# Coding norms & parallelization

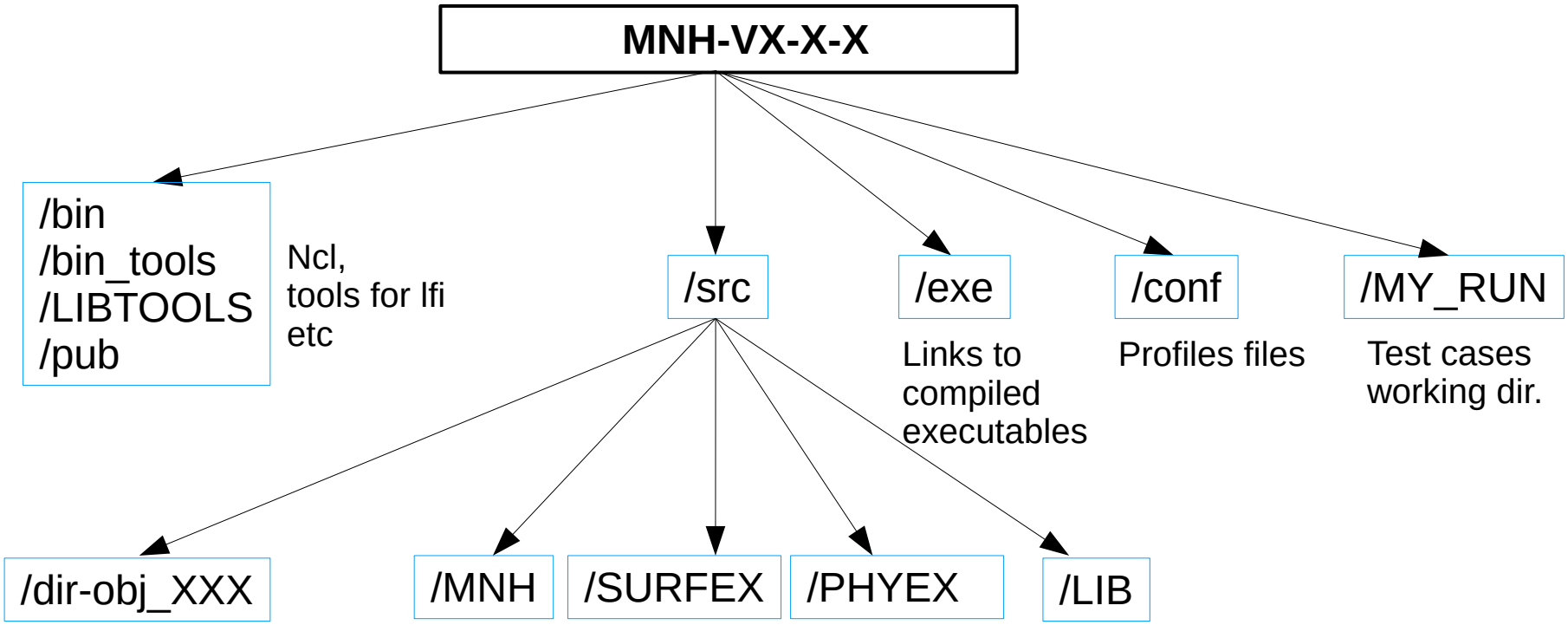
Méso-NH tutorial
12-15 November 2024

# Outline

- **How to read the code?** *(users)*
  - Tree
  - Fortran 95 norms
  - MesoNH norms
- **How to develop in Meso-NH?** *(developers)*
  - Basic rules
  - GIT
  - Specificities for PHYEX
- **Parallelization**
  - **Principles**
  - **Large grids and scalability**
  - **Reproducibility**
- **I/O**
- **FAQ**

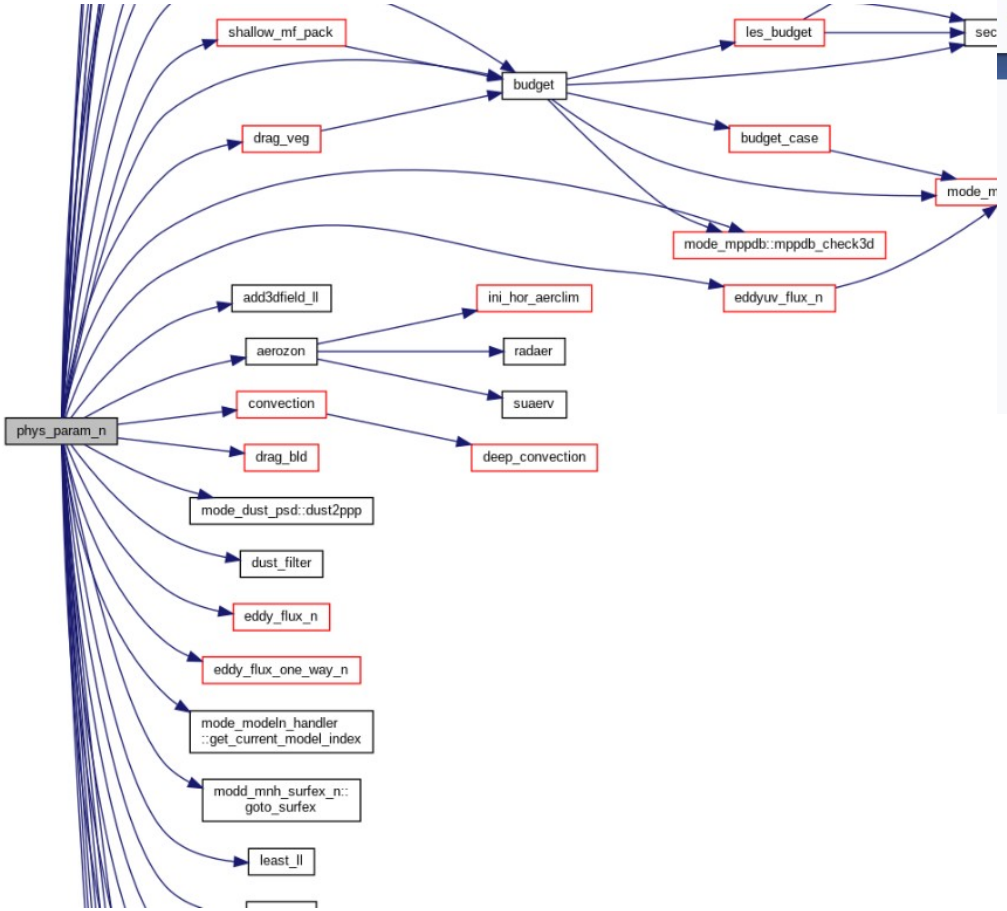
METEO FRANCE

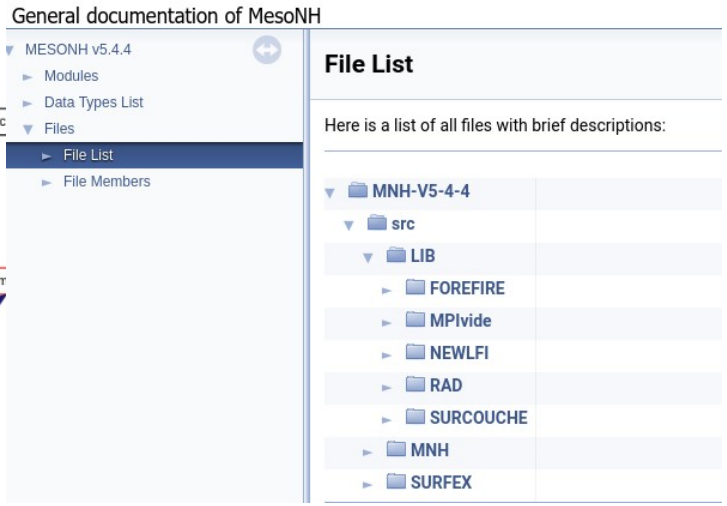
# Tree of the PACK



# Tree of the code: Doxygen

- **Generate the tree yourself**




Guide: mesonh.aero.obs-mip.fr/mesonh57/doxygenTree

METEO FRANCE

# Read the code

- ## Fortran 95
  - up to 132 characters per line
  - No blank line (use !)
  - Continuation character &
  - CODE IN CAPITAL LETTERS! comments in small letters

- ## Variable declaration
  - **Global**: used everywhere, declaration in module
  - **Dummy argument** in routine/function

e.g.: REAL, DIMENSION(:,:), INTENT(INOUT) :: PUT  ! u-wind at time t

  - **Local**: only known in the routine

  - All variables must be declared: IMPLICIT NONE

METEO FRANCE

# Read the code: name of variables

- **DOCTOR norm convention**

| Type Status | INTEGER | REAL | LOGICAL | CHARACTER | TYPE |
|---|---|---|---|---|---|
| Global | **N** | **X** | **L** (not LP) | **C** | **T** (not TP,TS,TZ) |
| Dummy argument | **K** | **P** (not PP) | **O** | **H** | **TP** |
| Local | **I** (not IS) | **Z** (not ZS) | **G** (not GS) | **Y** (not YS, YP) | **TZ** |
| Loop control | **J** (not JP) | - | - | - | - |

METEO FRANCE

# Read the code: name of variables

- **Example: dummy argument variables**

```
INTEGER,                  INTENT(IN)   :: KKA           !near ground array index
INTEGER,                  INTENT(IN)   :: KKU           !uppest atmosphere array index
INTEGER,                  INTENT(IN)   :: KKL           !vert. levels type 1=MNH -1=ARO
INTEGER,                  INTENT(IN)   :: KRR           ! number of moist var.
INTEGER,                  INTENT(IN)   :: KRRL          ! number of liquid water var.
INTEGER,                  INTENT(IN)   :: KRRI          ! number of ice water var.
LOGICAL,                  INTENT(IN)   ::  OCLOSE_OUT   ! switch for syncronous
                                                        ! file opening
LOGICAL,                  INTENT(IN)   ::  OTURB_FLX    ! switch to write the
                                  ! turbulent fluxes in the syncronous FM-file
CHARACTER(LEN=80),        INTENT(IN)   ::  HTURBDIM     ! dimensionality of the
                                                        ! turbulence scheme
CHARACTER(LEN=80),        INTENT(IN)   ::  HTOM         ! type of Third Order Moment
REAL,                     INTENT(IN)   ::  PIMPL, PEXPL ! Coef. for temporal disc.
REAL,                     INTENT(IN)   ::  PTSTEP       ! Double Time Step
TYPE(TFILEDATA),          INTENT(IN)   ::  TPFILE       ! Output file
!
REAL, DIMENSION(:,:,:), INTENT(IN)   ::  PDZZ, PDXX, PDYY, PDZX, PDZY
                                                        ! Metric coefficients
REAL, DIMENSION(:,:),   INTENT(IN)   ::  PDIRCOSZW    ! Director Cosinus of the
                                                        ! normal to the ground surface
REAL, DIMENSION(:,:,:), INTENT(IN)   ::  PZZ          ! altitudes
!
```

# Read the code: modules

- **Module = units of compilation**

  MODULE name_mod

  declarations

  END MODULE name_mod

  - Use in another subroutine: **USE** name_mod

- **4 types**

  — MODD_xxx: variables Declarations

  - For MODD, we encourage to select the variables needed

    USE MODD_xxx, **ONLY** : var1, var2

  — MODN_xxx: Namelist declaration

  — MODE_xxx: Functions

  — MODI_xxx: included in the routine Interface (check at the call that arguments are correct) ⇒ will be removed soon (created at compilation)

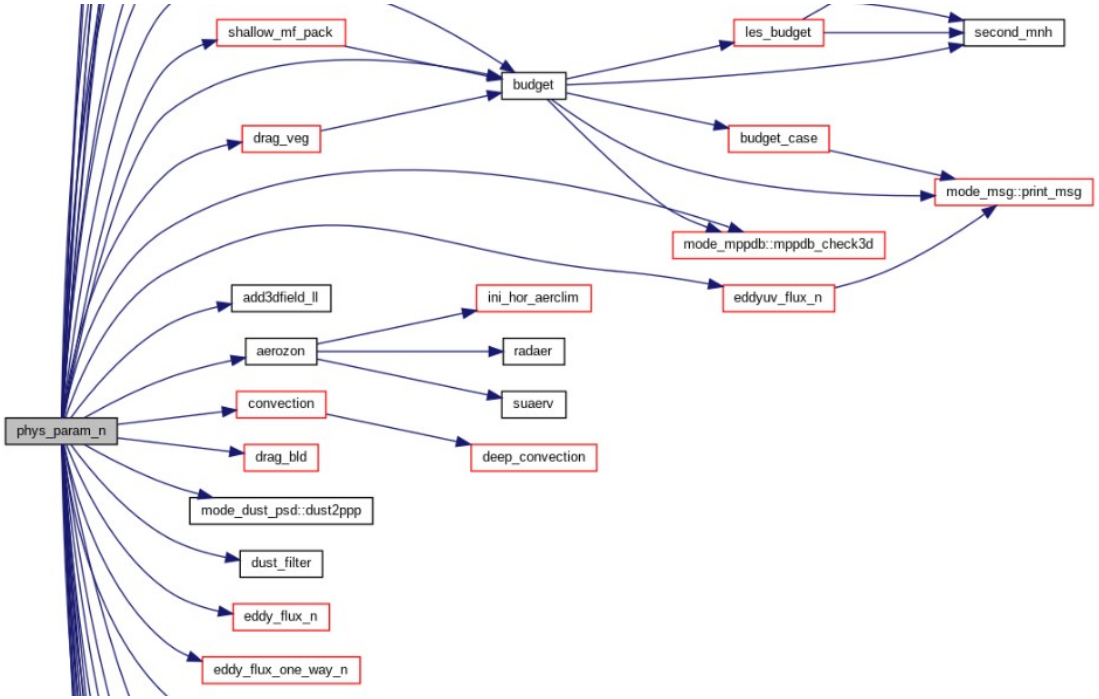
METEO FRANCE

# Read the code: modules

- **Examples**
  - **MODD**: modd_blank.f90
  - **MODN**: modn_backup.f90
  - **MODE**: mode_thermo.f90
  - **MODI**: spll_modi_phys_param_n.f90 (in */ src/dir-obj…./MASTER/)*
    - created from phys_paramn.f90 (/src/MNH)

# Read the code: _n files

- **xxxn files are for grid-nesting**
  - e.g. phys_paramn.f90, modn_advn.f90
  - Routines called X times for X sub-models
  - Routines without _n cannot use modules with _n
  - Variables relative to the horizontal domain (i,j) must be introduced by dummy arguments



# Outline

- How to read the code? *(users)*
  - Tree
  - Fortran 95 norms
  - MesoNH norms
- **How to develop in Meso-NH?** *(developers)*
  - Basic rules
  - GIT
  - Specificities for PHYEX
- Parallelization
  - Principles
  - Large grids and scalability
  - Reproducibility
- FAQ

METEO FRANCE

# Develop in MNH: good practices

- **Array**
  - Dynamical allocation

```
REAL, DIMENSION(:,:,:), ALLOCATABLE :: ZZS
ALLOCATE(ZZS(IIU,IJU))
IF(ALLOCATED(ZZS)) …
DEALLOCATE(ZZS)
```

  - Use (:,:,:) for readability and compilation optimization

    → matrix computation

```
ZVMOD(:,:) = SQRT(ZU(:,:)**2 + ZV(:,:)**2)
```

- **Pointer**
  - Declaration + Initialization (with NULLIFY() or => NULL() )

```
CHARACTER(LEN=8),DIMENSION(:), POINTER   :: NAME
NAME=>NULL()
```

METEO FRANCE

# Do not duplicate code

- **If you use twice a code ⇒ create a routine/function**
  - A large number of routines is OK: avoid « CONTAINS »
  - Avoid code duplication, ~1 million lines
  - Can be used by other developers later

- **Check existing functions (in mode_)**
  - Mode_thermo ($r_v$ from RH, P, $\theta_{v,}$ saturation, etc)
  - mode_datetime: handling time
  - mode_xxx: for each dvpmnt type
  - shuman.f90: operators (mean, gradient)

METEO FRANCE

# Discrete operators

- **Mean & finite difference** USE MODI_SHUMAN **(shuman.f90)**

e.g. $\overline{\alpha}^{z}$

MXF, MXM, MYF, MYM, MZF, MZM : means

- ► Means in X,Y or Z directions for a variable at Flux or Mass point

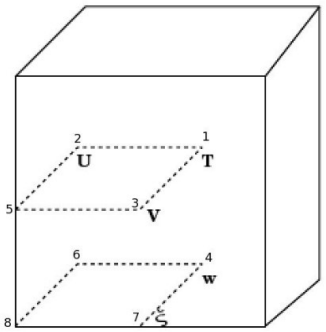
- 

e.g. $\frac{\partial \alpha}{\partial z}$

DXF, DXM, DYF, DYM, DZM, DZF

- ► Finite difference in X,Y or Z directions for a variable at Flux or Mass point

- **Gradients** USE MODI_GRADIENT_M




GX_M_M, GY_M_M, GZ_M_M

- ► Gradients along X,Y or Z for a Mass variable, results at Mass point

GX_M_U, GY_M_V, GZ_M_W

- ► Gradients along X,Y or Z for a Mass variable, results at flux U,V or W point

# Available free-to-use variables: modd_blankn

- **For testing**

```
REAL, SAVE          :: XDUMMY1, XDUMMY2, XDUMMY3, XDUMMY4, &
                       XDUMMY5, XDUMMY6, XDUMMY7, XDUMMY8
INTEGER, SAVE       :: NDUMMY1, NDUMMY2, NDUMMY3, NDUMMY4, &
                       NDUMMY5, NDUMMY6, NDUMMY7, NDUMMY8
LOGICAL, SAVE       :: LDUMMY1, LDUMMY2, LDUMMY3, LDUMMY4, &
                       LDUMMY5, LDUMMY6, LDUMMY7, LDUMMY8
CHARACTER(LEN=80), SAVE :: CDUMMY1, CDUMMY2, CDUMMY3, CDUMMY4, &
                       CDUMMY5, CDUMMY6, CDUMMY7, CDUMMY8
!
REAL,    SAVE, DIMENSION(JPDUMMY) :: XDUMMY
INTEGER, SAVE, DIMENSION(JPDUMMY) :: NDUMMY
LOGICAL. SAVE, DIMENSION(JPDUMMY) :: LDUMMY
CHARACTER(LEN=80), SAVE, DIMENSION(JPDUMMY) :: CDUMMY
!
END MODULE MODD_BLANKn
```

- **Use in your code**

USE MODD_BLANKn, ONLY : XDUMMY1

MYVAR(:,:) = MYVAR(:,:)*XDUMMY1

**In namelist**

&NAM_BLANKn  XDUMMY1=1.0 /

METEO FRANCE

# Create a new routine: format

```
!MNH_LIC Copyright 1998-2020 CNRS, Meteo-France and Universite Paul Sabatier
!MNH_LIC This is part of the Meso-NH software governed by the CeCILL-C licence
!MNH_LIC version 1. See LICENSE, CeCILL-C_V1-en.txt and CeCILL-C_V1-fr.txt
!MNH_LIC for details. version 1.
!-----------------------------------------------------------------
!     ################################
      MODULE MODI_READ_ALL_DATA_GRIB_CASE
!     ################################
INTERFACE
SUBROUTINE READ_ALL_DATA_GRIB_CASE(HFILE,TPPRE_REAL1,HGRIB,TPPGDFILE,    &
                   PTIME_HORI,KVERB,ODUMMY_REAL                          )
!
USE MODD_IO_ll, ONLY: TFILEDATA
!
CHARACTER(LEN=4),  INTENT(IN)    :: HFILE       ! which file ('ATM0','ATM1' or 'CHEM')
TYPE(TFILEDATA),POINTER,INTENT(INOUT) :: TPPRE_REAL1 ! PRE_REAL1 file
CHARACTER(LEN=28), INTENT(IN)    :: HGRIB       ! name of the GRIB file
TYPE(TFILEDATA),   INTENT(IN)    :: TPPGDFILE   ! physiographic data file
INTEGER,           INTENT(IN)    :: KVERB       ! verbosity level
LOGICAL,           INTENT(IN)    :: ODUMMY_REAL ! flag to interpolate dummy fields
REAL,              INTENT(INOUT) :: PTIME_HORI  ! time spent in hor. interpolations
!
END SUBROUTINE READ_ALL_DATA_GRIB_CASE
!
END INTERFACE
END MODULE MODI_READ_ALL_DATA_GRIB_CASE
```

Licence

Module
Interface

With declaration
of dummy arguments

```
!     ######################################################################
      SUBROUTINE READ_ALL_DATA_GRIB_CASE(HFILE,TPPRE_REAL1,HGRIB,TPPGDFILE,    &
                      PTIME_HORI,KVERB,ODUMMY_REAL                             )
!     ######################################################################
!
!!****  *READ_ALL_DATA_GRIB_CASE* - reads data for the initialization of real cases.
!!
!!    PURPOSE
!!    -------
!     This routine reads the two input files :
!       The PGD which is closed after reading
!       The GRIB file
!     Projection is read in READ_LFIFM_PGD (MODD_GRID).
!     Grid and definition of large domain are read in PGD file and Grib files.
!     The PGD files are also read in READ_LFIFM_PGD.
!     The PGD file is closed.
!     The MESO-NH domain is defined from PRE_REAL1.nam inputs in SET_SUBDOMAIN_CEP.
!     Vertical grid is defined in READ_VER_GRID.
!     PGD fields are stored on MESO-NH domain (in TRUNC_PGD).
!!
!!**  METHOD
!!    ------
!!  0. Declarations
!!    1. Declaration of arguments
!!    2. Declaration of local variables
!!  1. Read PGD file
!!    1. Domain restriction
!!    2. Coordinate conversion to lat,lon system
!!  2. Read Grib fields
!!  3. Vertical grid
!!  4. Free all temporary allocations
!!
!!    EXTERNAL
!!    --------
!!    subroutine READ_LFIFM_PGD    : to read PGD file
!!    subroutine SET_SUBDOMAIN     : to define the horizontal MESO-NH domain.
!!    subroutine READ_VER_GRID     : to read the vertical grid in namelist file.
!!    subroutine HORIBL            : horizontal bilinear interpolation
!!    subroutine XYTOLATLON        : projection from conformal to lat,lon
!!
!!    Module     MODI_SET_SUBDOMAIN     : interface for subroutine SET_SUBDOMAIN
!!    Module     MODI_READ_VER_GRID     : interface for subroutine READ_VER_GRID
!!    Module     MODI_HORIBL            : interface for subroutine HORIBL
!!    Module     MODI_XYTOLATLON        : interface for subroutine XYTOLATLON
!!
!!    IMPLICIT ARGUMENTS
!!    ------------------
!!
!!      Module MODD_CONF      : contains configuration variables for all models.
!!         NVERB : verbosity level for output-listing
!!      Module MODD_LUNIT     : contains logical unit names for all models
!!         CLUOUT0 : name of output-listing
!!      Module MODD_PGDDIM    : contains dimension of PGD fields
```

Self doc of the routine

PURPOSE

METHOD

EXTERNAL routines and functions used

IMPLICIT ARGUMENTS

```
!!    REFERENCE
!!    ---------
!!
!!      Book 1 : Informations on ISBA model (soil moisture)
!!      "Encoding and decoding Grib data", John D.Chambers, ECMWF, October 95
!!      "A guide to Grib", John D.Stackpole, National weather service, March 94
!!
!!    AUTHOR
!!    ------
!!
!!      J. Pettre and V. Bousquet
!!
!!    MODIFICATIONS
!!    -------------
!!      Original    20/11/98
!!                  15/03/99 (V. Masson) phasing with new PGD fields
!!                  28/05/99 (V. Bousquet) bug in wind interpolated variable for
!!                                         Arpege
!!                  31/05/99 (V. Masson) set pressure points (given on a regular grid at ECMWF)
!!                           on orography points (assuming the last are included in the former)
!!                  08/03/2018 (P.Wautelet)  replace ADD_FORECAST_TO_DATE by DATETIME_CORRECTDATE
!!  Philippe Wautelet: 05/2016-04/2018: new data structures and calls for I/O
!!         Pergaud  : 2018 add GFS
!!                    01/2019 (G.Delautier via Q.Rodier) for GRIB2 ARPEGE and AROME from EPYGRAM
!!      Bielli S. 02/2019  Sea salt : significant sea wave height influences salt emission; 5 salt
!  P. Wautelet 14/03/2019: correct ZWS when variable not present in file
!  Q. Rodier   27/01/2020: switch of GRIB number ID for Orograpgy and hydrometeors in ARPEGE/AROME
!  Q. Rodier   21/04/2020: correction GFS u and v wind component written in the right vertical ord
!  Q. Rodier   02/09/2020 : Read and interpol geopotential height for interpolation on isobaric su
!-------------------------------------------------------------------------------
!
!*      0. DECLARATIONS
!------------
!
USE MODE_DATETIME
USE MODE_FM,    ONLY: IO_FILE_CLOSE_ll
USE MODE_IO_ll, ONLY: UPCASE
!
USE MODI_READ_HGRID_n
!
USE MODD_FIELD_n, ONLY: XZWS, XZWS_DEFAULT
USE MODD_IO_ll, ONLY: TFILEDATA
!
USE GRIB_API
!
IMPLICIT NONE
!
!* 0.1. Declaration of arguments
!       ------------------------
!
CHARACTER(LEN=4),  INTENT(IN)    :: HFILE       ! which file ('ATM0','ATM1' or 'CHEM')
TYPE(TFILEDATA),POINTER,INTENT(INOUT) :: TPPRE_REAL1! PRE_REAL1 file
CHARACTER(LEN=28), INTENT(IN)    :: HGRIB       ! name of the GRIB file
```

REFERENCE (scientific doc or publication)

Author(s)

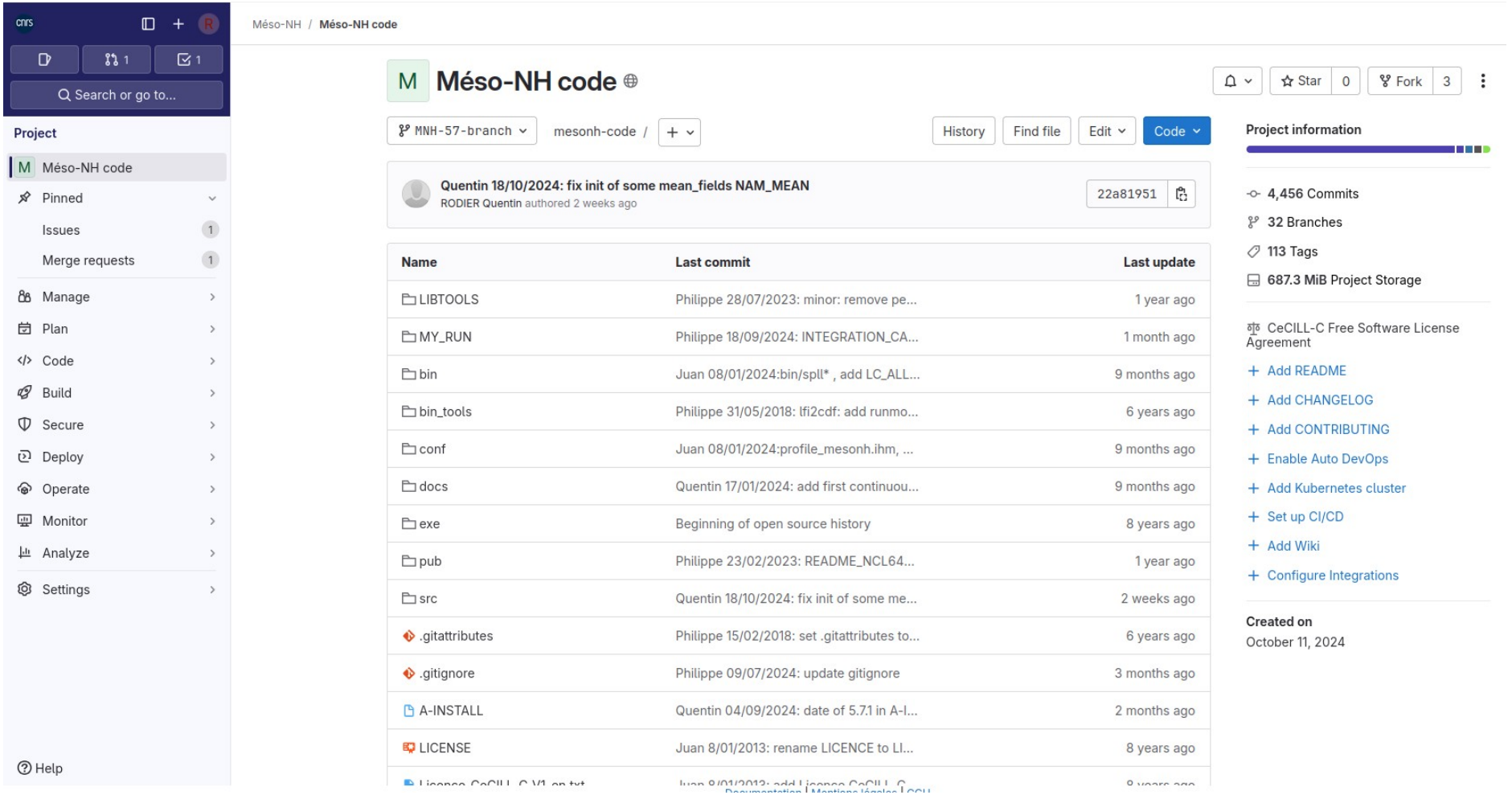
Modifications

0. Declarations
Use modules

Declarations
of arguments

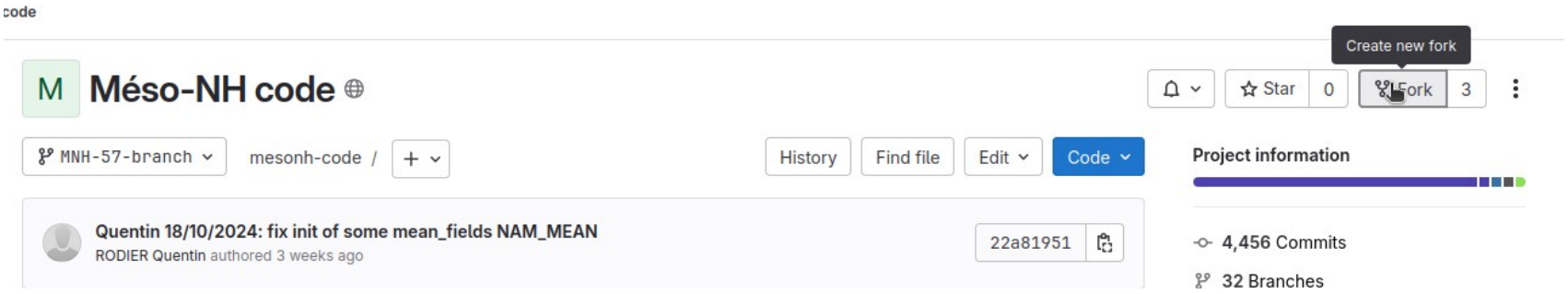
# Versioning with GIT

- **We use a gitlab instance (KODA)**
- git clone https://src.koda.cnrs.fr/mesonh/mesonh-code.git



# Versioning with GIT : how to contribute

- You do have a JANUS (CNRS) id
- **Create a fork from the official repo**




- **On your fork :**
  - Create your own branch
  - Develop inside, commit
  - From time to time, merge from the official branch
  - Once ready, make a merge-request

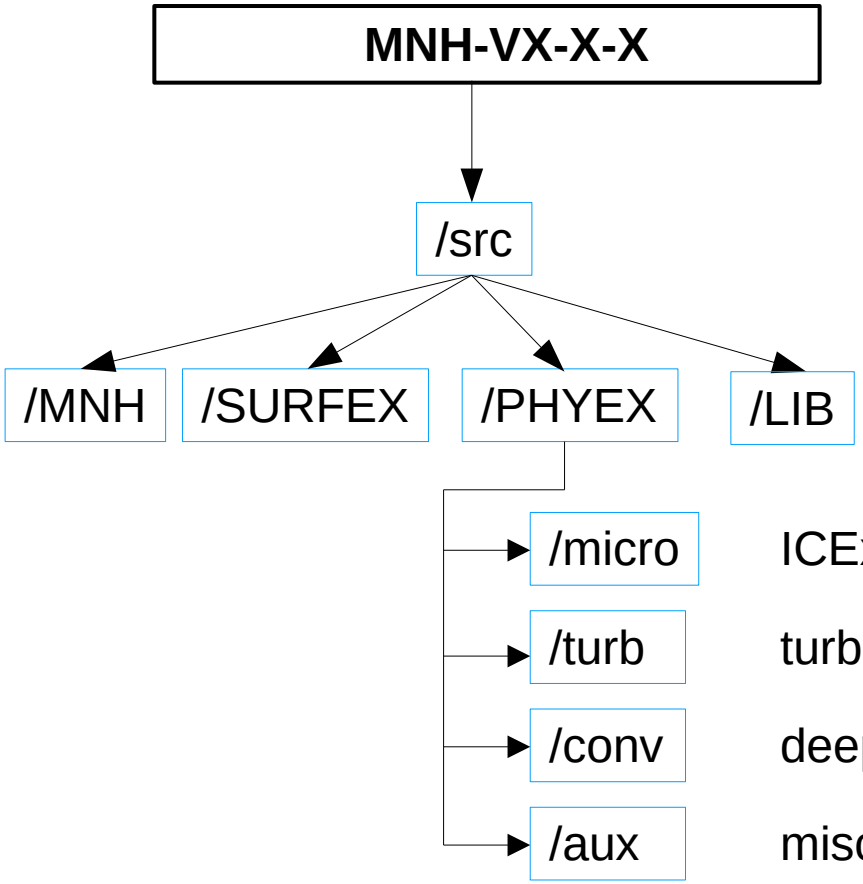
METEO FRANCE

# Versioning with GIT : how to contribute

- You **do not** have a JANUS (CNRS) id
- **Send a token request at** mesonhsupport@obs-mip.fr
- You will have written permission on your own branch only
  - Develop inside, commit
  - From time to time, merge from the official branch
  - Once ready, send us an email or @tag us on the repo

# PHYsics EXternalized

- Keep the same files for Méso-NH and AROME
- Adaptation to GPU baseline (for AROME)
- https://github.com/UMR-CNRM/PHYEX



# PHYsics EXternalized

- From 5.6.0, specific norms (more strict):
  - no allocatables : use automatic arrays,
  - dimensions of dummy argument arrays are explicit (no (:,:,:) )
    except variables declared with the PARAMETER attribute
  - no variable from modules (e.g. MODD ) can be used
    variables must be added in argument at the interface of PHYEX.
  - no functions returning arrays : use a subroutine.
  - CONTAINS subroutine included in a subroutine must not declare local variables. Declare these variables in the first subroutine.
  - horizontal dimensions are packed into one dimension (NI*NJ)

METEO FRANCE

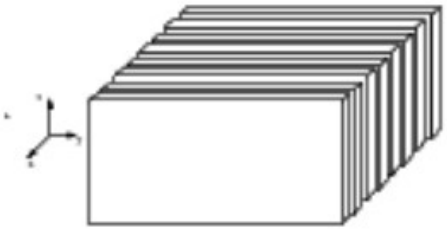
# Outline

- How to read the code? *(users)*
  - Tree
  - Fortran 95 norms
  - MesoNH norms
- How to develop in Meso-NH? *(developers)*
  - Basic rules
  - GIT
  - Specificities for PHYEX
- **Parallelization**
  - Principles
  - Large grids and scalability
  - Reproductibility
- FAQ

# Parallelization

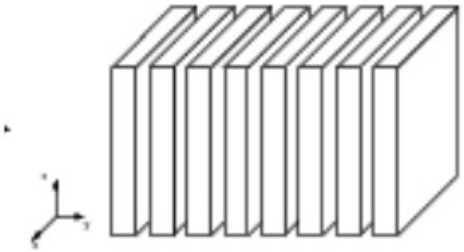
- **All the steps of Meso-NH are parallelized** (except SPECTRE)

  mpirun -np ${MPI_TASKS_TOTAL} -ppn ${MPI_TASKS_PER_NODE}



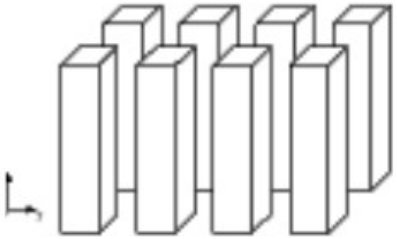
- **3 types of decomposition**
  - CSPLIT='YSPLITTING' (default option)

    Adapted to vector machines



  - CSPLIT='XSPLITTING'

    Adapted to vector machines

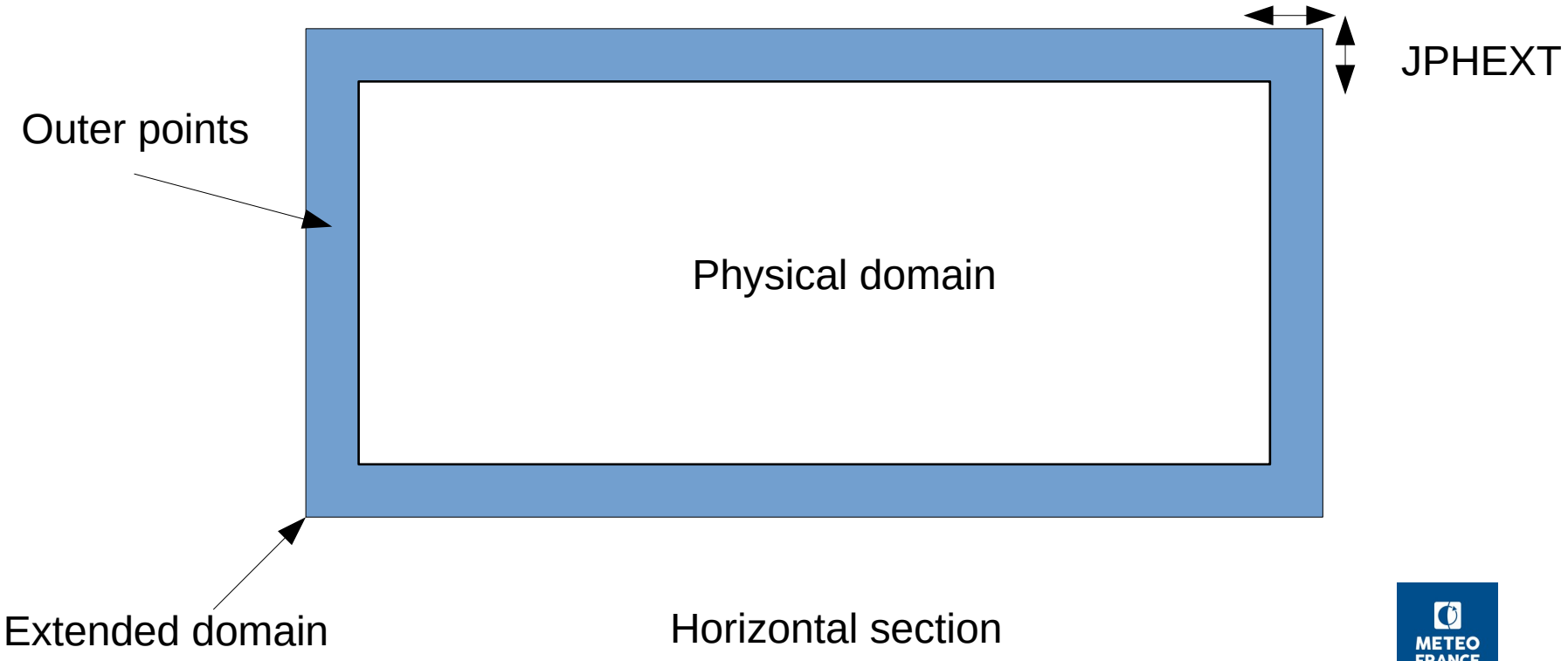


  - **CSPLIT='BSPLITTING'**

    Adapted to scalar machines with a lot of processors

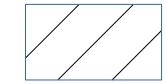
# Physical and extended domain

- **To deal with LBC, the model arrays are over-dimensioned**
  - along vertical dim. by JPVEXT points
  - along horizontal dim. by JPHEXT points
  - JPHEXT=JPVEXT=1 by default (JPHEXT=3 for WENO5 in CYCL)

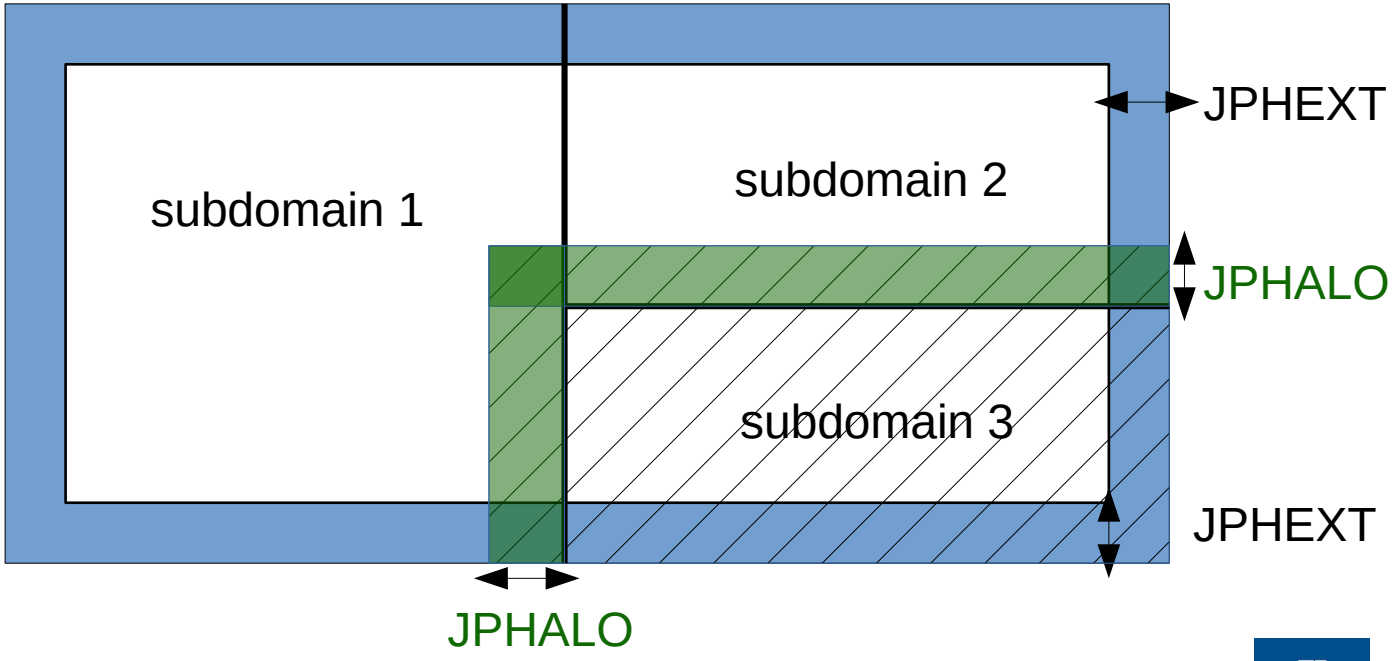


# Parallel decomposition

- **Horizontal decomposition**
  - Each subdomain is allocated on a different core
  - Finite difference requires data along the border of adjacent physical subdomains ⇒ extended domain
  - Overlap area = HALO of size JPHALO=1 (by default)



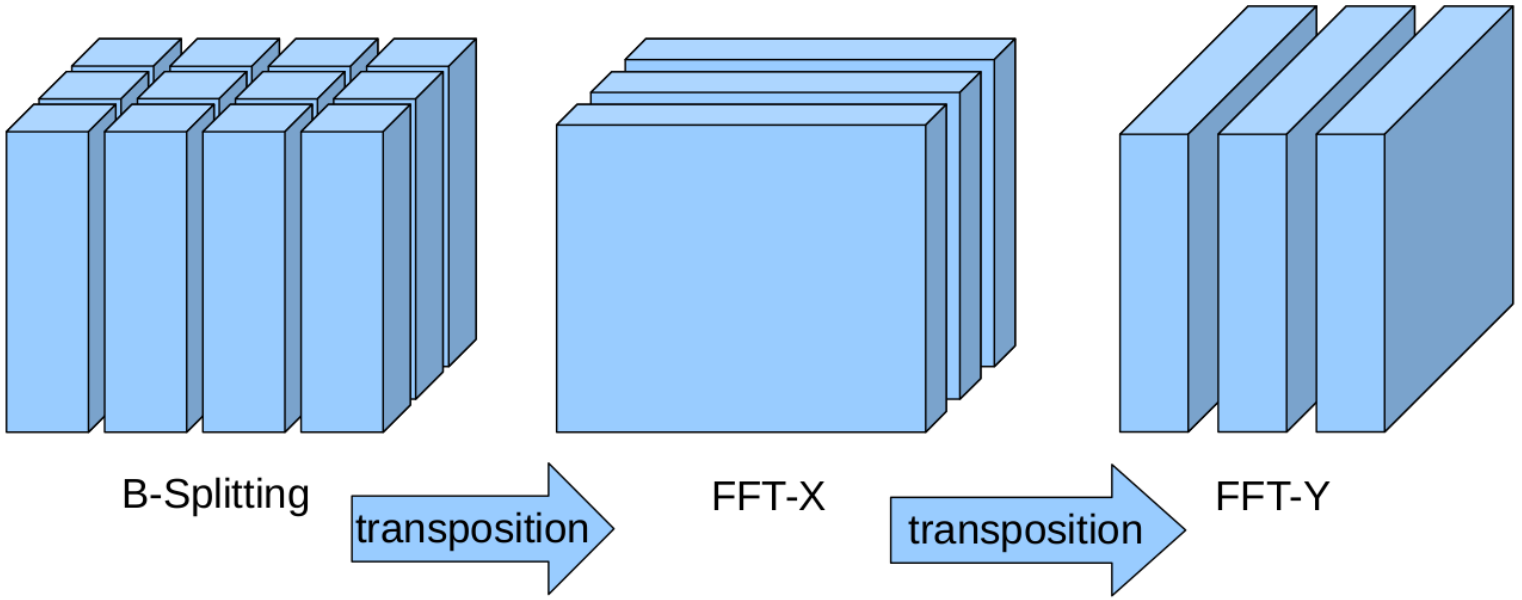
Extended subdomain




JPHALO=3 with WENO5

# Large grids

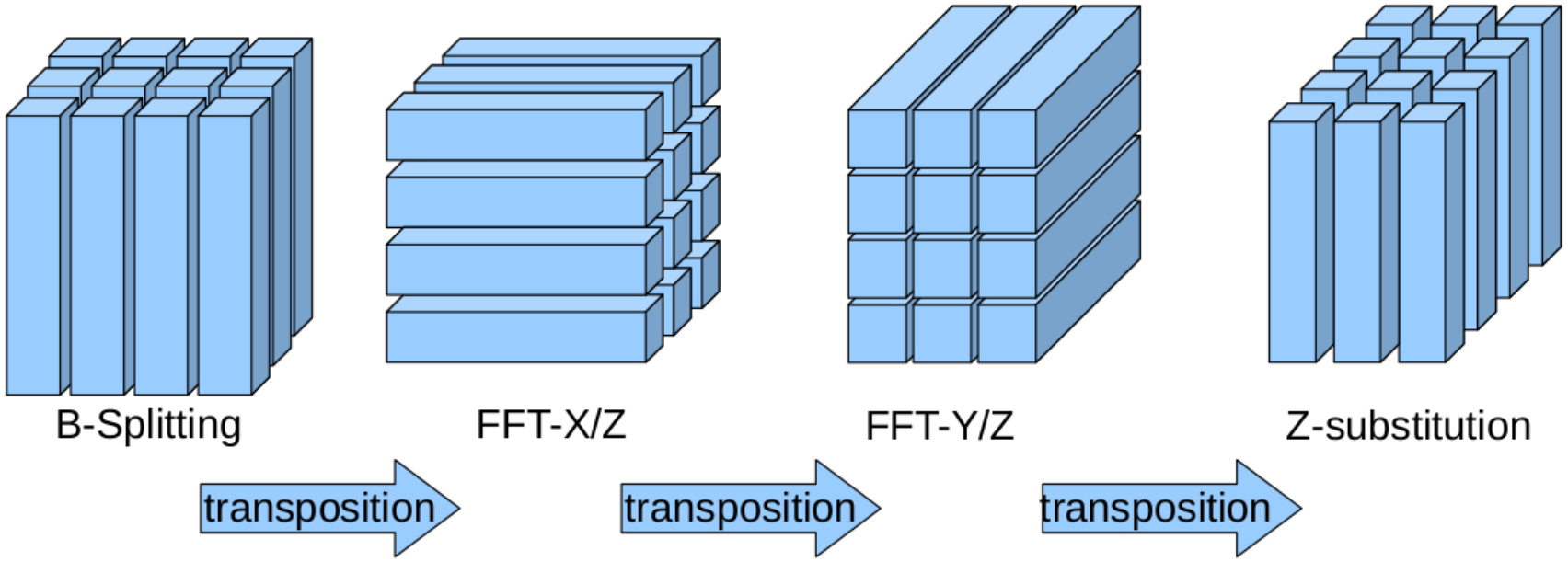
- **Before, the pressure solver was limiting** (CPRESOPT='CRESI')




  - Maximum cores = min (dimX, dimY)

# Large grids

- **Now, parallelization of the pressure solver in Z** (CPRESOPT='**Z**RESI')




  - Maximum cores = **dimX * dimY**
  - Will be the default value of CPRESOPT (not yet)

# Large grids: optimization

- **Large grids**
  - With BSPLITTING, try to set up a nb of cores such as the decomposition will give square subdomains.
  - Avoid prime numbers

- **Scalability**
  - The scalability is good down to subdomains size of **8x8** pts

    For square subdomains, we recommend to use maximum
    $(SmallDim/8)^2$ cores

    with SmallDim = the smallest dimension within all domains

  - Example: 1st domain of 256x256x80 points
    2nd domain of 128x128x80 points

    ⇒ Maximum nb of cores = $(128/8)^2$ = 256 cores

METEO FRANCE

# Online doc (books and guides)

- 7.1 Initialization
  - 7.1.1 **SET_DIM_ll**
  - 7.1.2 **SET_JP_ll**
  - 7.1.3 **SET_DAD_ll**
  - 7.1.4 **SET_LB_X_ll**
  - 7.1.5 **SET_XRATIO_ll**
  - 7.1.6 **SET_XOR_ll**
  - 7.1.7 **SET_XEND_ll**
  - 7.1.8 **INI_PARA_ll**
  - 7.1.9 **END_PARA_ll**
- 7.2 Halo
  - 7.2.1 **ADD2DFIELD_ll, ADD3DFIELD_ll**
  - 7.2.2 **ADD1DFIELD_ll**
  - 7.2.3 **DEL2DFIELD_ll, DEL3DFIELD_ll**
  - 7.2.4 **ADD_FIELD2_ll**
  - 7.2.5 **DEL_FIELD2_ll**
  - 7.2.6 **UPDATE_HALO_ll** → To update the halo with the values computed by the neighboring subdomains
  - 7.2.7 **UPDATE_1DHALO_ll**
  - 7.2.8 **UPDATE_HALO2_ll**
  - 7.2.9 **UPDATE_BOUNDARIES_ll**
- 7.3 Data distribution
  - 7.3.1 **REMAP_2WAY_X_ll**
  - 7.3.2 **REMAP_X_Y_ll**
  - 7.3.3 **REMAP_Y_X_ll**
  - 7.3.4 **REMAP_X_2WAY_ll**
  - 7.3.5 **EXTRACT_ll**
  - 7.3.6 **GET_SLICE_ll**
- 7.4 Domain informations
  - 7.4.1 **GET_DIM_EXT_ll** → To get the dimension of the extended subdomain
  - 7.4.2 **GET_DIM_PHYS_ll** → To get the dimension of the physical subdomain
  - 7.4.3 **GET_OR_ll**
  - 7.4.4 **GET_GLOBALDIMS_ll**
  - 7.4.5 **GET_INDICE_ll** → To get the origin and end coordinates of the physical local subdomain
  - 7.4.6 **GET_PHYSICAL_ll**
  - 7.4.7 **GET_INTERSECTION_ll**
  - 7.4.8 **LNORTH_ll, LWEST_ll, LSOUTH_ll, LEAST_ll** Returns a boolean which is True if the process is situated at the north... of the physical domain
- 7.5 Min Max
  - 7.5.1 **GMAXLOC_ll**
  - 7.5.2 **GMINLOC_ll**
  - 7.5.3 **MAX_ll, MIN_ll**
- 7.6 Sums
  - 7.6.1 **REDUCESUM_ll**
  - 7.6.2 **SUM_DIM1_ll, SUM_DIM2_ll**
  - 7.6.3 **SUM_1DFIELD_ll**
  - 7.6.4 **SUM_1D_ll**
  - 7.6.5 **SUM_2D_ll**
  - 7.6.6 **SUM_3D_ll**
  - 7.6.7 **SUMMASK_ll**
  - 7.6.8 **SUMMASKCOMP_ll**

METEO FRANCE

# Reproducibility

- **Repro = result independent of the number of processes**
  - On the same machine, same compiler, with an optimization level of DEBUG or O2
  - i.e. Repro. not garanteed between different machines, compilers or in O3 (aggressive compiler optimization level)

- **Help us keep the code reproducible**
  - Do not use functions such as ALL, ANY, COUNT, MAXVAL, MINVAL, SUM, MINLOC, MAXLOC ⇒ local applications
  - Use the parallelized versions **_ll** (ex. MAX_ll, SUM_3D_ll)
  - Avoid anticipated exit of a loop (EXIT, CYCLE, RETURN)

  General view: replace

  « if the process has converged for every point, we stop » (EXIT, CYCLE, RETURN)

  By

  « we compute on all the points where the process has not converged yet » (WHERE)

- **A checking tool exists (MPPDB_CHECK): ask us for help**

METEO FRANCE

# Input / Outputs

- **Jupyter Notebook tutorial and documentation**

  github.com/**PhilippeWautelet**/2022-mesonh-io-lecture/blob/main/io_mnh.ipynb

- Files structure (dimensions, netCDF groups, category)

- Main namelists and options (backup, frequent outputs, budgets, LES budgets, etc)

- How to :
  - write/read new variables
  - Handle metadata

# FAQ Frequent errors

- **Large grid / High resolution**

  1) ***Insufficient virtual memory*** (often at PREP_REAL_CASE or MESONH)

  The RSSMax value returns the memory used on the most consuming node (must be < 256GB on Belenos/Taranis)

  - Increase the number of nodes or
    decrease the number of cores by node

  - Check you have ulimit -s unlimited on bash

# FAQ Frequent errors

- **Large grid / High resolution**

2) ***INCREASE YOUR HALO_PREP IN NAM_PREP_SURF_ATM*** *(at PREP_REAL_CASE)*

***Some points lack data and are too far away from other points. Please define a higher halo value in &NAM_PGDFILE NHALO=xxx***

(at PREP_PGD)

Problem in the extra/interpolation of surface fields

SURFEX needs more point of the same COVER to interpolate prognostic fields data

- Increase the NHALO (maximum = max(dim) / 2)

  &NAM_PREP_SURF_ATM NHALO_PREP for prep

  &NAM_PGDFILE NHALO / for pgd

  Warning : increasing the NHALO increases significally the duration of the PGD/PREP

- In last cases, set NHALO/NHALO_PREP=0 : SURFEX will interpolate by using the whole domain field instead of the local field + NHALO. If not enough data is found, you must change your domain coverage

# FAQ Frequent errors

- **Large grid / High resolution**

  3) ***Error occurs in MPI_BSEND*** *or* ***MPI_ERR_BUFFER***

  (often at PREP_REAL_CASE or MESONH)

  Problem in the MPI buffer size

  The process #0 does not have enough memory to write one 3D field

  - Increase the MPI buffer size (40 by default)

    & NAM_CONFZ MPI_BUFFER_SIZE=200 /

  - Recommended value = 2 * size of a 3D field in MB

    $$\text{MPI\_BUFFER\_SIZE} = \mathbf{2 * N_x * N_y * N_z * 8 / 10^6}$$

# FAQ Frequent errors

- **Small grid**

  Use a very small number of processes, tempting to use a lot with HPC
  - SPAWNING usually works with 1 process
  - MESONH: try first on one node and a few processes

- **Read warnings in**
  - OUTPUT_LISTING files
  - .eo job outputs

- **Do not hesitate to report bugs**
  - mesonhsupport@obs-mip.fr

METEO FRANCE