# BASICS ABOUT ARPEGE/IFS, ALADIN, AROME (CY45).

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

# ARPEGE, ALADIN, ALARO, HIRALD, AROME.

## Several models in the same code

One code, but several models shared between different European (and also some non-European countries) :

- ARPEGE : spectral global model for METEO-FRANCE applications.
- IFS : spectral global model for ECMWF applications.
- ALADIN : spectral limited area model (mesh-size often between 3 km and 10 km).
- ALARO : cf. ALADIN but for some ALADIN partners.
- AROME : non-hydrostatic spectral limited area model for METEO-FRANCE applications (mesh-size 1.3 km).
- ARPEGE/CLIMAT and IFS/CLIMAT : climate versions of ARPEGE and IFS.
- Code stored under GIT.
- Around 14000 routines (around 4 millions code lines) spread among sub-projects.

# ARPEGE, ALADIN, ALARO, HIRALD, AROME.

## Several models in the same code

One code, but several models shared between different European (and also some non-European countries) :

- ARPEGE : spectral global model for METEO-FRANCE applications.
- IFS : spectral global model for ECMWF applications.
- ALADIN : spectral limited area model (mesh-size often between 3 km and 10 km).
- ALARO : cf. ALADIN but for some ALADIN partners.
- AROME : non-hydrostatic spectral limited area model for METEO-FRANCE applications (mesh-size 1.3 km).
- ARPEGE/CLIMAT and IFS/CLIMAT : climate versions of ARPEGE and IFS.
- Code stored under GIT.
- Around 14000 routines (around 4 millions code lines) spread among sub-projects.

# ARPEGE, ALADIN, ALARO, HIRALD, AROME.

## Several models in the same code

One code, but several models shared between different European (and also some non-European countries) :

- ARPEGE : spectral global model for METEO-FRANCE applications.
- IFS : spectral global model for ECMWF applications.
- ALADIN : spectral limited area model (mesh-size often between 3 km and 10 km).
- ALARO : cf. ALADIN but for some ALADIN partners.
- AROME : non-hydrostatic spectral limited area model for METEO-FRANCE applications (mesh-size 1.3 km).
- ARPEGE/CLIMAT and IFS/CLIMAT : climate versions of ARPEGE and IFS.
- Code stored under GIT.
- Around 14000 routines (around 4 millions code lines) spread among sub-projects.

# ARPEGE, ALADIN, ALARO, HIRALD, AROME.

## Brief history

- ARPEGE/IFS : project started in 1987.
- First operational implementation of ARPEGE : september 1992.
- First operational implementation of IFS : 1994.
- ALADIN : project started in 1990.
- First operational implementation of ALADIN : mid 1990s (1995-96 at METEO-FRANCE).
- First operational implementation of AROME : end 2008.

# ARPEGE, ALADIN, ALARO, HIRALD, AROME.

## Brief history

- ARPEGE/IFS : project started in 1987.
- First operational implementation of ARPEGE : september 1992.
- First operational implementation of IFS : 1994.
- ALADIN : project started in 1990.
- First operational implementation of ALADIN : mid 1990s (1995-96 at METEO-FRANCE).
- First operational implementation of AROME : end 2008.

# ARPEGE, ALADIN, ALARO, HIRALD, AROME.

## Brief history

- ARPEGE/IFS : project started in 1987.
- First operational implementation of ARPEGE : september 1992.
- First operational implementation of IFS : 1994.
- ALADIN : project started in 1990.
- First operational implementation of ALADIN : mid 1990s (1995-96 at METEO-FRANCE).
- First operational implementation of AROME : end 2008.

# ARPEGE, ALADIN, ALARO, HIRALD, AROME.

## Brief history

- ARPEGE/IFS : project started in 1987.
- First operational implementation of ARPEGE : september 1992.
- First operational implementation of IFS : 1994.
- ALADIN : project started in 1990.
- First operational implementation of ALADIN : mid 1990s (1995-96 at METEO-FRANCE).
- First operational implementation of AROME : end 2008.

# CODE IS STRUCTURED IN PROJECTS.

## Projects used in forecasts :

- **ARPIFS** : ARPEGE or common ARPEGE-ALADIN routines.
- TRANS : spectral transforms for spherical geometry.
- IFSAUX : some application routines (IO on files, DM environment).
- ALGOR : linear algebra, minimizers other than CONGRAD.
- **ALADIN** : specific LAM routines (LAM, not used at ECMWF).
- ETRANS : spectral transforms for plane geometry (LAM models).
- BIPER : bi-periodicisation package (LAM models).
- COUPLING : coupling package (LAM models).
- SURF : ECMWF surface scheme.
- MPA : upper air MESO-NH/AROME physics.
- MSE : surface processes in MESO-NH/AROME (interface for SURFEX).
- SURFEX : surface processes in MESO-NH/AROME.
- Remark : there are mirror routines between ARPIFS and ALADIN. **ETOTO** is the LAM counterpart of routine **TOTO** ; **SUETOTO** is the LAM counterpart of set-up routine **SUTOTO**.

# CODE IS STRUCTURED IN PROJECTS.

## Projects used in forecasts :

- **ARPIFS** : ARPEGE or common ARPEGE-ALADIN routines.
- TRANS : spectral transforms for spherical geometry.
- IFSAUX : some application routines (IO on files, DM environment).
- ALGOR : linear algebra, minimizers other than CONGRAD.
- **ALADIN** : specific LAM routines (LAM, not used at ECMWF).
- ETRANS : spectral transforms for plane geometry (LAM models).
- BIPER : bi-periodicisation package (LAM models).
- COUPLING : coupling package (LAM models).
- SURF : ECMWF surface scheme.
- MPA : upper air MESO-NH/AROME physics.
- MSE : surface processes in MESO-NH/AROME (interface for SURFEX).
- SURFEX : surface processes in MESO-NH/AROME.
- Remark : there are mirror routines between ARPIFS and ALADIN. **ETOTO** is the LAM counterpart of routine **TOTO** ; **SUETOTO** is the LAM counterpart of set-up routine **SUTOTO**.

# CODE IS STRUCTURED IN PROJECTS.

## Projects used in forecasts :

- **ARPIFS** : ARPEGE or common ARPEGE-ALADIN routines.
- TRANS : spectral transforms for spherical geometry.
- IFSAUX : some application routines (IO on files, DM environment).
- ALGOR : linear algebra, minimizers other than CONGRAD.
- **ALADIN** : specific LAM routines (LAM, not used at ECMWF).
- ETRANS : spectral transforms for plane geometry (LAM models).
- BIPER : bi-periodicisation package (LAM models).
- COUPLING : coupling package (LAM models).
- SURF : ECMWF surface scheme.
- MPA : upper air MESO-NH/AROME physics.
- MSE : surface processes in MESO-NH/AROME (interface for SURFEX).
- SURFEX : surface processes in MESO-NH/AROME.
- Remark : there are mirror routines between ARPIFS and ALADIN. **ETOTO** is the LAM counterpart of routine **TOTO** ; **SUETOTO** is the LAM counterpart of set-up routine **SUTOTO**.

# CODE IS STRUCTURED IN PROJECTS.

## Projects used in forecasts :

- **ARPIFS** : ARPEGE or common ARPEGE-ALADIN routines.
- TRANS : spectral transforms for spherical geometry.
- IFSAUX : some application routines (IO on files, DM environment).
- ALGOR : linear algebra, minimizers other than CONGRAD.
- **ALADIN** : specific LAM routines (LAM, not used at ECMWF).
- ETRANS : spectral transforms for plane geometry (LAM models).
- BIPER : bi-periodicisation package (LAM models).
- COUPLING : coupling package (LAM models).
- SURF : ECMWF surface scheme.
- MPA : upper air MESO-NH/AROME physics.
- MSE : surface processes in MESO-NH/AROME (interface for SURFEX).
- SURFEX : surface processes in MESO-NH/AROME.
- Remark : there are mirror routines between ARPIFS and ALADIN. **ETOTO** is the LAM counterpart of routine **TOTO** ; **SUETOTO** is the LAM counterpart of set-up routine **SUTOTO**.

# CODE IS STRUCTURED IN PROJECTS.

## Projects used in forecasts :

- **ARPIFS** : ARPEGE or common ARPEGE-ALADIN routines.
- TRANS : spectral transforms for spherical geometry.
- IFSAUX : some application routines (IO on files, DM environment).
- ALGOR : linear algebra, minimizers other than CONGRAD.
- **ALADIN** : specific LAM routines (LAM, not used at ECMWF).
- ETRANS : spectral transforms for plane geometry (LAM models).
- BIPER : bi-periodicisation package (LAM models).
- COUPLING : coupling package (LAM models).
- SURF : ECMWF surface scheme.
- MPA : upper air MESO-NH/AROME physics.
- MSE : surface processes in MESO-NH/AROME (interface for SURFEX).
- SURFEX : surface processes in MESO-NH/AROME.
- Remark : there are mirror routines between ARPIFS and ALADIN. **ETOTO** is the LAM counterpart of routine **TOTO** ; **SUETOTO** is the LAM counterpart of set-up routine **SUTOTO**.

# CODE IS STRUCTURED IN PROJECTS (CONT'D).

## Projects used in assimilation :

- AEOLUS : package for pre-processing satellite lidar wind data.
- BLACKLIST : package for blacklisting.
- OBSTAT : statistics of observation feedback data (only used at ECMWF).
- ODB : ODB (Observational DataBase software).
- SATRAD : satellite data handling package.
- SCAT : scatterometers handling.

## Miscellaneous utilitaries :

- UTILITIES : utilitary package (not used at ECMWF).
- SCRIPTS : scripts used at ECMWF.

# CODE IS STRUCTURED IN PROJECTS (CONT'D).

## Projects used in assimilation :

- AEOLUS : package for pre-processing satellite lidar wind data.
- BLACKLIST : package for blacklisting.
- OBSTAT : statistics of observation feedback data (only used at ECMWF).
- ODB : ODB (Observational DataBase software).
- SATRAD : satellite data handling package.
- SCAT : scatterometers handling.

## Miscellaneous utilitaries :

- UTILITIES : utilitary package (not used at ECMWF).
- SCRIPTS : scripts used at ECMWF.

# CODE IS STRUCTURED IN PROJECTS (CONT'D).

## Projects used in assimilation :

- AEOLUS : package for pre-processing satellite lidar wind data.
- BLACKLIST : package for blacklisting.
- OBSTAT : statistics of observation feedback data (only used at ECMWF).
- ODB : ODB (Observational DataBase software).
- SATRAD : satellite data handling package.
- SCAT : scatterometers handling.

## Miscellaneous utilitaries :

- UTILITIES : utilitary package (not used at ECMWF).
- SCRIPTS : scripts used at ECMWF.

# AND EACH PROJECT IS SUBDIVIDED IN DIRECTORIES.

## Example for project ARPIFS (not comprehensive).

- adiab : adiabatic dynamics, adiabatic diagnostics, semi-implicit scheme, horizontal diffusion.
- control : control routines, like CNT4 or STEPO.
- module : all the types of modules.
- namelist : all namelists.
- phys_dmn : physics parameterizations used at METEO-FRANCE.
- setup : a subset of setup routines.
- transform : hat routines for spectral transforms.

# AND EACH PROJECT IS SUBDIVIDED IN DIRECTORIES.

## Example for project ARPIFS (not comprehensive).

- adiab : adiabatic dynamics, adiabatic diagnostics, semi-implicit scheme, horizontal diffusion.
- control : control routines, like CNT4 or STEPO.
- module : all the types of modules.
- namelist : all namelists.
- phys_dmn : physics parameterizations used at METEO-FRANCE.
- setup : a subset of setup routines.
- transform : hat routines for spectral transforms.

# Variable NCONF.

## Range of configurations.

- 0- 99 : 3-D integration job.
- 100-199 : variational job.
- 200-299 : 2-D integration job.
- 300-349 : KALMAN filter.
- 350-399 : predictability model (currently not used).
- 400-499 : test of the adjoint.
- 500-599 : test of the tangent linear model.
- 600-699 : eigenvalue/vector solvers.
- 700-799 : optimal interpolation.
- 800-899 : sensitivity experiments.
- 900-999 : miscellaneous other configurations.
- There are actually around 20 existing configurations.

# Variable NCONF.

## Range of configurations.

- 0- 99 : 3-D integration job.
- 100-199 : variational job.
- 200-299 : 2-D integration job.
- 300-349 : KALMAN filter.
- 350-399 : predictability model (currently not used).
- 400-499 : test of the adjoint.
- 500-599 : test of the tangent linear model.
- 600-699 : eigenvalue/vector solvers.
- 700-799 : optimal interpolation.
- 800-899 : sensitivity experiments.
- 900-999 : miscellaneous other configurations.
- There are actually around 20 existing configurations.

# Variable NCONF.

## Range of configurations.

- 0- 99 : 3-D integration job.
- 100-199 : variational job.
- 200-299 : 2-D integration job.
- 300-349 : KALMAN filter.
- 350-399 : predictability model (currently not used).
- 400-499 : test of the adjoint.
- 500-599 : test of the tangent linear model.
- 600-699 : eigenvalue/vector solvers.
- 700-799 : optimal interpolation.
- 800-899 : sensitivity experiments.
- 900-999 : miscellaneous other configurations.
- There are actually around 20 existing configurations.

# Variable NCONF : example.

## Examples of configurations.

- 1 : forecast.
- 131 : 4DVAR-assimilation.
- 401 : test of the adjoint.
- 501 : test of the tangent linear model.
- 601 : make eigenvectors (for example for PEARP).
- 701 : CANARI surface assimilation.
- 903 : some off-line FULL-POS configurations.
- 923 : make climatology files.

# Variable NCONF : example.

## Examples of configurations.

- 1 : forecast.
- 131 : 4DVAR-assimilation.
- 401 : test of the adjoint.
- 501 : test of the tangent linear model.
- 601 : make eigenvectors (for example for PEARP).
- 701 : CANARI surface assimilation.
- 903 : some off-line FULL-POS configurations.
- 923 : make climatology files.

# GEOMETRY ASPECTS : GLOBAL MODELS.

## Global models.

- Spectral model : fields have a spectral representation defined by a couple of wavenumbers $(m, n)$ ($n$ and $m$ are respectively the total and zonal wavenumbers).
- Triangular truncation $N_{\rm s}$. $n$ varies between 0 and $N_{\rm s}$ ; for each $n$, $\mid m \mid$ varies between 0 and $n$.
- Grid-point calculations on reduced Gaussian grid. There are **NDLON** longitudes and **NDGLG** latitudes. **NDLON** is very close (or equal) to 2∗**NDGLG**.
- Variable mesh : stretching/tilting defined by a high resolution pole and a stretching coefficient **RSTRET** (Schmidt, 1977).

# GEOMETRY ASPECTS : GLOBAL MODELS.

## Global models.

- Spectral model : fields have a spectral representation defined by a couple of wavenumbers $(m, n)$ ($n$ and $m$ are respectively the total and zonal wavenumbers).
- Triangular truncation $N_{\rm s}$. $n$ varies between 0 and $N_{\rm s}$ ; for each $n$, $\mid m \mid$ varies between 0 and $n$.
- Grid-point calculations on reduced Gaussian grid. There are **NDLON** longitudes and **NDGLG** latitudes. **NDLON** is very close (or equal) to 2$*$**NDGLG**.
- Variable mesh : stretching/tilting defined by a high resolution pole and a stretching coefficient **RSTRET** (Schmidt, 1977).

# GEOMETRY ASPECTS : LIMITED AREA MODELS.

## LAM models.

- Spectral model : fields have a spectral representation defined by a couple of wavenumbers $(m, n)$ ($n$ and $m$ are respectively the meridian and zonal wavenumbers).
- Elliptic truncation, with a zonal truncation equal to $N_{\rm ms}$ and a meridian truncation equal to $N_{\rm s}$. Couple $(m, n)$ matches $0 \leq [(n/N_{\rm s})^2 + (m/N_{\rm ms})^2] \leq 1$.
- Grid-point calculations on a limited area plane projection (Lambert, Mercator). There are **NDLON** longitudes and **NDGLG** latitudes.
- Limited area domain is divided into three zones : C (inner), I (intermediate), E (extension).
- Bi-periodicity is done via extension zone.
- For LBC (= lateral boundary conditions), Davies relaxation in I zone (Davies, 1976).

# GEOMETRY ASPECTS : LIMITED AREA MODELS.

## LAM models.

- Spectral model : fields have a spectral representation defined by a couple of wavenumbers $(m, n)$ ($n$ and $m$ are respectively the meridian and zonal wavenumbers).
- Elliptic truncation, with a zonal truncation equal to $N_{\rm ms}$ and a meridian truncation equal to $N_{\rm s}$. Couple $(m, n)$ matches $0 \leq [(n/N_{\rm s})^2 + (m/N_{\rm ms})^2] \leq 1$.
- Grid-point calculations on a limited area plane projection (Lambert, Mercator). There are **NDLON** longitudes and **NDGLG** latitudes.
- Limited area domain is divided into three zones : C (inner), I (intermediate), E (extension).
- Bi-periodicity is done via extension zone.
- For LBC (= lateral boundary conditions), Davies relaxation in I zone (Davies, 1976).

# FORECASTS AND DYNAMICAL CORES.

## Dynamical cores for forecasts.

- Hydrostatic (primitive equation) model (configuration 1).
- Fully elastic non-hydrostatic model (configuration 1 with **LNHDYN**=T).
- Shallow-water model (configuration 201).

## Prognostic and diagnostic variables.

- A prognostic variable is a variable defined by a temporal equation ($\frac{dX}{dt} = RHS$).
- Example of prognostic variables in a hydrostatic model : $U$, $V$, $T$, $q$.
- Other computed variables are diagnostic variables.
- Example of diagnostic variables : $\omega/\Pi$, $\Phi$.

# FORECASTS AND DYNAMICAL CORES.

## Dynamical cores for forecasts.

- Hydrostatic (primitive equation) model (configuration 1).
- Fully elastic non-hydrostatic model (configuration 1 with **LNHDYN**=T).
- Shallow-water model (configuration 201).

## Prognostic and diagnostic variables.

- A prognostic variable is a variable defined by a temporal equation ($\frac{dX}{dt} = RHS$).
- Example of prognostic variables in a hydrostatic model : $U$, $V$, $T$, $q$.
- Other computed variables are diagnostic variables.
- Example of diagnostic variables : $\omega/\Pi$, $\Phi$.

# FORECASTS AND DYNAMICAL CORES.

## Dynamical cores for forecasts.

- Hydrostatic (primitive equation) model (configuration 1).
- Fully elastic non-hydrostatic model (configuration 1 with **LNHDYN**=T).
- Shallow-water model (configuration 201).

## Prognostic and diagnostic variables.

- A prognostic variable is a variable defined by a temporal equation ($\frac{dX}{dt} = RHS$).
- Example of prognostic variables in a hydrostatic model : $U$, $V$, $T$, $q$.
- Other computed variables are diagnostic variables.
- Example of diagnostic variables : $\omega/\Pi$, $\Phi$.

# FORECASTS AND DYNAMICAL CORES.

## Dynamical cores for forecasts.

- Hydrostatic (primitive equation) model (configuration 1).
- Fully elastic non-hydrostatic model (configuration 1 with **LNHDYN**=T).
- Shallow-water model (configuration 201).

## Prognostic and diagnostic variables.

- A prognostic variable is a variable defined by a temporal equation ($\frac{dX}{dt} = RHS$).
- Example of prognostic variables in a hydrostatic model : $U$, $V$, $T$, $q$.
- Other computed variables are diagnostic variables.
- Example of diagnostic variables : $\omega/\Pi$, $\Phi$.

# EQUATIONS : EULERIAN AND SEMI-LAGRANGIAN ASPECTS.

- Eulerian formulation :

$$\frac{\partial X}{\partial t} = -\vec{V}\nabla X - \dot{\eta}\frac{\partial X}{\partial \eta} + \mathcal{A} + \mathcal{F}$$

  ($\mathcal{A}$ = non linear (NL) + linear adiabatic terms, $\mathcal{F}$ = physics).
  Stability condition = CFL criterion.
  Always discretised as a leap-frog scheme.

- Semi-Lagrangian formulation :

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

  Stability condition = Lipschitz criterion, less stringent (the trajectories $O$ - $F$ must not cross each other). Physics often impose a slightly more stringent stability condition.
  Can be discretised as a leap-frog (three-time level) SL scheme or as a two-time level SL scheme (cheaper).

# EQUATIONS : EULERIAN AND SEMI-LAGRANGIAN ASPECTS.

- Eulerian formulation :

$$\frac{\partial X}{\partial t} = -\vec{V}\nabla X - \dot{\eta}\frac{\partial X}{\partial \eta} + \mathcal{A} + \mathcal{F}$$

  ($\mathcal{A}$ = non linear (NL) + linear adiabatic terms, $\mathcal{F}$ = physics).
  Stability condition = CFL criterion.
  Always discretised as a leap-frog scheme.

- Semi-Lagrangian formulation :

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

  Stability condition = Lipschitz criterion, less stringent (the trajectories $O$ - $F$ must not cross each other). Physics often impose a slightly more stringent stability condition.
  Can be discretised as a leap-frog (three-time level) SL scheme or as a two-time level SL scheme (cheaper).

# EQUATIONS : EULERIAN AND SEMI-LAGRANGIAN ASPECTS.

- Eulerian formulation :

$$\frac{\partial X}{\partial t} = -\vec{V}\nabla X - \dot{\eta}\frac{\partial X}{\partial \eta} + \mathcal{A} + \mathcal{F}$$

  ($\mathcal{A}$ = non linear (NL) + linear adiabatic terms, $\mathcal{F}$ = physics).
  Stability condition = CFL criterion.
  Always discretised as a leap-frog scheme.

- Semi-Lagrangian formulation :

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

  Stability condition = Lipschitz criterion, less stringent (the trajectories $O$ - $F$ must not cross each other). Physics often impose a slightly more stringent stability condition.
  Can be discretised as a leap-frog (three-time level) SL scheme or as a two-time level SL scheme (cheaper).

# EQUATIONS : PROGNOSTIC VARIABLES.

**$X$ represents the prognostic variables :**

- In a hydrostatic model, $X$ may be $U$, $V$, $T$, $\log \Pi_{\rm s}$, $q$.
- Link with definitions of GMV and GFL (see below).

# EQUATIONS : PROGNOSTIC VARIABLES.

**$X$ represents the prognostic variables :**

- In a hydrostatic model, $X$ may be $U$, $V$, $T$, $\log \Pi_{\mathrm{s}}$, $q$.
- Link with definitions of GMV and GFL (see below).

# EQUATIONS : DYNAMICS.

$\mathcal{A}$ represents all the effects which can be explicitly represented (often called "adiabatic effects"). Examples :

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

# EQUATIONS : DYNAMICS.

$\mathcal{A}$ represents all the effects which can be explicitly represented (often called "adiabatic effects"). Examples :

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

# EQUATIONS : PHYSICS.

$\mathcal{F}$ represents all the sub-scale effects (often called "diabatic effects" or "physics"). Examples :

- Radiation.
- Stratiform precipitations.
- Convection, and convective precipitations (example : PCMT).
- Vertical diffusion.
- Microphysics.
- Orographic gravity wave drag.
- Exchanges with the surface, interaction with the surface vegetation (examples : ISBA, SURFEX).
- Remark : there are several physics packages in the code.

# EQUATIONS : PHYSICS.

$\mathcal{F}$ represents all the sub-scale effects (often called "diabatic effects" or "physics"). Examples :

- Radiation.
- Stratiform precipitations.
- Convection, and convective precipitations (example : PCMT).
- Vertical diffusion.
- Microphysics.
- Orographic gravity wave drag.
- Exchanges with the surface, interaction with the surface vegetation (examples : ISBA, SURFEX).
- Remark : there are several physics packages in the code.

# EQUATIONS : PHYSICS.

$\mathcal{F}$ represents all the sub-scale effects (often called "diabatic effects" or "physics"). Examples :

- Radiation.
- Stratiform precipitations.
- Convection, and convective precipitations (example : PCMT).
- Vertical diffusion.
- Microphysics.
- Orographic gravity wave drag.
- Exchanges with the surface, interaction with the surface vegetation (examples : ISBA, SURFEX).
- Remark : there are several physics packages in the code.

# EQUATIONS : EULERIAN AND SEMI-LAGRANGIAN DISCRETISATIONS.

- Eulerian discretisation :

$$X(t+\Delta t) - \Delta t\mathcal{L}(t+\Delta t) = X(t-\Delta t)$$

$$-2\Delta t\left[\vec{V}\nabla X\right](t) - 2\Delta t\left[\dot{\eta}\frac{\partial X}{\partial \eta}\right](t) + 2\Delta t[\mathcal{A}(t) - \mathcal{L}(t)] + \Delta t\mathcal{L}(t-\Delta t) + 2\Delta t\mathcal{F}(t-\Delta t)$$

  $\mathcal{L}$ : linear terms.
  All terms are evaluated at the same model grid-point $F$.

- LSETTLS-type two-time level semi-Lagrangian discretisation without uncentering :

$$X(t+\Delta t, F) - 0.5\Delta t\mathcal{L}(t+\Delta t, F) = X(t, O) + \{[0.5\Delta t\mathcal{A}(t) - 0.5\Delta t\mathcal{L}(t)]\}_F$$

$$+\{[\Delta t\mathcal{A}(t) - \Delta t\mathcal{L}(t)] - [0.5\Delta t\mathcal{A}(t-\Delta t) - 0.5\Delta t\mathcal{L}(t-\Delta t)] + [0.5\Delta t\mathcal{L}(t) + \Delta t\mathcal{F}(t)]\}_O$$

  Requires the calculation of an origin point $O$ and interpolations at this point.

- Trajectories are great circles on the geographical sphere in global models, and straight lines on the projection plane in LAM models.
  The computation of the origin point $O$ is performed by an iterative method (2 to 5 iter) described by Robert (1981) and adapted to the sphere by M. Rochas.
  In LAM models, $O$ bounded inside C+I except for the analytical calculation of the Coriolis term.

- Interpolations : generally 32 points or trilinear interpolations, but possible choice of quasi-monotonic interpolations, SLHD interpolations, spline cubic interpolations.

# EQUATIONS : EULERIAN AND SEMI-LAGRANGIAN DISCRETISATIONS.

- Eulerian discretisation :

$$X(t+\Delta t) - \Delta t \mathcal{L}(t+\Delta t) = X(t-\Delta t)$$

$$-2\Delta t \left[\vec{V}\nabla X\right](t) - 2\Delta t \left[\dot{\eta}\frac{\partial X}{\partial \eta}\right](t) + 2\Delta t[\mathcal{A}(t) - \mathcal{L}(t)] + \Delta t \mathcal{L}(t-\Delta t) + 2\Delta t \mathcal{F}(t-\Delta t)$$

  $\mathcal{L}$ : linear terms.
  All terms are evaluated at the same model grid-point $F$.

- LSETTLS-type two-time level semi-Lagrangian discretisation without uncentering :

$$X(t+\Delta t, F) - 0.5\Delta t \mathcal{L}(t+\Delta t, F) = X(t, O) + \{[0.5\Delta t \mathcal{A}(t) - 0.5\Delta t \mathcal{L}(t)]\}_F$$

$$+\{[\Delta t \mathcal{A}(t) - \Delta t \mathcal{L}(t)] - [0.5\Delta t \mathcal{A}(t-\Delta t) - 0.5\Delta t \mathcal{L}(t-\Delta t)] + [0.5\Delta t \mathcal{L}(t) + \Delta t \mathcal{F}(t)]\}_O$$

  Requires the calculation of an origin point $O$ and interpolations at this point.

- Trajectories are great circles on the geographical sphere in global models, and straight lines on the projection plane in LAM models.
  The computation of the origin point $O$ is performed by an iterative method (2 to 5 iter) described by Robert (1981) and adapted to the sphere by M. Rochas.
  In LAM models, $O$ bounded inside C+I except for the analytical calculation of the Coriolis term.

- Interpolations : generally 32 points or trilinear interpolations, but possible choice of quasi-monotonic interpolations, SLHD interpolations, spline cubic interpolations.

# EQUATIONS : EULERIAN AND SEMI-LAGRANGIAN DISCRETISATIONS.

- Eulerian discretisation :

$$X(t+\Delta t) - \Delta t \mathcal{L}(t+\Delta t) = X(t-\Delta t)$$

$$-2\Delta t \left[\vec{V}\nabla X\right](t) - 2\Delta t \left[\dot{\eta}\frac{\partial X}{\partial \eta}\right](t) + 2\Delta t[\mathcal{A}(t) - \mathcal{L}(t)] + \Delta t \mathcal{L}(t-\Delta t) + 2\Delta t \mathcal{F}(t-\Delta t)$$

  $\mathcal{L}$ : linear terms.
  All terms are evaluated at the same model grid-point $F$.

- LSETTLS-type two-time level semi-Lagrangian discretisation without uncentering :

$$X(t+\Delta t, F) - 0.5\Delta t \mathcal{L}(t+\Delta t, F) = X(t, O) + \{[0.5\Delta t \mathcal{A}(t) - 0.5\Delta t \mathcal{L}(t)]\}_F$$

$$+\{[\Delta t \mathcal{A}(t) - \Delta t \mathcal{L}(t)] - [0.5\Delta t \mathcal{A}(t-\Delta t) - 0.5\Delta t \mathcal{L}(t-\Delta t)] + [0.5\Delta t \mathcal{L}(t) + \Delta t \mathcal{F}(t)]\}_O$$

  Requires the calculation of an origin point $O$ and interpolations at this point.

- Trajectories are great circles on the geographical sphere in global models, and straight lines on the projection plane in LAM models.
  The computation of the origin point $O$ is performed by an iterative method (2 to 5 iter) described by Robert (1981) and adapted to the sphere by M. Rochas.
  In LAM models, $O$ bounded inside C+I except for the analytical calculation of the Coriolis term.

- Interpolations : generally 32 points or trilinear interpolations, but possible choice of quasi-monotonic interpolations, SLHD interpolations, spline cubic interpolations.

# EQUATIONS : EULERIAN AND SEMI-LAGRANGIAN DISCRETISATIONS.

- Eulerian discretisation :

$$X(t+\Delta t) - \Delta t\mathcal{L}(t+\Delta t) = X(t-\Delta t)$$

$$-2\Delta t\left[\vec{V}\nabla X\right](t) - 2\Delta t\left[\dot{\eta}\frac{\partial X}{\partial \eta}\right](t) + 2\Delta t[\mathcal{A}(t) - \mathcal{L}(t)] + \Delta t\mathcal{L}(t-\Delta t) + 2\Delta t\mathcal{F}(t-\Delta t)$$

  $\mathcal{L}$ : linear terms.
  All terms are evaluated at the same model grid-point $F$.

- LSETTLS-type two-time level semi-Lagrangian discretisation without uncentering :

$$X(t+\Delta t, F) - 0.5\Delta t\mathcal{L}(t+\Delta t, F) = X(t, O) + \{[0.5\Delta t\mathcal{A}(t) - 0.5\Delta t\mathcal{L}(t)]\}_F$$

$$+\{[\Delta t\mathcal{A}(t) - \Delta t\mathcal{L}(t)] - [0.5\Delta t\mathcal{A}(t-\Delta t) - 0.5\Delta t\mathcal{L}(t-\Delta t)] + [0.5\Delta t\mathcal{L}(t) + \Delta t\mathcal{F}(t)]\}_O$$

  Requires the calculation of an origin point $O$ and interpolations at this point.

- Trajectories are great circles on the geographical sphere in global models, and straight lines on the projection plane in LAM models.
  The computation of the origin point $O$ is performed by an iterative method (2 to 5 iter) described by Robert (1981) and adapted to the sphere by M. Rochas.
  In LAM models, $O$ bounded inside C+I except for the analytical calculation of the Coriolis term.

- Interpolations : generally 32 points or trilinear interpolations, but possible choice of quasi-monotonic interpolations, SLHD interpolations, spline cubic interpolations.

# EQUATIONS : EULERIAN AND SEMI-LAGRANGIAN DISCRETISATIONS.

- Eulerian discretisation :

$$X(t+\Delta t) - \Delta t\mathcal{L}(t+\Delta t) = X(t-\Delta t)$$

$$-2\Delta t\left[\vec{V}\nabla X\right](t) - 2\Delta t\left[\dot{\eta}\frac{\partial X}{\partial \eta}\right](t) + 2\Delta t[\mathcal{A}(t) - \mathcal{L}(t)] + \Delta t\mathcal{L}(t-\Delta t) + 2\Delta t\mathcal{F}(t-\Delta t)$$

  $\mathcal{L}$ : linear terms.
  All terms are evaluated at the same model grid-point $F$.

- LSETTLS-type two-time level semi-Lagrangian discretisation without uncentering :

$$X(t+\Delta t, F) - 0.5\Delta t\mathcal{L}(t+\Delta t, F) = X(t, O) + \{[0.5\Delta t\mathcal{A}(t) - 0.5\Delta t\mathcal{L}(t)]\}_F$$

$$+\{[\Delta t\mathcal{A}(t) - \Delta t\mathcal{L}(t)] - [0.5\Delta t\mathcal{A}(t-\Delta t) - 0.5\Delta t\mathcal{L}(t-\Delta t)] + [0.5\Delta t\mathcal{L}(t) + \Delta t\mathcal{F}(t)]\}_O$$

  Requires the calculation of an origin point $O$ and interpolations at this point.

- Trajectories are great circles on the geographical sphere in global models, and straight lines on the projection plane in LAM models.
  The computation of the origin point $O$ is performed by an iterative method (2 to 5 iter) described by Robert (1981) and adapted to the sphere by M. Rochas.
  In LAM models, $O$ bounded inside C+I except for the analytical calculation of the Coriolis term.

- Interpolations : generally 32 points or trilinear interpolations, but possible choice of quasi-monotonic interpolations, SLHD interpolations, spline cubic interpolations.

# EQUATIONS : GRID-POINT AND SPECTRAL CALCULATIONS.

## Calculations in grid-point space :

- Explicit dynamics.
- Advection, if Eulerian advection.
- Physics.
- Lateral coupling for LAM models.

## Calculations in spectral space :

- Inversion of Helmholtz equations in the semi-implicit scheme (treatment of term $\mathcal{L}$).
- Horizontal diffusion.
- Spectral nudging (near the top) for LAM models.

# EQUATIONS : GRID-POINT AND SPECTRAL CALCULATIONS.

## Calculations in grid-point space :

- Explicit dynamics.
- Advection, if Eulerian advection.
- Physics.
- Lateral coupling for LAM models.

## Calculations in spectral space :

- Inversion of Helmholtz equations in the semi-implicit scheme (treatment of term $\mathcal{L}$).
- Horizontal diffusion.
- Spectral nudging (near the top) for LAM models.

# EQUATIONS : GRID-POINT AND SPECTRAL CALCULATIONS.

## Calculations in grid-point space :

- Explicit dynamics.
- Advection, if Eulerian advection.
- Physics.
- Lateral coupling for LAM models.

## Calculations in spectral space :

- Inversion of Helmholtz equations in the semi-implicit scheme (treatment of term $\mathcal{L}$).
- Horizontal diffusion.
- Spectral nudging (near the top) for LAM models.

# THE DIFFERENT OOPS-ORIENTED OBJECTS.

## List of objects :

- Around 10000 variables; need to gather them in objects.
- Variables are shared into some main objects, for example :
  - INIT : variables like **NCONF**, **LNHDYN**.
  - GEOMETRY : variables describing horizontal and vertical geometry (examples : number of latitudes, longitudes, levels).
  - FIELDS : fields, like GMV, GFL (see below).
  - MODEL : model variables (for example horizontal diffusion coefficients).
  - MTRAJ : trajectory variables.
- Each of these main objects has subdivisions.
- In a model execution under OOPS, several model versions (or "instanciations") may be launched, for example with different horizontal resolutions.
  - "INIT" object variables are identical for all instanciations.
  - GEOMETRY, FIELDS, MODEL, TRAJ objects variables may be different for each instanciation.
  - Variables YRGEOMETRY, YRMODEL, YRFIELDS, YRMTRAJ (declared in **CNT0**) respectively contain GEOMETRY, FIELDS, MODEL, TRAJ objects variables.

# THE DIFFERENT OOPS-ORIENTED OBJECTS.

## List of objects :

- Around 10000 variables ; need to gather them in objects.
- Variables are shared into some main objects, for example :
  - INIT : variables like **NCONF**, **LNHDYN**.
  - GEOMETRY : variables describing horizontal and vertical geometry (examples : number of latitudes, longitudes, levels).
  - FIELDS : fields, like GMV, GFL (see below).
  - MODEL : model variables (for example horizontal diffusion coefficients).
  - MTRAJ : trajectory variables.
- Each of these main objects has subdivisions.
- In a model execution under OOPS, several model versions (or "instanciations") may be launched, for example with different horizontal resolutions.
  - "INIT" object variables are identical for all instanciations.
  - GEOMETRY, FIELDS, MODEL, TRAJ objects variables may be different for each instanciation.
  - Variables YRGEOMETRY, YRMODEL, YRFIELDS, YRMTRAJ (declared in **CNT0**) respectively contain GEOMETRY, FIELDS, MODEL, TRAJ objects variables.

# THE DIFFERENT OOPS-ORIENTED OBJECTS.

## List of objects :

- Around 10000 variables ; need to gather them in objects.
- Variables are shared into some main objects, for example :
  - INIT : variables like **NCONF**, **LNHDYN**.
  - GEOMETRY : variables describing horizontal and vertical geometry (examples : number of latitudes, longitudes, levels).
  - FIELDS : fields, like GMV, GFL (see below).
  - MODEL : model variables (for example horizontal diffusion coefficients).
  - MTRAJ : trajectory variables.
- Each of these main objects has subdivisions.
- In a model execution under OOPS, several model versions (or “instanciations”) may be launched, for example with different horizontal resolutions.
  - “INIT” object variables are identical for all instanciations.
  - GEOMETRY, FIELDS, MODEL, TRAJ objects variables may be different for each instanciation.
  - Variables YRGEOMETRY, YRMODEL, YRFIELDS, YRMTRAJ (declared in **CNT0**) respectively contain GEOMETRY, FIELDS, MODEL, TRAJ objects variables.

# THE DIFFERENT OOPS-ORIENTED OBJECTS.

## List of objects :

- Around 10000 variables ; need to gather them in objects.
- Variables are shared into some main objects, for example :
  - INIT : variables like **NCONF**, **LNHDYN**.
  - GEOMETRY : variables describing horizontal and vertical geometry (examples : number of latitudes, longitudes, levels).
  - FIELDS : fields, like GMV, GFL (see below).
  - MODEL : model variables (for example horizontal diffusion coefficients).
  - MTRAJ : trajectory variables.
- Each of these main objects has subdivisions.
- In a model execution under OOPS, several model versions (or “instanciations”) may be launched, for example with different horizontal resolutions.
  - “INIT” object variables are identical for all instanciations.
  - GEOMETRY, FIELDS, MODEL, TRAJ objects variables may be different for each instanciation.
  - Variables YRGEOMETRY, YRMODEL, YRFIELDS, YRMTRAJ (declared in **CNT0**) respectively contain GEOMETRY, FIELDS, MODEL, TRAJ objects variables.

# THE DIFFERENT OOPS-ORIENTED OBJECTS.

## List of objects :

- Around 10000 variables; need to gather them in objects.
- Variables are shared into some main objects, for example :
  - INIT : variables like **NCONF**, **LNHDYN**.
  - GEOMETRY : variables describing horizontal and vertical geometry (examples : number of latitudes, longitudes, levels).
  - FIELDS : fields, like GMV, GFL (see below).
  - MODEL : model variables (for example horizontal diffusion coefficients).
  - MTRAJ : trajectory variables.
- Each of these main objects has subdivisions.
- In a model execution under OOPS, several model versions (or "instanciations") may be launched, for example with different horizontal resolutions.
  - "INIT" object variables are identical for all instanciations.
  - GEOMETRY, FIELDS, MODEL, TRAJ objects variables may be different for each instanciation.
  - Variables YRGEOMETRY, YRMODEL, YRFIELDS, YRMTRAJ (declared in **CNT0**) respectively contain GEOMETRY, FIELDS, MODEL, TRAJ objects variables.

# GROUPS OF PROGNOSTIC VARIABLES IN "FIELDS" OBJECT.

## Division into GMV, GMVS, GFL, surface fields :

- Upper-air quantities :
  - **For a given dynamical core, GMV+GMVS defines the dynamical core.**
  - For GMV (3D) variables, $\mathcal{A}$ and $\mathcal{L}$ are non-zero. Example : wind components (VOR/DIV in spectral calculations), temperature, additional NH variables. The GMV variables other than the wind components or divergence/vorticity are the "thermodynamical variables" (there are NFTHER thermodynamical variables in the model).
  - GMVS (2D) variables ($\mathcal{A}$ and $\mathcal{L}$ are non-zero). Example : logarithm of surface pressure.
  - **For a given dynamical core, GFL variables are additional variables which do not change the definition of the dynamical core.**
  - For GFL (3D) variables, $\mathcal{A}$ and $\mathcal{L}$ are zero. Example : humidity, liquid water, ice, TKE, ozone, etc...
    This list also contains some pseudo-historic variables (ex CPF = convective precipitation flux).
- Surface prognostic quantities : buffers SP_... of the surface dataflow. Examples : temperature and water content of the soil reservoirs.

# GROUPS OF PROGNOSTIC VARIABLES IN "FIELDS" OBJECT.

## Division into GMV, GMVS, GFL, surface fields :

- Upper-air quantities :
  - **For a given dynamical core, GMV+GMVS defines the dynamical core.**
  - For GMV (3D) variables, $\mathcal{A}$ and $\mathcal{L}$ are non-zero. Example : wind components (VOR/DIV in spectral calculations), temperature, additional NH variables. The GMV variables other than the wind components or divergence/vorticity are the "thermodynamical variables" (there are NFTHER thermodynamical variables in the model).
  - GMVS (2D) variables ($\mathcal{A}$ and $\mathcal{L}$ are non-zero). Example : logarithm of surface pressure.
  - **For a given dynamical core, GFL variables are additional variables which do not change the definition of the dynamical core.**
  - For GFL (3D) variables, $\mathcal{A}$ and $\mathcal{L}$ are zero. Example : humidity, liquid water, ice, TKE, ozone, etc...
    This list also contains some pseudo-historic variables (ex CPF = convective precipitation flux).
- Surface prognostic quantities : buffers SP_.. of the surface dataflow. Examples : temperature and water content of the soil reservoirs.

# GROUPS OF PROGNOSTIC VARIABLES IN "FIELDS" OBJECT.

## Division into GMV, GMVS, GFL, surface fields :

- Upper-air quantities :
  - **For a given dynamical core, GMV+GMVS defines the dynamical core.**
  - For GMV (3D) variables, $\mathcal{A}$ and $\mathcal{L}$ are non-zero. Example : wind components (VOR/DIV in spectral calculations), temperature, additional NH variables. The GMV variables other than the wind components or divergence/vorticity are the "thermodynamical variables" (there are NFTHER thermodynamical variables in the model).
  - GMVS (2D) variables ($\mathcal{A}$ and $\mathcal{L}$ are non-zero). Example : logarithm of surface pressure.
  - **For a given dynamical core, GFL variables are additional variables which do not change the definition of the dynamical core.**
  - For GFL (3D) variables, $\mathcal{A}$ and $\mathcal{L}$ are zero. Example : humidity, liquid water, ice, TKE, ozone, etc...
    This list also contains some pseudo-historic variables (ex CPF = convective precipitation flux).
- Surface prognostic quantities : buffers SP_... of the surface dataflow. Examples : temperature and water content of the soil reservoirs.

# GROUPS OF PROGNOSTIC VARIABLES IN "FIELDS" OBJECT.

## Division into GMV, GMVS, GFL, surface fields :

- Upper-air quantities :
  - **For a given dynamical core, GMV+GMVS defines the dynamical core.**
  - For GMV (3D) variables, $\mathcal{A}$ and $\mathcal{L}$ are non-zero. Example : wind components (VOR/DIV in spectral calculations), temperature, additional NH variables. The GMV variables other than the wind components or divergence/vorticity are the "thermodynamical variables" (there are NFTHER thermodynamical variables in the model).
  - GMVS (2D) variables ($\mathcal{A}$ and $\mathcal{L}$ are non-zero). Example : logarithm of surface pressure.
  - **For a given dynamical core, GFL variables are additional variables which do not change the definition of the dynamical core.**
  - For GFL (3D) variables, $\mathcal{A}$ and $\mathcal{L}$ are zero. Example : humidity, liquid water, ice, TKE, ozone, etc...
    This list also contains some pseudo-historic variables (ex CPF = convective precipitation flux).
- Surface prognostic quantities : buffers SP_.. of the surface dataflow. Examples : temperature and water content of the soil reservoirs.

# GROUPS OF PROGNOSTIC VARIABLES IN “FIELDS” OBJECT.

## Division into GMV, GMVS, GFL, surface fields :

- Upper-air quantities :
  - **For a given dynamical core, GMV+GMVS defines the dynamical core.**
  - For GMV (3D) variables, $\mathcal{A}$ and $\mathcal{L}$ are non-zero. Example : wind components (VOR/DIV in spectral calculations), temperature, additional NH variables. The GMV variables other than the wind components or divergence/vorticity are the ”thermodynamical variables” (there are NFTHER thermodynamical variables in the model).
  - GMVS (2D) variables ($\mathcal{A}$ and $\mathcal{L}$ are non-zero). Example : logarithm of surface pressure.
  - **For a given dynamical core, GFL variables are additional variables which do not change the definition of the dynamical core.**
  - For GFL (3D) variables, $\mathcal{A}$ and $\mathcal{L}$ are zero. Example : humidity, liquid water, ice, TKE, ozone, etc...
    This list also contains some pseudo-historic variables (ex CPF = convective precipitation flux).
- Surface prognostic quantities : buffers SP_.. of the surface dataflow. Examples : temperature and water content of the soil reservoirs.

# GROUPS OF PROGNOSTIC VARIABLES IN "FIELDS" OBJECT.

## Division into GMV, GMVS, GFL, surface fields :

- Upper-air quantities :
  - **For a given dynamical core, GMV+GMVS defines the dynamical core.**
  - For GMV (3D) variables, $\mathcal{A}$ and $\mathcal{L}$ are non-zero. Example : wind components (VOR/DIV in spectral calculations), temperature, additional NH variables. The GMV variables other than the wind components or divergence/vorticity are the "thermodynamical variables" (there are NFTHER thermodynamical variables in the model).
  - GMVS (2D) variables ($\mathcal{A}$ and $\mathcal{L}$ are non-zero). Example : logarithm of surface pressure.
  - **For a given dynamical core, GFL variables are additional variables which do not change the definition of the dynamical core.**
  - For GFL (3D) variables, $\mathcal{A}$ and $\mathcal{L}$ are zero. Example : humidity, liquid water, ice, TKE, ozone, etc...
    This list also contains some pseudo-historic variables (ex CPF = convective precipitation flux).
- Surface prognostic quantities : buffers SP_.. of the surface dataflow. Examples : temperature and water content of the soil reservoirs.

# SPECTRAL VARIABLES.

- YRFIELDS%YRSPEC%[X] : spectral variable for [X]. Example [X]=VOR,DIV,T,Q,SP.
- YRFIELDS%YRSPEC%GFL : all GFL spectral variables.
- YRFIELDS%YRSPEC%SP3D : all 3D variables.
- YRFIELDS%YRSPEC%SP2D : all 2D variables (+ the spectral orography).
- YRFIELDS%YRSPEC%SP1D : mean wind, in LAM models only.

# SPECTRAL VARIABLES.

- YRFIELDS%YRSPEC%[X] : spectral variable for [X]. Example [X]=VOR,DIV,T,Q,SP.
- YRFIELDS%YRSPEC%GFL : all GFL spectral variables.
- YRFIELDS%YRSPEC%SP3D : all 3D variables.
- YRFIELDS%YRSPEC%SP2D : all 2D variables (+ the spectral orography).
- YRFIELDS%YRSPEC%SP1D : mean wind, in LAM models only.

# GRID-POINT VARIABLES.

- YRFIELDS%YRGMV : gathers the $t - \Delta t$ and $t$ GMV variables (including horizontal derivatives).
- YRFIELDS%YRGMVT1 gathers the $t + \Delta t$ GMV variables.
- YRFIELDS%YRGMVS : gathers the $t - \Delta t$ and $t$ GMVS variables (including horizontal derivatives).
- YRFIELDS%YRGMVT1S gathers the $t + \Delta t$ GMVS variables.
- YRFIELDS%YRGFL : gathers the $t - \Delta t$ and $t$ GFL variables (including horizontal derivatives).
- YRFIELDS%YRGFLT1 gathers the $t + \Delta t$ GFL variables.
- YRFIELDS%YRSURF%SP_[group] : prognostic surface dataflow. In particuliar contains 2D surface variables used in the physics.
- YRFIELDS%YRSURF%SD_[group] : diagnostic surface dataflow.
- Individual variables :
  P[X]T0 : $X$ at $t$ ; (P[X]T0L,P[X]T0M) : grad($X$) at $t$.
  P[X]T9 : $X$ at $t$ ; (P[X]T9L,P[X]T9M) : grad($X$) at $t - \Delta t$.
  P[X]T1 : $X$ at $t + \Delta t$.
  Sometimes appendix F for full level, H for half level.
- Additional buffers are needed for some applications.

# GRID-POINT VARIABLES.

- YRFIELDS%YRGMV : gathers the $t - \Delta t$ and $t$ GMV variables (including horizontal derivatives).
- YRFIELDS%YRGMVT1 gathers the $t + \Delta t$ GMV variables.
- YRFIELDS%YRGMVS : gathers the $t - \Delta t$ and $t$ GMVS variables (including horizontal derivatives).
- YRFIELDS%YRGMVT1S gathers the $t + \Delta t$ GMVS variables.
- YRFIELDS%YRGFL : gathers the $t - \Delta t$ and $t$ GFL variables (including horizontal derivatives).
- YRFIELDS%YRGFLT1 gathers the $t + \Delta t$ GFL variables.
- YRFIELDS%YRSURF%SP_[group] : prognostic surface dataflow. In particuliar contains 2D surface variables used in the physics.
- YRFIELDS%YRSURF%SD_[group] : diagnostic surface dataflow.
- Individual variables :
  P[X]T0 : $X$ at $t$ ; (P[X]T0L,P[X]T0M) : grad($X$) at $t$.
  P[X]T9 : $X$ at $t$ ; (P[X]T9L,P[X]T9M) : grad($X$) at $t - \Delta t$.
  P[X]T1 : $X$ at $t + \Delta t$.
  Sometimes appendix F for full level, H for half level.
- Additional buffers are needed for some applications.

# GRID-POINT VARIABLES.

- YRFIELDS%YRGMV : gathers the $t - \Delta t$ and $t$ GMV variables (including horizontal derivatives).
- YRFIELDS%YRGMVT1 gathers the $t + \Delta t$ GMV variables.
- YRFIELDS%YRGMVS : gathers the $t - \Delta t$ and $t$ GMVS variables (including horizontal derivatives).
- YRFIELDS%YRGMVT1S gathers the $t + \Delta t$ GMVS variables.
- YRFIELDS%YRGFL : gathers the $t - \Delta t$ and $t$ GFL variables (including horizontal derivatives).
- YRFIELDS%YRGFLT1 gathers the $t + \Delta t$ GFL variables.
- YRFIELDS%YRSURF%SP_[group] : prognostic surface dataflow. In particuliar contains 2D surface variables used in the physics.
- YRFIELDS%YRSURF%SD_[group] : diagnostic surface dataflow.
- Individual variables :
  P[X]T0 : $X$ at $t$; (P[X]T0L,P[X]T0M) : grad($X$) at $t$.
  P[X]T9 : $X$ at $t$; (P[X]T9L,P[X]T9M) : grad($X$) at $t - \Delta t$.
  P[X]T1 : $X$ at $t + \Delta t$.
  Sometimes appendix F for full level, H for half level.
- Additional buffers are needed for some applications.

# GRID-POINT VARIABLES.

- YRFIELDS%YRGMV : gathers the $t - \Delta t$ and $t$ GMV variables (including horizontal derivatives).
- YRFIELDS%YRGMVT1 gathers the $t + \Delta t$ GMV variables.
- YRFIELDS%YRGMVS : gathers the $t - \Delta t$ and $t$ GMVS variables (including horizontal derivatives).
- YRFIELDS%YRGMVT1S gathers the $t + \Delta t$ GMVS variables.
- YRFIELDS%YRGFL : gathers the $t - \Delta t$ and $t$ GFL variables (including horizontal derivatives).
- YRFIELDS%YRGFLT1 gathers the $t + \Delta t$ GFL variables.
- YRFIELDS%YRSURF%SP_[group] : prognostic surface dataflow. In particuliar contains 2D surface variables used in the physics.
- YRFIELDS%YRSURF%SD_[group] : diagnostic surface dataflow.
- Individual variables :
  P[X]T0 : $X$ at $t$; (P[X]T0L,P[X]T0M) : grad($X$) at $t$.
  P[X]T9 : $X$ at $t$; (P[X]T9L,P[X]T9M) : grad($X$) at $t - \Delta t$.
  P[X]T1 : $X$ at $t + \Delta t$.
  Sometimes appendix F for full level, H for half level.
- Additional buffers are needed for some applications.

# GRID-POINT VARIABLES.

- YRFIELDS%YRGMV : gathers the $t - \Delta t$ and $t$ GMV variables (including horizontal derivatives).
- YRFIELDS%YRGMVT1 gathers the $t + \Delta t$ GMV variables.
- YRFIELDS%YRGMVS : gathers the $t - \Delta t$ and $t$ GMVS variables (including horizontal derivatives).
- YRFIELDS%YRGMVT1S gathers the $t + \Delta t$ GMVS variables.
- YRFIELDS%YRGFL : gathers the $t - \Delta t$ and $t$ GFL variables (including horizontal derivatives).
- YRFIELDS%YRGFLT1 gathers the $t + \Delta t$ GFL variables.
- YRFIELDS%YRSURF%SP_[group] : prognostic surface dataflow. In particuliar contains 2D surface variables used in the physics.
- YRFIELDS%YRSURF%SD_[group] : diagnostic surface dataflow.
- Individual variables :
  P[X]T0 : $X$ at $t$; (P[X]T0L,P[X]T0M) : grad($X$) at $t$.
  P[X]T9 : $X$ at $t$; (P[X]T9L,P[X]T9M) : grad($X$) at $t - \Delta t$.
  P[X]T1 : $X$ at $t + \Delta t$.
  Sometimes appendix F for full level, H for half level.
- Additional buffers are needed for some applications.

# GRID-POINT VARIABLES.

- YRFIELDS%YRGMV : gathers the $t - \Delta t$ and $t$ GMV variables (including horizontal derivatives).
- YRFIELDS%YRGMVT1 gathers the $t + \Delta t$ GMV variables.
- YRFIELDS%YRGMVS : gathers the $t - \Delta t$ and $t$ GMVS variables (including horizontal derivatives).
- YRFIELDS%YRGMVT1S gathers the $t + \Delta t$ GMVS variables.
- YRFIELDS%YRGFL : gathers the $t - \Delta t$ and $t$ GFL variables (including horizontal derivatives).
- YRFIELDS%YRGFLT1 gathers the $t + \Delta t$ GFL variables.
- YRFIELDS%YRSURF%SP_[group] : prognostic surface dataflow. In particuliar contains 2D surface variables used in the physics.
- YRFIELDS%YRSURF%SD_[group] : diagnostic surface dataflow.
- Individual variables :
  P[X]T0 : $X$ at $t$ ; (P[X]T0L,P[X]T0M) : grad($X$) at $t$.
  P[X]T9 : $X$ at $t$ ; (P[X]T9L,P[X]T9M) : grad($X$) at $t - \Delta t$.
  P[X]T1 : $X$ at $t + \Delta t$.
  Sometimes appendix F for full level, H for half level.
- Additional buffers are needed for some applications.

# GRID-POINT VARIABLES.

- YRFIELDS%YRGMV : gathers the $t - \Delta t$ and $t$ GMV variables (including horizontal derivatives).
- YRFIELDS%YRGMVT1 gathers the $t + \Delta t$ GMV variables.
- YRFIELDS%YRGMVS : gathers the $t - \Delta t$ and $t$ GMVS variables (including horizontal derivatives).
- YRFIELDS%YRGMVT1S gathers the $t + \Delta t$ GMVS variables.
- YRFIELDS%YRGFL : gathers the $t - \Delta t$ and $t$ GFL variables (including horizontal derivatives).
- YRFIELDS%YRGFLT1 gathers the $t + \Delta t$ GFL variables.
- YRFIELDS%YRSURF%SP_[group] : prognostic surface dataflow. In particuliar contains 2D surface variables used in the physics.
- YRFIELDS%YRSURF%SD_[group] : diagnostic surface dataflow.
- Individual variables :
  P[X]T0 : $X$ at $t$; (P[X]T0L,P[X]T0M) : grad($X$) at $t$.
  P[X]T9 : $X$ at $t$; (P[X]T9L,P[X]T9M) : grad($X$) at $t - \Delta t$.
  P[X]T1 : $X$ at $t + \Delta t$.
  Sometimes appendix F for full level, H for half level.
- Additional buffers are needed for some applications.

# TANGENT LINEAR AND ADJOINT CODES.

## Why ? :

- Some configurations, like minimisation in a 4D-VAR assimilation, require tangent linear (TL) and adjoint (AD) codes.

## Tangent linear (TL) :

- If the direct code computes the evolution of $X$ ($\frac{dX}{dt} = f(X)$), the tangent linear code computes the evolution of a small perturbation $\delta X$, assuming that the evolution of this perturbation is linear($\frac{d[\delta X]}{dt} = f^{'}(X)[\delta X]$).
- The tangent linear version of a routine **TOTO** has name **TOTOTL**.
- Before running the tangent linear code it is necessary to run the direct code, which provides a trajectory (stored in YRMTRAJ).

# TANGENT LINEAR AND ADJOINT CODES.

## Why ? :

- Some configurations, like minimisation in a 4D-VAR assimilation, require tangent linear (TL) and adjoint (AD) codes.

## Tangent linear (TL) :

- If the direct code computes the evolution of $X$ ($\frac{dX}{dt} = f(X)$), the tangent linear code computes the evolution of a small perturbation $\delta X$, assuming that the evolution of this perturbation is linear($\frac{d[\delta X]}{dt} = f'(X)[\delta X]$).
- The tangent linear version of a routine **TOTO** has name **TOTOTL**.
- Before running the tangent linear code it is necessary to run the direct code, which provides a trajectory (stored in YRMTRAJ).

# TANGENT LINEAR AND ADJOINT CODES.

## Why? :

- Some configurations, like minimisation in a 4D-VAR assimilation, require tangent linear (TL) and adjoint (AD) codes.

## Tangent linear (TL) :

- If the direct code computes the evolution of $X$ ($\frac{dX}{dt} = f(X)$), the tangent linear code computes the evolution of a small perturbation $\delta X$, assuming that the evolution of this perturbation is linear($\frac{d[\delta X]}{dt} = f'(X)[\delta X]$).
- The tangent linear version of a routine **TOTO** has name **TOTOTL**.
- Before running the tangent linear code it is necessary to run the direct code, which provides a trajectory (stored in YRMTRAJ).

# TANGENT LINEAR AND ADJOINT CODES (CONT'D).

## Adjoint (AD) :

- The TL code can be represented by the matricial product : $[\Delta X]_{N_{\rm stop}} = \mathtt{M}[\Delta X]_0$
- Taking the scalar product between $[\Delta X]_{N_{\rm stop}}$ and another vector denoted by $[\Delta Y]$ writes : $\left\langle [\Delta X]_{N_{\rm stop}}, [\Delta Y] \right\rangle = \langle \mathtt{M}[\Delta X]_0, [\Delta Y] \rangle$
- It can be rewritten : $\left\langle [\Delta X]_{N_{\rm stop}}, [\Delta Y] \right\rangle = \left\langle [\Delta X]_0, \mathtt{M}^T [\Delta Y] \right\rangle$
- $\mathtt{M}^T$ is the adjoint operator of $\mathtt{M}$.
- The adjoint version of a routine **TOTO** has name **TOTOAD**.

# TANGENT LINEAR AND ADJOINT CODES (CONT'D).

## Adjoint (AD) :

- The TL code can be represented by the matricial product : $[\Delta X]_{N_{\rm stop}} = \mathtt{M}[\Delta X]_0$
- Taking the scalar product between $[\Delta X]_{N_{\rm stop}}$ and another vector denoted by $[\Delta Y]$ writes : $\left\langle [\Delta X]_{N_{\rm stop}}, [\Delta Y] \right\rangle = \langle \mathtt{M}[\Delta X]_0, [\Delta Y] \rangle$
- It can be rewritten : $\left\langle [\Delta X]_{N_{\rm stop}}, [\Delta Y] \right\rangle = \left\langle [\Delta X]_0, \mathtt{M}^T [\Delta Y] \right\rangle$
- $\mathtt{M}^T$ is the adjoint operator of $\mathtt{M}$.
- The adjoint version of a routine **TOTO** has name **TOTOAD**.

# CODE ARCHITECTURE : SETUP.

## MASTER − > CNT0 − >

- **SU0YOMA** (setup of level 0, part A) − >
  - set-up before **SUGEOMETRY** : object INIT
  - **SUGEOMETRY** : object GEOMETRY
  - set-up after **SUGEOMETRY** : part of object MODEL
- **SU0YOMB** (setup of level 0, part B) : part of object MODEL
- Most namelists are read under **SU0YOMA** and **SU0YOMB**
- **CNT1** for conf 1-99 or 200-299
- CUN3 or CVA1 for conf 100-199
- CSEKF1 for conf 301-349
- CAD1 for conf 401-499
- CTL1 for conf 501-599
- CUN1 for conf 601-699
- CAN1 for conf 701-799
- CGR1 for conf 801-899
- CPREP1 for conf 901
- CPREP3 for conf 903
- INCLI0 for conf 923
- CSSTBLD for conf 931, CSEAICE for conf 932

# CODE ARCHITECTURE : SETUP.

## MASTER − > CNT0 − >

- **SU0YOMA** (setup of level 0, part A) − >
  - set-up before **SUGEOMETRY** : object INIT
  - **SUGEOMETRY** : object GEOMETRY
  - set-up after **SUGEOMETRY** : part of object MODEL
- **SU0YOMB** (setup of level 0, part B) : part of object MODEL
- Most namelists are read under **SU0YOMA** and **SU0YOMB**
- **CNT1** for conf 1-99 or 200-299
- CUN3 or CVA1 for conf 100-199
- CSEKF1 for conf 301-349
- CAD1 for conf 401-499
- CTL1 for conf 501-599
- CUN1 for conf 601-699
- CAN1 for conf 701-799
- CGR1 for conf 801-899
- CPREP1 for conf 901
- CPREP3 for conf 903
- INCLI0 for conf 923
- CSSTBLD for conf 931, CSEAICE for conf 932

# CODE ARCHITECTURE : SETUP.

## MASTER − > CNT0 − >

- **SU0YOMA** (setup of level 0, part A) − >
  - set-up before **SUGEOMETRY** : object INIT
  - **SUGEOMETRY** : object GEOMETRY
  - set-up after **SUGEOMETRY** : part of object MODEL
- **SU0YOMB** (setup of level 0, part B) : part of object MODEL
- Most namelists are read under **SU0YOMA** and **SU0YOMB**
- **CNT1** for conf 1-99 or 200-299
- CUN3 or CVA1 for conf 100-199
- CSEKF1 for conf 301-349
- CAD1 for conf 401-499
- CTL1 for conf 501-599
- CUN1 for conf 601-699
- CAN1 for conf 701-799
- CGR1 for conf 801-899
- CPREP1 for conf 901
- CPREP3 for conf 903
- INCLI0 for conf 923
- CSSTBLD for conf 931, CSEAICE for conf 932

# CODE ARCHITECTURE : SETUP.

## MASTER − > CNT0 − >

- **SU0YOMA** (setup of level 0, part A) − >
  - set-up before **SUGEOMETRY** : object INIT
  - **SUGEOMETRY** : object GEOMETRY
  - set-up after **SUGEOMETRY** : part of object MODEL
- **SU0YOMB** (setup of level 0, part B) : part of object MODEL
- Most namelists are read under **SU0YOMA** and **SU0YOMB**
- **CNT1** for conf 1-99 or 200-299
- CUN3 or CVA1 for conf 100-199
- CSEKF1 for conf 301-349
- CAD1 for conf 401-499
- CTL1 for conf 501-599
- CUN1 for conf 601-699
- CAN1 for conf 701-799
- CGR1 for conf 801-899
- CPREP1 for conf 901
- CPREP3 for conf 903
- INCLI0 for conf 923
- CSSTBLD for conf 931, CSEAICE for conf 932

# CODE ARCHITECTURE : SETUP.

## MASTER $->$ CNT0 $->$

- **SU0YOMA** (setup of level 0, part A) $->$
  - set-up before **SUGEOMETRY** : object INIT
  - **SUGEOMETRY** : object GEOMETRY
  - set-up after **SUGEOMETRY** : part of object MODEL
- **SU0YOMB** (setup of level 0, part B) : part of object MODEL
- Most namelists are read under **SU0YOMA** and **SU0YOMB**
- **CNT1** for conf 1-99 or 200-299
- CUN3 or CVA1 for conf 100-199
- CSEKF1 for conf 301-349
- CAD1 for conf 401-499
- CTL1 for conf 501-599
- CUN1 for conf 601-699
- CAN1 for conf 701-799
- CGR1 for conf 801-899
- CPREP1 for conf 901
- CPREP3 for conf 903
- INCLI0 for conf 923
- CSSTBLD for conf 931, CSEAICE for conf 932

# CODE ARCHITECTURE : CNT1.

## Setup for configuration 1 :

**CNT1** $->$

* SU1YOM (setup of level 1)
* **CNT2** $->$
  - SU2YOM (setup of level 2)
  - **CNT3** $->$
    * CSTA $->$ **SUINIF** (reads the initial files)
    * SU3YOM (setup of level 3)
    * **CNT4** $->$ some setup routines of level 4 and **STEPO**

# ORGANIGRAMME UNDER STEPO.

## STEPO = management of one timestep :

- $X(t)$ available as spectral variable.
- Write historic file [**IOPACK**].
- Inverse transforms + compute horizontal derivatives [**(E)TRANSINVH**]. Provides grid-point $X(t)$ and $\nabla X(t)$.
- Grid-point calculations [**GP_MODEL**] (explicit dynamics, physics, SL interpolations).
- Coupling (LAM models only) [**ECOUPL1**].
- Direct transforms [**(E)TRANSDIRH**] on provisional $X(t + \Delta t)$ variables.
  Remark for spectral transforms : Fourier + Legendre in ARPEGE (code in the TRANS library), double Fourier in LAM models (code in the ETRANS library).
- Spectral calculations [**(E)SPCM**] (SI scheme, horizontal diffusion).
- Provides final $X(t + \Delta t)$, which becomes $X(t)$ at the following timestep.

# ORGANIGRAMME UNDER STEPO.

## STEPO = management of one timestep :

- $X(t)$ available as spectral variable.
- Write historic file [**IOPACK**].
- Inverse transforms + compute horizontal derivatives [**(E)TRANSINVH**]. Provides grid-point $X(t)$ and $\nabla X(t)$.
- Grid-point calculations [**GP_MODEL**] (explicit dynamics, physics, SL interpolations).
- Coupling (LAM models only) [**ECOUPL1**].
- Direct transforms [**(E)TRANSDIRH**] on provisional $X(t + \Delta t)$ variables.
  Remark for spectral transforms : Fourier + Legendre in ARPEGE (code in the TRANS library), double Fourier in LAM models (code in the ETRANS library).
- Spectral calculations [**(E)SPCM**] (SI scheme, horizontal diffusion).
- Provides final $X(t + \Delta t)$, which becomes $X(t)$ at the following timestep.

# ORGANIGRAMME UNDER GP_MODEL FOR SEMI-LAGRANGIAN 3D MODEL.

**STEPO − > SCAN2M − > GP_MODEL − >**

* **CPG_DRV** − > **CPG** (unlagged dynamics, unlagged MF physics)
  - CPG_GP (dynamics calculations)
  - MF_PHYS (MF unlagged physics or AROME physics)
  - CPG_DIA − > (routines for some diagnostics : DDH, CFU, XFU)
  - CPG_DYN − >
  - CPG_END
* RADDRV (ECMWF lagged radiation scheme used at ECMWF)
* **CALL_SL** (semi-Lagrangian only) − >
  - some parallel environment routines spread in the code (SLCOMM.., (E)SLEXTPOL.. routines).
  - LAPINEA − > (E)LARMES : trajectory research, interpolation weights computation.
  - LAPINEB − > LARCINB and LARCINHB (interpolations, updates GFLT1,GMVT1,GMVT1S with the interpolated values).
* EC_PHYS_DRV (ECMWF lagged physics)
* **CPGLAG** (additional dynamics calculations)

# ORGANIGRAMME UNDER GP_MODEL FOR SEMI-LAGRANGIAN 3D MODEL.

**STEPO** – > **SCAN2M** – > **GP_MODEL** – >

* **CPG_DRV** – > **CPG** (unlagged dynamics, unlagged MF physics)
  - **CPG_GP** (dynamics calculations)
  - **MF_PHYS** (MF unlagged physics or AROME physics)
  - **CPG_DIA** – > (routines for some diagnostics : DDH, CFU, XFU)
  - **CPG_DYN** – >
    * CPEULDYN (Eulerian dynamics)
    * **LACDYN** (Semi-Lagrangian dynamics) : calls several LA.. routines, for example to fill PB1 (interpolation buffer), computes some linear terms.
    * VDIFLCZ (Buizza simplified physics)
  - **CPG_END**
* RADDRV (ECMWF lagged radiation scheme used at ECMWF)
* **CALL_SL** (semi-Lagrangian only) – >
  - some parallel environment routines spread in the code (SLCOMM.., (E)SLEXTPOL.. routines).
  - **LAPINEA** – > (E)LARMES : trajectory research, interpolation weights computation.
  - **LAPINEB** – > LARCINB and LARCINHB (interpolations, updates GFLT1,GMVT1,GMVT1S with the interpolated values).
* EC_PHYS_DRV (ECMWF lagged physics)
* **CPGLAG** (additional dynamics calculations)

# ORGANIGRAMME UNDER GP_MODEL FOR SEMI-LAGRANGIAN 3D MODEL.

**STEPO** – > **SCAN2M** – > **GP_MODEL** – >

- **CPG_DRV** – > **CPG** (unlagged dynamics, unlagged MF physics)
  - **CPG_GP** (dynamics calculations)
  - **MF_PHYS** (MF unlagged physics or AROME physics)
  - **CPG_DIA** – > (routines for some diagnostics : DDH, CFU, XFU)
  - **CPG_DYN** – >
    - CPEULDYN (Eulerian dynamics)
    - **LACDYN** (Semi-Lagrangian dynamics) : calls several LA.. routines, for example to fill PB1 (interpolation buffer), computes some linear terms.
    - VDIFLCZ (Buizza simplified physics)
  - **CPG_END**
- RADDRV (ECMWF lagged radiation scheme used at ECMWF)
- **CALL_SL** (semi-Lagrangian only) – >
  - some parallel environment routines spread in the code (SLCOMM.., (E)SLEXTPOL.. routines).
  - **LAPINEA** – > (E)LARMES : trajectory research, interpolation weights computation.
  - **LAPINEB** – > LARCINB and LARCINHB (interpolations, updates GFLT1,GMVT1,GMVT1S with the interpolated values).
- EC_PHYS_DRV (ECMWF lagged physics)
- **CPGLAG** (additional dynamics calculations)

# ORGANIGRAMME UNDER GP_MODEL FOR SEMI-LAGRANGIAN 3D MODEL.

**STEPO** − > **SCAN2M** − > **GP_MODEL** − >

* **CPG_DRV** − > **CPG** (unlagged dynamics, unlagged MF physics)
  - **CPG_GP** (dynamics calculations)
  - **MF_PHYS** (MF unlagged physics or AROME physics)
  - **CPG_DIA** − > (routines for some diagnostics : DDH, CFU, XFU)
  - **CPG_DYN** − >
    * CPEULDYN (Eulerian dynamics)
    * **LACDYN** (Semi-Lagrangian dynamics) : calls several LA.. routines, for example to fill PB1 (interpolation buffer), computes some linear terms.
    * VDIFLCZ (Buizza simplified physics)
  - **CPG_END**
* RADDRV (ECMWF lagged radiation scheme used at ECMWF)
* **CALL_SL** (semi-Lagrangian only) − >
  - some parallel environment routines spread in the code (SLCOMM.., (E)SLEXTPOL.. routines).
  - **LAPINEA** − > (E)LARMES : trajectory research, interpolation weights computation.
  - **LAPINEB** − > LARCINB and LARCINHB (interpolations, updates GFLT1,GMVT1,GMVT1S with the interpolated values).
* EC_PHYS_DRV (ECMWF lagged physics)
* **CPGLAG** (additional dynamics calculations)

# NAMING ROUTINES.

**Some routines names start by a specific prefix ; examples :**

- SU.. : set-up routines.
- CA.. : CANARI surface assimilation.
- LA.. : semi-Lagrangian advection routines.
- CP.. or GP.. : grid-point space calculations.
- GNH.. : non-hydrostatic grid-point space calculations.
- SP.. : spectral space calculations.

# NAMING ROUTINES.

**Some routines names start by a specific prefix ; examples :**

- SU.. : set-up routines.
- CA.. : CANARI surface assimilation.
- LA.. : semi-Lagrangian advection routines.
- CP.. or GP.. : grid-point space calculations.
- GNH.. : non-hydrostatic grid-point space calculations.
- SP.. : spectral space calculations.

# DIAGNOSTICS.

## Inventory :

- Write historic files.
- Post-processing : **FULL-POS**.
- Horizontal domains diagnostics : DDH.
- Cumulated fluxes : CFU.
- Instantaneous fluxes : XFU.
- Spectral norms and grid-point norms printings.
- There are other diagnostics spread in the code (for example in SURFEX, physics).

# DIAGNOSTICS.

## Inventory :

- Write historic files.
- Post-processing : **FULL-POS**.
- Horizontal domains diagnostics : DDH.
- Cumulated fluxes : CFU.
- Instantaneous fluxes : XFU.
- Spectral norms and grid-point norms printings.
- There are other diagnostics spread in the code (for example in SURFEX, physics).

# DIAGNOSTICS.

**Inventory :**

- Write historic files.
- Post-processing : **FULL-POS**.
- Horizontal domains diagnostics : DDH.
- Cumulated fluxes : CFU.
- Instantaneous fluxes : XFU.
- Spectral norms and grid-point norms printings.
- There are other diagnostics spread in the code (for example in SURFEX, physics).

# DIAGNOSTICS.

## Inventory :

- Write historic files.
- Post-processing : **FULL-POS**.
- Horizontal domains diagnostics : DDH.
- Cumulated fluxes : CFU.
- Instantaneous fluxes : XFU.
- Spectral norms and grid-point norms printings.
- There are other diagnostics spread in the code (for example in SURFEX, physics).

# DIAGNOSTICS.

## Inventory :

- Write historic files.
- Post-processing : **FULL-POS**.
- Horizontal domains diagnostics : DDH.
- Cumulated fluxes : CFU.
- Instantaneous fluxes : XFU.
- Spectral norms and grid-point norms printings.
- There are other diagnostics spread in the code (for example in SURFEX, physics).

# DIAGNOSTICS.

## Inventory :

- Write historic files.
- Post-processing : **FULL-POS**.
- Horizontal domains diagnostics : DDH.
- Cumulated fluxes : CFU.
- Instantaneous fluxes : XFU.
- Spectral norms and grid-point norms printings.
- There are other diagnostics spread in the code (for example in SURFEX, physics).

# FULL-POS.

- Post-processing for different types of variables : 3D dynamical variables, 2D dynamical variables, surface fields used in the physics, fields computed by the CFU or the XFU.
- Post-processing on different surfaces : hydrostatic pressure (ex : Z500), geopotential height, hybrid coordinate, potential temperature, potential vorticity, temperature, flight level, surface, sea level (ex : MSLP).
- Post-processing on different domains : whole Earth in spectral, grid-point or "lat-lon" grid representation ; LAM sub-domain in spectral or grid-point representation ; "lat-lon" sub-domain in grid-point representation.
- One application of FULL-POS is to change resolution (examples : to make coupling files, to change horizontal resolution in 4DVAR).

# FULL-POS.

- Post-processing for different types of variables : 3D dynamical variables, 2D dynamical variables, surface fields used in the physics, fields computed by the CFU or the XFU.
- Post-processing on different surfaces : hydrostatic pressure (ex : Z500), geopotential height, hybrid coordinate, potential temperature, potential vorticity, temperature, flight level, surface, sea level (ex : MSLP).
- Post-processing on different domains : whole Earth in spectral, grid-point or "lat-lon" grid representation ; LAM sub-domain in spectral or grid-point representation ; "lat-lon" sub-domain in grid-point representation.
- One application of FULL-POS is to change resolution (examples : to make coupling files, to change horizontal resolution in 4DVAR).

# FULL-POS.

- Post-processing for different types of variables : 3D dynamical variables, 2D dynamical variables, surface fields used in the physics, fields computed by the CFU or the XFU.
- Post-processing on different surfaces : hydrostatic pressure (ex : Z500), geopotential height, hybrid coordinate, potential temperature, potential vorticity, temperature, flight level, surface, sea level (ex : MSLP).
- Post-processing on different domains : whole Earth in spectral, grid-point or "lat-lon" grid representation ; LAM sub-domain in spectral or grid-point representation ; "lat-lon" sub-domain in grid-point representation.
- One application of FULL-POS is to change resolution (examples : to make coupling files, to change horizontal resolution in 4DVAR).

# FULL-POS.

- Post-processing for different types of variables : 3D dynamical variables, 2D dynamical variables, surface fields used in the physics, fields computed by the CFU or the XFU.
- Post-processing on different surfaces : hydrostatic pressure (ex : Z500), geopotential height, hybrid coordinate, potential temperature, potential vorticity, temperature, flight level, surface, sea level (ex : MSLP).
- Post-processing on different domains : whole Earth in spectral, grid-point or "lat-lon" grid representation ; LAM sub-domain in spectral or grid-point representation ; "lat-lon" sub-domain in grid-point representation.
- One application of FULL-POS is to change resolution (examples : to make coupling files, to change horizontal resolution in 4DVAR).

# FULL-POS.

- Post-processing for different types of variables : 3D dynamical variables, 2D dynamical variables, surface fields used in the physics, fields computed by the CFU or the XFU.
- Post-processing on different surfaces : hydrostatic pressure (ex : Z500), geopotential height, hybrid coordinate, potential temperature, potential vorticity, temperature, flight level, surface, sea level (ex : MSLP).
- Post-processing on different domains : whole Earth in spectral, grid-point or "lat-lon" grid representation ; LAM sub-domain in spectral or grid-point representation ; "lat-lon" sub-domain in grid-point representation.
- One application of FULL-POS is to change resolution (examples : to make coupling files, to change horizontal resolution in 4DVAR).

# DISTRIBUTED MEMORY AND OpenMP ASPECTS.

## Two ways of distribution :

- Message passing (MPI) : call to MPI.. routines.
- OpenMp : use of directives.

## MPI distribution :

- Two levels of distribution.
- There are **NPROC** processors.
- Two levels in grid-point calculations : **NPROC=NPRGPNS∗NPRGPEW**.
- Two levels in spectral calculations : **NPROC=NPRTRW∗NPRTRV**.
- There are other variables for IO server, IO.

# DISTRIBUTED MEMORY AND OpenMP ASPECTS.

## Two ways of distribution :

- Message passing (MPI) : call to MPI.. routines.
- OpenMp : use of directives.

## MPI distribution :

- Two levels of distribution.
- There are **NPROC** processors.
- Two levels in grid-point calculations : **NPROC=NPRGPNS*NPRGPEW**.
- Two levels in spectral calculations : **NPROC=NPRTRW*NPRTRV**.
- There are other variables for IO server, IO.

# DISTRIBUTED MEMORY AND OpenMP ASPECTS.

## Two ways of distribution :

- Message passing (MPI) : call to MPI.. routines.
- OpenMp : use of directives.

## MPI distribution :

- Two levels of distribution.
- There are **NPROC** processors.
- Two levels in grid-point calculations : **NPROC=NPRGPNS∗NPRGPEW**.
- Two levels in spectral calculations : **NPROC=NPRTRW∗NPRTRV**.
- There are other variables for IO server, IO.

# DISTRIBUTED MEMORY AND OpenMP ASPECTS.

## Two ways of distribution :

- Message passing (MPI) : call to MPI.. routines.
- OpenMp : use of directives.

## MPI distribution :

- Two levels of distribution.
- There are **NPROC** processors.
- Two levels in grid-point calculations : **NPROC=NPRGPNS*NPRGPEW**.
- Two levels in spectral calculations : **NPROC=NPRTRW*NPRTRV**.
- There are other variables for IO server, IO.

# DISTRIBUTED MEMORY AND DATA ORGANISATION.

## Horizontal representation in spectral space :

- In global models, **NSMAX** is the truncation.
- In LAM models, **NSMAX** and **NMSMAX** are the meridian and zonal truncations.
- A processor treats a subset of zonal wave numbers.

## Horizontal representation in grid-point space :

- For a 2D field, there are **NGPTOTG** grid-points, **NDGLG** latitudes, **NDLON** longitudes.
- A processor treats **NGPTOT** points (**NGPTOT** is processor dependent).
- In grid-point calculations, the **NGPTOT** points are sub-divided into **NGPBLKS** packets of **NPROMA** points.
- **NPROMA** is a tunable variable.

## Vertical representation :

- There are **NFLEVG** levels.
- When the second level of distribution is activated, some part of spectral calculations work on a subset of **NFLEVL** levels..
- Vertical discretisation : finite differences (VFD) or finite elements (VFE).

# DISTRIBUTED MEMORY AND DATA ORGANISATION.

## Horizontal representation in spectral space :

- In global models, **NSMAX** is the truncation.
- In LAM models, **NSMAX** and **NMSMAX** are the meridian and zonal truncations.
- A processor treats a subset of zonal wave numbers.

## Horizontal representation in grid-point space :

- For a 2D field, there are **NGPTOTG** grid-points, **NDGLG** latitudes, **NDLON** longitudes.
- A processor treats **NGPTOT** points (**NGPTOT** is processor dependent).
- In grid-point calculations, the **NGPTOT** points are sub-divided into **NGPBLKS** packets of **NPROMA** points.
- **NPROMA** is a tunable variable.

## Vertical representation :

- There are **NFLEVG** levels.
- When the second level of distribution is activated, some part of spectral calculations work on a subset of **NFLEVL** levels..
- Vertical discretisation : finite differences (VFD) or finite elements (VFE).

# DISTRIBUTED MEMORY AND DATA ORGANISATION.

## Horizontal representation in spectral space :

- In global models, **NSMAX** is the truncation.
- In LAM models, **NSMAX** and **NMSMAX** are the meridian and zonal truncations.
- A processor treats a subset of zonal wave numbers.

## Horizontal representation in grid-point space :

- For a 2D field, there are **NGPTOTG** grid-points, **NDGLG** latitudes, **NDLON** longitudes.
- A processor treats **NGPTOT** points (**NGPTOT** is processor dependent).
- In grid-point calculations, the **NGPTOT** points are sub-divided into **NGPBLKS** packets of **NPROMA** points.
- **NPROMA** is a tunable variable.

## Vertical representation :

- There are **NFLEVG** levels.
- When the second level of distribution is activated, some part of spectral calculations work on a subset of **NFLEVL** levels..
- Vertical discretisation : finite differences (VFD) or finite elements (VFE).

# DISTRIBUTED MEMORY AND DATA ORGANISATION.

## Horizontal representation in spectral space :

- In global models, **NSMAX** is the truncation.
- In LAM models, **NSMAX** and **NMSMAX** are the meridian and zonal truncations.
- A processor treats a subset of zonal wave numbers.

## Horizontal representation in grid-point space :

- For a 2D field, there are **NGPTOTG** grid-points, **NDGLG** latitudes, **NDLON** longitudes.
- A processor treats **NGPTOT** points (**NGPTOT** is processor dependent).
- In grid-point calculations, the **NGPTOT** points are sub-divided into **NGPBLKS** packets of **NPROMA** points.
- **NPROMA** is a tunable variable.

## Vertical representation :

- There are **NFLEVG** levels.
- When the second level of distribution is activated, some part of spectral calculations work on a subset of **NFLEVL** levels..
- Vertical discretisation : finite differences (VFD) or finite elements (VFE).

# MORE DOCUMENTATION.

## Where to find it ?.

- http ://www.cnrm.meteo.fr/gmapdoc (ARPEGE, ALADIN, AROME doc).
- Yessad, K., 2017 : Basics about ARPEGE/IFS, ALADIN and AROME in the cycle 45 of ARPEGE/IFS.

# MORE DOCUMENTATION.

## Where to find it ?.

- http ://www.cnrm.meteo.fr/gmapdoc (ARPEGE, ALADIN, AROME doc).
- Yessad, K., 2017 : Basics about ARPEGE/IFS, ALADIN and AROME in the cycle 45 of ARPEGE/IFS.

# THANK YOU / MERCI.