SEMI-IMPLICIT SPECTRAL COMPUTATIONS AND PREDICTOR-CORRECTOR SCHEMES IN THE CYCLE 46T1 OF ARPEGE/IFS.

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

December 19, 2018

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.

$*$ 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 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 may have an incremental formulation or a non-incremental formulation. The one which is coded in ARPEGE/IFS and its LAM version (for both hydrostatic and non-hydrostatic models) 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.

$*$ The different models described:

- 2D shallow-water model.
- 3D primitive equations model (denoted as HPE).
- 3D fully elastic NH model, with $\hat{Q}$ and d or d_4 as NH prognostic variables (denoted as NHEE).
- 3D quasi elastic NH model, with d_4 as NH prognostic variable, and $\hat{Q}$ as diagnostic variable (denoted as NHQE).

$*$ Other restrictions of this documentation:

- In the non-hydrostatic models, denotation d stands for vertical divergence if **NVDVAR**=3 and for d_4 if **NVDVAR**=4. In the NHEE system equations are valid for both **NVDVAR**=3 and **NVDVAR**=4. In the NHQE system equations are valid for **NVDVAR**=4 only.
- Atmosphere deep-layer effects are ignored in this documentation; radius is assumed to be equal to the mean radius denoted by a.

2 Denotations.

- L: number of layers of the model.
- 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.
- $\mathbf{V}$ 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; $D = M^2 * D^{'}$.
- ζ is the unreduced vorticity of horizontal wind, $\zeta^{'}$ is the reduced vorticity; $\zeta = M^2 * \zeta^{'}$.
- T is the temperature, $\tilde{T}$ is a modified temperature used in the NHQE model.
- T^* is a vertically-constant reference temperature which is used in the semi-implicit scheme.
- $T^*_{\rm a}$ is a cold vertically-variable reference temperature which is used in the semi-implicit scheme in the NH vertical divergence equation; it is recommended to have $T^*_{\rm a}$ lower than the current temperature.
- q is the humidity.
- Π is the hydrostatic pressure, $\Pi_{\rm s}$ is the hydrostatic surface pressure.
- A, B define hydrostatic pressure on the η levels: $\Pi = A + B\Pi_{\rm s}$.
- Π^* is a reference hydrostatic pressure and ${\Pi_{\rm s}}^*$ is a reference hydrostatic surface pressure, which are used in the semi-implicit scheme. These reference quantities are vertically dependent and "horizontally" (i.e. on η surfaces) constant. $\Delta\Pi^*$ are layer depths corresponding to a surface hydrostatic pressure equal to ${\Pi_{\rm s}}^*$.

- Π_{sst} is a reference hydrostatic pressure equal to the surface pressure of the standard atmosphere (variable **VP00**). Default value is 101325 Pa.
- $\Phi = gz$ is the geopotential height. Φ_{s} is the surface geopotential (i.e. the orography).
- $\mathbf{\Omega}$ 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.
- R is the gas constant for air and R_{d} the gas constant for dry air.
- c_{p} and $c_{\mathrm{p_d}}$ are respectively the specific heat at constant pressure for moist air and dry air.
- c_{v} and $c_{\mathrm{v_d}}$ are respectively the specific heat at constant volume for moist air and dry air.
- κ and κ_{d} are respectively R/c_{p} and $R_{\mathrm{d}}/c_{\mathrm{p_d}}$.
- ∇ is the unreduced first order horizontal gradient on η-surfaces; $\nabla^{'}$ is the reduced first order horizontal gradient; $\nabla = M * \nabla^{'}$.
- Non-hydrostatic variables:
  - p is the total pressure, p_{s} is the surface total pressure.
  - $\hat{Q}$ is the pressure departure variable. Expression of $\hat{Q}$ is $\hat{Q} = \log \frac{p}{\Pi}$.
  - d is the vertical divergence.
  - Variable $d_4 = d + \mathtt{X}$ can be also used as prognostic variable.
- β: tunable coefficient for the semi-implicit scheme (between 0 and 1).
- γ, τ, ν, μ, $\mathbf{G}^*$, $\mathbf{S}^*$, $\mathbf{N}^*$ are generic denotations for linear operators (see section 4).
- The NHQE model also uses the vertical integral operator $\mathbf{S}^*_\kappa$, and its inverse $\mathbf{S}^{*-1}_\kappa$.
- $\mathbf{L}^*$, $\mathbf{L}^{**}$, $\mathbf{Q}^*$, $\mathbf{T}^*$ and $\mathbf{T}^{**}$ are generic denotations for linear operators used in non-hydrostatic models (see section 4).
- The NHQE model also uses the vertical derivative operators $\mathbf{L}^*_\kappa$ and $\mathbf{L}^{**}_\kappa$ and there inverses.
- H, C, N are intermediate constants used in the semi-implicit scheme of the NHEE and NHQE models. Definitions are respectively:

$$H = (R_{\mathrm{d}} T^*)/g \tag{1}$$

$$C = \sqrt{R_{\mathrm{d}} T^* c_{\mathrm{p_d}}/c_{\mathrm{v_d}}} \tag{2}$$

$$N = g/\sqrt{c_{\mathrm{p_d}} T^*} \tag{3}$$

- 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.

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

$$\frac{\partial X}{\partial t} = -\mathbf{U}.\nabla_3 X + \mathcal{A} + \mathcal{F} \tag{4}$$

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.

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

$$\frac{dX}{dt} = \mathcal{A} + \mathcal{F} \tag{5}$$

In a three-time level semi-Lagrangian scheme (abbreviated into SL3TL 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 SL2TL 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.

$\ast$ **Adding of a semi-implicit correction S:** 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 (4) and (5) become respectively (6) and (7):

- Eulerian scheme:

$$\frac{\partial X}{\partial t} = -\mathbf{U}.\nabla_3 X + \mathcal{A} + \mathcal{F} + \mathcal{S} \tag{6}$$

- Semi-Lagrangian scheme:

$$\frac{dX}{dt} = \mathcal{A} + \mathcal{F} + \mathcal{S} \tag{7}$$

$\ast$ **Discretisation of equations (6) and (7):** Equations (6) and (7) give the following discretized equations, where Δt is the time step, $\mathcal{L}$ 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); more details can be found in documentations (IDEUL) and (IDSL):

- Eulerian scheme (all computations are done at the same grid point):

$$\mathcal{S} = -\beta\mathcal{L}^{t} + 0.5\beta\mathcal{L}^{t-\Delta t} + 0.5\beta\mathcal{L}^{t+\Delta t} \tag{8}$$

$$X^{t+\Delta t} - \beta\Delta t\mathcal{L}^{t+\Delta t} = X^{t-\Delta t} + 2\Delta t(-\mathbf{U}.\nabla_3 X + \mathcal{A} + \mathcal{F}) - 2\beta\Delta t\mathcal{L}^{t} + \beta\Delta t\mathcal{L}^{t-\Delta t} \tag{9}$$

- Three-time level semi-Lagrangian (SL3TL) scheme (without uncentering factor):

$$\mathcal{S} = -\beta\mathcal{L}^{t} + 0.5\beta\mathcal{L}^{t-\Delta t} + 0.5\beta\mathcal{L}^{t+\Delta t} \tag{10}$$

$$X^{t+\Delta t} - \beta\Delta t\mathcal{L}^{t+\Delta t} = X^{t-\Delta t} + 2\Delta t(\mathcal{A} + \mathcal{F}) - 2\beta\Delta t\mathcal{L}^{t} + \beta\Delta t\mathcal{L}^{t-\Delta t} \tag{11}$$

  where $X^{t+\Delta t} - \beta\Delta t\mathcal{L}^{t+\Delta t}$ is computed at the final grid point of the semi-Lagrangian trajectory, $X^{t-\Delta t}$ and $\beta\Delta t\mathcal{L}^{t-\Delta t}$ are computed at the origin point of the semi-Lagrangian trajectory, $-2\beta\Delta t\mathcal{L}^{t}$ 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.

- Two-time level semi-Lagrangian (SL2TL) scheme (without uncentering factor):

$$\mathcal{S} = -\beta\mathcal{L}^{t+0.5\Delta t} + 0.5\beta\mathcal{L}^{t} + 0.5\beta\mathcal{L}^{t+\Delta t} \tag{12}$$

$$X^{t+\Delta t} - 0.5\beta\Delta t\mathcal{L}^{t+\Delta t} = X^{t} + \Delta t(\mathcal{A} + \mathcal{F}) - \beta\Delta t\mathcal{L}^{t+0.5\Delta t} + 0.5\beta\Delta t\mathcal{L}^{t} \tag{13}$$

  where $X^{t+\Delta t} - 0.5\beta\Delta t\mathcal{L}^{t+\Delta t}$ is computed at the final grid point of the semi-Lagrangian trajectory, X^{t} and $0.5\beta\Delta t\mathcal{L}^{t}$ are computed at the origin point of the semi-Lagrangian trajectory, $-\beta\Delta t\mathcal{L}^{t+0.5\Delta t}$ and $\mathcal{A}$ are computed either at the medium point or as an average between the origin and final points of the trajectory.

$\mathcal{L}^{t+0.5\Delta t}$, $\mathcal{L}^{t}$ and $\mathcal{L}^{t-\Delta t}$ are computed in grid point space. The right-hand side members of equations (9), (11) and (13) 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:

$$X^{t+\Delta t} - \beta\Delta t\mathcal{L}^{t+\Delta t} = \mathcal{X}^* \tag{14}$$

for a leap-frog scheme, and:

$$X^{t+\Delta t} - 0.5\beta\Delta t\mathcal{L}^{t+\Delta t} = \mathcal{X}^* \tag{15}$$

for a two-time level semi-Lagrangian scheme, where $\mathcal{X}^*$ is known and $X^{t+\Delta t}$ is unknown.

3.3 Iterative centred-implicit schemes.

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.

$*$ **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.
- Iterations $(i > 0)$: the i-th iteration $(i > 0)$ computes $X^{t+\Delta t}_{(i)}$ (after inversion of Helmholtz equation) knowing $X^{t+\Delta t}_{(i-1)}$. The final value of $X^{t+\Delta t}$ is equal to $X^{t+\Delta t}_{(i=N_{\text{siter}})}$.
- Horizontal diffusion is always done at the last iteration, it can be done optionally at the other iterations.
- This scheme is controlled by the key **LPC_FULL**=.T. .
- 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 SL2TL scheme, $(\mathcal{A} - \beta\Delta t\mathcal{L})^{t+0.5\Delta t}_{(i=0)}$ is assumed to be equal to $(\mathcal{A} - \beta\Delta t\mathcal{L})^{t}$ if no extrapolation is done (case **LNESC**=.T.), and to a combination of $(\mathcal{A} - \beta\Delta t\mathcal{L})^{t}$ and $(\mathcal{A} - \beta\Delta t\mathcal{L})^{t-\Delta t}$ if extrapolation is done. For a SL2TL case with no uncentering factor that yields the following discretisations (physics is assumed to be unlagged):
  - no extrapolation:

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

    ($O(i = 0)$ and F are respectively the origin and final points of the semi-Lagrangian trajectory), which can be rewritten:

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

  - extrapolation: discretisation is identical to the case with no iterative centred-implicit scheme and extrapolation; 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):

    $$X^{t+\Delta t}_{(i)} - \Delta t\beta\mathcal{L}^{t+\Delta t}_{(i)}$$

    $$= X^{t-\Delta t} + 2\Delta tADV^t + [\Delta t\mathcal{A}^{t+\Delta t}_{(i-1)} - \Delta t\beta\mathcal{L}^{t+\Delta t}_{(i-1)}] + [\Delta t\mathcal{A}^{t-\Delta t} - \Delta t\beta\mathcal{L}^{t-\Delta t}] + \Delta t\beta\mathcal{L}^{t-\Delta t} + 2\Delta t\mathcal{F}^{t-\Delta t}$$

    which can be rewritten:

    $$X^{t+\Delta t}_{(i)} - \Delta t\beta\mathcal{L}^{t+\Delta t}_{(i)} = \Delta t\mathcal{A}^{t+\Delta t}_{(i-1)} - \Delta t\beta\mathcal{L}^{t+\Delta t}_{(i-1)} + [X^{t-\Delta t} + 2\Delta tADV^t + \Delta t\mathcal{A}^{t-\Delta t} + 2\Delta t\mathcal{F}^{t-\Delta t}]$$

– SL3TL (without uncentering factor):

$$[X_{(i)}^{t+\Delta t} - \Delta t\beta\mathcal{L}_{(i)}^{t+\Delta t}]_F$$

$$= X_{O(i)}^{t-\Delta t} + [\Delta t\mathcal{A}_{(i-1)}^{t+\Delta t} - \Delta t\beta\mathcal{L}_{(i-1)}^{t+\Delta t}]_F + [\Delta t\mathcal{A}^{t-\Delta t} - \Delta t\beta\mathcal{L}^{t-\Delta t}]_{O(i)} + [\Delta t\beta\mathcal{L}^{t-\Delta t} + 2\Delta t\mathcal{F}^{t-\Delta t}]_{O(i)}$$

which can be rewritten:

$$[X_{(i)}^{t+\Delta t} - \Delta t\beta\mathcal{L}_{(i)}^{t+\Delta t}]_F = X_{O(i)}^{t-\Delta t} + [\Delta t\mathcal{A}_{(i-1)}^{t+\Delta t} - \Delta t\beta\mathcal{L}_{(i-1)}^{t+\Delta t}]_F + [\Delta t\mathcal{A}^{t-\Delta t} + 2\Delta t\mathcal{F}^{t-\Delta t}]_{O(i)}$$

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. $O(i-1)$).

– SL2TL (without uncentering factor):

$$[X_{(i)}^{t+\Delta t} - 0.5\Delta t\beta\mathcal{L}_{(i)}^{t+\Delta t}]_F$$

$$= X_{O(i)}^{t} + [0.5\Delta t\mathcal{A}_{(i-1)}^{t+\Delta t} - 0.5\Delta t\beta\mathcal{L}_{(i-1)}^{t+\Delta t}]_F + [0.5\Delta t\mathcal{A}^{t} - 0.5\Delta t\beta\mathcal{L}^{t}]_{O(i)} + [0.5\Delta t\beta\mathcal{L}^{t} + \Delta t\mathcal{F}^{t}]_{O(i)}$$

which can be rewritten:

$$[X_{(i)}^{t+\Delta t} - 0.5\Delta t\beta\mathcal{L}_{(i)}^{t+\Delta t}]_F = X_{O(i)}^{t} + [0.5\Delta t\mathcal{A}_{(i-1)}^{t+\Delta t} - 0.5\Delta t\beta\mathcal{L}_{(i-1)}^{t+\Delta t}]_F + [0.5\Delta t\mathcal{A}^{t} + \Delta t\mathcal{F}^{t}]_{O(i)}$$

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. $O(i-1)$).

$*$ **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. Most parts of this documentation remain valid, the main differences with ARPEGE/IFS are:

- The LAM shallow-water model is not coded.
- Option **LIMPF**=.T. is not coded.
- Option **LESIDG**=.T. replaces **LSIDG**=.T., and is available only with the tilted-rotated Mercator projection. See (IDESIDG) about its implementation. But **LESIDG**=.T. code is currently incomplete and does not work.
- Some spectral calculations are done in routines named **ESP..SI** which are LAM counterparts of **SP..SI** routines. 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 (VFE).

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 $\mathcal{R}_{\rm inte}$) 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; one can use mixed configurations using VFE for vertical integrals and VFD (finite differences) for vertical derivatives.

4 Prognostic variables and quantities involved in the semi-implicit scheme.

4.1 Prognostic variables, and some denotations.

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 NHEE 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).

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

- VFD0: finite differences (**LVERTFE**=.F.) using **NDLNPR**=0.
- VFD1: finite differences (**LVERTFE**=.F.) using **NDLNPR**=1.
- VFD2: finite differences (**LVERTFE**=.F.) using **NDLNPR**=2.
- VFD: finite differences (**LVERTFE**=.F.), any value for **NDLNPR**.
- VFE: vertical finite elements (**LVERTFE**=.T.).

4.2 Operators "alpha", "delta" and Q*.

These operators are used for discretisations of some vertical integrals. They have a different expression according to the value of variables **NDLNPR**, **LVERTFE**.

- VFD0:
  - 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:

$$\alpha^*_l = 1 - (\Pi^*_{\bar{l}-1}/\Delta\Pi^*_l)\log(\Pi^*_{\bar{l}}/\Pi^*_{\bar{l}-1}) \tag{16}$$

$$\delta^*_l = \log(\Pi^*_{\bar{l}}/\Pi^*_{\bar{l}-1}) \tag{17}$$

  - 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.

- VFD1:
  - 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:

$$\alpha^*_l = 1 - \sqrt{\Pi^*_{\bar{l}-1}/\Pi^*_{\bar{l}}} = 1 - \Pi^*_l/\Pi^*_{\bar{l}} \tag{18}$$

$$\delta^*_l = \Delta\Pi^*_l/\Pi^*_l = \Delta\Pi^*_l/\sqrt{\Pi^*_{\bar{l}-1}\Pi^*_{\bar{l}}} \tag{19}$$

  Π^* is discretised as follows at full levels:

$$\Pi^*{}_l = \sqrt{\Pi^*{}_{\bar{l}}\Pi^*{}_{\bar{l}-1}} \tag{20}$$

- For the layer $l = 1$ if the pressure at the top of the model is zero:
  - $\alpha^*_{l=1} = 1$ and $\alpha^*_{\bar{l}=0} = 1$.
  - $\delta^*_{l=1} = 1 + \kappa_\mathrm{d}$.
  - $\Pi^*_{l=1} = \Delta\Pi^*_{l=1}/\delta^*_{l=1}$.

- VFD2: the only difference with VFD1 is the discretisation of $\delta^*_{l=1}$.

$$\delta^*_{l=1} = 1 + \delta^*_{l=2}(\Delta\Pi^*_{l=1}/\Delta\Pi^*_{l=2})\sqrt{\Pi^*_{\bar{l}=2}/\Pi^*_{\bar{l}=1}}$$

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

$$\alpha^*_l = (\Pi^*_{\bar{l}} - \Pi^*_l)/\Pi^*_l \tag{21}$$

$$\delta^*_l = \Delta\Pi^*_l/\Pi^*_l \tag{22}$$

  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 (21) and (22) 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 half-level data.

$\mathbf{Q}^*$ is a diagonal matrix: diagonal coefficients are equal to $\delta^* - 2\alpha^*$ for levels 2 to L, and 0 for level 1.

4.3 Vertical integrals and linear products.

These operators are used in both hydrostatic and non-hydrostatic models. Non linear counterparts of some of these vertical integrals are described in appendix 1 of documentation (IDEUL).

4.3.1 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: $\int_\eta^1 \frac{1}{\Pi^*}\frac{\partial\Pi^*}{\partial\eta}Zd\eta$
- $\gamma Z = R_\mathrm{d}(\mathbf{G}^*Z)$
- VFE expression of this discretisation is:

$$(\mathbf{G}^*Z)_l = [\mathcal{R}_\mathrm{inte}]_{(l,surf)}\left\langle\frac{Z\delta^*}{\Delta\eta}\right\rangle \tag{23}$$

- VFD expression of this discretisation is:

$$(\mathbf{G}^*Z)_l = \alpha^*_l Z_l + \sum_{k=l+1}^{L} Z_k\delta^*_k \tag{24}$$

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

4.3.2 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: $\frac{1}{\Pi^*}\int_0^\eta \frac{\partial\Pi^*}{\partial\eta}Zd\eta$
- $\tau Z = [(R_\mathrm{d}T^*)/c_{\mathrm{p_d}}](\mathbf{S}^*Z)$ if **LSPRT**=.F.
- $\tau Z = [(R_\mathrm{d}^2T^*)/(Rc_{\mathrm{p_d}})](\mathbf{S}^*Z)$ if **LSPRT**=.T. (equivalent to apply a linear operator in a system of equations using virtual temperature).
- VFE expression of this discretisation is:

$$(\mathbf{S}^*Z)_l = \frac{\delta^*_l}{\Delta\Pi^*_l}[\mathcal{R}_\mathrm{inte}]_{(top,l)}\left\langle\frac{\Delta\Pi^*Z}{\Delta\eta}\right\rangle \tag{25}$$

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

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

- VFD expression of this discretisation is:

$$(\mathbf{S}^* Z)_l = \left[\alpha_l^* Z_l + \frac{\delta_l^*}{\Delta\Pi_l^*} \sum_{k=1}^{l-1} \Delta\Pi_k^* Z_k \right] \tag{26}$$

The NHQE model uses the linear operator $\mathbf{S}_\kappa^*$ defined as:

$$\mathbf{S}_\kappa^* = \mathtt{I} - (1 - \kappa_\mathrm{d})\mathbf{S}^* = \mathtt{I} - \frac{1/\kappa_\mathrm{d} - 1}{T^*}\tau \tag{27}$$

and in some cases its inverse $\mathbf{S}_\kappa^{*-1}$.

4.3.3 Linear operator "ν" , and its dimensionless counterpart $\mathbf{N}^*$.

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:

$$(\mathbf{N}^* Z) = \frac{1}{\Pi_\mathrm{s}{}^*} [\mathcal{R}_\mathrm{inte}]_{(top,surf)} \left\langle \frac{\Delta\Pi^* Z}{\Delta\eta} \right\rangle \tag{28}$$

- VFD expression of this discretisation is:

$$(\mathbf{N}^* Z) = \frac{1}{\Pi_\mathrm{s}{}^*} \sum_{l=1}^{L} \Delta\Pi_l^* Z_l \tag{29}$$

4.3.4 Linear operator "μ", and its dimensionless counterpart $\mathtt{I}$.

This operator is applied to $\log(\Pi_\mathrm{s})$ to compute linear term in momentum equation.

- $\mathtt{I}$ 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).

4.3.5 Definition of constraint C1 in the NHEE model.

Continuous equations ensure the following identity:

$$\mathbf{G}^*\mathbf{S}^* - \mathbf{S}^* - \mathbf{G}^* + \mathbf{N}^* = 0 \tag{30}$$

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 matches equation (30).

- VFD0, VFD2 and VFE: constraint C1 is not ensured.
- VFD1: constraint C1 is ensured.

In the NHEE model, constraint C1 SHOULD be ensured if one wants the complete elimination of variables in order to provide a "one variable" Helmholtz equation with vertical divergence as unknown. This is why the VFD1 is preferred to VFD0 when using VFD discretisation. For VFE, Helmholtz equation treatment requires adaptations to take account of the fact that constraint C1 is not ensured. Elimination of equations can also be done in order to obtain a "one variable" Helmholtz equation with horizontal divergence as unknown: in this case elimination of variables is possible even if constraint C1 is not ensured.

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

$$COR = \frac{c_\mathrm{vd}}{R_\mathrm{d}^2 T^*}\gamma\tau - \frac{c_\mathrm{vd}}{R_\mathrm{d} c_\mathrm{pd}}\gamma - \frac{c_\mathrm{vd}}{R_\mathrm{d} T^*}\tau + \frac{c_\mathrm{vd}}{c_\mathrm{pd}}\nu = \frac{c_\mathrm{vd}}{c_\mathrm{pd}}[\mathbf{G}^*\mathbf{S}^* - \mathbf{S}^* - \mathbf{G}^* + \mathbf{N}^*] \tag{31}$$

4.4 Vertical derivatives and mixed operators for NHEE model.

4.4.1 Linear operator "∂^*".

It is defined by:

$$\partial^* Z = \Pi^* \frac{\partial Z}{\partial \Pi^*}$$

4.4.2 Linear operators $\mathbf{L}^*$ and $\mathbf{L}^{**}$.

This operator, called Laplacian operator, is applied to the pressure departure variable to compute linear term in the vertical divergence equation (NHEE model).

- For a variable Z, $(\mathbf{L}^* Z)$ is the vertical double derivative $\partial^*(\partial^* + \mathtt{I})Z = \left[\Pi^* \frac{\partial}{\partial \Pi^*}(\frac{\partial \Pi^* Z}{\partial \Pi^*})\right]$.
- VFD expression of this discretisation is:
  - Layers 2 to $L-1$:

$$(\mathbf{L}^* Z)_l = \mathbf{A}_l^* Z_{l-1} + \mathbf{B}_l^* Z_l + \mathbf{C}_l^* Z_{l+1} \tag{32}$$

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

$$\mathbf{A}_l^* = \frac{\Pi_{l-1}^*}{\delta_l^*(\Pi_l^* - \Pi_{l-1}^*)} \tag{33}$$

$$\mathbf{B}_l^* = -\frac{1}{\delta_l^*}\left(\frac{\Pi_l^*}{\Pi_l^* - \Pi_{l-1}^*} + \frac{\Pi_l^*}{\Pi_{l+1}^* - \Pi_l^*}\right) \tag{34}$$

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

Note that equation (32) can be rewritten:

$$(\mathbf{L}^* Z)_l = \mathbf{A}_l^*(Z_{l-1} - Z_l) + \mathbf{C}_l^*(Z_{l+1} - Z_l) \tag{36}$$

  - 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:

$$\mathbf{A}_1^* = \frac{\Pi_{\mathrm{top}}^*}{\delta_1^*(\Pi_1^* - \Pi_{\mathrm{top}}^*)} \tag{37}$$

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

$$\mathbf{C}_1^* = \frac{\Pi_2^*}{\delta_1^*(\Pi_2^* - \Pi_1^*)} \tag{38}$$

and, if the top hydrostatic pressure is zero:

$$\mathbf{B}_1^* = -\mathbf{C}_1^* \tag{39}$$

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

$$(\mathbf{L}^* Z)_1 = -\mathbf{A}_1^* Z_1 + \mathbf{C}_1^*(Z_2 - Z_1) \tag{40}$$

  - 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:

$$\mathbf{A}_L^* = \frac{\Pi_{L-1}^*}{\delta_L^*(\Pi_L^* - \Pi_{L-1}^*)} \tag{41}$$

Application of formula (36) would lead to $\mathbf{B}_L^* = -\mathbf{A}_L^*$ but we actually use the formula (32) with a slightly different expression for $\mathbf{B}_L^*$:

$$\mathbf{B}_L^* = -\frac{\Pi_L^*}{\delta_L^*(\Pi_L^* - \Pi_{L-1}^*)} \tag{42}$$

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

$$(\mathbf{L}^* Z)_L = \mathbf{A}_L^*(Z_{L-1} - Z_L) - \mathbf{A}_L^*(\Pi_L^*/\Pi_{L-1}^* - 1)Z_L \tag{43}$$

- VFE expression of this discretisation in the NHEE model: this topic is still in progress, at least two solutions have been implemented:
  - one option uses a second-order derivative operator $[\mathcal{R}_{\mathrm{dderi}}]$ (case **LVFE_LAPL_HALF**=F).
  - one option mixes a first-order VFE derivative operator $[\mathcal{R}_{\mathrm{deri}}]$ and a first-order VFD derivative operator (case **LVFE_LAPL_HALF**=T).
- Remarks for NHEE model:
  - stability is ensured only if the top hydrostatic pressure is zero.
  - VFD1 expressions have been chosen in order to match "constraint C2" (definition of "constraint C2" will be given below, in paragraph explaining linear operator $\mathbf{T}^*$).
  - There is a "weak constraint C2" notion for $\mathbf{L}^*$ saying that $\mathbf{L}^*$ is a tridiagonal operator.

To use a vertical dependent T^*_{a} and to make formulae simpler, it is more convenient to use $\mathbf{L}^{**}$:

$$\mathbf{L}^{**} = (T^*/T^*_{\mathrm{a}})\mathbf{L}^*$$

4.4.3 Linear operators $\mathbf{T}^*$ and $\mathbf{T}^{**}$, and definition of "constraint C2".

Operators $\mathbf{T}^*$ and $\mathbf{T}^{**}$ appear only when Helmholtz equation has vertical divergence as unknown.
General definition of $\mathbf{T}^*$ uses a combination of linear operators $\mathbf{L}^*$, τ and γ.

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

$$\mathbf{T}^* = \frac{g^2\mathbf{L}^*\left(\frac{c_{\mathrm{p_d}}}{R_{\mathrm{d}}T^*}\tau\gamma - \frac{c_{\mathrm{p_d}}^2}{c_{\mathrm{v_d}}T^*}\tau - \frac{c_{\mathrm{p_d}}}{c_{\mathrm{v_d}}}\gamma\right)}{R_{\mathrm{d}}N^2C^2} \tag{44}$$

Using definitions for C and H (see equations (2) and (1)) this equation can be rewritten:

$$\mathbf{T}^* = \frac{\mathbf{L}^*\left(T^*c_{\mathrm{p_d}}\tau\gamma - c_{\mathrm{p_d}}C^2\tau - C^2T^*\gamma\right)}{H^2N^2C^2} \tag{45}$$

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

$$\mathbf{T}^* = \frac{g^2\mathbf{L}^*\left(\mathbf{S}^*\mathbf{G}^* - (c_{\mathrm{p_d}}/c_{\mathrm{v_d}})\mathbf{S}^* - (c_{\mathrm{p_d}}/c_{\mathrm{v_d}})\mathbf{G}^*\right)}{N^2C^2} \tag{46}$$

We can now introduce definition of "constraint C2":

- Strong constraint C2: $\mathbf{T}^*$ is the identity matrix.
- Weak constraint C2: $\mathbf{T}^*$ matches (47):

  $$\mathbf{T}^* = (I + \mathbf{L}^*\mathbf{Q}^*) \tag{47}$$

  and does not depart too much from identity matrix.

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.

- VFD0, VFD2 and VFE: constraint C2 is not ensured.
- VFD1: 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 (44)).
- Discretisations of $\mathbf{T}^*$ matching constraint C2 or close to constraint C2 are preferred, in order to ensure numerical stability.

Discretisation:

- VFD1: constraint C2 is ensured, and formula (47) is used. In this case, $\mathbf{T}^*$ is a tri-diagonal operator, like $\mathbf{L}^*$. Expression of the elements of the associated matrix is:

$$\mathbf{T}^*{}_{(l,l)} = 1 - \frac{1}{\delta_l^*}\left(\frac{\Pi_l^*}{\Pi_l^* - \Pi_{l-1}^*} + \frac{\Pi_l^*}{\Pi_{l+1}^* - \Pi_l^*}\right)(\delta_l^* - 2\alpha_l^*) \tag{48}$$

$$\mathbf{T}^*{}_{(1,1)} = 1 \tag{49}$$

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

$$\mathbf{T}^*{}_{(l,l+1)} = \frac{1}{\delta_l^*}\left(\frac{\Pi_{l+1}^*}{\Pi_{l+1}^* - \Pi_l^*}\right)(\delta_{l+1}^* - 2\alpha_{l+1}^*) \tag{51}$$

  with some particular expressions at the top and the bottom.
- VFD0, VFD2 and VFE: constraint C2 is not ensured. Formula (44) is used (use VFE or VFD0 discretisations of γ, τ, $\mathbf{L}^*$).

To use a vertical dependent T_{a}^* and to make formulae simpler, it is more convenient to use $\mathbf{T}^{**}$:

$$\mathbf{T}^{**} = (T^*/T_{\mathrm{a}}^*)\mathbf{T}^*$$

4.5 Vertical derivatives and mixed operators for NHQE model.

4.5.1 Definition of constraint C2.

Definition is slightly different from the NHEE one. The weak C2 constraint says that operators like $\mathbf{L}^*$, $\mathbf{L}_\kappa^*$, $\mathbf{L}_\kappa^{**}$ are tridiagonal ones. When C2 constraint is matched, there are some properties between $\mathbf{L}^*$, $\mathbf{L}_\kappa^*$, $\mathbf{S}_\kappa^*$ which allow to compute for example $\mathbf{L}_\kappa^{**-1}$ and $\mathbf{L}_\kappa^{**-1}\mathbf{S}_\kappa^*$ without inverting matrices.

Constraint C2 requires VFD2 discretisation.

4.5.2 Linear operator ∂^*.

It is defined by:

$$\partial^* Z = \Pi^* \frac{\partial Z}{\partial \Pi^*}$$

Its calculation may be required in a subset of cases. VFD discretisation applied to a full level quantity Z is:

$$[\partial^* Z]_l = \alpha_l^*(\mathbf{A}_l^* Z_{l-1} + \mathbf{B}_l^* Z_l + \mathbf{C}_l^* Z_{l+1}) + \delta_l^* \mathbf{A}_l^*(Z_l - Z_{l-1}) \tag{52}$$

Quantities $\mathbf{A}^*$, $\mathbf{B}^*$ and $\mathbf{C}^*$ are those used in calculation of $\mathbf{L}^*$ (they match $\mathbf{A}^* + \mathbf{B}^* + \mathbf{C}^* = 0$).

- For layers 2 to $L-1$ this is equivalent to write:

$$[\partial^* Z]_l = (\delta_l^* - \alpha_l^*)\mathbf{A}_l^*(Z_l - Z_{l-1}) + \alpha_l^* \mathbf{C}_l^*(Z_{l+1} - Z_l)$$

- For layer 1: $\mathbf{A}^*$, $\mathbf{B}^*$ and $\mathbf{C}^*$ have specific values (see below), Z_0 is replaced by 0.
- For layer L: $\mathbf{A}^*$, $\mathbf{B}^*$ and $\mathbf{C}^*$ have specific values (see below), Z_{L+1} is replaced by 0.

Applying ∂^* at half level quantities is simpler:

$$[\partial^* Z]_l = \frac{Z_{\bar{l}} - Z_{\bar{l}-1}}{\delta_l} \tag{53}$$

Remarks:

- VFD2: calculation is not necessary. The two possible ways to compute $\mathbf{L}_\kappa^*$ (using and not using ∂^*) must give the same result, provided formula (52) is used.
- VFD for vertical derivatives, other cases (for example VFE) for vertical integrals: study is still in progress, use formula (52) for the time being.
- VFE for vertical derivatives: this case has not been yet studied nor coded.

4.5.3 Linear operator $\mathbf{L}^*$.

For a variable Z, $(\mathbf{L}^* Z)$ is the vertical double derivative $\partial^*(\partial^* + \mathtt{I})Z = \left[\Pi^* \frac{\partial}{\partial \Pi^*}(\frac{\partial \Pi^* Z}{\partial \Pi^*})\right]$.

NHQE discretisation slightly differs from the NHEE discretisation near the top and bottom.

Discretisation:

- VFD expression of this discretisation is:
  - Layers 3 to $L-1$: formulae (32), (33), (34), (35), (36) remain valid.
  - Layers 2: formulae (36) and (35) remain valid; $\mathbf{A}_2^*$ has a specific NHQE discretisation.

$$\mathbf{A}_2^* = \frac{\Pi_1^*}{\delta_2^*(\Pi_2^*/\delta_1^* - \Pi_1^*(\alpha_2^* - \delta_2^*))} \tag{54}$$

  - Layer 1: formulae (40) and (37) remain valid; $\mathbf{C}_1^*$ has a specific NHQE discretisation.

$$\mathbf{C}_1^* = \frac{\Pi_2^*}{\Pi_2^* - \Pi_1^*\delta_1^*(\alpha_2^* - \delta_2^*)} \tag{55}$$

  - Layer L:

$$(\mathbf{L}^* Z)_L = \mathbf{A}_L^*(Z_{L-1} - Z_L) - \mathbf{C}_L^* Z_L \tag{56}$$

    $\mathbf{A}_L^*$ matches the general expression:

$$\mathbf{A}_L^* = \frac{\Pi_{L-1}^*}{\delta_L^*(\Pi_L^* - \Pi_{L-1}^*)} \tag{57}$$

    Discretisation of $\mathbf{C}_L^*$ writes:

$$\mathbf{C}_L^* = \kappa_{\mathrm{d}} \frac{\Pi_{\mathrm{surf}}^*}{\delta_L^*(\kappa_{\mathrm{d}}\Pi_{\mathrm{surf}}^* + (1-\kappa_{\mathrm{d}})\Pi_L^*)} \tag{58}$$

- VFE expression of this discretisation in the NHQE model: this topic is still in progress, and no code is provided for the time being.
- Remark for NHQE model: stability is ensured only if the top hydrostatic pressure is zero.

4.5.4 Linear operator $\mathbf{L}_\kappa^*$.

This operator, is applied to the pressure departure variable to compute linear term in the vertical divergence equation.

For a variable Z, $(\mathbf{L}^* Z)$ is the vertical double derivative $\partial^*(\partial^* + \kappa_{\mathrm{d}}\mathtt{I})Z$.

$(\mathbf{L}^* Z)$ can also be written:

$$\mathbf{L}_\kappa^* Z = \mathbf{L}^* Z + (\kappa_{\mathrm{d}} - 1)\partial^* Z$$

Using VFD discretisation of $\mathbf{L}^*$ and ∂^*, VFD discretisation of $\mathbf{L}_\kappa^*$ (valid even if constraint C2 is not matched) writes:

$$[\mathbf{L}_\kappa^* Z]_l = [1 + (\kappa_{\mathrm{d}} - 1)(\alpha_l^* - \delta_l^*)]\mathbf{A}_l^*[Z_{l-1} - Z_l] + [1 + (\kappa_{\mathrm{d}} - 1)\alpha_l^*]\mathbf{C}_l^*[Z_{l+1} - Z_l]$$

For VFD2 case, and only in this case, $(\mathbf{L}^* Z)$ matches the following identity (constraint C2):

$$\mathbf{L}_\kappa^* Z = \mathbf{S}_\kappa^* \mathbf{L}^* Z$$

To use a vertical dependent T_{a}^* and to make formulae simpler, it is more convenient to use $\mathbf{L}_\kappa^{**}$:

$$\mathbf{L}_\kappa^{**} = (T^*/T_{\mathrm{a}}^*)\mathbf{L}_\kappa^*$$

4.5.5 Linear operator $\mathbf{S}_\kappa^{*-1}$.

Its discretisation is required only for case VFD2: we invert a triangular matrix.

We recall the VFD discretisation of $\mathbf{S}_\kappa^*$.

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

We assume that $Y_l = (\mathbf{S}_\kappa^* Z)_l$ is known for every level l; the unknown is Z_l.

- Let us start by $Z_{l=1}$, using:
$$(\mathbf{S}^*_\kappa Z)_{l=1} = Z_{l=1} + (\kappa_\mathrm{d} - 1)\alpha^*_{l=1} Z_{l=1}$$
Using identity $\alpha^*_{l=1} = 1$, that yields:
$$Z_{l=1} = (1/\kappa_\mathrm{d})(\mathbf{S}^*_\kappa Z)_{l=1}$$
- To compute Z_l one assumes that Z_k has been computed for k from 1 to $l-1$; on starts from:
$$(\mathbf{S}^*_\kappa Z)_l = Z_l + (\kappa_\mathrm{d} - 1)\left[\alpha^*_l Z_l + \frac{\delta^*_l}{\Delta\Pi^*_l}\sum_{k=1}^{l-1}\Delta\Pi^*_k Z_k\right]$$
which can be rewritten:
$$Z_l = \frac{1}{1 + (\kappa_\mathrm{d} - 1)\alpha^*_l}\left[(\mathbf{S}^*_\kappa Z)_l - (\kappa_\mathrm{d} - 1)\left(\frac{\delta^*_l}{\Delta\Pi^*_l}\sum_{k=1}^{l-1}\Delta\Pi^*_k Z_k\right)\right]$$

4.5.6 Linear operators $\mathbf{L}^{**-1}_\kappa$ and $\mathbf{L}^{**-1}_\kappa\mathbf{S}^*_\kappa$.

∗ **Case VFD2:** In this case we use properties matched with constraint C2 to compute them, without computing $\mathbf{L}^*_\kappa$.

The following properties are matched:

- $\mathbf{L}^{**}_\kappa$ is a tridiagonal matrix.
- $\mathbf{L}^*_\kappa$ is a tridiagonal matrix.
- $\mathbf{L}^{**}_\kappa = (T^*/T^*_\mathrm{a})\mathbf{L}^*_\kappa$ by définition.
- $\mathbf{L}^*_\kappa = \mathbf{S}^*_\kappa\mathbf{L}^*$ (constraint C2).
- $\mathbf{L}^{**}_\kappa = (T^*/T^*_\mathrm{a})\mathbf{S}^*_\kappa\mathbf{L}^*$ (constraint C2 and définition of $\mathbf{L}^{**}_\kappa$).
- But $\mathbf{L}^{**}$ is not equal to $(T^*/T^*_\mathrm{a})\mathbf{L}^*$, because (T^*/T^*_a) and $\mathbf{S}^*_\kappa$ are not commutative.
- $(c_{p_\mathrm{d}}/R^2_\mathrm{d}T^*)\mathbf{L}^*[\gamma\tau + R_\mathrm{d}T^*\nu] = -[\mathtt{I} + \mathbf{L}^*\mathbf{Q}^*]$ (constraint C2).
- Inverting above formulae yields:

  $\mathbf{L}^{*-1} = -((c_{p_\mathrm{d}}/R^2_\mathrm{d}T^*)[\gamma\tau + R_\mathrm{d}T^*\nu] + \mathbf{Q}^*)$ (constraint C2).

  $\mathbf{L}^{**-1}_\kappa = -((c_{p_\mathrm{d}}/R^2_\mathrm{d}T^*)[\gamma\tau + R_\mathrm{d}T^*\nu] + \mathbf{Q}^*)\mathbf{S}^{*-1}_\kappa(T^*_\mathrm{a}/T^*)$ (constraint C2).

  $\mathbf{L}^{**-1}_\kappa\mathbf{S}^*_\kappa = -((c_{p_\mathrm{d}}/R^2_\mathrm{d}T^*)[\gamma\tau + R_\mathrm{d}T^*\nu] + \mathbf{Q}^*)\mathbf{S}^{*-1}_\kappa(T^*_\mathrm{a}/T^*)\mathbf{S}^*_\kappa$ (constraint C2).
- Finally, computation of $\mathbf{S}^{*-1}_\kappa$ is needed to compute $\mathbf{L}^{**-1}_\kappa$ and $\mathbf{L}^{**-1}_\kappa\mathbf{S}^*_\kappa$.
- $\mathbf{S}^*_\kappa$ and $\mathbf{S}^{*-1}_\kappa$ are needed to compute $\mathbf{L}^{**-1}_\kappa\mathbf{S}^*_\kappa$.
- Some of these formulae can be simplified for vertically-constant T^*_a.
- Calculation of operators ∂^*, $\mathbf{L}^*$, $\mathbf{L}^*_\kappa$, $\mathbf{L}^{**}_\kappa$ is not necessary.

∗ **VFD discretisation for other cases:** That occurs for example when VFE are applied to vertical integrals, and VFD are applied to vertical derivatives. Constraint C2 is not matched, calculations follow a different way.

We use definition of $\mathbf{L}^{**}_\kappa$:
$$\mathbf{L}^{**}_\kappa Z = (T^*/T^*_\mathrm{a})\mathbf{L}^*Z + (\kappa_\mathrm{d} - 1)(T^*/T^*_\mathrm{a})\partial^*Z$$
VFD discretisation has been studied above for $\mathbf{L}^*Z$ and ∂^*Z.

We then invert $\mathbf{L}^{**}_\kappa$ by a linear algebra routine inverting matrices, to compute $\mathbf{L}^{**-1}_\kappa$. Applying a right multiplication by $\mathbf{S}^*_\kappa$ provides $\mathbf{L}^{**-1}_\kappa\mathbf{S}^*_\kappa$.

∗ **VFE discretisation:** This case has not been yet studied nor coded.

5 Semi-implicit scheme and inversion of Helmholtz equation.

5.1 Preliminary remarks and general formalism for Helmholtz equation.

Equations are written for a leap-frog scheme (Eulerian scheme or SL3TL scheme). For a SL2TL scheme replace Δt by $0.5\Delta t$.

Denotation d generally stands for d_4, in particuliar for the NHQE model.

Method for solving the semi-implicit system can be summarised as follows:

- Eliminate variables between the different equations, in order to obtain one equation (the Helmholtz equation) with one unknown. This unknown is generally $D'_{t+\Delta t}$, sometimes $d_{t+\Delta t}$. The RHS of the Helmholtz equation (denoted $\mathcal{R}$) is computed.
- Solve the Helmholtz equation: this is generally done by inversion of a diagonal linear system (unstretched global model, LAM model with small domain), or by inversion of a penta-diagonal linear system (for example stretched global model).
- Retrieve the other variables.

When $\overline{M} = M$, some calculations involve spectral multiplications by M^2. This is done by the product of a symmetric pentadiagonal matrix (useful coefficients are 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. **MXPTMA** is the routine doing the product. For more details see section 7.

If $D'_{t+\Delta t}$ is the unknown of the Helmholtz equation, the Helmholtz equation has the following shape:

$$(\mathtt{I} - \beta^2 \Delta t^2 \ \mathtt{B} \ \nabla'^2 \overline{M}^2) D'_{t+\Delta t} = \mathcal{R} \tag{59}$$

An alternate way to write it shows a symmetric matricial operator in the LHS:

$$(\nabla'^{-2} - \beta^2 \Delta t^2 \ \mathtt{B} \ \overline{M}^2) D'_{t+\Delta t} = \nabla'^{-2} \mathcal{R} \tag{60}$$

If $d_{t+\Delta t}$ is the unknown of the Helmholtz equation, the Helmholtz equation has the following shape:

$$(\mathtt{I} - \beta^2 \Delta t^2 \ \mathtt{B} \ \overline{M}^2 \nabla'^2) d_{t+\Delta t} = \mathcal{R} \tag{61}$$

An alternate way to write it shows a symmetric matricial operator in the LHS:

$$(\nabla'^{-2} - \beta^2 \Delta t^2 \ \mathtt{B} \ \overline{M}^2)[\nabla'^2 d_{t+\Delta t}] = \mathcal{R} \tag{62}$$

$\mathtt{B}$ may be timestep-dependent; it must be positive definite. To solve the above Helmholtz equations one uses the diagonalisation of $\mathtt{B}$. Algorithm works zonal wave number by zonal wave number m. For a given zonal wave number m:

- The right-hand side member $\mathcal{R}$ is computed for all total wave numbers n. For spherical geometry n varies between m and N_{s}.
- 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 $\mathtt{B}$, $\mathtt{Q}$, and the different linear operators defined in section 4 commute with the horizontal operators ∇'^2 and M^2. But M^2 does not commute with ∇'^2.
- Inversion of Helmholtz equation if $\overline{M} = c$ and if $D'_{t+\Delta t}$ is the unknown: algorithm is done according to the following steps:
  - Do a multiplication by $\mathtt{Q}$.
  - Helmholtz equation (59) becomes, for each eigenmode l:

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

  - 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: spectral coefficients of the right-hand side member of (63) are simply divided by the diagonal coefficients of this matrix. That yields $\mathtt{Q} D'_{t+\Delta t}$.
  - Multiplying by $\mathtt{Q}^{-1}$ one obtains $D'_{t+\Delta t}$.

- Inversion of Helmholtz equation if $\overline{M} = M$ and if $D'_{t+\Delta t}$ is the unknown: inversion of Helmholtz equation is more complicated 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.
  - Do a multiplication by `Q`.
  - Equations (59) and (60) respectively become:

$$(\mathtt{I} - \beta^2 \Delta t^2 a_l \nabla'^2 M^2)\mathtt{Q} D'_{t+\Delta t} = \mathtt{Q}\mathcal{R} \tag{64}$$

$$(\nabla'^{-2} - \beta^2 \Delta t^2 a_l M^2)\mathtt{Q} D'_{t+\Delta t} = \mathtt{Q}\nabla'^{-2}\mathcal{R} \tag{65}$$

  - Equation (65) is used for $m > 0$. 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. 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 $\mathtt{Q}D'_{t+\Delta t}$.
  - For the zonal wave number $m = 0$ equation (64) is used rather than (65) to avoid a division by 0 for $(m,n) = (0,0)$. 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**.
  - Multiplying by $\mathtt{Q}^{-1}$ one obtains $D'_{t+\Delta t}$.
- Inversion of Helmholtz equation if $\overline{M} = c$ and if $d_{t+\Delta t}$ is the unknown: algorithm is done according to the following steps:
  - Do a multiplication by `Q`.
  - Helmholtz equation (61) becomes, for each eigenmode l:

$$(\mathtt{I} - \beta^2 \Delta t^2 a_l \overline{M}^2 \nabla'^2)\mathtt{Q} d_{t+\Delta t} = \mathtt{Q}\mathcal{R} \tag{66}$$

  - 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: spectral coefficients of the right-hand side member of (66) are simply divided by the diagonal coefficients of this matrix. That yields $\mathtt{Q}d_{t+\Delta t}$.
  - Multiplying by $\mathtt{Q}^{-1}$ one obtains $d_{t+\Delta t}$.
- Inversion of Helmholtz equation if $\overline{M} = M$ and if $d_{t+\Delta t}$ is the unknown: inversion of Helmholtz equation is more complicated 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.
  - Do a multiplication by `Q`.
  - Equations (61) and (62) respectively become:

$$(\mathtt{I} - \beta^2 \Delta t^2 a_l \overline{M}^2 \nabla'^2)\mathtt{Q} d_{t+\Delta t} = \mathtt{Q}\mathcal{R} \tag{67}$$

$$(\nabla'^{-2} - \beta^2 \Delta t^2 a_l \overline{M}^2)\mathtt{Q}[\nabla'^2 d_{t+\Delta t}] = \mathtt{Q}\mathcal{R} \tag{68}$$

  - Equation (68) is used for $m > 0$. 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. 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 the useful coefficients of `L`, `U` are computed in the set-up routine **SUNHHEG** and stored in the array **SIHEG**. The right-hand side member of (68) 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 $\mathtt{Q}[\nabla'^2 d_{t+\Delta t}]$.
  - For the zonal wave number $m = 0$ equation (67) is used rather than (68) to avoid a division by 0 for $(m,n) = (0,0)$. 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 `L`, `U` are computed in the set-up routine **SUNHHEG** and stored in the arrays **SIHEG** and **SIHEG2**.
  - Multiplying by $\mathtt{Q}^{-1}$, then ∇'^{-2} (if $m > 0$), one obtains $d_{t+\Delta t}$.

5.2 Semi-implicit scheme for 3D hydrostatic primitive equations model.

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

- Continuity equation ($X = \log(\Pi_{\rm s})$):

$$\mathcal{L} = -\nu(\overline{M}^2 D^{'}) \tag{69}$$

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

$$\mathcal{L} = -\nabla^{'2}(\gamma T + \mu \log \Pi_{\rm s}) \tag{70}$$

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

$$\mathcal{L} = 0 \tag{71}$$

- Temperature equation ($X = T$):

$$\mathcal{L} = -\tau(\overline{M}^2 D^{'}) \tag{72}$$

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

$$\log(\Pi_{\rm s})_{t+\Delta t} + \beta\Delta t \overline{M}^2 \nu D^{'}_{t+\Delta t} = \mathcal{P}^* \tag{73}$$

$$D^{'}_{t+\Delta t} + \beta\Delta t \nabla^{'2}(\gamma T_{t+\Delta t} + \mu \log(\Pi_{\rm s})_{t+\Delta t}) = \mathcal{D}^{'*} \tag{74}$$

$$T_{t+\Delta t} + \beta\Delta t \overline{M}^2 \tau D^{'}_{t+\Delta t} = \mathcal{T}^* \tag{75}$$

$\mathcal{P}^*$, $\mathcal{D}^{'*}$, $\mathcal{T}^*$ correspond to $\mathcal{X}^*$ defined in equation (14) and are available in spectral arrays (**YDSP%SP**, **YDSP%DIV**, **YDSP%T**) at the beginning of the spectral computations. Equations (73) to (75) yield (76) (Helmholtz equation):

$$(\mathtt{I} - \beta^2 \Delta t^2\ \mathtt{B}\ \nabla^{'2} \overline{M}^2) D^{'}_{t+\Delta t} = \mathcal{R} \tag{76}$$

where $\mathtt{B} = \gamma\tau + \mu\nu$ is a matricial operator $L * L$ (precomputed in routines **SUDYN, SUBMAT** and stored in the array **SIB**), and:

$$\mathcal{R} = \mathcal{D}^{'*} - \beta\Delta t \nabla^{'2}(\gamma \mathcal{T}^* + \mu \mathcal{P}^*) \tag{77}$$

$*$ Spectral computations to solve system of equations (73) to (75). Algorithm works zonal wave number by zonal wave number m and is performed in the routine **SPCSI** (**ESPCSI** for LAM models). For a given zonal wave number m:

- After a preliminary memory transfer the right-hand side member of equation (76) is computed.
- Inversion of Helmholtz equation is done according to an algorithm described in subsection 5.1.
- Once known $D^{'}_{t+\Delta t}$ equation (73) provides $\log(\Pi_{\rm s})_{t+\Delta t}$ and equation (75) provides $T_{t+\Delta t}$.
- Semi-implicit scheme ends by a final memory transfer.

5.3 Semi-implicit scheme for 3D fully elastic nonhydrostatic primitive (NHEE) equations model: vertical divergence as unknown in Helmholtz equation.

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

- Continuity equation ($X = \log(\Pi_{\rm s})$):

$$\mathcal{L} = -\nu(\overline{M}^2 D^{'}) \tag{78}$$

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

$$\mathcal{L} = -\nabla^{'2}[\gamma T - T^*(\gamma \hat{Q}) + \mu \log(\Pi_{\rm s}) + R_{\rm d} T^* \hat{Q}] \tag{79}$$

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

$$\mathcal{L} = 0 \tag{80}$$

- Temperature equation ($X = T$):

$$\mathcal{L} = -\frac{R_{\rm d} T^*}{c_{\rm vd}}[\overline{M}^2 D^{'} + d] \tag{81}$$

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

$$\mathcal{L} = -\left[\frac{c_{\rm pd}}{c_{\rm vd}}(\overline{M}^2 D^{'} + d) - \frac{c_{\rm pd}}{R_{\rm d} T^*}\tau(\overline{M}^2 D^{'})\right] \tag{82}$$

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

$$\mathcal{L} = -\frac{g^2}{R_{\rm d} T^*}(\mathbf{L}^{**} \hat{Q}) \tag{83}$$

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

$$\log(\Pi_{\rm s})_{t+\Delta t} + \beta\Delta t\nu(\overline{M}^2 D'_{t+\Delta t}) = \mathcal{P}^* \tag{84}$$

$$D'_{t+\Delta t} + \beta\Delta t\nabla'^2[\gamma T_{t+\Delta t} - T^*(\gamma\hat{Q}_{t+\Delta t}) + \mu\log(\Pi_{\rm s})_{t+\Delta t} + R_{\rm d}T^*\hat{Q}_{t+\Delta t}] = \mathcal{D}'^* \tag{85}$$

$$\hat{Q}_{t+\Delta t} + \beta\Delta t\left[\frac{c_{\rm pd}}{c_{\rm vd}}(\overline{M}^2 D'_{t+\Delta t} + d_{t+\Delta t}) - \frac{c_{\rm pd}}{R_{\rm d}T^*}\tau(\overline{M}^2 D'_{t+\Delta t})\right] = \hat{\mathcal{Q}}^* \tag{86}$$

$$d_{t+\Delta t} + \beta\Delta t\frac{g^2}{R_{\rm d}T^*}(\mathbf{L}^{**}\hat{Q}_{t+\Delta t}) = \hat{\mathcal{D}}^* \tag{87}$$

$$T_{t+\Delta t} + \beta\Delta t\frac{R_{\rm d}T^*}{c_{\rm vd}}[\overline{M}^2 D'_{t+\Delta t} + d_{t+\Delta t}] = \mathcal{T}^* \tag{88}$$

$\mathcal{P}^*$, $\mathcal{D}'^*$, $\hat{\mathcal{Q}}^*$, $\hat{\mathcal{D}}^*$, $\mathcal{T}^*$ correspond to $\mathcal{X}^*$ defined in equation (14) and are available in spectral arrays (**YDSP%SP**, **YDSP%DIV**, **YDSP%SPD**, **YDSP%SVD**, **YDSP%T**) at the beginning of spectral computations.

$*$ **Elimination of T, $\hat{Q}$ and $\log(\Pi_{\rm s})$:** All the constants and the vertical operators commute with ∇'^2 and $\overline{M}$.

- Elimination of T, $\hat{Q}$ and $\log(\Pi_{\rm s})$ between equations (84), (85), (86) and (88) leads to equation (89):

$$\begin{aligned} D'_{t+\Delta t} - (\beta\Delta t)^2\nabla'^2[\{R_{\rm d}T^*(\tfrac{\gamma}{R_{\rm d}} - 1)(\tfrac{c_{\rm pd}}{R_{\rm d}T^*}\tau - \tfrac{c_{\rm pd}}{c_{\rm vd}}) + \tfrac{R_{\rm d}T^*}{c_{\rm vd}}\gamma + R_{\rm d}T^*\nu\}\overline{M}^2 D'_{t+\Delta t} \\ +\{-R_{\rm d}T^*\tfrac{c_{\rm pd}}{c_{\rm vd}}(\tfrac{\gamma}{R_{\rm d}} - 1) + \tfrac{R_{\rm d}T^*}{c_{\rm vd}}\gamma\}d_{t+\Delta t}] \\ = \mathcal{D}'^* + \beta\Delta t\nabla'^2[R_{\rm d}T^*(\tfrac{\gamma}{R_{\rm d}} - 1)\hat{\mathcal{Q}}^* - \gamma\mathcal{T}^* - R_{\rm d}T^*\mathcal{P}^*] \end{aligned} \tag{89}$$

  $\mathcal{D}'^{**}$ is defined by equation (90):

$$\mathcal{D}'^{**} = \mathcal{D}'^* + \beta\Delta t\nabla'^2\left[R_{\rm d}T^*(\frac{\gamma}{R_{\rm d}} - 1)\hat{\mathcal{Q}}^* - \gamma\mathcal{T}^* - R_{\rm d}T^*\mathcal{P}^*\right] \tag{90}$$

  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 (31)): this equation can be rewritten:

$$[-(\beta\Delta t)^2\nabla'^2(C^2 - T^*\gamma)]d_{t+\Delta t} + [\mathtt{I} - (\beta\Delta t)^2\nabla'^2(C^2(1 + COR))\overline{M}^2]D'_{t+\Delta t} = \mathcal{D}'^{**} \tag{91}$$

  Quantity COR is zero when the constraint "C1" is fulfilled. In the other cases, COR is generally non-zero but weak ($COR << 1$).

- Elimination of $\hat{Q}$ between equations (86) and (87) leads to equation (92):

$$d_{t+\Delta t} - (\beta\Delta t)^2\left[\frac{\mathbf{L}^{**}}{H^2}(-c_{\rm pd}\tau + C^2)\overline{M}^2 D'_{t+\Delta t} + \frac{C^2}{H^2}\mathbf{L}^{**}d_{t+\Delta t}\right] = \hat{\mathcal{D}}^{**} \tag{92}$$

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

$$\hat{\mathcal{D}}^{**} = \hat{\mathcal{D}}^* + \beta\Delta t\left[-\frac{g}{H}\mathbf{L}^{**}\hat{\mathcal{Q}}^*\right] \tag{93}$$

- Introducing the following denotations:
  - $\mathtt{B}_1 = -C^2$
  - $\mathtt{BC}_1 = -C^2 * COR$
  - $\mathtt{B}_2 = -C^2 + T^*\gamma$
  - $\mathtt{B}_3 = -\frac{1}{H^2}\mathbf{L}^{**}(-c_{\rm pd}\tau + C^2)$
  - $\mathtt{B}_4 = -\frac{C^2}{H^2}\mathbf{L}^{**}$

  equations (91) and (92) can be rewritten:

$$[\mathtt{I} + (\beta\Delta t)^2\nabla'^2(\mathtt{B}_1 + \mathtt{BC}_1)\overline{M}^2]D'_{t+\Delta t} + [(\beta\Delta t)^2\nabla'^2\mathtt{B}_2]d_{t+\Delta t} = \mathcal{D}'^{**} \tag{94}$$

$$[(\beta\Delta t)^2\mathtt{B}_3\overline{M}^2]D'_{t+\Delta t} + [\mathtt{I} + (\beta\Delta t)^2\mathtt{B}_4]d_{t+\Delta t} = \hat{\mathcal{D}}^{**} \tag{95}$$

∗ **Elimination of horizontal divergence $D^{'}$:** When $COR = 0$, elimination of $D^{'}$ between equations (91) and (92) leads to Helmholtz equation (98):

$$\left[\mathtt{I} - \beta^2\Delta t^2 C^2(\overline{M}^2\nabla^{'2} + \tfrac{\mathbf{L}^{**}}{H^2}) - \beta^4\Delta t^4 N^2 C^2\overline{M}^2\nabla^{'2}\mathbf{T}^{**}\right] d_{t+\Delta t}$$
$$= (\mathtt{I} - \beta^2\Delta t^2 C^2\overline{M}^2\nabla^{'2})\hat{\mathcal{D}}^{**} + \beta^2\Delta t^2\tfrac{\mathbf{L}^{**}}{H^2}(-c_{\mathrm{p_d}}\tau + C^2)\overline{M}^2\mathcal{D}^{'**} \quad (96)$$

which can be rewritten:

$$\left[\mathtt{I} - \beta^2\Delta t^2 C^2\left(\mathtt{I} - \beta^2\Delta t^2 C^2\tfrac{\mathbf{L}^{**}}{H^2}\right)^{-1}\left(\mathtt{I} + \beta^2\Delta t^2 N^2\mathbf{T}^{**}\right)\overline{M}^2\nabla^{'2}\right] d_{t+\Delta t}$$
$$= \left(\mathtt{I} - \beta^2\Delta t^2 C^2\tfrac{\mathbf{L}^{**}}{H^2}\right)^{-1}\left[(\mathtt{I} - \beta^2\Delta t^2 C^2\overline{M}^2\nabla^{'2})\hat{\mathcal{D}}^{**} + \beta^2\Delta t^2\tfrac{\mathbf{L}^{**}}{H^2}(-c_{\mathrm{p_d}}\tau + C^2)\overline{M}^2\mathcal{D}^{'**}\right] \quad (97)$$

∗ **Helmholtz equation:** We can notice that Helmholtz equation (97) writes:

$$(\mathtt{I} - \beta^2\Delta t^2\ \mathtt{B}\ \overline{M}^2\nabla^{'2})d_{t+\Delta t} = \mathcal{R} \quad (98)$$

where:

$$\mathtt{B}\ = C^2\left(\mathtt{I} - \beta^2\Delta t^2 C^2\frac{\mathbf{L}^{**}}{H^2}\right)^{-1}\left(\mathtt{I} + \beta^2\Delta t^2 N^2\mathbf{T}^{**}\right) \quad (99)$$

and:

$$\mathcal{R} = \left(\mathtt{I} - \beta^2\Delta t^2 C^2\frac{\mathbf{L}^{**}}{H^2}\right)^{-1}\left[(\mathtt{I} - \beta^2\Delta t^2 C^2\overline{M}^2\nabla^{'2})\hat{\mathcal{D}}^{**} + \beta^2\Delta t^2\frac{\mathbf{L}^{**}}{H^2}(-c_{\mathrm{p_d}}\tau + C^2)\overline{M}^2\mathcal{D}^{'**}\right] \quad (100)$$

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

- The unknown is $d_{t+\Delta t}$.
- The order of $\overline{M}^2$ and $\nabla^{'2}$ is inverted.
- `B` now depends on Δt, it must be recomputed each time the timestep is changed.
- `B` is now a tridiagonal matrix (at least for VFD discretisation).

∗ **Spectral computations to solve system of equations (84) to (88).** Algorithm works zonal wave number by zonal wave number m and is performed in the routine **SPNHSI** (**ESPNHSI** in LAM model). For a given zonal wave number m:

- After a preliminary memory transfer the right-hand side member of equation (98) is computed. We can do some remarks:
  - During the elimination of variables between the equations (84) to (88) 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 $(\mathtt{I} - \beta^2\Delta t^2(C^2/H^2)\mathbf{L}^{**})$ which appears in the RHS of (98) and also in `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 is done according to an algorithm described in subsection 5.1.
- Once known $d_{t+\Delta t}$ equation (91), which can be rewritten as follows:

$$D^{'}_{t+\Delta t} = [\mathtt{I} - (\beta\Delta t)^2\nabla^{'2}C^2\overline{M}^2]^{-1}[\mathcal{D}^{'**} + (\beta\Delta t)^2\nabla^{'2}(-T^*\gamma + C^2)d_{t+\Delta t}] \quad (101)$$

  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 `L`, `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 $\overline{M}^2 D^{'}_{t+\Delta t}$, equation (88) yields $T_{t+\Delta t}$, equation (86) yields $\hat{Q}_{t+\Delta t}$, and equation (84) yields $\log\Pi_{\mathrm{s}t+\Delta t}$.
- 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.

5.4 Semi-implicit scheme for 3D fully elastic nonhydrostatic primitive (NHEE) equations model: an alternate way to solve it.

The linear system to be solved is the same one, but elimination between equations (91) and (92) (or (94) and (95)) is now done on d to obtain an Helmholtz equation with D' as unknown.

$*$ **Elimination of vertical divergence and Helmholtz equation:**

- The following combination is used:

$$(\text{EQ } 94) - [(\beta\Delta t)^2\nabla'^2\mathtt{B}_2][\mathtt{I} + (\beta\Delta t)^2\mathtt{B}_4]^{-1}(\text{EQ } 95)$$

- And additionally we use the fact that ∇'^2 commute with $\mathtt{B}_1$, $\mathtt{BC}_1$, $\mathtt{B}_2$, $\mathtt{B}_3$ and $\mathtt{B}_4$.
- One obtains the following Helmholtz equation:

$$(\mathtt{I} - \beta^2\Delta t^2\ \mathtt{B}\ \nabla'^2\overline{M}^2)D'_{t+\Delta t} = \mathcal{R} \tag{102}$$

where:

$$\mathtt{B}\ = -(\mathtt{B}_1 +\ \mathtt{BC}_1\ ) + (\beta\Delta t)^2\mathtt{B}_2[\mathtt{I} + (\beta\Delta t)^2\mathtt{B}_4]^{-1}\mathtt{B}_3 \tag{103}$$

and:

$$\mathcal{R} = \mathcal{D}'^{**} - (\beta\Delta t)^2\nabla'^2\mathtt{B}_2[\mathtt{I} + (\beta\Delta t)^2\mathtt{B}_4]^{-1}\hat{\mathcal{D}}^{**} \tag{104}$$

Properties of $\mathtt{B}$:

- After calculations (not detailed) using expression of COR and the relationship $c_{p_\mathrm{d}} - c_{v\mathrm{d}} = R_\mathrm{d}$, $\mathtt{B}$ can be rewritten:

$$\mathtt{B}\ =\ \mathtt{B}_{\ \mathrm{hyd}} + \mathtt{B}_2[(\beta\Delta t)^2[\mathtt{I} + (\beta\Delta t)^2\mathtt{B}_4]^{-1} - \mathtt{B}_4{}^{-1}]\mathtt{B}_3 \tag{105}$$

where:

$$\mathtt{B}_{\ \mathrm{hyd}} = \gamma\tau + R_\mathrm{d}T^*\nu$$

- An alternate way to write $\mathtt{B}$ is:

$$\mathtt{B}\ =\ \mathtt{B}_{\ \mathrm{hyd}} + \frac{1}{C^2}[C^2 - T^*\gamma][(\beta\Delta t)^2\frac{C^2}{H^2}[\mathtt{I} + (\beta\Delta t)^2\mathtt{B}_4]^{-1}\mathbf{L}^{**} + \mathtt{I}][C^2 - c_{p_\mathrm{d}}\tau] \tag{106}$$

  Term COR and constraint C1 have disappeared.
- For large timesteps, $\mathtt{B}$ converges towards $\mathtt{B}_{\ \mathrm{hyd}}$.
- For very small timesteps, $\mathtt{B}$ converges towards $C^2(1 + COR)$.

Remarks and advantages compared to an Helmholtz equation with vertical divergence as unknown:

- The code is significantly simpler than with the old method.
- The code is closer to hydrostatic model design.
- Options **LSIDG** (variable mesh) and **LIMPF** (implicit Coriolis term) are simpler to implement (it is easier to re-use hydrostatic model pieces of code).
- There is no C1 constraint, quantity COR does not appear any longer.
- There is no iterative algorithm for VFE-NH.
- Operator $\mathbf{T}^{**}$ does not appear any longer.
- $\mathtt{B}$ easily writes as a sum of hydrostatic and anhydrostatic contributions.
- Calculations use quantity $[\mathtt{I}+(\beta\Delta t)^2\mathtt{B}_4]^{-1}$ which is already computed with the old method (array **SIFACI**).

∗ **Spectral computations to solve system of equations (84) to (88).** Algorithm works zonal wave number by zonal wave number m and is performed in the routine **SPNHEESI** (**ESPNHEESI** in LAM model). For a given zonal wave number m:

- After a preliminary memory transfer the right-hand side member of equation (102) is computed. We can do some remarks:
  - The matricial operator $(\mathtt{I} - \beta^2\Delta t^2(C^2/H^2)\mathbf{L}^{**})$ which appears in the RHS of (102) and also in `B` is pre-computed in routine **SUNHEESI** and is stored in the array **SIFAC**. Its inverse is stored in the array **SIFACI**.
- Inversion of Helmholtz equation is done according to an algorithm described in subsection 5.1.
- Once known $D^{'}_{t+\Delta t}$, calculation of $\overline{M}^2 D^{'}_{t+\Delta t}$ is done; when $\overline{M} = M$ this is a matricial product by a pentadiagonal matrix, and matricial products require additional calls to routine **MXPTMA**.
- After calculation of $\overline{M}^2 D^{'}_{t+\Delta t}$, equation (92) (or (95)) yields $d_{t+\Delta t}$, equation (88) yields $T_{t+\Delta t}$, equation (86) yields $\hat{Q}_{t+\Delta t}$, and equation (84) yields $\log \Pi_{\mathrm{s}t+\Delta t}$.
- Semi-implicit scheme ends by a final memory transfer.

5.5 Semi-implicit scheme for 3D quasi elastic nonhydrostatic primitive (NHQE) equations model.

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

- Continuity equation ($X = \log(\Pi_{\mathrm{s}})$):
$$\mathcal{L} = -\nu(\overline{M}^2 D^{'}) \tag{107}$$
- Divergence equation ($X = D^{'}$):
$$\mathcal{L} = -\nabla^{'2}[\gamma\tilde{T} + R_{\mathrm{d}}T^* \log(\Pi_{\mathrm{s}}) + R_{\mathrm{d}}T^*\hat{Q}] \tag{108}$$
- Vorticity equation ($X = \zeta^{'}$):
$$\mathcal{L} = 0 \tag{109}$$
- Temperature equation ($X = \tilde{T}$):
$$\mathcal{L} = -\tau(\overline{M}^2 D^{'}) \tag{110}$$
- Vertical divergence equation ($X = d$):
$$\mathcal{L} = -\frac{g^2}{R_{\mathrm{d}}T^*}(\mathbf{L}^{**}_{\kappa}\hat{Q}) = -\frac{R_{\mathrm{d}}T^*}{H^2}(\mathbf{L}^{**}_{\kappa}\hat{Q}) \tag{111}$$
- Closure equation, only valid for $d = d_4$:
$$\mathbf{S}_{\kappa}(\overline{M}^2 D^{'}) + d = 0 \tag{112}$$
  $\mathbf{S}_{\kappa}$ is the non-linear counterpart of $\mathbf{S}^*_{\kappa}$: defined with α, δ, Π, R_{d}, $c_{p_{\mathrm{d}}}$.

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

$$\log(\Pi_{\mathrm{s}})_{t+\Delta t} + \beta\Delta t\nu(\overline{M}^2 D^{'}_{t+\Delta t}) = \mathcal{P}^* \tag{113}$$

$$D^{'}_{t+\Delta t} + \beta\Delta t\nabla^{'2}[\gamma\tilde{T}_{t+\Delta t} + R_{\mathrm{d}}T^*\log(\Pi_{\mathrm{s}})_{t+\Delta t} + R_{\mathrm{d}}T^*\hat{Q}_{t+\Delta t}] = \mathcal{D}^{'*} \tag{114}$$

$$d_{t+\Delta t} + \beta\Delta t\frac{R_{\mathrm{d}}T^*}{H^2}(\mathbf{L}^{**}_{\kappa}\hat{Q}_{t+\Delta t}) = \hat{\mathcal{D}}^* \tag{115}$$

$$\tilde{T}_{t+\Delta t} + \beta\Delta t\tau(\overline{M}^2 D^{'}_{t+\Delta t}) = \mathcal{T}^* \tag{116}$$

$$\mathbf{S}^*_{\kappa}(\overline{M}^2 D^{'}_{t+\Delta t}) + d_{t+\Delta t} = (\mathbf{S}^*_{\kappa} - \mathbf{S}_{\kappa})(\overline{M}^2 D^{'}_{t+\Delta t}) \tag{117}$$

$\mathcal{P}^*$, $\mathcal{D}^{'*}$, $\hat{\mathcal{D}}^*$, $\mathcal{T}^*$ correspond to $\mathcal{X}^*$ defined in equation (14) and are available in spectral arrays (**YDSP%SP**, **YDSP%DIV**, **YDSP%SVD**, **YDSP%T**) at the beginning of spectral computations.

∗ Elimination of variables and Helmholtz equation:

- Elimination of d between equations (115) and (117) is done by applying (EQ 117) − (EQ 115), and leads to equation (118):

$$\mathbf{S}^*_\kappa(\overline{M}^2 D'_{t+\Delta t}) - \beta\Delta t \frac{R_\mathrm{d}T^*}{H^2}(\mathbf{L}^{**}_\kappa \hat{Q}_{t+\Delta t}) = (\mathbf{S}^*_\kappa - \mathbf{S}_\kappa)(\overline{M}^2 D'_{t+\Delta t}) - \hat{\mathcal{D}}^* \qquad (118)$$

  This equation must be iterated:

  - General iteration (i) writes:

$$\mathbf{S}^*_\kappa(\overline{M}^2 D'_{t+\Delta t,(i)}) - \beta\Delta t \frac{R_\mathrm{d}T^*}{H^2}(\mathbf{L}^{**}_\kappa \hat{Q}_{t+\Delta t,(i)}) = (\mathbf{S}^*_\kappa - \mathbf{S}_\kappa)(\overline{M}^2 D'_{t+\Delta t,(i-1)}) - \hat{\mathcal{D}}^*$$

  - Predictor step writes:

$$\mathbf{S}^*_\kappa(\overline{M}^2 D'_{t+\Delta t,(i=1)}) - \beta\Delta t \frac{R_\mathrm{d}T^*}{H^2}(\mathbf{L}^{**}_\kappa \hat{Q}_{t+\Delta t,(i=1)}) = (\mathbf{S}^*_\kappa - \mathbf{S}_\kappa)(\overline{M}^2 \mathcal{D}'^*) - \hat{\mathcal{D}}^*$$

- Elimination of $\tilde{T}$, d, $\log(\Pi_\mathrm{s})$ between equations (113), (114) and (116), is done by applying (EQ 114) − $[(\beta\Delta t)^2\nabla'^2 R_\mathrm{d}T^*]$(EQ 113) − $[(\beta\Delta t)^2\nabla'^2\gamma]$(EQ 116), and leads to equation (119):

$$[\mathtt{I} - (\beta\Delta t)^2(\gamma\tau + R_\mathrm{d}T^*\nu)\nabla'^2\overline{M}^2]D'_{t+\Delta t} + (\beta\Delta t)R_\mathrm{d}T^*\nabla'^2\hat{Q}_{t+\Delta t} = \mathcal{D}'^* - (\beta\Delta t)\nabla'^2(\gamma\mathcal{T}^* + R_\mathrm{d}T^*\mathcal{P}^*) \quad (119)$$

- Elimination of $\hat{Q}$ between equations (118) and (119), is done by applying $H^2\mathbf{L}^{**-1}_\kappa\nabla'^2$(EQ 118)+(EQ 119), and leads to Helmholtz equation:

$$\left[\mathtt{I} - \beta^2\Delta t^2\ \mathtt{B}\ \nabla'^2\overline{M}^2\right] D'_{t+\Delta t,(i)} = \mathcal{D}'^{**}_{(i)} \qquad (120)$$

  where:

$$\mathtt{B}\ = (\gamma\tau + R_\mathrm{d}T^*\nu) - \frac{H^2}{\beta^2\Delta t^2}\mathbf{L}^{**-1}_\kappa\mathbf{S}^*_\kappa = \mathtt{B}_{\ \mathrm{hyd}} - \frac{H^2}{\beta^2\Delta t^2}\mathbf{L}^{**-1}_\kappa\mathbf{S}^*_\kappa$$

  and:

$$\mathcal{D}'^{**}_{(i)} = \mathcal{D}'^* - (\beta\Delta t)\nabla'^2(\gamma\mathcal{T}^* + R_\mathrm{d}T^*\mathcal{P}^*) - H^2\mathbf{L}^{**-1}_\kappa\nabla'^2\hat{\mathcal{D}}^* + H^2\mathbf{L}^{**-1}_\kappa\nabla'^2(\mathbf{S}^*_\kappa - \mathbf{S}_\kappa)(\overline{M}^2 D'_{t+\Delta t,(i-1)})$$

  We can notice that Helmholtz equation writes:

$$(\mathtt{I} - \beta^2\Delta t^2\ \mathtt{B}\ \overline{M}^2\nabla'^2)d_{t+\Delta t} = \mathcal{R}$$

  with:

$$\mathcal{R} = \mathcal{D}'^{**}_{(i)}$$

∗ Some remarks about Helmholtz equation:

- Helmholtz equation requires iterations on D'. Due to memory distribution aspects, the loop doing iterations is in **(E)SPCM**.
- Recalculation of the other variables must be done only at the last iteration.
- Calculation of the non-linear term $\mathbf{S}_\kappa\overline{M}^2 D'_{t+\Delta t,(i-1)}$ requires grid-point space calculations (routine **(E)SPNHQESI_PRODSKAP**). Due to memory distribution aspects, **(E)SPNHQESI_PRODSKAP** is called from **(E)SPCM** and not from **(E)SPNHQESI**.
- For pure σ vertical coordinate $\mathbf{S}_\kappa = \mathbf{S}^*_\kappa$ and this is not necessary to iterate; this property is lost for hybrid coordinate.
- For predictor, $D'_{t+\Delta t,(i=0)}$ can be replaced by $\mathcal{D}'^*$.
- For large timesteps, $\mathtt{B}$ converges towards the HPE one.

∗ Spectral computations to solve system of equations (113) to (117). Algorithm works zonal wave number by zonal wave number m and is performed in the routine **SPNHQESI** (**ESPNHQESI** for LAM models). For a given zonal wave number m:

- After a preliminary memory transfer the right-hand side member of equation (76) is computed. A subset of calculations must be done at each iteration (i).
- Inversion of Helmholtz equation is done according to an algorithm described in subsection 5.1. It is done for all iterations.

- Once known $D^{'}_{t+\Delta t}$, the other variables are retrieved as follows:
  - Equation (118) allows to retrieve $\mathbf{L}_{\kappa}^{**}\hat{Q}_{t+\Delta t}$, then $\hat{Q}_{t+\Delta t}$ (multiplication by $\mathbf{L}_{\kappa}^{**-1}$).
  - Equation (117) allows to compute $d_{t+\Delta t}$.
  - Equation (113) allows to compute $\log(\Pi_{\rm s})_{t+\Delta t}$.
  - Equation (116) allows to compute $\tilde{T}_{t+\Delta t}$.

  This is done at the last iteration only.
- Semi-implicit scheme ends by a final memory transfer.

5.6 Semi-implicit scheme for 2D shallow-water model.

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

- Continuity equation ($X = \Phi$):
$$\mathcal{L} = -\Phi^{*}\overline{M}^{2}D^{'} \qquad (121)$$
- Divergence equation ($X = D^{'}$):
$$\mathcal{L} = -\nabla^{'2}(\Phi) \qquad (122)$$
- Vorticity equation ($X = \zeta^{'}$):
$$\mathcal{L} = 0 \qquad (123)$$

$*$ System to be solved:

$$\Phi_{t+\Delta t} + \beta\Delta t\overline{M}^{2}\Phi^{*}D^{'}_{t+\Delta t} = \mathcal{H}^{*} \qquad (124)$$
$$D^{'}_{t+\Delta t} + \beta\Delta t\nabla^{'2}(\Phi_{t+\Delta t}) = \mathcal{D}^{'*} \qquad (125)$$

$\mathcal{H}^{*}$, $\mathcal{D}^{'*}$ correspond to $\mathcal{X}^{*}$ defined in equations (14) and are available in spectral arrays (**YDSP%SP**, **YDSP%DIV**) at the beginning of the spectral computations. Equations (124) and (125) yield (126) (Helmholtz equation):

$$(\mathtt{I} - \beta^{2}\Delta t^{2}\ \mathtt{B}\ \nabla^{'2}\overline{M}^{2})D^{'}_{t+\Delta t} = \mathcal{R} \qquad (126)$$

where $\mathtt{B} = \Phi^{*}$ is a scalar (available in **SIVP**(1)), and:

$$\mathcal{R} = \mathcal{D}^{'*} - \beta\Delta t\nabla^{'2}(\mathcal{H}^{*}) \qquad (127)$$

$*$ Spectral computations to solve system of equations (124) and (125). Algorithm works zonal wave number by zonal wave number m and is performed in routine **SPC2**. For a given zonal wave number m:

- After a preliminary memory transfer the right-hand side member of equation (126) is computed.
- Inversion of Helmholtz equation is done according to an algorithm described in subsection 5.1. Note that $\mathtt{B}$ is the scalar variable Φ^{*}, so no diagonalisation is required.
- Once known $D^{'}_{t+\Delta t}$ equation (124) provides $\Phi_{t+\Delta t}$.
- Semi-implicit scheme ends by a final memory transfer.

6 Inclusion of Coriolis term in the semi-implicit scheme.

Although this term is non linear, it may be added to linear terms in the semi-implicit scheme. Equations are written for a leap-frog scheme (Eulerian scheme or SL3TL scheme). For a SL2TL 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 better 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).

This option is limited to global model with unstretched and untilted geometry. In unstretched and untilted global model 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. This simple property is lost in stretched or tilted geometry and in LAM models.

6.1 3D hydrostatic model.

$*$ **Expression of the linear term $\mathcal{L}$:** Equations (70) and (71) become respectively:

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

$$\mathcal{L} = -\nabla^{'2}(\gamma T + \mu\log\Pi_{\rm s}) - 2\nabla^{'}(\mathbf{\Omega}\wedge\mathbf{V}) \tag{128}$$

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

$$\mathcal{L} = -2\mathbf{k}.[\nabla^{'}\wedge(\mathbf{\Omega}\wedge\mathbf{V})] \tag{129}$$

$*$ **System to be solved:** Equations (73) and (75) are unchanged. Equation (74) is replaced by the two following equations, for divergence and vorticity. Restriction to unstretched untilted global model allows to replace $D^{'}$ by D, $\zeta^{'}$ by ζ, $\nabla^{'}$ by ∇.

$$D_{t+\Delta t} + \beta\Delta t\nabla^2(\gamma T_{t+\Delta t} + \mu\log(\Pi_{\rm s})_{t+\Delta t}) + \beta\Delta t(2\nabla(\mathbf{\Omega}\wedge\mathbf{V})) = \mathcal{D}^* \tag{130}$$

$$\zeta_{t+\Delta t} + \beta\Delta t(2\mathbf{k}.[\nabla\wedge(\mathbf{\Omega}\wedge\mathbf{V})]) = \zeta^* \tag{131}$$

$\mathcal{D}^*$, ζ^* correspond to $\mathcal{X}^*$ defined in equation (14) and are available in spectral arrays (**YDSP%DIV**, **YDSP%VOR**) at the beginning of the spectral computations.

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

$$\nabla\mathbf{Y} = \frac{1}{a\cos\theta}\left[\frac{\partial Y_{\rm x}}{\partial\lambda} + \frac{\partial(Y_{\rm y}\cos\theta)}{\partial\theta}\right] \tag{132}$$

$$\mathbf{k}.(\nabla\wedge\mathbf{Y}) = \frac{1}{a\cos\theta}\left[\frac{\partial Y_{\rm y}}{\partial\lambda} - \frac{\partial(Y_{\rm x}\cos\theta)}{\partial\theta}\right] \tag{133}$$

$*$ **Helmholtz equation:** Using relations (132) and (133) in equations (130) and (131) lead to the following equations:

$$D_{t+\Delta t} + \beta\Delta t\nabla^2(\gamma T_{t+\Delta t} + \mu\log(\Pi_{\rm s})_{t+\Delta t}) + \beta\Delta t(-2\Omega\sin\theta)\zeta_{t+\Delta t} + \beta\Delta t\left(\frac{2\Omega\cos\theta}{a}\right)U_{t+\Delta t} = \mathcal{D}^* \tag{134}$$

$$\zeta_{t+\Delta t} + \beta\Delta t(2\Omega\sin\theta)D_{t+\Delta t} + \beta\Delta t\left(\frac{2\Omega\cos\theta}{a}\right)V_{t+\Delta t} = \zeta^* \tag{135}$$

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

- Relationship between divergence and velocity potential χ:

$$D = \nabla^2\chi \tag{136}$$

- Relationship between vorticity and stream function ψ:

$$\zeta = \nabla^2\psi \tag{137}$$

- (136) and (137) can be rewritten, for each complex spectral component:

$$D_{(m,n)} = -\frac{n(n+1)}{a^2}\chi_{(m,n)} \tag{138}$$

$$\zeta_{(m,n)} = -\frac{n(n+1)}{a^2}\psi_{(m,n)} \tag{139}$$

- Relationship between U, ψ and χ:

$$(Ua\cos\theta)_{(m,n)} = im\chi_{(m,n)} + (n-1)e_{(m,n)}\psi_{(m,n-1)} - (n+2)e_{(m,n+1)}\psi_{(m,n+1)} \tag{140}$$

- Relationship between V, ψ and χ:

$$(Va\cos\theta)_{(m,n)} = im\psi_{(m,n)} - (n-1)e_{(m,n)}\chi_{(m,n-1)} + (n+2)e_{(m,n+1)}\chi_{(m,n+1)} \tag{141}$$

where $e_{(0,0)} = 0$ and $e_{(m,n)} = \sqrt{(n^2-m^2)/(4n^2-1)}$.
Equations (138) to (141) allow to eliminate $U_{t+\Delta t}$ and $V_{t+\Delta t}$ in equations (134) and (135).

$$\begin{aligned}\left[1 - i\tfrac{2\Omega\beta\Delta tm}{n(n+1)}\right] D_{(m,n),t+\Delta t} + \beta\Delta t\nabla^2(\gamma T_{(m,n),t+\Delta t} + \mu\log(\Pi_\mathrm{s})_{(m,n),t+\Delta t}) - \beta\Delta t(2\Omega(\sin\theta)\zeta)_{(m,n),t+\Delta t}\\ -\beta\Delta t\tfrac{2\Omega e_{(m,n)}}{n}\zeta_{(m,n-1),t+\Delta t} + \beta\Delta t\tfrac{2\Omega e_{(m,n+1)}}{n+1}\zeta_{(m,n+1),t+\Delta t} = \mathcal{D}^*_{(m,n)}\end{aligned} \tag{142}$$

$$\begin{aligned}\left[1 - i\tfrac{2\Omega\beta\Delta tm}{n(n+1)}\right] \zeta_{(m,n),t+\Delta t} + \beta\Delta t(2\Omega(\sin\theta)D)_{(m,n),t+\Delta t} + \beta\Delta t\tfrac{2\Omega e_{(m,n)}}{n}D_{(m,n-1),t+\Delta t}\\ -\beta\Delta t\tfrac{2\Omega e_{(m,n+1)}}{n+1}D_{(m,n+1),t+\Delta t} = \zeta^*_{(m,n)}\end{aligned} \tag{143}$$

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

$$[(\sin\theta)X]_{(m,n)} = e_{(m,n)}X_{(m,n-1)} + e_{(m,n+1)}X_{(m,n+1)} \tag{144}$$

Equations (142) and (143) become:

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

$$\begin{aligned}\left[1 - i\tfrac{2\Omega\beta\Delta tm}{n(n+1)}\right] \zeta_{(m,n),t+\Delta t} + \beta\Delta t\tfrac{2\Omega e_{(m,n)}(n+1)}{n}D_{(m,n-1),t+\Delta t} + \beta\Delta t\tfrac{2\Omega e_{(m,n+1)}n}{n+1}D_{(m,n+1),t+\Delta t}\\ = \zeta^*_{(m,n)}\end{aligned} \tag{146}$$

ζ is eliminated in equation (145) by using (146) in replacing n by $n-1$ then by $n+1$. Equation (145) becomes:

$$\begin{aligned}\left[\mathtt{I} - i\tfrac{2\Omega\beta\Delta tm}{n(n+1)} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e^2_{(m,n)}\tfrac{(n-1)(n+1)}{n^2} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e^2_{(m,n+1)}\tfrac{(n)(n+2)}{(n+1)^2}\right] D_{(m,n),t+\Delta t}\\ +\tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}e_{(m,n-1)}\tfrac{(n+1)}{(n-1)}D_{(m,n-2),t+\Delta t} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}e_{(m,n+2)}\tfrac{n}{(n+2)}D_{(m,n+2),t+\Delta t}\\ +\beta\Delta t\nabla^2(\gamma T_{(m,n),t+\Delta t} + \mu\log(\Pi_\mathrm{s})_{(m,n),t+\Delta t})\\ = \mathcal{D}^*_{(m,n)} + \tfrac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}\tfrac{(n+1)}{n}\zeta^*_{(m,n-1)} + \tfrac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}\tfrac{n}{(n+1)}\zeta^*_{(m,n+1)}\end{aligned} \tag{147}$$

$T_{t+\Delta t}$ and $\log(\Pi_\mathrm{s})_{t+\Delta t}$ are eliminated by using equations (73) and (75). That leads to Helmholtz equation (148):

$$\begin{aligned}\left[\mathtt{I} - i\tfrac{2\Omega\beta\Delta tm}{n(n+1)} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e^2_{(m,n)}\tfrac{(n-1)(n+1)}{n^2} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e^2_{(m,n+1)}\tfrac{(n)(n+2)}{(n+1)^2} - \beta^2\Delta t^2\ \mathtt{B}\ \nabla^2\right] D_{(m,n),t+\Delta t}\\ +\tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}e_{(m,n-1)}\tfrac{(n+1)}{(n-1)}D_{(m,n-2),t+\Delta t} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}e_{(m,n+2)}\tfrac{n}{(n+2)}D_{(m,n+2),t+\Delta t}\\ = \mathcal{D}^*_{(m,n)} - \beta\Delta t\nabla^2(\gamma\mathcal{T}^*_{(m,n)} + \mu\mathcal{P}^*_{(m,n)})\\ +\tfrac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}\tfrac{(n+1)}{n}\zeta^*_{(m,n-1)} + \tfrac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}\tfrac{n}{(n+1)}\zeta^*_{(m,n+1)}\end{aligned} \tag{148}$$

Equation (148) 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** via **SPCIMPFSOLVE**), for each zonal wave number m. All computations are currently (in cycle 46t1) done in **SPCSI**. For $m = 0$ complex operators become real operators.

$*$ **Determination of other quantities at $t+\Delta t$:** Equation (147) is used to compute $\zeta_{t+\Delta t}$ (for each zonal wave number m, multiplication by a complex tridiagonal matrix). Then equation (73) is used to compute $\log(\Pi_{\mathrm{s}})_{t+\Delta t}$ and equation (75) is used to compute $T_{t+\Delta t}$.

6.2 3D NHEE model: case with d as unknown in Helmholtz equation.

Extending such an algorithm to the 3D NHEE 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 $D^{'}$.

$*$ **Expression of the linear term $\mathcal{L}$:** Equations (79) and (80) become respectively:

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

$$\mathcal{L} = -\nabla^{'2}[\gamma T - T^*(\gamma\hat{Q}) + \mu\log(\Pi_{\mathrm{s}}) + R_{\mathrm{d}}T^*\hat{Q}] - 2\nabla^{'}(\mathbf{\Omega}\wedge\mathbf{V}) \qquad (149)$$

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

$$\mathcal{L} = -2\mathbf{k}.[\nabla^{'}\wedge(\mathbf{\Omega}\wedge\mathbf{V})] \qquad (150)$$

$*$ **Helmholtz equation:** Elimination of T, $\hat{Q}$ and $\log(\Pi_{\mathrm{s}})$ between equations (84), (86), (88), and the modified (85) is done like for the case **LIMPF**=.F. . We have now a 3-equations system with the unknowns d or d_4, $D^{'}$ and $\zeta^{'}$ (equation of $\zeta^{'}$ 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 $\zeta^{'}$ by D and ζ everywhere.

The relationships (136), (137), (136), (139), (138), (140), (141), (144) 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 $\zeta^{'}$.

To simplify, we adopt the following denotations:

- $\mathtt{B}_1 = -C^2$
- $\mathtt{BC}_1 = -C^2 * COR$
- $\mathtt{B}_2 = -C^2 + T^*\gamma$
- $\mathtt{B}_3 = -\frac{1}{H^2}\mathbf{L}^{**}(-c_{\mathrm{p_d}}\tau + C^2)$
- $\mathtt{B}_4 = -\frac{C^2}{H^2}\mathbf{L}^{**}$

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

$$\left[\mathtt{I} - i\tfrac{2\Omega\beta\Delta tm}{n(n+1)} + (\beta\Delta t)^2\nabla^2(\mathtt{B}_1 + \mathtt{BC}_1)\right] D_{(m,n),t+\Delta t} + \left[(\beta\Delta t)^2\nabla^2\mathtt{B}_2\right] d_{(m,n),t+\Delta t}$$
$$-\beta\Delta t\tfrac{2\Omega e_{(m,n)}(n+1)}{n}\zeta_{(m,n-1),t+\Delta t} - \beta\Delta t\tfrac{2\Omega e_{(m,n+1)}n}{n+1}\zeta_{(m,n+1),t+\Delta t} = \mathcal{D}^{**}_{(m,n)} \qquad (151)$$

$$\left[(\beta\Delta t)^2\mathtt{B}_3\right] D_{(m,n),t+\Delta t} + \left[\mathtt{I} + (\beta\Delta t)^2\mathtt{B}_4\right] d_{(m,n),t+\Delta t} = \hat{\mathcal{D}}^{**}_{(m,n)} \qquad (152)$$

$$\left[\mathtt{I} - i\tfrac{2\Omega\beta\Delta tm}{n(n+1)}\right]\zeta_{(m,n),t+\Delta t} + \beta\Delta t\tfrac{2\Omega e_{(m,n)}(n+1)}{n}D_{(m,n-1),t+\Delta t} + \beta\Delta t\tfrac{2\Omega e_{(m,n+1)}n}{n+1}D_{(m,n+1),t+\Delta t}$$
$$= \zeta^*_{(m,n)} \qquad (153)$$

ζ is eliminated in equation (151) by using (153) in replacing n by $n-1$ then by $n+1$ (exactly like we do in the hydrostatic model). Equation (151) becomes:

$$\left[\mathtt{I} - i\tfrac{2\Omega\beta\Delta tm}{n(n+1)} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e^2_{(m,n)}\tfrac{(n-1)(n+1)}{n^2} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e^2_{(m,n+1)}\tfrac{(n)(n+2)}{(n+1)^2} + (\beta\Delta t)^2\nabla^2(\mathtt{B}_1 + \mathtt{BC}_1)\right] D_{(m,n),t+\Delta t}$$
$$+\tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}e_{(m,n-1)}\tfrac{(n+1)}{(n-1)}D_{(m,n-2),t+\Delta t} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}e_{(m,n+2)}\tfrac{n}{(n+2)}D_{(m,n+2),t+\Delta t}$$
$$+\left[(\beta\Delta t)^2\nabla^2\mathtt{B}_2\right] d_{(m,n),t+\Delta t}$$
$$= \mathcal{D}^{**}_{(m,n)} + \tfrac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}\tfrac{(n+1)}{n}\zeta^*_{(m,n-1)} + \tfrac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}\tfrac{n}{(n+1)}\zeta^*_{(m,n+1)} \qquad (154)$$

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:

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

- 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 (152) 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:

$$\begin{aligned}
&\left[\mathtt{I} - (\beta\Delta t)^2 B\nabla^2 - i\tfrac{2\Omega\beta\Delta tm}{n(n+1)} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}} e^2_{(m,n)} \tfrac{(n-1)(n+1)}{n^2} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}} e^2_{(m,n+1)} \tfrac{(n)(n+2)}{(n+1)^2}\right] d_{(m,n),t+\Delta t} \\
&+ \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}} e_{(m,n)} e_{(m,n-1)} \tfrac{(n+1)}{(n-1)} d_{(m,n-2),t+\Delta t} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}} e_{(m,n+1)} e_{(m,n+2)} \tfrac{n}{(n+2)} d_{(m,n+2),t+\Delta t} \\
&= \left[\mathtt{I} + (\beta\Delta t)^2\mathtt{B}_4\right]^{-1} \left(-(\beta\Delta t)^2\mathtt{B}_3\mathcal{D}^{***}_{(m,n)} + \hat{\mathcal{D}}^{***}_{(m,n)}\right)
\end{aligned} \tag{156}$$

where

$$\begin{aligned}
&\hat{\mathcal{D}}^{***}_{(m,n)} = \\
&\left[\mathtt{I} + (\beta\Delta t)^2\mathtt{B}_1\nabla^2 - i\tfrac{2\Omega\beta\Delta tm}{n(n+1)} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}} e^2_{(m,n)} \tfrac{(n-1)(n+1)}{n^2} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}} e^2_{(m,n+1)} \tfrac{(n)(n+2)}{(n+1)^2}\right] \hat{\mathcal{D}}^{**}_{(m,n)} \\
&+ \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}} e_{(m,n)} e_{(m,n-1)} \tfrac{(n+1)}{(n-1)} \hat{\mathcal{D}}^{**}_{(m,n-2)} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}} e_{(m,n+1)} e_{(m,n+2)} \tfrac{n}{(n+2)} \hat{\mathcal{D}}^{**}_{(m,n+2)}
\end{aligned} \tag{157}$$

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 (156) 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 46t1) done in **SPNHSI**, by calling routine **SIMPLICO**. For $m = 0$ complex operators become real operators.

Once computed $d_{(m,n),t+\Delta t}$, equation (154) is used to retrieve $D_{(m,n),t+\Delta t}$. Equation (154) 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 (153) 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.

6.3 3D NHEE model: case with D' as unknown in Helmholtz equation.

Method is more consistent with what is done in the hydrostatic and NHQE models, and significantly simpler than with d as unknown. One starts from equations (152) and (154). The following combination of equations yields Helmholtz equation:

$$(\text{EQ } 154) - [(\beta\Delta t)^2\nabla^2\mathtt{B}_2][\mathtt{I} + (\beta\Delta t)^2\mathtt{B}_4]^{-1}(\text{EQ } 152)$$

Helmholtz equation writes:

$$\begin{aligned}
&\left[\mathtt{I} - i\tfrac{2\Omega\beta\Delta t m}{n(n+1)} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n-1)n}} e^2_{(m,n)} \tfrac{(n-1)(n+1)}{n^2} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n+1)(n+2)}} e^2_{(m,n+1)} \tfrac{(n)(n+2)}{(n+1)^2} - (\beta\Delta t)^2\ \mathtt{B}\ \nabla^2\right] D_{(m,n),t+\Delta t} \\
&+ \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n-1)n}} e_{(m,n)} e_{(m,n-1)} \tfrac{(n+1)}{(n-1)} D_{(m,n-2),t+\Delta t} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n+1)(n+2)}} e_{(m,n+1)} e_{(m,n+2)} \tfrac{n}{(n+2)} D_{(m,n+2),t+\Delta t} \\
&= \mathcal{D}^{**}_{(m,n)} - [(\beta\Delta t)^2\nabla^2\mathtt{B}_2][\mathtt{I} + (\beta\Delta t)^2\mathtt{B}_4]^{-1}\hat{\mathcal{D}}^{**}_{(m,n)} + \tfrac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta t m}{(n-1)n}} e_{(m,n)} \tfrac{(n+1)}{n} \zeta^*_{(m,n-1)} + \tfrac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta t m}{(n+1)(n+2)}} e_{(m,n+1)} \tfrac{n}{(n+1)} \zeta^*_{(m,n+1)}
\end{aligned} \quad (158)$$

The design of equation (158) is very close to equation (148) one and the same method is applied. Equation (158) 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** via **SPCIMPFSOLVE**), for each zonal wave number m. All computations are currently (in cycle 46t1) done in **SPNHQESI**. For $m = 0$ complex operators become real operators.

Once $D_{t+\Delta t}$ computed:

- Equation (153) yields $\zeta_{t+\Delta t}$
- Equation (92) (or (95)) yields $d_{t+\Delta t}$.
- Equation (88) yields $T_{t+\Delta t}$.
- Equation (86) yields $\hat{Q}_{t+\Delta t}$.
- Equation (84) yields $\log \Pi_{s\,t+\Delta t}$.

6.4 3D NHQE model.

Method is similar to the HPE equations, and calculations are not detailed. HPE calculations can be re-used, with the following replacements:

- When $(\gamma T + \mu\log(\Pi_s))$ appear in equations, replace it by $(\gamma T + \mu\log(\Pi_s) + \mu\hat{Q})$.
- In the RHS of equation (148), replace:

$$\mathcal{D}^*_{(m,n)} - \beta\Delta t\nabla^2(\gamma\mathcal{T}^*_{(m,n)} + \mu\mathcal{P}^*_{(m,n)})$$

  by:

$$\mathcal{D}^*_{(m,n)} - \beta\Delta t\nabla^2(\gamma\mathcal{T}^*_{(m,n)} + \mu\mathcal{P}^*_{(m,n)}) - H^2\mathbf{L}^{**-1}_\kappa\nabla'^2\hat{\mathcal{D}}^*_{(m,n)} + H^2\mathbf{L}^{**-1}_\kappa\nabla'^2\mathbf{S}^*_\kappa[D'_{t+\Delta t,(i-1)}]_{(m,n)} - H^2\mathbf{L}^{**-1}_\kappa\nabla'^2[\mathbf{S}_\kappa D'_{t+\Delta t,(i-1)}]_{(m,n)}$$

  $\mathbf{S}_\kappa$ is the non-linear counterpart of $\mathbf{S}^*_\kappa$.

6.5 2D shallow-water model.

Such an algorithm is also coded in the 2D shallow-water model in not stretched and not tilted geometry. Start from the HPE linear system, then:

- When $(\gamma T + \mu\log(\Pi_s))$ appear in equations, replace it by Φ.
- Replace $(\gamma\mathcal{T}^*_{(m,n)} + \mu\mathcal{P}^*_{(m,n)})$ by $\mathcal{H}^*_{(m,n)}$ in equation (148).
- Once computed $D'_{t+\Delta t}$ equation (124) provides $\Phi_{t+\Delta t}$.

7 Spectral multiplications by polynomial expressions of the mapping factor.

$*$ **Expression of mapping factor M in spectral space in spherical geometry.** Let us denote by:

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

Expression of M is:

$$M = a_{\rm c} + b_{\rm c}\xi \tag{159}$$

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

$$[MX]_{(m,n)} = b_{\rm c}e_{(m,n)}X_{(m,n-1)} + a_{\rm c}X_{(m,n)} + b_{\rm c}e_{(m,n+1)}X_{(m,n+1)} \tag{160}$$

It is easy from (160) 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 spherical geometry.**

$$[M^2X]_{(m,n)} = b_{\rm c}^2e_{(m,n)}e_{(m,n-1)}X_{(m,n-2)} + 2a_{\rm c}b_{\rm c}e_{(m,n)}X_{(m,n-1)} + (a_{\rm c}^2 + b_{\rm c}^2(e_{(m,n)}^2 + e_{(m,n+1)}^2))X_{(m,n)}$$
$$+2a_{\rm c}b_{\rm c}e_{(m,n+1)}X_{(m,n+1)} + b_{\rm c}^2e_{(m,n+1)}e_{(m,n+2)}X_{(m,n+2)} \tag{161}$$

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

8.1 Set-up and control routines until STEPO.

```
CNT0 ->
 * IFS_INIT -> SUCT0, SUDYNA
 * SU0YOMA -> SUGEOMETRY -> SUSMAP, SUESMAP, and so on
 * SU0YOMB ->
   - SUDYN ->
     * SUALDYN
     * SUALDYNB (global model) or SUELDYNB (LAM model).
     * SUSI (hydrostatic model) ->
       - GPHPRE
       - SUBMAT -> SITNU and SIGAM
       - EIGSOL, SCORDO
       - MINV_CALLER
       - SUHEG -> SUHER and SUHES (LSIDG=T only)
       - SUEHEG -> SUHER and SUHES (LESIDG=T only)
     * SUNHSI (NHEE model) ->
       - GPHPRE
       - SUNHBMAT -> SISEVE and MINV_CALLER (NHEE model)
       - EIGSOL, SCORDO
       - MINV_CALLER
       - SUNHHEG -> SUHER and SUHES (LSIDG=T only, NHEE model)
       - SUENHHEG -> SUHER and SUHES (LESIDG=T only, NHEE model)
     * SUNHSI_TESTCONV (NHEE model)
     * SUNHEESI (NHEE model) ->
       - GPHPRE
       - SUNHEEBMAT -> SIGAM, SITNU, SISEVE and MINV_CALLER
       - EIGSOL, SCORDO
       - MINV_CALLER
       - SUHEG -> SUHER and SUHES (LSIDG=T only)
       - SUEHEG -> SUHER and SUHES (LESIDG=T only)
     * SUNHQESI (NHQE model) ->
       - GPHPRE
       - SUBMAT -> SITNU, SIGAM, SILKAPI
       - EIGSOL, SCORDO
       - MINV_CALLER
       - SUHEG -> SUHER and SUHES (LSIDG=T only)
 * CNT1 -> CNT2 -> CNT3 -> CNT4 ->
   - SUHEG -> SUHER and SUHES (LSIDG=T only)
   - SUEHEG -> SUHER and SUHES (LESIDG=T only)
   - SUNHSI (NHEE model) -> (see above)
   - SUNHEESI (NHEE model) -> (see above)
   - SUNHQESI (NHQE 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**.

8.2 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 -> VERDISINT
     * SIGAM -> VERDISINT
     * SPCIMPFSOLVE (LIMPF=T only) -> TRSTOM, SIMPLICO, TRMTOS
     * MXMAOP
     * MXTURS, MXTURE, MXPTMA (LSIDG=T or LESIDG=T only)
   - SPNHSI (NHEE model) ->
     * SIDD -> SIGAM and SISEVE
     * SI_CCCOR
     * SISEVE -> VERDISINT
     * SITNU -> VERDISINT
     * SIMPLICO and SI_MXPTCO (LIMPF=T only)
     * MXMAOP
     * MXTURS, MXTURE, MXPTMA (LSIDG=T or LESIDG=T only)
     * SIGAM -> VERDISINT
   - SPNHEESI (NHEE model) ->
     * SITNU -> VERDISINT
```

```
      * SIGAM -> VERDISINT
      * SIDD -> SIGAM and SISEVE
      * SISEVE -> VERDISINT
      * SPCIMPFSOLVE (LIMPF=T only) -> TRSTOM, SIMPLICO, TRMTOS
      * MXMAOP
      * MXTURS, MXTURE, MXPTMA (LSIDG=T or LESIDG=T only)
    - SPNHQESI (NHQE model) ->
      * SITNU -> VERDISINT
      * SIGAM -> VERDISINT
      * SISKAP
      * SPCIMPFSOLVE (LIMPF=T only) -> TRSTOM, SIMPLICO, TRMTOS
      * MXMAOP
      * MXTURS, MXTURE, MXPTMA (LSIDG=T only)
    - SPNHQESI_PRODSKAP (NHQE model) -> not described
    - TRSTOM (transposition routine for distributed memory)
    - SPCIMPFPOST (LIMPF=T only)
    - SPCHOR (horizontal diffusion)
  * SPC2M -> SPC2 ->
    - SIMPLICO (LIMPF=T only)
    - MXTURS, MXTURE, MXPTMA (LSIDG=T or LESIDG=T only)
    - BALADSM
```

In LAM models, **ESPCM**, **ESPCSI**, **ESPNHSI**, **ESPNHEESI**, **ESPNHQESI**, **ESPCHOR** are called instead of **SPCM**, **SPCSI**, **SPNHSI**, **SPNHEESI**, **SPNHQESI**, **SPCHOR**.

8.3 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, MXTURE, MXPTMA (LSIDG=T or LESIDG=T only)
   - 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**.

8.4 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 NHEE SI scheme (Helmholtz equation with vertical divergence as unknown).
- **SUNHSI_TESTCONV**: test convergence necessary condition of the **NITERHELM** algorithm (NHEE SI scheme).
- **SUNHEESI**: set-up for NHEE SI scheme (Helmholtz equation with horizontal divergence as unknown).
- **SUNHQESI**: set-up for NHQE SI scheme.
- **GPHPRE**: some reference vertical-dependent hydrostatic pressure quantities.
- **SCORDO**: reorders eigenvalues and eigenvectors in order to have ascending values of eigenvectors.
- **SUBMAT**, **SUNH(EE)BMAT**, **SUNHQEBMAT**: computes array **SIB** containing operator `B` respectively for 3D hydrostatic model, 3D NHEE model, 3D NHQE model.
- **SUGEOMETRY**: geometry set-up.
- **SUSMAP**: computes array **SCGMAP** containing coefficients for spectral multiplication by M^2.
- **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 dynamical cores having 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 NHEE model when Helmholtz equation has vertical divergence as unknown.
- **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**: semi-implicit scheme spectral computations in the NHEE model (Helmholtz equation with vertical divergence as unknown).
- **SPNHEESI**: semi-implicit scheme spectral computations in the NHEE model (Helmholtz equation with horizontal divergence as unknown).
- **SPNHQESI**: semi-implicit scheme spectral computations in the NHQE model.
- **SPNHQESI_PRODSKAP**: semi-implicit scheme spectral computations in the NHQE model: code requiring grid-point calculations.
- **SPC2**: spectral space computations, including the semi-implicit scheme and all horizontal diffusion schemes in the 2D model.
- **SPCM**: distributed memory interface for **SPCSI**, **SPNHSI** and **SPNHQESI**.
- **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.
- **SITNU**: computes linear applications by operators τ and ν.
- **SIGAM**: computes a linear application by operator γ.
- **VERDISINT**: interface for vertical integrations or vertical derivatives (vertical finite element scheme).
- **VERINT**: does vertical integrations (vertical finite element scheme).
- **VERDER**: does vertical derivations (vertical finite element scheme).
- **SI_CCCOR**: computes COR (NHEE model).
- **SISEVE**: computes a linear application by operator $\mathbf{L}^*$ or $\mathbf{L}^{**}$ in the NHEE model.
- **SINHQE_SEVE**: computes a linear application by operator $\mathbf{L}^*$ or $\mathbf{L}^{**}$ in the NHQE model.
- **SINHQE_VDERI**: computes a linear application by operator ∂^* in the NHQE model.
- **SIDD**: performs the elimination of T, $\log \Pi_s$ and $\hat{Q}$ in the linear system of NHEE equations (in order to compute the RHS of the Helmholtz equation).
- **SISKAP**: computes $\mathbf{S}^*_\kappa$ in the NHQE model.
- **SISKAPI**: computes $\mathbf{S}^{*-1}_\kappa$ in the NHQE model.
- **SILKAP**: computes $\mathbf{L}^{**}_\kappa$ in the NHQE model.
- **SILKAPI**: computes $\mathbf{L}^{**-1}_\kappa$ and product $\mathbf{L}^{**-1}_\kappa \mathbf{S}^*_\kappa$ in the NHQE model.
- **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_CALLER**: inverts a matrix (calls **MINV** with an additional rescaling).
- **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.

9 Other remarks.

9.1 Adjoint and tangent linear codes.

These codes have been updated for semi-implicit computations in cycle 46t1, 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 46t1 (except for non-hydrostatic model). Notice that tangent linear of **SPCSI** is **SPCSI** itself.

9.2 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**, **LANHQESI**) 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:

- one call to **SITNU** and two calls to **SIGAM** for 3D hydrostatic model.
- one call to **SIPTP**, two calls to **SIDD**, one call to **SISEVE** for 3D NHEE model.
- one call to **SITNU**, two calls to **SIGAM**, one call to **SISKAP** for 3D NHQE model.
- 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 $X(t-\Delta t) - 2X(t)$.
- SL3TL advection: linear terms for $X(t)$ and $X(t-\Delta t)$.
- SL2TL advection: linear terms for $X(t)$.
- Coupler: linear terms for coupler instant.

9.3 Place of inversion of semi-implicit scheme in spectral space.

- The semi-implicit calculations are done after mass corrector computations.
- The semi-implicit calculations are done before spectral nudging (LAM models).
- The semi-implicit calculations are done before horizontal diffusion.
- The semi-implicit calculations are done before nudging (climate models).
- If an ICI scheme is activated (**LPC_FULL=T**), spectral semi-implicit calculations should be done at all iterations.

9.4 Some distributed memory features in spectral calculations.

- 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**.
- The total number of processors is **NPROC=NPRTRW∗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** is the same one as in **SPCSI**.

10 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.
  - LNHEE and LNHQE (in **NAMCT0**), LNHDYN.
  - 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, LMIXETTLS (extrapolation).
  - LAPRXPK, NDLNPR, RHYDR0 (vertical discretisation).
  - LSLINLC2, LSLINL.
  - NPDVAR, NVDVAR, ND4SYS, LNHX, LNHXDER (NH models).
  - LGWADV, NGWADVSI, LRDBBC (treatment of vertical divergence equation in NH model).
  - LSI_NHEE.

  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 ($\Pi_s{}^*$), 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 $\mathtt{B}$ in Helmholtz equation).
  - SIVP (eigenvalues a_l of $\mathtt{B}$), SIMI ($\mathtt{Q}$ = eigenvectors of $\mathtt{B}$), SIMO ($\mathtt{Q}^{-1}$).
  - SIFAC: matricial operator for NHEE model only, i.e. $\mathtt{I} - \beta^2\Delta t^2 C^2 (1/H^2)\mathbf{L}^{**}$.
  - SIFACI: inverse of SIFAC.
  - SIHEG:
    - ∗ Hydrostatic model, NHQE model or 2D shallow-water model: for $m > 0$ contains the non-zero diagonals of $\mathtt{LU}$ decomposition of $(\nabla^{'-2} - \beta^2\Delta t^2 a_l M^2)$; for $m = 0$, contains the non-zero diagonals of $\mathtt{L}$ of the $\mathtt{LU}$ decomposition of $(\mathtt{I} - \beta^2\Delta t^2 a_l \nabla^{'2} M^2)$.
    - ∗ NHEE 3D model: for $m > 0$ contains the non-zero diagonals of $\mathtt{LU}$ decomposition of $(\nabla^{'-2} - \beta^2\Delta t^2 a_l M^2)$; for $m = 0$ contains the non-zero diagonals of $\mathtt{L}$ of the $\mathtt{LU}$ decomposition of $(\mathtt{I} - \beta^2\Delta t^2 a_l M^2 \nabla^{'2})$.
  - SIHEG2:

    * Hydrostatic model, NHQE model or 2D shallow-water model: for $m = 0$ contains the non-zero diagonals of `U` of the `LU` decomposition of $(\mathtt{I} - \beta^2 \Delta t^2 a_l \nabla'^2 M^2)$.
    * NHEE 3D model: for $m = 0$ 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 NHEE 3D model (Helmholtz equation with vertical divergence only): for $m > 0$ contains the non-zero diagonals of `LU` decomposition of $(\nabla'^{-2} - \beta^2 \Delta t^2 C M^2)$; for $m = 0$ 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 NHEE 3D model (Helmholtz equation with vertical divergence only): for $m = 0$ contains the non-zero diagonals of `U` of the `LU` decomposition of $(\mathtt{I} - \beta^2 \Delta t^2 \nabla'^2 C M^2)$
  - SI_ILAPKSSI for NHQE 3D model: contains $\mathbf{L}_\kappa^{**-1}$ and product $\mathbf{L}_\kappa^{**-1} \mathbf{S}_\kappa^*$.

  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).

11 References.

11.1 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., 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.
- 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.
- 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.
- 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.

For non-hydrostatic models aspects:

- 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.
- 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.
- 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.*

11.2 Internal notes and documentation.

- (TDECDYN) 2017: IFS technical documentation (CY43R3). Part III: dynamics and numerical procedures. Available at "https://software.ecmwf.int/wiki/display/IFS/Official+IFS+Documentation".
- (TDECTEC) 2017: IFS technical documentation (CY43R3). 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.umr-cnrm.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., 2018: Basics about ARPEGE/IFS, ALADIN and AROME in the cycle 46t1 of ARPEGE/IFS (internal note, available on the intranet server "http://www.umr-cnrm.fr/gmapdoc/").
- (IDDH) Yessad, K., 2018: Horizontal diffusion in the cycle 46t1 of ARPEGE/IFS (internal note, available on the intranet server "http://www.umr-cnrm.fr/gmapdoc/").
- (IDTS) Yessad, K., 2018: Spectral transforms in the cycle 46t1 of ARPEGE/IFS (internal note, available on the intranet server "http://www.umr-cnrm.fr/gmapdoc/").
- (IDSL) Yessad, K., 2018: Semi-Lagrangian computations in the cycle 46t1 of ARPEGE/IFS (internal note, available on the intranet server "http://www.umr-cnrm.fr/gmapdoc/").
- (IDDM) Yessad, K., 2018: Distributed memory features in the cycle 46t1 of ARPEGE/IFS (internal note, available on the intranet server "http://www.umr-cnrm.fr/gmapdoc/").
- (IDEUL) Yessad, K., 2018: Integration of the model equations, and Eulerian dynamics, in the cycle 46t1 of ARPEGE/IFS (internal note, available on the intranet server "http://www.umr-cnrm.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.umr-cnrm.fr/gmapdoc/".