FULL-POS IN THE CYCLE 30 OF ARPEGE/IFS.

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

June 15, 2005

Contents

1 Introduction.

1.1 Aim of this documentation.

This documentation has to aim to describe FULL-POS in the cycle 30 of ARPEGE/IFS. This documentation is a complement to the FULL-POS users' guide written by Ryad El Khatib and contains a large part of this users' guide.

1.2 Modifications since cycle 29.

- New post-processable (DYN3D-GFL) variables:
  - q_r: rain.
  - q_s: snow.
  - q_g: graupels.
  - q_{TKE}: turbulent kinetic energy.
  - q_{aero}: aerosols.
- The organigramme has been modified under **SPACONVERT** (now calls **SPNH_CONV_NHVAR**).
- The organigramme has been modified under **POS**. For example, routine **EGPHI** has been replaced by the sequence **GPGRXYB** + **GPGRGEO**; **EVERDIA** has been renamed into **GPVERDIA**.
- **FPACHMT** calls **ACSOLW** instead of **ACSOL**.

2 Post-processable fields.

Post-processing can be made on pressure levels, height levels, potential vorticity levels, potential temperature levels or η-levels. Each variable is defined by a sequence of letters [X] which appears in the name of some variables of **YOMAFN**. For example X=T for upper air temperature.

2.1 3D dynamical fields (DYN3D).

* Non derived:

- nd01 and nd02) Horizontal wind components U and V ([U] and [V]).
- nd03) Velocity potential χ [KHI] (has to be fitted).
- nd04) Stream function ψ [PSI] (has to be fitted).
- nd05) Temperature T [T].
- nd06) Specific humidity (moisture) q [Q].
- nd07) Ozone $O3$ [O3MX].
- nd08) Convective precipitation flux CPF [CPF].
- nd09) Stratiform precipitation flux SPF [SPF].
- nd10) Liquid water content q_l [W].
- nd11) Ice content q_i [S].
- nd12) Cloud fraction q_a [CLF].
- nd13) Snow q_s [SN].
- nd14) Rain q_r [RR].
- nd15) Graupels q_g [GR].
- nd16) Turbulent kinetic energy TKE [TKE].
- nd17) Geopotential height gz [Z].
- nd18) Wind velocity [WND].
- nd19) Relative humidity HU [HU].
- nd20) Potential temperature Θ [TH].
- nd21) Moist (irreversible) pseudo-adiabatic potential temperature $\Theta_w^{'}$ [THPW].
- nd22) Equivalent potential temperature Θ_e [ETH].
- nd23) Virtual potential temperature Θ_v [IET].
- nd24) Hydrostatic pressure Π [P].
- nd25) Pressure departure variable [PD] (for NH only).
- nd26) Vertical divergence variable d [VD] (for NH only).
- nd27) Montgomery geopotential $\Phi_{\rm mg}$ [MG].
- nd28) Aerosols $AERO$ [AERO].

*** Extra GFL variables:** Possibility to post-process some extra GFL fields [SCVA].

*** Additional free upper air fields:** Possibility to post-process some additional free upper air fields [FUA].

* Derived:

- de01) True vertical velocity w [VW] (for NH ALADIN only).
- de02) Potential vorticity PV [PV].
- de03) Pressure coordinate vertical velocity ω [VV].
- de04) Stretching deformation STD [STD].
- de05) Shearing deformation SHD [SHD].
- de06) Absolute vorticity $\zeta + f$ [ABS].
- de07) Relative vorticity ζ [VOR].
- de08) Divergence D [DIV].

2.2 2D dynamical fields (DYN2D).

- 01) Surface pressure Π_s [SP].
- 02) Logarithm of surface pressure [LNSP].
- 03) Mean sea level pressure Π_{MSL} [MSL].
- 04) Interpolated (spectral) model orography [FIS].
- 05) Surface NH vertical velocity w_s [WWS].
- 06) Pressure jet [PJET].
- 07) ICAO tropopause temperature [TCAO].
- 08) ICAO tropopause pressure [PCAO].
- 09) Pressure of iso-T=0 [PT0B].
- 10) Altitude of iso-T=0 [HT0B].
- 11) Pressure of iso-T=-10C [PT1B].
- 12) Altitude of iso-T=-10C [HT1B].
- 13) Altitude of iso-$\Theta_w^{'}$=0 [HTPW].
- 14) Mapping factor M [GM].
- 15) Tropopause folding indicator of the iso-2 PVU surface [FOL].
- 16) U-component of wind at 10 meters (pbl) [UCLS].
- 17) V-component of wind at 10 meters (pbl) [VCLS].
- 18) Temperature at 2 meters (pbl) [TCLS].
- 19) Specific humidity at 2 meters (pbl) [QCLS].
- 20) Relative humidity at 2 meters (pbl) [RCLS].
- 21) Wind velocity at 2 meters (pbl) [FCLS].
- 22) Maximum temperature at 2 meters [TX].
- 23) Minimum temperature at 2 meters [TN].
- 24) Maximum relative humidity in CLS HUX [HUX].
- 25) Minimum relative humidity in CLS HUN [HUN].
- 26) CAPE (Convective available potential energy) [CAPE].
- 27) CIEN (Convective inhibition energy) [CIEN].
- 28) Moisture convergence [MOCO].
- 29) Total water vapour content in a vertical column [TVW].
- 30) U-momentum of gusts out of the model [UGST].
- 31) V-momentum of gusts out of the model [VGST].
- 32) Gusts out of the model [FGST].
- 33) Planetary boundary layer depth [HCLP].
- 34) Analysed surface moisture convergence [ASMC].
- 35) Forecast surface moisture convergence [SMC].
- 36) Additional surface fields [FSU].
- 37) ICAO jet zonal component of wind [UJET].
- 38) ICAO jet meridian component of wind [VJET].

2.3 Surface physical fields (PHYSOL).

- 01) Wind gust at 10 m [10FG] (ECMWF).
- 02) U-wind at 10 m [10U] (ECMWF).
- 03) V-wind at 10 m [10V] (ECMWF).
- 04) Dewpoint at 2 m [2D] (ECMWF).
- 05) Temperature at 2 m [2T] (ECMWF).
- 06) Anisotropy coefficient of orography [ACOT].
- 07) Surface albedo [AL].
- 08) Albedo [ALB] (ECMWF).
- 09) Surface albedo for bare soil [ALS] (MF).
- 10) Surface snow albedo [ALSN] (MF).
- 11) Surface albedo for soil with vegetation [ALV] (MF).
- 12) Angle of surface orography [ANOR] (ECMWF).
- 13) Percentage of clay within soil [ARG] (MF).
- 14) Snow albedo [ASN] (ECMWF).
- 15) Surface albedo for non snowed areas [BAAL] (MF).
- 16) Boundary layer dissipation [BLD] (ECMWF).
- 17) Height of boundary layer [BLH] (ECMWF).
- 18) (Roughness length of bare surface)$*g$ [BSR].
- 19) Budget values [BV].
- 20) Convective cloud cover [CCC] (ECMWF).
- 21) Climatological relative deep soil wetness [CDSW].
- 22) Charnock parameter [CHAR] (ECMWF).
- 23) Sea ice cover [CI] (ECMWF).
- 24) Convective precipitation [CP] (ECMWF).
- 25) Convective snow fall [CSF] (ECMWF).
- 26) Climatological relative surface soil wetness [CSSW].
- 27) High vegetation cover [CVH] (ECMWF).
- 28) Low vegetation cover [CVL] (ECMWF).
- 29) Soil depth [D2] (MF).
- 30) Maximum soil depth [D2MX] (MF).
- 31) Direction of main axis of orography [DPAT].
- 32) Deep soil temperature [DST].
- 33) Deep soil wetness [DSW].
- 34) Water evaporation [E] (ECMWF).
- 35) Emissivity [EMIS].
- 36) Evaporation of snow [ES] (ECMWF).
- 37) U-wind stress [EWSS] (ECMWF).
- 38) Frozen deep soil wetness [FDSW] (MF).
- 39) Frozen surface soil wetness [FSSW] (MF).
- 40) (Output grid-point orography)$*g$ [GFIS].
- 41) Gravity wave dissipation [GWD] (ECMWF).
- 42) High cloud cover [HCC] (ECMWF).

- 43) Resistance to evapotranspiration [HV] (MF).
- 44) Interception content [IC] (MF).
- 45) Interpolated dynamic surface $g * z_0$ [IDZ0] (MF).
- 46) Instantaneous surface moisture flux [IE] (ECMWF).
- 47) Instantaneous surface zonal component of stress [IEWSS].
- 48) Increment to mini relative moisture [INCHN] (ECMWF).
- 49) Increment to maxi relative moisture [INCHX] (ECMWF).
- 50) Increment to mini temperature [INCTN] (ECMWF).
- 51) Increment to maxi temperature [INCTX] (ECMWF).
- 52) Instantaneous surface meridian component of stress [INSSS].
- 53) Anisotropy of surface orography [ISOR] (ECMWF).
- 54) Instantaneous surface heat flux [ISSHF].
- 55) Ice surface first level temperature [ISTL1] (ECMWF).
- 56) Ice surface second level temperature [ISTL2] (ECMWF).
- 57) Ice surface third level temperature [ISTL3] (ECMWF).
- 58) Ice surface forth level temperature [ISTL4] (ECMWF).
- 59) Interpolated thermal surface $g * z_0$ [ITZ0] (MF).
- 60) Index of vegetation [IVEG] (MF).
- 61) Leaf area index [LAI] (MF).
- 62) Percentage of land [LAN].
- 63) Low cloud cover [LCC] (ECMWF).
- 64) Zonal gravity wave stress [LGWS] (ECMWF).
- 65) Large scale snowfall [LSF] (ECMWF).
- 66) Land/sea mask [LSM].
- 67) Large scale precipitation [LSP] (ECMWF).
- 68) Large scale precipitation fraction [LSPF] (ECMWF).
- 69) Logarithm of surface roughness [LSRH] (ECMWF).
- 70) Logarithm of z_0 times heat flux [LZ0H] (ECMWF).
- 71) Medium cloud cover [MCC] (ECMWF).
- 72) Meridian gravity wave stress [MGWS] (ECMWF).
- 73) Minimum temperature at 2 m [MN2T] (ECMWF).
- 74) Mean sea level ??? [MSLD] (ECMWF).
- 75) Maximum temperature at 2 m [MX2T] (ECMWF).
- 76) V-wind stress [NSSS] (ECMWF).
- 77) Anisotropy vector U-momentum [PADOU].
- 78) Anisotropy vector V-momentum [PADOV].
- 79) Analysed RMS of geopotential height (CANARI) [PCAAG].
- 80) Forecast RMS of geopotential height (CANARI) [PCAPG].
- 81) Surface relative moisture [PSRHU].
- 82) Proportion of urbanisation [PURB].
- 83) Maximum proportion of vegetation [PVGMX].
- 84) Interpolated surface temperature [RDST].
- 85) Relaxation deep soil wetness [RDSW].

- 86) Runoff [RO] (ECMWF).
- 87) Minimum stomatal resistance [RSMIN] (MF).
- 88) Snow density [RSN] (ECMWF).
- 89) Percentage of sand within soil [SAB] (MF).
- 90) Snow depth [SD].
- 91) (Standard deviation of orography)$*g$ [SDOG].
- 92) Snowfall [SF] (ECMWF).
- 93) (Interpolated grid-point orography)$*g$ [SFIS].
- 94) Skin temperature [SKT] (ECMWF).
- 95) Surface latent heat flux [SLHF] (ECMWF).
- 96) Slope of surface orography [SLOR] (ECMWF).
- 97) Snow melt [SMLT] (ECMWF).
- 98) Surface snow density [SNDE] (MF).
- 99) (Surface roughness)$*g$ [SR].
- 100) Skin wetness [SRC] (ECMWF).
- 101) Surface sensible heat flux [SSHF] (ECMWF).
- 102) Surface solar radiation [SSR] (ECMWF).
- 103) Surface solar radiation clear sky [SSRC] (ECMWF).
- 104) Downward surface solar radiation [SSRD] (ECMWF).
- 105) Sea surface temperature [SST] (ECMWF).
- 106) Surface soil wetness [SSW].
- 107) Surface temperature [ST].
- 108) Soil first level temperature [STL1] (ECMWF).
- 109) Soil second level temperature [STL2] (ECMWF).
- 110) Soil third level temperature [STL3] (ECMWF).
- 111) Soil fourth level temperature [STL4] (ECMWF).
- 112) Surface thermal radiation [STR] (ECMWF).
- 113) Surface thermal radiation clear sky [STRC] (ECMWF).
- 114) Downward surface thermic radiation [STRD] (ECMWF).
- 115) Sunshine duration [SUND] (ECMWF).
- 116) Soil first level wetness [SWL1] (ECMWF).
- 117) Soil second level wetness [SWL2] (ECMWF).
- 118) Soil third level wetness [SWL3] (ECMWF).
- 119) Soil fourth level wetness [SWL4] (ECMWF).
- 120) Total cloud cover [TCC] (ECMWF).
- 121) Total ozone content in a vertical column [TCO3] (ECMWF).
- 122) Total water content in a vertical column [TCW] (ECMWF).
- 123) Total water vapor content in a vert. col. [TCWV] (ECMWF).
- 124) Total liquid water content in a vert. col. [TCLW] (ECMWF).
- 125) Total ice water content in a vert. col. [TCIW] (ECMWF).
- 126) Temperature of snow layer [TSN] (ECMWF).
- 127) Top solar radiation [TSR] (ECMWF).
- 128) Top solar radiation clear sky [TSRC] (ECMWF).

- 129) Top thermal radiation [TTR] (ECMWF).
- 130) Top thermal radiation clear sky [TTRC] (ECMWF).
- 131) High vegetation type [TVH] (ECMWF).
- 132) Low vegetation type [TVL] (ECMWF).
- 133) Percentage of vegetation [VEG].
- 134) Surface roughness [Z0F] (ECMWF).
- 135) (Roughness length for heat)$*g$ [Z0H] (MF).
- 136) (Vegetation roughness length)$*g$ [Z0V] (MF).
- 137) Marine aerosols [ASEA] (MF).
- 138) Continental aerosols [ALAN] (MF).
- 139) Carbone aerosols [ASOO] (MF).
- 140) Desert aerosols [ADES] (MF).
- 141) First ozone profile [O3A] (MF).
- 142) Second ozone profile [O3B] (MF).
- 143) Third ozone profile [O3C] (MF).
- 144) Surface photo active radiation [SPAR] (ECMWF).
- 145) Surface UV-B radiation [SUVB] (ECMWF).
- 146) Convective available potential energy [CAPE] (ECMWF).
- 147) Extra surface fields [VX2] (ECMWF).
- 148) Free surface fields [FSU].

Remark: some of them are used only at ECMWF and can be diagnosed at METEO-FRANCE via some 'DYN2D' or 'CFU' or 'XFU' quantities; some other ones are used only at METEO-FRANCE. Some fields seem to be present twice; for example the only difference between "(Surface roughness)$*g$ [SR]" and "Surface roughness [Z0F] (ECMWF)" is the multiplicative constant g.

2.4 Surface cumulated fluxes (CFU).

- 01) Large scale precipitation [CLSP].
- 02) Convective precipitation [CCP].
- 03) Large scale snow fall [CLSS].
- 04) Convective snow fall [CCSF].
- 05) U-stress [CUSS].
- 06) V-stress [CVSS].
- 07) Surface sensible heat flux [CSSH].
- 08) Surface latent heat flux [CSLH].
- 09) Tendency of surface pressure [CTSP].
- 10) Total cloud cover [CTCC].
- 11) Boundary layer dissipation [CBLD].
- 12) Surface solar radiation [CSSR].
- 13) Surface thermal radiation [CSTR].
- 14) Top solar radiation [CTSR].
- 15) Top thermal radiation [CTTR].
- 16) Convective cloud cover [CCCC].
- 17) High cloud cover [CHCC].
- 18) Medium cloud cover [CMCC].

- 19) Low cloud cover [CLCC].
- 20) U-gravity-wave stress [CUGW].
- 21) V-gravity-wave stress [CVGW].
- 22) Water evaporation [CE].
- 23) Snow sublimation [CS].
- 24) Latent heat [CLHE].
- 25) Latent heat sublimation [CLHS].
- 26) Soil moisture [CWS].
- 27) Snow mass [CSNS].
- 28) Total precipitable water [CQTO].
- 29) Total ozone [CTO3].
- 30) Top mesospheric enthalpy [CTME].
- 31) Solid specific moisture [CICE].
- 32) Liquid specific moisture [CLI].
- 33) Contribution of convection to U [CCVU].
- 34) Contribution of convection to V [CCVV].
- 35) Contribution of convection to q [CCVQ].
- 36) Contribution of convection to c_pT [CCVS].
- 37) Contribution of turbulence to q [CTUQ].
- 38) Contribution of turbulence to c_pT [CTUS].
- 39) Surface clear sky shortwave radiative flux [CSOC].
- 40) Surface clear sky longwave radiative flux [CTHC].
- 41) Surface parallel solar flux [CSOP].
- 42) Top parallel solar flux [CTOP].
- 43) Surface down solar flux [CSOD].
- 44) Surface down thermic flux [CTHD].
- 45) Melt snow [CFON].
- 46) Heat flux in soil [CCHS].
- 47) Water flux in soil [CEAS].
- 48) Surface soil runoff [CSRU].
- 49) Deep soil runoff [CDRU].
- 50) Interception soil layer runoff [CIRU].
- 51) Evapotranspiration flux [CETP].
- 52) Transpiration flux [CTP].
- 53) Surface downward moon radiation [CSMR].
- 54) Top clear sky shortwave radiative flux [CTSOC].
- 55) Top clear sky longwave radiative flux [CTTHC].
- 56) Duration of total precipitations [CDUTP].

2.5 Surface instantaneous fluxes (XFU).

- 01) Total cloud cover [XTCC].
- 02) U-component of wind at 10 meters (pbl) [X10U].
- 03) V-component of wind at 10 meters (pbl)[X10V].
- 04) Temperature at 2 meters (pbl)[X2T].
- 05) Specific humidity at 2 meters (pbl)[X2SH].
- 06) Relative humidity at 2 meters (pbl)[X2RH].
- 07) Convective cloud cover [XCCC].
- 08) High cloud cover [XHCC].
- 09) Medium cloud cover [XMCC].
- 10) Low cloud cover [XLCC].
- 11) Maximum temperature at 2 meters [X2TX].
- 12) Minimum temperature at 2 meters [X2TN].
- 13) Contribution of convection to U [XCVU].
- 14) Contribution of convection to V [XCVV].
- 15) Contribution of convection to q [XCVQ].
- 16) Contribution of convection to c_pT [XCVS].
- 17) Contribution of turbulence to U [XTUU].
- 18) Contribution of turbulence to V [XTUV].
- 19) Contribution of turbulence to q [XTUQ].
- 20) Contribution of turbulence to c_pT [XTUS].
- 21) Contribution of gravity wave drag to U [XGDU].
- 22) Contribution of gravity wave drag to V [XGDV].
- 23) Large scale precipitation [XLSP].
- 24) Convective precipitation [XCP].
- 25) Large scale snow fall [XLSS].
- 26) Convective snow fall [XCSF].
- 27) Surface solar radiation [XSSR].
- 28) Surface thermal radiation [XSTR].
- 29) Top solar radiation [XTSR].
- 30) Top thermal radiation [XTTR].
- 31) Wind at 10 m [X10FF].
- 32) CAPE [XCAPE].
- 33) Moisture convergence [XMOCO].
- 34) Height of the PBL out of the model [XCLPH].
- 35) Gusts out pf the model [XGUST].
- 36) U-momentum of gusts out of the model [XUGST].
- 37) V-momentum of gusts out of the model [XVGST].
- 38) Maximum relative humidity at 2 meters [XX2HU].
- 39) Minimum relative humidity at 2 meters [XX2HV].
- 40) IR cloudy radiance from model [XIRCD].
- 41) IR clear sky radiance from model [XIRCL].
- 42) WV cloudy radiance from model [XWVCD].
- 43) WV clear sky radiance from model [XWVCL].

2.6 Other notations.

- L: number of layers of the model.
- $\left(\frac{dT}{dz}\right)^{st}$: standard atmosphere vertical gradient of the temperature in the troposphere (0.0065 K/m).
- R_d: dry air constant.
- g: gravity acceleration.

3 Horizontal interpolations.

Horizontal interpolations can be bilinear interpolations or 12 points cubic interpolations.

3.1 Bilinear horizontal interpolations.

3.1.1 Horizontal interpolation grid and weights for bi-linear interpolations.

A 16 points horizontal grid is defined as it is shown in figure 3.1. The interpolation point O is between B_1, C_1, B_2 and C_2. Λ and Θ are the longitudes and latitudes on the computational sphere (departure geometry). The following weights are defined as follows:

- zonal weight number 1:
$$ZDLO1 = \frac{\Lambda_O - \Lambda_{B_1}}{\Lambda_{C_1} - \Lambda_{B_1}}$$
- zonal weight number 2:
$$ZDLO2 = \frac{\Lambda_O - \Lambda_{B_2}}{\Lambda_{C_2} - \Lambda_{B_2}}$$
- meridian weight:
$$ZDLAT = \frac{\Theta_O - \Theta_{B_1}}{\Theta_{B_2} - \Theta_{B_1}}$$

3.1.2 Bilinear interpolation.

For a quantity X, are computed successively:

- a linear interpolation on the longitude number 1:
$X_1 = X_{B_1} + ZDLO1(X_{C_1} - X_{B_1})$.
- a linear interpolation on the longitude number 2:
$X_2 = X_{B_2} + ZDLO2(X_{C_2} - X_{B_2})$.
- a meridian linear interpolation:
$X_{interpolated} = X_1 + ZDLAT(X_2 - X_1)$.

In the FULL-POS code the weights are pre-computed in routines **SUHOW2** and **SUHOWLSM**, so the separation of zonal and meridian interpolations is not visible in the interpolation routines.

3.2 12 points horizontal interpolations.

3.2.1 Horizontal interpolation grid and weights for 12 points cubic interpolations.

A 16 points horizontal grid is defined as it is shown in figure 3.2. The interpolation point O is between B_1, C_1, B_2 and C_2. The following weights are defined as follows:

- zonal weight number 0:
$$ZDLO0 = \frac{\Lambda_O - \Lambda_{B_0}}{\Lambda_{C_0} - \Lambda_{B_0}}$$
- zonal weight number 1:
$$ZDLO1 = \frac{\Lambda_O - \Lambda_{B_1}}{\Lambda_{C_1} - \Lambda_{B_1}}$$
- zonal weight number 2:
$$ZDLO2 = \frac{\Lambda_O - \Lambda_{B_2}}{\Lambda_{C_2} - \Lambda_{B_2}}$$
- zonal weight number 3:
$$ZDLO3 = \frac{\Lambda_O - \Lambda_{B_3}}{\Lambda_{C_3} - \Lambda_{B_3}}$$

- meridian weights:

$$ZCLA2 = \frac{(\Theta_O - \Theta_{B_0})(\Theta_O - \Theta_{B_2})(\Theta_O - \Theta_{B_3})}{(\Theta_{B_1} - \Theta_{B_0})(\Theta_{B_1} - \Theta_{B_2})(\Theta_{B_1} - \Theta_{B_3})}$$
$$ZCLA3 = \frac{(\Theta_O - \Theta_{B_0})(\Theta_O - \Theta_{B_1})(\Theta_O - \Theta_{B_3})}{(\Theta_{B_2} - \Theta_{B_0})(\Theta_{B_2} - \Theta_{B_1})(\Theta_{B_2} - \Theta_{B_3})}$$
$$ZCLA4 = \frac{(\Theta_O - \Theta_{B_0})(\Theta_O - \Theta_{B_1})(\Theta_O - \Theta_{B_2})}{(\Theta_{B_3} - \Theta_{B_0})(\Theta_{B_3} - \Theta_{B_1})(\Theta_{B_3} - \Theta_{B_2})}$$

3.2.2 Horizontal 12 points interpolation.

Let us define:

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

For a quantity X, are computed successively:

- a linear interpolation on the longitude number 0:
  $X_0 = X_{B_0} + ZDLO0(X_{C_0} - X_{B_0})$.
- a cubic 4 points interpolation on the longitude number 1:
  $X_1 = X_{A_1} + f_2(ZDLO1)(X_{B_1} - X_{A_1}) + f_3(ZDLO1)(X_{C_1} - X_{A_1}) + f_4(ZDLO1)(X_{D_1} - X_{A_1})$.
- a cubic 4 points interpolation on the longitude number 2:
  $X_2 = X_{A_2} + f_2(ZDLO2)(X_{B_2} - X_{A_2}) + f_3(ZDLO2)(X_{C_2} - X_{A_2}) + f_4(ZDLO2)(X_{D_2} - X_{A_2})$.
- a linear interpolation on the longitude number 3:
  $X_3 = X_{B_3} + ZDLO3(X_{C_3} - X_{B_3})$.
- a meridian cubic 4 points interpolation:
  $X_{interpolated} = X_0 + ZCLA2(X_1 - X_0) + ZCLA3(X_2 - X_0) + ZCLA4(X_3 - X_0)$.

In the FULL-POS code the weights are pre-computed in routines **SUHOW2** and **SUHOWLSM**, so the separation of zonal and meridian interpolations is not visible in the interpolation routines.

3.3 Location of computations.

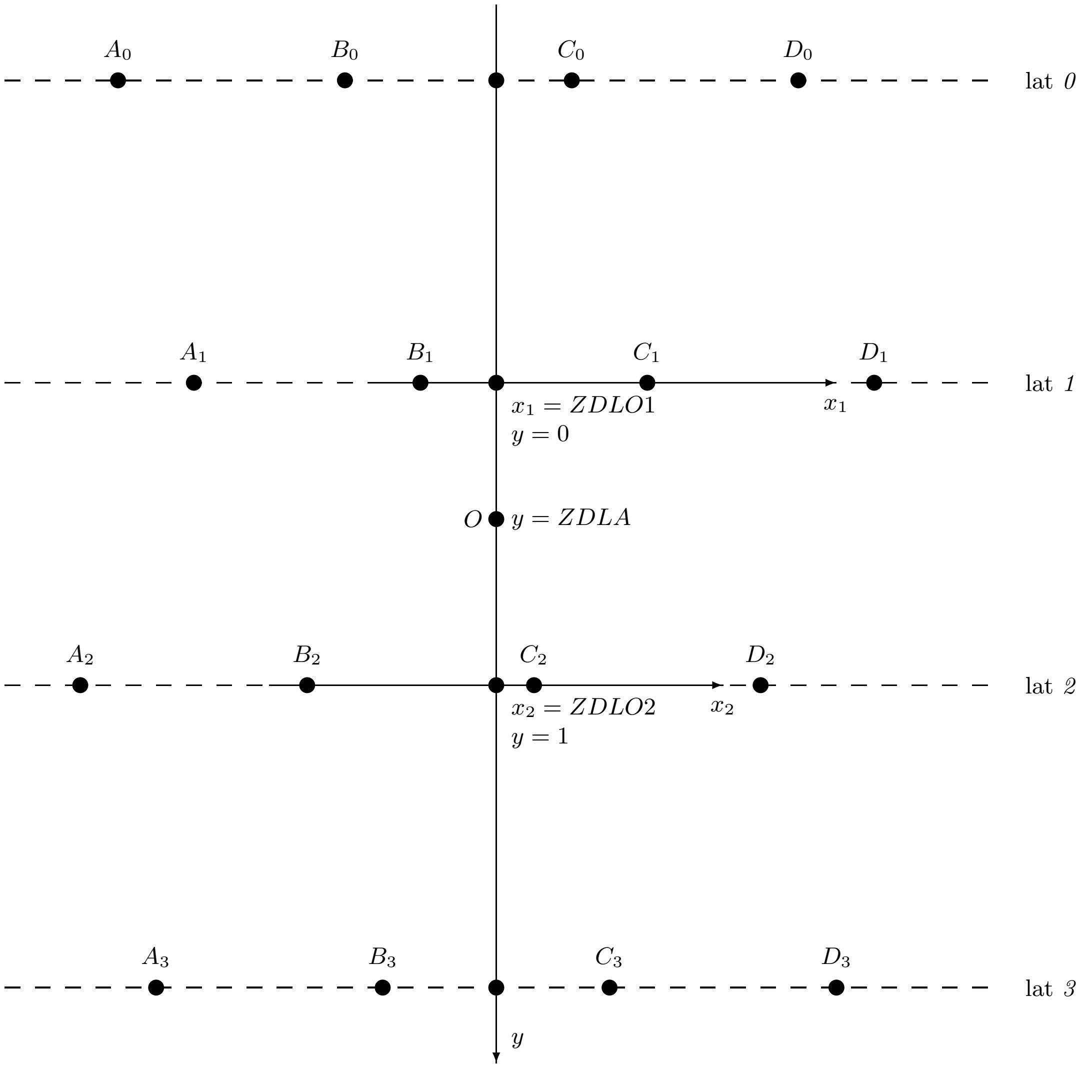
Once the coordinates of the interpolation points known:

- The "north-western" model grid point coordinates are computed in the routine **SUHOW1**. This is the model (departure geometry) grid point which is immediately at the north-west of the interpolation point.
- The weights not modified by the land-sea mask are computed in routine **SUHOW2**.
- The weights modified by the land-sea mask are computed in routine **SUHOWLSM**. This is equivalent to use weights not modified by the land-sea mask and to multiply the field to be interpolated by the land-sea mask (0 if sea, 1 if land).
- The horizontal bilinear interpolations are done by routine **FPINT4**.
- The horizontal 12 points interpolations are done by routine **FPINT12**.
- Horizontal interpolations need intermediate quantities computed in routine **FPSCAW**.

Additional modifications can be done after 12 points interpolations: they are done by routine **FPHOR12**, for example add a monotonicity condition. More details about these additional modifications and the post-processed fields concerned by these modifications are described in the part describing each routine.

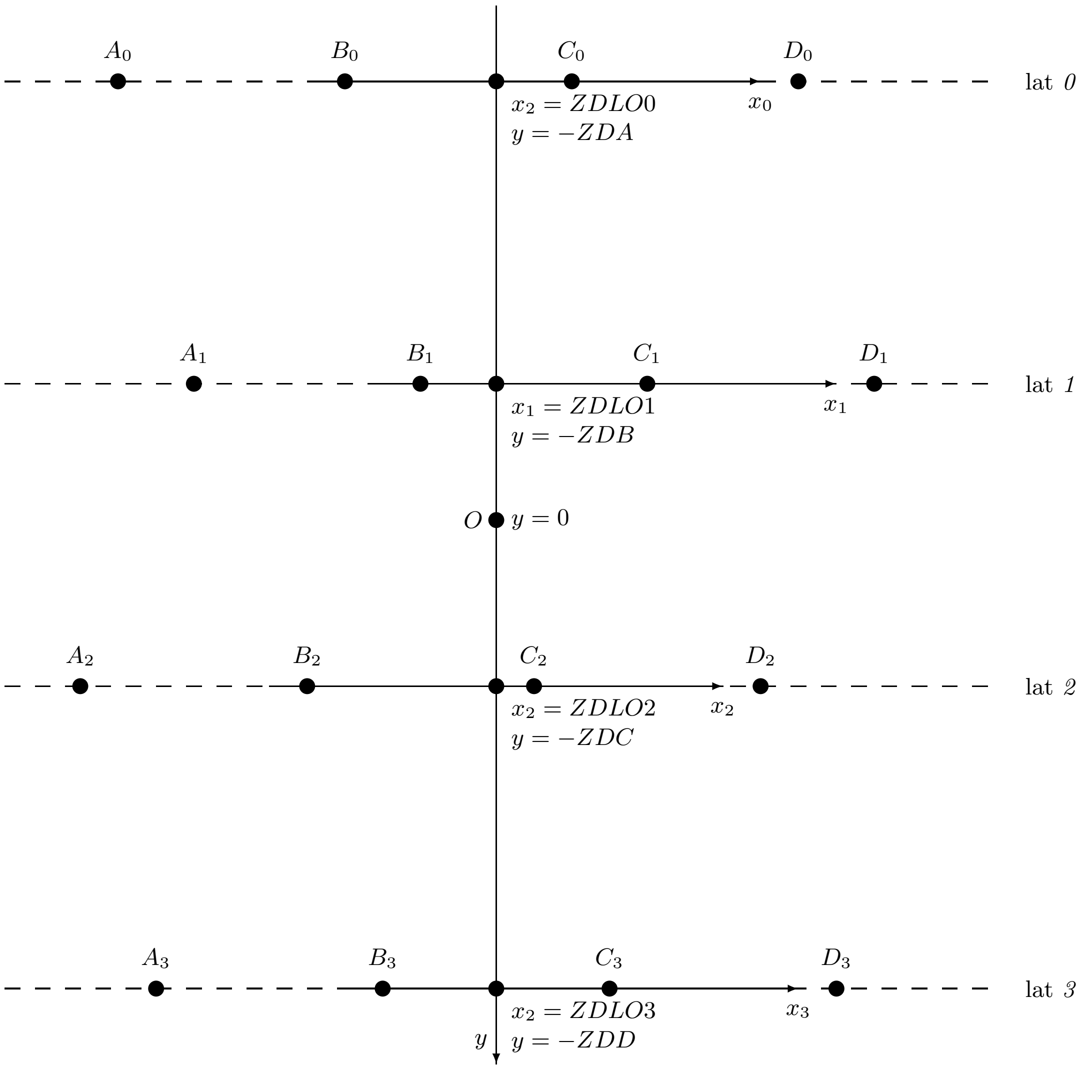
3.4 Plane geometry (ALADIN).

All previous formulae for weight computation can be used for an irregular latitude spacing and a different number of points on each longitude. The ALADIN grid has a horizontal regular spacing, so the previous formulae can be simplified. **SUEHOW1**, **SUEHOW2** and **SUEHOWLSM** are called instead of **SUHOW1**, **SUHOW2** and **SUHOWLSM**.



$x_1(B_1) = x_2(B_2) = 0$; $x_1(C_1) = x_2(C_2) = 1$.

Figure 3.1: Interpolation horizontal grid for bilinear interpolations.



$x_0(B_0) = x_1(B_1) = x_2(B_2) = x_3(B_3) = 0$; $x_0(C_0) = x_1(C_1) = x_2(C_2) = x_3(C_3) = 1$.

Figure 3.2: Interpolation horizontal grid for 12 points interpolations.

4 Vertical interpolations and extrapolations.

4.1 General considerations.

For 3D variables to be vertically interpolated, vertical interpolations are generally linear interpolations between the layers where are defined model variables. The treatment of the extrapolations above the upper layer, the extrapolations below the lower layer or the surface depend on the variable considered. In particular cases some variables can be diagnosed using the vertically interpolated value of some other variables.

The "**OLD**" versions of routines when available are used at ECMWF (switch **LOLDPP**=.T.). METEO-FRANCE uses the option **LOLDPP**=.F. .

4.2 More details for 3D dynamical variables.

4.2.1 Wind components, wind velocity.

∗ Way of interpolating if LOLDPP=.T. (routine PPUV_OLD):

- Linear interpolation between the layer 2 and the lower layer.
- The coordinate used for linear interpolation is the logarithm of the pressure.
- Quadratic interpolation between the layer 1 and the layer 2 using the values of the layers 1, 2 and 3.
- Quadratic interpolation between the top and the layer 1 using the values of the top, layers 1 and 2; the value of the top is obtained by a linear extrapolation from the values of the layers 1 and 2.
- The coordinate used for quadratic interpolation is the logarithm of the pressure.
- Extrapolation below the middle of the lower layer and below the surface assumes that the quantity is constant.

∗ Way of interpolating if LOLDPP=.F. (routine PPUV): The same as for **LOLDPP**=.T. but quadratic interpolations are replaced by linear interpolations.

4.2.2 Temperature.

Applies to temperature if the vertical coordinate of post-processing is not the potential vorticity, otherwise see routine **PP2DINT**.

∗ Way of interpolating if LOLDPP=.F. (routine PPT):

- Quadratic interpolation between the middles of the upper and lower layers.
- Quadratic interpolation between the top and the middle of the upper layer: the top value of the temperature is assumed to be equal to the value of the middle of the upper layer; due to the fact that the interpolation is a quadratic one, that does not mean that the temperature is constant in this atmosphere depth.
- The coordinate used for quadratic interpolation is the logarithm of pressure. For more details about the quadratic interpolation used, which is a quadratic analytic expression of the logarithm of pressure, and the reason of using a quadratic interpolation, see (Undén, 1995).
- A surface temperature T_{surf} is computed as follows:

$$T_{\mathrm{surf}} = T_L + \left(\frac{dT}{dz}\right)^{st} \frac{R_d}{g} \left(\frac{\Pi_s}{\Pi_L} - 1\right) T_L \tag{1}$$

- Extrapolation below the middle of the lower layer and the surface is a linear interpolation between T_L and T_{surf}.
- Extrapolation under the surface is made according a more complicated algorithm:

$$T_{\mathrm{extrapolated}} = T_{\mathrm{surf}} \left(1 + y + \frac{y^2}{2} + \frac{y^3}{6}\right) \tag{2}$$

where:

$$y = \Gamma \frac{R_d}{g} \log\left(\frac{\Pi_s}{\Pi_L}\right) \tag{3}$$

Expression of Γ: $\Gamma = \left(\frac{dT}{dz}\right)^{st}$ if $\Phi_s/g < 2000$ m; if $\Phi_s/g \geq 2000$ m expression of Γ is more complicated:

$$\Gamma = \frac{g}{\Phi_s} \max(T_0^{'} - T_{\text{surf}}, 0) \tag{4}$$

Expression of $T_0^{'}$:

– if $\Phi_s/g > 2500$ m:

$$T_0^{'} = \min\left(T_{\text{surf}} + \left(\frac{dT}{dz}\right)^{st} \frac{\Phi_s}{g}, 298\text{K}\right) \tag{5}$$

– if $\Phi_s/g \leq 2500$ m and $\Phi_s/g \geq 2000$ m: $T_0^{'}$ is computed by a linear interpolation (coordinate of interpolation is Φ_s) between the two values $\min\left(T_{\text{surf}} + \left(\frac{dT}{dz}\right)^{st} \frac{\Phi_s}{g}, 298\text{K}\right)$ and $T_{\text{surf}} + \left(\frac{dT}{dz}\right)^{st} \frac{\Phi_s}{g}$.

∗ Way of interpolating if LOLDPP=.T. (routine PPT_OLD):

- Linear interpolation (between the upper and the lower layer).
- The coordinate used for linear interpolation is the pressure.
- Extrapolation above the middle of the upper layer assumes that the quantity is constant.
- Extrapolation below the middle of the lower layer and the surface is a linear interpolation between T_L and T_{surf} like in **PPT**.
- Extrapolation under the surface is made according the same algorithm as in **PPT** (code of part 1.4 is different in **PPT** and in **PPT_OLD** but actually does the same calculations).

4.2.3 Geopotential height.

Applies to geopotential height (and Montgomery geopotential) if the vertical coordinate of post-processing is not the potential vorticity, otherwise see routine **PP2DINT**.

∗ Way of interpolating if LOLDPP=.T. (routine PPGEOP_OLD):

- The variable interpolated is a geopotential height departure from a reference defined by a standard atmosphere without any orography. After the interpolation an increment is added, sum of the surface orography and the "standard" geopotential height depth between the pressure level of interpolation and the actual surface. This method avoids to introduce interpolations for the standard component of the geopotential height which can be computed analytically (in routine **PPSTA**).
- Linear interpolation between the layer 2 and the surface.
- The coordinate used for linear interpolation is the logarithm of the pressure.
- Quadratic interpolation between the layer 1 and the layer 2 using the values of the layers 1, 2 and 3.
- Quadratic interpolation between the top and the layer 1 using the values of the top, layers 1 and 2.
- The coordinate used for quadratic interpolation is the logarithm of the pressure.
- Extrapolation below surface uses the surface temperature T_{surf} of equation (1).

$$gz_{\text{extrapolated}} = \Phi_s - R_d T_{\text{surf}} \log\left(\frac{\Pi_s}{\Pi_L}\right)\left(1 + \frac{y}{2} + \frac{y^2}{6}\right) \tag{6}$$

  where y is defined by formula (3) with $\Gamma = \left(\frac{dT}{dz}\right)^{st}$ in all cases.
- For more details about this algorithm, see (Andersson and Courtier, 1992) which is still valid for the cycle 30.

∗ Way of interpolating if LOLDPP=.F. (routine PPGEOP):

- The variable interpolated is a geopotential height departure from a reference defined by a standard atmosphere without any orography. After the interpolation an increment is added, sum of the surface orography and the "standard" geopotential height depth between the pressure level of interpolation and the actual surface. This method avoids to introduce interpolations for the standard component of the geopotential height which can be computed analytically (in routine **PPSTA**).
- Quadratic interpolation between the middles of the upper and lower layers.

- Quadratic interpolation between the top and the middle of the upper layer.
- The coordinate used for quadratic interpolation is the logarithm of pressure. The quadratic interpolation is not exactly the same as for **LOLDPP**=.T., it is a quadratic analytic expression of the logarithm of pressure of the same type as the one used to post-process the temperature for **LOLDPP**=.F. . For more details about the quadratic interpolation used, and the reason of using a quadratic interpolation, see (Undén, 1995).
- Linear interpolation between the lower layer and the surface, as for **LOLDPP**=.T. .
- Extrapolation below surface uses the same algorithm as for **LOLDPP**=.T. .

4.2.4 Variables interpolated using routine PP2DINT.

∗ List of variables:

- Geopotential height gz if vertical coordinate is potential vorticity.
- Temperature T if vertical coordinate is potential vorticity.
- Relative vorticity ζ.
- Divergence D.
- Potential temperature Θ if vertical coordinate is not potential temperature.
- Velocity potential χ.
- Stream function ψ.
- Equivalent potential temperature Θ_e.
- Absolute vorticity $\zeta + f$.
- Stretching deformation STD.
- Shearing deformation SHD.
- Potential vorticity PV.
- True vertical velocity w for non hydrostatic model (defined on interlayers).

∗ Way of interpolating:

- Linear interpolation (between the upper and the lower layer for quantities defined on the middle of layers, between the layer 1 and the surface for quantities defined on interlayers).
- The coordinate used for linear interpolation is the pressure.
- Extrapolation above the middle of the upper layer assumes that the quantity is constant.
- Extrapolation below the middle of the lower layer and below the surface assumes that the quantity is constant.

4.2.5 GFL variables (moisture, ozone, liquid water, solid water, cloud fraction, rain, snow, graupels, TKE, aerosols, extra GFL variables), pressure departure variable: variables using routine PPQ.

∗ Way of interpolating (routine PPQ):

- Linear interpolation (between the upper and the lower layer).
- The coordinate used for linear interpolation is the pressure.
- Extrapolation above the middle of the upper layer assumes that the quantity is constant.
- Extrapolation below the middle of the lower layer and below the surface assumes that the quantity is constant.

4.2.6 Relative humidity (routine PPRH).

∗ Way of interpolating (routine PPRH):

- Linear interpolation (between the upper and the lower layer).
- The coordinate used for linear interpolation is the pressure.
- Extrapolation above the middle of the upper layer assumes that the quantity is constant.
- Extrapolation below the middle of the lower layer and below the surface assumes that the quantity is constant.

4.2.7 Pressure coordinate vertical velocity ω (routine PPVVEL).

$*$ Way of interpolating (routine PPVVEL):

- Linear interpolation (between the upper and the lower layer).
- The coordinate used for linear interpolation is the pressure.
- Extrapolation above the middle of the upper layer is a linear interpolation between a zero value at the top and the value of the upper layer.
- Extrapolation between the middle of the lower layer and the surface assumes that the quantity is constant.
- Extrapolation below the surface assumes that the quantity is zero.

4.2.8 Moist (irreversible) pseudo-adiabatic potential temperature $\Theta_w^{'}$ (routine PPTHPW).

Routine **PPTHPW** is a diagnostic one. It takes as input the vertically post-processed pressure, temperature, moisture, liquid water and ice and computes $\Theta_w^{'}$ at the post-processing levels using a diagnostic (and rather complicated) algorithm.

4.2.9 Vertical divergence variable (currently d).

One uses the vertically post-processed vertical velocity w (done by **PP2DINT**) then diagnoses the post-processed vertical divergence (for both **NVDVAR**=3 or 4).

4.3 2D dynamical variables which need extrapolations.

4.3.1 Mean sea level pressure Π_{MSL} (routine PPPMER).

If $\mid \Phi_s \mid$ is lower than 0.001 J/kg the mean sea level pressure is set to the surface pressure. In the other cases one uses the following algorithm:

- One computes the surface temperature T_{surf} of equation (1) and the " mean sea level " temperature $T_0 = T_{\mathrm{surf}} + \left(\frac{dT}{dz}\right)^{st} \frac{\Phi_s}{g}$.
- To avoid extrapolation of too low pressures over high and warm surfaces the following modifications are done:
  - if $T_0 > 290.5$ K and $T_{\mathrm{surf}} \leq 290.5$ K, Γ is defined by:

$$\Gamma = (290.5 - T_{\mathrm{surf}}) \frac{g}{\Phi_s} \tag{7}$$

  - if $T_0 > 290.5$ K and $T_{\mathrm{surf}} > 290.5$ K, Γ is set to 0, T_{surf} is modified and set to 0.5$*$(290.5 K + old value of T_{surf}).
- To avoid extrapolation of too high pressures over cold surfaces the following modifications are done when $T_{\mathrm{surf}} < 255$ K: Γ is set to $\left(\frac{dT}{dz}\right)^{st}$ and T_{surf} is modified and set to 0.5$*$(255 K + old value of T_{surf}).
- In the other cases Γ is set to $\left(\frac{dT}{dz}\right)^{st}$.
- Mean sea level pressure is computed as follows:

$$\Pi_{\mathrm{MSL}} = \Pi_s \exp\left[\frac{\Phi_s}{R_d T_{\mathrm{surf}}}\left(1 - \frac{x}{2} + \frac{x^2}{3}\right)\right] \tag{8}$$

  where:

$$x = \frac{\Gamma \Phi_s}{g T_{\mathrm{surf}}} \tag{9}$$

5 Filtering in spectral space.

5.1 General considerations.

Three filterings are successively applied:

- 1) A "THX" filter on derivatives in ARPEGE, or a bell-shaped filter on derivatives for ALADIN.
- 2) A bell-shaped filter on all fields (derivatives and not derivatives).
- 3) A "THX" filter on all fields.

Filtering is done in routine **pp_obs/SPOS** (**pp_obs/ESPOS** for ALADIN).

5.2 "THX" (ARPEGE) or bell-shaped (ALADIN) filtering on derivatives.

$*$ **Cases where this filtering is used:** This filtering applies in the spectral space to absolute vorticity, relative vorticity, divergence, vertical velocity, stretching and shearing deformations, and potential vorticity (and extend to all variables if the vertical coordinate of post-processing is the potential vorticity). This filter is active if:

- **CFPFMT** is not 'MODEL' in namelist **NAMFPC**.
- **LFPFIL**=.TRUE. in namelist **NAMFPF** for ARPEGE, or **LFPBED**=.TRUE. in namelist **NAMFPF** for ALADIN.
- Variable **NFMAX** is smaller than the "equivalent unstretched sphere" truncation N_c (in practical N_c is between $1.1 * c * N_s$ and $1.2 * c * N_s$, where c is the stretching factor).

If you wish to keep these fields unfiltered, then just set **LFPFIL**=.FALSE. in ARPEGE/IFS, or **LFPBED**=.FALSE. in ALADIN (namelist **NAMFPF**). On the other hand, you can keep the filter active but you can tune the filtering function.

$*$ **Function of filtering:** In ARPEGE/IFS, this function looks like a smoothed step function; for a given total wavenumber n in the unstretched spectral space (i.e. the spectral space of the "equivalent unstretched sphere" of truncation N_c), the formula is:

$$f_{\mathrm{THX}}(n) = \frac{1 - \tanh(e^{-k}(n - n_0))}{2} \tag{10}$$

(The use of the function hyperbolic tangent is the reason of the nickname "THX" for this filter). It means that this function equals roughly 1 if n is less than n_0, and 0 if it is bigger than n_0.

$*$ **Tunable parameters in the previous function:** k and n_0 are tunable parameters:

- n_0 is in the variable **NFMAX**. If **CFPFMT**='GAUSS' or 'MODEL' in namelist **NAMFPC**, the default value is **NFPMAX**$*$**FPSTRET**; else, it is the truncation of the Gaussian grid which would have been defined by **NLAT** latitudes and **NLON** longitudes with default so that $3 * \mathbf{NFMAX} + 1 = \min(\mathbf{NLAT}, \mathbf{NLON})$ (see **YOMFPD** and namelist **NAMFPD**).
- k is stored in variable **RFPBED**.

$*$ **Operations done in SPOS for this filtering:** One assumes that **CFPFMT** is not 'MODEL'.

- Spectral post-processed data are stored in array **SPBFP**. This is true for both filtered and unfiltered data.
- **LFPFIL**=.T. and $N_s < N_c$: the previous function $f_{\mathrm{THX}}(n)$ is computed in the "equivalent unstretched sphere" of truncation N_c, so reading dilatation and contraction matrixes respectively denoted by $\mathcal{D}$ and $\mathcal{C}$ (computed by the configuration 911 of ARPEGE/IFS, see the corresponding documentation (IDRD)) is necessary. The operator applied to spectral fields in the computational sphere is a matricial operator $\mathcal{C} * f_{\mathrm{THX}}(n) * \mathcal{D}$ pre-computed in the routine **pp_obs/FPFILTER** (called by **setup/SU3FPOS**) and stored in the array **RFPMAT**. Filtering is done only if **NFMAX** $< N_c$.

  Remarks about this matricial operator:

  - It is now possible to store this matricial operator on a file by setting **LFPWRFIL**=.T. in **NAMFPF**.
  - This file can be read by setting **LFPRDFIL**=.T. in **NAMFPF**. It saves a lot of CPU time compared to the method of reading the dilatation/contraction matrices.

- **LFPFIL**=.T. and $N_s = N_c$: identity $N_s = N_c$ is satisfied if the model resolution has no stretching. $f_{\mathrm{THX}}(n)$ is stored in the array **RFPFIL**. Filtering is done only if **NFMAX** $< N_s$.
- **LFPFIL**=.F.: no filtering.

∗ **ALADIN:** In ALADIN, this filter is a "bell-shaped" function and is done in the computational geometry. For a given pair of wavenumbers (n, m), the formula is:

$$f_{\mathrm{bsAL}}(n,m) = e^{\frac{-k}{2}((n/N)^2+(m/M)^2)} \tag{11}$$

where N is the model truncation (**NSMAX**) and k a tunable variable. k is in variable **RFPBED** of namelist **NAMFPF**.

5.3 "bell-shaped" filtering on non-derivative fields.

∗ **Way of filtering:** It is also possible to filter the non-derivative post-processed fields through "bell-shaped" filters. Separate "Bell-shaped" filters are available for geopotential height, temperature, mean sea level pressure, relative humidity and all other non-derivatives. By default, these filters are active for geopotential height, temperature, mean sea level pressure and relative humidity. In ALADIN, the formula is the same as above (formula (11)). In ARPEGE/IFS, for a given wavenumber n in the stretched spectral space, the formula is:

$$f_{\mathrm{bsARP}}(n) = e^{\frac{-k}{2}(n/N_s)^2} \tag{12}$$

where N_s is the model truncation (**NSMAX**) and k a tunable variable. In **SPOS**, the bell-shaped filtering is done by multiplying the array **SPBFP** by $f_{\mathrm{bsARP}}(n)$, the result is still in **SPBFP**.

∗ **Variables controlling bell-shaped filtering:**

- Switch **LFPBED** for derivatives (**LFPBED** is used only in ALADIN): .TRUE. if filtering, .FALSE. if no filtering.
- Variable **RFPBED** for derivatives (**RFPBED** is used only in ALADIN): to control the intensity of the filtering (variable k of formula (12)).

6 Post-processing files.

- Departure file (ICMSH[code]INIT) is read on logical unit number 81.
- If FULL-POS configurations are not of the type 927 (**LFPSPEC**=.F.) post-processing fields are written on logical unit number 54 (file PF[code]DDDDDD+0000), where DDDDDD is the content of the character variable **CFPDOM** of **YOMFPC**.
- If FULL-POS configurations are of the type 927 (**LFPSPEC**=.T.) post-processing fields are written on logical unit number 54 (file PF[code]DDDDDD+0000), and intermediate data corresponding to model fields interpolated in the final geometry at the end of the first part of the FULL-POS run are written on logical unit 91. The second part of the FULL-POS run first reads unit 91, does spectral fit or spectral transformations of some fields and writes them on unit 91; then unit 91 is read as an ARPEGE file like a departure file.
- Departure geometry climatology is read on logical unit 10 (file Const.Clim).
- Arrival geometry climatology is read on logical unit 54 (file Const.Clim.DDDDDD).

7 Organigramme

When not precised, directory is **pp_obs** (except for **LFI...** routines which are in the auxiliary library).

7.1 Setup routines and arborescence above STEPO.

∗ **General architecture under CNT0:** Only features concerning FULL-POS are mentioned.

```
control/CNT0 ->
* setup/SU0YOMA ->
  - setup/SUCT0
  - setup/SUAFN ->
    * setup/SUAFN1
    * setup/SUAFN2 -> CTRL_TFP
    * setup/SUAFN3 -> setup/suafn3.F90/PRINT_TFP and setup/suafn3.F90/PRINT_GFP
  - setup/SUFPC ->
    * FA and LFI routines
    * setup/SUFPDIM
  - setup/SUDIM2
  - setup/SUFPD -> ald/utility/EGGX_N -> (arborescence not detailed)
  - setup/SUGPPRP -> pp_obs/FPINIPHY
  - setup/SUALLO -> (arborescence not detailed)
* setup/SU0YOMB ->
  - setup/SUMP -> (arborescence not detailed)
  - setup/SUCFU -> setup/SUCFUFP
  - setup/SUXFU -> setup/SUXFUFP
  - setup/SUSC2A
  - setup/SUSC2B ->
    * module/ALLOCATE_GRIDPOINT_BUFFER for allocations
    * parallel/SLCSET -> parallel/SLRSET -> parallel/SET2PE,
      processor communication routines MPL_SEND and MPL_RECV.
    * Some transposition routines (do not touch FULL-POS).
  - setup/SUBFPOS -> (see below architecture under SUBFPOS).
* control/CNT1 -> (see below arborescence under CNT1).
```

∗ **General architecture under CNT1:** Only features concerning FULL-POS are mentioned.

```
control/CNT1 ->
* setup/SU1YOM ->
  - setup/SUINIF -> (see below arborescence under SUINIF).
  - other routines without any relationship with FULL-POS.
* control/CNT2 -> control/CNT3 -> (see below arborescence under CNT3).
```

∗ **General architecture under CNT3:** Only features concerning FULL-POS are mentioned.

```
control/CNT3 ->
* setup/SU3FPOS ->
  - setup/SURFPBUF ->
    * c9xx/RDCLIMO -> (see below general architecture under RDCLIMO)
    * c9xx/RDECCLIMO (GRIB version of RDCLIMO) -> GRIB routines.
    * utility/SC2WRGFP
    * processor communication routines parallel/ISNDGPFFP, parallel/IRCVGPFFP,
      parallel/DISGRIDFP,
      which call parallel/SET2PE, MPL_SEND and MPL_RECV;
      processor communication routine MPL_BARRIER.
    * ALADIN processor communication routines parallel/ISNDEZON,
      parallel/IRCVEZON (not detailed).
    * setup/SUFPSC2B
    * utility/SC2WRGFP
  - setup/SUHOW1 (ARPEGE) or setup/SUEHOW1 (ALADIN)
  - setup/SUWFPBUF ->
    * setup/SUHOW2 (ARPEGE) or setup/SUEHOW2 (ALADIN)
    * pp_obs/CPCLIMI ->
      - pp_obs/FPSCAW
      - utility/SC2RDG -> utility/EMPTB.
      - parallel/SLCOMM (DM) -> processor communication routines MPL_SEND and MPL_RECV.
      - parallel/SLEXTPOL (DM)
    * pp_obs/CPCLIMO ->
      - ARPEGE: setup/SUHOW2 (ARPEGE) or setup/SUEHOW2 (ALADIN)
      - pp_obs/FPSCAW
      - utility/SC2RDG -> utility/EMPTB.
      - parallel/SLCOMM (DM) -> processor communication routines MPL_SEND and MPL_RECV.
      - parallel/SLEXTPOL (DM)
      - module/type_fprqphys.F90/ALLOCATE_FPRQPHY
      - pp_obs/FPINTPHY -> FPINT12 and FPINT4.
      - pp_obs/FPCORPHY -> sur/external/SURF_INQ, FPLAKE, FPHOR12 and phys_dmn/ACSOLW.
      - pp_obs/FPSAMPL -> FPINT12 and FPINT4.
    * c9xx/RDCLIMO -> (see below general architecture under RDCLIMO)
    * c9xx/RDECCLIMO (GRIB version of RDCLIMO) -> GRIB routines.
    * processor communication routines parallel/ISNDGPFFP, parallel/IRCVGPFFP,
      which call parallel/SET2PE, MPL_SEND and MPL_RECV;
      processor communication routine MPL_BARRIER.
    * setup/SUHOWLSM (ARPEGE) or setup/SUEHOWLSM (ALADIN)
      - processor communication routine MPL_ALLREDUCE
    * setup/SUFPSC2B
    * utility/SC2WRGFP
* control/CNT4 -> (see below arborescence under CNT4).
* setup/SU4FPOS ->
  - pp_obs/PPRESET (ECMWF, arborescence not detailed).
  - pp_obs/PPREQ
  - setup/SUFPPHY -> setup/SUFPDOM
  - setup/SUFPDYN -> setup/SUFPDOM
* control/CPREP4 -> (see below arborescence under CPREP4).
```

∗ **General architecture under CNT4:** Only features concerning FULL-POS are mentioned.

```
control/CNT4 ->
* utility/RDSPEC -> (arborescence not detailed).
* setup/SU4FPOS ->
  - pp_obs/PPRESET (ECMWF, arborescence not detailed).
  - pp_obs/PPREQ
  - setup/SUFPPHY -> setup/SUFPDOM
  - setup/SUFPDYN -> setup/SUFPDOM
* pp_obs/GRIDFPOS -> (see below arborescence under GRIDFPOS).
* pp_obs/DYNFPOS -> (see below arborescence under DYNFPOS).
* control/CPREP4 -> (see below arborescence under CPREP4).
```

∗ **General architecture under SUBFPOS:**

```
setup/SUBFPOS ->
* setup/SUALFPOS -> module/type_fprqdyns.F90/ALLOCATE_FPRQDYN and module/type_fprqdyns.F90/INQUIRE_FPRQDYN
* setup/SUFPG ->
  - setup/SUFPG1 ->
    * setup/DFLTVV1
    * setup/SUGAW (arborescence is not described in detail).
    * setup/SUNMEN
  - setup/SUFPG2 ->
    * the ALADIN routine ald/pp_obs/SUFPMOVE (arborescence not described in detail)
    * ald/utility/EGGX_N -> (arborescence not described in detail)
    * transform/TRACARE
    * setup/SUGENORD
    * ald/utility/EGGDIR
    * ald/utility/EGGRVS
    * ald/utility/ELALO2XY
    * transform/TRARECA
  - the ALADIN routine SUEFPG3.
  - the ALADIN routine SUFPEZO (arborescence not described in detail).
  - setup/SUMPFPOS -> setup/SUPROCFP -> setup/SUPROCGP -> parallel/SET2PE
* setup/SUFPWIDE
  - parallel/SLCSET -> parallel/SLRSET -> parallel/SET2PE,
    processor communication routines MPL_SEND and MPL_RECV.
* setup/SUFPF
* setup/SUFPOPH -> setup/SUFRAME -> FACADE (or SUEFRAME if plane geometry,
  or SUFPGRIB if GRIB files).
* setup/SUFPSC2 ->
  - parallel/BROADCINT -> processor communication routines MPL_SEND and MPL_RECV.
* setup/SURFPDS -> setup/surfpds.F90/SET_NEXT
* setup/SUWFPDS
* setup/SUFPTR2 -> setup/sufptr2.F90/SET_NEXT
* setup/SUFPIOS
```

∗ **General architecture under SUINIF:** Only features concerning FULL-POS are mentioned. For arborescence under **transform/REESPE**, **transform/SPEREE**, **transform/UVSPE**, **transform/TRANSDIRH**, see documentation (IDTS) about spectral transforms.

```
setup/SUINIF ->
* pp_obs/SPECFITADM -> (see below arborescence under SPECFITADM)
* pp_obs/SPECFITG (GRIB version of SPECFITADM, arborescence not detailed)
* setup/SUSPEC ->
  - setup/SUSPECB for 2D model (arborescence not detailed)
  - setup/SUSPECA, ARPEGE files (arborescence not detailed)
  - setup/SUSPECG for GRIB files (arborescence not detailed)
  - setup/SUSPECG1 for GRIB files (arborescence not detailed)
  - setup/SUSPECTCFOU for transmission coefficients (arborescence not detailed)
  - setup/SUOROG (arborescence not detailed)
* setup/SUGRIDF -> (see below arborescence under SUGRIDF)
* setup/SUGRIDU -> (arborescence not detailed)
* adiab/SPECRTDM, or ESPECRT for ALADIN (arborescence not detailed)
* setup/SUGRCFU for ARPEGE files (arborescence not detailed)
* setup/SUGRXFU for ARPEGE files (arborescence not detailed)
```

∗ **General architecture under SPECFITADM:** Only features concerning FULL-POS are mentioned. For arborescence under **transform/REESPE**, **transform/SPEREE**, **transform/UVSPE** see documentation (IDTS) about spectral transforms.

```
pp_obs/SPECFITADM ->
* parallel/RDPXFA ->
  - utility/OPENFA -> utility/CHIEN, utility/CCHIEN, LFI end FA routines.
  - parallel/DISGRID_C (processor communication routine, arborescence not detailed)
  - LFI... and FA... routines.
* setup/SUGRIDF -> (see below arborescence under SUGRIDF)
* utility/SC2RDG -> utility/EMPTB
* utility/SC2WRG -> utility/FILLB
* setup/SUGRCFU -> (arborescence not detailed)
* setup/SUGRXFU -> (arborescence not detailed)
* transform/TRANSDIRH (ARPEGE) or transform/ETRANSDIRH (ALADIN)
* transform/REESPE (ARPEGE) or transform/EREESPE (ALADIN)
* transform/SPEREE (ARPEGE) or transform/ESPEREE (ALADIN)
* setup/SUOROG (arborescence not detailed)
* transform/SPECRTDM (ARPEGE) or transform/ESPECRT (ALADIN)
```

∗ General architecture under SUGRIDF:

```
setup/SUGRIDF ->
* setup/SUGRIDADM for DM code, ARPEGE files (arborescence not detailed)
* setup/SUCACLIADM for DM code, ARPEGE files (arborescence not detailed)
* setup/SUGRCLIA for ARPEGE files (arborescence not detailed)
* setup/SUGRIDVADM for DM code, ARPEGE files (arborescence not detailed)
* setup/SUGRIDG for GRIB files (arborescence not detailed)
* utility/SC2RDG -> utility/EMPTB
* utility/SC2WRG -> utility/FILLB
* utility/GPNORM2 (arborescence not detailed)
```

* **General architecture under RDCLIMO:**

```
c9xx/RDCLIMO ->
* dia/INI1WRFP
* dia/INI3WRFP
* dia/SUPPDATE -> some FA and LFI routines.
* pp_obs/OPENFPFA -> utility/CHIEN, utility/ECHIEN (ALADIN routine), LFI end FA routines.
* some FA routines.
```

* **General architecture under GRIDFPOS:** Only features concerning FULL-POS are mentioned.

```
pp_obs/GRIDFPOS ->
* pp_obs/PREGPFPOS ->
  - utility/CPGRIDF
  - utility/SC2RDG -> utility/EMPTB
  - utility/SC2WRG -> utility/FILLB
  - utility/PKGRIDA (or PKGRIDG if use of GRIB files, not described in detail) ->
    * some FA routines.
    * utility/SC2RDG -> utility/EMPTB
    * processor communication routines parallel/DIWRGRID, parallel/DISGRID_C,
      which call parallel/SET2PE, MPL_SEND and MPL_RECV.
    * utility/INCGPF ->
      - utility/SC2RDG -> utility/EMPTB
      - utility/SC2WRG -> utility/FILLB
* pp_obs/FPMODCFU (CFU) and pp_obs/FPMODXFU (XFU) -> pp_obs/FPMODPREC ->
  - utility/SC2RDG -> utility/EMPTB
  - utility/SC2WRG -> utility/FILLB
  - parallel/SLCOMM (DM) -> processor communication routines MPL_SEND and MPL_RECV.
  - parallel/SLEXTPOL (DM)
* utility/MAXGPFV ->
  - parallel/SET2PE
  - processor communication routines parallel/OSNDGPF, parallel/ORCVGPF
    which call parallel/SET2PE, MPL_SEND and MPL_RECV;
    processor communication routines MPL_SEND, MPL_RECV and MPL_BARRIER.
  - utility/SC2RDG -> utility/EMPTB
* module/type_fprqphys.F90/ALLOCATE_FPRQPHY and module/type_fprqphys.F90/INQUIRE_FPRQPHY
* control/SCAN2H -> control/SCAN2MDM (DM) ->
  (see below arborescence under SCAN2MDM).
* setup/SUFPSC2B
* the ALADIN routine FPEZO2H (not described).
* dia/WRHFP (WRMLFPL or WRMLFP if use of GRIB files, not described in detail) ->
  (see below general architecture under WRHFP)
```

* **General architecture under DYNFPOS:** Only features concerning FULL-POS are mentioned.

```
pp_obs/DYNFPOS ->
* setup/SUFPCONF
* setup/SUVFPOS ->
* pp_obs/FPFILTER -> dia/INI3WRFP, LFI routines, MXMAOP
* pp_obs/CPVPOSPR
* setup/SUVFPOSL
* control/STEPO -> (see below arborescence under STEPO).
```

∗ **General architecture under CPREP4:** Only features concerning FULL-POS are mentioned. **CPREP4** is divided in two parts calling the same routines.

```
control/CPREP4 ->
* utility/SUALSPA
* setup/SUINIF -> (see paragraph about arborescence under SUINIF).
* control/STEPO -> (see below arborescence under STEPO).
* pp_obs/GRIDFPOS -> (see above arborescence under GRIDFPOS).
* pp_obs/DYNFPOS -> (see above arborescence under DYNFPOS).
* utility/FREEMEM -> several utility/DEAL... routines (DEALLO,
  DEALFPOS, DEALDDH, DEALCOS, DEALCTV, DEALGES, DEALCAN, DEALSPA, DEALSC2,
  DEALSCR, DEALMOD, DEALNMI, DEALTC), utility/DEELLO for ALADIN,
  (E)TRANS_END (project tfl or tal).
* control/CNT0 -> (see above arborescence under CNT0).
* FA and LFI routines.
* distributed memory routines (MPL_BROADCAST).
```

∗ **General architecture under STEPO:** Only features concerning FULL-POS are mentioned.

```
control/STEPO ->
* Gestion of read/write: utility/IOPACK (see below arborescence under IOPACK).
* Gestion of read/write for in-line FULL-POS:
  - pp_obs/PRESPFPOS ->
    * utility/RDSPEC (arborescence not detailed)
    * utility/WRSPEC (arborescence not detailed)
    * utility/CPGRIDF
    * pp_obs/SPACONVERT ->
      - adiab/SPNH_CONV_NHVAR (for non hydrostatic variables in ALADIN) ->
        * transform/ETRANSINV_NHCONV (for ALADIN) or transform/TRANSINV_NHCONV (for ARPEGE)
          (arborescence not detailed, does inverse spectral transforms)
        * adiab/GNH_CONV_NHVAR (arborescence not detailed, does grid-point calculations)
        * transform/ETRANSDIR_NHCONV (for ALADIN) or transform/TRANSDIR_NHCONV (for ARPEGE)
          (arborescence not detailed, does direct spectral transforms)
      - utility/ESPCONVERT (for ALADIN).
      - utility/SPCONVERT (for ARPEGE).
    * utility/PKSPECA for ARPEGE files (which calls FA and LFI routines,
      parallel/DIWRSPE0, parallel/DISSPEC0).
    * utility/PKSPECG for GRIB files (arborescence not detailed).
    * utility/PKGRIDA (or PKGRIDG if use of GRIB files, not described in detail) ->
      - some FA routines.
      - utility/SC2RDG -> utility/EMPTB
      - processor communication routines parallel/DIWRGRID, parallel/DISGRID_C,
        which call parallel/SET2PE, MPL_SEND and MPL_RECV.
      - utility/INCGPF ->
        * utility/SC2RDG -> utility/EMPTB
        * utility/SC2WRG -> utility/FILLB
    * utility/SUALSPA
* Inverse spectral transforms: transform/TRANSINVH -> transform/TRANSINV_FP ->
  pp_obs/SUFPVSET_INV, pp_obs_FPTSA_INV and ``TFL" routines
  (or the equivalent ALADIN routines for ALADIN).
* Grid point computations:
  control/SCAN2H -> control/SCAN2MDM (DM) ->
  (see below arborescence under SCAN2MDM).
* Lateral coupling in ALADIN (ECOUPL1).
* Direct spectral transforms: transform/TRANSDIRH -> transform/TRANSDIR_FP ->
  pp_obs/SUFPVSET_DIR, pp_obs_FPTSA_DIR and ``TFL" routines
  (or the equivalent ALADIN routines for ALADIN).
* Spectral computations: control/SPCH (see below).
* FULL-POS residual computations.
  - dia/ESPFP (for ALADIN, arborescence not described in detail)
  - setup/SUFPSC2B
  - the ALADIN routine FPEZO2H (not described).
```

∗ General architecture under IOPACK: Part 6 of **IOPACK** concerns FULL-POS.

```
utility/IOPACK ->
* dia/WRPLFP for GRIB files, not described in detail.
* dia/WRPVLFP for GRIB files, not described in detail.
* dia/WRTHLFP for GRIB files, not described in detail.
* dia/WRMLFP for GRIB files, not described in detail.
* dia/WRSFP -> (see below general architecture under WRSFP).
* dia/WRHFP (WRMLFPL or WRMLFP if use of GRIB files, not described in detail) ->
  (see below general architecture under WRHFP).
```

∗ General architecture under WRSFP:

```
dia/WRSFP ->
* dia/INI1WRFP
* setup/SUMPIOH
* dia/FPSPNORM ->
  - setup/SUMETRIC (ARPEGE) or setup/SUEMETRIC (ALADIN)
  - dia/SPNORMBL for ARPEGE (arborescence not described in detail)
  - parallel/COMMSPNORM1 for ARPEGE (arborescence not described in detail)
  - dia/SPNORMAVE for ARPEGE (arborescence not described in detail)
* parallel/TRWVTOF, which calls parallel/SET2PE and the processor
  communication routines MPL_PROBE, MPL_SEND and MPL_RECV.
* dia/INI2WRFP
* utility/PREPACKA -> some FA routines.
* parallel/GATHFLNM -> MPL_SEND and MPL_RECV.
* dia/INI3WRFP
* dia/SUPPDATE -> some FA and LFI routines.
* pp_obs/OPENFPFA -> utility/CHIEN, utility/ECHIEN (ALADIN routine), LFI end FA routines.
* utility/NEWFA which calls some FA and LFI routines.
* some FA and LFI routines.
* dia/WRHFP -> (see below the organigramme of WRHFP).
```

∗ General architecture under WRHFP:

```
dia/WRHFP (WRMLFPL or WRMLFP if use of GRIB files, not described in detail) ->
* dia/INI1WRFP
* dia/INI3WRFP
* dia/SUPPDATE -> some FA and LFI routines.
* setup/SUMPIOH
* utility/EXTGPF ->
  - utility/SC2RDG -> utility/EMPTB
  - processor communication routine parallel/DIWRGRFP, which calls MPL_SEND, MPL_PROBE and MPL_RECV.
* utility/EXTFPF ->
  - utility/SC2RDGFP
  - processor communication routine parallel/DIWRGRFP, which calls MPL_SEND, MPL_PROBE and MPL_RECV.
* the ALADIN routine EXBIP.
* dia/INI2WRFP
* the ALADIN routine EXARP.
* dia/FPGPNORM
* utility/PREPACKA -> some FA routines.
* parallel/GATHFLNM -> MPL_SEND and MPL_RECV.
* pp_obs/OPENFPFA -> utility/CHIEN, utility/ECHIEN (ALADIN routine), LFI end FA routines.
* utility/NEWFA which calls some FA and LFI routines.
* some FA and LFI routines.
```

7.2 Grid-point computations.

$*$ **General architecture of SCAN2MDM (distributed memory): control/SCAN2H $\rightarrow$ control/SCAN2MDM** (DM) $\rightarrow$:

- Some memory transfers and pointer computations before grid-point computations.
- Comparison with observations (non-lagged part, then information communication between processors, then lagged part).
- **GP_MODEL**: model grid-point computations (non-lagged part, then information communication between processors, then lagged part).
- Post-processing (grid-point part, see below for more details).
- Grid-point computations for analysis.

∗ **General architecture concerning FULL-POS in SCAN2MDM:** FULL-POS under **control/SCAN2MDM** includes:

```
* setup/SUFPSC2B
* Vertical interpolations. VPOS ->
  - utility/SC2RDG -> utility/EMPTB
  - adiab/GPPRE.
  - CTSTAR.
  - POS -> (see below).
  - PHYMFPOS ->
    * adiab/GPPRE
    * FPACHMT ->
      - Several adiab/GP... routines
      - phys_dmn/ACSOLW, phys_dmn/ACHMT and phys_dmn/ACCLPH (arborescence not detailed)
    * FPCICA -> FPCINCAPE
    * pp_obs/phymfpos.F90/STORE_DATA
  - utility/SC2WRG -> utility/FILLB
* Biperiodicization of vertically post-processed fields in ALADIN: ald/pp_obs/EBIPOS ->
  - FPBIPER
  - the processor communication routines parallel/DIWRGRID and parallel/DISGRID_C
* Diagnostic of the post-processable variables which are computed from
  the post-processed model variables (height or eta-levels
  post-processing and change of horizontal geometry).
  ENDVPOS ->
  - utility/SC2RDGFP
  - ENDPOS ->
    * Several adiab/GP... routines: GPRCP, GPTET, GPEPT, GPPRE, GPIET.
    * CTSTAR
    * FPPS ->
      - Several adiab/GP... routines: GPRCP, GPPREH, GPXYB, GPGEO.
      - PPPMER
    * PPLETA
    * c9xx/APACHE -> (see below)
    * FPACHMT ->
      - Several adiab/GP... routines
      - phys_dmn/ACSOLW, phys_dmn/ACHMT and phys_dmn/ACCLPH (arborescence not detailed)
    * FPCICA -> FPCINCAPE
    * PPLTEMP
    * FPPS
      - Several adiab/GP... routines: GPRCP, GPPREH, GPXYB, GPGEO.
      - PPPMER
    * PPWETPOINT
    * PPTHPW
    * POAERO -> TJQUD, TJCUBI and TJQUAA
    * endpos.F90/STORE_DATA
  - utility/SC2WRGFP
* Non lagged part of horizontal interpolations. HPOS ->
  - phys_dmn/ACSOLW
  - CVLANISO
  - utility/SC2RDG -> utility/EMPTB
* For distributed memory, some routines doing halo constitution and communications
  between processors before performing interpolations under HPOSLAG.
  - parallel/SLCOMM (DM) -> processor communication routines MPL_SEND and MPL_RECV.
  - parallel/SLEXTPOL
* Lagged part of horizontal interpolations and some vertical interpolations
  made after horizontal interpolations. HPOSLAG ->
  - FPOSHOR -> (see below)
```

∗ Organigramme of POS:

```
* Several adiab/GP... routines: GPRCP, GPRT, GPPREH, GPXYB, GPPREF, GPGRXYB,
  GPGRGEO, GPCTY, GPVERDIA, GPUVS, GPGW, GPGEO, GPPVO, GPEPT, GPIET, GPRH.
* CTSTAR
* FPPS
  - Several adiab/GP... routines: GPRCP, GPPREH, GPXYB, GPGEO.
  - PPPMER
* A series of routines computing intermediate quantities and weights for vertical interpolations.
  - PPLTP
  - PPLTETA
  - PPLETA -> adiab/GPPREH, adiab/GPXYB and adiab/GPPREF
  - PPINIT
  - PPFLEV
  - PPSTA
* A series of interpolation routines.
  - PPUV ->
    * PPUV_OLD -> PPINTP and PPITPQ
    * PPINTP
    * (PPITPQ) (currently commented).
  - PPT ->
    * PPT_OLD -> PPINTP
    * PPSTA
    * PPINTP
  - PPQ -> PPINTP
  - c9xx/APACHE -> (see below)
  - PPGEOP ->
    * PPGEOP_OLD -> PPSTA, adiab/GPGEO, PPITPQ and PPINTP.
    * PPSTA
    * adiab/GPGEO
  - PP2DINT
  - PPRH -> adiab/GPRH and PPINTP
  - adiab/GPRH
  - PPVVEL -> adiab/GPCTY and PPINTP
  - adiab/GPCTY
  - PPPMER
  - PPLTP
  - PPTHPW
  - POAERO -> TJQUD, TJCUBI and TJQUAA
  - PPLTEMP
  - FPPS
  - PPWETPOINT
* Storage of post-processed fields.
```

∗ Organigramme of FPOSHOR:

```
* utility/SC2RDGFP
* FPSCAW
* FPINTDYN -> FPINT12 and FPINT4.
* FPCORDYN -> FPHOR12.
* FPGEO
* FPSAMPL -> FPINT12 and FPINT4.
* FPCLIPHY -> CVLANISO.
* FPNILPHY -> CVLANISO.
* FPINTPHY -> FPINT12 and FPINT4.
* FPCORPHY -> sur/external/SURF_INQ, FPLAKE, FPHOR12, phys_dmn/ACSOLW, FPLAKE.
* utility/SC2WRGFP
```

∗ Organigramme of c9xx/APACHE:

```
* Several adiab/GP... routines: GPPREH, GPXYB, GPPREF, GPRCP and GPPRE
* PPINIT
* FPPS ->
  - Several adiab/GP... routines: GPRCP, GPPREH, GPXYB, GPGEO.
  - PPPMER
* PPLETA
  - Several adiab/GP... routines
* FPVIEW
* c9xx/AVAL ->
  - PPFLEV
  - PPSTA
  - PPT ->
    * PPT_OLD -> PPINTP
    * PPSTA
    * PPINTP
  - PPQ -> PPINTP
  - PPRH -> adiab/GPRH and PPINTP
  - PPUV ->
    * PPUV_OLD -> PPINTP and PPITPQ
    * PPINTP
    * (PPITPQ) (currently commented).
  - PPGEOP ->
    * PPGEOP_OLD -> PPSTA, adiab/GPGEO, PPITPQ and PPINTP.
    * PPSTA
    * adiab/GPGEO
  - PP2DINT
  - Several adiab/GP... routines: GPPREH, GPXYB, GPPREF and GPRCP
  - c9xx/BOB -> PPINTP
```

7.3 Spectral transforms.

See documentation (IDTS) about spectral transforms of model fields.

7.4 Spectral computations.

`control/STEPO -> control/SPCH -> control/SPCM -> pp_obs/SPOS -> utility/MXMAOP.`

7.5 Action and brief description of each routine.

- Expression "full level" is synonym of "middle of layer".
- Expression "half level" is synonym of "interlayer".
- For meaning of [L5] see section "Sequences of calls of post-processing".
- Expression "DM-local" for a quantity means "local to the couple of processors (*proca*,*procb*)": each processor has its own value for the quantity. Expression "DM-local computations" means that the computations are made independently in each processor on "DM-local" quantities, leading to results internal to each processor, which can be different from a processor to another one.
- Expression "DM-global" for a quantity means that it has a unic value available in all the processors. Expression "DM-global computations" means that the computations are either made in one processor, then the results are dispatched in all the processors, or the same computations are made in all the processors, leading to the same results in all the processors.
- In a routine description the mention "For distributed memory computations are DM-local" means that all calculations made by this routine are DM-local; the mention "For distributed memory computations are DM-global" means that all calculations made by this routine are DM-global; when no information is provided it means that a part of calculations are DM-local and the other part is DM-global.
- Expression "main" processor currently refers to the processor number 1: (*proca*,*procb*)=(1,1).

∗ Grid-point and spectral routines of directory "adiab":

- **GP...** and **GNH...** routines computing some dynamical grid-point quantities; for distributed memory computations are DM-local. For more details about what do these routines, see section (24) of documentation (IDEUL).
- **SPECRTDM**: spectral space conversions between real temperature and virtual temperature for distributed memory.

∗ Routines of directory "c9xx".

- **APACHE**: interface routine for some vertical interpolations, called if post-processing on height or η-levels and change of horizontal geometry. For distributed memory computations are DM-local.
- **AVAL**: inner interface routine for some vertical interpolations. For distributed memory computations are DM-local.
- **BOB**: vertical interpolation of the B-coefficients of the vertical hybrid coordinate, to the levels of post-processing. For distributed memory computations are DM-local.
- **RDCLIMO** (**RDECCLIMO** at ECMWF if GRIB files used): opens output climatology files and reads the needed fields. In case of a file ARPEGE, poles values are not read. For distributed memory computations are DM-global.

∗ Control routines of directory "control":

- **CNT0**: controls integration job at level 0.
- **CNT1**: controls integration job at level 1.
- **CNT2**: controls integration job at level 2.
- **CNT3**: controls integration job at level 3.
- **CNT4**: controls integration job at level 4.

- **CPREP4**: control routine called only if post-processing needs a change of horizontal geometry and if spectral fit or computations are made in the arrival geometry: pseudo configurations 927 (for example file ARPEGE → file ARPEGE), E927 (for example file ALADIN → file ALADIN). Calls twice **CNT0** after other routines, the first call of **CNT0** to manage post-processing computations in the arrival geometry and the second call to **CNT0** if required to return to the departure (model) geometry.
- **SCAN2H**: control routine for grid-point computations.
- **SCAN2MDM**: interface for grid-point computations for distributed memory calculations.
- **SPCH**: control routine for spectral computations.
- **SPCM**: interface for spectral computations.
- **STEPO**: control routine for one time integration step.

∗ Routines of directory "dia".

- **ESPFP**: ALADIN routine writing vertical post-processed data on a FA file, equivalent of **WRSFP**.
- **FPGPNORM**: writes out norms on the horizontally post-processed fields (including mean value and extrema) for grid-point fields.
- **FPSPNORM**: writes out norms on the horizontally post-processed fields (including mean value and extrema) for spectral fields.
- **INI1WRFP**: preparation to write out the horizontally post-processed fields to ARPEGE/ALADIN files: computes variables which do not depend of the fields themselves, prior to writing out on post-processed files.
- **INI2WRFP**: preparation to write out the horizontally post-processed fields to ARPEGE/ALADIN files: computes variables which depend of the fields themselves, prior to writing out on post-processed files.
- **INI3WRFP**: preparation to open a file and write out the horizontally post-processed fields to ARPEGE/ALADIN files: computes variables needed before opening files for post-processing purpose.
- **SPNORMAVE**: computes the average of norms in spectral space for each level and for all levels, uses data computed in **SPNORMBL**. For distributed memory computations are DM-global.
- **SPNORMBL**: computes norms in spectral space (interface routine). For distributed memory this routine is not parallelised and can do computations for all data on one processor only, so data have to be previously collected on one processor.
- **SUPPDATE**: initializes the date to write out on an output file (historical file or post-processing file).
- **WRHFP**: writes out the horizontally post-processed fields, sorts out the output points for each domain, then writes records on files. Only grid-point data are written.
- **WRMLFP**: GRIB routine used instead of **WRHFPG** in some cases (especially in FULL-POS configurations other than 927-type ones). Writes data on model layers.
- **WRPLFP**: GRIB routine used instead of **WRHFPG** in some cases (especially in FULL-POS configurations other than 927-type ones). Writes data on pressure layers.
- **WRPVLFP**: GRIB routine. Writes data on potential vorticity layers.
- **WRSFP**: writes out the vertically post-processed dynamical fields to ARPEGE/ALADIN file. Spectral fields are written out as spectral coefficients (if spectral output is wanted), grid-point fields are written out as grid-point fields. The horizontal grid is the model one.
- **WRTHLFP**: GRIB routine. Writes data on potential temperature layers.

∗ Distributed memory environment routines (directory "parallel"): For the "parallel" routines present in the organigramme, see documentation (IDDM) about distributed memory features.

∗ Routines of directory "phys_dmn".

- **ACCLPH**: interpolation of some variables (gz, U, V) at the top of the planetary boundary layer.
- **ACHMT**: calculation of some surface characteristics; interpolation of some variables (U, V, T, q) at some chosen heights.
- **ACSOLW**: computation of some surface coefficients and variables.

* Routines of directory "pp_obs".

- **CPCLIMI**: fills interpolation buffers with some surface fields (land-sea mask) previous to compute weights with land-sea mask.
- **CPCLIMO**: computes output climatology from input one (land-sea mask): extracts fields from model buffer/work file or array, then performs horizontal interpolations or sampling according to the switch **LFPOSHOR**.
- **CPVPOSPR**: computes field pointers for vertical post-processing. Computes the pointers of the post-processing fields so that the fields are stored one after the others in the arrays GT0 (fitted fields) and GAUX (unfitted fields). The fields are stored in the following order: 1) underived 3D fields; 2) underived 2D fields; 3) derived 3D fields; 4) derived 2D fields.
- **CTSTAR**: computes the standard surface temperature and the surface temperature to be used for extrapolations of temperature and geopotential height. For distributed memory computations are DM-local.
- **CVLANISO**: makes a transformation for surface orography in order to have the horizontal gradient (departure data are standard deviation, anisotropy coefficient and direction of the principal axis).
- **DYNFPOS**: interface routine managing vertical then horizontal interpolations for post-processing on dynamical 3D and 2D fields, calls several sequences of **STEPO**.
- **ENDPOS**: finish post-processing calculations for output height levels or η-levels. For distributed memory computations are DM-local.
- **ENDVPOS**: interface routine managing all post-processing computations which can be done only after horizontal interpolations. Called if dynamical variables other than model variables are post-processed on height or η-levels. For distributed memory computations are DM-local.
- **EBIPOS**: bi-periodicization of vertically post-processed fields in ALADIN.
- **FPACHMT**: FULL-POS interface to ACHMT, to compute pressure, humidity, temperature and wind.
- **FPBIPER**: bi-periodicize the post-processed fields, or just fill the extension zone with the mean value of C+I area.
- **FPCICA**: routine computing CAPE (convective available potential energy) and CIN (convective inhibition): interface to **FPCINCAPE**.
- **FPCINCAPE**: routine computing CAPE (convective available potential energy) and CIN (convective inhibition).
- **FPCLIPHY**: writes the output climatology in the post-processing buffers and initializes a logical climatological mask. For distributed memory computations are DM-local.
- **FPCORDYN**: corrects dynamical fields after the horizontal interpolations which may create overshoots, or to recover the requested fields. For distributed memory computations are DM-local.
- **FPCORPHY**: corrects physical fields after the horizontal interpolations which may create overshoots, or to recover the requested fields. For distributed memory computations are DM-local.
- **FPFILTER**: set-up routine which computes a matricial operator which is a product of a transformation from computational sphere to "equivalent" not dilated sphere, a "THX"-filtering and the inverse transformation from "equivalent" not dilated sphere to computational sphere. The operator is stored in the array **RFPMAT**. Sphere to sphere transformations are made by reading a LFI file containing dilatation/contraction matrixes which can be initialised by a configuration 911 (see the ad-hoc documentation (IDRD)). **FPFILTER** is called only in the following cases:
  - filtering in spectral space activated, **LFPFIL**=.TRUE. (if ARPEGE) or **LFPBED**=.TRUE. (if ALADIN) in **NAMFPF**.
  - the geometry where the spectral fit is done is a stretched one: for example the initial geometry if not 927-type FULL-POS configurations, and **CFPFMT** is not 'MODEL'.
  - post-processing for any variable if potential vorticity vertical coordinate.
  - post-processing of horizontal derivatives (vorticity, divergence, vertical velocity, deformations) for other vertical coordinates.

  For distributed memory computations are DM-local: **RFPMAT** has a DM-global dimension but only the zonal wavenumbers useful for the current processor are computed.

- **FPGEO**: transforms the post-processed fields into the output geometry. For distributed memory computations are DM-local.
- **FPHOR12**: Corrections after 12-points horizontal interpolations, for example to prevent overshoots. Corrections can be done according to the values of the dummy argument **KORR**. See comments in **FPHOR12** for the different options available for **KORR**. For distributed memory computations are DM-local.
- **FPINIPHY**: controls that the pointers of physical fields to be post-processed are defined.
- **FPINTDYN**: interface for 12-points or 4-points horizontal interpolations, for fields coming from the dynamics. For distributed memory computations are DM-local.
- **FPINTPHY**: interface for 12-points or 4-points horizontal interpolations, for fields coming from the physics.

  - Land-sea mask is taken into account for the surface variables other than surface orography.
  - A monotonic correction is done for precipitations.
  - For minima and maxima, interpolations are replaced by the research of the nearest point.

  For distributed memory computations are DM-local.
- **FPINT4**: 4 points horizontal interpolations. If **LDNRST** is .TRUE., interpolations are replaced by the research of the nearest point. For distributed memory computations are DM-local.
- **FPINT12**: 12 points horizontal interpolations. If **LDNRST** is .TRUE., interpolations are replaced by the research of the nearest point. If **LDMONO** is .TRUE., a monotonic correction is done. For distributed memory computations are DM-local.
- **FPLAKE**: corrector for lakes and islands: overwrite with climatologic data. For distributed memory computations are DM-local.
- **FPMODCFU**: preliminary modification of cumulated fluxes for post-processing: prepares input fields before horizontal interpolations.
- **FPMODPREC**: to modify precipitation fields before horizontal post-processing: to avoid enlargement of the precipitation areas, input fields are transformed so that there are negative values at the points with no precipitations. Output fields will be filtered after horizontal interpolations. This negative value is computed as the opposite of the maximum value of the nearest points: 4 points if the grid is regular; 5 or 6 points is the grid is reduced (in this case one takes the linearly interpolated value for the external latitudes). For distributed memory computations are DM-local. The way to modify values is coded like horizontal interpolations, but in this particular case the interpolation points are model grid-points, so weights computations are useless, one only needs to know the grid-point coordinates of the surrounding points.
- **FPMODXFU**: preliminary modification of instantaneous fluxes for post-processing: prepares input fields before horizontal interpolations.
- **FPNILPHY**: computes the physical fields which cannot be directly interpolated but need auxiliary computation or need to be set-up by a straight value (this is the case for some ISBA fields); generates an abort for some fields which cannot be treated by FULL-POS.
  The following fields have a special treatment:

  - Resistance to evapotranspiration: reset to 1. .
  - Interception content: reset to 0. .
  - Anisotropy coefficient and direction of main axis of orography: transformed into a vector via routine **CVLANISO** before interpolation.

  The following fields cannot be treated (abort):

  - Index of vegetation.
  - Stomatal minimum resistance.
  - Percentage of clay within soil.
  - Percentage of sand within soil.
  - Soil depth.

  - Leaf area index.
  - (Roughness length of bare surface)$*g$.
  - Surface albedo for non snowed areas.
  - (Roughness length for heat)$*g$.

  For distributed memory computations are DM-local.

- **FPOSHOR**: inner interface routine managing lagged part of horizontal interpolations. For distributed memory computations are DM-local.
- **FPPS**: computes pressures corresponding to heights levels or η-levels of post-processing. For distributed memory computations are DM-local.
- **FPSAMPL**: FULL-POS horizontal sampling; to sample input buffer (when the output grid is a subset of the input grid or the same grid); does a simple memory transfer or calls the 12-points or 4-points horizontal interpolations with option "search of the nearest point" according to the dummy variable **KW**. For distributed memory computations are DM-local.
- **FPSCAW**: for horizontal interpolations of FULL-POS, finds the indices in interpolation buffers where information of the surrounding model grid points can be found. For distributed memory computations are DM-local.
- **GPOS**: saves in grid-point arrays the spectral fields that will be needed for post-processing on height levels above an outside orography. For distributed memory computations are DM-local.
- **GRIDFPOS**: interface routine managing post-processing of grid point fields. Performs horizontal post-processing on physical fields and fluxes and writes post-processed fields on a file.
- **HPOS**: interface routine managing non lagged part of horizontal interpolations. Fills buffers for quantities to be interpolated (arrays **PFPBUF1**). Treats: 1) fitted fields coming from vertical interpolations, 2) unfitted fields coming from vertical interpolations, 3) physical fields and auxiliary surface fields, 4) cumulated fluxes, 5) instantaneous fields. A memory transfer by calling routine **SC2RDG** is generally necessary to get quantities for series 2 to 5. For distributed memory computations are DM-local.
- **HPOSLAG**: interface routine managing lagged part of horizontal interpolations and some vertical interpolations made after horizontal interpolations. Works like the sequence **adiab/CALL_SL** $\rightarrow$ **adiab/LAPINEA + LAPINEB** $\rightarrow$ (subroutines) for semi-Lagrangian scheme, but code is simpler because there is no trajectory computation: points where quantities have to be interpolated are already known, so coordinates and weights for interpolations can be precomputed in a set-up routine. First surface fields are interpolated (if post-processing on height or η-levels), then dynamical fields and physical fields. If post-processing is made on height or η-levels there are additional vertical interpolations to take account of the difference of orography between the departure geometry and the arrival geometry. For distributed memory computations are DM-local.
- **OPENFPFA**: opens an ARPEGE/ALADIN existing file of post-processing format, and controls the framework (namelist vs file frame).
- **PHYMFPOS**: physico-dynamic post-processing interface for Meteo-France.
- **POAERO**: post-processing of jet and tropopause heights. For distributed memory computations are DM-local.
- **POS**: inner interface for vertical post-processing. For distributed memory computations are DM-local.
- **PPFLEV**: finds full or half levels under specified pressures. Called if post-processing on pressure, potential vorticity or potential temperature levels ([L5]='B', 'V' or 'T'). For distributed memory computations are DM-local.
- **PPGEOP**: vertical interpolations on pressure levels for geopotential height. For distributed memory computations are DM-local.
- **PPGEOP_OLD**: old version of routine **PPGEOP** used at ECMWF. For distributed memory computations are DM-local.
- **PPINIT**: computes intermediate quantities connected with full levels and half levels pressures, prior to vertical interpolations on pressure or height levels. Called in all cases except the case where post-processing is done on model η-levels ([L5]='M' or 'S'). For distributed memory computations are DM-local.

- **PPINTP**: vertical linear interpolation on pressure levels. For distributed memory computations are DM-local.
- **PPITPQ**: vertical quadratic 3 points interpolation. For distributed memory computations are DM-local.
- **PPLETA**: computes the pressures of post-processing on η-levels. Called if post-processing on η-levels ([L5]='S'). For distributed memory computations are DM-local.
- **PPLTEMP**: computes the pressure value for a given temperature in a vertical profile.
- **PPLTETA**: computes the pressures of post-processing on θ-levels. Called if post-processing on potential temperature levels ([L5]='T'). For distributed memory computations are DM-local.
- **PPLTP**: computes the pressures of post-processing on potential vorticity levels. Called if post-processing on potential vorticity levels ([L5]='V'). For distributed memory computations are DM-local.
- **PPPMER**: computes mean sea level pressure. For distributed memory computations are DM-local.
- **PPQ**: vertical interpolations on pressure levels for humidity, liquid water, ice, cloudiness, extra GFL variables. For distributed memory computations are DM-local.
- **PPREQ**: search for a suitable post-processing namelist file and reads it.
- **PPRESET**: reset the setup of FULL-POS (currently used at ECMWF only to change the FULL-POS options at each post-processing timestep).
- **PPRH**: vertical interpolations on pressure levels or height levels for relative humidity. For distributed memory computations are DM-local.
- **PPSTA**: integrates standard atmosphere for geopotential height. For distributed memory computations are DM-local.
- **PPT**: vertical interpolations on pressure levels for temperature. For distributed memory computations are DM-local.
- **PPT_OLD**: old version of routine **PPT** used at ECMWF. For distributed memory computations are DM-local.
- **PPTHPW**: computation of the moist irreversible adiabatic potential temperature, vertical interpolation on pressure or height levels. For distributed memory computations are DM-local.
- **PPUV**: vertical interpolations on pressure levels for wind components. For distributed memory computations are DM-local.
- **PPUV_OLD**: old version of routine **PPUV** used at ECMWF. For distributed memory computations are DM-local.
- **PPVVEL**: vertical interpolations on pressure levels or height levels for variables linked to vertical velocity. For distributed memory computations are DM-local.
- **PPWETPOINT**: computation of the wetpoint temperature.
- **PP2DINT**: vertical linear interpolation (on pressure levels or height levels): used for potential vorticity, potential temperature, equivalent potential temperature, divergence, vorticity, stretching deformation, shearing deformation, velocity potential, stream function, absolute vorticity, wet potential vorticity. For distributed memory computations are DM-local.
- **PREGPFPOS**: saves model grid-point data and pack current data before go post-processing.
- **PRESPFPOS**: to save and possibly pack the data before post-processing. If in-line vertical post-processing: pack spectral arrays, save and pack gridpoint upperair fields, save gridpoint surface fields and pack surface temperature. After post-processing: reset gridpoint orography, recover gridpoint upperair fields, recover gridpoint surface fields (for surface temperature).
- **SPACONVERT**: to convert model spectral fields into the fields actually written on file, and vice-versa.
- **SPECFITADM**: spectral transformation of model fields (except wind components) and surface orography read on a grid-point field, computation of spectral velocity potential and stream function on model η-levels, writing of all these fields on a file. This routine is called in the second part of a 927-type FULL-POS configuration.
- **SPECFITG**: GRIB version of **SPECFITADM**.
- **SPOS**: filters some spectral quantities in model spectral space.

- **SUFPMOVE**: ALADIN routine to define the translation of the fields to apply during post-processing; the translation is defined as the trajectory from the modelized to the observed cyclone.
- **TJCUBI**: post-processing of jet and tropopause: finds coefficients of a cubic expression. For distributed memory computations are DM-local.
- **TJQUAA**: post-processing of jet and tropopause: performs the least square fit by a quadratic function. For distributed memory computations are DM-local.
- **TJQUD**: post-processing of jet and tropopause: finds coefficients of a quadratic expression. For distributed memory computations are DM-local.
- **VPOS**: interface routine managing all vertical interpolations which can be done before horizontal interpolations. **VPOS** does one call to **POS** for "basic" variables, and special separate calls to **POS** for variables linked with the wind field, such as divergence, (relative) vorticity, absolute vorticity, velocity potential and stream function. For distributed memory computations are DM-local.

∗ Set-up routines of directory "setup":

- **DFLTVV1**: initialises default values of functions A and B of hybrid vertical coordinate of the arrival geometry of FULL-POS. For distributed memory computations are DM-global.
- **SUAFN**: initialises ARPEGE field descriptors. Reads namelist **NAMAFN**, initialises **YOMAFN**. For distributed memory computations are DM-global.
- **SUAFN1**: initialises ARPEGE field descriptors: part of calculations done before the namelist reading. For distributed memory computations are DM-global.
- **SUAFN2**: initialises ARPEGE field descriptors: part of calculations done after the namelist reading. For distributed memory computations are DM-global.
- **SUAFN3**: prints ARPEGE field descriptors. For distributed memory computations are DM-global.
- **SUALFPOS**: memory allocation for FULL-POS arrays.
- **SUALLO**: memory allocation: general interface routine.
- **SUALMP2**: allocation of arrays used for distributed memory; includes quantities used in the distributed memory version of FULL-POS.
- **SUBFPOS**: interface routine, calls some other setup routines.
- **SUCACLIADM**: interface for reading the climatologic grid point fields for CANARI (ARPEGE files): distributed memory code.
- **SUCFU**: initialises the control of cumulated fluxes.
- **SUCFUFP**: initialises cumulated fluxes switches for FULL-POS. For distributed memory computations are DM-global.
- **SUCT0**: routine to initialize level 0 control module.
- **SUDIM2**: initialises dimensioning quantities, reads namelist **NAMDIM**, initialises **YOMDIM**.
- **SUFPC**: initialises FULL-POS post-processing scientific and technical options, reads namelist **NAMFPC**, initialises **YOMFPC**. For distributed memory computations are DM-global.
- **SUFPCONF**: initialise FULL-POS configuration letters for **STEPO**. Initialise for each scan the configuration letters for vertical post-processing, horizontal post-processing and for **IOPACK**. For distributed memory computations are DM-global.
- **SUFPD**: initialise FULL-POS horizontal subdomains. Computes total number of output points. Reads namelist **NAMFPD**, initialises **YOMFPD**. For distributed memory computations are DM-global.
- **SUFPDIM**: initialise dimensioning quantities for FULL-POS. Computes number of derived/underived fields, number of fitted/unfitted fields, number of scalar/vector fields as if for a given post-processing time step, all the fields were requested at all levels for all subdomains. For distributed memory computations are DM-global.
- **SUFPDOM**: initialise list of FULL-POS post-processing subdomains. To read the list of subdomains as a character string and convert it into an integer array. In the string, the subdomain names are separated with a ":". If the string is blank, all the subdomains are required. For distributed memory computations are DM-global.

- **SUFPDYN**: initialise dynamical requests of post-processing at a given time-step. In this routine, we consider that, for a given level type, all the fields are post-processed at all levels and for all subdomains. Reads namelists containing local variables (**NAMFPDY2**, **NAMFPDYP**, **NAMFPDYH**, **NAMFPDYV**, **NAMFPDYT** and **NAMFPDYS**).
- **SUFPF**: initialises the profile of the spectral post-processing filter. Reads namelist **NAMFPF** and initialises **YOMFPF**.
- **SUFPG1**: initialise output geometry: sets default values, then reads namelist **NAMFPG** and checks it. For distributed memory computations are DM-global.
- **SUFPG2**: initialise output geometry: computes DM-global quantities containing the coordinates of the output points in the model geometry, the output mapping factor, and the relative rotation matrix. When an output point is at a geographical pole, the equation of the compass is degenerating, so a direct (and exact) computation is needed. For distributed memory computations are DM-global.
- **SUFPG**: initialise output geometry: calls **SUFPG1** and **SUFPG2**, makes some additional tests then computes the DM-local versions of the DM-global quantities computed in **SUFPG2**. Initialises **YOMFPG** and **YOMFPGIND**.
- **SUFPIOS**: set-up for using work files on post-processing buffers. To set-up **YOMFPIOS**. Opens work files.
- **SUFPOPH**: initialise post-processing file-handling parameters (**YOMFPOP**). For distributed memory computations are DM-global.
- **SUFPPHY**: initialise the requests of post-processing at a given time-step. In this routine, we consider that, for a given level type, all the fields are post-processed at all levels and for all subdomains. Reads namelist containing local variables (**NAMFPPHY**).
- **SUFPSC2**: reads namelist **NAMFPSC2** and sets-up **YOMFPSC2** which contains the control parameters and arrays to scan the post-processing buffers per row.
- **SUFPSC2B**: sets up control arrays of type **NSTA(X)FPB** of **YOMRFPB**, **YOMWFPB**, **YOMDFPB**, **YOMPFPB**, **YOMAFPB**, **YOMFPEZO**.
- **SUFPTR2**: computes the number of fields **NFPVT0** and the associated pointers for the auxiliary grid-point quantities.
- **SUFPWIDE**: initialises the control variables depending on the variable **NFPWIDE**, which are required in routine **SCAN2MDM**: interpolation buffers quantities, some distributed memory quantities.
- **SUFRAME** (or **SUEFRAME** if plane geometry, or **SUFPGRIB** if GRIB files): sets-up the frame of ARPEGE file, in the current case for the FULL-POS files. For distributed memory computations are DM-global.
- **SUGAW**: initialises the Gaussian latitudes and weights for the geometry of FULL-POS grid. Called if the FULL-POS grid is a global Gaussian one. For distributed memory computations are DM-global.
- **SUGENORD**: computes the rotation matrix to find the geographical north, from grid-points of a computational transformed sphere; this matrix is necessary to convert vectors which are initially given in the repere of the model (transformed) geometry (for example after the horizontal interpolations) towards the geographical repere. For distributed memory computations are DM-global.
- **SUGPPRP**: initialises relative pointers for physics arrays. For distributed memory computations are DM-global.
- **SUGRCFU**: reads the cumulated fluxes on ARPEGE files.
- **SUGRCLIA**: interface for reading the climatologic grid point fields (ARPEGE files).
- **SUGRIDADM**: interface for reading the surface grid point fields of the departure files (ARPEGE files): distributed memory code.
- **SUGRIDG**: interface for reading the surface grid point fields of the departure files (GRIB files).
- **SUGRIDVADM**: interface for reading some grid point fields (ARPEGE files): distributed memory code.
- **SUGRXFU**: reads the instantaneous fluxes on ARPEGE files.
- **SUGRIDF**: interface for reading the surface grid point fields of the departure files (ARPEGE files). For distributed memory computations are DM-global.

- **SUGRIDU**: interface for reading the upper air grid point fields of the departure files. For distributed memory computations are DM-global.
- **SUHOW1** (**SUEHOW1** for ALADIN): initialises the coordinates of the nearest model grid point of the FULL-POS interpolation points: the general rule is to take the grid-point which is immediately at the north-west (relative to the transformed geometry) of the interpolation point, except in the case where the north-western point is the pole, in this case one takes the south-western model grid-point. Additional coordinates are computed. For distributed memory computations are DM-local.
- **SUHOW2** (**SUEHOW2** for ALADIN): computes weights without land-sea mask and surface temperature for bilinear or 12 points horizontal interpolations used in FULL-POS. For distributed memory computations are DM-local.
- **SUHOWLSM** (**SUEHOWLSM** for ALADIN): computes weights for bilinear or 12 points horizontal interpolations used in FULL-POS. Possibility to use the land/sea/ice mask or the surface temperature. When the option **NFPLAKE**=-1 is activated and when the four points surrounding the interpolation point are on the earth (resp. in a lake) and the interpolation point is in a lake (resp. on the earth) there is creation of a lake (resp. an island). For distributed memory computations are DM-local.
- **SUINIF**: interface routine for reading the departure files.
- **SUMETRIC**: computes metrics for spectral norms computation.
- **SUMP**: initialises quantities and arrays involved in distributed memory.
- **SUMPFPOS**: initialises some FULL-POS quantities and arrays involved in distributed memory.
- **SUMPIOH**: To setup in a contiguous way the distribution of fields when file reading/writing among processors.
- **SUNMEN**: initialises array **NFPMEN** which contains for each Gaussian latitude of the FULL-POS grid the highest zonal wave number represented. Called if the FULL-POS grid is a global Gaussian one.
- **SUOROG**: interface for reading the grid-point orography fields of the departure files.
- **SUPROCFP**: initialisation of the array **NUMPROCFP** (gives the number of the processor which makes calculations for each FULL-POS interpolation point).
- **SUPROCGP**: initialisation of a local array **IPROCGP** (gives the number of the processor which makes calculations for each model grid point).
- **SURFPBUF**: initialise buffer **RFPBUF** of **YOMRFPB**, which contains the output climatology and geometry.
  - If variable **NFPCLI** is 1,2 or 3 the output geometry climatology is read on a file by routine **RDCLIMO** or **RDECCLIMO** according to use of ARPEGE or GRIB files.
  - Output geometry climatology is stored in the array **ZFIELD** for mapping factor, local repere (north direction).
  - These data are stored in the buffer array **RFPBUF**.
  - Remark: for distributed memory, all computations are DM-local (each processor treats its interpolation points) but diagnostic prints and reading climatology on a file is made on the "main" processor (or the **NSTRIN** first processors when possible, i.e. when **LFPEZO**=.F. and **NFPCLI**=2 or 3) for all the grid-points, so processor communication is necessary after reading climatologies to dispatch data in all the processors.
- **SURFPDS**: initialises the descriptors of the auxiliary fields needed for horizontal post-processing, initialises **YOMRFPDS**. For distributed memory computations are DM-global.
- **SUSC2A** and **SUSC2B**: initialises some control variables required in routine **SCAN2MDM** including variables depending on **NSLWIDE**: interpolation buffers quantities, some distributed memory quantities.
- **SUSPEC**: interface for reading the spectral fields of the departure files (ARPEGE files).
- **SUSPECA**: interface for reading the spectral fields of the departure files (ARPEGE files).
- **SUSPECB**: interface for reading the spectral fields of the departure files for 2D model.
- **SUSPECG**: interface for reading the spectral fields of the departure files (GRIB files).
- **SUSPECG1**: initializes the spectral fields using artificial data (GRIB files).

- **SUSPECTCFOU**: interface for reading the Fourier coefficients fields of the transmission coefficients files (LFI files).
- **SUVFPOS**: initialises and sorts the list of the dynamical fields to post-process for a given level type of post-processing. Fields are stored as follows: 1) fitted underivated 3D fields; 2) fitted derivated 3D fields; 3) fitted underivated 2D fields; 4) fitted derivated 2D fields; 5) unfitted 3D fields; 6) unfitted 2D fields.
- **SUVFPOSL**: initialises **YOMFP4L** which contains the working variables and arrays to write out the dynamical post-processed fields.
- **SUVPOS**: initialises **YOMVPOS** which contains control variables for vertical post-processing.
- **SUWFPBUF**: initialises buffer **WFPBUF** of **YOMWFPB**, which contains the working fields for horizontal interpolations: size of the box, indexes and weights. Reads or computes input/output land-sea mask, computes weights and index, then fills buffer.
  - Fills interpolation buffers containing the model grid-point land-sea mask and surface temperature (currently surface temperature is not used in weight computations, so the array is filled with -999) in routine **CPCLIMI**.
  - Fills the output geometry land-sea mask and surface temperature (currently surface temperature is not used in weight computations, so the array is filled with -999): such quantities can be read on a file if **NFPCLI**=1,2 or 3 by routines **RDCLIMO** or **RDECCLIMO** or interpolated if **NFPCLI**=0 by routine **CPCLIMO**.
  - Computes weights for horizontal interpolations: a set of weights without land-sea mask and a set to weights with land-sea mask.
  - Stores the weights in the buffer **WFPBUF**. Some additional quantities can be stored: model Gaussian latitude index immediately at the north of the interpolation point (lowerly bounded by 1), longitudes indices of the points denoted by B_0, B_1, B_2 and B_3 on figure 3.2, number of the FULL-POS subdomain to which belongs each interpolation point.
  - Remark: for distributed memory, all computations are DM-local (each processor treats its interpolation points) but reading climatology on a file is made on the "main" processor for all the grid-points, so processor communication is necessary after reading climatologies to dispatch data in all the processors.
- **SUWFPDS**: initialises the FULL-POS working fields descriptors needed for horizontal post-processing, initialises **YOMWFPDS**. For distributed memory computations are DM-global.
- **SUXFU**: initialises the control of instantaneous fluxes.
- **SUXFUFP**: initialises instantaneous fluxes switches for FULL-POS. For distributed memory computations are DM-global.
- **SU0YOMA**: 0-level interface routine for set-up: first part.
- **SU0YOMB**: 0-level interface routine for set-up: second part.
- **SU1YOM**: 1-level interface routine for set-up.
- **SU3FPOS**: 3-level interface routine for set-up of FULL-POS features. Initialises buffers **RFPBUF** and **WFPBUF**.
- **SU4FPOS**: 4-level interface routine for set-up of FULL-POS features. Initialises blocks **YOM4FPOS** and **PTRFP4**, which contain the requests for post-processing at a given time-step. The grid-point fields are stored in the following order: physics, then cumulated fluxes CFU, then instantaneous fluxes XFU.

∗ **Routines of directory "transform".** For more details see documentation (IDTS) about spectral transforms; only specific FULL-POS routines are here described.

- **TRANSDIRH**: interface routine for direct spectral transforms.
- **TRANSINVH**: interface routine for inverse spectral transforms.
- **REESPE**: spectral transformation from grid point space to spectral space.
- **SPEREE**: spectral transformation from spectral space to grid-point space.
- **TRACARE**: computes geographical coordinates, once given the computational sphere coordinates.

- **TRARECA**: computes the computational sphere coordinates, once given the geographical coordinates.
- **UPDVPOS**: updates **YOMVPOS** which contains control variables for vertical post-processing.
- **UVSPE**: spectral transformation from grid point space to spectral space: takes the grid-point wind components, transform them into the spectral divergence/vorticity or velocity potential/stream function.

* Routines of directory "utility".

- **CHIEN** (or **CCHIEN** if ALADIN): It controls consistency between defined geometry and ARPEGE file. In the case of inconsistency it calls **ABOR1**. For distributed memory computations are DM-global.
- **CPGRIDF**: copies a grid point buffer/work file on another buffer/work file. For distributed memory computations are DM-local.
- **DEALFPOS**, **DEALDDH**, **DEALCOS**, **DEALCTV**, **DEALGES**, **DEALCAN**, **DEALSPA**, **DEALSC2**, **DEALSCR**, **DEALMOD**, **DEALNMI**, **DEALLO**: deallocate space for global variables. Respectively for:
  - Full-POS variables.
  - DDH variables.
  - Cost function variables.
  - Control variables (arrays depending on the size of the control variable).
  - Arrays needed for first guess constraint and for CANARI OI.
  - Analysis grid and diagnostic fields.
  - Spectral arrays.
  - Model grid-point buffers and other arrays allocated in **SUSC2A** and **SUSC2B**.
  - Screening of observations.
  - Model fields for screening.
  - Fields used in the NMI.
  - Other arrays.

  Remark: **DEALCAN** is provisionally in directory **canari** and has to be moved later in directory **utility**.
- **EGGDIR**: see **SUGENORD** but used for ALADIN: the departure geometry is the ALADIN one. For distributed memory computations are DM-global.
- **EGGRVS**: geography of grid-points, inversion from grid to geographical sphere ARPEGE-ALADIN. For distributed memory computations are DM-global.
- **EGGX**: geography of grid-points ARPEGE-ALADIN. For distributed memory computations are DM-global.
- **EMPTB**: memory transfer routine; calls **EXPAND21** if unpacking is required. For distributed memory computations are DM-local.
- **EXTFPF**: extracts 2D fields from grid-point post-processing file. Writes out the post-processed fields of the post-processing grid-point buffer. Performs a transpose from ordering by the **NFPROMA** based sub-rows of packets used in the grid-point calculations to an ordering by fields. The result is in the form of an array **PREAL** containing the extracted fields sorted out from the first subdomain to the last subdomain. For distributed memory the first part of this routine (data extraction) performs DM-local computations on the current processor data only. Ends by communications between processors to collect data on the **NSTROUT** processors in order to have data ready to be written on files.
- **EXTGPF**: extracts 2D fields from grid-point file. Writes out the model level fields of the model grid-point buffer. Performs a transpose from ordering by the **NPROMA** based packets used in the grid-point calculations to an ordering by fields. The result is in the form of an array **PREAL** containing the extracted fields stored from the North pole to the South pole (poles excluded) stored in a compact way taking into account the reduced grid. The result is suitable for being fed directly to the output routines **FAIENC** or grib packing routines **GRIBEX**.
- **FILLB**: memory transfer routine; calls **PACK21** if packing is required. Does the inverse operation of routine **EMPTB**. For distributed memory computations are DM-local.

- **FREEMEM**: interface routine to free memory used by pointer allocations.
- **GPNORM2**: computes grid-point norms of data structures written with **SC2WRG**.
- **INCGPF**: includes 2D fields into array/work file. Writes in model fields into the model grid-point buffer. Performs a transpose from ordering by fields to an ordering by the **NPROMA** based packets used in the grid-point calculations. Warning: **PREAL** never contain the poles values.
- **IOPACK**: interface for writing data on ARPEGE, ALADIN or GRIB files.
- **MAXGPFV**: computes maximum value of a grid-point field. Extracts a field from model work file or array, then computes its maximum value. If the field is not found, nothing is done. In distributed memory, each processor reads its own data in the buffer **GPPBUF**, then data are collected in the "main" processor, then the maximum of the field is computed in the "main" processor and communicated to all the other processors.
- **MXMAOP**: product matrix times matrix, also used in spectral computations and spectral transforms.
- **NEWFA**: opens a new ARPEGE/ALADIN file.
- **OPENFA**: opens an ARPEGE/ALADIN file and does tests of compatibility with model.
- **PREPACKA**: preliminar packing of gridpoint or spectral data before writing out to file: packs a chunk of DM-global fields (either spectral or gridpoint) prior to writing out on an ARPEGE/ALADIN file. WARNING : spectral data is expected to be ordered like in a FA file.
- **PKGRIDA**: packs model grid point variables by writing to then reading from an ARPEGE file. At the beginning and the end of the routine data are in a buffer. If distributed memory destoring from buffer is DM-local, then a communication between processors collects data in the "main" processor, then packing by writing/reading on an ARPEGE file is made in the "main" processor, then packed data are dispatched in all the processors, then each processor stores its own data on a buffer.
- **PKSPECA**: packs model spectral variables by writing to then reading from an ARPEGE file.
- **PKSPECG**: packs model spectral variables by writing to then reading from a GRIB file.
- **RDSPEC**: memory transfer of spectral data.
- **PKGRIDG**: the same as **PKGRIDA** but for GRIB files.
- **SC2RDG**: memory transfer routine for **NPROMA**-arrays; reads data in a buffer and copies it in a grid-point array. For distributed memory computations are DM-local.
- **SC2RDGFP**: version of **SC2RDG** used in FULL-POS for arrays with another dimension than **NPROMA**. For distributed memory computations are DM-local.
- **SC2WRG**: memory transfer routine for **NPROMA**-arrays; reads data in a grid-point array and copies it in a buffer. Does the inverse operation of routine **SC2RDG**. For distributed memory computations are DM-local.
- **SC2WRGFP**: version of **SC2WRG** used in FULL-POS for arrays with another dimension than **NPROMA**. For distributed memory computations are DM-local.
- **SPCONVERT** (**ESPCONVERT** + **NHCONVERT** for ALADIN): to convert model spectral fields into the fields actually written on file, NH variables are scaled.
- **SUALSPA**: allocation of spectral arrays.
- **WRSPEC**: memory transfer of spectral data.

8 Sequences of calls of post-processing.

8.1 Vertical coordinates of post-processing.

Post-processing can be made on pressure levels, height levels, potential vorticity levels, potential temperature levels or η-levels. For post-processing on pressure levels a vertical coordinate linked to pressure (pressure or logarithm of pressure, according to the post-processed variable) is used to compute weights for vertical interpolations. For post-processing on height levels a vertical coordinate linked to geopotential height is used to compute weights for vertical interpolations. For post-processing on potential vorticity levels, potential temperature levels or η-levels an intermediate state through a pressure-type coordinate is needed to compute vertical weights.

8.2 Sequences of calls: general considerations.

Post-processing for each coordinate is made by different sequences under routine **control/STEPO**. Vertical interpolations and horizontal interpolations are made by different sequences under **control/STEPO**. A sequence is defined by nine letters (or zeros) [L1][L2][L3][L4][L5][L6][L7][L8][L9] (variable **CLCONF** in routine **control/CNT4** and **CDCONF** in routine **control/STEPO**).

- L1 controls the file write/read.
- L2 controls the inverse Legendre transforms.
- L3 controls the inverse Fourier transforms.
- L4 controls the grid-point computations for dynamics and physics.
- L5 controls the grid-point computations for post-processing.
- L6 controls the grid-point computations for assimilation.
- L7 controls the direct Fourier transforms.
- L8 controls the direct Legendre transforms.
- L9 controls the spectral computations.

For example a model integration time-step is defined by the sequence [L1]AAA00AAA. Additional sequences can be performed by calls to **control/SCAN2MDM** (DM) under routines other than **control/STEPO**. The next subsections will detail sequences for different type of output post-processed data. Note that writing fields post-processed by a sequence [L2][L3][L4][L5][L6][L7][L8][L9] is managed by the [L1] of the following sequence.

∗ Abbreviations:

- DYN3D: 3D dynamical fields.
- DYN2D: 2D dynamical fields.
- DYN2DGP: 2D dynamical fields always written in grid-point.
- MSLP: mean sea level pressure.
- PHIS: surface orography.
- PS: surface pressure.
- PHYSOL: surface physical fields.
- CFU: cumulated fluxes.
- XFU: instantaneous fluxes.

∗ **Remark:** CFU and XFU are treated like PHYSOL and not detailed on the following tables.

8.3 Global spectral post-processed fields in the same horizontal geometry as the model geometry.

Variable **CFPFMT** is equal to 'MODEL' in **NAMFPC**.

Vertical coordinate for interpolation → Type of interpolated variable ↓	extrapolations for 2D fields.	pressure (with spectral fit)	height (without spectral fit)	PV (with spectral fit)	Θ (with spectral fit)	η (with spectral fit)
DYN3D	/	AAA0B0PPP E00000000	AAA0H0000 U00000000	AAA0V0PPP Y00000000	AAA0T0PPP M00000000	AAA0S0PPP Z00000000
DYN2D	AAA0B0PPP E00000000	/	/	/	/	/
PHYSOL, CFU or XFU	[*] A00000000	/	/	/	/	/
DYN3D + DYN2D + PHYSOL	/	[*] (PHYSOL) AAA0B0PPP E00000000 (DYN3D+DYN2D)	[*] (PHYSOL) AAA0H0000 U00000000 (DYN3D+DYN2D)	[*] (PHYSOL) AAA0V0PPP Y00000000 (DYN3D+DYN2D)	[*] (PHYSOL) AAA0T0PPP M00000000 (DYN3D+DYN2D)	[*] (PHYSOL) AAA0S0PPP Z00000000 (DYN3D+DYN2D)

∗ Remarks:

- [∗] corresponds to a sequence **GRIDFPOS** → (**SCAN2H** → **SCAN2MDM** → (**HPOS**+**HPOSLAG**) with [L5]='P') and **WRHFP** (with [L1]='I') by-passing **STEPO**. Equivalent to call **STEPO** first with sequence 0000P0000 then with sequence I00000000. Since there is no change of horizontal geometry here, horizontal interpolations are replaced by simple memory transfers. This sequence treats surface physical fields, CFU or XFU fields.
- All sequences called by **STEPO** call **VPOS**, but not **HPOS**, **HPOSLAG** and **ENDVPOS**. **STEPO** is generally called by **DYNFPOS** (under **CNT4**), except the last call to **STEPO** (where only [L1] is non zero) which is called directly by **CNT4**.
- **CPREP4** is never called.
- Headline of organigramme under **CNT4** is generally:
  - **SU4FPOS**
  - **GRIDFPOS** →
    - **SCAN2H** → **SCAN2MDM** → (**HPOS**+**HPOSLAG**) for physics or CFU or XFU if post-processing of such fields is required.
    - **WRHFP** to write post-processed physics or CFU or XFU fields.
  - **DYNFPOS** (always called): **DYNFPOS** calls one **STEPO** sequence (call to vertical interpolations only). Direct spectral transforms are called for fields to be fitted. Writing on files is made under routine **WRSFP**.
  - **STEPO** (last call for the case where only [L1] is non zero); writing on files is made under routine **WRSFP** for spectral fields and under routine **WRHFP** for grid-point fields.
- For η vertical coordinate, the sequence is the same either the η values are a subset of model values or not.
- All 3D variables are directly interpolated by vertical interpolations under **VPOS**, there is no call to **ENDVPOS**.
- Spectral fit is compulsory to post-process velocity potential and stream function.
- For height coordinate, the switch controlling spectral fit is not available in namelist and is set to .FALSE. when only 3D dynamical fields are post-processed. So there is no spectral fit and 3D dynamical fields are provided in grid-point values.
- When spectral fit is activated, post-processed fields are spectral ones for 3D dynamical fields, surface pressure, mean sea level pressure and orography, grid-point ones for the other 2D dynamical fields, the surface physical fields, the CFU and XFU.
- Names of articles in namelist **NAMAFN** have to be the same as the names in the input file.
- Unavailable variables:
  - Post-processing of velocity potential and stream function is not available for height coordinate.
  - Post-processing of pressure (resp. potential vorticity, potential temperature) is not made even if required for pressure (resp. potential vorticity, potential temperature) coordinate.
  - Post-processing of stretching deformation and shearing deformation is not available if FULL-POS is made in a semi-Lagrangian job.
  - Post-processing of temperature and $\Theta_w^{'}$ does not work for height coordinate.

8.4 Global grid-point post-processed fields in the same horizontal geometry or in a different horizontal geometry as the model geometry.

Variable **CFPFMT** is equal to 'GAUSS' and variable **LFPSPEC** is .FALSE. in **NAMFPC**. This configuration does not require any spectral fit in the arrival geometry. Spectral fit can be done in the departure geometry. The case where there is no change of horizontal geometry uses the same computations except the fact that horizontal interpolations are not done and are replaced by simple memory transfers.

8.4.1 Spectral fit.

Vertical coordinate for interpolation → Type of interpolated variable ↓	extrapolations for 2D fields.	pressure (with spectral fit)	height (without spectral fit)	PV (with spectral fit)	Θ (with spectral fit)	η (with spectral fit)
DYN3D	/	AAA0B0PPP 0PP0A0000 E00000000	AAA0M0000 0000I0000 0000Z0000 U00000000	AAA0V0PPP 0PP0A0000 Y00000000	AAA0T0PPP 0PP0A0000 M00000000	AAA0M0000 0000I0000 0000E0000 Z00000000
DYN2D	AAA0B0PPP 0PP0A0000 E000M0000 0000I0000 0000Z0000 U00000000	/	/	/	/	/
PHYSOL, CFU or XFU	[*] A00000000	/	/	/	/	/
DYN3D + DYN2D + PHYSOL	/	[*] (PHYSOL) AAA0B0PPP 0PP0A0000 E000M0000 0000I0000 0000Z0000 U00000000	[*] (PHYSOL) AAA0B0PPP 0PP0A0000 E000M0000 0000I0000 0000Z0000 U00000000	[*] (PHYSOL) AAA0M0000 0000I0000 0000Z0000 U000V0PPP 0PP0A0000 Y00000000	[*] (PHYSOL) AAA0M0000 0000I0000 0000Z0000 U000T0PPP 0PP0A0000 M00000000	[*] (PHYSOL) AAA0B0PPP 0PP0A0000 E000M0000 0000I0000 0000E0000 Z00000000

∗ Remarks:

- [∗] corresponds to a sequence **GRIDFPOS** → (**SCAN2H** → **SCAN2MDM** → (**HPOS**+**HPOSLAG**) with [L5]='P') and **WRHFP** (with [L1]='I') by-passing **STEPO**. Equivalent to call **STEPO** first with sequence 0000P0000 then with sequence I00000000. This sequence treats surface physical fields, CFU or XFU fields.
- For each of these configurations one has a preliminary [L5]='G' sequence **GRIDFPOS** → **SCAN2H** (replaces the treatment of the old buffer **FPBUF2** which does not exist any longer).
- Sequences called by **STEPO** with [L5]='A', 'Z' or 'E' call (**HPOS**+**HPOSLAG**).
- All sequences called by **STEPO** with [L5]='B', 'M', 'V' or 'T' call **VPOS**. **STEPO** is generally called by **DYNFPOS** (under **CNT4**), except the last call to **STEPO** (where only [L1] is non zero) which is called directly by **CNT4**.
- Sequences called by **STEPO** with [L5]='Z' or 'E' call **ENDVPOS**, but exist only if variables other than model variables are post-processed.
- **CPREP4** is never called.
- Headline of organigramme under **CNT4** is generally:
  - **SU4FPOS**
  - **GRIDFPOS** →
    - **SCAN2H** → **SCAN2MDM** → (**HPOS**+**HPOSLAG**) for physics or CFU or XFU if post-processing of such fields is required.
    - **WRHFP** to write post-processed physics or CFU or XFU fields.
  - **DYNFPOS** (always called): **DYNFPOS** calls one or several **STEPO** sequences according to the 3D or 2D dynamical fields to post-process: call to horizontal interpolations only if [L5]='A' or 'I', vertical interpolations before horizontal interpolations only if [L5]='B', 'M', 'V' or 'T', vertical interpolations after horizontal interpolations only if [L5]='I', **ENDVPOS** only if [L5]='Z' or 'E'. Direct spectral transforms are called for fields to be fitted.

- **STEPO** (last call for the case where only [L1] is non zero, routine **WRHFP**).

- For η vertical coordinate, the sequence is the same either the η values are a subset of model values or not.
- **ENDVPOS** is called only if post-processing of 3D variables other than model variables is wanted, and for height or η coordinates only ([L5]='Z' or 'E').
- **APACHE** is called only if post-processing of model variables and height or η vertical coordinate (case [L5]='M').
- Spectral fit is compulsory to post-process velocity potential and stream function.
- For height coordinate, the switch controlling spectral fit is not available in namelist and is set to .FALSE. when only 3D dynamical fields are post-processed. So there is no spectral fit.
- For η coordinate, there is no spectral fit for 3D dynamical variables.
- When spectral fit is activated, post-processed fields are fitted for 3D dynamical fields (except if η vertical coordinate), mean sea level pressure and orography.
- Names of articles in namelist **NAMAFN** have to be the same as the names in the input file.
- Restrictions of use for η-coordinate or height coordinate:
  - 3D variables which can be post-processed:
    * Temperature T.
    * Horizontal wind components U and V, wind velocity.
    * Specific humidity (moisture) q.
    * Relative humidity HU.
    * Potential temperature Θ.
    * Moist (irreversible) pseudo-adiabatic potential temperature $\Theta_w^{'}$.
    * Equivalent potential temperature Θ_e.
    * Extra GFL variables q_{SV}.
    * Pressure Π.
    * Possibility to post-process some additional free upper air fields.
  - 3D variables which cannot be post-processed (abort due to the fact that there is no spectral fit):
    * Velocity potential χ.
    * Stream function ψ.
  - Other 3D variables which cannot be post-processed (post-processing not made even if required):
    * Geopotential height gz.
    * Ozone $O3$.
    * Ice content q_i.
    * ω (pressure coordinate vertical velocity).
    * Relative vorticity ζ.
    * Divergence D.
    * Liquid water content q_l.
    * Cloudiness q_a.
    * Absolute vorticity $\zeta + f$.
    * Stretching deformation STD.
    * Shearing deformation SHD.
    * Potential vorticity PV.
    * Montgomery geopotential Φ_{mg}.
- Other restrictions of use:
  - Post-processing of pressure (resp. potential vorticity, potential temperature) is not made even if required for pressure (resp. potential vorticity, potential temperature) coordinate.
  - Post-processing of stretching deformation and shearing deformation is not available if FULL-POS is made in a semi-Lagrangian job.

8.4.2 No spectral fit.

Sequence
[L1][L2][L3][L4][L5]0PPP followed by 0PP0[L5][L6][L7][L8][L9] is replaced by sequence [L1][L2][L3][L4][L5]0000 followed by 0000[L5][L6][L7][L8][L9]. Routine **SPOS** is no longer called.

8.5 Global spectral post-processed fields in a horizontal geometry different from the model geometry.

Variable **CFPFMT** is equal to 'GAUSS' and variable **LFPSPEC** is .TRUE. in **NAMFPC** (pseudo-configuration 927). This configuration requires a spectral fit in the arrival geometry.

Routine **CPREP4** is always called. Organigramme has recursivity and is of the following type: **CNT0** (model geometry) → set-up, control routines, including **CNT4**. Headline of organigramme under this call to **CNT4** does not depend on the vertical coordinate of post-processing and on the list of post-processed variables, and is generally:

- **SU4FPOS**
- **CPREP4** →
  - **GRIDFPOS** →
    - **SCAN2H** → **SCAN2MDM** → (**HPOS**+**HPOSLAG**) for physics or CFU or XFU if post-processing of such fields is required.
    - **WRHFP** to write post-processed physics or CFU or XFU fields on unit 91.
  - **DYNFPOS** (always called):
    - first call to **STEPO**: inverse spectral transform; then (**SCAN2H** → **SCAN2MDM** →): vertical interpolations (**VPOS**) on the arrival η-levels (departure horizontal geometry), then horizontal interpolations (**HPOS** + **HPOSLAG**) on the arrival (vertical and horizontal) geometry for grid-point model variables (wind components, temperature, humidity, surface pressure, surface orography).
    - second call to **STEPO**: writing grid-point interpolated fields on model layers on unit 91 (routine **WRHFP**).
  - **CNT0** (output geometry).

CNT0 (output geometry) → set-up, control routines, including:

- a call to **SPECFITADM** under **CNT1**: spectral transformation of model fields (except wind components) and surface orography in unit 91, computation of spectral velocity potential and stream function on model η-levels, writing of all these fields on unit 91.
- a call to **CNT4**.

Headline of organigramme under this call to **CNT4** is generally:

- **SU4FPOS**
- **CPREP4** →
  - **GRIDFPOS** →
    - **SCAN2H** → **SCAN2MDM** → (**HPOS**+**HPOSLAG**) for physics or CFU or XFU if post-processing of such fields is required (horizontal interpolations are not done and replaced by simple memory transfers).
    - **WRHFP** to write post-processed physics or CFU or XFU fields on unit 54.
  - **DYNFPOS** (always called): **DYNFPOS** calls one or several **STEPO** sequences according to the 3D or 2D dynamical fields to post-process (call to vertical interpolations, horizontal interpolations are neutral here and replaced by simple memory transfers). Direct spectral transforms are called for fields to be fitted.
  - **STEPO** (last call for the case where only [L1] is non zero, routine **WRSFP** to write spectral fields and routine **WRHFP** to write grid-point fields).
  - **CNT0** (return to model geometry to continue integration if required).

Post-processed fields are written on unit 54.

Under the first call to **CNT0** one finds the following sequences of **STEPO**: AAA0M0000 (vertical interpolations on departure geometry, simple copy on model η-levels), then 0000A0000 (horizontal interpolations on grid-points of the arrival geometry), then Z00000000.

Under the second call to **CNT0** one finds the following sequences of **STEPO**: 0AA0B0PPP then E00000000.

∗ Remarks, restrictions of use:

- **APACHE** is called if post-processing on height or η-levels.
- **ENDVPOS** is theoretically called if post-processing on height or η-levels to post-process dynamical variables which are not model variables (does not work in cycle 30). In practical configurations calling **ENDVPOS** are not completely coded, so **ENDVPOS** is never called for configurations which work.
- It is necessary to post-process at least physical surface fields, in the contrary fields are not written in the first call to **CNT0** but read in the second call to **CNT0** (abort when reading inexistant fields).
- Names of articles in namelist **NAMAFN** have to be the same as the names in the input file.
- For η vertical coordinate, the sequence is the same either the η values are a subset of model values or not.
- Spectral fit is compulsory to post-process velocity potential and stream function.
- For height coordinate, the switch controlling spectral fit is not available in namelist and is set to .FALSE. when only 3D dynamical fields are post-processed. So there is no spectral fit and 3D dynamical fields are provided in grid-point values.
- When spectral fit is activated, post-processed fields are spectral ones for 3D dynamical fields, surface pressure, mean sea level pressure and orography, grid-point ones for the other 2D dynamical fields, the surface physical fields, the CFU and XFU.
- Restrictions of use for η-coordinate:
  - If no change of vertical coordinate, the number of levels to be post-processed has to be the model number of levels.
  - If change of set of vertical levels, the number of levels to be post-processed has to be the number of levels specified in namelist **NAMFPG**.
  - 3D variables which can be post-processed: temperature, velocity potential (if spectral fit), stream function (if spectral fit), horizontal wind components, specific humidity, wind velocity, absolute vorticity, pressure, non hydrostatic variables.
  - 2D variables which can be post-processed: surface pressure, spectral orography.
  - It works for the “true” 927: post-processing of temperature, velocity potential, stream function, specific humidity, non hydrostatic variables if NH, surface pressure, spectral orography. If these variables are not asked together, FULL-POS may sometimes abort with diagnostic “ETA-LEVEL SELECTION IMPOSSIBLE”.
- Restrictions of use for height coordinate: generally it does not work.
- Restrictions of use for pressure coordinate:
  - it works when the second part of the 927-configuration does only one sequence of **STEPO**, this is the case when surface pressure is not processed together with other dynamical 3D or 2D fields.
- Restrictions of use for potential temperature or potential vorticity coordinate:
  - it works when the second part of the 927-configuration does only one sequence of **STEPO**, this is the case when only 3D dynamical fields are asked.
- Other restrictions of use:
  - Post-processing of pressure (resp. potential vorticity, potential temperature, geopotential height) is not made even if required for pressure (resp. potential vorticity, potential temperature, height) coordinate.
  - Post-processing of stretching deformation and shearing deformation is not available if FULL-POS is made in a semi-Lagrangian job.

A solution to run not working configurations is to perform two calls to FULL-POS, first a configuration 927 which creates an historic file in the final geometry (new η coordinate), then a FULL-POS run without change of horizontal geometry and with global spectral outputs (**CFPFMT**='MODEL').

8.6 Grid-point post-processed fields in a ALADIN limited area domain.

Variable **CFPFMT** is equal to 'LELAM' and variable **LFPSPEC** is .FALSE. in **NAMFPC**.

Same sequences and same organigramme as in the case **CFPFMT**='GAUSS' and **LFPSPEC**=.FALSE. .

Such a job requires an executable with the ALADIN routines (but **LELAM** is set to .FALSE.).

8.7 Grid-point post-processed fields in a latitude-longitude limited area domain.

Variable **CFPFMT** is equal to 'LALON' and variable **LFPSPEC** is .FALSE. in **NAMFPC**.

Same sequences and same organigramme as in the case **CFPFMT**='GAUSS' and **LFPSPEC**=.FALSE. .

8.8 Mixed ARPEGE/IFS-ALADIN FULL-POS configurations: spectral post-processed fields in a limited area domain.

∗ **Spectral post-processed fields in a limited area domain.** Variable **CFPFMT** is equal to 'LELAM' and variable **LFPSPEC** is .TRUE. in **NAMFPC** (pseudo-configuration E927). ALADIN libraries are necessary in this case (take cycle AL30 ALADIN library and cycle 30 ARPEGE/IFS library). The first call to **CNT0** makes an ARPEGE/IFS run, the second one makes an ALADIN run.
Same sequences and same organigramme as in the case **CFPFMT**='GAUSS' and **LFPSPEC**=.TRUE. with change of horizontal geometry (pseudo-configuration 927).

∗ **Bogussing (post-processing in a subset of an ARPEGE domain from an ALADIN file).** This configuration has to be described later.

8.9 Pure ALADIN FULL-POS configurations.

ALADIN libraries are necessary in this case (take cycle AL30 ALADIN library and cycle 30 ARPEGE/IFS library).

- **CFPFMT**='MODEL', **LFPSPEC**=.FALSE.: spectral outputs on an ALADIN domain which is identical to the model ALADIN domain. Same sequences as in the case (**CFPFMT**='MODEL', **LFPSPEC**=.FALSE.) of ARPEGE/IFS. One call to **CNT0**, no call to **CPREP4**.
- **CFPFMT**='LELAM', **LFPSPEC**=.FALSE.: grid-point outputs on an ALADIN domain which is included in the model ALADIN domain. Cannot work if the required post-processing ALADIN domain is not included in the model ALADIN domain. Same sequences as in the case (**CFPFMT**='LELAM', **LFPSPEC**=.FALSE.) of ARPEGE/IFS. One call to **CNT0**, no call to **CPREP4**.
- **CFPFMT**='LALON', **LFPSPEC**=.FALSE.: grid-point outputs on some LALON domains which are included in the model ALADIN domain. Cannot work if the required post-processing LALON domains are not included in the model ALADIN domain. Same sequences as in the case (**CFPFMT**='LALON', **LFPSPEC**=.FALSE.) of ARPEGE/IFS. One call to **CNT0**, no call to **CPREP4**.
- **CFPFMT**='GAUSS', **LFPSPEC**=.FALSE. has no sense for ALADIN.
- **CFPFMT**='LELAM', **LFPSPEC**=.TRUE.: pseudo-configuration EE927, spectral outputs on an ALADIN domain which is included in the model ALADIN domain. Cannot work if the required post-processing ALADIN domain is not included in the model ALADIN domain. Same sequences as in the cases (**CFPFMT**='LELAM', **LFPSPEC**=.TRUE.) of the pseudo-configurations 927 end E927. Two calls to **CNT0** with **LELAM**=.T., **CPREP4** is called.

9 Some distributed memory features:

9.1 Calculations packets:

∗ **Grid-point computations:** The total number of processors involved in the A-level parallelisation is **NPRGPNS**. The total number of processors involved in the B-level parallelisation is **NPRGPEW**. One processor treats **NGPTOT** points (a part of the Gaussian grid points). The total amount of grid-points for all the processors is **NGPTOTG**. The maximum value of **NGPTOT** is **NGPTOTMX**. In the grid-point space there is a subdivision of the current processor grid-points into **NGPBLKS**=int[(**NGPTOT**+**NPROMA**-1)/**NPROMA**] packets of length **NPROMA** (the useful number of values in each packet is lower or equal than **NPROMA**). These packets do not contain neither extra-longitudes nor extra-polar latitudes data. A **NPROMA**-packet does not always contain a set of complete latitudes. This subdivision into **NPROMA**-packet only concern not lagged computations. Lagged computations (interpolations) are made at one time for all the **NFPRGPL** FULL-POS points treated by the current processor. For the FULL-POS lagged computations there is a second subdivision into sub-packets of length **NFPROMA** containing useful points for computations at the points of the arrival horizontal geometry (routines **SUWFPBUF**, **HPOSLAG** and **ENDVPOS**). More details will be given later for the data transmission for horizontal interpolations. One 2D field has **NGPTOTG** points divided into **NPRGPNS*****NPRGPEW** sets of **NGPTOT** points treated by each processor. **NGPTOT** does not take account of the extra-longitudes and the extra-polar latitudes. All these variables take account of the reduced Gaussian grid. It is assumed and hardcoded that there are one western extra-longitude and two eastern extra-longitudes. The DM-global longitude $jlon = 1$ is always the "Greenwich" meridian of the computational sphere. All the vertical levels and the variables corresponding to a same grid-point are treated by the same processor. There are necessary transpositions (reorganisation of data) between grid point computations and Fourier transforms because Fourier transforms need complete latitudes.

∗ **Fourier transforms:** The total number of processors involved in the A-level parallelisation is **NPRTRNS**. The total number of processors involved in the B-level parallelisation is **NPRTRV**. Fourier transforms are done latitude by latitude for each **NPROMA**-packet. A processor treats a subset of latitudes, one latitude at the time. A processor can treat only a subset of fields to be post-processed (if **NPRTRV**>1). Data reorganisation and transpositions are necessary in the Fourier space between the zonal wave structure necessary for Legendre transforms and the latitudinal structure necessary for Fourier transforms.

∗ **Legendre transforms:** 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 **NPRTRV**. Legendre transforms are done zonal wave number by zonal wave number. A processor treats a subset of zonal wave numbers, one zonal wave number at the time.

∗ **Filtering in spectral space:** The total number of processors involved in the A-level parallelisation is **NPRTRM**. The total number of processors involved in the B-level parallelisation is **NPRTRV**. Filtering in spectral space is done zonal wave number by zonal wave number. A processor treats a subset of zonal wave numbers, all this work being done in one call of **SPOS**.

9.2 Transmission of data necessary for FULL-POS horizontal interpolations from HPOS to HPOSLAG: interpolation buffers.

First one associates to each interpolation point, which is generally not a model grid-point, an "associated" model grid-point which currently satisfies to the following rule:

- the "associated" model grid-point is on the longitude immediately west to the interpolation point.
- the "associated" model grid-point is on the latitude immediately north to the interpolation point if the interpolation point is not between the Gaussian North pole and the first Gaussian latitude; in the contrary one takes the first Gaussian latitude.

This associated model grid-point is always between the latitudes 1 and **NDGLG** (it is never a pole or an extra-polar latitude point). The processor which treats the FULL-POS interpolation point is the processor which treats this associated model grid-point.

Interpolations use data of points which are not necessary on the same latitude and longitude as the interpolation point. Thus interpolation routines need to have access to a limited number of surrounding latitudes and longitudes which are not necessary treated by the current processor. Interpolation points are collected so that a call to **HPOSLAG** for horizontal interpolations treats a set of points, the "associated" model grid-points set of which is a **NGPTOT** set which is treated by one processor. The number of surrounding latitudes and longitudes rows necessary for interpolations but which do not belong to the current processor is precomputed in the subroutine **SUFPWIDE** (variable **NFPWIDE**). This is a sort of "halo" belonging to some other processors. Due to the "halo" there is still need to split calculations into not lagged ones (**HPOS**) and lagged ones (**HPOSLAG**). Quantities to be interpolated are computed in the non-lagged part and interpolations are performed in the lagged part. In **HPOS**, only data of the current processor (without any extra-longitudinal data nor polar and extra-polar data) are computed. For all the **NPROMA**-packages treated by the current processor these data are stored in the arrays **PFPBUF1** in **HPOS** (corresponding to local array **ZFPBUF1AUX** in the routine **SCAN2MDM**). The first dimension of **PFPBUF1** is **NPROMA**. The second dimension of **PFPBUF1** is **NFPBFIN**: for each buffer this is the number of 2D quantities to be interpolated. **ZFPBUF1AUX** has only two dimensions in **SCAN2MDM**. Then a memory transfer is immediately done after each call to **HPOS** in a bigger interpolation buffer where some place is let for extra-longitudes (assumption is made that interpolations use only the data of 2 longitudes west and 2 longitudes east) and the halo (buffers **ZFPBUF1** in **SCAN2MDM**). This memory transfer uses a precomputed intermediate array **NFPCORE**. Then some communication routines are called in **SCAN2MDM** to constitute the halo. **SLCOMM** does processor communication to constitute the halo (receives and sends data from some other processors). **SLEXTPOL** adds data of extra-polar latitudes in the halo when necessary. The first dimension of **ZFPBUF1** is **NAFPB1** which is the total number of points one needs for the interpolations (**NAFPB1** is greater than **NGPTOT**). The second dimension of **ZFPBUF1** is the same as for **PFPBUF1**: the number of 2D fields to be interpolated. When all the **NAFPB1** dataset is constituted, the lagged part **HPOSLAG** is called which do horizontal interpolations for all the **NFPRGPL** data to be interpolated for the current processor.

The lagged part of the computations (**HPOSLAG**, **ENDVPOS**) has a subdivision into **NFPROMA**-packets. The number of packets is **NFPBLOCS**.

Set-up routines **CPCLIMI** and **CPCLIMO** use the same logics.

9.3 B-level parallelization:

The B-level parallelization is a circular one in spectral space and spectral transforms; the list of fields to be interpolated is considered as a list of 2D fields and these 2D fields are spread among the **NPRTRV** processors (fields numbers 1 to **NPRTRV** on processors 1 to **NPRTRV**, then fields numbers **NPRTRV**+1 to 2∗**NPRTRV** on processors 1 to **NPRTRV**, etc). When it is necessary some fields are kept together (for example U and V on a same layer) so the distribution is not completely circular.

10 Module and namelist variables to be known:

10.1 Modules for declarations.

10.1.1 Parameter PARFPOS.

Contains basic dimensions for FULL-POS post-processing (in parenthesis, value currently attributed). All variables are DM-global.

```
* JPOSDOM : maximum number of horizontal (sub)domains (15).
* JPOSLEN : maximum length of a (sub)domain name (10).
* JPOSLIS : maximum number of groups of subdomains (10).
* JPOSDIR : maximum length of the path (or prefix) for the output files (180).
* JPOSLE  : maximum number of eta-levels on the output subdomain (= JPMXLE).
* JPOSGL  : maximum number of latitude rows of the output Gaussian grid (= JPMXGL).
* JPOSCFU : maximum number of cumulated fluxes (63).
* JPOSXFU : maximum number of instantaneous fluxes (63).
* JPOS3P  : maximum number of post-processing pressure levels (43).
* JPOS3H  : maximum number of post-processing height (above orography) levels (127).
* JPOS3TH : maximum number of post-processing potential temperature levels (15).
* JPOS3PV : maximum number of post-processing potential vorticity levels (15).
* JPOS3S  : maximum number of post-processing eta-levels (= JPOSLE).
* JPOSSCVA: maximum number of post-processable extra GFL variables (5).
* JPOSAERO: maximum number of post-processable aerosols (3).
* JPOSVX2 : maximum number of post-processable extra surface fields (70).
* JPOSFSU : maximum number of free gp/sp surface fields (15).
* JPOSSGP : maximum number of surface gridpoint fields (157+JPOSVX2+JPOSFSU).
* JPOSPHY : maximum number of surface fields 'PHYSOL' (157+JPOSVX2+JPOSFSU+JPOSCFU+JPOSXFU).
* JPOS3DF : maximum number of specific 3D dynamical fields (63+JPOSSCVA+JPOSVX2+JPOSAERO).
* JPOS2DF : maximum number of specific 2D dynamical fields (63+JPOSFSU).
* JPOSDYN : Maximum number of dynamic fields (JPOS3DF+JPOS2DF).
```

10.1.2 Pointer PTRFP4.

Contains control variables for post-processing of physical fields.

∗ Physics and fluxes:

```
* YRQGFP: current request description for calculation.
```

10.1.3 PTRFPB2.

Contains variables and pointers for auxiliary surface grid point fields used for horizontal post-processing. Variables are initialised in setup routine **setup/SUFPTR2**. No variable in namelist. All variables are DM-global.

∗ List of variables:

```
* MAFPTR : field pointers (fixed dimension).
* MFPVT0 : internal codes of auxiliary surface grid point fields.
* NFPVT0 : number of auxiliary 'surface' grid point fields.
```

10.1.4 YOM4FPOS.

Contains variables relative to FULL-POS working arrays (level 4). Variables are initialised in setup routine **setup/SU4FPOS**. No variable in namelist. All variables are DM-global.

∗ **Post-processing on 3D dynamical variables:**

```
* NTFPP  : total number of fields in P levels
* NTFPH  : total number of fields in H levels
* NTFPTH : total number of fields in THETA levels
* NTFPPV : total number of fields in PV levels
* NTFPS  : total number of fields in eta levels

* NOFPP  : total number of P levels for each field
* NOFPH  : total number of H levels for each field
* NOFPTH : total number of THETA levels for each field
* NOFPPV : total number of PV levels for each field
* NOFPS  : total number of eta levels for each field

* NAFPP  : total number of subdomains for each P level of each field
* NAFPH  : total number of subdomains for each H level of each field
* NAFPT  : total number of subdomains for each THETA level of each field
* NAFPV  : total number of subdomains for each PV level of each field
* NAFPS  : total number of subdomains for each eta level of each field

* NCPFP  : field pointers in P levels
* NCHFP  : field pointers in H levels
* NCTFP  : field pointers in THETA levels
* NCVFP  : field pointers in PV levels
* NCSFP  : field pointers in eta levels

* NPFP   : P     level pointers for each field
* NHFP   : H     level pointers for each field
* NTFP   : THETA level pointers for each field
* NVFP   : PV    level pointers for each field
* NSFP   : eta   level pointers for each field

* NIPFP  : subdomain index for each P level of each field
* NIHFP  : subdomain index for each H level of each field
* NITFP  : subdomain index for each THETA level of each field
* NIVFP  : subdomain index for each PV level of each field
* NISFP  : subdomain index for each eta level of each field
```

∗ **Post-processing on 2D dynamical variables:**

```
* NOFP2F : total number of 2D fields
* NOFP2A : total number of subdomains for each 2D field
* NC2FP  : 2-D field pointers
* NI2FP  : subdomain index for each 2D field
```

∗ **Post-processing on lagged physical and instantaneous fluxes variables:**

```
* NFPOSPHY: number of fields
* NCPHYFP : fields codes
* NAPHYFP : number of subdomains for each field
* NIPHYFP : subdomains pointers for each field
```

10.1.5 YOMAFN.

Contains ARPEGE fields descriptors. Variables are initialised under setup routine **setup/SUAFN**. All variables are DM-global.

∗ **DYN3D and DYN2D fields:** The global descriptor for all fields is **TFP_DYNDS**; for each field [X] there is an individual descriptor **TFP_[X]**. See section 2 for the list of DYN3D and DYN2D post-processable variables and for the code [X] (for example **TFP_[T]** is the descriptor for upper air temperature).

∗ **PHYSOL, CFU and XFU fields:** The global descriptor for all fields is **GFP_PHYDS**; for each field [X] there is an individual descriptor **GFP_[X]**. See section 2 for the list of PHYSOL, CFU and XFU post-processable variables and for the code [X] (for example **GFP_[LSM]** is the descriptor for land-sea mask).

∗ **Other remarks:** The way of computing the descriptors is done in routines **SUAFN1**, **SUAFN2** and **SUAFN3**. **SUAFN2** provides the ordering of individual fields in the global descriptor, this is this order which is adopted in the current documentation. **SUAFN1** uses a code-alphabetical order. The order in **SUAFN3** is still different and it is desirable in the future that the ordering of prints in **SUAFN3** match at least one of the orderings of **SUAFN1** or **SUAFN2**.

∗ **Namelist NAMAFN.** The following variables of **YOMAFN** are in namelist **NAMAFN**:

```
GFP_VX2   ,GFP_FSU   ,GFP_10U   ,GFP_10V   ,GFP_2D    ,GFP_2T    ,
GFP_SUND  ,GFP_BLH   ,GFP_ACOT  ,GFP_AL    ,GFP_ALB   ,GFP_ALSN  ,
GFP_ANOR  ,GFP_ARG   ,GFP_BAAL  ,GFP_BLD   ,GFP_BSR   ,GFP_CBLD  ,
GFP_CCC   ,GFP_CCCC  ,GFP_CCHS  ,GFP_CCP   ,GFP_CCSF  ,GFP_CCVQ  ,
GFP_CCVS  ,GFP_CCVU  ,GFP_CCVV  ,GFP_CDRU  ,GFP_CDSW  ,GFP_CE    ,
GFP_CEAS  ,GFP_CETP  ,GFP_CFON  ,GFP_CHAR  ,GFP_CHCC  ,GFP_CICE  ,
GFP_CIRU  ,GFP_CLCC  ,GFP_CLHE  ,GFP_CLHS  ,GFP_CLI   ,GFP_CLSP  ,
GFP_CLSS  ,GFP_CMCC  ,GFP_CP    ,GFP_CQTO  ,GFP_CS    ,GFP_CSF   ,
GFP_CSLH  ,GFP_CSNS  ,GFP_CSOC  ,GFP_CSOD  ,GFP_CSOP  ,GFP_CSRU  ,
GFP_CSSH  ,GFP_CSSR  ,GFP_CSSW  ,GFP_CSTR  ,GFP_CTCC  ,GFP_CTHC  ,
GFP_CTHD  ,GFP_CTME  ,GFP_CTO3  ,GFP_CTOP  ,GFP_CTP   ,GFP_CTSP  ,
GFP_CTSR  ,GFP_CTTR  ,GFP_CTUQ  ,GFP_CTUS  ,GFP_CUGW  ,GFP_CUSS  ,
GFP_CVGW  ,GFP_CVSS  ,GFP_CTSOC ,GFP_CTTHC ,GFP_CWS   ,GFP_D2    ,
GFP_DPAT  ,GFP_XX2HU ,GFP_XN2HU ,GFP_DST   ,GFP_DSW   ,GFP_E     ,
GFP_EMIS  ,GFP_EWSS  ,GFP_FDSW  ,GFP_GFIS  ,GFP_GWD   ,GFP_HCC   ,
GFP_HV    ,GFP_IC    ,GFP_IE    ,GFP_IEWSS ,GFP_INSSS ,GFP_ISOR  ,
GFP_ISSHF ,GFP_IVEG  ,GFP_LAI   ,GFP_LAN   ,GFP_LCC   ,GFP_LGWS  ,
GFP_LSF   ,GFP_LSM   ,GFP_LSP   ,GFP_LSRH  ,GFP_LZ0H  ,GFP_MCC   ,
GFP_MGWS  ,GFP_MN2T  ,GFP_MSLD  ,GFP_MX2T  ,GFP_NSSS  ,GFP_RDST  ,
GFP_RDSW  ,GFP_RO    ,GFP_RSMIN ,GFP_SAB   ,GFP_SD    ,GFP_SDOG  ,
GFP_SF    ,GFP_SKT   ,GFP_SLHF  ,GFP_SLOR  ,GFP_SNDE  ,GFP_SR    ,
GFP_SRC   ,GFP_SSHF  ,GFP_SSR   ,GFP_SSRD  ,GFP_SSW   ,GFP_ST    ,
GFP_STL1  ,GFP_STL2  ,GFP_STL3  ,GFP_STL4  ,GFP_STR   ,GFP_STRD  ,
GFP_SWL1  ,GFP_SWL2  ,GFP_SWL3  ,GFP_SWL4  ,GFP_TCC   ,GFP_TCO3  ,
GFP_TCW   ,GFP_TCWV  ,GFP_TSN   ,GFP_TSR   ,GFP_TTR   ,GFP_PSRHU ,
GFP_PADOU ,GFP_PADOV ,GFP_PCAAG ,GFP_VEG   ,GFP_X10U  ,GFP_X10V  ,
GFP_X2RH  ,GFP_X2SH  ,GFP_X2T   ,GFP_XCCC  ,GFP_XGDU  ,GFP_XGDV  ,
GFP_XHCC  ,GFP_XLCC  ,GFP_XLSP  ,GFP_XLSS  ,GFP_XMCC  ,GFP_PCAPG ,
GFP_XCP   ,GFP_XCSF  ,GFP_XCVQ  ,GFP_XCVS  ,GFP_XCVU  ,GFP_XCVV  ,
GFP_XN2T  ,GFP_XSSR  ,GFP_XSTR  ,GFP_XTCC  ,GFP_XTSR  ,GFP_XTTR  ,
GFP_XTUQ  ,GFP_XTUS  ,GFP_XTUU  ,GFP_XTUV  ,GFP_XX2T  ,GFP_Z0F   ,
GFP_Z0H   ,GFP_X10FF ,GFP_XCAPE ,GFP_XMOCO ,GFP_XCLPH ,GFP_XGUST ,
GFP_XUGST ,GFP_XVGST ,GFP_CSMR  ,GFP_FSSW  ,GFP_CDUTP ,GFP_ALS   ,
GFP_ALV   ,GFP_SFIS  ,GFP_PVGMX ,GFP_Z0V   ,GFP_PURB  ,GFP_D2MX  ,
GFP_SPAR  ,GFP_SUVB  ,GFP_CAPE  ,GFP_TCLW  ,GFP_TCIW  ,
GFP_ASEA  ,GFP_ALAN  ,GFP_ASOO  ,GFP_ADES  ,
GFP_O3A   ,GFP_O3B   ,GFP_O3C   ,
GFP_XIRCD ,GFP_XIRCL ,GFP_XWVCD ,GFP_XWVCL ,
TFP_Z     ,TFP_T     ,TFP_U     ,TFP_V     ,TFP_Q     ,TFP_HU    ,
TFP_VV    ,TFP_VOR   ,TFP_DIV   ,TFP_PSI   ,TFP_KHI   ,TFP_W     ,
TFP_S     ,TFP_SN    ,TFP_RR    ,TFP_GR    ,TFP_TKE   ,
TFP_TH    ,TFP_THPW  ,TFP_CLF   ,TFP_O3MX  ,
TFP_WND   ,TFP_ETH   ,TFP_ABS   ,TFP_STD   ,TFP_PV    ,TFP_TWV   ,
TFP_SHD   ,TFP_P     ,TFP_PD    ,TFP_VD    ,TFP_VW    ,TFP_FUA   ,
TFP_GFL   ,TFP_MG    ,TFP_HUX   ,TFP_SP    ,TFP_MSL   ,TFP_FIS   ,
TFP_GM    ,TFP_FOL   ,TFP_WWS   ,TFP_LNSP  ,TFP_UCLS  ,TFP_VCLS  ,
TFP_TCLS  ,TFP_QCLS  ,TFP_RCLS  ,TFP_FCLS  ,TFP_CAPE  ,TFP_CIEN  ,
TFP_MOCO  ,TFP_UGST  ,TFP_VGST  ,TFP_FGST  ,TFP_HCLP  ,TFP_UJET  ,
TFP_VJET  ,TFP_PJET  ,TFP_TCAO  ,TFP_PCAO  ,TFP_FSU   ,TFP_TX    ,
```

```
TFP_TN    ,TFP_PTOB  ,TFP_HTOB  ,TFP_HTPW  ,TFP_HUN   ,TFP_IET   ,
TFP_PT1B  ,TFP_HT1B  ,TFP_SMC   ,TFP_ASMC  ,TFP_CPF   ,TFP_SPF   ,
TFP_HT1B  ,TFP_SMC   ,TFP_ASMC  ,TFP_CPF   ,TFP_SPF   ,TFP_AERO
```

Names of articles in namelist **NAMAFN** have to be the same as the names in the input file.

10.1.6 YOMAFPB.

Contains buffer for basic fields resulting from APACHE in FULL-POS. Variables are initialised in setup routine **setup/SUFPIOS** via **setup/SUFPSC2B**. No variable in namelist. All variables are DM-local.

∗ List of variables:

```
* NLENAFPBL: length of buffer.
* AFPBUF   : buffer.
* NSTAAFPB : start adresses for post-processed packets of points within buffer.
```

10.1.7 YOMAFPDS.

Absolute pointers of the basic fields which have been horizontally then vertically post-processed. Variables are initialised in setup routine **setup/SUVFPOS**. No variable in namelist. All variables are DM-global.

```
* NAFPOS : total number of fields in buffer
* MFPUT2 : field pointer for zonal component of wind
* MFPVT2 : field pointer for meridional component of wind
* MFPTT2 : field pointer for temperature
* MFPWVE2: field pointer for true vertical velocity ON UPPER HALF LEVELS
* MFPPDE2: field pointer for n.-h. pressure departure variable
* MFPSVT2: field pointer for extra GFL variables
* MFPLPT2: field pointer for Ln of surface pressure
* MFPFIS2: field pointer for surface geopotential
* MFPWWS2: field pointer for surface true vertical velocity
```

10.1.8 YOMCAPE.

Contains variables to control CAPE computation in FULL-POS. The following variables also present in namelist **NAMCAPE** can be useful for FULL-POS (these variables are DM-global).

```
* NCAPEITER : NUMBER OF ITERATIONS IN THE NEWTON LOOPS.
              Default is the same as NBITER in YOMPHY.
* NETAPES   : NUMBER OF INTERMEDIATE-LAYERS USED FOR CALCULATION OF
              VERTICAL ASCENT BETWEEN TWO MODEL PRESSURE LEVELS.
              Default value is 2.
* GCAPEPSD  : DEPTH OF LAYER ABOVE THE GROUND IN WHICH MOST
              UNSTABLE PARCEL IS SEARCHED FOR (UNIT: Pa)
              USED JUST FOR TYPE 2 OF CAPE. Default value is 30000.
              (CAPE Pressure Search Depth)
* GCAPERET  : FRACTION OF THE CONDENSATES WHICH IS RETAINED,
              I.E. WHICH DOES NOT PRECIPITATE.
              IF GCAPERET=1. ==> REVERSIBLE MOIST ASCENT.
                IT IS ASSUMED THAT ALL THE PARCEL'S CONDENSED
                WATER IS RETAINED, THUS CLOUD CONDENSATES
                REDUCE THE BUOYANCY.
              IF GCAPERET=0. ==> "IRREVERSIBLE" (PSEUDO-ADIABATIC) MOIST ASCENT.
                CLOUD CONDENSATES PRECIPITATE  INSTANTANEOUSLY
                AND THUS DO NOT AFFECT THE BUOYANCY.
              GCAPERET CAN BE USED WITH VALUES BETWEEN 0. AND 1..
              Default value is 0.
```

10.1.9 YOMCT0.

Contains control variables. The following variables also present in namelist **NAMCT0** can be useful for FULL-POS (these variables are DM-global).

```
* NCONF   : configuration
* LFPOS   : .T. = use of Full-POS rather than POS
* CFPNCF  : name of Full-POS control file (pseudo-configuration 927)
* LFPART2 : .T. = second part of interpolations for changing geometry (pseudo-configuration 927)
* CFDIRLST: path of postprocessing listing file
* CNPPATH : directory of postprocessing namelist files
* NFRPOS  : post-processing frequency
* NPOSTS  : post-processing control array
```

The following variables also present in namelist **NAMPAR0** can be useful for FULL-POS when running distributed memory jobs (these variables are DM-global).

```
* NPRGPNS : number of processors used in A-direction during grid-point phase in N-S direction.
* NPRGPEW : number of processors used in B-direction during grid-point phase in E-W direction.
* NPRTRW  : number of processors used in A-direction during Fourier space, Legendre
            transform phase and spectral space in wave space.
* NPRTRV  : number of processors used in B-direction during transform phase and
            spectral space in vertical direction.
* NPROC   : total number of processors requested.
* NOUTPUT : type of output.
            0 = no diagnostic output
            1 = only diagnostic output from processor number 1
            2 = diagnostic output from all processors into separate files
            Default value is 1.
* LMPOFF  : if .TRUE., no communication routine is called if 1 processor mode.
            Default value is .FALSE.
* LMPDIAG : extensive message passing diagnostic output requested if .TRUE.
            Default value is .FALSE.
```

The following variables are not present in namelist and control distributed memory. These variables are DM-global.

```
* NPRTRM : number of processors used during spectral computations in zonal wave direction.
* NPRTRN : number of processors used during spectral computations in total wave direction.
* NPRTRNS: number of processors used in A-direction during Fourier transform phase
           in North-South direction.
```

10.1.10 YOMDFPB.

Contains buffer for horizontally post-processed main fields. No variable in namelist. All variables are DM-local.

* List of variables:

```
* NLENHFPB : Core-zone buffer length
* NELENHFP : Extension-zone buffer length
* NSTAHFPB : Start adresses of NFPROMA-sized blocks in Core-zone buffer
* NESTAHFP : Start adresses of NFPROMEL-sized blocks in Extension-zone buffer
* HFPBUF   : Core-zone buffer for horizontally post-processed main fields
* HEFPBUF  : Extension-zone buffer for horizontally post-processed main fields
```

10.1.11 YOMDIM.

Contains dimensioning variables. The following variables are related with FULL-POS.

* Variables computed in SUDIM1 and SUDIM2:

```
* NDLON    : maximum number of Gaussian longitudes (DM-global).
* LOPTPROMA: .T./.F.: value of NPROMA optimised/not optimised.
* NPROMA   : working dimension of grid-point packets for grid-point computations.
* NGPBLKS  : number of NPROMA-subpackets in a processor (DM-local).
* NDGSAFPH : modified lower bound for latitude (DM-local).
* NDGENFPH : modified upper bound for latitude (DM-local).
```

*** Namelist NAMDIM:** The following variables of **YOMDIM** are in namelist **NAMDIM**: **NPROMA**.

10.1.12 YOMFP4L.

Contains information relative to lagged variables needed to write out post-processed fields. Variables are initialised in the sequence **pp_obs/DYNFPOS** → **setup/SUVFPOSL**. No variable in namelist. All variables are DM-global.

* A-DM-global and B-DM-local variables.

```
* NDYNFPL : number of fields.
* NFLDFPL : field pointer.
* NDOMFPL : number of subdomains for each field.
* NINDFPL : indices of subdomains for each field.
* LFPSRFL : T=surface field; F=upper-air field.
* RLEVFPL : level value.
* NFPSLOCG: DM-local index for each DM-local field for the DM-local array.
```

* DM-global variables.

```
* NFPSPBL : number of output spectral fields.
* NFPSPBG : number of output grid-point fields.
* NFPSFLDG: internal code for the fields.
* NFPSSETG: V-set where each fields can be found.
* LFPSSRFG: DM-global version of variable LFPSRFL.
* RFPSLEVG: DM-global version of variable RLEVFPL.
```

10.1.13 YOMFPC.

Contains scientific and technical variables for post-processing. Variables are initialised in setup routine **setup/SUFPC**. All variables are DM-global.

* Technical variables.

```
* CFPDIR  : prefix (path) for the output files (default value is 'PF').
* CFPIDEN : identificator of the output files.
* CFPFMT  : format of the output files, can take the following values:
            * 'MODEL' for output in spherical harmonics.
            * 'GAUSS' for output in grid-point space on Gaussian grid (covering the global sphere).
            * 'LELAM' for output on a grid of kind 'ALADIN' (spectral or grid-point coefficients).
            * 'LALON' for a grid of kind "latitudes * longitudes".
            Default is 'GAUSS' in ARPEGE/IFS, 'LELAM' in ALADIN.
* CFPDOM, C1FPDOM: names of the subdomains. Names have at maximum 7 characters.
            * If CFPFMT='GAUSS' or 'LELAM' only one output domain is allowed.
            * If CFPFMT='LALON' the maximum of output subdomains allowed is 10.
            By default, one output domain is requested, CFPDOM(1)='000' and CFPDOM(i)='' for i>1.
* CFP3DF, C1FP3DF (and (CFP3DFS, C1FP3DFS), (CFP3DFP, C1FP3DFP), (CFP3DFH, C1FP3DFH),
            (CFP3DFT, C1FP3DFT), (CFP3DFV, C1FP3DFV), respectively for post-processing on model
            layers, pressure levels, height levels, potential temperature levels, potential
            vorticity levels): names of the 3D dynamics fields to be post-processed. Names have
            at maximum 12 characters. By default CFP3DF contains blanks (no 3D dynamical field
            to post-process).
```

```
* CFP2DF, C1FP2DF: names of the 2D dynamics fields to be post-processed. Names have at maximum
           16 characters. By default CFP2DF contains blanks (no 2D dynamical field to
           post-process).
* CFPPHY, C1FPPHY: names of the physical fields to be post-processed. Names have at maximum
           16 characters. By default CFPPHY contains blanks (no physical field to post-process).
* CFPCFU, C1FPCFU: names of the cumulated fluxes fields to be post-processed. Names have at
           maximum 16 characters. By default CFPCFU contains blanks (no cumulated fluxes field
           to post-process).
* CFPXFU, C1FPXFU: names of the instantaneous fluxes fields to be post-processed. Names have at
           maximum 16 characters. By default CFPXFU contains blanks (no instantaneous fluxes
           field to post-process).
* Remark: C1FPXXX=CFPXXX(1)//CFPXXX(2)//...//CFPXXX(JPOSXXX) for XXX=DOM,3DF,2DF,PHY,CFU,XFU.
* MFP3DFS, MFP3DFP, MFP3DFH, MFP3DFT, MFP3DFV, respectively for post-processing on
           model layers, pressure levels, height levels, potential temperature levels,
           potential vorticity levels: gribcodes of the 3D dynamics fields to be post-processed
           (variable used at ECMWF only).
* MFP2DF, (resp. MFPPHY): gribcodes of the 2D dynamics (resp. physical)
           fields to be post-processed (variable used at ECMWF only).
* RFP3P (resp. RFP3H, RFP3TH, RFP3PV, NRFP3S): list of post-processing pressure
           (resp. height, potential temperature, potential vorticity, "eta") levels.
* LFPCNT  : control varying output variables according to time step.
* LTRACEFP: trace for FULL-POS (additional prints on listing, default value is .FALSE.).
* NFPGRIB : level of GRIB coding in output file ARPEGE/ALADIN for grid-point arrays:
           0 if no packing at all, 1 if standard GRIB encoding, 2 if modified GRIB encoding
           (default value is 2).
* NFPDOM  : useful dimension of CFPDOM.
* NFP3DF (resp. NFP3DFS, NFP3DFP, NFP3DFH, NFP3DFT, NFP3DFV): useful dimension of CFP3DF
           (resp. for post-processing on model layers, pressure levels, height levels, potential
           temperature levels, potential vorticity levels). Default value is the number of 3D
           wanted dynamical fields.
* NFP2DF (resp. NFPPHY, NFPCFU, NFPXFU): useful dimension of CFP2DF (resp. CFPPHY, CFPCFU, CFPXFU).
           Default value is the number of wanted fields.
* NFP3P (resp. NFP3H, NFP3TH, NFP3V, NFP3S): useful dimension of RFP3P (resp. RFP3H, RFP3TH,
           RFP3PV, NRFP3S).
* NFPXLEV : maximum number of post-processing levels.
* MFP3DYN : maximum number of 3D-dynamical fields needed for LFPART2.
* MFP2DYN : maximum number of 2D-dynamical fields needed for LFPART2.
* NFPDPHY : maximum number of physical fields.
* LFPNHPD : .TRUE. if post-processing of NH pressure departure variable is requested.
* LFPNHVD : .TRUE. if post-processing of NH vertical divergence variable is requested.
* LFPNHVW : .TRUE. if post-processing of NH true vertical velocity is requested.
* LFPLOSP : If .TRUE., fill "ps" array with log(ps),
           fill log(ps) array always with log(ps).
           If .FALSE., fill "ps" array always with "ps",
           fill log(ps) array with log(ps).
* LFPRH100: .TRUE. to convert relative humidity in percent.
```

* Scientific variables.

```
* LFPSPEC : .T. if post-processed dynamical fields are written out as spectral coefficients,
           .F. if post-processed dynamical fields are written out as grid point values (default
           value is .FALSE., except if second part of a pseudo-configuration 927 or E927).
* LFITP, (resp. LFITT, LFITV): .T. post-processed fields on pressure
           (resp. height, potential temperature, potential vorticity, "eta") levels
           should be fitted, .F. otherwise.
* NFPCLI  : usage level for climatology:
           * 0: no climatology
           * 1: orography and land-sea mask of output only
           * 2: all available climatological fields of the current month
           * 3: shifting mean from the climatological fields of the current month to the ones
               of the closest month
```

```
* NFPINDYN (resp. NFPINPHY): type of interpolations for dynamical (resp. physical) fields
           (4 for bilinear, 12 for 12-points); default value is 12.
* LFPQ     : * .TRUE. if specific humidity is interpolated
           * .FALSE. if relative humidity is interpolated
* LASQ     : .T. if apparent surface humidity is set to 80 per cent saturation (default = .F.).
* WSXI     : maximum surface moisture in input.
* WDXI     : maximum deep soil moisture in input.
* WSXO     : maximum surface moisture in output. Default value is -999.
* WDXO     : maximum deep soil moisture in output. Default value is -999.
* FPBL     : critical thickness of PBL. Default value is 17500 Pa.
* RFPCORR : critical orography difference for correcting surface
           temperature through standard profile, in J/kg. Default value is 300*"g".
* RFPCSAB : critical sand percentage difference for computing relative soil moisture in ISBA;
           default value is 0.01.
* RFPCD2  : critical soil depth difference for computing relative soil moisture in ISBA; default
           value is 0.001.
* RFPVCAP : minimum pressure of model level to provide an equatorial cap in the computation of
           variables on constant PV surfaces.
           Default value is 8900 at ECMWF, 15000 at METEO-FRANCE.
* NFPLNPR : 0 for conventional formulation of hydrostatic pressure depth,
           1 for formulation of hydrostatic pressure depth used in the NH model.
* LFPMOIS : month allowed for climatology usage:
           * .F. => month of the model (forecast).
           * .T. => month of the file.
* NFPINCR : sign of the increment when post-processing from ALADIN to ARPEGE.
* NFPLAKE : To overwrite created lakes or islands by specific data:
           *  0 => do not overwrite
           * -1 => overwrite with roughly interpolated data
           * +1 => overwrite with climatology
* NFPCAPE : kind of computation for CAPE and CIN:
           * 1 => from bottom model layer
           * 2 => from the most unstable layer
           * 3 => from mto standard height (2 meters) as recomputed values
           * 4 => from mto standard height (2 meters) out of fluxes (for analysis)
```

* Remarks.

- Note that if you ask for post-processing of dynamical fields which are not spectral in the model, then these post-processed fields will not be spectrally fitted, even if the corresponding switch **LFIT...** is .TRUE. .
- Note that if you wish to post-process upper air dynamical fields on height levels or hybrid levels, it is not possible to apply such spectral fit because the horizontal interpolations are performed before the vertical interpolation in order to respect the displacement of the planetary boundary layer.

* Namelist NAMFPC:
The following variables of **YOMFPC** are in namelist **NAMFPC**:

```
CFPDIR,CFPFMT,CFPDOM,CFP3DF,CFP2DF,CFPPHY,
CFPCFU,CFPXFU,LFPSPEC,RFPVCAP,NFPCAPE,RFP3P,RFP3H,
RFP3TH,RFP3PV,NRFP3S,NFPGRIB,NFPLNPR,RFPCSAB,RFPCD2,
LFPLOSP,LFPRH100,LFITP,LFITT,LFITV,
RFPCORR,NFPCLI,LFPQ,LTRACEFP,LASQ,LFPMOIS,MFP2DF,
MFPPHY,NFP2DF,NFPPHY,NFP3DFS,NFP3DFP,NFP3DFH,NFP3DFT,
NFP3DFV,MFP3DFS,MFP3DFP,MFP3DFH,MFP3DFT,MFP3DFV,NFPINCR,
NFPINDYN,NFPINPHY,NFPLAKE
```

10.1.14 YOMFPCT0.

Variables concerning the control of post-processing. Variables are initialised in setup routine **setup/SUFPC**. No variable in namelist.

∗ List of variables:

```
* NFPCT0: kind of call to CNT0.
         * 1 for standard post-processing;
         * 0 for second (internal) part of pseudo-configuration 927;
         * -1 for last call to CNT0 in the first part of pseudo-configuration 927;
         * -2 for last but one call to CNT0 in the first part of pseudo-configuration 927.
* FPINCR: increment value while bogussing.
```

10.1.15 YOMFPD.

Variables concerning the boundaries and the horizontal dimensions of each output subdomain. Variables are initialised in setup routine **setup/SUFPD**.

∗ For all kinds of output (sub)domains:

```
* NLAT    : number of latitudes for each output (sub)domain;
            corresponds to the variable NDGLG of the output grid(s).
* NLON    : number of longitudes for each output (sub)domain;
            corresponds to the variable NDLON of the output grid(s).
* RLATC   : domain centre in degrees for each output (sub)domain.
* RLONC   : domain centre in degrees for each output (sub)domain.
* RDELX   : resolution in in degrees for each output (sub)domain.
* RDELY   : resolution in in degrees for each output (sub)domain.
* RLATN   : northern latitude in degrees for each output (sub)domain.
* RLATS   : southern latitude in degrees for each output (sub)domain.
* RLONW   : western longitude in degrees for each output (sub)domain.
* RLONE   : eastern longitude in degrees for each output (sub)domain.
* NFPSIZEG: number of points in each complete core of subdomain.
* NFPSIZEL: number of points in each complete extension zone of subdomain.
```

These variables have many possible default values, according to the content of the namelist **NAMFPC**. They are DM-global.

∗ Default values for NLAT, NLON, RLONC, RLATC, RDELX, RDELY: They are initialised in routine **SUFPD**; for more details see the content of this routine.

∗ Additional variables required for a grid ALADIN: If you ask for a grid ALADIN (**CFPFMT**='LELAM'), there are additionnal variables that you can specify (these variables are DM-global):

```
* NFPLUX : actual last row of longitude.
* NFPGUX : actual last row of latitude.
```

There are many possible default values, according to the value of **CFPDOM** in **NAMFPC**; for **NFPLUX** and **NFPGUX** there are two possible default values according to the value of **NFPDOM**, the first one is valid when **NFPDOM**=0, the second one is valid when **NFPDOM**=1 or -1:

CFPDOM	NFPLUX	NFPGUX
'BELG'	72; 61	72; 61
'SLOV'	48; 37	48; 37
'MARO'	160; 149	160; 149
'OPMA'	108; 97	108; 97
'LACE14'	216; 205	192; 181
'LACE12'	240; 229	216; 205
'LACE20'	150; 139	135; 124
'ROUM'	72; 61	72; 61
'FRAN'	200; 189	200; 189
other	180; 169	180; 169

For **CFPFMT**='MODEL','GAUSS' or 'LALON' these variables are not useful, they have default values equal to 0 excepted in the case ALADIN when **CFPFMT**='MODEL' or 'LALON': defaults are **NFPLUX=NDLUXG**, **NFPGUX=NDGUXG**.

* **Namelist NAMFPD:** The following variables of **YOMFPD** are in namelist **NAMFPD**:

```
NLAT,NLON,RLONC,RLATC,RDELX,RDELY,NFPLUX,NFPGUX
```

10.1.16 YOMFPDIM.

```
=== Full-POS DIMENSIONNING ====
* NFPGT1 : number of fields out of vertical post-processing.
* NFPAUXB: number of vertically post-processed fields to remain unfitted (Both 2D and 3D).
* NFPVEC : number of vector fields to be fitted.
* NFPVECG: DM-global number of vector fields to be fitted.
* NFPSCA : number of scalar fields to be fitted.
* NFPSCAG: DM-global number of scalar fields to be fitted.
* NFPUVMN: number of pairs of true (ie : U,V not Vor, Div) wind components to be fitted.
* NFPSPD : number of derived vertically pp. fields to be fitted,
           for one horizontal subdomain.
* NFPSPB : number of derived vertically pp. fields to be fitted,
           for the max. number of horizontal domains.
* NFPGT0B: number of fields to be horizontally post-processed (Both 2D and 3D).
* NFPDYNB: number of post-processed dynamic fields.
* NFPIVEC: number of spectral vector fields to be transformed in gridpoints.
* NFPISCA: number of spectral scalar fields to be transformed in gridpoints.
* NFPIVECG: DM-global number of spectral vector fields to be transformed in gridpoints.
* NFPISCAG: DM-global number of spectral scalar fields to be transformed in gridpoints.
* NAFPOS : number of primitive fields for post-processing on surface-dependent levels.
* NFPBFIN: number of gridpoint fields in interpolation buffer.
* NFPBFOUT: number of gridpoint fields in output gridpoint buffer.

! N.B. : NFPSPD,NFPSPB,NFPSCA,NFPVEC,NFPISCA,NFPIVEC are DM-local per V-set.
```

10.1.17 YOMFPEZO.

Variables related to postprocessing on extension zone of ALADIN output geometry. Variables are initialised in ALADIN setup routine **setup/SUFPEZO**. No variable in namelist.

* List of variables:

```
* NFPEL      : number of gridpoints in DM-local extension zone.
* NFPEX      : maximum number of gridpoints in DM-local extension zone.
* NFPEG      : number of gridpoints in DM-global extension zone.
* NFPSTAE    : start indexes of DM-local extension zones within DM-global extension zone.
* NSTREFP    : number of processors for the DM computations of post-processing extension zone.
* NFPEBLKS   : number of blocks in DM-local extension zone buffer.
* NFPROMEL   : maximum length of a block in DM-local extension zone buffer.
* NEFPSTART  : Start index of points in a working array.
* NEFPEND    : End index of points in a working array.
* EFPGM      : mapping factor over DM-local/DM-global extension zone.
* EFPNORX    : compass (first component) over DM-local/DM-global extension zone.
* EFPNORY    : compass (second component) over DM-local/DM-global extension zone.
```

10.1.18 YOMFPF.

Variables defining the FULL-POS filter. Variables are initialised in setup routine **setup/SUFPF**. All variables are DM-global.

* List of variables:

```
* LFPBED  : .TRUE. to use the old "THX" filter on the homogenous resolution
            space (arpege only) ; else the new gaussian filter is used.
* LFPMAT  : .TRUE. if filtering matrix in the homogenous resolution space must be computed.
* LFPWRFIL: Write out filtering matrixes on files.
```

```
* LFPRDFIL: Read filtering matrixes from files.
* LFPFIL  : .TRUE. if the filter for dilated fields is active (arpege only) for
            the corresponding (sub)domain.
* NFPGLFI : granularity factor for LFI filtering matrixes.
* NFPCMAX : maximum truncation in the homogenous resolution of the input model.
* NFMAX   : maximum truncation of the output subdomains.
            If negative : means no filter at all.
* RFPLTF  : coefficient of the exponential function in the new gaussian filter.
            Also used for the current exponential function in aladin when fields
            are post-processed on a surface to be filtered.
* RFPBED  : coefficient of the exponential function in the old "THX" filter.
* RFPFIL  : Filtering function per wave number in the homogenous resolution spectrum.
* RFPMAT  : Filtering matrix in the homogenous resolution space.
```

Variable **LFPBED** is .FALSE. in the first part of a 927-type configuration **LFPSPEC**=.TRUE. .

Furthermore, in ARPEGE/IFS, it is also possible to perform *at the end* an overtruncation of each post-processed field. This overtruncation is equivalent to the so-called "THX" filter, but it is applied in the stretched spectral space and then n_0 (see formula (10)) is equal to the variable **NPMAX** in the namelist **NAMDIM**. Note that this overtruncation is active only if **NPMAX** is less than **NSMAX**.

∗ Namelist NAMFPF:

The following variables of **YOMFPF** are in namelist **NAMFPF**:

```
NFMAX,LFPBED,RFPBED,LFPWRFIL,LFPRDFIL,NFPGLFI
```

10.1.19 YOMFPG.

Variables defining the characteristics of the (transformed) output geometry. Variables are initialised in setup routine **setup/SUFPG**.

∗ Variables concerning all kinds of output subdomains:

```
* NFPMAX   : truncation order (DM-global).
             * If CFPFMT='GAUSS', NFPMAX is the truncation corresponding to the output grid;
               default value is NSMAX.
             * If CFPFMT='LELAM', NFPMAX is the meridian truncation;
               default value satisfies to the identity 3*NFPMAX+1=NLAT.
             * If CFPFMT='LALON', NFPMAX is the truncation of the Gaussian grid which would
               have been defined by NLAT latitudes and NLON longitudes; default value matches
               the identity 3*NFPMAX+1=min(NLAT,NLON).
* NFPLEV   : number of levels; default is NFLEVG.
* FPVALH   : A coefficients of vertical system (DM-global); default value is VAH.
* FPVBH    : B coefficients of vertical system (DM-global); default value is VBH.
* RFPLA    : latitudes of the output points (DM-local).
* RFPLO    : longitudes of the output points (DM-local).
* RFPGM    : mapping factor (DM-local).
* RFPNORX  : zonal component of rotation matrix for vector outputs (DM-local).
* RFPNORY  : meridian component of rotation matrix for vector outputs (DM-local).
* NFPNUMD  : subdomain index of the output points (DM-local).
* NFPRGPG  : effective number of output points (DM-global).
* NFPRGPL  : DM-local effective number of output points for one processor (DM-local).
* NFPRGPLX : maximum of the NFPRGPL of all the processors (DM-global).
* LFPOSHOR : .TRUE. for actual horizontal post-processing (DM-global).
* LFPEZO   : .TRUE. for actual biperiodicisation (DM-global).
* LFPOSBUF : .TRUE. to set up FULL-POS buffer, .FALSE. for model-shaped buffer (DM-global).
```

∗ Variables concerning Gaussian grid as output subdomain:

```
* FPMUCEN : sine of the latitude of the pole of interest (DM-global);
            default value is RMUCEN for global outputs, 1 for limited area outputs.
* FPLOCEN : longitude of the pole of interest (DM-global);
```

```
            default value is RLOCEN for global outputs, 0 for limited area outputs.
* NFPHTYP (DM-global):
            * 0: regular grid.
            * 1: number of points proportional to sqrt(1-mu**2).
            * 2: number of points read on namelist NAMFPG.
            * default value is NHTYP if NLAT=NDGLG, zero in the other cases.
* FPNLGINC: increment to get non-linear grid (DM-global); default value is 1.
* NFPRGRI : number of active points on a parallel (DM-global); default value is NLOENG if
            NLAT=NDGLG, NLON in the other cases.
* NFPMEN  : maximum zonal wave number for a given parallel (DM-global).
* FPSTRET : stretching factor (DM-global); default value is RSTRET for global outputs,
            1 for limited area outputs.
* NFPTTYP (DM-global):
            * 1: no tilting; pole of high resolution at the northern pole of the real Earth.
            * 2: tilting; pole of high resolution at a different location than the northern
              pole of the real Earth.
            * Default value is NSTTYP for global outputs, 1 for limited area outputs.
* RFPMU   : array containing the sines of the output Gaussian latitudes (DM-global).
```

$*$ **Variables concerning ALADIN grid as output subdomain:** All these variables are DM-global.

```
* NMFPMAX   : meridian truncation order; default value satisfies to the identity
              3*NFPMAX+1=NLAT if CFPFMT='LELAM' and default value is 1 in the other cases.
* LFPMAP    : .T./.F. if the domain is defined through its coordinates/wavelengths.
* FPLX      : wavelength in longitude; default is 5448300. in ARPEGE/IFS if CFPFMT='LELAM',
              0 in ARPEGE/IFS if CFPFMT='MODEL', 'LALON' or 'GAUSS', 0 in ALADIN
              if CFPFMT='GAUSS' and ELX in ALADIN if CFPFMT='MODEL', 'LALON' or 'LELAM'.
* FPLY      : wavelength in latitude; default is 5448300. in ARPEGE/IFS if CFPFMT='LELAM',
              0 in ARPEGE/IFS if CFPFMT='MODEL', 'LALON' or 'GAUSS', 0 in ALADIN
              if CFPFMT='GAUSS' and ELY in ALADIN if CFPFMT='MODEL', 'LALON' or 'LELAM'.
* (FPLON0,FPLAT0): (LA0 geographical longitude, FI0 geographical latitude) of reference
              for the projection, in degrees; default value is 0. in ARPEGE/IFS excepted in
              the case CFPFMT='LELAM' (various default values according to CFPDOM, see routine
              SUFPG1), 0. in ALADIN if CFPFMT='GAUSS' and ELON0 in ALADIN if CFPFMT='MODEL',
              'LALON' or 'LELAM'.
* LFPMRT    : If .T. in Mercator case => use of Rotated/Tilted option.
* FPRPK     : K projection parameter and definition; default value is -9. in ARPEGE/IFS
              excepted in the case CFPFMT='LELAM' (various default values according to CFPDOM,
              see routine SUFPG1), -9. in ALADIN if CFPFMT='GAUSS' and ERPK in ALADIN if
              CFPFMT='MODEL', 'LALON' or 'LELAM'.
```

$*$ **Namelist NAMFPG:** The following variables of **YOMFPG** are in namelist **NAMFPG**:

```
NFPMAX,NFPLEV,FPMUCEN,FPLOCEN,FPVALH,FPVBH,
NFPHTYP,NFPRGRI,FPSTRET,NFPTTYP,NMFPMAX,
FPLON0,FPLAT0,LFPMRT,FPNLGINC
```

10.1.20 YOMFPGIND.

Variables defining some indexes. Variables are initialised in setup routine **setup/SUFPG**. No variable in namelist.

$*$ **List of variables:**

```
* NFPRGPNUM: for processor "jroc", NFPRGPNUM(jroc) is the "NFPRGPL" of the processor "jroc".
* NFPRGPIND: if "iloc" is the DM-local index in a NFPRGPL-array and "jroc" the processor,
             NFPRGPIND(iloc,jroc) gives the DM-global index "iglo" in a NFPRGPG-array.
```

10.1.21 YOMFPIOS.

Contains control variables for FULL-POS file read/write. Variables are initialised in setup routine **setup/SUFPIOS**.

∗ **Writing to output files: variable NFPXFLD.** To write post-processed fields on an output file, you first extract them from an array (or a work file). Rather than extracting the fields one after the other, the fields are grouped in packets, and these packets of fields are extracted one after the other. You can specify the maximum number of fields in a packet by modifying the variable **NFPXFLD** in the namelist **NAMFPIOS**. The default value is the maximum possible depending on the number of fields to produce. Raising the value of **NFPXFLD** should save CPU time to the detriment of the memory cost, and vice versa.

∗ **Additional remarks and some default values:** For the horizontal post-processing, the data flows have been separated in 4 parts:

- post-processed dynamical data.
- post-processed physical and fluxes data.
- weights for horizontal interpolations.
- output geometry and climatology.
- horizontally pre-processed dynamical data (for post-processing on height levels or hybrid levels).

∗ **Namelist NAMFPIOS:** The following variable of **YOMFPIOS** is in namelist **NAMFPIOS**: **NFPXFLD**.

10.1.22 YOMFPOP.

Contains post-processing file-handling variables. Variables are initialised in setup routine **setup/SUFPOPH**. No variable in namelist.

∗ **List of variables:** All variables are DM-global.

```
* CFPFN  : path file name for output files.
* CFPCA  : names of output frames.
* CFPFNBG: path file name for background files while bogussing.
* CFPCABG: names of background frames while bogussing.
```

10.1.23 YOMFPSC2.

Contains control variables for horizontal post-processing.

∗ **List of variables:**

```
* NFPSORT  : Location of points in post-processing buffer.
* NFPSTART : Start index of points in a working array.
* NFPEND   : End index of points in a working array.
* NFPROMA  : Size of working arrays for horizontal post-processing.
* NFPBLOCS : Number of NFPROMA-sized blocs.
```

∗ **Namelist NAMFPSC2:** The following variable of **YOMFPSC2** is in namelist **NAMFPSC2**: **NFPROMA**.

10.1.24 YOMFPSP.

Contains FULL-POS spectral arrays. Variables are initialised after direct spectral transforms. All variables are DM-global. No variable in namelist.

∗ **List of variables:**

```
* SPBFP: derived vertically post-processed fields for all required subdomains.
* UMEAN : Mean wind - first component (for LAM).
* VMEAN : Mean wind - second component (for LAM).
```

10.1.25 YOMGEM.

Some of the variables can be useful for FULL-POS, at least the following ones:

```
* NGPTOT     : Total number of grid columns on a PE
* NGPTOT_CAP : Size of grid points arrays for ALADIN
* NGPTOTMX   : Maximum number of grid columns on any of the PEs
* NGPTOTG    : Total number of grid columns on the Globe
* NGPTOTL    : Total number of grid columns on on eacch PE
```

10.1.26 YOMIOS.

Some variables in **YOMIOS** (which does not contain only FULL-POS variables) can be useful for FULL-POS. These variables are DM-global. For the vertical post-processing, the data flows have been separated in 2 parts: fitted post-processed fields and unfitted post-processed fields.

```
* NPCKFGX: packing factor for read/write on vertically post-processed fields; default is 1.
```

∗ Other useful variables in YOMIOS and NAMIOS:

```
* NPCKFGP: packing factor for read/write on surface fields.
* NPCKFCF: packing factor for read/write on cumulated fluxes.
* NPCKFXF: packing factor for read/write on instantaneous fluxes.
```

Default value of **NPCK...** variables is 1. See routine **SUIOS** for more details.

10.1.27 YOMMP.

Variables specific to the distributed memory environment. For more details see documentation (IDDM) about distributed memory features.

10.1.28 YOMMPG.

Variables specific to the distributed memory environment. DM-global variables mainly used to handle reading and writing of grib data. For more details see documentation (IDDM) about distributed memory features.

10.1.29 YOMOP.

Some variables in **YOMOP** (which does not contain only FULL-POS variables) and namelist **NAMOPH** can be useful for FULL-POS.

```
* LINC   : .T. to get time stamp in hours rather than in time steps
           (default value is .FALSE.). DM-global variable.
* LTIMEP0: .T. to write in the file frame the date of a 0h-range forecast rather than the
           date of an analysis (default value is .TRUE. for configuration 1). DM-global variable.
* NCADFORM: format of *FA* frames in new limited area files (0=old format, 1=new format).
```

10.1.30 YOMPFPB.

Contains buffer for horizontally post-processed auxiliary surface fields. No variable in namelist.

∗ List of variables:

```
* NLENGFPB : Core-zone buffer length
* NELENGFP : Extension-zone buffer length
* NSTAGFPB : Start adresses of NFPROMA-sized blocks in Core-zone buffer
* NESTAGFP : Start adresses of NFPROMEL-sized blocks in Extension-zone buffer
* GFPBUF   : Core-zone buffer for horizontally post-processed auxiliary surface fields
* GEFPBUF  : Extension-zone buffer for horizontally post-processed auxiliary surface fields
```

10.1.31 YOMRFPB.

Contains buffer of auxiliary fields (geometry, climatology) in post processing geometry. No variable in namelist.

* List of variables:

```
* NLENRFPB : Core-zone buffer length
* NELENRFP : Extension-zone buffer length
* NSTARFPB : Start adresses of NFPROMA-sized blocks in Core-zone buffer
* NESTARFP : Start adresses of NFPROMEL-sized blocks in Extension-zone buffer
* RFPBUF   : Core-zone buffer
* REFPBUF  : Extension-zone buffer
```

10.1.32 YOMRFPDS.

Description of auxiliary fields for horizontal post-processing. Variables are initialised in setup routine **setup/SURFPDS**. Variables are DM-global. No variable in namelist.

* List of variables:

```
* MOFPTR : Field pointers (fixed dimension) for climatology fields.
* MFPVSO : internal codes of climatology fields.
* NFPVSO : number of climatology fields.
* NRFPOS : total number of fields in buffer (climatology + geometry).
* MFPGMO : field pointer for output map factor.
* MFPGNXO: field pointer for output X-component of compass.
* MFPGNYO: field pointer for output Y-component of compass.
```

10.1.33 YOMSC2.

Contains parameters used to control vectorisation and memory space. Some of these parameters are used in FULL-POS. No variable in namelist.

- **NSLWIDE**: number of rows the model lagged part runs behind (DM-global).
- **NFPWIDE**: number of rows the FULL-POS lagged part runs behind (DM-global).
- **NDIST**: start address for each row in grid-point calculations (DM-local).
- **NDIEND**: end address for each row in grid-point calculations (DM-local).
- **NCIST**: start address for each row of grid-point field in buffer (DM-local).
- **NCIEND**: end address for each row of grid-point field in buffer (DM-local).

10.1.34 YOMSTA.

The following variable in **YOMSTA** and namelist **NAMSTA** is involved in FULL-POS: **HEXTRAP**. **HEXTRAP** is the height from where the temperature is extrapolated.

10.1.35 YOMVFP.

Contains switches to use FULL-POS in an incremental variational application. Variables are initialised in setup routine **var/SUVAR** (move to directory **setup** will be desirable in the future). Variables are DM-global.

* List of variables:

```
* LSPINT: .T. => if LARCHFP is also .T., follow same data flow but the low resolution and
          analyses will be directly read into the high resolution run and padded with zeros.
          Used at ECMWF only. Default is .F.
```

*** Namelist NAMVFP:** The following variables of **YOMVFP** are in namelist **NAMVFP**: **LARCHFP**, **LSPINT**.

10.1.36 YOMVPOS.

Contains control variables for vertical post-processing. Variables are initialised in setup routine **setup/SUVFPOS** and can be modified by routine **transform/UPDVPOS**. All variables are DM-global. No variable in namelist.

* Control variables.

```
* NXFPLEV  : number of post-processing levels.
* XFPLEV   : values of post-processing levels. Array dimensioned to NXFPLEV.
* QFPTYPE  : derived type variable containing most of control variables.
* NFPOSDYN : number of requested fields for the current scan.
* NFPTRDYN : internal codes of the requested fields for the current scan.
* MYSFSET  : index of my spectral fields set.
```

10.1.37 YOMWFPB.

Contains buffer of working fields (indices, weights) for horizontal post-processing. Variables are initialised in setup routines **setup/SUWFPBUF** for buffer **WFPBUF** and **setup/SUFPIOS** via **setup/SUFPSC2B** for other variables. No variable in namelist.

* List of variables:

```
* NLENWFPBL: length of buffer (DM-local).
* WFPBUF   : buffer containing working fields for horizontal interpolations
             (size of the box, indices, weights, land/sea mask)
             (DM-local).
* NSTAWFPB : start adresses for post-processed packets of points within buffer (DM-local).
```

10.1.38 YOMWFPDS.

Description of working fields for horizontal post-processing. Variables are initialised in setup routine **setup/SUWFPDS**. Variables are DM-global. No variable in namelist.

* Field pointers:

- **MBOX** : for the size of the box.
- **MILA** : pointer for the array containing the nearest northern latitude ($lat1$ of figure 3.2).
- **MILON** : pointer for the array containing the longitude index of the point A_1 of figure 3.2 .
- **MILOS** : pointer for the array containing the longitude index of the point A_2 of figure 3.2 .
- **MILONN**: pointer for the array containing the longitude index of the point A_0 of figure 3.2 .
- **MILOSS**: pointer for the array containing the longitude index of the point A_3 of figure 3.2 .
- **MWSTD** : for the first weight without land-sea mask (12-points interpolations).
- **MWLSM** : for the first weight with land-sea mask for scalars (12-points interpolations).
- **MWSTD4**: for the first weight without land-sea mask (4-points interpolations).
- **MWLSM4**: for the first weight with land-sea mask for scalars (4-points interpolations).
- **MWIC** : field pointer for the nature of the point (lake, island, or else).

* Other variables:

- **NWFPOS**: total number of fields in **WFPBUF**.

10.1.39 Additional namelists, containing local variables.

These namelists are **NAMFPPHY** (read in routine **setup/SUFPPHY**), **NAMFPDY2**, **NAMFPDYP**, **NAMFPDYH**, **NAMFPDYV**, **NAMFPDYT** and **NAMFPDYS** (read in routine **setup/SUFPDYN**). These namelists can be used to make an accurate list of post-processed fields.

In ordinary case, at each post-processing time step, all the fields that are written in the namelist **NAMFPC** are post-processed at all the levels and for all the output domains written in the namelist **NAMFPC**. However, it is possible to get, at each post-processing time step, exactly the fields you wish, and nothing more: in that case, you have to make other namelists file which will contain the selection of the fields you wish to get. First, you have to set in **NAMCT0** the variable **CNPPATH** as the directory where the selection files will be. Under this directory, the name of a selection file must be xxtDDDDHHMM, where DDDDHHMM specifies the date/time of the post-processing time step. Then a selection file must contain 7 new namelists (see above).

∗ **Namelist NAMFPPHY:** Namelist **NAMFPPHY** is for physical fields and fluxes; it contains the following DM-global parameters:

- **CLPHY**: names of selected physical fields; array of 16 characters.
- **CLDPHY**: names of selected subdomains, separated by ":", for each selected physical field; characters array (example: 'DOM1:DOM2:DOM3').
- **CLCFU**: names of selected cumulated fluxes fields; array of 16 characters.
- **CLDCFU**: names of selected subdomains, separated by ":", for each selected cumulated flux; characters array (example : 'DOM1:DOM2:DOM3').
- **CLXFU**: names of selected instantaneous fluxes fields; array of 16 characters.
- **CLDXFU**: names of selected subdomains, separated by ":", for each selected instantaneous flux; characters array (example : 'DOM1:DOM2:DOM3').

∗ **Namelist NAMFPDY2:** Namelist **NAMFPDY2** is for 2D dynamical fields; it contains the following DM-global parameters:

- **CL2DF**: names of selected fields; array of 16 characters.
- **CLD2DF**: names of selected subdomains, separated by ":", for each selected field; characters array (example : 'DOM1:DOM2:DOM3').

∗ **Namelists NAMFPDYP, NAMFPDYH, NAMFPDYV, NAMFPDYT and NAMFPDYS:** These namelists are for fields post-processed respectively on pressure levels, height levels, potential vorticity levels, isentropic levels and hybrid levels; they contain the same following DM-global parameters:

- **CL3DF**: names of selected fields; array of 16 characters.
- **IL3DF**: list of selected indexes of post-processing levels, for each selected field; integer array of 2 dimensions; first subscript is for selected post-processing level; second one is the local field index.
- **CLD3DF**: names of selected subdomains, separated by ":", for each selected level and each selected field; characters array of 2 dimensions; first subscript is the local index of the selected level; second one is the local field index (example : 'DOM1:DOM2:DOM3').

Note that all the fields or levels or domains written in a selection file should be included in the main namelist **NAMFPC**, otherwise the job will abort. Default value of namelist quantities is 0 for **IL3DF** and ' ' for character quantities.

10.2 Modules for type definition.

10.2.1 FULLPOS_TYPES.

See comments inside this module. Defines the type **FULLPOS_TYPES** and contains also some routines working on this type of variables.

10.2.2 TYPE_FPDSPHYS.

See comments inside this module. Defines the type **TYPE_FPDSPHY** and contains also some routines working on this type of variables.

10.2.3 TYPE_FPRQDYNS.

See comments inside this module. Concerns DYN3D and DYN2D post-processable fields. Defines the type **TYPE_FPRQDYN** and contains also some routines working on this type of variables.

10.2.4 TYPE_FPRQPHYS.

See comments inside this module. Concerns PHYSOL, CFU and XFU post-processable fields. Defines the type **TYPE_FPRQPHY** and contains also some routines working on this type of variables.

11 References

11.1 Some internal notes and other ARPEGE notes.

- (TDECDYN) 2004: IFS technical documentation (CY28R1). Part III: dynamics and numerical procedures.
- (TDECTEC) 2004: IFS technical documentation (CY28R1). Part VI: technical and computational procedures.
- Andersson, E., and Ph. Courtier, 1992: Post-processing changes for IFS cycle 9. (Research Department memorandum, internal note).
- (IDFPSG) El Khatib, R., 2002: FULL-POS scientists' guide for ARPEGE/ALADIN cycle ARP24T1/AL15. Internal note (currently uncomplete), 7pp, available on the intranet server "http://intra.cnrm.meteo.fr/gmod/index.html".
- (IDFPTG) El Khatib, R., 2002: FULL-POS technical guide for ARPEGE/ALADIN cycle 25T1. Internal note (currently uncomplete), 39pp, available on the intranet server "http://intra.cnrm.meteo.fr/gmod/index.html".
- (IDFPUG) El Khatib, R., 2002: FULL-POS user's guide for ARPEGE/ALADIN cycle 25T1. Internal note, 67pp, available on the intranet server "http://intra.cnrm.meteo.fr/gmod/index.html".
- Estrade J.-F., 1997: Développement ARPEGE/ALADIN. Mémoire distribuée/ mémoire partagée (internal note in French).
- Janoušek, M., 2001: New port of the new geographical routines to ALADIN/ARPEGE (internal note).
- (NTA30) Rochas, M., et Ph. Courtier, 1992: La méthode spectrale en météorologie. Note de travail ARPEGE numéro **30**, 58pp.
- Undén, P., 1995: IFS post-processing changes proposed by METEO-FRANCE and an improved formulation (internal note).
- (IDBAS) Yessad, K., 2005: Basics about ARPEGE/IFS, ALADIN and AROME in the cycle 30 of ARPEGE/IFS (internal note).
- (IDEUL) Yessad, K., 2005: Integration of the model equations, and Eulerian dynamics, in the cycle 30 of ARPEGE/IFS (internal note).
- (IDSL) Yessad, K., 2005: Semi-Lagrangian computations in the cycle 30 of ARPEGE/IFS (internal note).
- (IDSI) Yessad, K., 2005: Semi-implicit spectral computations in the cycle 30 of ARPEGE/IFS (internal note).
- (IDDH) Yessad, K., 2005: Horizontal diffusion in the cycle 30 of ARPEGE/IFS (internal note).
- (IDTS) Yessad, K., 2005: Spectral transforms in the cycle 30 of ARPEGE/IFS (internal note).
- (IDRD) Yessad, K., 2005: Sphere to sphere transforms in spectral space in the cycle 30 of ARPEGE/IFS: configurations 911 and 912, TRAGEO. (internal note).
- (IDDM) Yessad, K., 2005: Distributed memory features in the cycle 30 of ARPEGE/IFS (internal note).