

FULL-POS IN THE CYCLE 43 OF ARPEGE/IFS.

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

February 23, 2016

Abstract:

This documentation describes the software FULL-POS doing post-processing on different kind of vertical levels. In particular, post-processable variables and organigramme are given. Some aspects of horizontal and vertical interpolators (which may be used in some other applications) are also described.

Résumé:

Cette documentation décrit le logiciel de diagnostics FULL-POS, qui fait du post-traitement sur différents types de niveaux verticaux. On y décrit en particulier la liste des variables post-traitables et l'organigramme. Certains aspects des interpolateurs horizontaux et verticaux (susceptibles d'être utilisés dans d'autres applications) y sont également décrits.

Contents

1	Introduction.	4
1.1	Aim of this documentation.	4
1.2	Modifications since cycle 42.	4
2	Post-processable fields.	5
2.1	3D dynamical fields (DYN3D).	5
2.2	2D dynamical fields (DYN2D).	6
2.3	Surface physical fields (PHYSOL).	7
2.4	Surface cumulated fluxes (CFU).	13
2.5	Surface instantaneous fluxes (XFU).	14
2.6	Surface dataflow for surface variables and post-processing.	15
2.7	Other notations.	15
3	Horizontal interpolations.	16
3.1	Bilinear horizontal interpolations.	16
3.1.1	Horizontal interpolation grid and weights for bi-linear interpolations.	16
3.1.2	Bilinear interpolation.	16
3.2	12 points horizontal interpolations.	16
3.2.1	Horizontal interpolation grid and weights for 12 points cubic interpolations.	16
3.2.2	Horizontal 12 points interpolation.	17
3.2.3	Multilinear interpolations.	17
3.3	Location of computations.	17
3.4	Plane geometry (LAM models).	17
4	Vertical interpolations and extrapolations.	20
4.1	General considerations.	20
4.2	More details for 3D dynamical variables.	20
4.2.1	Wind components, wind velocity.	20
4.2.2	Temperature.	20
4.2.3	Geopotential height.	21
4.2.4	Variables interpolated using routine PP2DINT	22
4.2.5	Post-processable GFL variables (like moisture) and pressure departure variable: variables using routine PPQ	22
4.2.6	Pressure coordinate vertical velocity ω (routine PPVVEL).	22
4.2.7	Moist (irreversible) pseudo-adiabatic potential temperature Θ'_w (routine PPTHWP).	23
4.2.8	Vertical divergence variable (currently d).	23
4.3	2D dynamical variables which need extrapolations.	23
4.3.1	Mean sea level pressure Π_{MSL} (routine PPPMER).	23
5	Filtering in spectral space.	24
5.1	General considerations.	24
5.2	Filtering in the code.	24
6	Post-processing files.	25
7	Organigramme.	26
7.1	FULL-POS via configuration 1.	26
7.1.1	Setup routines and call tree above STEPO_FPOS.	26
7.1.2	General architecture under STEPO_FPOS.	28
7.1.3	Vertical interpolator.	29
7.1.4	Horizontal interpolator.	30
7.2	FULL-POS via configuration 903.	31
7.3	Action and brief description of each routine.	32
7.3.1	Setup routines and call tree above STEPO_FPOS.	32
7.3.2	General architecture under STEPO_FPOS.	35
7.3.3	General architecture under VPOS and ENDVPOS (vertical interpolator).	36
7.3.4	General architecture under FPOSHOR (horizontal interpolator).	37

8 Sequences of calls of post-processing.	38
8.1 Vertical coordinates of post-processing.	38
8.2 Sequences of calls: general considerations.	38
8.3 Different choices for horizontal output.	38
9 Some distributed memory features.	40
9.1 Calculations packets.	40
9.2 Transmission of data necessary for FULL-POS horizontal interpolations from HPOS to FPOSHOR: interpolation buffers.	41
9.3 B-level parallelization.	41
9.4 Case LEQ_REGIONS=T.	42
10 Module and namelist variables to be known.	43
11 References.	45
11.1 Some internal notes and other ARPEGE notes.	45

1 Introduction.

1.1 Aim of this documentation.

This documentation has to aim to describe FULL-POS in the cycle 43 of ARPEGE/IFS.

1.2 Modifications since cycle 42.

- New configuration 903 for off-line FULL-POS (the old way to do off-line FULL-POS still exists).
- Post-processing of surface and mean sea level total pressure.

2 Post-processable fields.

Post-processing can be done on pressure levels, height levels, potential vorticity levels, potential temperature levels, temperature levels, flight 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. They are ordered according to SUAFN2.

2.1 3D dynamical fields (DYN3D).

* Non derived:

- Horizontal wind components U and V ([U] and [V]). It is possible to post-process the geographical wind or the reduced wind.
- Velocity potential χ [KHI] (has to be fitted).
- Stream function ψ [PSI] (has to be fitted).
- Temperature T [T].
- Specific humidity (moisture) q [Q].
- Ozone $O3$ [O3MX].
- Liquid water content q_l [L].
- Ice content q_i [I].
- Total liquid water content for radiation q_{rad} [LRAD].
- Total ice content for radiation q_{irad} [IRAD].
- Cloud fraction q_a [CLF].
- Snow q_s [SN].
- Rain q_r [RR].
- Graupels q_g [GR].
- Hail q_h [HL].
- Turbulent kinetic energy TKE [TKE].
- First variable for EFB turbulent parameterization $EFB1$ [EFB1].
- Second variable for EFB turbulent parameterization $EFB2$ [EFB2].
- Third variable for EFB turbulent parameterization $EFB3$ [EFB3].
- Geopotential height gz [Z].
- Wind velocity [WND].
- Relative humidity HU [HU].
- Potential temperature Θ [TH].
- Moist (irreversible) pseudo-adiabatic potential temperature Θ'_w [THPW].
- Equivalent potential temperature Θ_e [ETH].
- Isobaric equivalent temperature Θ_{vm} taking account of q only for ALARO [IET].
- Simulated reflectivity [SRE].
- Virtual potential temperature Θ_v taking account of q and some other hydrometers [THV].
- Hydrostatic pressure Π [P].
- Pressure departure variable [PD] (for NH only).
- Vertical divergence variable d [VD] (for NH only).
- Montgomery geopotential Φ_{mg} [MG].
- Pressure of iso- T [PTB].
- Altitude of iso- T [HTB].
- Density ρ [RHO].
- QNH [QNH].
- Downdraught mesh fraction q_{dal} [DAL].
- Downdraught vertical velocity q_{dom} [DOM].
- Updraught mesh fraction q_{ual} [UAL].
- Updraught vertical velocity q_{uom} [UOM].
- Prognostic pseudo-historic entrainment q_{uen} [UEN].

- Pseudo-historic convective cloudiness q_{unebh} [UNEBH].
- Total turbulent energy q_{tte} [TTE].
- Prognostic mixing length q_{mxl} [MXL].
- Convective liquid water content q_{lconv} [LCONV].
- Convective ice content q_{iconv} [ICONV].
- Convective rain q_{rconv} [RCONV].
- Convective snow content q_{sconv} [SCONV].
- Shear source term for turbulence q_{shtur} [SHTUR].
- Flux form source term for turbulence: moisture q_{fqtur} [FQTUR].
- Flux form source term for turbulence: enthalpy q_{fstur} [FSTUR].
- Rasch-Kristjansson enthalpy tendency q_{rktth} [RKTH].
- Rasch-Kristjansson water vapour tendency q_{rktqv} [RKTQV].
- Rasch-Kristjansson condensates tendency q_{rktqc} [RKTQC].
- Greenhouse gases GHG [GHG].
- Tracers $TRAC$ [TRAC].
- Chemicals for ECMWF physics $CHEM$ [CHEM].
- Aerosols $AERO$ [AERO].
- Methane loss rate $LRCH4$ [LRCH4].
- ERA40 diagnostic fields $ERA40$ [ERA40].
- Diagnostic fields for NORO GWD scheme $NOGW$ [NOGW].
- Output aerosols $AEROUT$ [AEROUT].
- Output fields from UV processor UVP [UVP].
- Extra GFL variables q_{ext} [EXT].
- Free upper air fields [FUA].
- Eddy diffusivity rate [EDR].

* **Derived:**

- True vertical velocity w [VW].
- Potential vorticity PV [PV].
- Pressure coordinate vertical velocity ω [VV].
- Eta coordinate vertical velocity $\dot{\eta}$ [ETAD].
- Stretching deformation STD [STD].
- Shearing deformation SHD [SHD].
- Absolute vorticity $\zeta + f$ [ABS].
- Relative vorticity ζ [VOR].
- Divergence D [DIV].

2.2 2D dynamical fields (DYN2D).

- Surface (hydrostatic) pressure Π_s [SP].
- Logarithm of surface (hydrostatic) pressure [LNSP].
- Surface total pressure p_s [SPNH].
- Mean sea level (hydrostatic) pressure Π_{MSL} [MSL].
- Mean sea level total pressure p_{MSL} [MSLNH].
- Filtered logarithm of surface pressure diagnostics ([CUF1] to [CUF5]).
- Interpolated (spectral) model orography [FIS].
- Surface NH vertical velocity w_s [WWS].
- Pressure jet [PJET].
- ICAO tropopause temperature [TCAO].
- ICAO tropopause pressure [PCAO].

- Altitude of iso- $\Theta'_w=0$ Celsius [HTPW].
- Altitude of iso- $\Theta'_w=1$ Celsius [HTPW1].
- Mapping factor M [GM].
- Tropopause folding indicator of the iso-2 PVU surface [FOL].
- U and V components of wind at 10 meters (pbl) ([UCLS] and [VCLS]).
- Temperature at 2 meters (pbl) [TCLS].
- Specific humidity at 2 meters (pbl) [QCLS].
- Relative humidity at 2 meters (pbl) [RCLS].
- Wind velocity at 2 meters (pbl) [FCLS].
- Mask extra domain [MSK].
- Maximum temperature at 2 meters [TX].
- Minimum temperature at 2 meters [TN].
- Maximum relative humidity in CLS [HUX].
- Minimum relative humidity in CLS [HUN].
- Maximum simulated reflectivities [SREX].
- CAPE (Convective available potential energy) [CAPE].
- CIEN (Convective inhibition energy) [CIEN].
- Moisture convergence [MOCO].
- Total water vapour content in a vertical column [TVW].
- U and V momentum of gusts out of the model ([UGST] and [VGST]).
- Gusts out of the model [FGST].
- Planetary boundary layer depth [HCLP].
- Forecast surface moisture convergence [SMC].
- Analysed surface moisture convergence [ASMC].
- MSAT7 MVIRI channels 1 and 2 ([MSAT7C1] and [MSAT7C2]).
- MSAT8 MVIRI channels 1 to 8 ([MSAT8C1] to [MSAT8C8]).
- MSAT9 MVIRI channels 1 to 8 ([MSAT9C1] to [MSAT9C8]).
- GOES11 IMAGER channels 1 to 4 ([GOES11C1] to [GOES11C4]).
- GOES12 IMAGER channels 1 to 4 ([GOES12C1] to [GOES12C4]).
- MTSAT1 IMAGER channels 1 to 4 ([MTSAT1C1] to [MTSAT1C4]).
- Lifting condensation level [LCL].
- Free convection level [FCL].
- Equilibrium level [EL].
- Additional surface fields [FSU].
- ICAO jet zonal component of wind [UJET].
- ICAO jet meridian component of wind [VJET].

2.3 Surface physical fields (PHYSOL).

- SD filtered orography [SDFOR] (ECMWF).
- CO2 biospheric flux [CO2NBF] (ECMWF).
- GPP flux adjustment coefficients [CGPP] (ECMWF).
- REC flux adjustment coefficients [CREC] (ECMWF).
- CO2 oceanic flux [CO2OF] (ECMWF).
- CO2 anthropic emission [CO2APF] (ECMWF).
- CO2 biomass burning [CO2FIRE] (ECMWF).
- CH4 surface flux [CH4AG] (ECMWF).
- CH4 fire emission [CH4F] (ECMWF).
- Wind gust at 10 m [10FG] (ECMWF).
- Max wind gust at 10 m during last 3h [10FG3] (ECMWF).

- Max wind gust at 10 m during last 6h [10FG6] (ECMWF).
- Wind gust at 10 m (instantaneous) [I10FG] (ECMWF).
- U-wind at 10 m [10U] (ECMWF).
- V-wind at 10 m [10V] (ECMWF).
- U-wind at 100 m [100U] (ECMWF).
- V-wind at 100 m [100V] (ECMWF).
- Friction velocity [ZUST] (ECMWF).
- Neutral U-wind at 10 m [10NU] (ECMWF).
- Neutral V-wind at 10 m [10NV] (ECMWF).
- Dewpoint at 2 m [2D] (ECMWF).
- Temperature at 2 m [2T] (ECMWF).
- Anisotropy coefficient of orography [ACOT].
- Surface albedo [AL].
- Albedo [ALB] (ECMWF).
- Surface albedo for bare soil [ALS] (MF).
- Surface snow albedo [ALSN] (MF).
- MODIS albedo UVis parallel radiation [ALUVP] (ECMWF).
- MODIS albedo UVis diffuse radiation [ALUVD] (ECMWF).
- MODIS albedo N-IR parallel radiation [ALNIP] (ECMWF).
- MODIS albedo N-IR diffuse radiation [ALNID] (ECMWF).
- Dust emission potential [AERDEP] (ECMWF).
- Lifting Threshold Speed [AERLTS] (ECMWF).
- Soil Clay Content [AERSCC] (ECMWF).
- Black Carbon biogenic flux [BCBF] (ECMWF).
- Black Carbon fossil fuel flux [BCFF] (ECMWF).
- Black Carbon GFED flux [BCGF] (ECMWF).
- Biomass burning injection height [INJF] (ECMWF).
- Organic matter biogenic flux [OMBF] (ECMWF).
- Organic matter fossil fuel flux [OMFF] (ECMWF).
- Organic matter GFED flux [OMGF] (ECMWF).
- Sulphate Low-level emission flux [SO2L] (ECMWF).
- Sulphate High-level emission flux [SO2H] (ECMWF).
- Sulphate GFED flux [SOGF] (ECMWF).
- Volcanic Continuous emission SO2 [VOLC] (ECMWF).
- Volcanic Explosive emission SO2 [VOLE] (ECMWF).
- Secondary organic matter [SOA] (ECMWF).
- Oceanic DMS (dimethylsulfide) [DMSO] (ECMWF).
- Optical depth of total aerosols [ODTO] (ECMWF).
- Optical depth of sea salt aerosols [ODSS] (ECMWF).
- Optical depth of dust aerosols [ODDU] (ECMWF).
- Optical depth of organic matter aerosols [ODOM] (ECMWF).
- Optical depth of black C aerosols [ODBC] (ECMWF).
- Optical depth of sulphate aerosols [ODSU] (ECMWF).
- Optical depth of volcanic flying ash aerosols [ODVFA] (ECMWF).
- Optical depth of volcanic sulphate aerosols [ODVSU] (ECMWF).
- Particulate matter “le 1 um” [AEPM1] (ECMWF).
- Particulate matter “le 2.5 um” [AEPM25] (ECMWF).
- Particulate matter “le 10 um” [AEPM10] (ECMWF).
- Tracers surface flux [TRACFLX] (ECMWF).
- Surface albedo for soil with vegetation [ALV] (MF).

- Angle of surface orography [ANOR] (ECMWF).
- Percentage of clay within soil [ARG] (MF).
- Snow albedo [ASN] (ECMWF).
- Surface albedo for non snowed areas [BAAL] (MF).
- Boundary layer dissipation [BLD] (ECMWF).
- Height of boundary layer [BLH] (ECMWF).
- (Roughness length of bare surface)* g [BSR].
- Budget values [BV].
- Cloud base level [CBASE].
- Convective cloud cover [CCC] (ECMWF).
- Climatological relative deep soil wetness [CDSW].
- Charnock parameter [CHAR] (ECMWF).
- Sea ice cover [CI] (ECMWF).
- Convective precipitation [CP] (ECMWF).
- Convective rainfall rate [CRR] (ECMWF).
- Convective snowfall rate [CSFR] (ECMWF).
- Convective snow fall [CSF] (ECMWF).
- Climatological relative surface soil wetness [CSSW].
- High vegetation cover [CVH] (ECMWF).
- Low vegetation cover [CVL] (ECMWF).
- LAI low vegetation cover [LAIL] (ECMWF).
- LAI high vegetation cover [LAIH] (ECMWF).
- Soil depth [D2] (MF).
- Maximum soil depth [D2MX] (MF).
- Direction of main axis of orography [DPAT].
- Deep soil temperature [DST].
- Deep soil wetness [DSW].
- Water evaporation [E] (ECMWF).
- Emissivity [EMIS].
- Evaporation of snow [ES] (ECMWF).
- U -wind stress [EWSS] (ECMWF).
- Frozen deep soil wetness [FDSW] (MF).
- Frozen surface soil wetness [FSSW] (MF).
- Freezing rain [FZRA] (ECMWF).
- (Output grid-point orography)* g [GFIS].
- Gravity wave dissipation [GWD] (ECMWF).
- High cloud cover [HCC] (ECMWF).
- Resistance to evapotranspiration [HV] (MF).
- Interception content [IC] (MF).
- Interpolated dynamic surface $g * z_0$ [IDZ0] (MF).
- Instantaneous surface moisture flux [IE] (ECMWF).
- Instantaneous surface zonal component of stress [IEWSS].
- Increment to mini relative moisture [INCHN] (ECMWF).
- Increment to maxi relative moisture [INCHX] (ECMWF).
- Increment to mini temperature [INCTN] (ECMWF).
- Increment to maxi temperature [INCTX] (ECMWF).
- Instantaneous surface meridian component of stress [INSSS].
- Anisotropy of surface orography [ISOR] (ECMWF).
- Instantaneous surface heat flux [ISSHF].
- Ice surface first level temperature [ISTL1] (ECMWF).

- Ice surface second level temperature [ISTL2] (ECMWF).
- Ice surface third level temperature [ISTL3] (ECMWF).
- Ice surface fourth level temperature [ISTL4] (ECMWF).
- Interpolated thermal surface $g * z_0$ [ITZ0] (MF).
- Index of vegetation [IVEG] (MF).
- Leaf area index [LAI] (MF).
- Percentage of land [LAN].
- Low cloud cover [LCC] (ECMWF).
- Zonal gravity wave stress [LGWS] (ECMWF).
- Large-scale rainfall rate [LSRR] (ECMWF).
- Large-scale snowfall rate [LSSFR] (ECMWF).
- Large scale snowfall [LSF] (ECMWF).
- Land/sea mask [LSM].
- Large scale precipitation [LSP] (ECMWF).
- Large scale precipitation fraction [LSPF] (ECMWF).
- Instantaneous large scale precipitation fraction [ILSPF] (ECMWF).
- Logarithm of surface roughness [LSRH] (ECMWF).
- Logarithm of z_0 times heat flux [LZ0H] (ECMWF).
- Medium cloud cover [MCC] (ECMWF).
- Meridian gravity wave stress [MGWS] (ECMWF).
- Minimum temperature at 2 m [MN2T] (ECMWF).
- Minimum temperature at 2 m during last 3 hours [MN2T3] (ECMWF).
- Minimum temperature at 2 m during last 6 hours [MN2T6] (ECMWF).
- Minimum total precipitation since last p.p. [MNTPR] (ECMWF).
- Minimum total precipitation since last 3 hours [MNTPR3] (ECMWF).
- Minimum total precipitation since last 6 hours [MNTPR6] (ECMWF).
- Mean sea level pressure [MSLD] (ECMWF).
- Surface pressure [SP] (ECMWF).
- Maximum temperature at 2 m [MX2T] (ECMWF).
- Maximum temperature at 2 m during last 3 hours [MX2T3] (ECMWF).
- Maximum temperature at 2 m during last 6 hours [MX2T6] (ECMWF).
- Maximum total precipitation since last p.p. [MXTPR] (ECMWF).
- Maximum total precipitation since last 3 hours [MXTPR3] (ECMWF).
- Maximum total precipitation since last 6 hours [MXTPR6] (ECMWF).
- V -wind stress [NSSS] (ECMWF).
- Anisotropy vector U -momentum [PADOU].
- Anisotropy vector V -momentum [PADOV].
- Analysed RMS of geopotential height (CANARI) [PCAAG].
- Forecast RMS of geopotential height (CANARI) [PCAPG].
- Potential water evaporation [PEV] (ECMWF).
- Surface relative moisture [PSRHU].
- Precipitation type [PTYPE] (ECMWF).
- Proportion of urbanisation [PURB].
- Maximum proportion of vegetation [PVGMX].
- Interpolated surface temperature [RDST].
- Relaxation deep soil wetness [RDSW].
- Runoff [RO] (ECMWF).
- Surface runoff [SRO] (ECMWF).
- Sub-surface runoff [SSRO] (ECMWF).
- Net ecosystem exchange for CO2 [NEE] (ECMWF).

- Gross primary production for CO2 [GPP] (ECMWF).
- Ecosystem respiration for CO2 [REC] (ECMWF).
- Instantaneous net ecosystem exchange for CO2 [INEE] (ECMWF).
- Instantaneous gross primary production for CO2 [IGPP] (ECMWF).
- Instantaneous ecosystem respiration for CO2 [IREC] (ECMWF).
- Minimum stomatal resistance [RSMIN] (MF).
- Snow density [RSN] (ECMWF).
- Percentage of sand within soil [SAB] (MF).
- Snow depth [SD].
- (Standard deviation of orography)*g [SDOG].
- Snowfall [SF] (ECMWF).
- (Interpolated grid-point orography)*g [SFIS].
- Skin temperature [SKT] (ECMWF).
- Surface latent heat flux [SLHF] (ECMWF).
- Slope of surface orography [SLOR] (ECMWF).
- Snow melt [SMLT] (ECMWF).
- Surface snow density [SNDE] (MF).
- Soil type [SOTY] (ECMWF).
- Lake cover [CLK] (ECMWF).
- Lake depth [DL] (ECMWF).
- Lake mix layer temperature [LMLT] (ECMWF).
- Lake mix layer depth [LMLD] (ECMWF).
- Lake bottom layer temperature [LBLT] (ECMWF).
- Lake total layer temperature [LTLT] (ECMWF).
- Lake shape factor [LSHF] (ECMWF).
- Lake ice temperature [LICT] (ECMWF).
- Lake ice thickness [LICD] (ECMWF).
- (Surface roughness)*g [SR].
- Skin wetness [SRC] (ECMWF).
- Surface sensible heat flux [SSHF] (ECMWF).
- Surface solar radiation [SSR] (ECMWF).
- Surface solar radiation clear sky [SSRC] (ECMWF).
- Downward surface solar radiation [SSRD] (ECMWF).
- Clear sky downward surface solar radiation [SSRDC] (ECMWF).
- Sea surface temperature [SST] (ECMWF).
- Surface soil wetness [SSW].
- Surface temperature [ST].
- Ocean U-current [UCUR].
- Ocean V-current [VCUR].
- Soil first level temperature [STL1] (ECMWF).
- Soil second level temperature [STL2] (ECMWF).
- Soil third level temperature [STL3] (ECMWF).
- Soil fourth level temperature [STL4] (ECMWF).
- Surface thermal radiation [STR] (ECMWF).
- Surface thermal radiation clear sky [STRC] (ECMWF).
- Downward surface thermic radiation [STRD] (ECMWF).
- Clear sky downward surface thermic radiation [STRDC] (ECMWF).
- Sunshine duration [SUND] (ECMWF).
- Soil first level wetness [SWL1] (ECMWF).
- Soil second level wetness [SWL2] (ECMWF).

- Soil third level wetness [SWL3] (ECMWF).
- Soil fourth level wetness [SWL4] (ECMWF).
- Total cloud cover [TCC] (ECMWF).
- Total ozone content in a vertical column [TCO3] (ECMWF).
- Total greenhouse gases content in a vertical column [TCGHG] (ECMWF).
- Total tracers content in a vertical column [TCTRAC] (ECMWF).
- Total chemicals (for ECMWF physics) content in a vertical column [TCCHEM] (ECMWF).
- Total water content in a vertical column [TCW] (ECMWF).
- Total water vapour content in a vertical column [TCWV] (ECMWF).
- Total liquid water content in a vertical column [TCLW] (ECMWF).
- Total ice water content in a vertical column [TCIW] (ECMWF).
- Total rain water content in a vertical column [TCRW] (ECMWF).
- Total snow water content in a vertical column [TCSW] (ECMWF).
- Total supercooled liquid water content in a vertical column [TCSLW] (ECMWF).
- Total precipitation [TP] (ECMWF).
- Temperature of snow layer [TSN] (ECMWF).
- Top solar radiation [TSR] (ECMWF).
- Top solar radiation clear sky [TSRC] (ECMWF).
- Top thermal radiation [TTR] (ECMWF).
- Top thermal radiation clear sky [TTRC] (ECMWF).
- High vegetation type [TVH] (ECMWF).
- Low vegetation type [TVL] (ECMWF).
- Percentage of vegetation [VEG].
- Vertically integrated mass divergence [VIMD] (ECMWF).
- Horizontal visibility [VISIH] (ECMWF).
- Surface roughness [ZOF] (ECMWF).
- (Roughness length for heat)*g [ZOH] (MF).
- (Vegetation roughness length)*g [ZOV] (MF).
- 0 degree level [0DEGL] (ECMWF).
- Convective inhibition [CIN] (ECMWF).
- Convective K-index [KINDEX] (ECMWF).
- Convective TT-index [TTINDEX] (ECMWF).
- Marine aerosols [ASEA] (MF).
- Continental aerosols [ALAN] (MF).
- Carbon (soot) aerosols [ASOO] (MF).
- Desert aerosols [ADES] (MF).
- Sulfate aerosols [ASUL] (MF).
- Volcano aerosols [AVOL] (MF).
- First ozone profile [O3A] (MF).
- Second ozone profile [O3B] (MF).
- Third ozone profile [O3C] (MF).
- Surface photo active radiation [SPAR] (ECMWF).
- Surface UV-B radiation [SUVB] (ECMWF).
- Convective available potential energy [CAPE] (ECMWF).
- Surface clear sky parallel radiation [SPARC] (ECMWF).
- TOA incident solar radiation [STINC] (ECMWF).
- Surface total sky direct shortwave radiation [SFDIR] (ECMWF).
- Surface clear sky direct shortwave radiation [SCDIR] (ECMWF).
- Min. Refractivity Gradient in TPL [DNDZN] (ECMWF).
- Mean Refractivity Gradient in TPL [DNDZA] (ECMWF).

- Duct Base Height [DCTB] (ECMWF).
- Trapping Layer Base Height [TPLB] (ECMWF).
- Trapping Layer Top Height [TPLT] (ECMWF).
- Extra surface fields [VX2] (ECMWF).
- Free surface fields [FSU].

Remarks:

- Some of them are diagnosed at ECMWF via some PHYSOL quantities, and at METEO-FRANCE via some XFU quantities: this is the case of several “VDIAG” surface variables (example: the high, medium and low cloud cover).
- The same quantity can be sometimes found in the topics DYN2D, PHYSOL and XFU (example: the wind-components at 10 m).
- Some of them are used only at ECMWF (example: the boundary layer dissipation).
- Some of them are used only at METEO-FRANCE (example: the A, B, C climatological ozone profiles).
- Some of them have a different generic name at METEO-FRANCE and ECMWF. Example: surface snow albedo ([ALSN] in PHYSOL at METEO-FRANCE, [ASN] in PHYSOL at ECMWF).
- Some of the PHYSOL variables are variables of the surface data flow, but not always with the same generic name. For example for the surface snow albedo one uses the following generic names: [A] in the surface dataflow, [ALSN] in PHYSOL-pp at METEO-FRANCE, [ASN] in PHYSOL-pp at ECMWF.

2.4 Surface cumulated fluxes (CFU).

- Large scale liquid precipitation [CLSP].
- Convective liquid precipitation [CCP].
- Large scale snow fall [CLSS].
- Large scale graupel fall [CLSG].
- Large scale hail fall [CLSH].
- Convective snow fall [CCSF].
- Convective graupel fall [CCSG].
- Convective hail fall [CCSH].
- U -stress [CUSS].
- V -stress [CVSS].
- Surface sensible heat flux [CSSH].
- Surface latent heat flux [CSLH].
- Tendency of surface pressure [CTSP].
- Total cloud cover [CTCC].
- Boundary layer dissipation [CBLD].
- Surface solar radiation [CSSR].
- Surface thermal radiation [CSTR].
- Top solar radiation [CTSR].
- Top thermal radiation [CTTR].
- Convective cloud cover [CCCC].
- High cloud cover [CHCC].
- Medium cloud cover [CMCC].
- Low cloud cover [CLCC].
- U -gravity-wave stress [CUGW].
- V -gravity-wave stress [CVGW].
- U -total stress [CUTO].
- V -total stress [CVTO].
- Water evaporation [CE].
- Snow sublimation [CS].
- Water and snow sublimation [CT].

- Latent heat [CLHE].
- Latent heat sublimation [CLHS].
- Total latent heat [CLHT].
- Soil moisture [CWS].
- Snow mass [CSNS].
- Total precipitable water [CQTO].
- Total ozone [CTO3].
- Top mesospheric enthalpy [CTME].
- Solid specific moisture [CICE].
- Liquid specific moisture [CLI].
- Contribution of convection to U [CCVU].
- Contribution of convection to V [CCVV].
- Contribution of convection to q [CCVQ].
- Contribution of convection to $c_p T$ [CCVS].
- Contribution of turbulence to q [CTUQ].
- Contribution of turbulence to $c_p T$ [CTUS].
- Surface clear sky shortwave radiative flux [CSOC].
- Surface clear sky longwave radiative flux [CTHC].
- Surface parallel solar flux [CSOP].
- Top parallel solar flux [CTOP].
- Surface down solar flux [CSOD].
- Surface down thermic flux [CTHD].
- Surface direct normal irradiance [CDNI].
- Melt snow [CFON].
- Heat flux in soil [CCHS].
- Water flux in soil [CEAS].
- Surface soil runoff [CSRU].
- Deep soil runoff [CDRU].
- Interception soil layer runoff [CIRU].
- Evapotranspiration flux [CETP].
- Transpiration flux [CTP].
- Surface downward moon radiation [CSMR].
- Top clear sky shortwave radiative flux [CTSOC].
- Top clear sky longwave radiative flux [CTTHC].
- Duration of total precipitations [CDUTP].

2.5 Surface instantaneous fluxes (XFU).

- Total cloud cover [XTCC].
- U -component of wind at 10 meters (pbl) [X10U].
- V -component of wind at 10 meters (pbl)[X10V].
- Mean U -component of wind at 10 meters (pbl) [XM10U].
- Mean V -component of wind at 10 meters (pbl) [XM10V].
- Temperature at 2 meters (pbl)[X2T].
- Specific humidity at 2 meters (pbl)[X2SH].
- Relative humidity at 2 meters (pbl)[X2RH].
- Convective cloud cover [XCCC].
- High cloud cover [XHCC].
- Medium cloud cover [XMCC].
- Low cloud cover [XLCC].

- Maximum temperature at 2 meters [XX2T].
- Minimum temperature at 2 meters [XN2T].
- Contribution of convection to U [XCVU].
- Contribution of convection to V [XCVV].
- Contribution of convection to q [XCVQ].
- Contribution of convection to $c_p T$ [XCVS].
- Contribution of turbulence to U [XTUU].
- Contribution of turbulence to V [XTUV].
- Contribution of turbulence to q [XTUQ].
- Contribution of turbulence to $c_p T$ [XTUS].
- Contribution of gravity wave drag to U [XGDU].
- Contribution of gravity wave drag to V [XGDV].
- Large scale precipitation [XLSP].
- Convective precipitation [XCP].
- Large scale snow fall [XLSS].
- Large scale graupel fall [XLSG].
- Large scale hail fall [XLSH].
- Convective snow fall [XCSF].
- Convective graupel fall [XCSG].
- Convective hail fall [XCSH].
- Surface solar radiation [XSSR].
- Surface thermal radiation [XSTR].
- Top solar radiation [XTSR].
- Top thermal radiation [XTTR].
- Wind at 10 m [X10FF].
- CAPE [XCAPE].
- Moisture convergence [XMOCO].
- Height of the PBL out of the model [XCLPH].
- Gusts out of the model [XGUST].
- U -momentum of gusts out of the model [XUGST].
- V -momentum of gusts out of the model [XVGST].
- Maximum relative humidity at 2 meters [XX2HU].
- Minimum relative humidity at 2 meters [XN2HU].
- Θ'_w surface flux [XTHW].
- Hail diagnostic (AROME) [XXDIAGH].

2.6 Surface dataflow for surface variables and post-processing.

Root names for post-processed surface fields may differ from names in `module/surface_fields_mix.F90` (surface dataflow). See appendix 1 for more details.

2.7 Other notations.

- L : number of layers of the model.
- $\left(\frac{dT}{dz}\right)^{\text{st}}$: 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 **SUHOWLMSM**, so splitting between 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:

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

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

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

3.2.2 Horizontal 12 points interpolation.

Let us define:

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

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:
 $X_1 = X_{A_1} + f_1(ZDLO1)(X_{B_1} - X_{A_1}) + f_2(ZDLO1)(X_{C_1} - X_{A_1}) + f_3(ZDLO1)(X_{D_1} - X_{A_1})$.
- a cubic 4 points interpolation on the longitude number 2:
 $X_2 = X_{A_2} + f_1(ZDLO2)(X_{B_2} - X_{A_2}) + f_2(ZDLO2)(X_{C_2} - X_{A_2}) + f_3(ZDLO2)(X_{D_2} - X_{A_2})$.
- 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:
 $X_{interpolated} = X_0 + ZCLA1(X_1 - X_0) + ZCLA2(X_2 - X_0) + ZCLA3(X_3 - X_0)$.

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

3.2.3 Multilinear interpolations.

More diffusive multi-linear interpolations are also available. Combine the above 4 or 12 points interpolations with:

- bilinear interpolations using points (B_0, C_0, B_3, C_3) .
- bilinear interpolations using points (A_1, D_1, A_2, D_2) .
- bilinear interpolations using points (A_0, D_0, A_3, D_3) .

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 (LAM models).

All previous formulae for weight computation can be used for an irregular latitude spacing and a different number of points on each longitude. The LAM 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**.

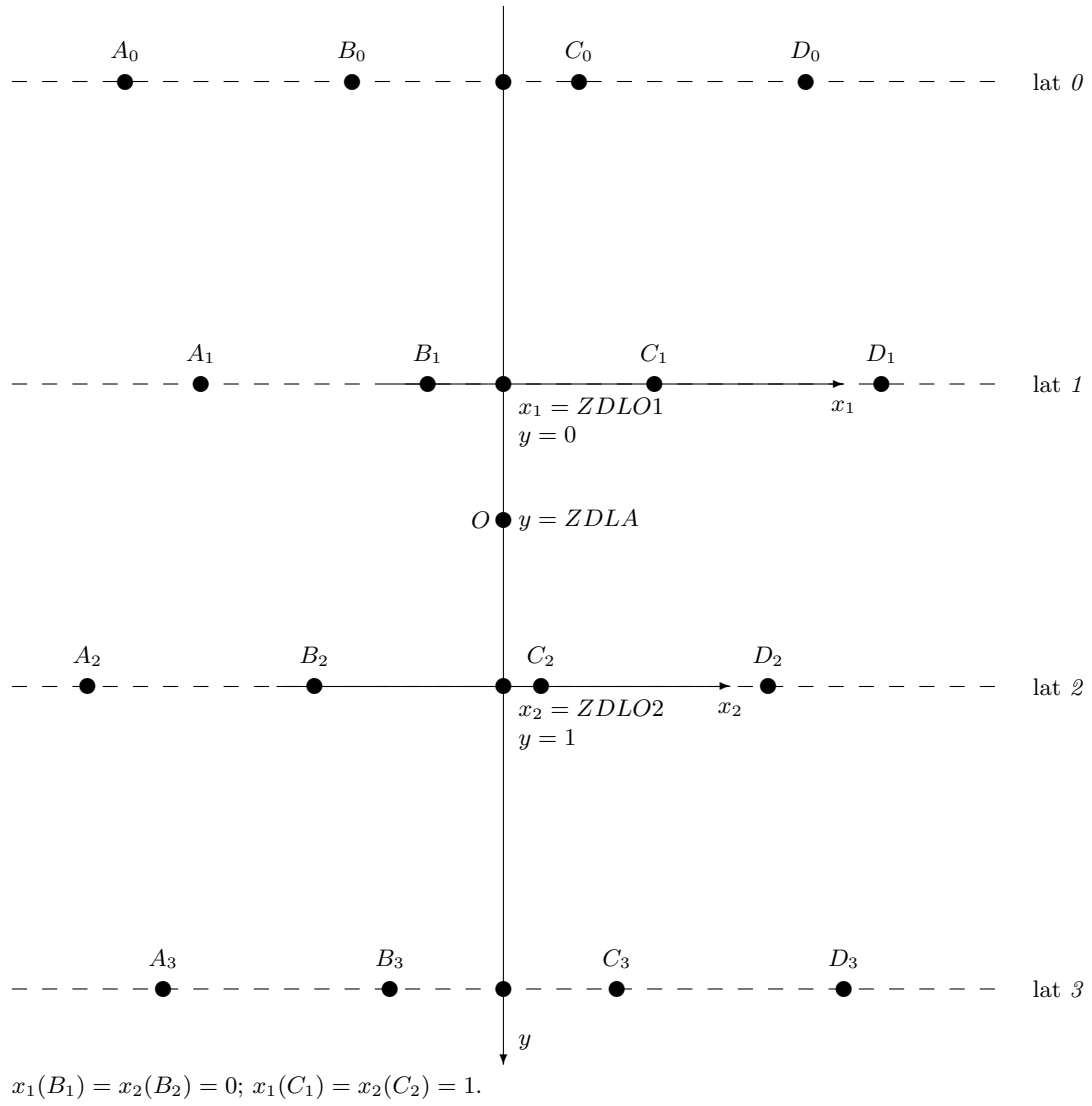


Figure 3.1: Interpolation horizontal grid for bilinear interpolations.

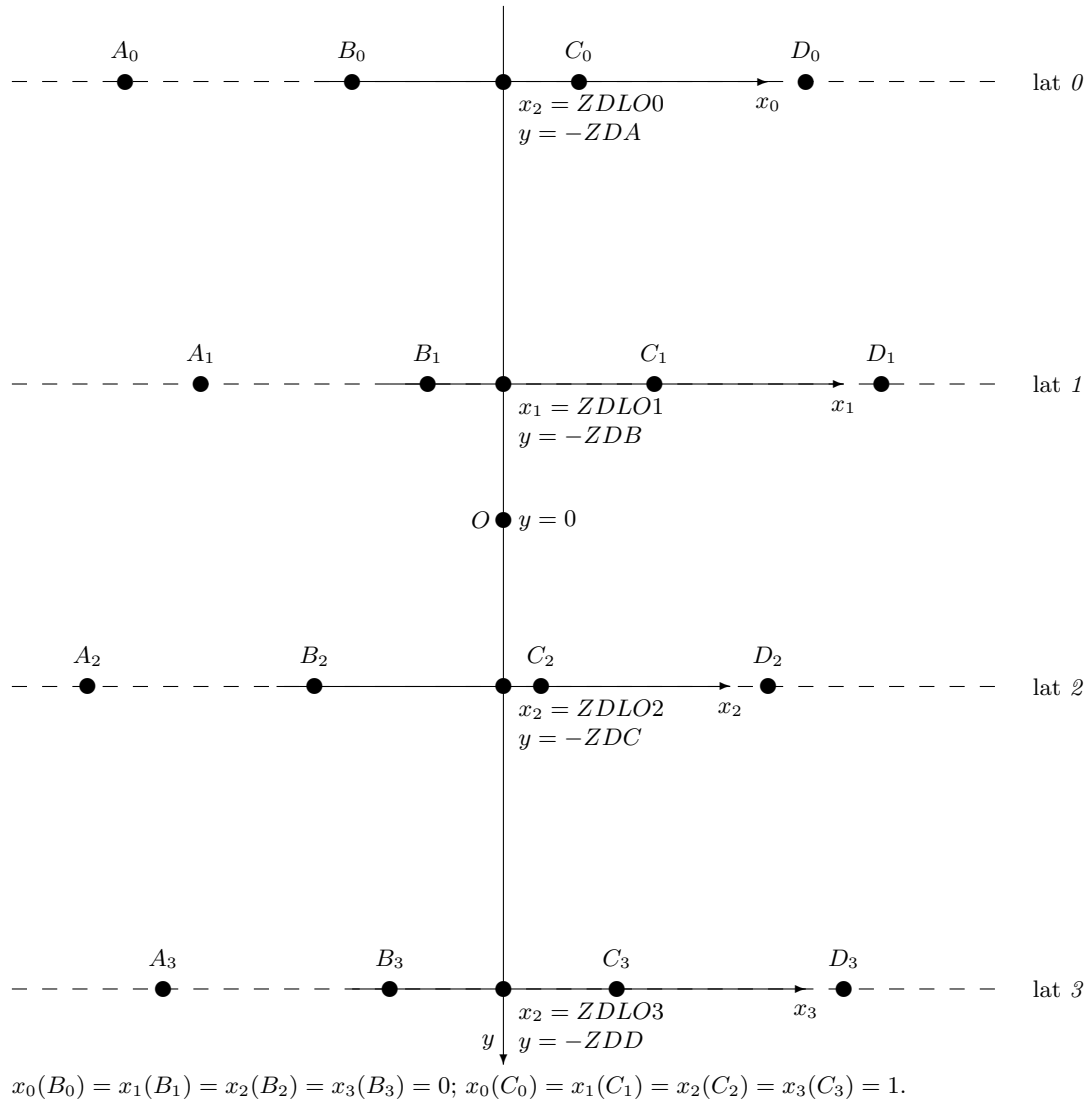


Figure 3.2: Interpolation horizontal grid for 12 points interpolations.

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 (key **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.:****

- 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.:**** 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:

$$T_{\text{surf}} = T_{Le} + \left(\frac{dT}{dz}\right)^{\text{st}} \frac{R_d}{g} \left(\frac{\Pi_s}{\Pi_{Le}} - 1\right) T_{Le} \quad (1)$$

Le is the level number defined by variable **NLEXTRAP**, matching with height **HEXTRAP**.

- 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 done according a more complicated algorithm:

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

where:

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

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

$$\Gamma = \frac{g}{\Phi_s} \max(T'_0 - T_{\text{surf}}, 0) \quad (4)$$

Expression of T'_0 :

– if $\Phi_s/g > 2500$ m:

$$T'_0 = \min\left(T_{\text{surf}} + \left(\frac{dT}{dz}\right)^{\text{st}} \frac{\Phi_s}{g}, 298\text{K}\right) \quad (5)$$

– 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 $\min\left(T_{\text{surf}} + \left(\frac{dT}{dz}\right)^{\text{st}} \frac{\Phi_s}{g}, 298\text{K}\right)$ and $T_{\text{surf}} + \left(\frac{dT}{dz}\right)^{\text{st}} \frac{\Phi_s}{g}$.

* **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 done 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.:**

- 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).

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

where y is defined by formula (3) with $\Gamma = \left(\frac{dT}{dz}\right)^{\text{st}}$ in all cases.

- For more details about this algorithm, see (Andersson and Courtier, 1992) which is still valid for the cycle 43.

* **Way of interpolating if LOLDPP=.F.:**

- 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.
- Virtual potential temperature Θ_v 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 at half levels).

* 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 at half levels).
- 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 Post-processable GFL variables (like moisture) and 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.

This method is also applied to relative humidity.

4.2.6 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.

4.2.7 Moist (irreversible) pseudo-adiabatic potential temperature Θ'_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 Θ'_w at the post-processing levels using a diagnostic (and rather complicated) algorithm.

4.2.8 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 $|\Phi_s|$ 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 $T_0 = T_{\text{surf}} + \left(\frac{dT}{dz}\right)^{\text{st}} \frac{\Phi_s}{g}$.
- To avoid extrapolation of too low pressures over high and warm surfaces the following modifications are done:

- if $T_0 > 290.5$ K and $T_{\text{surf}} \leq 290.5$ K, Γ is defined by:

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

- if $T_0 > 290.5$ K and $T_{\text{surf}} > 290.5$ K, Γ is set to 0, T_{surf} is modified and set to $0.5 \cdot (290.5 \text{ K} + \text{old value of } T_{\text{surf}})$.

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

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

where:

$$x = \frac{\Gamma \Phi_s}{g T_{\text{surf}}} \quad (9)$$

5 Filtering in spectral space.

5.1 General considerations.

- two filters are available for global models: a low-pass filter (also called THX filter because it uses a tanh function) and a bell-shaped filter (also called Gaussian filter). Variable **LFPBED** allows to choose the filter (THX filter if **LFPBED=T**).
- derivatives in stretched global models: require to do a filter which is equivalent to filter on an equivalent unstretched geometry.
- filtering is done in routines **SPOS** (low-pass filter) and **SPOSGF** (Gaussian filter).
- namelist tunable variables are generally in **NAMFPPF**.

* Low-pass filter in global models:

This function looks like a smoothed step function; for a given total wavenumber n the formula is:

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

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 parameter n_0 is stored in variable **NFMAX**.
- tunable parameter k is stored in variable **RFPBED**.

* Bell-shaped filter in global models:

For a given total wavenumber n the formula is:

$$f_{\text{bsARP}}(n) = e^{-4\frac{k}{2}(n/\min(n_0, N_s))^2} \quad (11)$$

where N_s is the model truncation (**NSMAX**) and k a tunable variable stored in variable **RFPBED**.

* Bell-shaped filter in LAM models:

For a given pair of wavenumbers (n, m) , the formula is:

$$f_{\text{bsAL}}(n, m) = e^{-6\frac{k}{2}((n/\min(n_0, N_s))^2 + (m/\min(n_0, N_{\text{ms}}))^2)} \quad (12)$$

where N_s is the model meridian truncation (**NSMAX**), N_{ms} is the model zonal truncation (**NMSMAX**), and k a tunable variable stored in variable **RFPBED**.

5.2 Filtering in the code.

* Fields which can be filtered in computational space:

Filtering is done in routine **SPOS** or **SPOSGF**: multiplication of array **PSPBFP** (containing post-processed data to be filtered) by array **RFPFIL** (in **SPOS**) containing the filtering operator. In **SPOSGF** the filtering operator is hard-coded (no storage in **RFPFIL**).

* Fields which must be filtered in an equivalent not-stretched space:

This case occurs in case of stretched global geometry, to filter some “derivative fields”. It is available for low-pass filter. This filter is active if:

- **LFPFIL=.TRUE.** in namelist **NAMFPPF**.
- 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).

Filtering is done as follows:

- array **PSPBFP** is multiplied by array **RFPMAT** containing the filtering operator.
- **RFPMAT** contains $\mathcal{C} * f * \mathcal{D}$ and is computed in routine **CPFPFILTER**. f denotes the filter; \mathcal{D} and \mathcal{C} are dilatation and contraction matrices (they can be computed by the formerly numbered configuration 911, now in “utilities”, see the corresponding documentation (IDRD); they can also be computed internally in **RDFPFILTER**).
- Remarks about this matricial operator:
 - It is possible to store this matricial operator on a file by setting **LFPWRFIL=.T.** in **NAMFPPF**.
 - This file can be read by setting **LFPRDFIL=.T.** in **NAMFPPF**.

6 Post-processing files.

- Departure file (**CFNISH**= ICMSH[code]INIT) is read on logical unit number **NINISH**=81.
- Departure geometry climatology is read on logical unit **NULCLI**=10 (file Const.Clim).
- Arrival geometry climatology is read on logical unit **NULFP01**=54 (file Const.Clim.DDDDDD).
- For NFPOS=1 or 2 post-processing fields are written on logical unit number **NULFP01**=54 (file **CPFPN**=PF[code]DDDDDD+0000), where DDDDDD is the content of the character variable **CFPDOM** of **YOMFPC**.

7 Organigramme.

Call-tree is very complex and some parts are not described extensively.

7.1 FULL-POS via configuration 1.

7.1.1 Setup routines and call tree above STEPO_FPOS.

* General architecture under CNT0 (only FULL-POS features are mentioned):

```
CNT0 ->
* SUOYOMA ->
  - Most routines called before SUGOMETRY, in particular SUCT0, SUCST
  - SUGOMETRY -> (call tree not detailed)
  - SUFPC -> (call tree not detailed)
  - SUAFN -> SUAFN1, SUAFN2 and SUAFN3 (call tree not detailed)
  - SUFP_CTL
  - SUFPD -> EGGX_N -> (call tree not detailed)
* SUOYOMB ->
  - SUPPVI
  - SUFPG
  - SUPHY -> SUMTS
  - SUCFU -> SUFPCFU
  - SUXFU -> SUFPXFU
  - SUMCUF -> SUFPCUF
  - SUSC2B ->
    * ALLOCATE_GRIDPOINT_BUFFER for allocations
    * SLCSET -> (call tree not detailed)
  - SUBFPOS ->
    * FPINIPHY
    * SUALFPOS -> (call tree not detailed)
    * SUFPTRANS -> (call tree not detailed)
    * SUFPDISTRIB -> SUFPG2, SUEFPG3, SUMPFFPOS, SUMPFFPOS_DEP, SUFPSC2_DEP, SUFPSC2
      (lower call tree not detailed)
    * SUFPWFPDS
    * SUFPWIDE -> SLCSET
    * SUFPF -> UPDTRANS, FPFILTER, RDPFFILTER, CPFPFILTER, WRFPFILTER
      (E)TRANS_INQ, (lower call tree not detailed)
    * SUFPOPH -> SU(E)FRAME, SUFPGRIB, EGGX_N, TRANS_INQ (lower call tree not detailed)
    * SUFPRFPDS -> (call tree not detailed)
    * SUFPTR2 -> (call tree not detailed)
    * SUFPIOS
    * SUFPMAPF -> (call tree not detailed)
    * SUFPRFPBUF_CLIM -> SUMPIOH, DISGRIDFP, RDCLIMO, RDECCLIMO (lower call tree not detailed)
    * SUFPRFPBUF_GEOM
    * SUFPWFPBUF -> SU(E)HOW1, SU(E)HOW2, SU(E)HOX1
* CNT1 -> (see below call tree under CNT1)
```

* General architecture under CNT1 (only FULL-POS features are mentioned):

```
CNT1 ->
* SU1YOM ->
  - SUINIF ->
    * SUSPEC -> (call tree not detailed)
    * SUGRIDF, SUGRIDO and SUGRIDU -> (call tree not detailed)
    * SUGRCFU and SUGRXFU -> (call tree not detailed)
  - SUCT1
* CNT2 -> CNT3 ->
  - CNT4 ->
    * RDSPEC -> (call tree not detailed)
```

- * SU4FPOS -> PPRESET, PPREQ, SUFPPHY and SUFPDYN (lower call tree not detailed)
- * GRIDFPOS -> (see below call tree under GRIDFPOS)
- * GRIDFPOS_SAVEFU -> CPGRIDF
- * PREGPFPOS -> CPGRIDF, PKSURFA -> (lower call tree not detailed)
- * PREDYNFPOS ->
 - WRSPEC (to save fields)
 - PRESFPPOS
 - STEPO (simply to do inverse spectral transforms)
 - RDSPEC (to restore fields)
- * DYNFPOS -> (see below call tree under DYNFPOS)
- * FPWRNCF
- SU4FPOS -> PPRESET, PPREQ, SUFPPHY and SUFPDYN (lower call tree not detailed)

*** General architecture under GRIDFPOS:**

- GRIDFPOS ->
 - * GRIDFPOSSFY_INIT
 - * FPMODCFU (CFU) and FPMODXFU (XFU) -> (call tree not detailed)
 - * MAXGPFV -> (call tree not detailed)
 - * SUFPSUW ->
 - CPCLIMI
 - * HPOS -> (see below call tree under HPOS)
 - * processor communication routines SLCOMM and SLEXPOL
 - * FPOSHOR -> (see below call tree under FPOSHOR)
 - * FPTRATOD and FPTRDTOA (transposition)
 - * FPOSHORLAG -> (see below call tree under FPOSHORLAG)
 - FPSCAW, SU(E)HOW1, SU(E)HOWLSM, SUFPCIP -> (lower call tree not detailed)
 - * SCAN2M_HPOS -> (see below call tree under STEPO_FPOS -> SCAN2M_HPOS)
 - * FPOSHOR -> (see below call tree under FPOSHOR)
 - * FPTRDTOA (transposition)
 - * FPOSHORLAG -> (see below call tree under FPOSHORLAG)
 - * FPEZO2H -> (call tree not detailed)
 - * SUFPOROG -> (call tree not detailed)
 - * FP2SX1 -> (call tree not detailed)
 - * WRHFP (ARPEGE files), WRMLFP (GRIB files) -> (call tree not detailed)

*** General architecture under DYNFPOS:**

- DYNFPOS ->
 - * SUFPCONF
 - * SUVFPOS ->
 - SUFPFIT
 - UPDVPOS -> SUVPOS
 - * OPDIS
 - * STEPO_FPOS -> (see below call tree under STEPO_FPOS)

7.1.2 General architecture under STEPO_FPOS.

Lower routines appearing in this call-tree may call routines not detailed there. In particular, complex call-tree under **WRHFP** and **WRSFP** is not described.

```
STEPO_FPOS ->
* Vertical interpolations:
  - CPVPOSPR -> SUFPFIT
  - SUFPFIT
  - SCAN2M_VPOS ->
    * VPOS_PREP
    * (E)SPEREE
    * VPOS -> (see below call tree under VPOS)
    * EBIPOS
* Model direct transforms:
  - TRANSDIR_FP
  - FPTSA_DIR
* Spectral calculations (filters):
  - SPOSGF
  - SPOS
* Model inverse transforms:
  - FPTSA_INV
  - TRANSINV_FP
* Horizontal interpolations:
  - SCAN2M_HPOS ->
    * HPOS -> ACSOLW, CVLANISO, SC2RDG
    * SLCOMM and (E)SLEXPOL
  - SCAN2M_MPOS ->
    * VPOS_PREP
    * (E)SPEREE
    * VPOS -> (see below call tree under VPOS)
    * HPOS -> ACSOLW, CVLANISO, SC2RDG
    * SLCOMM and (E)SLEXPOL
    * EBIPOS
  - FPOSHOR -> (see below call tree under FPOSHOR)
  - FPTRDTOA (transposition)
  - FPOSHORLAG -> (see below call tree under FPOSHORLAG)
  - FPEZ02H
* Lagged vertical interpolations:
  - SUVFPOS ->
    * SUFPFIT
    * UPDVPOS -> SUVPOS
  - CPVPOSPR -> SUFPFIT
  - SUFPFIT
  - FPSPECFITG
  - ENDVPOS -> (see below call tree under ENDVPOS)
  - FPEZ02H
* Output direct transforms:
  - TRANSDIR_FP
  - FPTSA_DIR
* Spectral calculations (filters) in output space:
  - SPOSGF
  - SPOS
* Output inverse transforms:
  - FPTSA_INV
  - TRANSINV_FP
* IO:
  - SUVFPOSL
  - IOFPOS ->
    * ESPFP
    * WRSFP and WRHFP (ARPEGE files)
    * WRPLFP and WRMLFP (GRIB files)
```

7.1.3 Vertical interpolator.

* General architecture under VPOS and POS:

```
VPOS ->
- GPHPRE
- CTSTAR
- SC2RDG
- POS ->
  * pos.F90/SUPTRPPGFL_POS
  * POS_PREPGFL
  * Several adiabatic GP... and GNH... routines (in part 2.1)
  * Additional adiabatic GP... routines, and PPCVIRT (in part 2.2)
  * CTSTAR
  * FPPS -> PPPMER
  * A series of routines computing intermediate quantities and weights for vertical interpolations.
    - PPLTP
    - PPLTETA
    - PPLETA -> several adiabatic GP... routines
    - PPINIT
    - PPFLEV
    - PPSTA
  * A series of vertical interpolation routines.
    - PPUV -> PPINTP and PPITPQ
    - PPT ->
      * PPT_OLD -> PPINTP
      * PPSTA
      * PPINTP
    - PPQ -> PPINTP
    - PPGEOP -> PPSTA, GPGE0, PPITPQ and PPINTP.
    - PP2DINT
    - PPVVEL -> PPINTP
    - PPPMER
    - PPLTP
    - PPTHWP
    - POAERO -> TJQUD, TJCUBI and TJQUAA
    - PPLTEMP
    - FPPS -> PPPMER
    - PPWETPOINT
  * Storage of post-processed fields.
- PHYMFPOS ->
  * adiabatic GP... and GNH... routines.
  * FPACHMT -> ACSOLW and ACHMT (call tree not detailed)
  * FPCICA -> FPCINCAPE
  * phymfpos.F90/STORE_DATA
  * MTS_PHYS (call tree not detailed)
- SC2WRG
```

* General architecture under ENDVPOS:

```
ENDVPOS ->
- ENDPOS ->
  * endpos.F90/SUPTRPPGFL_ENDPOS
  * ENDPOS_PREPGFL
  * Several adiabatic GP... routines (mostly in part 1.3).
  * CTSTAR
  * FPPS -> PPPMER
  * PPLETA -> several adiabatic GP... routines
  * APACHE -> (see below call tree under APACHE)
  * FPACHMT -> ACSOLW and ACHMT (call tree not detailed)
```

- * FPCICA -> FPCINCAPE
- * PPLTEMP
- * PPWETPOINT
- * PPTHWP
- * PPCVIRT
- * POAERO -> TJQUD, TJCUBI and TJQUAA
- * endpos.F90/STORE_DATA

*** General architecture under APACHE:**

APACHE ->

- * Several adiabatic GP... routines.
- * PPINIT
- * FPVIEW
- * Several adiabatic GP... routines.
- * PPFLEV
- * PPSTA
- * PPUV -> PPINTP and PPITPQ
- * PPT ->
 - PPT_OLD -> PPINTP
 - PPSTA
 - PPINTP
- * PPQ -> PPINTP
- * PP2DINT
- * PPGEOP -> PPSTA, GPGEOP, PPITPQ and PPINTP.

7.1.4 Horizontal interpolator.

*** Organigramme of FPOSHOR:**

FPOSHOR ->

- * FPCLIPHY, FPNILPHY (to compute LLCLI, LLNIL)
- * FPSCAW
- * FPSCAX
- * FPINTDYN -> FPINT12 and FPINT4.
- * FPINTPHY -> FPINT12, FPINT4, FPINT4X, FPAVG, FPNEAR.
- * FPSAMPL -> FPINT12 and FPINT4.

*** Organigramme of FPOSHORLAG:**

FPOSHORLAG ->

- * FPCORDYN -> FPHOR12.
- * FPGEOP
- * FPCLIPHY -> CVLANISO.
- * FPNILPHY -> CVLANISO.
- * FPCORPHY -> (call tree not detailed)

7.2 FULL-POS via configuration 903.

* General architecture under CNT0 (only FULL-POS features are mentioned):

CNT0 ->

- * SUOYOMA -> (see above)
- * SUOYOMB -> (see above)
- * CPREP3 -> (see below call tree under CPREP3)

* General architecture under CPREP3:

CPREP3 ->

- * SUCT1
- * CNT3_WAIT -> (call tree not detailed)
- * SU4FPOS -> PPRESET, PPREQ, SUFPPHY and SUFPDYN (lower call tree not detailed)
- * FILEDATE -> (call tree not detailed)
- * SUCST
- * SUFPINIF -> (call tree similar to SUINIF one)
- * SUFPRFPBUF_CLIM -> SUMP10H, DISGRIDFP, RDCLIMO, RDECCLIMO (lower call tree not detailed)
- * GRIDFPOS -> (see above call tree under GRIDFPOS)
- * DYNFPOS -> (see above call tree under DYNFPOS)
- * GRIBIOFLUSH, if .NOT.LARPEGEF -> (call tree not detailed)
- * SIGPOST -> FPWRNCF
- * IO_SERV_SYNC -> (call tree not detailed)
- * LOGDIS

7.3 Action and brief description of each routine.

- Expression “DM-local” for a quantity means “local to the couple of processors (*proca,procb*)”: each processor has its own value for the quantity. Expression “DM-local computations” means that the computations are done independently in each processor on “DM-local” quantities, leading to results internal to each processor, which can be different from a processor to another one.
- Expression “DM-global” for a quantity means that it has a unique value available in all the processors. Expression “DM-global computations” means that the computations are either done in one processor, then the results are dispatched in all the processors, or the same computations are done in all the processors, leading to the same results in all the processors.
- In a routine description the mention “For distributed memory computations are DM-local” means that all calculations done by this routine are DM-local; the mention “For distributed memory computations are DM-global” means that all calculations done by this routine are DM-global; when no information is provided it means that a part of calculations are DM-local and the other part is DM-global.
- Expression “main” processor currently refers to the processor number 1: (*proca,procb*)=(1,1).

7.3.1 Setup routines and call tree above STEPO_FPOS.

* **General architecture under CNT0:**

- **CNT0:** controls integration job at level 0.
- **SU0YOMA:** 0-level interface routine for set-up: first part.
- **SU0YOMB:** 0-level interface routine for set-up: second part.
- **SUCT0:** routine to initialize level 0 control module.
- **SUCST:** routine to initialize universal and astronomical constants.
- **SUGEOMETRY:** routine to initialize model geometry.
- **SUA FN:** initialises ARPEGE field descriptors. Reads namelist **NAMAFN**, initialises **YOMAFN**.
- **SUA FN1** and **SUA FN2:** initialises ARPEGE field descriptors: part of calculations done before (in **SUA FN1**) and after (in **SUA FN2**) the namelist reading.
- **SUA FN3:** prints ARPEGE field descriptors.
- **SUFPC:** initialises FULL-POS post-processing scientific and technical options; in particular initialise some dimensioning quantities. Reads namelist **NAMFPC**, initialises **YOMFPC**.
- **SUFPC_TL:** control FULL-POS processing options with respect to the model setup.
- **SUFPCD:** initialise FULL-POS horizontal subdomains. Computes total number of output points. Reads namelist **NAMFPCD**, initialises **YOMFPCD**.
- **EGGX_N:** geography of grid-points ARPEGE-ALADIN.
- **SUPPVI:** initialize variables used in the vertical interpolator.
- **SUFPCG:** initialise output geometry (some **YOMFPCG** quantities).
- **SUSC2B:** initialises some control variables (interpolation buffers quantities, some distributed memory quantities).
- **SUPHY:** header setup routine for the different packages of physics.
- **SUMTS:** setup routine for TOVS radiative transfer code.
- **SUCFU:** initialises the control of cumulated fluxes.
- **SUFPCFU:** initialises cumulated fluxes keys for FULL-POS.
- **SUXFU:** initialises the control of instantaneous fluxes.
- **SUFPCXFU:** initialises instantaneous fluxes switches for FULL-POS.
- **SUMCUF:** set-up for CUF diagnostics.
- **SUFPCUF:** set-up for post-processing of CUF diagnostics.
- **SUBFPOS:** general set-up routine for FULL-POS called under **SU0YOMB**.

* **General architecture under SUBFPOS:**

- **CPFPFILTER:** filtering for stretched models:
 - reads or computes dilatation/contraction matrices.
 - computes filtering matrices; stores them in **RFPMAT**.
- **DISGRIDFP:** memory distribution routine in grid-point space for FULL-POS fields.
- **FPFILTER:** fills **RFPFIL** (filtering operator).
- **FPINIPHY:** controls that the pointers of physical fields to be post-processed are defined.
- **RDFPFILTER:** filtering for stretched models; reads filtering matrices and fills **RFPMAT** (filtering operator).
- **SLCSET:** set-up for quantities involved in the halo necessary for horizontal interpolations.
- **SUALFPOS:** memory allocation for FULL-POS arrays.
- **SUFPG2:** initialise output geometry (some **YOMFPD** and **YOMFPG** quantities).
- **SUEFPG3:** initialise specific output geometry quantities for LAM models.
- **SUFPTTRANS:** initialize the transform package for each output resolution, fills **YOMFPTRANS**.
- **SUFPDISTRIB:** initialize the post-processing horizontal distribution in the departure and the arrival geometries; initialize the array slicing for vector or memory cache optimization.
- **SUFPWFPDS:** initialises the FULL-POS working fields descriptors needed for horizontal post-processing, initialises **YOMWFPDS**.
- **SUFPWIDE:** initialises the control variables depending on the size of the halo (horizontal interpolations): interpolation buffers quantities, some distributed memory quantities.
- **SUFPPF:** initialises the profile of the spectral post-processing filter. Reads namelist **NAMFPF** and initialises **YOMFPF**.
- **SUFPOPH:** initialise post-processing file-handling parameters (**YOMFPOP**).
- **SUFPRFPDS:** initialises the descriptors of the auxiliary fields needed for horizontal post-processing, initialises **YOMRFPDS**.
- **SUFPTR2:** computes the number of fields **NFPVT0** and the associated pointers for the auxiliary grid-point quantities.
- **SUFPIOS:** set-up for using work files on post-processing buffers. To set-up **YOMFPIOS**. Opens work files.
- **SUFPMAPF:** setup map factor **RFPGM** for LAM geometries.
- **SUFPRFPBUF_CLIM**, **RDCLIMO** and **RDECCLIMO:** 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.
- **SUFPRFPBUF_GEOM:** fills part of **RFPBUF** which contains geometry.
- **SUFPWFPBUF:** computes the weights **WSTD04** and **WSTD12** for interpolations.
- **SUFPSC2** and **SUFPSC2_DEP:** reads namelist **NAMFPSC2** and **NAMFPSC2_DEP** and sets-up **YOMFPSC2** which contains the control parameters and arrays to scan the post-processing buffers per row. When **NFPDISTRIB** ≥ 1 only **NAMFPSC2_DEP** is read.
- **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.
- **SUHOW1** (**SUEHOW1** for LAM models): initialises the coordinates of the nearest model grid point of the FULL-POS interpolation points.
- **SUHOW2** (**SUEHOW2** for LAM models): computes weights without land-sea mask and surface temperature for horizontal interpolations used in FULL-POS.
- **SUHOX1** (**SUEHOX1** for LAM models): cf. **SUHOW1** in a more optimised way.
- **SUMPFPOS:** initialises some FULL-POS quantities and arrays involved in distributed memory linked to arrival geometry.
- **SUMPFPOS_DEP:** initialises some FULL-POS quantities and arrays involved in distributed memory linked to departure geometry.
- **TRANS_INQ:** fetches quantities computed in the transform package.
- **WRFPFILTER:** filtering for stretched models; write **RFPMAT** on a file.

* **General architecture under CNT1:**

- **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.
- **SU1YOM:** 1-level interface routine for set-up.
- **SUINIF:** interface routine for reading the departure files.
- **SUSPEC:** interface for reading the spectral fields of the departure files (ARPEGE files).
- **SUGRIDF:** interface for reading the surface grid point fields of the departure files (ARPEGE files).
- **SUGRIDO:** interface for reading the ocean mixed layer model grid point fields of the departure files.
- **SUGRIDU:** interface for reading the upper air grid point fields of the departure files.
- **SUGRCFU:** reads the cumulated fluxes on ARPEGE files.
- **SUGRXFU:** reads the instantaneous fluxes on ARPEGE files.
- **RDSPEC:** memory transfer of spectral data.
- **SU4FPOS:** 4-level interface routine for set-up of FULL-POS features. Initialises blocks **YOM4FPOS** and **PTRFP4**.
- **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).
- **SUFPPHY:** initialise the requests of post-processing at a given time-step. Reads namelist **NAMFPPHY**.
- **SUFPDYN:** initialise dynamical requests of post-processing at a given time-step. Reads namelists **NAMFPDY...**
- **PREDYNFPOS:** to save and and pre-condition input data before post-processing.
- **WRSPEC:** memory transfer of spectral data.
- **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.
- **FPWRNCF:** to write FULL-POS status.

* **General architecture under CPREP3:**

- **SUCT1:** sets-up **YOMCT1**.
- **FILEDATE:** extracts the date of a file.
- **SUFPINIF:** interface routine for reading the departure files (specific version of **SUINIF** for conf 903).
- **GRIBIOFLUSH:** flush GRIB IOs at the end of a step.
- **SIGPOST:** post events to signal completion of IOs.
- **IO_SERV_SYNC:** send a synchronous signal to the IO server.
- **LOGDIS:** displays the current state of the model execution.

* **General architecture under GRIDFPOS:**

- **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.
- **CPGRIDF:** copies a grid point buffer/work file on another buffer/work file.
- **PKGRIDA:** packs model grid point variables by writing to then reading from an ARPEGE file.
- **PKSURFA:** packs model surface fields by writing to, then reading from an ARPEGE file.
- **SC2RDG:** memory transfer routine for **NPROMA**-arrays; reads data in a buffer and copies it in a grid-point array.
- **SC2WRG:** does the inverse operation of routine **SC2RDG**.

- **FPMODCFU** (resp. **FPMODXFU**): preliminary modification of cumulated fluxes (resp. instantaneous fluxes) for post-processing: prepares input fields before horizontal interpolations.
- **MAXGPFV**: computes maximum value of a grid-point field.
- **SUFPSUW**: computes weights for horizontal interpolations of surface fields.
- **CPCLIMI**: fills interpolation buffers with some surface fields (for example land-sea mask) previous to compute weights with land-sea mask.
- **FPSCAW**: for horizontal interpolator, computes interpolation grid and some indexes to be used with interpolation buffer.
- **FPSCAX**: cf. **FPSCAW**, in a more optimised way.
- **FPTRATOD**: transposition routine between arrival geometry DM-distribution and departure geometry DM-distribution.
- **FPTRDTPA**: does the inverse operation of routine **FPTRATOD**.
- **SUHOWLSM** (**SUEHOWLSM** for LAM models): cf. **SU(E)HOW2** but weights take account of land-sea mask or surface temperature.
- **SUFPCIP**: setup localization of created isolated points (lake or island) in the horizontal interpolation step.
- **FPEZO2H**: post-processing of extension zone in LAM models.
- **SUFPOROG**: set-up output orography.
- **FP2SX1**: FULL-POS interface for SURFEX.

* **General architecture under DYNFPOS:**

- **DYNFPOS**: interface routine managing vertical then horizontal interpolations for post-processing on dynamical 3D and 2D fields, calls several sequences of **STEPO_FPOS**.
- **SUFPCONF**: initialise FULL-POS configuration letters for **STEPO_FPOS**.
- **SUVFPOS**: initialises and sorts the list of the dynamical fields to post-process for a given level type of post-processing.
- **UPDVPOS**: updates **YOMVPOS** which contains control variables for vertical post-processing.
- **SUVPOS**: initialises **YOMVPOS** which contains control variables for vertical post-processing.
- **OPDIS**: prepares a text array containing relevant information about the current state of model execution.

7.3.2 General architecture under STEPO_FPOS.

- **STEPO_FPOS**: controls FULL-POS job at lowest level.
- **CPVPOSPR**: computes field pointers for vertical post-processing.
- **SUFPFIT**: for a given post-processing step: define whether fit is active or not.
- **VPOS_PREP**: computes input quantities for **VPOS**.
- **EBIPOS**: bi-periodicization of vertically post-processed fields in LAM models.
- **TRANSDIR_FP**: direct spectral transforms for post-processed quantities.
- **TRANSINV_FP**: inverse spectral transforms for post-processed quantities.
- **FPTSA_DIR**: memory transfer in spectral space between arrays coming from spectral transforms and arrays entering spectral calculations.
- **FPTSA_INV**: does the inverse operation of routine **FPTSA_DIR**.
- **SPOS**: filters some spectral quantities in spectral space; in particular does filtering requiring array **RFPMAT**.
- **SPOSGF**: filters some spectral quantities in spectral space by a Gaussian filter.
- **SCAN2M_VPOS**: post-processing in grid-point space: control routine for unlagged vertical interpolations.
- **SCAN2M_HPOS**: post-processing in grid-point space: control routine for horizontal interpolations.

- **SCAN2M_MPOS**: post-processing in grid-point space: control routine for both vertical and horizontal interpolations.
- **HPOS**: interface routine managing non lagged part of horizontal interpolations. Fills buffer for quantities to be interpolated (array **PPBBUF1**).
- **ACSOLW**: computation of some surface coefficients and variables.
- **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).
- **SLCOMM** does processor communication to fill the halo required for horizontal interpolations.
- **SLEXTPOL** adds data of extra-polar latitudes in the halo when necessary.
- **SUVFPOSL**: initialises **YOMFP4L** which contains the working variables and arrays to write out the dynamical post-processed fields.
- **IOFPOS**: manages file writing for post-processed quantities.
- **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.
- **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.
- **ESPFP**: LAM model routine writing vertical post-processed data on a FA file, equivalent of **WRSFP**.
- **WRMLFP**: GRIB routine writing data on model layers.
- **WRPLFP**: GRIB routine writing data on pressure layers, potential vorticity layers, potential temperature layers.
- **FPSPECFITG**: spectral fit of some model fields, for ECMWF configurations.

7.3.3 General architecture under VPOS and ENDVPOS (vertical interpolator).

- **VPOS**: interface routine managing all vertical interpolations which can be done before horizontal interpolations. In particular **VPOS** does one call to **POS**.
- **POS**: inner interface for vertical post-processing.
- **POS_PREPGFL**: GFL preliminary calculations for **POS**.
- **GPHPRE**: computes half-level and full-level pressures and intermediate quantities on η -levels.
- **CTSTAR**: computes the standard surface temperature and the surface temperature to be used for extrapolations of temperature and geopotential height.
- **PPCVIRT**: transforms a temperature variable into its virtual counterpart.
- **FPPS**: computes the pressures of a set of surfaces, given their geopotential heights.
- **PPPMER**: computes mean sea level pressure.
- **PPLTP**: computes the pressures of post-processing on potential vorticity levels.
- **PPLTETA**: computes the pressures of post-processing on Θ -levels.
- **PPLETA**: computes the pressures of post-processing on η -levels.
- **PPINIT**: computes intermediate quantities connected with full levels and half levels pressures, prior to vertical interpolations on pressure or height levels.
- **PPFLEV**: finds full or half levels under specified pressures.
- **PPSTA**: integrates standard atmosphere for geopotential height.
- **PPUV**: vertical interpolations on pressure levels for wind components.
- **PPT**: vertical interpolations on pressure levels for temperature.
- **PPT_OLD**: old version of routine **PPT** used at ECMWF.
- **PPQ**: vertical interpolations on pressure levels for humidity and most of post-processable GFL variables.
- **PPGEOP**: vertical interpolations on pressure levels for geopotential height.
- **PP2DINT**: vertical linear interpolation (on pressure levels or height levels).

- **PPVEL**: vertical interpolations on pressure levels or height levels for variables linked to vertical velocity.
- **PPHPW**: computation of the moist irreversible adiabatic potential temperature, vertical interpolation on pressure or height levels.
- **POAERO**: post-processing of jet and tropopause heights.
- **TJQUD**, **TJCUBI** and **TJQUAA**: post-processing of jet and ICAO tropopause.
- **PPLTEMP**: computes the pressure value for a given temperature in a vertical profile.
- **PPWETPOINT**: computation of the wetpoint temperature.
- **PPINTP**: vertical linear interpolation on pressure levels.
- **PPITPQ**: vertical quadratic 3 points interpolation.
- **PHYMFPOS**: physico-dynamic post-processing interface for METEO-FRANCE physics.
- **FPACHMT**: FULL-POS interface to ACHMT, to compute pressure, humidity, temperature and wind.
- **ACHMT**: calculation of some surface characteristics; interpolation of some variables (U , V , T , q) at some chosen heights.
- **FPCICA**: interface to **FPCINCAPE** (see below).
- **FPCINCAPE**: routine computing CAPE (convective available potential energy) and CIN (convective inhibition).
- **MTS_PHYS**: physics for TOVS radiative transfer code.
- **ENDVPOS**: interface routine managing all post-processing computations which can be done only after horizontal interpolations.
- **ENDPOS**: finish post-processing calculations for output height levels or η -levels.
- **ENDPOS_PREPGFL**: GFL preliminary calculations for **ENDPOS**.
- **APACHE**: interface routine for some vertical interpolations, called for example if post-processing on height or η -levels.
- **FPVIEW**: compute weights for vertical interpolations combining 2 profiles.

7.3.4 General architecture under FPOSHOR (horizontal interpolator).

- **FPOSHOR**: interface routine managing lagged part of horizontal interpolations.
- **FPOSHORLAG**: interface routine managing lagged part of horizontal interpolations: second part.
- **FPCLIPHY**: writes the output climatology in the post-processing buffers and initializes a logical climatological mask.
- **FPNILPHY**: computes the physical fields which cannot be directly interpolated but need auxiliary computation or need to be set-up by a straight value.
- **FPINTDYN**: interface for horizontal interpolations, for fields coming from the dynamics.
- **FPINTPHY**: interface for horizontal interpolations, for fields coming from the physics.
- **FPSAMPL**: horizontal sampling, when interpolation points are on the input grid.
- **FPINT4**: 4 points horizontal interpolations; also used for some multi-linear interpolations.
- **FPINT4X**: 4 points horizontal interpolations, with possibility to take account of missing values (less than 4 points may be used in this case).
- **FPINT12**: 12 points horizontal interpolations; also used for some multi-linear interpolations.
- **FPAVG**: FULL-POS interpolation based on average over the FULL-POS halo.
- **FPNEAR**: interpolate field for FULL-POS using the nearest point in the FULL-POS halo.
- **FPCORDYN** (resp. **FPCORPHY**): corrects dynamical (resp. physical) fields after the horizontal interpolations which may create overshoots, or to recover the requested fields.
- **FPHOR12**: corrections after 12-points horizontal interpolations, for example to prevent overshoots.
- **FPGEO**: transforms the post-processed fields into the output geometry.

8 Sequences of calls of post-processing.

8.1 Vertical coordinates of post-processing.

Post-processing can be done on pressure levels, height levels, potential vorticity levels, potential temperature levels, temperature levels, flight 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 done by different sequences under routine **STEPO_FPOS**. Vertical interpolations and horizontal interpolations are done by different sequences under **STEPO_FPOS**. **STEPO_FPOS** has its own sequence **CLCONF** (now completely different from the model one), defined by nine letters (or zeros) [L1][L2][L3][L4][L5][L6][L7][L8][L9]

- L1 controls vertical post-processing.
- L2 controls model direct spectral transforms.
- L3 controls spectral filters in model space.
- L4 controls model inverse spectral transforms.
- L5 controls the grid-point horizontal interpolations for post-processing.
- L6 controls lagged vertical post-processing.
- L7 controls output direct spectral transforms.
- L8 controls spectral filters in output spectral space.
- L9 controls output inverse spectral transforms.

No extensive inventory will be given in this documentation for letters used in the different elements of **CLCONF**, but we can say for example that:

- ‘0’ means that no action is done.
- ‘P’ or ‘F’ is generally used for elements controlling spectral transforms when they must be done.

For more details reader can refer to technical document (IDFPOS2).

8.3 Different choices for horizontal output.

* Groups of post-processed fields and 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.

Each group may have a specific treatment; CFU and XFU are treated like PHYSOL.

* **Choice of horizontal geometry and representation (spectral vs grid-point) for output:** Output data can be written on the following output grids and with the following representations:

- Spectral representation, Gaussian grid.
- Grid-point representation, Gaussian grid.
- Spectral representation, LELAM grid.
- Grid-point representation, LELAM grid.
- Grid-point representation, LALON grid.

It is done according to variables **CFPFMT** in **NAMFPC** and **NFPOS** in **NAMCT0**.

- (NFPOS, CFPFMT) = (2, 'MODEL'): output horizontal geometry is identical to model one, and output fields have the same representation as model ones.
- (NFPOS, CFPFMT) = (1, 'GAUSS'): for global model input geometry only. Output horizontal geometry is a Gaussian grid one, which can be different (or not) from the model grid. Output fields are written in grid-point representation.
- (NFPOS, CFPFMT) = (2, 'GAUSS'): for global model input geometry only. Output horizontal geometry is a Gaussian grid one, which can be different (or not) from the model grid. Output fields have the same representation as model ones.
- (NFPOS, CFPFMT) = (1, 'LELAM'): for both global model and LAM input geometry. Output horizontal geometry is a LELAM grid one, which can be different (or not) from the model grid. Output fields are written in grid-point representation.
- (NFPOS, CFPFMT) = (2, 'LELAM'): for both global model and LAM input geometry. Output horizontal geometry is a LELAM grid one, which can be different (or not) from the model grid. Output fields have the same representation as model ones.
- (NFPOS, CFPFMT) = (1, 'LALON'): for both global model and LAM input geometry. Output horizontal geometry is a LALON grid one. Output fields are written in grid-point representation.

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**= $\text{int}[(\text{NGPTOT}+\text{NPROMA}-1)/\text{NPROMA}]$ 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 done at one time for all the **NFPRGPL** (resp. **NFPRGPL_DEP**) FULL-POS points treated by the current processor in the arrival (resp. departure) horizontal geometry distribution. For the FULL-POS lagged computations there is a second subdivision into sub-packets of length **NFPROMA_DEP** (departure geometry distribution) or **NFPROMA** (arrival geometry distribution) containing useful points for computations at the points of the arrival horizontal geometry (routines **FPOSHOR**, **FPOSHORLAG** 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.

If **NFPDISTRIB**=0, the arrival horizontal geometry distribution is set equal to the departure horizontal geometry distribution. For example **NFPRGPL**=**NFPRGPL_DEP**, **NFPROMA**=**NFPROMA_DEP**. Such configuration is always used when the arrival horizontal geometry is identical to the departure horizontal geometry.

If **NFPDISTRIB**=1 or 2, the arrival horizontal geometry distribution is set different from the departure horizontal geometry distribution. The variation of **NFPRGPL** between each processor is minimized to ensure a good load balance. If there is one domain (**NFPDOM**=1), data of the **NFPRGPG**-arrays are dispatched first in the processor number 1, then in the processor number 2, etc... For example if **NFPRGPG**=64 and **NPROC**=2, **NFPRGPL** will be equal to 32 for both processors, **NFPRGPG**-arrays indices number 1 to 32 (resp. 33 to 64) will be treated by processor 1 (resp. processor 2). If there are several domains, such distribution is done first for the first domain, then for the second domain, etc., so each processor has a subset of points of all domains. **NFPDISTRIB**=1 or 2 is recommended if the arrival horizontal geometry is significantly different from the departure horizontal geometry (cases where the load balance can be not good in the departure horizontal geometry distribution); **NFPDISTRIB**>=1 is compulsory when the arrival horizontal geometry is different from the departure one and when some SURFEX surface fields are post-processed.

Horizontal interpolations should be done in the departure horizontal geometry distribution (this is the case of all the interpolations done at the beginning of routine **FPOSHOR**). Calculations which do not require horizontal interpolations are done in the arrival horizontal geometry distribution (this is the case of all the calculations done in part 5 of **FPOSHOR**). All readings and writings on FULL-POS buffers are done in the arrival horizontal geometry distribution. All the post-processing file writings are done in the arrival horizontal geometry distribution.

The way of dispatching the points among the processors in the departure horizontal geometry distribution is the same for **NFPDISTRIB**=1 or 2 and **NFPDISTRIB**=0 (see part 9.2).

NFPDISTRIB=2 sets gridpoint distributions according to the spectral transforms package. **NFPDISTRIB**=2 is needed for **NFPOS**=2 and **CFPFMT**/='MODEL'.

* **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 **NPRTRW**. 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** and **SPOSGF**.

9.2 Transmission of data necessary for FULL-POS horizontal interpolations from HPOS to FPOSHOR: 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). In the departure horizontal geometry distribution, 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 **FPOSHOR** 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 **YRFP%NSLWIDE**). This is a sort of “halo” belonging to some other processors. Due to the “halo” there is still need to split calculations into not lagged ones (**HPOS**) and lagged ones (**FPOSHOR**). 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 **SCAN2M_HPOS**). 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 **SCAN2M_HPOS**. 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 done that interpolations use only the data of 2 longitudes west and 2 longitudes east) and the halo (buffers **ZFPBUF1** in **SCAN2M_HPOS**). This memory transfer uses a precomputed intermediate array **YRFP%NSLCORE**. Then some communication routines are called in **SCAN2M_HPOS** 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 **YRFP%NASLB1** which is the total number of points one needs for the interpolations (**YRFP%NASLB1** 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 **YRFP%NASLB1** dataset is constituted, the lagged part **FPOSHOR** is called which do horizontal interpolations for all the **NFPRGPL_DEP** data to be interpolated for the current processor.

The lagged part of the computations (**FPOSHOR**, **ENDVPOS**) has a subdivision into **NFPROMA_DEP**-packets for calculations done in the departure horizontal geometry distribution, and **NFPROMA**-packets for calculations done in the arrival horizontal geometry distribution. The number of packets is **NFPBLOCS_DEP** (departure horizontal geometry distribution) and **NFPBLOCS** (arrival horizontal geometry distribution).

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.

9.4 Case **LEQ_REGIONS=T**.

This case is relevant only when **NPRGPEW**>1 (B-level parallelisation at least in the grid-point calculations). This is an optimised version of the **LEQ_REGIONS=F** case which is well designed for reduced Gaussian grid and it improves the load balance in this case. A comprehensive description can be found in (Mozdzynski, 2006). To sum-up, we can say that:

- the A-level grid-point distribution splits the Earth into **N_REGIONS_NS** bands. **N_REGIONS_NS** can be slightly different from **NPRGPNS**.
- for each band *proca*, the B-level grid-point distribution splits the band into **N_REGIONS**(*proca*) zones: the minimum value of **N_REGIONS** is at the poles of the computational sphere (equal to 1 in the examples provided by Mozdzynski); the maximum value of **N_REGIONS** is at the equator of the computational sphere and this maximum is equal to **N_REGIONS_EW**. The meridian variations of **N_REGIONS** are highly correlated to those of **NLOENG**.
- In the examples provided by Mozdzynski, **NPRGPNS=NPRGPEW=NPRTRW=NPRTRV** and we notice that **N_REGIONS_NS** is slightly below **NPRGPNS**, and that **N_REGIONS_EW** is slightly below **2*NPRGPEW**.

When **LEQ_REGIONS=F**, variables **N_REGIONS_NS**, **N_REGIONS** and **N_REGIONS_EW** are still used but in this case:

- **N_REGIONS_NS=NPRGPNS**.
- **N_REGIONS=NPRGPEW** everywhere.
- **N_REGIONS_EW=NPRGPEW**.

10 Module and namelist variables to be known.

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

- **EINT_MOD**: externalisable part of interpolators, and halo management (including distributed memory aspects to communicate the halo among the different processors). In particular, variable **YRFP** is used in FULL-POS horizontal interpolator.
- **EXTFPSELECT_MOD**: contains encapsulated routines doing memory transfers from/to buffers.
- **FPGPNORM_MOD**: contains encapsulated routines doing grid-point norms calculations.
- **FULLPOS_MIX**: defines the type **FULLPOS_TYPES** and contains also some routines working on this type of variables.
- **PARFPOS**: contains basic dimensions for FULL-POS post-processing.
- **PTRFP4**: contains control variables for post-processing of physical fields.
- **PTRFPB2**: contains variables and pointers for auxiliary surface grid point fields used for horizontal post-processing.
- **TYPE_FPDSPHY**: defines the type **TYPE_FPDSPHY** and contains also some routines working on this type of variables.
- **TYPE_FPRQDYN**: defines the type **TYPE_FPRQDYN** and contains also some routines working on this type of variables. Concerns DYN3D and DYN2D post-processable fields.
- **TYPE_FPRQPHY**: defines the type **TYPE_FPRQPHY** and contains also some routines working on this type of variables. Concerns PHYSOL, CFU and XFU post-processable fields.
- **YEMFPG**: output geometry, LAM features.
- **YOM4FPOS**: contains variables relative to FULL-POS working arrays (level 4).
- **YOMAFN**: contains ARPEGE fields descriptors. Some of these variables are in namelist **NAMAFN**.
- **YOMCAPE**: contains variables to control CAPE computation in FULL-POS. Some of these variables are in namelist **NAMCAPE**.
- **YOMARG** (0-level control, former command line) and **YOMCT0** (0-level control):
 - NCONF (in **NAMARG**).
 - NFPOS.
 - LECPOS.
 - CFPNCF, CFDIRLST, CNPPATH.
 - NFRPOS, NPOSTS.

Some of these variables are in namelist **NAMCT0**.

- **YOMCT1** (1-level control): in particular N1POS. Some of these variables are in namelist **NAMCT1**.
- **YOMDFPB**: contains buffer for horizontally post-processed main fields.
- **YOMDIM**: contains dimensioning variables. In particular NDLO, LOPTPROMA, NPROMA, NGPBLKS, NDGSAPPH, NDGENFPH. Some of these variables are in namelist **NAMDIM**.
- **YOMFP4L**: contains information relative to lagged variables needed to write out post-processed fields.
- **YOMFPC**: contains scientific and technical variables for post-processing. Some of these variables are in namelist **NAMFPC**.
- **YOMFPCT0**: contains variables concerning the control of post-processing.
- **YOMFPD**: contains variables concerning the boundaries and the horizontal dimensions of each output subdomain. Some of these variables are in namelist **NAMFPD**.
- **YOMFPDIM**: contains FULL-POS dimensions.
- **YOMFPEZO**: contains variables related to post-processing on extension zone of LAM model output geometry.
- **YOMFPF**: contains variables for FULL-POS filtering. Some of these variables are in namelist **NAMFPF**.
- **YOMFPG**: contains variables defining the characteristics of the (transformed) output geometry. Some of these variables are in namelist **NAMFPG**.
- **YOMFPGIND**: contains variables defining some indexes.
- **YOMFPIOS**: contains control variables for FULL-POS file read/write. Some of these variables are in namelist **NAMFPIOS**.
- **YOMFPOP**: contains post-processing file-handling variables.

- **YOMFPSC2**: contains control variables for horizontal post-processing and in particular some array indexes and dimensions. Some of these variables are in namelists **NAMFPSC2** and **NAMFPSC2_DEP**.
- **YOMFPSP**: contains FULL-POS spectral arrays.
- **YOMFPTRANS**: contains variables for spectral transforms done on output domains.
- **YOMIOS**: contains packing factors and pathnames for some files. Some of these variables are in namelist **NAMIOS**.
- **YOMMP0** and **YOMMP**: distributed memory environment, see documentation (IDDM) for more details.
- **YOMOPH0** and **YOMOPH**: contains file-handling variables. Some of these variables are in namelist **NAMOPH**.
- **YOMFPFB**: contains buffer for horizontally post-processed auxiliary surface fields.
- **YOMPPVI**: contains variables for vertical interpolator used for example in FULL-POS. Some of these variables are in namelist **NAMPPVI**.
- **YOMRFPB**: contains buffer of auxiliary fields (geometry, climatology) in post processing geometry.
- **YOMRFPDS**: contains variables for description of auxiliary fields for horizontal post-processing.
- **YOMSTA**: contains constants for standard atmosphere. In particular **HEXTRAP**.
- **YOMVPOS**: contains control variables for vertical post-processing.
- **YOMWFPB**: contains indexes and weights for horizontal interpolations.
- **YOMWFPS**: contains variables for internal control of horizontal interpolations.
- Modules for geometry:
 - **YOMCSGEOM** (computational space grid-point geometry): **YRCSGEOM**, **YRCSGEOM_NB**.
 - **YOMGSGEOM** (geographic space grid-point geometry): **YRGSGEOM**, **YRGSGEOM_NB**.
 - **YOMOROG** (orography): **YROROG**.
 - **YOMGEM** (other variables for horizontal geometry): all variables. Some of these variables are in namelist **NAMGEM**.
 - **YOMVERT** (vertical geometry): “VAB” and “VETA” variables.
 - **YOMVV1** (namelist variables for vertical geometry). Variables are in namelist **NAMVV1**.
 - **YEMGEO** (LAM model geometry): all variables. Some of these variables are in namelist **NEMGEO**.

Additional namelists elements: these namelists are **NAMFPPHY** (for physical fields and fluxes), **NAMFPDY2** (for 2D dynamical fields), **NAMFPDYP** (fields post-processed on pressure levels), **NAMFPDYH** (fields post-processed on height levels), **NAMFPDYV** (fields post-processed on potential vorticity levels), **NAMFPDYI** (fields post-processed on isentropic levels), **NAMFPDYT** (fields post-processed on temperature levels), **NAMFPDYF** (fields post-processed on flight levels), and **NAMFPDYS** (fields post-processed on hybrid levels). These namelists can be used to make an accurate list of post-processed fields.

11 References.

11.1 Some internal notes and other ARPEGE notes.

- (TDECDYN) 2015: IFS technical documentation (CY41R1). Part III: dynamics and numerical procedures. Available at “<https://software.ecmwf.int/wiki/display/IFS/Official+IFS+Documentation>”.
- (TDECTEC) 2015: IFS technical documentation (CY41R1). Part VI: technical and computational procedures. Available at “<https://software.ecmwf.int/wiki/display/IFS/Official+IFS+Documentation>”.
- Andersson, E., and Ph. Courtier, 1992: Post-processing changes for IFS cycle 9. (Research Department memorandum, internal note).
- (IDFPOS2) El Khatib R., 2012: FULLPOS-2: design, specifications study and development (version 6). Internal note, 15pp.
- 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).
- (IDEQR) Mozdzyński, G., 2006: A new partitioning approach for IFS. Internal note, 6pp.
- (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., 2016: Basics about ARPEGE/IFS, ALADIN and AROME in the cycle 43 of ARPEGE/IFS (internal note).
- (IDEUL) Yessad, K., 2016: Integration of the model equations, and Eulerian dynamics, in the cycle 43 of ARPEGE/IFS (internal note).
- (IDTS) Yessad, K., 2016: Spectral transforms in the cycle 43 of ARPEGE/IFS (internal note).
- (IDRD) Yessad, K., 2016: Sphere to sphere transforms in spectral space in the cycle 43 of ARPEGE/IFS: configuration 911. (internal note).
- (IDDM) Yessad, K., 2016: Distributed memory features in the cycle 43 of ARPEGE/IFS (internal note).