SEMI-IMPLICIT SPECTRAL COMPUTATIONS AND PREDICTOR-CORRECTOR SCHEMES IN THE CYCLE 43 OF ARPEGE/IFS.

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

February 23, 2016

Abstract:

This documentation describes two types of schemes allowing to stabilize the numerical discretizations of the models: the semi-implicit scheme and the iterative centred-implicit scheme (sometimes called predictor-corrector scheme). An algorithmic description to different types of equations, and some technical information (organigramme) are provided.

Résumé:

Cette documentation décrit deux types de schémas permettant de stabiliser les discrétisations numériques des modèles: le schéma semi-implicite et le schéma ICI (centré implicite itératif) encore appelé prédicteur-correcteur. On fournit une application de ces algorithmes à différents jeux d'équations, ainsi que des informations techniques (organigramme).

Contents

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 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).

1.3   The different models described.

- 2D shallow-water model.
- 3D primitive equations model (denoted as HYD).
- 3D NH model with $\hat{Q}$ and d or d_4 NH prognostic variables (denoted as NH-PDVD).
- 3D NH model with Φ and gw NH prognostic variables (denoted as NH-GEOGW).

1.4   Other restrictions of this documentation.

- In the non-hydrostatic model with $\hat{Q}$ and d or d_4 NH prognostic variables, only the option with $\hat{Q}$ and d as prognostic variables (options **NPDVAR**=2, **NVDVAR**=3) is currently described. The choice of d_4 as prognostic variable **NVDVAR**=4 does not change the expression of linear terms.

- 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.
  For the (Wood and Staniforth, 2003) NH deep-layer system, hydrostatic pressures (Π,Π_s) must be replaced by mass-integrated coordinates ($\tilde{\Pi},\tilde{\Pi}_s$) in the linear system.

1.5   Modifications since cycle 42.

No significant modification.

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_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, ζ' is the reduced vorticity. ζ and ζ' 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. 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.

- q is the humidity.

- Π is the hydrostatic pressure, Π_s is the hydrostatic surface pressure. Π^* is a reference hydrostatic pressure and Π_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 Π_s^* is generally between 800 hPa and 1000 hPa. $\Delta\Pi^*$ are layer depths corresponding to a surface hydrostatic pressure equal to Π_s^*.

- Π_{sst} 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_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), Φ_s is the surface geopotential (i.e. the orography). In the thin layer equations, $\Phi = gz$. Φ_s is assumed to be always equal to gz_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_{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. ∇' 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:

$$D_3 = \nabla \mathbf{V} + \frac{p}{\frac{\partial \Pi}{\partial \eta} RT} \nabla \Phi \left(\frac{\partial \mathbf{V}}{\partial \eta} \right) - \frac{gp}{\frac{\partial \Pi}{\partial \eta} RT} \left(\frac{\partial w}{\partial \eta} \right) \tag{2}$$

- 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, ∂^*, $\mathbf{L}^*$, $\mathbf{G}^*$, $\mathbf{S}^*$, $\mathbf{N}^*$, $\mathbf{Q}^*$ 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:

$$H = \frac{R_{\mathrm{d}} T^*}{g} \tag{4}$$

$$C = \sqrt{R_{\mathrm{d}} T^* \frac{c_{\mathrm{pd}}}{c_{\mathrm{vd}}}} \tag{5}$$

$$N = \frac{g}{\sqrt{c_{\mathrm{pd}} T^*}} \tag{6}$$

- For a variable X defined at full levels, $\langle X \rangle$ is the vector of coordinates $(X_1; ...; X_l; ...; X_L)$.

- $\mathcal{R}_{\mathrm{inte}}$ is the vertical integration operator used in the case **LVERTFE**=.T. :

  - $\int_{\eta=0}^{\eta=1} X d\eta$ is discretised by $[\mathcal{R}_{\mathrm{inte}}]_{(top,surf)} \langle X \rangle$.
  - $\int_{\eta=0}^{\eta=\eta_l} X d\eta$ is discretised by $[\mathcal{R}_{\mathrm{inte}}]_{(top,l)} \langle X \rangle$.
  - $\int_{\eta=\eta_l}^{\eta=1} X d\eta$ is discretised by $[\mathcal{R}_{\mathrm{inte}}]_{(l,surf)} \langle X \rangle$.

- $\mathcal{R}_{\mathrm{deri}}$ 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.

3.3.1 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.

3.3.2 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 $X^{t+\Delta t}_{(i=N_{\text{siter}})}$.

- 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 recommended for the non-hydrostatic model with a SL2TL scheme to ensure stability.

- For unlagged physics, the physics has to be computed for the iteration $(i = 0)$ only. For lagged physics, iterations 0 to $N_{\text{siter}} - 1$ are adiabatic ones, iteration N_{siter} is diabatic one.

∗ **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 **LNESC**=.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 **LNESC**=.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:

  $$(X^{t+\Delta t}_{(i=0)} - 0.5\Delta t\beta\mathcal{B}^{t+\Delta t}_{(i=0)})_F = [0.5\Delta t\mathcal{A}^t - 0.5\Delta t\beta\mathcal{B}^t]_F + [X^t + 0.5\Delta t\mathcal{A}^t + \Delta t\mathcal{F}^t]_{O(i=0)}$$

7

- 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   Limited area models (ALADIN, AROME).

Particular features for LAM models are not described in detail, only brief comments are mentioned. Concerning the semi-implicit and iterative centred-implicit algorithms one can consider that the major part of this documentation is still valid for LAM models; the main differences with ARPEGE/IFS are:

- The LAM model shallow-water model is not coded.
- Option **LESIDG**=.T. replaces **LSIDG**=.T., 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 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**); the algorithm is the same as in global model 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 $\mathcal{R}_{\text{inte}}$) computed in the setup code under **SUVERTFE**; the vertical integration is done by routine **VERINT**.

- In NH models there are also vertical derivatives with a specific treatment. The NH-GEOGW model is coded only with VFE (for integrals and derivatives) because this is the only way to keep all the prognostic variables at full levels. The spectral part of the semi-implicit scheme is not designed to mix half level and full level variables.

- In the NH-PDVD model, it is possible for example:

  - to use VFE only in the explicit model, and to keep the finite differences vertical discretisation in the linear model (that allows to ensure constraints C1 and C2, see below).

  - to use VFE in both linear and non linear terms, at least for LGWADV=T.

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 $\mathbf{V}$, temperature T, and the two additional non-hydrostatic variables in a non-hydrostatic model. Details about equations of the NH variables in the NH-PDVD model (choice of the two additional prognostic variables, discretisations, linearisation for semi-implicit scheme) can be found for example in (IDNHPB).

- 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 $\mathbf{V}$, and the equivalent height $\Phi - \Phi_{\mathrm{s}}$ (continuity equation).

4.2 Quantities used for vertical discretisations and linear operators.

The following quantities and operators may have different discretisations according to the fact that finite differences (FD) or finite elements (VFE) are used for vertical discretisations in the model. The following abbreviations will be used:

- FD0: finite differences (**LVERTFE**=.F.) using **NDLNPR**=0.
- FD1: finite differences (**LVERTFE**=.F.) using **NDLNPR**=1.
- FD: finite differences (**LVERTFE**=.F.), any value for **NDLNPR**.
- VFE: vertical finite elements (**LVERTFE**=.T.).

4.2.1 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**.

- FD0:
  - For 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 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.

- FD1:
  - For 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:

$$\alpha_l^* = 1 - \sqrt{\frac{\Pi_{\bar{l}-1}^*}{\Pi_{\bar{l}}^*}} = 1 - \frac{\Pi_l^*}{\Pi_{\bar{l}}^*} \tag{21}$$

10

$$\delta_l^* = \frac{\Delta\Pi_l^*}{\Pi_l^*} = \frac{\Delta\Pi_l^*}{\sqrt{\Pi_{\overline{l-1}}^*\Pi_{\overline{l}}^*}} \tag{22}$$

Π^* is discretised as follows at full levels:

$$\Pi^*{}_l = \sqrt{\Pi^*{}_{\overline{l}}\Pi^*{}_{\overline{l-1}}} \tag{23}$$

Notation $\alpha_{\overline{l-1}}^*$ is used for quantity $1 - \frac{\Pi_{\overline{l-1}}^*}{\Pi_l^*}$ instead of notation β_l^* of (Bubnová et al., 1995).

   – For the layer $l = 1$ if the pressure at the top of the model is zero:

        ∗ $\alpha_{l=1}^* = 1$ and $\alpha_{l=0}^* = 1$.

        ∗ $\delta_{l=1}^* = 1 + c_{\mathrm{pd}}/R_{\mathrm{d}}$.

        ∗ $\Pi_{l=1}^* = \Delta\Pi_{l=1}^*/\delta_{l=1}^*$.

- VFE: for a layer l between 1 and L, α^* and δ^* are discretised as follows at full levels:

$$\alpha_l^* = \frac{\Pi_{\overline{l}}^* - \Pi_l^*}{\Pi_l^*} \tag{24}$$

$$\delta_l^* = \frac{\Delta\Pi_l^*}{\Pi_l^*} \tag{25}$$

where $\Pi_l^* = A_l + B_l\Pi_{\mathrm{s}}^*$. See documentation (IDEUL) for computation of A_l and B_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.

4.2.2   Vertical integrals and linear products.

These operators are used in both hydrostatic and non-hydrostatic models.

∗ **Linear operator "γ", and its dimensionless counterpart $\mathbf{G}^*$:** this operator is applied to temperature and pressure departure variable to compute linear term in momentum equation.

- For a variable Z, $(\mathbf{G}^*Z)$ is a discretisation of vertical integral: $\int_\eta^1 \frac{1}{\Pi^*}\frac{\partial\Pi^*}{\partial\eta}Z d\eta$

- $\gamma Z = R_{\mathrm{d}}(\mathbf{G}^*Z)$

- VFE expression of this discretisation is:

$$(\mathbf{G}^*Z)_l = [\mathcal{R}_{\mathrm{inte}}]_{(l,surf)}\left\langle\frac{Z\delta^*}{\Delta\eta}\right\rangle \tag{26}$$

- FD expression of this discretisation is:

$$(\mathbf{G}^*Z)_l = \alpha_l^* Z_l + \sum_{k=l+1}^{L} Z_k \delta_k^* \tag{27}$$

Remark: if **LSPRT**=.T., $\mathbf{G}^*$ is applied to virtual temperature instead of real temperature.

∗ **Linear operator "τ", and its dimensionless counterpart $\mathbf{S}^*$:** this operator is applied to divergence to compute linear term in temperature equation.

- For a variable Z, $(\mathbf{S}^*Z)$ is a discretisation of vertical integral: $\frac{1}{\Pi^*}\int_0^\eta \frac{\partial\Pi^*}{\partial\eta}Z d\eta$

- $\tau Z = [(R_{\mathrm{d}}T^*)/c_{\mathrm{pd}}](\mathbf{S}^*Z)$ if **LSPRT**=.F.

- $\tau Z = [(R_{\mathrm{d}}^2 T^*)/(Rc_{\mathrm{pd}})](\mathbf{S}^*Z)$ if **LSPRT**=.T. (equivalent to apply a linear operator in a system of equations using virtual temperature).

- VFE expression of this discretisation is:

$$(\mathbf{S}^*Z)_l = \frac{\delta_l^*}{\Delta\Pi_l^*}[\mathcal{R}_{\mathrm{inte}}]_{(top,l)}\left\langle\frac{\Delta\Pi^*Z}{\Delta\eta}\right\rangle \tag{28}$$

Remark: according to the expression of δ_l^* in this case, this equation can be rewritten:

$$(\mathbf{S}^*Z)_l = \frac{1}{\Pi_l^*}[\mathcal{R}_{\mathrm{inte}}]_{(top,l)}\left\langle\frac{\Delta\Pi^*Z}{\Delta\eta}\right\rangle$$

- FD expression of this discretisation is:

$$(\mathbf{S}^*Z)_l = \left[\alpha_l^* Z_l + \frac{\delta_l^*}{\Delta\Pi_l^*}\sum_{k=1}^{l-1}\Delta\Pi_k^* Z_k\right] \tag{29}$$

11

∗ **Linear operator "ν" , and its dimensionless counterpart N*:** this operator is applied to divergence to compute linear term in continuity equation.

- For a variable Z, $(\mathbf{N}^*Z)$ is a discretisation of vertical integral: $\frac{1}{\Pi^*}\int_0^1 \frac{\partial \Pi^*}{\partial \eta} Z d\eta$

- $\nu Z = \mathbf{N}^*Z$ (because use of $\log \Pi_{\mathrm{s}}$).

- VFE expression of this discretisation is:

$$(\mathbf{N}^*Z) = \frac{1}{\Pi_{\mathrm{s}}^*}[\mathcal{R}_{\mathrm{inte}}]_{(top,surf)}\left\langle \frac{\Delta\Pi^*Z}{\Delta\eta}\right\rangle \tag{30}$$

- FD expression of this discretisation is:

$$(\mathbf{N}^*Z) = \frac{1}{\Pi_{\mathrm{s}}^*}\sum_{l=1}^{L}\Delta\Pi_l^* Z_l \tag{31}$$

∗ **Linear operator "μ", and its dimensionless counterpart I:** this operator is applied to $\log(\Pi_{\mathrm{s}})$ to compute linear term in momentum equation.

- I is the identity matrix: $\mathtt{I}Z = Z$.

- $(\mu Z) = R_{\mathrm{d}}T^*Z$ if **LSPRT**=.F.

- $(\mu Z) = (R_{\mathrm{d}}^2/R)T^*Z$ if **LSPRT**=.T. (equivalent to apply a linear operator in a system of equations using virtual temperature).

∗ **Relationship between these vertical operators and definition of constraint C1:**

Continuous equations ensure the following identity:

$$\mathbf{G}^*\mathbf{S}^* - \mathbf{S}^* - \mathbf{G}^* + \mathbf{N}^* = 0 \tag{32}$$

According to the vertical discretisation used, this identity may or may not be matched by the discretised operators. Definition of "constraint C1" is: the discretised operators match equation (32).

- FD0 and VFE: constraint C1 is not ensured.

- FD1: constraint C1 is ensured.

In the NH-PDVD model, constraint C1 SHOULD be ensured if one wants the complete elimination of variables in order to provide a "one variable" Helmholtz equation computing the vertical divergence. This is why the FD1 is preferred to FD0 when using FD discretisation. For VFE, Helmholtz equation treatment requires adaptations to take account of the fact that constraint C1 is not ensured.

When constraint C1 is not ensured, we introduce the dimensionless quantity COR:

$$COR = \frac{c_{\mathrm{vd}}}{R_{\mathrm{d}}^2 T^*}\gamma\tau - \frac{c_{\mathrm{vd}}}{R_{\mathrm{d}}c_{\mathrm{pd}}}\gamma - \frac{c_{\mathrm{vd}}}{R_{\mathrm{d}}T^*}\tau + \frac{c_{\mathrm{vd}}}{c_{\mathrm{pd}}}\nu = \frac{c_{\mathrm{vd}}}{c_{\mathrm{pd}}}[\mathbf{G}^*\mathbf{S}^* - \mathbf{S}^* - \mathbf{G}^* + \mathbf{N}^*] \tag{33}$$

4.2.3 Vertical derivatives and mixed operators (NH models only).

These operators are used in non-hydrostatic models only.

∗ **Linear operator "∂^*":** this operator is the following first-order derivative: $\partial^*Z = \frac{\partial Z}{\partial \log \Pi^*}$.
This operator is currently only used in the VFE NH-GEOGW scheme, discretisation involves the operator $[\mathcal{R}_{\mathrm{deri}}]$. Actually ∂^* and ∂^*+1 appear, and we cannot exclude a priori the possibility to use two slightly different first order derivative operators, in order to ensure good properties for the product $\partial^*(\partial^*+1)$ (see the following paragraph).

∗ **Linear operator "$\mathbf{L}^*$":** this operator, called Laplacian operator, is applied to the pressure departure variable to compute linear term in the vertical divergence equation (NH-PDVD model). It also appears in the NH-GEOGW scheme.

- For a variable Z, $(\mathbf{L}^*Z)$ is the vertical double derivative $\left[\Pi^*\frac{\partial}{\partial\Pi^*}(\frac{\partial\Pi^*Z}{\partial\Pi^*})\right]$; this is equivalent to write $\mathbf{L}^* = \partial^*(\partial^*+1)$.

- FD expression of this discretisation is:

– Layers 2 to $L-1$:
$$(\mathbf{L}^*Z)_l = \mathbf{A}_l^* Z_{l-1} + \mathbf{B}_l^* Z_l + \mathbf{C}_l^* Z_{l+1} \tag{34}$$

Expressions of $\mathbf{A}^*$, $\mathbf{B}^*$ and $\mathbf{C}^*$ are:

$$\mathbf{A}_l^* = \frac{\Pi_{l-1}^*}{\delta_l^*(\Pi_l^* - \Pi_{l-1}^*)} \tag{35}$$

$$\mathbf{B}_l^* = -\frac{1}{\delta_l^*}\left(\frac{\Pi_l^*}{\Pi_l^* - \Pi_{l-1}^*} + \frac{\Pi_l^*}{\Pi_{l+1}^* - \Pi_l^*}\right) \tag{36}$$

$$\mathbf{C}_l^* = \frac{\Pi_{l+1}^*}{\delta_l^*(\Pi_{l+1}^* - \Pi_l^*)} \tag{37}$$

Note that equation (34) can be rewritten:

$$(\mathbf{L}^*Z)_l = \mathbf{A}_l^*(Z_{l-1} - Z_l) + \mathbf{C}_l^*(Z_{l+1} - Z_l) \tag{38}$$

– Layer 1: The quantity Z to which is applied $\mathbf{L}^*$ is assumed to be zero at the top of the model, that means that the term $\mathbf{A}_1^*(Z_0 - Z_1)$ has to be replaced by $-\mathbf{A}_1^* Z_1$. In practical, $\mathbf{A}_1^*$ has to be set to zero if the top hydrostatic pressure is zero, and the general formula valid for any non-zero top pressure is:

$$\mathbf{A}_1^* = \frac{\Pi_{\text{top}}^*}{\delta_1^*(\Pi_1^* - \Pi_{\text{top}}^*)} \tag{39}$$

$\mathbf{C}_1^*$ matches the general expression:

$$\mathbf{C}_1^* = \frac{\Pi_2^*}{\delta_1^*(\Pi_2^* - \Pi_1^*)} \tag{40}$$

and, if the top hydrostatic pressure is zero:

$$\mathbf{B}_1^* = -\mathbf{C}_1^* \tag{41}$$

This upper condition is stable at least when the top hydrostatic pressure is zero.
That leads to the following formula for $(\mathbf{L}^*Z)_1$:

$$(\mathbf{L}^*Z)_1 = -\mathbf{A}_1^* Z_1 + \mathbf{C}_1^*(Z_2 - Z_1) \tag{42}$$

– Layer L: The quantity Z to which is applied $\mathbf{L}^*$ is assumed to be constant below the full level $l = L$, that means that the term $\mathbf{C}_L^*(Z_{L+1} - Z_L)$ has to be replaced by 0. In practical, $\mathbf{C}_L^*$ is set to zero. $\mathbf{A}_L^*$ matches the general expression:

$$\mathbf{A}_L^* = \frac{\Pi_{L-1}^*}{\delta_L^*(\Pi_L^* - \Pi_{L-1}^*)} \tag{43}$$

Application of formula (38) would lead to $\mathbf{B}_L^* = -\mathbf{A}_L^*$ but we actually use the formula (34) with a slightly different expression for $\mathbf{B}_L^*$:

$$\mathbf{B}_L^* = -\frac{\Pi_L^*}{\delta_L^*(\Pi_L^* - \Pi_{L-1}^*)} \tag{44}$$

That leads to the following formula for $(\mathbf{L}^*Z)_L$:

$$(\mathbf{L}^*Z)_L = \mathbf{A}_L^*(Z_{L-1} - Z_L) - \mathbf{A}_L^*\left(\frac{\Pi_L^*}{\Pi_{L-1}^*} - 1\right)Z_L \tag{45}$$

- VFE expression of this discretisation in the NH-PDVD model: this topic is still in progress, at least two solutions have been implemented:

  – one option uses a second-order derivative operator $[\mathcal{R}_{\text{dderi}}]$ (case **LVFE_LAPL_HALF**=F).
  – one option mixes a first-order VFE derivative operator $[\mathcal{R}_{\text{deri}}]$ and a first-order FD derivative operator (case **LVFE_LAPL_HALF**=T).

- VFE expression of this discretisation in the NH-GEOGW model: it is currently computed as the product of two first-order derivatives ($\mathbf{L}^* = \partial^*(\partial^* + 1)$).

- Remarks for NH-PDVD model:

  – stability is ensured only if the top hydrostatic pressure is zero.
  – FD1 expressions have been chosen in order to match "constraint C2" (definition of "constraint C2" will be given below, in paragraph explaining linear operator $\mathbf{T}^*$).

13

∗ Linear operator "T∗" for NH-PDVD model, and definition of "constraint C2":

General definition of $\mathbf{T}^*$ uses a combination of linear operators $\mathbf{L}^*$, τ and γ.

Definition of $\mathbf{T}^*$ writes:

$$\mathbf{T}^* = \frac{g^2 \mathbf{L}^* \left(\frac{c_{\mathrm{p_d}}}{R_{\mathrm{d}} T^*} \tau\gamma - \frac{c_{\mathrm{p_d}}^2}{c_{\mathrm{v_d}} T^*} \tau - \frac{c_{\mathrm{p_d}}}{c_{\mathrm{v_d}}} \gamma \right)}{R_{\mathrm{d}} N^2 C^2} \tag{46}$$

Using definitions for C and H (see equations (5) and (4)) this equation can be rewritten:

$$\mathbf{T}^* = \frac{\mathbf{L}^* \left(T^* c_{\mathrm{p_d}} \tau\gamma - c_{\mathrm{p_d}} C^2 \tau - C^2 T^* \gamma \right)}{H^2 N^2 C^2} \tag{47}$$

Using $\mathbf{G}^*$ and $\mathbf{S}^*$ this equation can be rewritten:

$$\mathbf{T}^* = \frac{g^2 \mathbf{L}^* \left(\mathbf{S}^* \mathbf{G}^* - (c_{\mathrm{p_d}}/c_{\mathrm{v_d}}) \mathbf{S}^* - (c_{\mathrm{p_d}}/c_{\mathrm{v_d}}) \mathbf{G}^* \right)}{N^2 C^2} \tag{48}$$

We can now introduce definition of "constraint C2":

- Strong constraint C2: $\mathbf{T}^*$ is the identity matrix.
- Weak constraint C2: $\mathbf{T}^*$ matches (49):

$$\mathbf{T}^* = (I + \mathbf{L}^* \mathbf{Q}^*) \tag{49}$$

  and does not depart too much from identity matrix. $\mathbf{Q}^*$ is a diagonal matrix equal to $\delta^* - 2\alpha^*$.

Continuous equations ensure "strong" constraint C2. In some cases this is possible to match "weak" constraint C2 in discretised equations, and from now "constraint C2" will mean "weak constraint C2".

According to the vertical discretisation used, "constraint C2" may or may not be matched by the discretised operators.

- FD0 and VFE: constraint C2 is not ensured.
- FD1: constraint C2 is ensured.
- We can notice that either C1 and C2 are both ensured, or none of C1 and C2 are ensured.
- Elimination between equations to obtain Helmholtz equation does not assume that constraint C2 is ensured (use of formula (46)).
- Discretisations of $\mathbf{T}^*$ matching constraint C2 or close to constraint C2 are preferred, in order to ensure numerical stability.

Discretisation:

- FD1: constraint C2 is ensured, and formula (49) is used. In this case, $\mathbf{T}^*$ is a tri-diagonal operator, like $\mathbf{L}^*$. Expression of the elements of the associated matrix is:

$$\mathbf{T}^*_{(l,l)} = 1 - \frac{1}{\delta_l^*} \left(\frac{\Pi_l^*}{\Pi_l^* - \Pi_{l-1}^*} + \frac{\Pi_l^*}{\Pi_{l+1}^* - \Pi_l^*} \right) (\delta_l^* - 2\alpha_l^*) \tag{50}$$

$$\mathbf{T}^*_{(1,1)} = 1 \tag{51}$$

$$\mathbf{T}^*_{(l,l-1)} = \frac{1}{\delta_l^*} \left(\frac{\Pi_{l-1}^*}{\Pi_l^* - \Pi_{l-1}^*} \right) (\delta_{l-1}^* - 2\alpha_{l-1}^*) \tag{52}$$

$$\mathbf{T}^*_{(l,l+1)} = \frac{1}{\delta_l^*} \left(\frac{\Pi_{l+1}^*}{\Pi_{l+1}^* - \Pi_l^*} \right) (\delta_{l+1}^* - 2\alpha_{l+1}^*) \tag{53}$$

  with some particular expressions at the top and the bottom.
  Discretisation of diagonal matrix $\mathbf{Q}^*$ is: $\mathbf{Q}^*_{(l,l)} = (\delta_l^* - 2\alpha_l^*)$, and if the top hydrostatic pressure is zero, $\mathbf{Q}^*_{(1,1)} = 0$.
- FD0 and VFE: constraint C2 is not ensured. Formula (46) is used (use VFE or FD0 discretisations of γ, τ, $\mathbf{L}^*$).

14

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 ($X = \log(\Pi_\mathrm{s})$):

$$\mathcal{B} = -\nu(\overline{M}^2 D^{'}) \tag{54}$$

- Divergence equation ($X = D^{'}$):

$$\mathcal{B} = -\nabla^{'2}(\gamma T + \mu \log \Pi_\mathrm{s}) \tag{55}$$

- Vorticity equation ($X = \zeta^{'}$):

$$\mathcal{B} = 0 \tag{56}$$

- Temperature equation ($X = T$):

$$\mathcal{B} = -\tau(\overline{M}^2 D^{'}) \tag{57}$$

∗ **System to be solved:** Equations are written for $\log(\Pi_\mathrm{s})$ as a prognostic variable for continuity equation.

$$\log(\Pi_\mathrm{s})_{t+\Delta t} + \beta\Delta t \overline{M}^2 \nu D^{'}_{t+\Delta t} = \mathcal{P}^* \tag{58}$$

$$D^{'}_{t+\Delta t} + \beta\Delta t \nabla^{'2}(\gamma T_{t+\Delta t} + \mu \log(\Pi_\mathrm{s})_{t+\Delta t}) = \mathcal{D}^{'*} \tag{59}$$

$$T_{t+\Delta t} + \beta\Delta t \overline{M}^2 \tau D^{'}_{t+\Delta t} = \mathcal{T}^* \tag{60}$$

$\mathcal{P}^*$, $\mathcal{D}^{'*}$, $\mathcal{T}^*$ correspond to $\mathcal{X}^*$ defined in equation (17) and are available in spectral arrays (**SPA3%SP**, **SPA3%DIV**, **SPA3%T**) at the beginning of the spectral computations. Equations (58) to (60) yield (61) (Helmholtz equation):

$$(\mathtt{I} - \beta^2\Delta t^2 \; \mathtt{B} \; \nabla^{'2}\overline{M}^2)D^{'}_{t+\Delta t} = \mathcal{D}^{'*} - \beta\Delta t\nabla^{'2}(\gamma\mathcal{T}^* + \mu\mathcal{P}^*) \tag{61}$$

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 (61) as:

$$(\nabla^{'-2} - \beta^2\Delta t^2 \; \mathtt{B} \; M^2)D^{'}_{t+\Delta t} = \nabla^{'-2}\mathcal{D}^{'*} - \beta\Delta t(\gamma\mathcal{T}^* + \mu\mathcal{P}^*) \tag{62}$$

which shows a symmetric matricial operator in the left hand side.

∗ **Spectral computations to solve system of equations (58) to (60).** Algorithm works zonal wave number by zonal wave number m ($\mid m \mid$ varies between 0 and the truncation N_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 (61) is computed for all total wave numbers n between m and N_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 (61) becomes, for each eigenmode l:

$$(\mathtt{I} - \beta^2\Delta t^2 a_l \nabla^{'2}\overline{M}^2)\mathtt{Q}D^{'}_{t+\Delta t} = \mathtt{Q}(\mathcal{D}^{'*} - \beta\Delta t\nabla^{'2}(\gamma\mathcal{T}^* + \mu\mathcal{P}^*)) \tag{63}$$

  - 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 $(N_\mathrm{s} + 1 - \mid m \mid) * (N_\mathrm{s} + 1 - \mid m \mid)$: spectral coefficients of the right-hand side member of (63) are simply divided by the diagonal coefficients of this matrix. The result is then multiplied by $\mathtt{Q}$.

15

- Inversion of Helmholtz equation for the case "unreduced divergence" (case **LSIDG**=.T. or **LESIDG**=.T.): 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 B is used: $B = Q^{-1} A Q$, where A is a diagonal $L * L$ matrix, the diagonal coefficients a_l of which are stored in the array **SIVP**. Q is a $L * L$ matrix stored in the array **SIMI**, Q^{-1} is stored in the array **SIMO**. Note that the vertical operators ν, μ, τ, γ, B , Q commute with the horizontal operators $\nabla^{'2}$ and M^2.

  - Helmholtz equation (61) (resp. (62)) becomes, for each eigenmode l:

  $$(I - \beta^2 \Delta t^2 a_l \nabla^{'2} M^2) Q D'_{t+\Delta t} = Q(\mathcal{D}'^* - \beta \Delta t \nabla^{'2}(\gamma \mathcal{T}^* + \mu \mathcal{P}^*)) \tag{64}$$

  resp.:

  $$(\nabla^{'-2} - \beta^2 \Delta t^2 a_l M^2) Q D'_{t+\Delta t} = Q(\nabla^{'-2} \mathcal{D}'^* - \beta \Delta t(\gamma \mathcal{T}^* + \mu \mathcal{P}^*)) \tag{65}$$

  - 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 $(N_s + 1 - |m|) * (N_s + 1 - |m|)$. 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 (65) 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 $Q D'_{t+\Delta t}$. Multiplying by Q^{-1} one obtains $D'_{t+\Delta t}$.

  - For the zonal wave number $m = 0$ equation (64) is used rather than (65) 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 $(I - \beta^2 \Delta t^2 a_l \nabla^{'2} M^2)$ 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 (58) provides $\log(\Pi_s)_{t+\Delta t}$ and equation (60) provides $T_{t+\Delta t}$. For the case **LSIDG**=.T. only (resp. **LESIDG**=.T. in LAM models), spectral multiplications by M^2 are performed by the product of a symmetric pentadiagonal matrix of dimensions $(N_s + 1 - |m|) * (N_s + 1 - |m|)$ (useful coefficients computed in routine **SUSMAP** (resp. **SUESMAP** in LAM models) and stored in the array **SCGMAP** (resp. **ESCGMAP** in LAM models)) by a vector containing spectral coefficients (m, n) for n varying from $|m|$ to N_s.

- Semi-implicit scheme ends by a final memory transfer.

5.2   3D NH-PDVD model.

The code described is valid for options **NPDVAR**=2 and **NVDVAR**=3. For **NVDVAR**=4 the linear terms are the same as for **NVDVAR**=3.

∗ **Expression of the linear term $\mathcal{B}$ for GMV and GMVS variables:**

- Continuity equation ($X = \log(\Pi_s)$):

$$\mathcal{B} = -\nu(\overline{M}^2 D') \tag{66}$$

- Divergence equation ($X = D'$):

$$\mathcal{B} = -\nabla^{'2}[\gamma T - T^*(\gamma \hat{Q}) + \mu \log(\Pi_s) + R_d T^* \hat{Q}] \tag{67}$$

- Vorticity equation ($X = \zeta'$):

$$\mathcal{B} = 0 \tag{68}$$

- Temperature equation ($X = T$):

$$\mathcal{B} = -\frac{R_d T^*}{c_{vd}}[\overline{M}^2 D' + d] \tag{69}$$

- Pressure departure variable equation ($X = \hat{Q}$):

$$\mathcal{B} = -\left[\frac{c_{pd}}{c_{vd}}(\overline{M}^2 D' + d) - \frac{c_{pd}}{R_d T^*}\tau(\overline{M}^2 D')\right] \tag{70}$$

16

- Vertical divergence equation ($X = d$):

$$\mathcal{B} = -\frac{g^2}{R_{\rm d}T_{\rm a}^*}(\mathbf{L}^*\hat{Q}) \tag{71}$$

∗ **System to be solved:** Equations are written for $\log(\Pi_{\rm s})$ as a prognostic variable for continuity equation.

$$\log(\Pi_{\rm s})_{t+\Delta t} + \beta\Delta t\nu(\overline{M}^2 D_{t+\Delta t}^{'}) = \mathcal{P}^* \tag{72}$$

$$D_{t+\Delta t}^{'} + \beta\Delta t\nabla^{'2}[\gamma T_{t+\Delta t} - T^*(\gamma\hat{Q}_{t+\Delta t}) + \mu\log(\Pi_{\rm s})_{t+\Delta t} + R_{\rm d}T^*\hat{Q}_{t+\Delta t}] = \mathcal{D}^{'*} \tag{73}$$

$$\hat{Q}_{t+\Delta t} + \beta\Delta t\left[\frac{c_{\rm pd}}{c_{\rm vd}}(\overline{M}^2 D_{t+\Delta t}^{'} + d_{t+\Delta t}) - \frac{c_{\rm pd}}{R_{\rm d}T^*}\tau(\overline{M}^2 D_{t+\Delta t}^{'})\right] = \hat{Q}^* \tag{74}$$

$$d_{t+\Delta t} + \beta\Delta t\frac{g^2}{R_{\rm d}T_{\rm a}^*}(\mathbf{L}^*\hat{Q}_{t+\Delta t}) = \hat{D}^* \tag{75}$$

$$T_{t+\Delta t} + \beta\Delta t\frac{R_{\rm d}T^*}{c_{\rm vd}}[\overline{M}^2 D_{t+\Delta t}^{'} + d_{t+\Delta t}] = \mathcal{T}^* \tag{76}$$

$\mathcal{P}^*$, $\mathcal{D}^{'*}$, $\hat{Q}^*$, $\hat{D}^*$, $\mathcal{T}^*$ correspond to $\mathcal{X}^*$ defined in equation (17) and are available in spectral arrays (**SPA3%SP**, **SPA3%DIV**, **SPA3%SPD**, **SPA3%SVD**, **SPA3%T**) at the beginning of spectral computations.

∗ **Elimination of variables:** The following calculations are valid for both $\overline{M}$ horizontally constant and $\overline{M}$ horizontally variable. All the constants and the vertical operators commute with $\nabla^{'2}$ and $\overline{M}$.

- Elimination of T, $\hat{Q}$ and $\log(\Pi_{\rm s})$ between equations (72), (73), (74) and (76) leads to equation (77):

$$D_{t+\Delta t}^{'} - (\beta\Delta t)^2\nabla^{'2}[\{R_{\rm d}T^*(\frac{\gamma}{R_{\rm d}} - 1)(\frac{c_{\rm pd}}{R_{\rm d}T^*}\tau - \frac{c_{\rm pd}}{c_{\rm vd}}) + \frac{R_{\rm d}T^*}{c_{\rm vd}}\gamma + R_{\rm d}T^*\nu\}\overline{M}^2 D_{t+\Delta t}^{'}$$

$$+\{-R_{\rm d}T^*\frac{c_{\rm pd}}{c_{\rm vd}}(\frac{\gamma}{R_{\rm d}} - 1) + \frac{R_{\rm d}T^*}{c_{\rm vd}}\gamma\}d_{t+\Delta t}]$$

$$= \mathcal{D}^{'*} + \beta\Delta t\nabla^{'2}[R_{\rm d}T^*(\frac{\gamma}{R_{\rm d}} - 1)\hat{Q}^* - \gamma\mathcal{T}^* - R_{\rm d}T^*\mathcal{P}^*] \tag{77}$$

$\mathcal{D}^{'**}$ is defined by equation (78):

$$\mathcal{D}^{'**} = \mathcal{D}^{'*} + \beta\Delta t\nabla^{'2}\left[R_{\rm d}T^*(\frac{\gamma}{R_{\rm d}} - 1)\hat{Q}^* - \gamma\mathcal{T}^* - R_{\rm d}T^*\mathcal{P}^*\right] \tag{78}$$

We use the relationship $c_{\rm pd} - c_{\rm vd} = R_{\rm d}$, the definition of C and we isolate the term COR (see equation (33)): this equation can be rewritten:

$$[-(\beta\Delta t)^2\nabla^{'2}(C^2 - T^*\gamma)]d_{t+\Delta t} + [\mathtt{I} - (\beta\Delta t)^2\nabla^{'2}(C^2(1 + COR))\overline{M}^2]D_{t+\Delta t}^{'} = \mathcal{D}^{'**} \tag{79}$$

Quantity COR is zero when the constraint "C1" is fulfilled. This is the case for the finite-difference vertical discretisation with **NDLNPR**=1. In the other cases, COR is generally non-zero but weak ($COR << 1$).

- Elimination of $\hat{Q}$ between equations (74) and (75) leads to equation (80):

$$d_{t+\Delta t} - (\beta\Delta t)^2\frac{T^*}{T_{\rm a}^*}\left[\frac{\mathbf{L}^*}{H^2}(-c_{\rm pd}\tau + C^2)\overline{M}^2 D_{t+\Delta t}^{'} + \frac{C^2}{H^2}\mathbf{L}^*d_{t+\Delta t}\right] = \hat{D}^{**} \tag{80}$$

where $\hat{D}^{**}$ is defined by:

$$\hat{D}^{**} = \hat{D}^* + \beta\Delta t\left[-\frac{g}{H}\frac{T^*}{T_{\rm a}^*}\mathbf{L}^*\hat{Q}^*\right] \tag{81}$$

- When $COR = 0$, elimination of $D^{'}$ between equations (79) and (80) leads to Helmholtz equation (84):

$$\left[\mathtt{I} - \beta^2\Delta t^2C^2(\overline{M}^2\nabla^{'2} + \frac{T^*}{T_{\rm a}^*}\frac{\mathbf{L}^*}{H^2}) - \beta^4\Delta t^4N^2C^2\overline{M}^2\nabla^{'2}\frac{T^*}{T_{\rm a}^*}\mathbf{T}^*\right]d_{t+\Delta t}$$

$$= (\mathtt{I} - \beta^2\Delta t^2C^2\overline{M}^2\nabla^{'2})\hat{D}^{**} + \beta^2\Delta t^2\frac{T^*}{T_{\rm a}^*}\frac{\mathbf{L}^*}{H^2}(-c_{\rm pd}\tau + C^2)\overline{M}^2\mathcal{D}^{'**} \tag{82}$$

which can be rewritten:

$$\left[\left(\mathtt{I} - \beta^2\Delta t^2C^2\frac{T^*}{T_{\rm a}^*}\frac{\mathbf{L}^*}{H^2}\right) - \left(\beta^2\Delta t^2C^2 + \beta^4\Delta t^4N^2C^2\frac{T^*}{T_{\rm a}^*}\mathbf{T}^*\right)\overline{M}^2\nabla^{'2}\right]d_{t+\Delta t}$$

$$= (\mathtt{I} - \beta^2\Delta t^2C^2\overline{M}^2\nabla^{'2})\hat{D}^{**} + \beta^2\Delta t^2\frac{T^*}{T_{\rm a}^*}\frac{\mathbf{L}^*}{H^2}(-c_{\rm pd}\tau + C^2)\overline{M}^2\mathcal{D}^{'**} \tag{83}$$

17

or:

$$\left[\mathtt{I} - \beta^2 \Delta t^2 C^2 \left(\mathtt{I} - \beta^2 \Delta t^2 C^2 \frac{T^*}{T_a^*} \frac{\mathbf{L}^*}{H^2} \right)^{-1} \left(\mathtt{I} + \beta^2 \Delta t^2 N^2 \frac{T^*}{T_a^*} \mathbf{T}^* \right) \overline{M}^2 \nabla'^2 \right] d_{t+\Delta t}$$

$$= \left(\mathtt{I} - \beta^2 \Delta t^2 C^2 \frac{T^*}{T_a^*} \frac{\mathbf{L}^*}{H^2} \right)^{-1} \left[(\mathtt{I} - \beta^2 \Delta t^2 C^2 \overline{M}^2 \nabla'^2) \hat{\mathcal{D}}^{**} + \beta^2 \Delta t^2 \frac{T^*}{T_a^*} \frac{\mathbf{L}^*}{H^2} (-c_{\mathrm{Pd}} \tau + C^2) \overline{M}^2 \mathcal{D}'^{**} \right] \quad (84)$$

- In the LHS of equation (84), we denote by $\mathtt{B}$ the following matrix, by analogy with the hydrostatic model:

$$\mathtt{B} = C^2 \left(\mathtt{I} - \beta^2 \Delta t^2 C^2 \frac{T^*}{T_a^*} \frac{\mathbf{L}^*}{H^2} \right)^{-1} \left(\mathtt{I} + \beta^2 \Delta t^2 N^2 \frac{T^*}{T_a^*} \mathbf{T}^* \right) \quad (85)$$

The LHS of equation (84) can be rewritten:

$$\left[\mathtt{I} - \beta^2 \Delta t^2 \; \mathtt{B} \; \overline{M}^2 \nabla'^2 \right]$$

If we compare with the LHS of the Helmholtz equation in the hydrostatic case we can notice three things:

- The order of $\overline{M}^2$ and ∇'^2 is inverted (these two operator do not commute if $\overline{M}$ is not constant).
- $\mathtt{B}$ now depends on Δt, it must be recomputed each time the timestep is changed.
- $\mathtt{B}$ is now a tridiagonal matrix (at least in the **LVERTFE**=.F. discretisation).

- When $\overline{M} = M$ (case **LSIDG**=.T. or **LESIDG**=.T.) it is more convenient to rewrite equation (84) as:

$$\left[\nabla'^{-2} - \beta^2 \Delta t^2 C^2 \left(\mathtt{I} - \beta^2 \Delta t^2 C^2 \frac{T^*}{T_a^*} \frac{\mathbf{L}^*}{H^2} \right)^{-1} \left(\mathtt{I} + \beta^2 \Delta t^2 N^2 \frac{T^*}{T_a^*} \mathbf{T}^* \right) \overline{M}^2 \right] [\nabla'^2 d_{t+\Delta t}]$$

$$= \left(\mathtt{I} - \beta^2 \Delta t^2 C^2 \frac{T^*}{T_a^*} \frac{\mathbf{L}^*}{H^2} \right)^{-1} \left[(\mathtt{I} - \beta^2 \Delta t^2 C^2 \overline{M}^2 \nabla'^2) \hat{\mathcal{D}}^{**} + \beta^2 \Delta t^2 \frac{T^*}{T_a^*} \frac{\mathbf{L}^*}{H^2} (-c_{\mathrm{Pd}} \tau + C^2) \overline{M}^2 \mathcal{D}'^{**} \right] \quad (86)$$

which shows a symmetric matricial operator in the left hand side.

$*$ **Spectral computations to solve system of equations (72) to (76).** The algorithm has some similarities with what is done in the hydrostatic model. Algorithm works zonal wave number by zonal wave number m ($| m |$ varies between 0 and the truncation N_s) and performed in the routine **SPNHSI** before all horizontal diffusion schemes. For a given zonal wave number m:

- After a preliminary memory transfer the right-hand side member of equation (84) is computed for all total wave numbers n between m and N_s. We can do some remarks:

  - During the elimination of variables between the equations (72) to (76) we have to perform matricial multiplications by $(\mathtt{I} - \beta^2 \Delta t^2 C^2 \overline{M}^2 \nabla'^2)$ and $\overline{M}^2$. When $\overline{M}$ is horizontally constant (case **LSIDG**=.F., **LESIDG**=.F.) these operators are purely diagonal; when $\overline{M} = M$ these operators are pentadiagonal and matricial products require additional calls to routine **MXPTMA**.
  - The matricial operator

$$\left(\mathtt{I} - \beta^2 \Delta t^2 C^2 \frac{T^*}{T_a^*} \frac{\mathbf{L}^*}{H^2} \right)$$

which appears in the RHS of (84) and also in $\mathtt{B}$ is pre-computed in routine **SUNHSI** and is stored in the array **SIFAC**. Its inverse is stored in the array **SIFACI**.

- Inversion of Helmholtz equation for the case "constant $\overline{M}$" (case **LSIDG**=.F., **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, $\mathtt{B}$, $\mathtt{Q}$, $\mathbf{L}^*$, $\mathbf{T}^*$ commute with the horizontal operator ∇'^2.
  - Helmholtz equation (61) becomes, for each eigenmode l:

$$(\mathtt{I} - \beta^2 \Delta t^2 a_l \overline{M}^2 \nabla'^2) \mathtt{Q} d_{t+\Delta t}$$

$$= \mathtt{Q} \left(\mathtt{I} - \beta^2 \Delta t^2 C^2 \frac{T^*}{T_a^*} \frac{\mathbf{L}^*}{H^2} \right)^{-1} \left[(\mathtt{I} - \beta^2 \Delta t^2 C^2 \overline{M}^2 \nabla'^2) \hat{\mathcal{D}}^{**} + \beta^2 \Delta t^2 \frac{T^*}{T_a^*} \frac{\mathbf{L}^*}{H^2} (-c_{\mathrm{Pd}} \tau + C^2) \overline{M}^2 \mathcal{D}'^{**} \right] (87)$$

- For each eigenmode l and each zonal wave number m: $(\mathtt{I} - \beta^2\Delta t^2 a_l\overline{M}^2\nabla^{'2})$ is a diagonal matricial operator $(N_s+1-\mid m\mid)*(N_s+1-\mid m\mid)$: spectral coefficients of the right-hand side member of (87) are simply divided by the diagonal coefficients of this matrix. The result is then multiplied by $\mathtt{Q}^{-1}$.

- Inversion of Helmholtz equation for the case "$\overline{M}=M$" (case **LSIDG**=.T. or **LESIDG**=.T.): Inversion of Helmholtz equation is more complicated than in the case of semi-implicit scheme with constant $\overline{M}$ because the left-hand side member of Helmholtz equation contains values of d 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, $\mathtt{B}$, $\mathtt{Q}$, $\mathbf{L}^*$, $\mathbf{T}^*$ commute with the horizontal operators $\nabla^{'2}$ and M^2.

  - Helmholtz equation (84) (resp. (86)) becomes, for each eigenmode l:

  $$(\mathtt{I}-\beta^2\Delta t^2 a_l\overline{M}^2\nabla^{'2})\mathtt{Q}d_{t+\Delta t}=\mathtt{Q}\left(\mathtt{I}-\beta^2\Delta t^2 C^2\tfrac{T^*}{T_a^*}\tfrac{\mathbf{L}^*}{H^2}\right)^{-1}$$
  $$\left[(\mathtt{I}-\beta^2\Delta t^2 C^2\overline{M}^2\nabla^{'2})\hat{\mathcal{D}}^{**}+\beta^2\Delta t^2\tfrac{T^*}{T_a^*}\tfrac{\mathbf{L}^*}{H^2}(-c_{\mathrm{p}\mathrm{d}}\tau+C^2)\overline{M}^2\mathcal{D}^{'**}\right] \tag{88}$$

  resp.:

  $$(\nabla^{'-2}-\beta^2\Delta t^2 a_l\overline{M}^2)\mathtt{Q}[\nabla^{'2}d_{t+\Delta t}]=\mathtt{Q}\left(\mathtt{I}-\beta^2\Delta t^2 C^2\tfrac{T^*}{T_a^*}\tfrac{\mathbf{L}^*}{H^2}\right)^{-1}$$
  $$\left[(\mathtt{I}-\beta^2\Delta t^2 C^2\overline{M}^2\nabla^{'2})\hat{\mathcal{D}}^{**}+\beta^2\Delta t^2\tfrac{T^*}{T_a^*}\tfrac{\mathbf{L}^*}{H^2}(-c_{\mathrm{p}\mathrm{d}}\tau+C^2)\overline{M}^2\mathcal{D}^{'**}\right] \tag{89}$$

  - 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 $(N_s+1-\mid m\mid)*(N_s+1-\mid m\mid)$. 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 the useful coefficients of $\mathtt{L}$, $\mathtt{U}$ are computed in the set-up routine **SUNHHEG** and stored in the array **SIHEG**.

  - The right-hand side member of (89) is computed, then multiplied by the inverse of the symmetric pentadiagonal operator $(\nabla^{'-2}-\beta^2\Delta t^2 a_l M^2)$ (resolution of two tridiagonal triangular systems by routine **MXTURS**). That yields $\mathtt{Q}[\nabla^{'2}d_{t+\Delta t}]$. Multiplying by $\mathtt{Q}^{-1}$ then $\nabla^{'-2}$ one obtains $d_{t+\Delta t}$.

  - For the zonal wave number $m=0$ equation (88) is used rather than (89) 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 $(\mathtt{I}-\beta^2\Delta t^2 a_l M^2\nabla^{'2})$ is factorised and inverted. All useful coefficients of $\mathtt{L}$, $\mathtt{U}$ are computed in the set-up routine **SUNHHEG** and stored in the arrays **SIHEG** and **SIHEG2**.

- Once known $d_{t+\Delta t}$ equation (79), which can be rewritten as follows:

$$D'_{t+\Delta t}=[\mathtt{I}-(\beta\Delta t)^2\nabla^{'2}C^2\overline{M}^2]^{-1}[\mathcal{D}^{'**}+(\beta\Delta t)^2\nabla^{'2}(-T^*\gamma+C^2)d_{t+\Delta t}] \tag{90}$$

yields $D'_{t+\Delta t}$. Note that when $\overline{M}$ is constant, $[\mathtt{I}-(\beta\Delta t)^2\nabla^{'2}C^2\overline{M}^2]$ is a purely diagonal operator, but when $\overline{M}=M$ $[\mathtt{I}-(\beta\Delta t)^2\nabla^{'2}C^2\overline{M}^2]$ is a pentadiagonal operator which has to be inverted by a LU method (all useful coefficients of $\mathtt{L}$, $\mathtt{U}$ are computed in the set-up routine **SUNHHEG** and are stored in **SIHEGB** and **SIHEGB2**) and that introduces additional calls to routines **MXTURS** or **MXTURE**. For $m>0$ we prefer to invert the symmetric operator $[\nabla^{'-2}-(\beta\Delta t)^2 C^2\overline{M}^2]$.

- After calculation of $\overline{M}^2 D'_{t+\Delta t}$ (which requires an additional call to routine **MXPTMA** if $\overline{M}=M$), equation (76) yields $T_{t+\Delta t}$, equation (74) yields $\hat{Q}_{t+\Delta t}$, and equation (72) yields $\log\Pi_{st+\Delta t}$. For the case **LSIDG**=.T. only (resp. **LESIDG**=.T. in LAM models), spectral multiplications by M^2 are performed by the product of a symmetric pentadiagonal matrix of dimensions $(N_s+1-\mid m\mid)*(N_s+1-\mid m\mid)$ (useful coefficients computed in routine **SUSMAP** (resp. **SUESMAP** in LAM models) and stored in the array **SCGMAP** (resp. **ESCGMAP** in LAM models)) 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.

∗ **Case where the constraint C1 is not matched (non zero COR):** An iterative algorithm has been implemented, which can briefly described as follows:

- The total number of iterations is **NITERHELM**.
- For the predictor step, we replace COR by 0 and we do the eliminations and the inversion of the Helmholtz equation like previously described.
- For the corrector step, the term containing COR is put in the RHS, it is multiplied by $D'_{t+\Delta t}$ at the previous iteration. The LHS is unchanged, so the elimination and the Helmholtz solving can be done like in the predictor step. We rather take as unknowns the increments between the current iteration and the predictor step, that allows to simplify the calculation of the RHS for the corrector step (some terms become 0).

Some preliminary tests done with this iterative algorithm unfortunately show that it does not converge (COR is too big, especially at the top of the model, and ill-conditioned).

5.3 3D NH-GEOGW model.

The code described is valid for NH prognostic variables Φ and gw but linear terms do not change if taking $\Phi - B\Phi_s$ or $gw - Bgw_s$.

∗ **Expression of the linear term $\mathcal{B}$ for GMV and GMVS variables:**

- Continuity equation ($X = \log(\Pi_s)$):
$$\mathcal{B} = -\nu(\overline{M}^2 D') \tag{91}$$

- Divergence equation ($X = D'$):
$$\mathcal{B} = -\nabla'^2[R_d T + R_d T^* \log(\Pi_s) + (\partial^* + 1)\Phi] \tag{92}$$

- Vorticity equation ($X = \zeta'$):
$$\mathcal{B} = 0 \tag{93}$$

- Temperature equation ($X = T$):
$$\mathcal{B} = -\frac{R_d T^*}{c_{vd}}[\overline{M}^2 D' - \frac{1}{R_d T_a^*}\partial^*(gw)] \tag{94}$$

- Geopotential equation ($X = \Phi$):
$$\mathcal{B} = -[-R_d T^*(\nu - \frac{c_{pd}}{R_d T^*}\tau)(\overline{M}^2 D') - (gw)] \tag{95}$$

- Vertical velocity equation ($X = gw$):
$$\mathcal{B} = -\frac{g^2}{R_d T_a^*}[-R_d(\partial^* + 1)T - (\partial^* + 1)(\partial^*\Phi)] \tag{96}$$

∗ **System to be solved:** Equations are written for $\log(\Pi_s)$ as a prognostic variable for continuity equation.

$$\log(\Pi_s)_{t+\Delta t} + \beta\Delta t\nu(\overline{M}^2 D'_{t+\Delta t}) = \mathcal{P}^* \tag{97}$$

$$D'_{t+\Delta t} + \beta\Delta t\nabla'^2[R_d T_{t+\Delta t} + R_d T^*\log(\Pi_s)_{t+\Delta t} + (\partial^* + 1)\Phi_{t+\Delta t}] = \mathcal{D}'^* \tag{98}$$

$$T_{t+\Delta t} + \beta\Delta t\frac{R_d T^*}{c_{vd}}[\overline{M}^2 D'_{t+\Delta t} - \frac{1}{R_d T_a^*}\partial^*(gw_{t+\Delta t})] = \mathcal{T}^* \tag{99}$$

$$\Phi_{t+\Delta t} + \beta\Delta t\left[-(gw_{t+\Delta t}) - R_d T^*(\nu - \frac{c_{pd}}{R_d T^*}\tau)(\overline{M}^2 D'_{t+\Delta t})\right] = \oplus^* \tag{100}$$

$$gw_{t+\Delta t} + \beta\Delta t\frac{g^2}{R_d T_a^*}[-R_d(\partial^* + 1)T_{t+\Delta t} - (\partial^* + 1)(\partial^*\Phi_{t+\Delta t})] = g\mathcal{W}^* \tag{101}$$

$\mathcal{P}^*, \mathcal{D}'^*, \mathcal{T}^*, \oplus^*, g\mathcal{W}^*$, correspond to $\mathcal{X}^*$ defined in equation (17) and are available in spectral arrays (**SPA3%SP**, **SPA3%DIV**, **SPA3%T**, **SPA3%SPD**, **SPA3%SVD**) at the beginning of spectral computations.
We can notice that this linear system contains vertical integrals and vertical derivatives; equation (101) uses something looking like the Laplacian operator $\mathbf{L}^*$.

∗ **Elimination of variables:** The following calculations are valid for both $\overline{M}$ horizontally constant and $\overline{M}$ horizontally variable. All the constants and the vertical operators commute with $\nabla^{'2}$ and $\overline{M}$.

We introduce the following denotations:

$$\mathcal{D}^{'**} = \mathcal{D}^{'*} - \beta\Delta t\nabla^{'2}\left[R_\mathrm{d}T^*\mathcal{P}^* + R_\mathrm{d}\mathcal{T}^* + (\partial^* + 1)\oplus^*\right] \tag{102}$$

$$\mathcal{W}^{**} = \mathcal{W}^* + \beta\Delta t\frac{g}{R_\mathrm{d}T_\mathrm{a}^*}\left[R_\mathrm{d}(\partial^* + 1)\mathcal{T}^* + (\partial^* + 1)(\partial^*\oplus^*)\right] \tag{103}$$

$$\mathtt{B_1} = -\left[\frac{R_\mathrm{d}^2T^*}{c_\mathrm{vd}} + R_\mathrm{d}T^*\nu\right] + \left[R_\mathrm{d}T^*(\partial^* + 1)(\nu - \frac{c_\mathrm{pd}}{R_\mathrm{d}T^*}\tau)\right] \tag{104}$$

$$\mathtt{B_2} = \frac{R_\mathrm{d}}{c_\mathrm{vd}}\frac{T^*}{T_\mathrm{a}^*}\partial^* + (\partial^* + 1) \tag{105}$$

$$\mathtt{B_3} = g^2\frac{R_\mathrm{d}}{c_\mathrm{vd}}\frac{T^*}{T_\mathrm{a}^*}(\partial^* + 1) - g^2\frac{T^*}{T_\mathrm{a}^*}(\partial^* + 1)(\partial^*(\nu - \frac{c_\mathrm{pd}}{R_\mathrm{d}T^*}\tau)) \tag{106}$$

$$\mathtt{B_4} = -\frac{g^2T^*}{c_\mathrm{vd}T_\mathrm{a}^{*2}}(\partial^* + 1)\partial^* - \frac{g^2}{R_\mathrm{d}T_\mathrm{a}^*}(\partial^* + 1)\partial^* \tag{107}$$

- Elimination of T, Φ and $\log(\Pi_\mathrm{s})$ between equations (97), (99), (100), and (98) leads to equation (108):

$$[\mathtt{I} + (\beta\Delta t)^2\mathtt{B_1}\nabla^{'2}\overline{M}^2]D'_{t+\Delta t} + [(\beta\Delta t)^2\mathtt{B_2}\nabla^{'2}]gw_{t+\Delta t} = \mathcal{D}^{'**} \tag{108}$$

- Elimination of T and Φ between equations (99), (100) and (101) leads to equation (109):

$$[(\beta\Delta t)^2\mathtt{B_3}\overline{M}^2]D'_{t+\Delta t} + [\mathtt{I} + (\beta\Delta t)^2\mathtt{B_4}]gw_{t+\Delta t} = g\mathcal{W}^{**} \tag{109}$$

The elimination which requires the least constraints is the one providing an Helmholtz equation with the unknown $D'_{t+\Delta t}$. In this case the commutativity between $(\partial^* + 1)$ and ∂^* is a sufficient condition: it is ensured if we take exactly the same operator ∂^* everywhere. Elimination of $gw_{t+\Delta t}$ leads to Helmholtz equation (110):

$$[\mathtt{I} - (\beta\Delta t)^2\ \mathtt{B}\ \nabla^{'2}\overline{M}^2]D'_{t+\Delta t} = \mathcal{D}^{'**} - [\mathtt{I} + (\beta\Delta t)^2\mathtt{B_4}]^{-1}[(\beta\Delta t)^2\mathtt{B_2}\nabla^{'2}]g\mathcal{W}^{**} \tag{110}$$

where:

$$\mathtt{B}\ = [\mathtt{I} + (\beta\Delta t)^2\mathtt{B_4}]^{-1}[-\mathtt{B_1} - (\beta\Delta t)^2(\mathtt{B_4B_1} - \mathtt{B_2B_3})] \tag{111}$$

When $\overline{M} = M$ (case **LSIDG**=.T. or **LESIDG**=.T.) it is more convenient to rewrite equation (110) as:

$$[\nabla^{'-2} - (\beta\Delta t)^2\ \mathtt{B}\ \overline{M}^2]D'_{t+\Delta t} = \nabla^{'-2}\mathcal{D}^{'**} - [\mathtt{I} + (\beta\Delta t)^2\mathtt{B_4}]^{-1}[(\beta\Delta t)^2\mathtt{B_2}]g\mathcal{W}^{**} \tag{112}$$

which shows a symmetric matricial operator in the left hand side.

We can do some remarks:
- The product of $\overline{M}^2$ and $\nabla^{'2}$ appears as $\nabla^{'2}\overline{M}^2$, like in the hydrostatic model: this is because the elimination has been in order to have $D'_{t+\Delta t}$ as unknown.
- Like in the NH-PDVD model, $\mathtt{B}$ depends on Δt, it must be recomputed each time the timestep is changed.
- When Δt becomes 0, $\mathtt{B}$ converges towards $-\mathtt{B_1}$.
- $\mathtt{B}$ contains vertical integrals, the Laplacian operator and first-order derivatives, and the inverse of a matrix containing the Laplacian term: in practical the positive definiteness of $\mathtt{B}$ is not easy to match (this is true in continuous equations but this property is lost in the discretised equations as they are discretised in cycle 43, for non-evanescent timesteps), and we have also some difficulties to specify the upper and lower boundary conditions for vertical derivatives.

∗ **Spectral computations to solve system of equations (97) to (101).** The algorithm has some similarities with what is done in the hydrostatic model and the NH-PDVD model. Algorithm works zonal wave number by zonal wave number m ($\mid m \mid$ varies between 0 and the truncation N_s) and performed in the routine **SPNHSI_GEOGW** before all horizontal diffusion schemes. For a given zonal wave number m:

- After a preliminary memory transfer the right-hand side member of equation (110) is computed for all total wave numbers n between m and N_s. We can do some remarks:

  - The matricial operator $[\mathtt{I} + (\beta\Delta t)^2\mathtt{B_4}]$ which appears in the RHS of (110) and also in $\mathtt{B}$ is precomputed in routine **SUNHSI** and is stored in the array **SIFAC**. Its inverse is stored in the array **SIFACI**.

21

- 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 B is used: B $= \mathbb{Q}^{-1}$ A $\mathbb{Q}$, where A is a diagonal $L * L$ matrix, the diagonal coefficients a_l of which are stored in the array **SIVP**. $\mathbb{Q}$ is a $L * L$ matrix stored in the array **SIMI**, $\mathbb{Q}^{-1}$ is stored in the array **SIMO**. Note that the vertical operators ν, τ, γ, ∂^*, B, $\mathbb{Q}$ commute with the horizontal operator ∇'^2.

  - Helmholtz equation (110) becomes, for each eigenmode l:

  $$(\mathtt{I} - \beta^2 \Delta t^2 a_l \nabla'^2 \overline{M}^2)\mathbb{Q}D'_{t+\Delta t} = \mathbb{Q}(\mathcal{D}'^{**} - [\mathtt{I} + (\beta\Delta t)^2\mathtt{B_4}]^{-1}[(\beta\Delta t)^2\mathtt{B_2}\nabla'^2]g\mathcal{W}^{**}) \qquad (113)$$

  - 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 $(N_s + 1 - |m|) * (N_s + 1 - |m|)$: spectral coefficients of the right-hand side member of (113) are simply divided by the diagonal coefficients of this matrix. The result is then multiplied by $\mathbb{Q}$.

- Inversion of Helmholtz equation for the case "unreduced divergence" (case **LSIDG**=.T. or **LESIDG**=.T.): 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 ∇'^2.

  - First the diagonalisation of B is used: B $= \mathbb{Q}^{-1}$ A $\mathbb{Q}$, where A is a diagonal $L * L$ matrix, the diagonal coefficients a_l of which are stored in the array **SIVP**. $\mathbb{Q}$ is a $L * L$ matrix stored in the array **SIMI**, $\mathbb{Q}^{-1}$ is stored in the array **SIMO**. Note that the vertical operators ν, τ, γ, ∂^*, B, $\mathbb{Q}$ commute with the horizontal operators ∇'^2 and M^2.

  - Helmholtz equation (110) (resp. (112)) becomes, for each eigenmode l:

  $$(\mathtt{I} - \beta^2 \Delta t^2 a_l \nabla'^2 M^2)\mathbb{Q}D'_{t+\Delta t} = \mathbb{Q}(\mathcal{D}'^{**} - [\mathtt{I} + (\beta\Delta t)^2\mathtt{B_4}]^{-1}[(\beta\Delta t)^2\mathtt{B_2}\nabla'^2]g\mathcal{W}^{**}) \qquad (114)$$

  resp.:

  $$(\nabla'^{-2} - \beta^2 \Delta t^2 a_l M^2)\mathbb{Q}D'_{t+\Delta t} = \mathbb{Q}(\nabla'^{-2}\mathcal{D}'^{**} - [\mathtt{I} + (\beta\Delta t)^2\mathtt{B_4}]^{-1}[(\beta\Delta t)^2\mathtt{B_2}]g\mathcal{W}^{**}) \qquad (115)$$

  - 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 $(N_s + 1 - |m|) * (N_s + 1 - |m|)$. 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 (115) 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 $\mathbb{Q}D'_{t+\Delta t}$. Multiplying by $\mathbb{Q}^{-1}$ one obtains $D'_{t+\Delta t}$.

  - For the zonal wave number $m = 0$ equation (114) is used rather than (115) because, for the total wave number $n = 0$, ∇'^2 is equivalent to a multiplication by 0 and ∇'^{-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 $D'_{t+\Delta t}$, and after calculation of $\overline{M}^2 D'_{t+\Delta t}$ (which requires an additional call to routine **MXPTMA** if $\overline{M} = M$), equation (109) yields $gw_{t+\Delta t}$ (use matrix stored in **SIFACI**). Equation (97) yields $\log\Pi_{s\,t+\Delta t}$, equation (99) yields $T_{t+\Delta t}$, and equation (100) yields $\Phi_{t+\Delta t}$.

- Semi-implicit scheme ends by a final memory transfer.

∗ **Upper and lower boundary conditions for ∂^*.** This point is not easy to solve in the linear model because these boundary conditions are not always well known. There is a possibility to ignore them: in this case they are hidden in the operator $\mathcal{R}_{\mathrm{deri}}$, but we do not guarantee that they are always proper ones, especially at the bottom boundary. There is also possibility to specify them explicitly (use a $\mathcal{R}_{\mathrm{deri}}$ operator which takes account of them). Here is currently the status of what is done:

- All first-order vertical derivatives are assumed to be 0 at the top and bottom: in particular, $\partial^* Z$ is assumed to be 0 at the top and bottom in $(\partial^* + 1)(\partial^* Z)$.

- $(\nu - \frac{c_{\mathrm{p_d}}}{R_{\mathrm{d}}T^*}\tau)$: top condition equal to ν, bottom condition equal to 0.

- ∂^* is assumed to match $\partial^*\nu = 0$ (this is currently the case).

- For other quantities, top condition is assumed constant.

- Bottom condition is assumed constant for T.

- Bottom condition is not convenient to specify for Φ or gw for calculations done in spectral space, because we are not assumed to have surface values (actually we have it for Φ but not for gw). This is one of the reasons (but not the only one) why it seems better to take the prognostic variables $\Phi - B\Phi_s$ and $gw - Bgw_s$: bottom conditions are 0 (but the assumption that the first-order vertical derivative is zero is not very good for this case!).

5.4   2D shallow-water model.

∗ **Expression of the linear term $\mathcal{B}$:**

- Continuity equation ($X = \Phi$):

$$\mathcal{B} = -\Phi^* \overline{M}^2 D' \tag{116}$$

- Divergence equation ($X = D'$):

$$\mathcal{B} = -\nabla'^2(\Phi) \tag{117}$$

- Vorticity equation ($X = \zeta'$):

$$\mathcal{B} = 0 \tag{118}$$

∗ **System to be solved:**

$$\Phi_{t+\Delta t} + \beta\Delta t \overline{M}^2 \Phi^* D'_{t+\Delta t} = \mathcal{H}^* \tag{119}$$

$$D'_{t+\Delta t} + \beta\Delta t \nabla'^2(\Phi_{t+\Delta t}) = \mathcal{D}'^* \tag{120}$$

$\mathcal{H}^*$, $\mathcal{D}'^*$ correspond to $\mathcal{X}^*$ defined in equations (17) and are available in spectral arrays (**SPA3%SP**, **SPA3%DIV**) at the beginning of the spectral computations. Equations (119) and (120) yield (121) (Helmholtz equation):

$$(1 - \beta^2\Delta t^2 \Phi^* \nabla'^2 \overline{M}^2)D'_{t+\Delta t} = \mathcal{D}'^* - \beta\Delta t \nabla'^2(\mathcal{H}^*) \tag{121}$$

When $\overline{M} = M$ (case **LSIDG**=.T.) it is more convenient to rewrite equation (121) as:

$$(\nabla'^{-2} - \beta^2\Delta t^2 \Phi^* M^2)D'_{t+\Delta t} = \nabla'^{-2}\mathcal{D}'^* - \beta\Delta t(\mathcal{H}^*) \tag{122}$$

which shows a symmetric matricial operator in the left hand side.

∗ **Spectral computations to solve system of equations (119) and (120).**   Algorithm works zonal wave number by zonal wave number m ($| m |$ varies between 0 and the truncation N_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 (121) is computed for all total wave numbers n between m and N_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 $(N_s + 1 - | m |) * (N_s + 1 - | m |)$. Spectral coefficients of the right-hand side member of (121) 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 $(N_s + 1 - | m |) * (N_s + 1 - | m |)$. 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 (122) 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 (121) is used preferably than (122) because, for the total wave number $n = 0$, ∇'^2 is equivalent to a multiplication by 0 and ∇'^{-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 (119) 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.

5.5 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. designed for LAM models, coded only for a tilted-rotated Mercator projection.

5.6 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 Eulerian 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.

∗ **LAM models:** The previous point is not an issue in most applications of LAM models 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 long range forecasts (but not stability issues: **LIMPF=.F.** is as stable as **LIMPF=.T.** but may lead to worse scores for long-range forecasts).

6.1 Semi-implicit scheme including Coriolis term in the 3D hydrostatic model.

6.1.1 Thin layer equations.

∗ **Expression of the linear term** $\mathcal{B}$**:** Equations (55) and (56) become respectively:

- Divergence equation ($X = D'$):

$$\mathcal{B} = -\nabla'^2(\gamma T + \mu \log \Pi_{\mathrm{s}}) - 2\nabla'(\mathbf{\Omega} \wedge \mathbf{V}) \tag{123}$$

- Vorticity equation ($X = \zeta'$):

$$\mathcal{B} = -2\mathbf{k}.[\nabla' \wedge (\mathbf{\Omega} \wedge \mathbf{V})] \tag{124}$$

∗ **System to be solved:** Equations (58) and (60) are unchanged. Equation (59) is replaced by the two following equations, for divergence and vorticity.

$$D'_{t+\Delta t} + \beta\Delta t\nabla'^2(\gamma T_{t+\Delta t} + \mu\log(\Pi_{\mathrm{s}})_{t+\Delta t}) + \beta\Delta t(2\nabla'(\mathbf{\Omega} \wedge \mathbf{V})) = \mathcal{D}'^* \tag{125}$$

$$\zeta'_{t+\Delta t} + \beta\Delta t(2\mathbf{k}.[\nabla' \wedge (\mathbf{\Omega} \wedge \mathbf{V})]) = \zeta'^* \tag{126}$$

$\mathcal{D}'^*$, ζ^* correspond to $\mathcal{X}^*$ defined in equation (17) and are available in spectral arrays (**SPA3%DIV**, **SPA3%VOR**) at the beginning of the spectral computations.

∗ **Restriction to not stretched and not tilted model:** Inclusion of Coriolis term in 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 (125) and (126) become:

$$D_{t+\Delta t} + \beta\Delta t\nabla^2(\gamma T_{t+\Delta t} + \mu\log(\Pi_{\mathrm{s}})_{t+\Delta t}) + \beta\Delta t(2\nabla(\mathbf{\Omega} \wedge \mathbf{V})) = \mathcal{D}^* \tag{127}$$

$$\zeta_{t+\Delta t} + \beta\Delta t(2\mathbf{k}.[\nabla \wedge (\mathbf{\Omega} \wedge \mathbf{V})]) = \zeta^* \tag{128}$$

∗ **Divergence and vorticity in spherical geometry:** For a vector $\mathbf{Y}$ of horizontal components Y_{x} and Y_{y}, the divergence and vertical component of vorticity write as:

$$\nabla \mathbf{Y} = \frac{1}{a \cos \theta} \left[\frac{\partial Y_{\mathrm{x}}}{\partial \lambda} + \frac{\partial (Y_{\mathrm{y}} \cos \theta)}{\partial \theta} \right] \tag{129}$$

$$\mathbf{k}.(\nabla \wedge \mathbf{Y}) = \frac{1}{a \cos \theta} \left[\frac{\partial Y_{\mathrm{y}}}{\partial \lambda} - \frac{\partial (Y_{\mathrm{x}} \cos \theta)}{\partial \theta} \right] \tag{130}$$

∗ **Helmholtz equation:** Using relations (129) and (130) in equations (127) and (128) lead to the following equations:

$$D_{t+\Delta t} + \beta \Delta t \nabla^2 (\gamma T_{t+\Delta t} + \mu \log(\Pi_{\mathrm{s}})_{t+\Delta t}) + \beta \Delta t (-2\Omega \sin \theta) \zeta_{t+\Delta t} + \beta \Delta t \left(\frac{2\Omega \cos \theta}{a} \right) U_{t+\Delta t} = \mathcal{D}^* \tag{131}$$

$$\zeta_{t+\Delta t} + \beta \Delta t (2\Omega \sin \theta) D_{t+\Delta t} + \beta \Delta t \left(\frac{2\Omega \cos \theta}{a} \right) V_{t+\Delta t} = \zeta^* \tag{132}$$

The four following expressions are used, for each complex spectral coefficient:

- Relationship between divergence and velocity potential χ:

$$D = \nabla^2 \chi \tag{133}$$

- Relationship between vorticity and stream function ψ:

$$\zeta = \nabla^2 \psi \tag{134}$$

- (133) and (134) can be rewritten, for each complex spectral component:

$$D_{(m,n)} = -\frac{n(n+1)}{a^2} \chi_{(m,n)} \tag{135}$$

$$\zeta_{(m,n)} = -\frac{n(n+1)}{a^2} \psi_{(m,n)} \tag{136}$$

- Relationship between U, ψ and χ:

$$(Ua \cos \theta)_{(m,n)} = im \chi_{(m,n)} + (n-1) e_{(m,n)} \psi_{(m,n-1)} - (n+2) e_{(m,n+1)} \psi_{(m,n+1)} \tag{137}$$

- Relationship between V, ψ and χ:

$$(Va \cos \theta)_{(m,n)} = im \psi_{(m,n)} - (n-1) e_{(m,n)} \chi_{(m,n-1)} + (n+2) e_{(m,n+1)} \chi_{(m,n+1)} \tag{138}$$

where $e_{(0,0)} = 0$ and:

$$e_{(m,n)} = \sqrt{\frac{n^2 - m^2}{4n^2 - 1}} \tag{139}$$

Equations (135) to (138) allow to eliminate $U_{t+\Delta t}$ and $V_{t+\Delta t}$ in equations (131) and (132).

$$\left[1 - i \frac{2\Omega \beta \Delta t m}{n(n+1)} \right] D_{(m,n),t+\Delta t} + \beta \Delta t \nabla^2 (\gamma T_{(m,n),t+\Delta t} + \mu \log(\Pi_{\mathrm{s}})_{(m,n),t+\Delta t}) - \beta \Delta t (2\Omega(\sin \theta)\zeta)_{(m,n),t+\Delta t}$$

$$-\beta \Delta t \frac{2\Omega e_{(m,n)}}{n} \zeta_{(m,n-1),t+\Delta t} + \beta \Delta t \frac{2\Omega e_{(m,n+1)}}{n+1} \zeta_{(m,n+1),t+\Delta t} = \mathcal{D}^*_{(m,n)} \tag{140}$$

$$\left[1 - i \frac{2\Omega \beta \Delta t m}{n(n+1)} \right] \zeta_{(m,n),t+\Delta t} + \beta \Delta t (2\Omega(\sin \theta)D)_{(m,n),t+\Delta t} + \beta \Delta t \frac{2\Omega e_{(m,n)}}{n} D_{(m,n-1),t+\Delta t}$$

$$-\beta \Delta t \frac{2\Omega e_{(m,n+1)}}{n+1} D_{(m,n+1),t+\Delta t} = \zeta^*_{(m,n)} \tag{141}$$

Multiplication by $\sin \theta$ is eliminated by using the following relationship valid for any variable X, in not stretched and not tilted geometry:

$$[(\sin \theta)X]_{(m,n)} = e_{(m,n)} X_{(m,n-1)} + e_{(m,n+1)} X_{(m,n+1)} \tag{142}$$

Equations (140) and (141) become:

$$\left[1 - i \frac{2\Omega \beta \Delta t m}{n(n+1)} \right] D_{(m,n),t+\Delta t} + \beta \Delta t \nabla^2 (\gamma T_{(m,n),t+\Delta t} + \mu \log(\Pi_{\mathrm{s}})_{(m,n),t+\Delta t}) - \beta \Delta t \frac{2\Omega e_{(m,n)}(n+1)}{n} \zeta_{(m,n-1),t+\Delta t}$$

$$-\beta \Delta t \frac{2\Omega e_{(m,n+1)}n}{n+1} \zeta_{(m,n+1),t+\Delta t} = \mathcal{D}^*_{(m,n)} \tag{143}$$

26

$$\left[1 - i\frac{2\Omega\beta\Delta tm}{n(n+1)}\right]\zeta_{(m,n),t+\Delta t} + \beta\Delta t\frac{2\Omega e_{(m,n)}(n+1)}{n}D_{(m,n-1),t+\Delta t} + \beta\Delta t\frac{2\Omega e_{(m,n+1)}n}{n+1}D_{(m,n+1),t+\Delta t}$$
$$= \zeta_{(m,n)}^* \tag{144}$$

ζ is eliminated in equation (143) by using (144) in replacing n by $n-1$ then by $n+1$. Equation (143) becomes:

$$\left[\mathtt{I} - i\frac{2\Omega\beta\Delta tm}{n(n+1)} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}^2\frac{(n-1)(n+1)}{n^2} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}^2\frac{(n)(n+2)}{(n+1)^2}\right]D_{(m,n),t+\Delta t}$$
$$+ \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}e_{(m,n-1)}\frac{(n+1)}{(n-1)}D_{(m,n-2),t+\Delta t} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}e_{(m,n+2)}\frac{n}{(n+2)}D_{(m,n+2),t+\Delta t}$$
$$+ \beta\Delta t\nabla^2(\gamma T_{(m,n),t+\Delta t} + \mu\log(\Pi_\mathrm{s})_{(m,n),t+\Delta t})$$
$$= \mathcal{D}_{(m,n)}^* + \frac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}\frac{(n+1)}{n}\zeta_{(m,n-1)}^* + \frac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}\frac{n}{(n+1)}\zeta_{(m,n+1)}^* \tag{145}$$

$T_{t+\Delta t}$ and $\log(\Pi_\mathrm{s})_{t+\Delta t}$ are eliminated by using equations (58) and (60). That leads to Helmholtz equation (146):

$$\left[\mathtt{I} - i\frac{2\Omega\beta\Delta tm}{n(n+1)} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}^2\frac{(n-1)(n+1)}{n^2} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}^2\frac{(n)(n+2)}{(n+1)^2} - \beta^2\Delta t^2\,\mathtt{B}\,\nabla^2\right]D_{(m,n),t+\Delta t}$$
$$+ \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}e_{(m,n-1)}\frac{(n+1)}{(n-1)}D_{(m,n-2),t+\Delta t} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}e_{(m,n+2)}\frac{n}{(n+2)}D_{(m,n+2),t+\Delta t}$$
$$= \mathcal{D}_{(m,n)}^* - \beta\Delta t\nabla^2(\gamma\mathcal{T}_{(m,n)}^* + \mu\mathcal{P}_{(m,n)}^*)$$
$$+ \frac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}\frac{(n+1)}{n}\zeta_{(m,n-1)}^* + \frac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}\frac{n}{(n+1)}\zeta_{(m,n+1)}^* \tag{146}$$

Equation (146) 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 43) done in **SPCSI**. For $m=0$ complex operators become real operators.

∗ **Determination of other quantities at $t + \Delta t$:** Equation (145) is used to compute $\zeta_{t+\Delta t}$ (for each zonal wave number m, multiplication by a complex tridiagonal matrix). Then equation (58) is used to compute $\log(\Pi_\mathrm{s})_{t+\Delta t}$ and equation (60) is used to compute $T_{t+\Delta t}$.

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

Equations (123) to (126) 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 (127) to (146) remain valid, replacing D by $(r_\mathrm{s}/a)D$, ζ by $(r_\mathrm{s}/a)\zeta$, ∇ by $(r_\mathrm{s}/a)\nabla$. The code of spectral computations is unchanged.

6.2   Semi-implicit scheme including Coriolis term in the 3D NH-PDVD model.

Extending such an algorithm to the 3D NH-PDVD model in not stretched and not tilted geometry is possible but not straightforward. Most difficulties come from the fact that the unknown in the Helmholtz equation is d and not D'.

6.2.1   Thin layer equations.

∗ **Expression of the linear term $\mathcal{B}$:**   Equations (67) and (68) become respectively:

- Divergence equation ($X = D'$):

$$\mathcal{B} = -\nabla'^2[\gamma T - T^*(\gamma\hat{Q}) + \mu\log(\Pi_\mathrm{s}) + R_\mathrm{d}T^*\hat{Q}] - 2\nabla'(\mathbf{\Omega}\wedge\mathbf{V}) \tag{147}$$

- Vorticity equation ($X = \zeta'$):

$$\mathcal{B} = -2\mathbf{k}.[\nabla'\wedge(\mathbf{\Omega}\wedge\mathbf{V})] \tag{148}$$

* **Helmholtz equation:** Elimination of T, $\hat{Q}$ and $\log(\Pi_s)$ between equations (72), (74), (76), and the modified (73) is done like for the case **LIMPF**=.F. . We have now a 3-equations system with the unknowns d or d_4, D' and ζ' (equation of ζ' is identical to the hydrostatic one). Since **LIMPF** can work only with a not stretched not tilted spherical geometry, we omit the mapping factor and we replace D' and ζ' by D and ζ everywhere.

The relationships (133), (134), (133), (136), (135), (137), (138), (139), (142) are still used, in the same manner as in the hydrostatic model. Using these relationships allows to replace the occurrences of U and V exactly as it is done in the hydrostatic model (calculations are not detailed) and to have only the spectral components of d (or d_4), D' and ζ'.

To simplify, we adopt the following denotations (consistent with the ones used in the NH-GEOGW model):

- $\mathtt{B_1} = -C^2$
- $\mathtt{BC_1} = -C^2 * COR$
- $\mathtt{B_2} = -C^2 + T^*\gamma$
- $\mathtt{B_3} = -\frac{T^*}{T_a^*}\frac{1}{H^2}\mathbf{L}^*(-c_{\mathrm{pd}}\tau + C^2)$
- $\mathtt{B_4} = -\frac{T^*}{T_a^*}\frac{C^2}{H^2}\mathbf{L}^*$

Equations for d (or d_4), D and ζ become:

$$\left[\mathtt{I} - i\frac{2\Omega\beta\Delta t m}{n(n+1)} + (\beta\Delta t)^2\nabla^2(\mathtt{B_1} + \mathtt{BC_1})\right]D_{(m,n),t+\Delta t} + \left[(\beta\Delta t)^2\nabla^2\mathtt{B_2}\right]d_{(m,n),t+\Delta t}$$
$$-\beta\Delta t\frac{2\Omega e_{(m,n)}(n+1)}{n}\zeta_{(m,n-1),t+\Delta t} - \beta\Delta t\frac{2\Omega e_{(m,n+1)}n}{n+1}\zeta_{(m,n+1),t+\Delta t} = \mathcal{D}^{**}_{(m,n)} \qquad (149)$$

$$\left[(\beta\Delta t)^2\mathtt{B_3}\right]D_{(m,n),t+\Delta t}\left[\mathtt{I} + (\beta\Delta t)^2\mathtt{B_4}\right]d_{(m,n),t+\Delta t} = \hat{\mathcal{D}}^{**}_{(m,n)} \qquad (150)$$

$$\left[\mathtt{I} - i\frac{2\Omega\beta\Delta t m}{n(n+1)}\right]\zeta_{(m,n),t+\Delta t} + \beta\Delta t\frac{2\Omega e_{(m,n)}(n+1)}{n}D_{(m,n-1),t+\Delta t} + \beta\Delta t\frac{2\Omega e_{(m,n+1)}n}{n+1}D_{(m,n+1),t+\Delta t}$$
$$= \zeta^*_{(m,n)} \qquad (151)$$

ζ is eliminated in equation (149) by using (151) in replacing n by $n-1$ then by $n+1$ (exactly like we do in the hydrostatic model). Equation (149) becomes:

$$\left[\mathtt{I} - i\frac{2\Omega\beta\Delta t m}{n(n+1)} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n-1)n}}e^2_{(m,n)}\frac{(n-1)(n+1)}{n^2} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n+1)(n+2)}}e^2_{(m,n+1)}\frac{(n)(n+2)}{(n+1)^2} + (\beta\Delta t)^2\nabla^2(\mathtt{B_1} + \mathtt{BC_1})\right]D_{(m,n),t+\Delta t}$$
$$+ \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n-1)n}}e_{(m,n)}e_{(m,n-1)}\frac{(n+1)}{(n-1)}D_{(m,n-2),t+\Delta t} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n+1)(n+2)}}e_{(m,n+1)}e_{(m,n+2)}\frac{n}{(n+2)}D_{(m,n+2),t+\Delta t}$$
$$+ \left[(\beta\Delta t)^2\nabla^2\mathtt{B_2}\right]d_{(m,n),t+\Delta t}$$
$$= \mathcal{D}^{**}_{(m,n)} + \frac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta t m}{(n-1)n}}e_{(m,n)}\frac{(n+1)}{n}\zeta^*_{(m,n-1)} + \frac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta t m}{(n+1)(n+2)}}e_{(m,n+1)}\frac{n}{(n+1)}\zeta^*_{(m,n+1)} \qquad (152)$$

The similarities with the hydrostatic model are the following ones:

- The ζ^* terms added to $\mathcal{D}^{**}_{(m,n)}$ are the same ones than those added to $\mathcal{D}^{*}_{(m,n)}$ in the hydrostatic Helmholtz equation. We introduce the following quantity:

$$\mathcal{D}^{***}_{(m,n)} = \mathcal{D}^{**}_{(m,n)} + \frac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta t m}{(n-1)n}}e_{(m,n)}\frac{(n+1)}{n}\zeta^*_{(m,n-1)} + \frac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta t m}{(n+1)(n+2)}}e_{(m,n+1)}\frac{n}{(n+1)}\zeta^*_{(m,n+1)} \quad (153)$$

- In the LHS, the additional terms containing Ω (in factor of $D_{(m,n-2),t+\Delta t}$, $D_{(m,n),t+\Delta t}$ and $D_{(m,n+2),t+\Delta t}$) are the same ones as in the hydrostatic model.

That means that some pieces of code present in the hydrostatic code (the call to **SIMPLICO** and most of the input dummy arguments of **SIMPLICO**) can be re-used with no significant change in the NH model.

The main difference with the hydrostatic model, which will provide an additional difficulty, is the elimination between the D equation and the d equation. We first assume that $COR = 0$ (constraint C1) and, like in the case **LIMPF**=F, we do the elimination in order to have an Helmholtz equation with d as unknown.

Equation (150) is used three times, for the total wavenumbers $n-2$, n and $n+2$ to do the elimination of $D_{(m,n-2),t+\Delta t}$, $D_{(m,n),t+\Delta t}$ and $D_{(m,n-2),t+\Delta t}$. After this elimination, a left multiplication of the LHS and of the RHS by $\left[1+(\beta\Delta t)^2 \mathtt{B_4}\right]$ (which commutes with all the coefficients containing Ω) is performed. We obtain an Helmholtz equation containing $d_{(m,n-2),t+\Delta t}$, $d_{(m,n),t+\Delta t}$ and $d_{(m,n+2),t+\Delta t}$ in the LHS:

$$\left[\mathtt{I} - (\beta\Delta t)^2 B\nabla^2 - i\frac{2\Omega\beta\Delta tm}{n(n+1)} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}^2\frac{(n-1)(n+1)}{n^2} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}^2\frac{(n)(n+2)}{(n+1)^2}\right]d_{(m,n),t+\Delta t}$$

$$+\frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}e_{(m,n-1)}\frac{(n+1)}{(n-1)}d_{(m,n-2),t+\Delta t} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}e_{(m,n+2)}\frac{n}{(n+2)}d_{(m,n+2),t+\Delta t}$$

$$= \left[\mathtt{I}+(\beta\Delta t)^2\mathtt{B_4}\right]^{-1}\left(-(\beta\Delta t)^2\mathtt{B_3}\mathcal{D}_{(m,n)}^{***} + \hat{\mathcal{D}}_{(m,n)}^{***}\right) \tag{154}$$

where

$$\hat{\mathcal{D}}_{(m,n)}^{***} =$$

$$\left[\mathtt{I} + (\beta\Delta t)^2\mathtt{B_1}\nabla^2 - i\frac{2\Omega\beta\Delta tm}{n(n+1)} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}^2\frac{(n-1)(n+1)}{n^2} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}^2\frac{(n)(n+2)}{(n+1)^2}\right]\hat{\mathcal{D}}_{(m,n)}^{**}$$

$$+\frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}e_{(m,n-1)}\frac{(n+1)}{(n-1)}\hat{\mathcal{D}}_{(m,n-2)}^{**} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}e_{(m,n+2)}\frac{n}{(n+2)}\hat{\mathcal{D}}_{(m,n+2)}^{**} \tag{155}$$

We can first remark that the RHS of the Helmholtz equation requires an additional multiplication by a complex pentadiagonal matrix which is not present in the hydrostatic model. Such a multiplication is done in the routine **SI_MXPTCO** which re-uses some input dummy arguments of **SIMPLICO**. This multiplication computes a quantity which must be added to the $\mathcal{D}_{(m,n)}^{***}$ term, before applying the inverse of $\left[\mathtt{I}+(\beta\Delta t)^2\mathtt{B_4}\right]$.

Once computed its RHS, equation (154) is solved in eigenmodes space. Inversion of Helmholtz equation is equivalent to invert a pentadiagonal complex matrix (done in routine **SIMPLICO**), for each zonal wave number m; the coefficients are the same ones as in the hydrostatic model, the only difference being the content of B (filled in the set-up), so this is transparent in routine **SPNHSI**. All computations are currently (in cycle 43) done in **SPNHSI**, by calling routine **SIMPLICO**. For $m=0$ complex operators become real operators.

Once computed $d_{(m,n),t+\Delta t}$, equation (152) is used to retrieve $D_{(m,n),t+\Delta t}$. Equation (152) contains $D_{(m,n-2),t+\Delta t}$, $D_{(m,n),t+\Delta t}$ and $D_{(m,n+2),t+\Delta t}$, that means that we need a second complex matrix inversion (there was only one in the hydrostatic model). But we can notice that the coefficients in the LHS look like those present in the LHS of the Helmholtz equation:

- All the coefficients containing Ω are exactly the same ones of the Helmholtz equation.
- The coefficient $(\beta\Delta t)^2$ B ∇^2 of the Helmholtz equation is replaced by $-(\beta\Delta t)^2\mathtt{B_1}\nabla^2$, where $\mathtt{B_1}$ is already a (constant coefficients) diagonal matrix.

That means that, for this complex matrix inversion, routine **SIMPLICO** can be re-used without any change (the only input coefficient which changes is the one containing the eigenvalues).

Once computed $D_{t+\Delta t}$, equation (151) allows to retrieve $\zeta_{t+\Delta t}$, and this piece of calculations is identical to what is done in the hydrostatic model.

∗ **Determination of other quantities at $t+\Delta t$:** Retrieval of T, $\hat{Q}$ and $\log(\Pi_s)$ at $t+\Delta t$ is done exactly like in the **LIMPF**=F case.

∗ **Combination with the NITERHELM algorithm when the constraint C1 is not matched:** This is possible and now implemented.

6.2.2 Deep layer equations (according to Wood and Staniforth, 2003).

Equations (147) to (151) 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 (152) to (154) remain valid, replacing D by $(r/a)D$, ζ by $(r/a)\zeta$, ∇ by $(r/a)\nabla$. The code of spectral computations is unchanged.

6.3 Semi-implicit scheme including Coriolis term in the 3D NH-GEOGW model.

6.3.1 Thin layer equations.

∗ **Expression of the linear term $\mathcal{B}$:** Equations (92) and (93) become respectively:

- Divergence equation ($X = D'$):

$$\mathcal{B} = -\nabla'^2[R_d T + R_d T^* \log(\Pi_s) + (\partial^* + 1)\Phi] - 2\nabla'(\boldsymbol{\Omega} \wedge \mathbf{V}) \tag{156}$$

- Vorticity equation ($X = \zeta'$):

$$\mathcal{B} = -2\mathbf{k}.[\nabla' \wedge (\boldsymbol{\Omega} \wedge \mathbf{V})] \tag{157}$$

The other equations are unchanged.

∗ **Restriction to not stretched and not tilted model:** cf. what has been said for the hydrostatic model.

∗ **Helmholtz equation:** Elimination of T, Φ and $\log(\Pi_s)$ between equations (97), (100), (99), and the modified (98) is done like for the case **LIMPF**=.F. . Elimination of T and Φ in the (101) equation is done like for the case **LIMPF**=.F. . We have now a 3-equations system with the unknowns gw, D' and ζ' (equation of ζ' is identical to the hydrostatic one). Since **LIMPF** can work only with a not stretched not tilted spherical geometry, we omit the mapping factor and we replace D' and ζ' by D and ζ everywhere.

The relationships (133), (134), (133), (136), (135), (137), (138), (139), (142) are still used, in the same manner as in the hydrostatic model. Using these relationships allows to replace the occurrences of U and V exactly as it is done in the hydrostatic model (calculations are not detailed) and to have only the spectral components of gw, D' and ζ'.

At this stage the (101) equation is unchanged compared to its **LIMPF**=F form. Additional terms present in the divergence and vorticity equations (terms containing Ω) are the same as the ones in the hydrostatic system: elimination of ζ terms into the divergence equation is done exactly as in the hydrostatic model.

We use the matrix denotations $\mathtt{B_1}$, $\mathtt{B_2}$, $\mathtt{B_3}$, $\mathtt{B_4}$ defined by formulae (104) to (107). After some long calculations often identical of those of the hydrostatic model (elimination of ζ terms, see above), we obtain the following two-equations system, with unknowns D' and gw:

$$\left[\mathtt{I} - i\frac{2\Omega\beta\Delta t m}{n(n+1)} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n-1)n}}e_{(m,n)}^2\frac{(n-1)(n+1)}{n^2} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n+1)(n+2)}}e_{(m,n+1)}^2\frac{(n)(n+2)}{(n+1)^2} + (\beta\Delta t)^2\mathtt{B_1}\nabla^2 \right] D_{(m,n),t+\Delta t}$$

$$+ \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n-1)n}}e_{(m,n)}e_{(m,n-1)}\frac{(n+1)}{(n-1)}D_{(m,n-2),t+\Delta t} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n+1)(n+2)}}e_{(m,n+1)}e_{(m,n+2)}\frac{n}{(n+2)}D_{(m,n+2),t+\Delta t}$$

$$+ \left[(\beta\Delta t)^2\nabla^2\mathtt{B_2} \right] gw_{(m,n),t+\Delta t}$$

$$= \mathcal{D}_{(m,n)}^{**} + \frac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta t m}{(n-1)n}}e_{(m,n)}\frac{(n+1)}{n}\zeta_{(m,n-1)}^* + \frac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta t m}{(n+1)(n+2)}}e_{(m,n+1)}\frac{n}{(n+1)}\zeta_{(m,n+1)}^* \tag{158}$$

$$[(\beta\Delta t)^2\mathtt{B_3}\overline{M}^2]D_{(m,n),t+\Delta t} + [\mathtt{I} + (\beta\Delta t)^2\mathtt{B_4}]gw_{(m,n),t+\Delta t} = g\mathcal{W}_{(m,n)}^{**} \tag{159}$$

After elimination of $gw_{(m,n),t+\Delta t}$ between both equations we obtain the following Helmholtz equation:

$$\left[\mathtt{I} - i\frac{2\Omega\beta\Delta t m}{n(n+1)} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n-1)n}}e_{(m,n)}^2\frac{(n-1)(n+1)}{n^2} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n+1)(n+2)}}e_{(m,n+1)}^2\frac{(n)(n+2)}{(n+1)^2} - \beta^2\Delta t^2\ \mathtt{B}\ \nabla^2 \right] D_{(m,n),t+\Delta t}$$

$$+ \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n-1)n}}e_{(m,n)}e_{(m,n-1)}\frac{(n+1)}{(n-1)}D_{(m,n-2),t+\Delta t} + \frac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n+1)(n+2)}}e_{(m,n+1)}e_{(m,n+2)}\frac{n}{(n+2)}D_{(m,n+2),t+\Delta t}$$

$$= \mathcal{D}_{(m,n)}^{**} - [\mathtt{I} + (\beta\Delta t)^2\mathtt{B_4}]^{-1}[(\beta\Delta t)^2\mathtt{B_2}\nabla'^2]g\mathcal{W}_{(m,n)}^{**}$$

$$+ \frac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta t m}{(n-1)n}}e_{(m,n)}\frac{(n+1)}{n}\zeta_{(m,n-1)}^* + \frac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta t m}{(n+1)(n+2)}}e_{(m,n+1)}\frac{n}{(n+1)}\zeta_{(m,n+1)}^* \tag{160}$$

Equation (160) 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: this calculation is similar to the one done in the hydrostatic model. All computations are currently (in cycle 43) done in **SPNHSI_GEOGW**. For $m = 0$ complex operators become real operators.

Once computed $D_{t+\Delta t}$, retrieval of $gw_{t+\Delta t}$, then retrieval of $T_{t+\Delta t}$, $\Phi_{t+\Delta t}$, and $\log(\Pi_s)_{t+\Delta t}$ is done exactly like in the case **LIMPF**=F.

Retrieval of ζ uses the following equation (for each zonal wave number m, multiplication by a complex tridiagonal matrix, like in the hydrostatic model).

$$\left[1 - i\frac{2\Omega\beta\Delta t m}{n(n+1)}\right]\zeta_{(m,n),t+\Delta t} + \beta\Delta t\frac{2\Omega e_{(m,n)}(n+1)}{n}D_{(m,n-1),t+\Delta t} + \beta\Delta t\frac{2\Omega e_{(m,n+1)}n}{n+1}D_{(m,n+1),t+\Delta t}$$
$$= \zeta_{(m,n)}^* \tag{161}$$

6.3.2 Deep layer equations (according to Wood and Staniforth, 2003).

Not coded.

6.4 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 $(\gamma T + \mu\log(\Pi_s))$ by Φ in equation (123).
- Replace $(\gamma T_{t+\Delta t} + \mu\log(\Pi_s)_{t+\Delta t})$ by $\Phi_{t+\Delta t}$ in equations (125), (127) and (131).
- Replace $(\gamma T_{(m,n),t+\Delta t} + \mu\log(\Pi_s)_{(m,n),t+\Delta t})$ by $\Phi_{(m,n),t+\Delta t}$ in equations (140), (143) and (145).
- Replace $(\gamma\mathcal{T}_{(m,n)}^* + \mu\mathcal{P}_{(m,n)}^*)$ by $\mathcal{H}_{(m,n)}^*$ in equation (146).
- Once computed $D'_{t+\Delta t}$ equation (119) provides $\Phi_{t+\Delta t}$.

6.5 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, the shallow-water model and in the NH-GEOGW model, and treatment of implicit Coriolis term is similar in all these cases, with only one call to **SIMPLICO**.

In NH models where this is w or d which is the unknown in the Helmoltz equation, treatment of implicit Coriolis term is more tricky, because we must solve a modified Helmoltz equation with additional Ω terms, then retrieve the horizontal divergence via a modified equation, then retrieve the horizontal vorticity. In such configurations we can expect two calls of **SIMPLICO**, and one call to **SI_MXPTCO**. This is the case in the NH-PDVD model.

7 Spectral multiplications by polynomial expressions of the mapping factor.

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

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

Expression of M is:

$$M = a_c + b_c \xi \tag{162}$$

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

$$[MX]_{(m,n)} = b_c e_{(m,n)} X_{(m,n-1)} + a_c X_{(m,n)} + b_c e_{(m,n+1)} X_{(m,n+1)} \tag{163}$$

It is easy from (163) 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.**

$$[M^2 X]_{(m,n)} = b_c^2 e_{(m,n)} e_{(m,n-1)} X_{(m,n-2)} + 2a_c b_c e_{(m,n)} X_{(m,n-1)} + (a_c^2 + b_c^2(e_{(m,n)}^2 + e_{(m,n+1)}^2)) X_{(m,n)}$$
$$+ 2a_c b_c e_{(m,n+1)} X_{(m,n+1)} + b_c^2 e_{(m,n+1)} e_{(m,n+2)} X_{(m,n+2)} \tag{164}$$

This is equivalent to a multiplication by a pentadiagonal symmetric matrix in spectral space.

∗ **Expression of M and M^2 in spectral space in LAM models.** 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 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.

8 Order of spectral computations.

∗ **ARPEGE/IFS:** Spectral computations are done in the 3D model in the following order:

- Mass corrector computations (**SPCMASCOR**).
- Semi-implicit computations in the hydrostatic model (**SPCSI**).
- Semi-implicit computations in the NH-PDVD model (**SPNHSI**).
- Semi-implicit computations in the NH-GEOGW model (**SPNHSI_GEOGW**).
- Main horizontal diffusion scheme (**SPCHOR**).
- Nudging (i.e. linear relaxation towards a pre-defined state) (**SPCHOR**).

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

∗ **LAM models:** Spectral computations are done in the 3D model in the following order:

- Semi-implicit computations in the hydrostatic model (**ESPCSI**).
- Semi-implicit computations in the NH-PDVD model (**ESPNHSI**).
- Semi-implicit computations in the NH-GEOGW model (**ESPNHSI_GEOGW**).
- Spectral nudging (coupling) from a coarser model (**ESPSC2R** for temporal interpolation providing coupler at the current timestep, **ESPCPL** for spectral relaxation).
- Main horizontal diffusion scheme (**ESPCHOR**).

∗ **Case of iterative centred-implicit model**
(option LPC_FULL=.T.): The inversion of Helmholtz equation is done at all iterations and horizontal diffusion can be done at the last iteration or at all iterations according to the value of key **LRHDI_LASTITERPC**.

9 Organigramme of the spectral part of the semi-implicit computations.

9.1 Set-up and control routines until STEPO.

```
CNT0 ->
 * SUOYOMA ->
   - SUCT0
   - SUDYNA
 * SUOYOMB ->
   - SUDYN ->
     * SUALDYN
     * SUALDYNB (global model) or SUELDYNB (LAM model).
     * SUSMAP -> SUGMRE (global model)
     * SUESMAP (LAM model)
     * SUSI (hydrostatic model) ->
       - GPHPRE
       - SUBMAT -> SITNU and SIGAM
       - EIGSOL
       - SCORDO
       - MINV
       - SUHEG -> SUHER and SUHES (LSIDG=T only)
       - SUEHEG -> SUHER and SUHES (LESIDG=T only)
     * SUNHSI (NH models) ->
       - GPHPRE
       - SUNHBMAT -> SISEVE and MINV (NH-PDVD model)
       - SUNHBMAT_GEOGW -> SITNU, SIVDERI and MINV (NH-GEOGW model)
       - EIGSOL
       - SCORDO
       - MINV
       - SUNHHEG -> SUHER and SUHES (LSIDG=T only, NH-PDVD model)
       - SUENHHEG -> SUHER and SUHES (LESIDG=T only, NH-PDVD model)
       - SUHEG -> SUHER and SUHES (LSIDG=T only, NH-GEOGW model)
     * SUNHSI_TESTCONV (NH models)
 * CNT1 -> CNT2 -> CNT3 -> CNT4 ->
   - SUHEG -> SUHER and SUHES (LSIDG=T only)
   - SUEHEG -> SUHER and SUHES (LESIDG=T only)
   - SUNHSI (NH model) -> (see above)
   - SUENHSI (NH model) -> (see above)
   - STEPO -> (see below)
```

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

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

9.2 Direct code under STEPO or tangent linear code under STEPOTL.

```
STEPO or STEPOTL ->
 * SPCM ->
   - SPCIMPFINIT (LIMPF=T only)
   - TRMTOS (transposition routine for distributed memory)
   - SPCSI (hydrostatic model) ->
     * SITNU -> VERINT
     * SIGAM -> VERINT
     * SPCIMPFSOLVE (LIMPF=T only) -> TRSTOM, SIMPLICO, TRMTOS
     * MXMAOP
     * MXTURS and MXTURE (LSIDG=T or LESIDG=T only)
     * MXPTMA (LSIDG=T or LESIDG=T only)
   - SPNHSI (NH-PDVD model) ->
     * SIDD -> SIGAM and SISEVE
     * SI_CCCOR
     * MXPTMA (LSIDG=T or LESIDG=T only)
     * SISEVE -> VERDER
     * SITNU -> VERINT
     * SIMPLICO and SI_MXPTCO (LIMPF=T only)
     * MXMAOP
     * MXTURS and MXTURE (LSIDG=T or LESIDG=T only)
     * SIGAM -> VERINT
```

```
        - SPNHSI_GEOGW (NH-GEOGW model) ->
          * SITNU -> VERINT
          * SIVDERI -> VERDER
          * SIMPLICO (LIMPF=T only)
          * MXPTMA (LSIDG=T or LESIDG=T only)
          * MXMAOP
          * MXTURS and MXTURE (LSIDG=T or LESIDG=T only)
        - TRSTOM (transposition routine for distributed memory)
        - SPCIMPFPOST (LIMPF=T only)
        - SPCHOR (horizontal diffusion)
 * SPC2M -> SPC2 ->
        - SIMPLICO (LIMPF=T only)
        - MXTURS (LSIDG=T or LESIDG=T only)
        - MXTURE (LSIDG=T or LESIDG=T only)
        - MXPTMA (LSIDG=T or LESIDG=T only)
        - BALADSM
```

In LAM models:

- **ESPCM**, **ESPCSI**, **ESPNHSI**, **ESPNHSI_GEOGW**, **ESPCHOR** are called instead of **SPCM**, **SPCSI**, **SPNHSI**, **SPNHSI_GEOGW**, **SPCHOR**.

9.3   Adjoint code under STEPOAD.

```
STEPOAD ->
 * SPCMAD ->
        - BRPTOB -> PE2SET
        - SPCHORAD (horizontal diffusion)
        - SPCIMPFPOSTAD (LIMPF=T only)
        - TRMTOS (transposition routine for distributed memory)
        - SPCSIAD (hydrostatic model) ->
          * SITNUAD -> VERINTAD
          * SIGAMAD -> VERINTAD
          * SPCIMPFSOLVEAD (LIMPF=T only) -> TRSTOM, SIMPLICOAD, TRMTOS
          * MXMAOP
          * MXTURS and MXTURE (LSIDG=T or LESIDG=T only)
          * MXPTMA (LSIDG=T or LESIDG=T only)
        - [ SPNHSIAD (NH-PDVD model, not yet coded) ] ->
          * SIDDAD -> SIGAMAD and SISEVEAD
          * MXPTMA (LSIDG=T or LESIDG=T only)
          * SISEVEAD
          * SITNUAD -> VERINTAD
          * MXMAOP
          * MXTURS and MXTURE (LSIDG=T or LESIDG=T only)
          * SIGAMAD -> VERINTAD
        - TRSTOM (transposition routine for distributed memory)
        - SPCIMPFINITAD (LIMPF=T only)
```

In LAM models:

- **ESPCMAD**, **ESPCSIAD** and **ESPCHORAD** are called instead of **SPCMAD**, **SPCSIAD**, **SPNHSIAD** and **SPCHORAD**.

9.4   Action done by these routines.

∗ **Set-up routines:**

- **SUCT0**: computes 0-level control variables, and also some variables linked to dynamics.
- **SUDYNA**: set-up for dynamics, part A.
- **SUDYN**: set-up for dynamics, part B.
- **SUSI**: set-up for hydrostatic SI scheme.
- **SUNHSI**: set-up for non-hydrostatic SI schemes.
- **SUNHSI_TESTCONV**: test convergence necessary condition of the **NITERHELM** algorithm (NH-PDVD SI scheme).
- **GPHPRE**: some reference vertical-dependent hydrostatic pressure quantities.
- **SCORDO**: reorders eigenvalues and eigenvectors in order to have ascending values of eigenvectors.
- **SUBMAT**, **SUNHBMAT**, **SUNHBMAT_GEOGW**: computes array **SIB** containing operator B respectively for 3D hydrostatic model, 3D NH-PDVD model, 3D NH-GEOGW model.

- **SUSMAP**: computes array **SCGMAP** containing coefficients for spectral multiplication by M^2. **SUSMAP** calls intermediate routine **SUGMRE**.
- **SUESMAP**: computes array **ESCGMAP** containing coefficients for spectral multiplication by M^2 (LAM models, tilted-rotated Mercator projection).
- **SUHEG** (**SUEHEG** in LAM models): computes the LU factorisation of Helmholtz operator in case of semi-implicit scheme with unreduced divergence. Routine **SUHEG** (resp. **SUEHEG**) is called only if **LSIDG**=.T. (resp. **LESIDG**=.T.), for 3D hydrostatic model, 3D NH-GEOGW model and 2D shallow-water model (and more generally when the elimination gives an Helmholtz equation with D' as unknown).
- **SUNHHEG** (resp. **SUENHHEG** in LAM models): the same as **SUHEG** (resp. **SUEHEG** in LAM models) but for the NH-PDVD model.
- **SUALDYNB** (ARPEGE) or **SUELDYNB** (LAM models): allocation of some arrays used in the semi-implicit scheme.

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

∗ **Routines under STEPO, STEPOAD or STEPOTL:**
- **SPCSI**: semi-implicit scheme spectral computations in the hydrostatic model.
- **SPNHSI** (resp. **SPNHSI_GEOGW**): semi-implicit scheme spectral computations in the NH-PDVD (resp. NH-GEOGW) model.
- **SPC2**: spectral space computations, including the semi-implicit scheme and all horizontal diffusion schemes in the 2D model.
- **SPCM**: distributed memory interface for **SPCSI** and **SPNHSI**.
- **SPC2M**: distributed memory interface for **SPC2**.
- Their LAM models counterparts have names **ESPC..** instead of **SPC..**.
- These routines have adjoints (same names + "AD").
- **SPCIMPFINIT**, **SPCIMPFPOST**, **SPCIMPFSOLVE**: contain code for **LIMPF**=T case.
- **SI_CCCOR**: computes COR (NH-PDVD model).
- **SITNU**: computes a linear application by operator ν (see equations (31) and (30)).
- **SIGAM**: computes a linear application by operator γ (see equations (27) and (26)).
- **SISEVE**: computes a linear application by operator $\mathbf{L}^*$ in the NH-PDVD model (see equation (34)).
- **SIDD**: performs the elimination of T, $\log \Pi_s$ and $\hat{Q}$ in the linear system of NH-PDVD equations (in order to compute the RHS of the Helmholtz equation).
- **SIVDERI**: computes a linear application by operator ∂^*.
- **VERINT**: does vertical integrations (vertical finite element scheme).
- **VERDER**: does vertical derivations (vertical finite element scheme).
- **BALADSM**: solve linear balance equation in spectral space to convert vorticity into geopotential (used for vorticity 2D equation only).

∗ **Linear algebra routines (project XLA/ALGOR):**
- **EIGSOL**: finds eigenvalues and eigenvectors of a matrix.
- **MINV**: inverts a matrix.
- **MXTURE**: inverts a set of tridiagonal (lower or upper) triangular matrices.
- **MXTURS**: combines two calls to **MXTURE** to invert a set of symmetric pentadiagonal matrices, the decomposition LU of which is known.
- **MXPTMA**: products of a set of pentadiagonal matrices by a set of matrices or vectors.
- **MXMAOP**: matrix by matrix product.
- **SIMPLICO**: solves a set of set of complex pentadiagonal systems (in practical called when **LIMPF**=T).
- **SI_MXPTCO**: complex multiplications by a penta-diagonal matrix (in practical called when **LIMPF**=T).
- **SUHER**: performs the LU factorisations of a set of non symmetric pentadiagonal matrices.
- **SUHES**: performs the LU factorisations of a set of symmetric pentadiagonal matrices.

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

10   Other remarks.

∗ **Adjoint and tangent linear codes:**   These codes have been updated for semi-implicit computations in cycle 43, for 3D hydrostatic model and 2D model. Use of both options **LSIDG**=.F. and **LSIDG**=.T. is possible for adjoint and tangent linear codes of semi-implicit scheme in cycle 43 (except for non-hydrostatic model). Notice that tangent linear of **SPCSI** is **SPCSI** itself.

∗ **Place of calculation of linear terms in grid-point space:**

In the 3D model, we find such calculations in **CPEULDYN** for the Eulerian advection and in **LACDYN** (callees **LASSIE**, **LANHSI** and **LANHSI_GEOGW**) for the semi-Lagrangian advection. Grid-point coupling also needs to compute these terms to add them to couplers (routine **ESEIMPLS** called by **ECOUPL1**).
In the 2D model, we find such calculations in **CPG2** for the Eulerian advection and in **LACDYNSHW** for the semi-Lagrangian advection.

Linear term calculations are done according to the following pattern:
- 3D hydrostatic model:
    - one call to **SITNU**.
    - two calls to **SIGAM**.
    - if **LSPRT**=T, a conversion for the T-equation linear term.
    - if **LIMPF**=T, add Coriolis term to wind-equation linear term.
- 3D NH-PDVD model:
    - one call to **SIPTP**.
    - two calls to **SIDD**.
    - one call to **SISEVE**.
    - if **LSPRT**=T, a conversion for the T-equation linear term.
    - if **LIMPF**=T, add Coriolis term to wind-equation linear term.
- 3D NH-GEOGW model:
    - one call to **SITNU**.
    - several calls to **SIVDERI**.
    - additional in-lined calculations.
    - if **LSPRT**=T, a conversion for the T-equation linear term.
    - if **LIMPF**=T, add Coriolis term to wind-equation linear term.
- timestep of calculation (given for predictor step):
    - Eulerian advection: linear terms for $X(t - \Delta t) - 2X(t)$.
    - 3TLSL advection: linear terms for $X(t)$ and $X(t - \Delta t)$.
    - 2TLSL advection: linear terms for $X(t)$.
    - coupler: linear terms for coupler instant.

∗ **Some distributed memory features:**
- The total number of processors involved in the A-level parallelisation is **NPRTRW**.
- The total number of processors involved in the B-level parallelisation is **NPRTRN**.
- One processor treats only a subset of zonal wave numbers.
- If **LSIDG**=.T. or **LIMPF**=.T. spectral part of the semi-implicit scheme is done zonal wave number by zonal wave number. A call to **SPCSI** currently treats only one zonal wave number (case **LLONEM**=.T.).
- In the other cases a call to **SPCSI** can treat several wave numbers: currently all the zonal wave numbers treated by the current processor (case **LLONEM**=.F.).
- All the **NFLEVG** layers are treated together, there is no subdivision into packets of **NFLEVL** layers when the second level of parallelisation is activated contrary to the horizontal diffusion. That means that additional transpositions (**TRSTOM** in the direct code) are necessary between the semi-implicit calculations of **SPCSI** and the horizontal diffusion calculations of **SPCHOR** to convert the fields from the **NFLEVG** structure required in the semi-implicit calculations to the **NFLEVL** structure required in the horizontal diffusion calculations.
- In LAM models the way of distributing **ESPCSI**, **ESPNHSI** and **ESPNHSI_GEOGW** is the same one as in **SPCSI**, **SPNHSI** and **SPNHSI_GEOGW**.

11   Precomputed module and namelist quantities.

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

- Modules for geometry:

  - **SPGEOM_MOD** (spectral geometry).
  - **YOMVERT**: all variables.

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

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

- **YOMCVER** (vertical finite element discretisation keys): most of variables. Some of these variables are in namelist **NAMCVER**.

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

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

  - LPC_FULL, LPC_CHEAP (predictor-corrector scheme).
  - LNESC, LNESCT, LNESCV, LSETTLS, LSETTLST, LSETTLSV (extrapolation).
  - LSLINLC1, LSLINLC2, LSLINL.
  - LAPRXPK, NDLNPR, RHYDR0 (vertical discretisation).
  - NPDVAR, NVDVAR, ND4SYS, LNH_PDVD, LNH_GEOGW, LNHX, LNHXDER, RC_PD1 (NH model).
  - LGWADV, NGWADVSI, LRDBBC (treatment of vertical divergence equation in NH model).
  - LVERCOR, LRWSDLW, LRWSDLR, LRWSDLR2, LRWSDLG, LCURVW (deep-layer equations).

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

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

  - LSIDG, BETADT (β), RBT, RBTS2, NITERHELM, LIMPF (semi-implicit scheme).
  - REFGEO, SIPR (Π_s^*), SITR (T^*), SITRA (T_a^*), SITRUB, SIPRUB, SITIME, SIRPRG ($R_d T^*$), SIRPRN (equal to 1): reference values used in the semi-implicit scheme.
  - VESL, XIDT.
  - NSITER, NCURRENT_ITER, LRHDI_LASTITERPC (predictor-corrector scheme).
  - SIDELP ($\Delta\Pi^*$), SIRDEL ($1/\Delta\Pi^*$), SILNPR (δ^*), SIALPH (α^*).
  - SITLAF and SITLAH (full-level and half-level reference hydrostatic pressure).
  - SIDPHI ($\Delta\Phi^*$).
  - SIB (matrix `B` in Helmholtz equation).
  - SIVP (eigenvalues a_l of `B`), SIMI (`Q` = eigenvectors of `B`), SIMO (`Q`$^{-1}$).
  - SIFAC (matricial operator for NH model only)

$$\left(\mathtt{I} - \beta^2 \Delta t^2 C^2 \frac{T^*}{T_a^*} \frac{\mathbf{L}^*}{H^2} \right)$$

  - SIFACI: inverse of SIFAC.
  - SIHEG:
    * Hydrostatic model, NH-GEOGW model or 2D shallow-water model: for zonal wave numbers different from zero, contains the non-zero diagonals of `LU` decomposition of:

$$(\nabla'^{-2} - \beta^2 \Delta t^2 a_l M^2)$$

      for zonal wave number equal to 0, contains the non-zero diagonals of L of the `LU` decomposition of:

$$(\mathtt{I} - \beta^2 \Delta t^2 a_l \nabla'^2 M^2)$$

* NH-PDVD 3D model: for zonal wave numbers different from zero, contains the non-zero diagonals of `LU` decomposition of:

$$(\nabla^{'-2} - \beta^2 \Delta t^2 a_l M^2)$$

for zonal wave number equal to 0, contains the non-zero diagonals of `L` of the `LU` decomposition of:

$$(\mathtt{I} - \beta^2 \Delta t^2 a_l M^2 \nabla^{'2})$$

- SIHEG2:
    * Hydrostatic model, NH-GEOGW model or 2D shallow-water model: for zonal wave number equal to zero, contains the non-zero diagonals of `U` of the `LU` decomposition of:

$$(\mathtt{I} - \beta^2 \Delta t^2 a_l \nabla^{'2} M^2)$$

    * NH-PDVD 3D model: for zonal wave number equal to zero, contains the non-zero diagonals of `U` of the `LU` decomposition of:

$$(\mathtt{I} - \beta^2 \Delta t^2 a_l M^2 \nabla^{'2})$$

- SIHEGB for NH-PDVD 3D model: for zonal wave numbers different from zero, contains the non-zero diagonals of `LU` decomposition of:

$$(\nabla^{'-2} - \beta^2 \Delta t^2 C M^2)$$

for zonal wave number equal to 0, contains the non-zero diagonals of `L` of the `LU` decomposition of:

$$(\mathtt{I} - \beta^2 \Delta t^2 \nabla^{'2} C M^2)$$

- SIHEGB2 for NH-PDVD 3D model: for zonal wave number equal to zero, contains the non-zero diagonals of `U` of the `LU` decomposition of:

$$(\mathtt{I} - \beta^2 \Delta t^2 \nabla^{'2} C M^2)$$

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

- **YOMRIP** (date and timestep related variables). The following variables are attributes of YRRIP. Some of these variables are in namelist **NAMRIP**.

    - TSTEP, TDT (timestep).

- **YEMDYN** (LAM model dynamics): LESIDG, RTHRESIDG. RTHRESIDG is in namelist **NEMDYN**.

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

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

12 References.

12.1 Publications.

- Bénard, P., 2003: Stability of semi-implicit and iterative centred implicit time discretization for various equation systems used in NWP. *Mon. Wea. Rev.*, **131**, 2479-2491.

- Bénard, P., R. Laprise, J. Vivoda, and P. Smolíková, 2004: Stability of leapfrog constant-coefficients semi-implicit schemes for the fully elastic system of Euler equations: flat-terrain case. *Mon. Wea. Rev.*, **132**, 1306-1318.

- Bénard, P., J. Masek, and P. Smolíková, 2005: Stability of leapfrog constant-coefficients semi-implicit schemes for the fully elastic system of Euler equations: Case with orography. *Mon. Wea. Rev.*, **133**, 1065-1075.

- Bénard, P., 2004: On the use of a wider class of linear systems for the design of constant-coefficients semi-implicit time schemes in NWP. *Mon. Wea. Rev.*, **132**, 1319-1324.

- Bubnová, R., G. Hello, P. Bénard, and J.F. Geleyn, 1995: Integration of the fully elastic equations cast in the hydrostatic pressure terrain-following coordinate in the framework of the ARPEGE/Aladin NWP system. *Mon. Wea. Rev.*, **123**, 515-535.

- Courtier, Ph., C. Freydier, J.F. Geleyn, F. Rabier and M. Rochas, 1991: The ARPEGE project at METEO-FRANCE. ECMWF Seminar Proceedings 9-13 September 1991, Volume II, 193-231.

- Courtier, Ph., and J.F. Geleyn, 1988: A global numerical weather prediction model with variable resolution: Application to the shallow-water equations. *Quart. J. Roy. Meteor. Soc.*, **114**, 1321-1346.

- Geleyn, J.F., and R. Bubnová, 1995: The fully elastic equations cast in hydrostatic pressure coordinate: accuracy and stability aspects of the scheme as implemented in ARPEGE/Aladin. *Atm. Ocean, special issue memorial André Robert.*

- Laprise, R., 1992: The Euler equations of motion with hydrostatic pressure as an independent variable. *Mon. Wea. Rev.*, **120**, 197-207.

- Schmidt, F., 1977: Variable fine-mesh in spectral global model. *Beitr. Phys. Atmos.*, **50**, 211-227.

- Simmons, A. J., and B.J. Hoskins, 1978: Stability of the semi-implicit method of time integration. *Mon. Wea. Rev.*, **106**, 405-412.

- Temperton, C., 1997: Treatment of the Coriolis terms in semi-Lagrangian spectral models. *Atm. Ocean, 35:sup1, special issue memorial André Robert*, 293-302.

- Wood, N., and A. Staniforth, 2003: The deep-atmosphere Euler equations with a mass-based vertical coordinate. *Q. J. R. Meteorol. Soc.*, **129**, 1289-1300.

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

12.2 Internal notes and documentation.

- (TDECDYN) 2015: IFS technical documentation (CY41R1). Part III: dynamics and numerical procedures. Available at "https://software.ecmwf.int/wiki/display/IFS/Official+IFS+Documentation".

- (TDECTEC) 2015: IFS technical documentation (CY41R1). Part VI: technical and computational procedures. Available at "https://software.ecmwf.int/wiki/display/IFS/Official+IFS+Documentation".

- (IDNHPB) Bénard, P., 2013: Scientific documentation for ALADIN NH model (version 3.1). Internal note (94pp), available on "http://www.cnrm.meteo.fr/gmapdoc/".

- Bénard, P., 2002: Harmonization and rationalization of iterative time-schemes in IFS/ARPEGE/ALADIN. Internal note (11pp).

- (IDPC) Bénard, P., 2002: Incremental versus non-incremental predictor-corrector schemes. Appendix A of the previous internal note.

- Bénard, P., 2004: Study of the VFE discretisation in view of NH modelling. Internal note, 37pp.

- Bénard, P., 2007: A review on some sets of prognostic variables for Euler equations in mass-based coordinates. Internal note, 21pp.

- Simmons, A. J., and C. Temperton, 1996: Stability of a two-time-level semi-implicit integration scheme for gravity-wave motion. *ECMWF Technical Memorandum*, **226**, 33pp.

- (IDVNH1) Smolíková, P., 2001: New strategies for non-hydrostatic temporal scheme. Internal note.

- (IDVNH2) Smolíková, P., 2002: New NH variables: d_4 in three dimensions (in the cycle CY25T1). Internal note.

- (IDVNH3) Smolíková, P., 2003: The use of diagnostic BBC (boundary bottom condition) in semi-Lagrangian schemes. Internal note, 8pp.

- Temperton, C., 1996: Economical solution of the vertically-coupled semi-implicit equations. *ECMWF Technical Memorandum*, **227**, 3pp.

- (IDPCEU) Vivoda, J., 2004: Three-time level iterative centred implicit scheme with Eulerian advection treatment. Internal note, 16pp.

- (IDPCSL) Vivoda, J., 2005: Two-time level iterative centred implicit semi-Lagrangian scheme with grid-point prognostic variable gw and semi-implicit variable d or d_4. Internal note, 15pp.

- (IDBAS) Yessad, K., 2016: Basics about ARPEGE/IFS, ALADIN and AROME in the cycle 43 of ARPEGE/IFS (internal note, available on the intranet server "http://www.cnrm.meteo.fr/gmapdoc/").

- (IDDH) Yessad, K., 2016: Horizontal diffusion in the cycle 43 of ARPEGE/IFS (internal note, available on the intranet server "http://www.cnrm.meteo.fr/gmapdoc/").

- (IDTS) Yessad, K., 2016: Spectral transforms in the cycle 43 of ARPEGE/IFS (internal note, available on the intranet server "http://www.cnrm.meteo.fr/gmapdoc/").

- (IDSL) Yessad, K., 2016: Semi-Lagrangian computations in the cycle 43 of ARPEGE/IFS (internal note, available on the intranet server "http://www.cnrm.meteo.fr/gmapdoc/").

- (IDFPOS) Yessad, K., 2016: FULL-POS in the cycle 43 of ARPEGE/IFS (internal note, available on the intranet server "http://www.cnrm.meteo.fr/gmapdoc/").

- (IDRD) Yessad, K., 2016: Sphere to sphere transforms in spectral space in the cycle 43 of ARPEGE/IFS: configuration 911 (internal note, available on the intranet server "http://www.cnrm.meteo.fr/gmapdoc/").

- (IDDM) Yessad, K., 2016: Distributed memory features in the cycle 43 of ARPEGE/IFS (internal note, available on the intranet server "http://www.cnrm.meteo.fr/gmapdoc/").

- (IDEUL) Yessad, K., 2016: Integration of the model equations, and Eulerian dynamics, in the cycle 43 of ARPEGE/IFS (internal note, available on the intranet server "http://www.cnrm.meteo.fr/gmapdoc/").

- (IDESIDG) Yessad, K., 2006: Implementation of option **LESIDG** in ALADIN (internal note, 13pp).

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