DISTRIBUTED MEMORY FEATURES IN THE CYCLE 46T1 OF ARPEGE/IFS.

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

December 19, 2018

Abstract:

This documentation describes some aspects of the memory distribution for parallel environment (message passing). It is written as a complement of some other documentations (mainly the ECMWF documentation) so some points are not detailed here. A list of routines doing processor communication is provided.

Résumé:

Cette documentation décrit quelques uns des aspects de l'environnement distribué dans le code (message passing). Elle est écrite comme un complément à d'autres documentations existantes (en particulier celles du CEP), certains points n'y sont donc pas repris en détail. On fournit une liste aussi complète que possible des routines faisant de la communication entre processeurs.

Contents

1 Introduction and theoretical aspects.

In a distributed memory architecture, each processor has its own memory: that means that a quantity which is declared in several different processors corresponds to different parts of memory. The consequence is that some communications between processors will be necessary when:

- different parts of the code have not the same way of sharing data between processors.
- there is a need to collect data or to dispatch data.

This documentation will be a complement to the documentation (TDECTEC25) provided by ECMWF (IFS technical documentation, part VI, chapter 3) and a more recent version of this documentation (TDECTEC, chapter 2) for:

- features not described in detail in the ECMWF documentation.
- features which have changed from the last version of the ECMWF documentation.

The ECMWF documentation currently provided is valid for the cycle CY25R1 (TDECTEC25) and CY43R3 (TDECTEC) so some points may have been changed in the cycle 46t1 of ARPEGE/IFS.

For some convenience expressions such "DM-local" or "DM-global" will be used to describe some distributed memory features.

- 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 is DM-local and the other part is DM-global.

2 Number of processors.

2.1 Number of processors.

The total number of processors (without the IO server processors) is **NPROC**. Processors are divided into two levels of distribution.

	A-level parallelisation	B-level parallelisation
Spectral calculations where all zonal wavenumbers are independent - semi-implicit scheme: - horizontal diffusion:	 **NPRTRW** **NPRTRW**	 **NPRTRN** **NPRTRV**
Legendre transforms:	**NPRTRW**	**NPRTRV**
Fourier space calculations:	**NPRTRW**	**NPRTRV**
Fast Fourier transforms:	**NPRTRNS**	**NPRTRV**
Grid-point calculations:	**NPRGPNS**	**NPRGPEW**
Reading ARPEGE file:	**NSTRIN**	1
Writing on ARPEGE file:	**NSTROUT**	1

Remarks:

- **NPROC** is equal to **NPRTRW**∗**NPRTRV**.
- **NPROC** is equal to **NPRGPNS**∗**NPRGPEW**.
- **NPRTRN** is equal to **NPRTRV**.
- **NPRTRNS** is equal to **NPRTRW**.

2.2 Which are my A-set and B-set processors?

	A-level parallelisation	B-level parallelisation
Spectral calculations where all zonal wavenumbers are independent - semi-implicit scheme: - horizontal diffusion:	 **MYSETM** **MYSETM**	 **MYSETN** **MYSETV**
Legendre transforms:	**MYSETW**	**MYSETV**
Fourier space calculations:	**MYSETW**	**MYSETV**
Fast Fourier transforms:	**MYSETW**	**MYSETV**
Grid-point calculations:	**MYSETA**	**MYSETB**

My processor is **MYPROC**.

3 Distributed memory features.

3.1 Distributed memory features for grid-point calculations.

∗ **General case:** They are described in the ECMWF documentation (TDECTEC25) and (TDECTEC); some description is done also in MF documentations. For more details:

- Documentation (TDECTEC), part 2.2, and in particular part 2.2.4 for the interpolation halo aspects.
- Data independence: paragraph 3.3.1 of (TDECTEC25).
- Data structure: paragraphs 3.4.1, 3.4.2, 3.4.3 of (TDECTEC25).
- Data distribution: paragraph 3.5.1 of (TDECTEC25).
- Horizontal interpolations: part 3.7 of (TDECTEC25).
- Figures about horizontal distribution: A1, A2, A3, A4, A5 of (TDECTEC25).
- Figures about horizontal interpolation halo: A6, A7 of (TDECTEC25).
- Other information in MF documentation: (IDFPOS), (IDSL), (IDTS).

∗ **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 global models with 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 *jroca*, the B-level grid-point distribution splits the band into **N_REGIONS**(*jroca*) 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**.

∗ **Specific FULL-POS transpositions:** Additionally to information provided in FULL-POS documentation (IDFPOS), we can say that there are specific transposition routines to go from the departure horizontal geometry distributed environment towards the arrival one (**FPTRDTOA** from departure to arrival, **FPTRATOD** from arrival to departure).

3.2 Distributed memory features for fast Fourier transforms and Fourier space.

They are described in the Part VI of ECMWF documentation; some description is done also in MF documentations. For more details:

- Documentation (TDECTEC), part 2.3.
- Data independence: paragraph 3.3.2 of (TDECTEC25).
- Data structure: paragraphs 3.4.2, 3.4.3, 3.4.4 of (TDECTEC25).
- Data distribution: paragraph 3.5.2 of (TDECTEC25).
- Figures about horizontal distribution: A2 of (TDECTEC25).
- Figures about vertical distribution: A5 of (TDECTEC25).
- Other information in MF documentation: (IDFPOS), (IDDH), (IDTS).

3.3 Distributed memory features for Legendre transforms.

They are described in the Part VI of ECMWF documentation; some description is done also in MF documentations. For more details:

- Documentation (TDECTEC), part 2.4.
- Data independence: paragraph 3.3.3 of (TDECTEC25).
- Figures about horizontal spectral distribution: A8 of (TDECTEC25).
- Figures about vertical distribution: A5 of (TDECTEC25).
- Other information in MF documentation (IDFPOS), (IDTS).

3.4 Distributed memory features for spectral space calculations.

They are described in the Part VI of ECMWF documentation; some description is done also in MF documentations. For more details:

- Documentation (TDECTEC), part 2.5, for semi-implicit scheme.
- Data independence: paragraph 3.3.4 of (TDECTEC25).
- Figures about horizontal spectral distribution: A8 (horizontal diffusion), A9 (semi-implicit scheme), of (TDECTEC25).
- Figures about vertical distribution: A5 of (TDECTEC25).
- Other information in MF documentation (IDFPOS), (IDDH), (IDSI), (IDTS).

3.5 Distributed memory features for file reading or writing.

This point is not described at all in the ECMWF documentation. Some documentation can be found in (IDTECIO). Description is provided here for ARPEGE files only.

∗ **File reading:** ARPEGE files can be read by **NSTRIN** processors. Each processor reads a subset of fields. Currently the repartition of fields among processors is the following (circular distribution):

- fields number 1, **NSTRIN**+1, 2**NSTRIN**+1, etc.. are read on the processor number 1.
- fields number 2, **NSTRIN**+2, 2**NSTRIN**+2, etc.. are read on the processor number 2.
- ...
- fields number **NSTRIN**-1, 2**NSTRIN**-1, 3**NSTRIN**-1, etc.. are read on the processor number **NSTRIN**-1.
- fields number **NSTRIN**, 2**NSTRIN**, 3**NSTRIN**, etc.. are read on the processor number **NSTRIN**.

∗ **File writing:** **NSTROUT** processors can write on ARPEGE files. Each processor writes a subset of fields. Currently the repartition of fields among processors is the following (circular distribution):

- fields number 1, **NSTROUT**+1, 2**NSTROUT**+1, etc.. are written on the processor number 1.
- fields number 2, **NSTROUT**+2, 2**NSTROUT**+2, etc.. are written on the processor number 2.
- ...
- fields number **NSTROUT**-1, 2**NSTROUT**-1, 3**NSTROUT**-1, etc.. are written on the processor number **NSTROUT**-1.
- fields number **NSTROUT**, 2**NSTROUT**, 3**NSTROUT**, etc.. are written on the processor number **NSTROUT**.

∗ **IO server:** In this case, **NPROC_IO** processors are dedicated to IO treatment. The total amount of processors used by the job is **NPROC**+**NPROC_IO**.

3.6 Configurations which incomplete distributed memory features.

∗ **Configurations which work only for one processor, no code provided for several processors at all:** There are at least all configurations 9xx other than former configuration 911.

∗ **Configurations with A-level of distributed memory code only, no B-level distributed memory code at all:** There are at least the following configurations:

- Former configuration 911, now in "utilities" (distribution of latitudes: each processor has a contiguous number of latitudes close to **NDGLG**/**NPROC**, and for parts requiring a distribution of the zonal wavenumber the spectral calculations type of distribution is taken).

4 Processor communication routines, transposition routines.

4.1 MPL_... routines in ifsaux/module:

These routines are auto-documented so the reader can look at code sources for information.

MPL_... are ECMWF's message passing interface layer which presently calls **MPI_...** routines underneath.

The most frequently used routines are **MPL_ALLREDUCE**, **MPL_BROADCAST**, **MPL_RECV**, **MPL_SEND**, **MPL_BARRIER**.

MPL_DATA_MODULE contains variables controlling the execution of MPL. The other **MPL_...** routines contain encapsulated routines. See in directory ifsaux/module for the comprehensive list of routines.

4.2 Processor communication and transposition routines:

∗ **Halo management for horizontal interpolators:** These routines are currently stored in **arpifs/interpol** or **aladin/interpol**.

- **SLCOMM**: communicates grid-point data to make the halo necessary for horizontal interpolations (for ex. semi-Lagrangian scheme, FULL-POS, observation interpolator). For the semi-Lagrangian scheme this routine is used when the "on-demand" processor communications are not asked for.
- **SLCOMM2A**: used instead **SLCOMM** when the "on-demand" processor communications are asked for (semi-Lagrangian scheme only). Does the communication only on the part of the interpolation buffer which contains the RHS-equations quantities.
- **SLCOMM2**: old version of **SLCOMM2A** still used for some applications (not used any longer in the semi-Lagrangian scheme).
- **SLEXTPOL**: fills the halo of width band **YR..%NSLWIDE(N,S,E,W)** for non polar extra-polar latitudes. For the semi-Lagrangian scheme this routine is used when the "on-demand" processor communications are not asked for.
- In LAM models, **ESLEXTPOL** is used instead of **SLEXTPOL**.
- **SLEXTPOLAD**: adjoint code of **SLEXTPOL**.

∗ **List of (encapsulated) routines stored in directory arpifs/module:**

- **DISGRID_SEND+DISGRID_RECV** (in disgrid_mod.F90): communication of grid-point data. Dispatches a DM-global **NGPTOTG**-dimensioned real grid-point field initially present in the "main" processor, into DM-local **NGPTOT**-dimensioned real grid-point fields in each processor.
- **DIWRGRID_SEND+DIWRGRID_RECV** (in diwrgrid_mod.F90): communication of grid-point data; inverse action of **DISGRID_SEND+DISGRID_RECV**.
- **DIWRSPEC_SEND+DIWRSPEC_RECV** (in diwrspec_mod.F90): communication of spectral data (DM-local towards DM-global data).

∗ **List of routines stored in directory arpifs/parallel:**

- **BCASTCOV**: broadcasts model error covariance data to all processors.
- **BRPTOB**: useful when the second-level of parallelisation is activated. Communication of spectral upper-air temperature and surface pressure.
- **CASND1**: the "main" processor receives a DM-local integer array from all the other processors and computes a sum on all these DM-local data.
- **CASNDR1**: cf. **CASND1** but for real data.
- **COMMFCE2**: sends forecast error grid parameters from a "main" processor to all other processors.
- **COMMJBBAL**: communication of spectral GSA data to the "main" processor. The "main" processor collects DM-local real data from the other processors and creates DM-global data.
- **COMMJBDAT**: cf. **COMMJBBAL** but for some other fields.
- **COMMSPNORM**: sends DM-global spectral real data from the "main" processor to all other processors. Is used to compute spectral norms of model variables.
- **COMMSPNORM1**: cf. **COMMSPNORM** but for FULL-POS variables.
- **DDHRCV**: routine to input DDH restart data and distribute.
- **DDHSND**: routine to gather DDH restart data and output.
- **DISGRID_SURF_EXT**: communication of grid-point data for the externalised surface.
- **DISSPEC0**: communication of spectral data in an optimized way. Dispatches packets of DM-global **NSPEC2G**-dimensioned real spectral fields (each processor 1 to **NSTRIN** containing a subset of fields before the communication), into DM-local **NSPEC2**-dimensioned real spectral fields (each processor 1 to **NPROC** having the complete set of fields for the **NSPEC2** spectral coefficients).

- **DISTDDH**: communication for DDH (real data). The "main" processor receives the DM-local versions of "distance of single points", computes the DM-global maximum, and sends to the other processors this DM-global maximum.
- **DIWRGRFP**: communication of FULL-POS grid-point data; inverse action of **DISGRIDFP**.
- **DIWRGRID_SURF_EXT**: communication of grid-point data for the externalised surface; inverse action of **DISGRID_SURF_EXT**.
- **DLADDH**: communicates communication tables for latitude bands to master process. Used in the DDH. The "main" processor receives DM-local integer data from the other processors, collects them into a DM-global integer array and sends this DM-global array to the other processors.
- **DMADDH**: communicates communication tables for domain masks to master process. Used in the DDH. The "main" processor receives DM-local integer data from the other processors, collects them into a DM-global integer array and sends this DM-global array to the other processors.
- **DOT_PRODUCT_CTLVEC**: ??? (no comment in this routine); is used for control vector.
- **DRESDDH**: communicates intermediate results to the "main" processor. The "main" processor collects DM-local real data from the other processors and creates DM-global data.
- **FPTRATOD**: in FULL-POS, transposition from the arrival horizontal geometry DM-environment towards the departure one.
- **FPTRDTOA**: in FULL-POS, transposition from the departure horizontal geometry DM-environment towards the arrival one.
- **FPTRGTOA**: in FULL-POS, transposition from the DM-global horizontal geometry towards the arrival one.
- **GATHERBDY**: gathers a partitioned boundary array to global form. Takes DM-local **NGPTOT**-dimensioned real grid-point fields in each processor, sends them to the "main" processor where data are collected into a DM-global **NGPTOTG**-dimensioned real grid-point field.
- **GATHERCOST1**: gathers partitioned spectral cost function contributions. Takes DM-local **NUMP**-dimensioned real spectral fields in each processor, sends them to the "main" processor where data are collected into a DM-global **NVARMAX**-dimensioned real spectral field.
- **GATHERCOST2**: gathers partitioned spectral cost function contributions. Takes DM-local (**NUMP**,**NFLEVL**)-dimensioned real spectral fields in each processor, sends them to the "main" processor where data are collected into a DM-global (**NVARMAX**,**NFLEVG**)-dimensioned real spectral field.
- **GATHEREIGMD**: gathers partitioned eigenvectors to global form. Takes DM-local **NUMP**-dimensioned real spectral fields in each processor, sends them to the "main" processor where data are collected into a DM-global **NSMAX**-dimensioned real spectral field.
- **GATHERGPF**: gathers global gridpoint fields.
- **GATHERGPF_WAVELET**: gathers global gridpoint fields for a set of wavelet scales.
- **GATHERSPA**: gathers partitioned spectral arrays to global form. Takes DM-local (**NSPEC2**,**NFLEVL**)-dimensioned real spectral fields in each processor, sends them to the "main" processor where data are collected into a DM-global (**NSPEC2G**,**NFLEVG**)-dimensioned real spectral field.
- **GATHERT**: gathers diffused temperature values. Takes DM-local (**NUMP**,**NFLEVL**)-dimensioned real spectral fields in each processor, sends them to the "main" processor where data are collected into a semi-DM-global (**NUMP**,**NFLEVG**)-dimensioned real spectral field.
- **GATHERTC**: gathers the time correction observation errors message passing. The "main" processor receives DM-local real data from the other processors, gathers them in DM-global data, then sends these DM-global data to the other processors.
- **GPNORM1**: communication of grid-point data (cf. **DISGRID_SEND**+**DISGRID_RECV**) and prints norms. Takes DM-local **NGPTOT**-dimensioned real grid-point fields in each processor, sends them to the "main" processor where data are collected into a DM-global **NGPTOTG**-dimensioned real grid-point field, and prints DM-global norms.
- **MYRECVSET** (function) returns set number to receive from.
- **MYSENDSET** (function) returns set number to send to.
- **READ_SPEC**: reads GRIB spectral fields from files, one file per level.
- **READ_SPEC_GRIB**: reads spectral fields from GRIB file and distributes them.
- **READ_SPEC_FROMFA**: cf. **READ_SPEC** but reads ARPEGE FA files.
- **TRMTOS**: transpose spectral data from partitioning over total wave numbers and vertical layers (DM-local (**NSPEC2**,**NFLEVL**)-dimensioned real spectral fields) necessary for example for spectral horizontal diffusion, to a finer partitioning over total wave numbers without vertical partitioning (DM-local (**NSPEC2V**,**NFLEVG**)-dimensioned or (**NSPEC2VF**,**NFLEVG**)-dimensioned real spectral fields) necessary for example for spectral semi-implicit scheme.
- **TRMTOS_SPEC**: transpose spectral data distributed by columns (**SPECTRAL_FIELDS** type) into **SPECTRAL_COLUMNS** type.
- **TRSTOM**: inverse action of **TRMTOS**.

- **TRSTOM_SPEC**: transpose spectral data distributed by columns (**SPECTRAL_COLUMNS** type) into **SPECTRAL_FIELDS** type.
- **TRWVTOF**: transposition routine to communicate a set of DM-global spectral fields from W-set processors to the processors in charge of DM-global work prior to writing out fields (usually : packing). The distribution per fields of DM-global spectra is contiguous among a subset of processors.
- **WHICHPROC**: returns the processor number.
- **WRGP_SURF**: communication of grid-point data for the externalised surface (calls **DIWRGRID_SURF_EXT**).
- **WRITE_SPEC** and **WRITE_SPEC_TRAJ**: writes spectral fields in files, one file per level; this routine does communications between processors.
- **WRITE_SPEC_GRIB**: gathers global versions of spectral arrays on one PE and writes it in GRIB file. If the resolution defined in the GRIB header is different from the model resolution the fields are truncated to that resolution.

∗ **List of routines stored in directory aladin/parallel:**

- **ECOMMJBBAL** and **ECOMMJBBALBETA**: LAM model versions of **COMMJBBAL**.
- **ECOMMSPNORM**: LAM model version of **COMMSPNORM**.
- **EGATHEREIGMD**: LAM model version of **GATHEREIGMD**.

∗ **List of routines stored in projects "trans" and "etrans":** The list is the following one:

- **DIST_GRID_CTL** and **DIST_GRID_32_CTL**: distributes global gridpoint array to processors.
- **DIST_SPEC_CONTROL** (**EDIST_SPEC_CONTROL** for LAM models): distributes global spectral array to processors.
- **EQ_REGIONS_MOD**: set of routines used for **LEQ_REGIONS**=T.
- **GATH_GRID_CTL** and **GATH_GRID_32_CTL**: gathers global gridpoint array from processors.
- **GATH_SPEC_CONTROL** (**EGATH_SPEC_CONTROL** for LAM models): gathers global spectral array from processors.
- **INIGPTR**: compute tables to assist GP to/from Fourier space transpositions.
- **TRGTOL**: transpose grid-point data from partitioning over sets of **NGPTOT** grid-points containing complete columns (all the **NFLEVG** model layers) and incomplete latitudes to partitioning over complete latitudes (a processor contains a subset of complete latitudes) and layers if **NPRTRV**>1 (a processor contains **NFLEVL** layers). The initial partitioning is necessary for grid-point computations which need complete columns. The final partitioning is necessary for fast Fourier transforms which need complete latitudes but can be done on a subset of complete latitudes and layers.
- **TRLTOG**: inverse action of **TRGTOL**.
- **TRLTOM**: transpose Fourier buffer data from partitioning over latitudes (necessary for direct fast Fourier transforms) to partitioning over zonal wave numbers (necessary for direct Legendre transforms).
- **TRMTOL**: inverse action of **TRLTOM**.

∗ **List of routines stored in project "ifsaux":** They are generally in directory **parallel**.

- **CMPL_BINDING**: contains some hat routines for MPL routines.
- **COML_BINDING**: contains some hat routines for OML routines.

∗ **Remarks:**

- Some other routines spread in several directories do some memory distribution, at least partly.

5 Setup routines and other routines computing distributed memory environment.

∗ **Halo management for horizontal interpolators:** These routines are currently stored in **arpifs/interpol**.

- **SLCSET**: computes the halo for the local processor (some attributes of EINT_MOD variables). For more details see part 2.2.4 of (TDECTEC).
- **SLRSET**: computes the send-list and receive-list information used by routine **SLCOMM** (some attributes of EINT_MOD variables). For more details see part 2.2.4 of (TDECTEC). See remark above for the other interpolators (FULL-POS, observation, ECMWF radiation).

∗ List of other setup routines of directory "parallel":

- **PE2SET**: once known the number of the processor between 1 and **NPROC**, gives:
  - the "A-set" number of the processor between 1 and **NPRGPNS** for grid-point computations.
  - the "B-set" number of the processor between 1 and **NPRGPEW** for grid-point computations.
  - the "A-set" number of the processor between 1 and **NPRTRW** for spectral computations.
  - the "B-set" number of the processor between 1 and **NPRTRV** for spectral computations.
- **SET2PE**: gives the number of the processor between 1 and **NPROC**, once known the following numbers:
  - the "A-set" number of the processor between 1 and **NPRGPNS** for grid-point computations.
  - the "B-set" number of the processor between 1 and **NPRGPEW** for grid-point computations.
  - the "A-set" number of the processor between 1 and **NPRTRW** for spectral computations.
  - the "B-set" number of the processor between 1 and **NPRTRV** for spectral computations.

For more details, see part 3.8.3 of (TDECTEC25).

∗ List of setup routines of directory "fullpos":

- **SUMPFPOS** and **SUMPFPOS_DEP**: computes control array for the distribution and to distribute some global fields for FULLPOS.

∗ List of setup routines of directory "setup":

- "Init object" routines filling **YOMMP0**: they are called before **SUGEOMETRY**. In particuliar, they fill **YOMMP0** variables.
  - **SUMPINI**: computes the number of processors and some variables required before calling **SUCT0**. Reads namelist **NAMPAR0**.
  - **SUMPINI_PRT**: prints **YOMMP0** quantities computed by **SUMPINI**.
  - **SUMPOUT**, called by **SUMPINI_PRT**: determines if the current processor writes outputs or not.
  - **SUMP0**: reads namelist **NAMPAR1**.
  - **SUTRANS0**.
- "Geometry object" routines filling **YOMMP**: they are called under **SUGEOMETRY**.
  - **SUMP**: fills **YOMMP** (geometry-dependent distributed memory environment variables).
  - **SUEMP**: fills **YEMMP** (additional quantities for LAM model).
  - **SUALMP1** and **SUALMP2**, called by **SU(E)MP**: allocates some distributed memory environment variables.
- Other routines:
  - **SUMPIOH**: to setup in a contiguous way the distribution of fields when read/write among processors.

∗ List of setup routines of directory "io_serv":

- **IO_SERV_INIT**: set-up for IO server, fills **YOMIO_SERV**.

∗ List of setup routines of "trans" package:

- **SUMP_TRANS0**: sets up distributed environment for the transform package (part 0).
- **SUMP_TRANS**: sets up distributed environment for the transform package (part 2).
- **SUMP_TRANS_PRELEG**: sets up distributed environment for the transform package (part 1).
- **SUMPLAT** and **SUMPLATF**: computes distributed environment about latitude partitioning in grid-point and Fourier spaces.
- **SUMPLATB**: part of calculations under **SUMPLAT** which are different for spherical geometry and plane geometry (in this case **SUEMPLAT** is called instead).
- **SUSTAONL**: defines the repartition between processors of the grid-point columns, defines the grid-point columns which belong to the current processor.
- **SUWAVEDI**: initialises arrays controlling spectral wave distribution.

For more details, see part 3.8.3 of (TDECTEC25).

∗ List of other routines:

- **parallel/GL2LL**: routine to convert from DM-global computational sphere lat/lon numbers to DM-local computational sphere lat/lon numbers.

6 Module and namelist variables.

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:

- **CONTROL_VECTORS_COMM_MOD** (control vector): contains encapsulated subroutines, some of them are linked with distributed memory.
- **EINT_MOD** (externalisable part of interpolators): in particular, variables **YRSL**, **YRRI**, **YRRO**, **YRFP**, **YRAD**, some attributes of which are used to communicate the halo among the different processors.
- **YOMMASK** (must go later in **EINT_MOD**).
- **OML_MOD** (in **ifsaux/module**): no comment provided in this module (seems linked with OpenMP or distributed memory).
- **YOMDIM**, **YOMDIMV** and **YOMDIMF** (dimensioning): most of variables. Some of these variables are in namelist **NAMDIM**.
- **YOMFPGEO** (output geometry for FULL-POS), in particular attributes NFPRGPL, NFPRGPG, NFPRGPLX, NFPRGPNUM, NFPRGPIND.
- **YOMFPG**, **YOMFPGIND**.
- **YOMGEM** (grid-point horizontal geometry), in particular attributes NLOEN, NLOENG, NMEN, NMENG, NDGLU, NSTAGP, NTSTAGP, NGPTOT, NGPTOTL, NGPTOT_CAP, NGPTOTG, NGPTOTMX.
- **YOMGLOBS** (contains some information about processor communications for the observations).
- **YOMGSTATS** (in **ifsaux/module**): module for timing statistics. Some of these variables are in namelist **NAMPAR0**.
- **YOMIO_SERV** (IO server), in particular attribute NPROCS_IO. Some of these variables are in namelist **NAMIO_SERV**.
- **YOMLAP** and **YEMLAP** (variables defining some spectral indexes arrays and spectral Laplacian operators).
- **YOMLEG** (description of Legendre polynomials).
- **YOMMP0**: distributed memory environment variables which are not geometry-dependent, and which belong to the "init" object. Some of these variables are in namelists **NAMPAR0** and **NAMPAR1**.
- **YOMMP**: distributed memory environment variables which are geometry-dependent, and which belong to the "geometry" object.
- **YOMMPI** (in **ifsaux/module**; contains identifiers used by MPI).
- **YOMTAG** (tag identifiers used in message passing communication).

Some variables have a DM-global and a DM-local version, for example:

Module	DM-global variable	DM-local variable
YOMDIM	NDGLG	NDGLL
	NDGSAG	NDGSAL
	NDGENG	NDGENL
	NSPECG	NSPEC
	NSPEC2G	NSPEC2
	NDGUNG	NDGUNL
	NDGUXG	NDGUXL
	NDLUNG	NDLUNL
	NDLUXG	NDLUXL
YOMDIMV	NFLEVG	NFLEVL
YOMFPGEO	NFPRGPG	NFPRGPL
YOMGEM	NGPTOTG	NGPTOT
	NLOENG	NLOEN
	NMENG	NMEN
YOMLAP	NASMOG	NASMO

7 References.

- Courtier, Ph., C. Freydier, J. F. Geleyn, F. Rabier and M. Rochas, 1991: The ARPEGE project at METEO-FRANCE. ECMWF Seminar Proceedings 9-13 September 1991, Volume II, 193-231.
- (IDIDM) El Khatib, R., 2002: An introduction to the memory-distributed aspect of the code ARPEGE/IFS/ALADIN. Internal note, 11pp, available on the internet server "http://www.umr-cnrm.fr/gmapdoc/".
- Estrade, J.F., 1997: Développement ARPEGE/ALADIN. Mémoire distribuée/mémoire partagée (internal note in French).
- (TDECTEC25) 2002: IFS technical documentation (CY25R1). Part VI: technical and computational procedures.
- (TDECTEC) 2017: IFS technical documentation (CY43R3). Part VI: technical and computational procedures. Available at "https://software.ecmwf.int/wiki/display/IFS/Official+IFS+Documentation".
- (IDTECIO) Marguinaud, Ph., 2013: ARPEGE/ALADIN/AROME IO in 39t1. Internal note, 18pp. Available on the intranet server "http://www.umr-cnrm.fr/gmapdoc/".
- (IDEQR) Mozdzynski, G., 2006: A new partitioning approach for IFS. Internal note, 6pp.
- Rochas, M., et Ph. Courtier, 1992: La méthode spectrale en météorologie. Note de travail ARPEGE numéro **30**, 58pp.
- (IDBAS) Yessad, K., 2018: Basics about ARPEGE/IFS, ALADIN and AROME in the cycle 46t1 of ARPEGE/IFS (internal note).
- (IDEUL) Yessad, K., 2018: Integration of the model equations, and Eulerian dynamics, in the cycle 46t1 of ARPEGE/IFS (internal note).
- (IDSL) Yessad, K., 2018: Semi-Lagrangian computations in the cycle 46t1 of ARPEGE/IFS (internal note).
- (IDSI) Yessad, K., 2018: Semi-implicit spectral computations in the cycle 46t1 of ARPEGE/IFS (internal note).
- (IDDH) Yessad, K., 2018: Horizontal diffusion in the cycle 46t1 of ARPEGE/IFS (internal note).
- (IDTS) Yessad, K., 2018: Spectral transforms in the cycle 46t1 of ARPEGE/IFS (internal note).
- (IDFPOS) Yessad, K., 2018: FULL-POS in the cycle 46t1 of ARPEGE/IFS (internal note).
- (IDRD) Yessad, K., 2018: Sphere to sphere transforms in spectral space in the cycle 46t1 of ARPEGE/IFS (internal note).
- (IDLAM) Zagar, M., and C. Fischer, 2007: The ARPEGE/ALADIN Tech'Book: Implications of LAM aspects on the global model code for CY33/AL33. Internal note, 31pp, available on the internet server "http://www.umr-cnrm.fr/gmapdoc/".