THE EXTERNALIZED SURFACE USER'S GUIDE
V7.3

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

1. How to install the software

1.1. Export off-line version of SURFEX (until version 7.1 included)

Instructions to install surfex on a linux-PC and to run a 1d example.

1. **select a directory** where installation has to be done: for example $HOME or $HOME/MYDIR, where MYDIR is an existing directory (if not, it has to be created by the user).
   From now on, it is supposed that the user has defined a MYDIR directory.

2. **download** EXPORT_v7.1.gz and move it into $HOME/MYDIR/EXPORT_v7.1.gz

3. **extract files from archive**: tar zxvf EXPORT_v7.1.gz (or gunzip EXPORT_v7.1.gz and then tar xvf EXPORT_v7.1.tar) at this stage directory EXPORT_v7.1 is created in MYDIR and contains all software peaces.

4. **initialize environment variables needed for surfex**:
   1. set main environment variable:
   2. export SURFEX_EXPORT="$HOME/MYDIR/EXPORT_v7.1"
   3. then run a configuration script included in the archive:
   4. . $SURFEX_EXPORT/conf/profile_surfex

5. **run install_surfex**:
   if your linux distribution is ubuntu, don't run install_surfex but install_surfex_ubuntu.
   install_surfex / install_surfex_ubuntu:
   1. realizes the compilation of surfex sources
   2. prepares executable files for pgd, prep, offline and sxpost applications stored in $SURFEX_EXPORT/src/exe directory
      If everything goes well until this step, then surfex has been successfully installed on you computer.

6. **How to install a pre-defined experiment**:
   1. go to $SURFEX_EXPERIMENT/forcing directory
   2. run 1_compile_and_link.bash script in order to prepare executable that will then be used to install an experiment.
   3. run 2_prepare_files.bash to know the available experiments
   4. run 2_prepare_files.bash with a name of experiment as argument:
      for example "2_prepare_files.bash hapex"
      a namelist MY_PARAM.nam will open (vi editor), simply quit (use command :q)
      Some information will then be written on the screen and should look like:
      $SURFEX_EXPORT="/home/lemoigne/surfex/EXPORT_v7.1/"
      -- namelist NAM_MY_PARAM read
      > ==
      > PREP_INPUT_EXPERIMENT: YEXPER = HAPEX
      > PREP_INPUT_EXPERIMENT: INI = 1

```
> PREP_INPUT_EXPERIMENT: INPTS = 17521
> PREP_INPUT_EXPERIMENT: JNPTS = 17521
> PREP_INPUT_EXPERIMENT: ZTSTEPFRC = 1800.
> PREP_INPUT_EXPERIMENT: YFORCING_FILETYPE = NETCDF
> ==========================================
YFILE_FORCIN=../Data/hapex/HAPEX.DAT.30
-rw-r--r-- 1 lemoigne mc2 1543644 jui 22 16:51
/home/lemoigne/surfex/EXPORT_v7.0//EXP/forcing/FORCING.nc
-rw-r--r-- 1 lemoigne mc2 644 jui 22 16:51
/home/lemoigne/surfex/EXPORT_v7.0//EXP/forcing/PARAMS.nc
============================================
> input files moved to /home/lemoigne/surfex/EXPORT_v7.0//EXP/rundir/hapex
============================================
```

7. **once the installation is done, go to $SURFEX_EXPERIMENT/rundir/hapex directory and launch successively**:
    1. pgd.exe
    2. prep.exe
    3. offline.exe

8. **to view output 1d**, you can use *vespa* tool. Type vespa to get available fields and then vespa FIELDNAME to plot FIELDNAME

9. **How to rerun a pre-defined experiment with new inputs**:

    1. you can define new surface characteristics by modifying file $SURFEX_EXPERIMENT/rundir/hapex/OPTIONS.nam and then run pgd.exe, prep.exe and offline.exe
    2. you can define new initial values for state variables by modifying file $SURFEX_EXPERIMENT/rundir/hapex/OPTIONS.nam and then run prep.exe and offline.exe
    3. you can modify the forcing characteristics
        1. you can rerun $SURFEX_EXPERIMENT/forcing/2_prepapre_files.bash and modify namelist MY_PARAM to select the number of time steps you want to treat (parameter NUMBER_OF_TIME_STEPS_FINAL) the format of the input forcing files (parameter YFORCING_FILETYPE)
        2. then go to $SURFEX_EXPERIMENT/rundir/hapex and rerun pgd.exe, prep.exe and offline.exe

10. **How to create a new experiment**:
    1. you need to modify $SURFEX_EXPERIMENT/forcing/src/my_forcing.f90 to introduce the call to the new program that is going to read your dataset
    2. you need to create a new subroutine named $SURFEX_EXPERIMENT/forcing/src/my_forc_xxxx.f90 that corresponds to experiment xxxx
    3. go then to $SURFEX_EXPERIMENT/forcing and run successively:

    1. 1_compile_and_link.bash (to account for your new subroutine)
    2. 2_prepare_files.bash (to create input files related to your experiment)
    3. then go to $SURFEX_EXPERIMENT/rundir/xxxx and run pgd.exe, prep.exe and offline.exe

11. **How to compile your own source for surfex**
    1. cp the sources (from OFF_LINE or SURFEX directories) that you want to modify onto $SURFEX_EXPORT/src/MYSRC
    2. go to $SURFEX_EXPORT/src/MYSRC and make your modifications
    3. go to $SURFEX_EXPORT/src and simply launch command "make" new executable files will be created in exe directory and will replace the old ones.

12. **How to include the BUGFIX (when exists)**
    1. download BUGFIX_EXPORT_v7.1.gz
    2. extract files from archive BUGFIX_EXPORT_v7.1.gz, a directory named MYSRC and containing bug fixes will be created
    3. place extracted MYSRC in $SURFEX_EXPORT/src/ as replacement of the old one
    4. go to $SURFEX_EXPORT/src and simply launch command "make" new executable files will be created in exe directory and will replace the old ones.

1.2. Export off-line version of SURFEX (from version 7_2)

From version 7_2 of Surfex, the configuration and installation environment of off-line Surfex is adapted to this from Meso-nh.

This leads to changes in configuration and installation processes.

Instructions to install surfex on a linux-PC and to run a 1d example.

1. **select a directory** where installation has to be done: for example $HOME or $HOME/MYDIR, where MYDIR is an existing directory (if not, it has to be created by the user).
   From now on, it is supposed that the user has defined a MYDIR directory.
   Caution: none of the directory and up-directories may contain dots (.) in their names.

2. **download** EXPORT_v7_2_0.gz and move it into $HOME/MYDIR/EXPORT_v7_2_0.tar.gz (You can also get the package from SVN directly).
3. **extract files from archive**: tar zxvf EXPORT_v7_2_0.gz (or gunzip EXPORT_v7_2_0.gz and then tar xvf EXPORT_v7_2_0.tar) at this stage directory EXPORT_v7_2 is created in MYDIR and contains all software peaces.

4. **initialize environment variables needed for surfex**: go into src directory and run **./configure**.
   Then, execute the profile file for this master version of surfex: **. ../conf/profile_surfex-LXgfortran-SFX-V7-2-0-MPIAUTO-DEBUG**
5. **special case of ubuntu linux distribution**:
   if your linux distribution is ubuntu, edit the file: Makefile in src directory, comment lines: ARCHOOK="linuxgfortran" and ARCHOOK="ifort32" and uncomment lines #ARCHOOK="linuxgfortran_ubuntu" and #ARCHOOK="ifort32_ubuntu" for the compilation of DR_HOOK library.
6. **compile the master version of the code**:
   in the src directory, run **make**, and then **make installmaster**.
   Master executables are created in directory **exe**.
   If everything goes well until this step, then master surfex has been successfully installed on you computer.

7. **How to install a pre-defined experiment**:
   1. in another terminal, in src directory, do **export VER_USER=FORC**.
   2. run **./configure**.
   3. execute the profile file corresponding to this user version of surfex: **. ../conf/profile_surfex-LXgfortran-SFX-V7-2-0-FORC-MPIAUTO-DEBUG**.
   4. run **make user** and **make installuser** to create the scecific executables in directory **exe**.

5. go into **MY_RUN/FORCING** directory and run **prepare_forcing.bash** with a name of experiment as argument:
   for example "**./prepare_forcing.bash hapex**"
   a namelist MY_PARAM.nam will open (vi editor), simply quit (use command :q)
   Some information will then be written on the screen and should look like:
   $SRC_SURFEX="/home/lemoigne/surfex/EXPORT_v7_2"
   -- namelist NAM_MY_PARAM read
   > ==
   > PREP_INPUT_EXPERIMENT: YEXPER = HAPEX
   > PREP_INPUT_EXPERIMENT: INI = 1
   > PREP_INPUT_EXPERIMENT: INPTS = 17521
   > PREP_INPUT_EXPERIMENT: JNPTS = 17521
   > PREP_INPUT_EXPERIMENT: ZTSTEPFRC = 1800.
   > PREP_INPUT_EXPERIMENT: YFORCING_FILETYPE = NETCDF
   > ==
   YFILE_FORCIN=../DATA/hapex/HAPEX.DAT.30
   -rw-r--r-- 1 lemoigne mc2 1543644 jui 22 16:51
   /home/lemoigne/surfex/EXPORT_v7.2/MY_RUN/FORCING/FORCING.nc
   ==
   > input files moved to
   /home/lemoigne/surfex/EXPORT_v7_2/MY_RUN/KTEST/hapex
   ==

8. **once the installation is done, go to $SRC_SURFEX/MY_RUN/KTEST/hapex directory and launch successively**:
   1. pgd.exe
   2. prep.exe
   3. offline.exe

9. **to view output 1d**, you can use *vespa* tool. Type vespa to get available fields and then vespa FIELDNAME to plot FIELDNAME

10. **How to rerun a pre-defined experiment with new inputs**:

    1. you can define new surface characteristics by modifying file $SRC_SURFEX/MY_RUN/KTEST/hapex/OPTIONS.nam and then run pgd.exe, prep.exe and offline.exe
    2. you can define new initial values for state variables by modifying file $SRC_SURFEX/MY_RUN/KTEST/hapex/OPTIONS.nam and then run prep.exe and offline.exe
    3. you can modify the forcing characteristics
       1. you can rerun $SRC_SURFEX/MY_RUN/FORCING/prepapre_forcing.bash and modify namelist MY_PARAM to select the number of time steps you want to treat (parameter NUMBER_OF_TIME_STEPS_FINAL) the format of the input forcing files (parameter YFORCING_FILETYPE)
       2. then go to $SRC_SURFEX/MY_RUN/KTEST/hapex and rerun pgd.exe,

prep.exe and offline.exe

11. **How to create a new experiment**:
    1. you need to modify $SRC_SURFEX/src/FORC/my_forcing.f90 to introduce the call to the new program that is going to read your dataset
    2. you need to create a new subroutine named $SRC_SURFEX/src/FORC/my_forc_xxxx.f90 that corresponds to experiment xxxx
    3. then run successively:
        1. in src directory, make user (verify that $VER_USER=FORC and that corresponding profile file has been executed).
        2. in MY_RUN/FORCING directory, prepare_forcing.bash (to create input files related to your experiment)
        3. then go to $SRC_SURFEX/MY_RUN/KTEST/xxxx and run pgd.exe, prep.exe and offline.exe

12. **How to compile your own source for surfex**
    1. choose a name for your own source directory in src, for example MYSRC. Cp the sources (from OFF_LINE or SURFEX directories) that you want to modify onto $SRC_SURFEX/src/MYSRC
    2. go to $SRC_SURFEX/src/MYSRC and make your modifications
    3. go to $SRC_SURFEX/src and launch successively **export VER_USER=MYSRC**, **./configure**, **. ../conf/profile_surfex-LXgfortran-SFX-V7-2-0-MYSRC-MPIAUTO-DEBUG**, **make user** and **make installuser**. New executable files for MYSRC will be created in exe directory.

2. Overview of the externalized surface sequence

The externalized surface facilities do not contain only the program to run the physical surface schemes, but also those producing the initial surface fields (before the run) and the diagnostics (during or after the run). All these facilities are listed, below, and they separate in 4 main parts:

2.1. The sequence

1. **PGD** (routine pgd_surf_atm.f90): this program computes the physiographic data file (called PGD file below). At this step, you perform 3 main tasks:
    1. You choose the surface schemes you will use.
    2. You choose and define the grid for the surface
    3. The physiographic fields are defined on this grid.
       Therefore, the PGD file contains the spatial characteristics of the surface and all the physiographic data necessary to run the interactive surface schemes for vegetation and town.
2. **PREP** (routine prep_surf_atm_n.f90): this program performs the initialization of the surface scheme prognostic variables, as temperatures profiles, water and ice soil contents, interception reservoirs, snow reservoirs.
3. **run of the schemes** (routine coupling_surf_atm_n.f90): this performs the physical evolution of the surface schemes. It is necessary that this part, contrary to the 2 previous ones, is to be coupled within an atmospheric forcing (provided either in off-line mode or via a coupling with an atmospheric model).
4. **DIAG** (routine diag_surf_atm_n.f90): this computes diagnostics linked to the surface (e.g. surface energy balance terms, variables at 2m of height, etc...). It can be used either during the run (adding these diagnostics in the output file(s) of the run), or independantly from the run, for a given surface state (still, an instantaneous atmospheric forcing is necessary for this evaluation).

In addition, in order to read or write the prognostic variables or the diagnostics variables, respectively, in the surface files, the following subroutines are used: init_surf_atm_n.f90, write_surf_atm_n.f90 and write_diag_surf_atm_n.f90.

2.2. The atmospheric models using the externalized surface

The externalized surface can presently be used in:

1. in offline mode
2. MESONH
3. AROME

For each model, additionnal possibilities of the surface, especially the ability to read and write in files with particular formats, are added:

2.2.1. In offline mode

In this case, several types of files can be used:

- *ASCII files*, not efficient in term of storage, but completely portable.
- *TEXTE files*, not efficient in term of storage, but completely portable.
- *netcdf files*, that can be used by the program code "OFFLIN".
- *BINARY files*, increases the efficiency of the system.
- *LFI files*, increases the efficiency of the system. This special format is used in meso-NH and Arome models for surface fields.
- *FA files*, This special format is used for Arpege and Aladin models.

currently, **PGD** and **PREP** steps may be done using ASCII, LFI or FA files, and also the run produces time series of each variable (prognostic or diagnostic) in any of the formats listed above files and the output instant of the run in an ASCII, LFI or FA file.

The namelists are all included in the namelist file named OPTIONS.nam

2.2.1.1. namelist NAM_IO_OFFLINE

This namelist is the main namelist used in the off-line mode.

Fortran name	Fortran type	values	default value
CSURF_FILETYPE	string of 6 characters	"FA " , "ASCII ", "LFI "	"ASCII "
CTIMESERIES_FILETYPE	string of 6 characters	"NETCDF", "OFFLIN", "NONE ", "ASCII ", "TEXTE ", "BINARY", "FA ", "LFI "	"NONE "
CFORCING_FILETYPE	string of 6 characters	"NETCDF", "BINARY", "ASCII"	"NETCDF"
CPGDFILE	string of 28 characters		"PGD"
CPREPFILE	string of 28 characters		"PREP"
CSURFFILE	string of 28 characters		"SURFOUT"
LPRINT	logical		.FALSE.
LRESTART	logical		.FALSE.
LINQUIRE	logical		.FALSE.
XTSTEP_SURF	real		300.
XTSTEP_OUTPUT	real		1800.
LDIAG_FA_NOCOMPACT	logical		.FALSE.
LSET_FORC_ZS	logical		.FALSE.
LWRITE_COORD	logical		.FALSE.
LOUT_TILENAME	logical		.FALSE.
LLIMIT_QAIR	logical		.FALSE.
NB_READ_FORC	integer		Number of forcing time steps
LLAND_USE	logical		.FALSE.
NPROMA	integer		none
NI	integer		none
NJ	integer		none
XIO_FRAC	real		1.
YALG_MPI	character(LEN=4)	"LIN ","ADJ ","TILL","TILA"	"LIN"

- CSURF_FILETYPE: type of Surfex surface files created during PGD or PREP steps .
- CTIMESERIES_FILETYPE: type of the files containing the output diagnostic time series .
- CFORCING_FILETYPE: type of atmospheric forcing files .
- CPGDFILE: name of the PGD file .
- CPREPFILE: name of the PREP file .

* CSURFFILE: name of the final output surfex file (restart file) .
* LPRINT: write information on screen during run .
* LRESTART: write restart file .
* LINQUIRE: enable test of inquiry mode .
* XTSTEP_SURF: surface time step .
* XTSTEP_OUTPUT: time step of the output time series .
* LDIAG_FA_NOCOMPACT : fa compaction for diagnostic files.
* LSET_FORC_ZS: if T, the orography of the forcing file is set to the same value as in surface file .
* LWRITE_COORD: enables write of fields XLAT and XLON in output file .
* LOUT_TIMENAME: change name of output file at the end of the day .
* LLIMIT_QAIR: General flag for coherence between forcing Qair and calculated Qsat(Tair).
* NB_READ_FORC: subdivisions of the reading of forcings. Can vary from 1 (all forcing data read in one time) to the total number of forcing time steps (what was done until now). It's usefull especially for netcdf forcing files on tori and yuki.
* LLAND_USE: if LLAND_USE = .TRUE., fractions of vegtypes can be given at INIT level, by the namelist NAM_LAND_USE, and other surface parameters are calculated through ECOCLIMAP. It allows to make a restart with new fractions of vegtypes. But for the moment, the water balance is not kept in this case (it will be done in next version).
* NPROMA, NI, NJ : parameters needed for OPEN-MP offline driver from GMAP, but not used in classical offline mode (size of openMP packets, domain sizes)
* XIO_FRAC : the I/O processor/thread will be affected XIO_FRAC * number of points affected to other processors/threads.
* YALG_MPI : type of algorithm used to distribute points in case of MPI parallelization:
    * LIN : linear distribution
    * ADJ : distribution grouping geographically adjacent points
    * TILL : distribution to balance points from same tiles and types of vegetation between processors.
    * TILA : distribution that combines ADJ and TILL.

2.2.1.2. namelist NAM_LAND_USE

This namelist is needed when LLAND_USE = .TRUE. (NAM_IO_OFFLINE). The file referenced in this namelist has to be formatted as a Surfex PREP file and to contain at least 13 record: DIM_FULL, VEGTYPE_P1, ... VEGTYPE_P12.
If CFTYP_VEGTYPE = 'OFFLIN', the file is a NETCDF file and its name needs to be PARAMS.nc.

Fortran name	Fortran type	values	default value
CFNAM_VEGTYPE	character (LEN=28)	' '	
CFTYP_VEGTYPE	character (LEN=6)	'ASCII', 'FA ','LFI ', 'OFFLIN'	none

2.2.1.3. Namelist NAM_ZS_FILTER

Fortran name	Fortran type	values	default value
NZSFILTER	integer		1

- NZSFILTER : number of iterations of the spatial filter applied to smooth the orography (integer, 1 iteration removes the $2\Delta x$ signal, 50% of the $4\Delta x$ signal, 25% of the $6\Delta x$ signal, etcÂ [1]

2.2.1.4. Namelist NAM_NACVEG

declaration of keys for ISBA assimilation scheme (2DVAR, Bouyssel et al.)

Fortran name	Fortran type	values	default value
NECHGU	integer		6
RCLIMCA	real		0.
RCLISST	real		0.05
SIGH2M0	real		0.1
SIGT2M0	real		1.0
SIGWG0	real		
SIGWGB	real		0.06
SIGW2B	real		0.03
LOBSWG	logical		
LOBS2M	logical		F
LIMVEG	logical		T
SPRECIP2	real		4.0
RTHR_QC	real		3.0
SIGWG0_MAX	real		6.0
RSCAL_JAC	real		4.0
LPRINT	logical		T
LAROME	logical		T

2.2.2. in MESONH

In this case, MESONH FM files are used. The parallelization of the surface fields is done during the reading or writing of the fields by the FMREAD and FMWRIT routines.

2.2.2.1. Initialization of surface fields integrated in MESONH programs

In MESONH, there are usually 2 ways to produce initial files, depending if you want to use real or ideal atmospheric conditions. However, from the surface point of view, there is no difference between these 2 main possibilities of fields (real -e.g. from operationnal surface scheme in an operationnal model- or ideal -e.g. uniform-), whatever the treatment done for the atmospheric fields. This is allowed because the same externalized routines corresponding to PGD and PREP are used:

In the case of realistic atmospheric fields, the MESONH programs calling the surface are:

1. PREP_PGD : it uses the **PGD** facility of the surface
2. PREP_NEST_PGD : surface fields are only read and rewritten, except the orography that is modified (the modification of the orography itself is considered as an atmospheric model routine, as orography is also a field of the atmospheric model).
3. PREP_REAL_CASE : it uses the **PREP** facility of the surface, that can produce either ideal or realistic surface fields.
4. SPAWNING : it does not produce surface fields any more. The surface fields will be recreated during the PREP_REAL_CASE step following the SPAWNING.
   In the case of ideal atmospheric fields, the MESONH program calling the surface is PREP_IDEAL_CASE : it uses both the **PGD** and **PREP** facilities of the surface. Ideal or realistic (the latter only in conformal projection) physiographic fields can be either produced or read from a file. Then the prognostic surface variables, either ideal or realistic, can be computed by **PREP**.

If you use MESONH atmospheric model, the input and output surface files are the same as the atmospheric ones, so there is no need to specify via surface namelists any information about the input or output file names.

2.2.2.2. Namelist NAM_PGDFILE

Note however that, in PREP_PGD (just before the call to the surface physiographic computation in PGD, for which the namelists are described in the next chapter), there is a namelist to define the output physiographic file:

Fortran name	Fortran type
CPGDFILE	string of 28 characters

2.2.2.3. MESONH run and diagnostics

Then, the MESONH run can be done. During this one, the diagnostics can be, or not, be computed.

In DIAG, the surface diagnostics can also be recomputed.

2.2.3. in AROME

In this case, MESONH FM files are also used, for the surface only. The parallelization of the surface fields is done during the reading or writing of the fields by parallelization routines of ALADIN atmospheric model.

3. Off-line Guide

3.1. The input files

The use of the externalized surface software in off-line mode requires the preparation of several types of file, especially the input data necessary for the run and the definition of the options specified in the namelist .

- **OPTIONS.nam** is the namelist name used in the off-line model. The same namelist is used for the PGD, PREP and RUN facilities. The description of the different namelist blocks for PGD ("The physiographic fields") and PREP ("Initialization of the prognostic fields") tools are described in the next chapter. The namelist block where functionnalities of the off-line run in terms of Input/Output is named **NAM_IO_OFFLINE**
- the princip of an off-line simulation is that the atmospheric variables are known in advance. Thus, time series of air temperature, humidity, wind speed, precipitation, pressure and radiation terms are known. These data are pre-treated in order to be written in specific files (see below) called forcing files.
- like for any model, some parameters related to the scheme have to be set and state variable have to be initialized. These two tasks are performed by mean of tools PGD and PREP which lead to create the initial file used in the simulation

3.2. forcing files

3.2.1. Forcing format in ASCII and binary cases

There are:

- one ASCII/binary file by atmospheric parameter beginning by **Forc_..**
- one ASCII configuration file named **Params_config.txt**

The forcing parameters are:

- Atmospheric temperature: Forc_TA.txt / Forc_TA.bin (K)
- Atmospheric humidity: Forc_QA.txt / Forc_QA.bin (kg/m3)
- Atmospheric pression: Forc_PS.txt / Forc_PS.bin (Pa)
- Rain precipitation: Forc_RAIN.txt / Forc_RAIN.bin (kg/m2/s)
- Snow precipitation: Forc_SNOW.txt / Forc_SNOW.bin (kg/m2/s)
- Wind speed: Forc_WIND.txt / Forc_WIND.bin (m/s)
- Wind direction: Forc_DIR.txt / Forc_DIR.bin (degrees from N, clockwise)
- Long-wave radiation: Forc_LW.txt / Forc_LW.bin (W/m2)
- direct short-wave radiation: Forc_DIR_SW.txt / Forc_DIR_SW.bin (W/m2)
- diffuse short-wave radiation: Forc_SCA_SW.txt / Forc_SCA_SW.bin (W/m2)
- flux of CO2 : Forc_CO2.txt / Forc_CO2.bin (kg/kg)

The **Forc_...** files contain a line by forcing time step. This line contains the value of the forcing parameters for each point of the user domain.

The **Params_config.txt** file contain following information:

- Y/N (only in binary case) to specify if the forcing data must be swapped
- number of points
- number of forcing time steps during the run
- forcing time step (seconds)
- year
- month
- day
- hour (seconds)
- longitude for each point of the domain (degrees)
- latitude for each point of the domain (degrees)
- altitude of each point of the domain (m)
- height of temperature forcing for each point of the domain (m)
- height of wind forcing for each point of the domain (m)

3.2.2. Forcing format in NETCDF case

There is one file : **FORCING.nc**

Dimensions:

- time
- xx
- yy

Variables:

- time(time): units = "minutes since 1986-01-01 00:00:00" (example)

- FORC_TIME_STEP: forcing time step (s)

- LON(yy,xx): longitude (degrees)
- LAT(yy,xx): latitude (degrees)

- ZS(yy,xx): surface orography (m)
- UREF(yy,xx): reference height for the wind (m)
- ZREF(yy,xx): reference height for the temperature (m)

The forcing parameters are:

- Atmospheric temperature: Tair(time,yy,xx) (K)
- Atmospheric humidity: Qair(time,yy,xx) (kg/m3)
- Atmospheric pression: PSurf(time,yy,xx) (Pa)
- Rain precipitation: Rainf(time,yy,xx) (kg/m2/s)
- Snow precipitation: Snowf(time,yy,xx) (kg/m2/s)
- Wind speed: Wind(time,yy,xx) (m/s)
- Wind direction: Wind_DIR(time,yy,xx) (degrees from N, clockwise)
- Long-wave radiation: LWdown(time,yy,xx) (W/m2)
- direct short-wave radiation: DIR_SWdown(time,yy,xx) (W/m2)
- diffuse short-wave radiation: SCA_SWdown(time,yy,xx) (W/m2)
- flux of CO2 : CO2air(time,yy,xx) (kg/kg)

3.2.3. creation of forcing files

For the preparation of forcing files, specific programs are used and are located in $SRC_SURFEX/src/FORC.

The structure of $SRC_SURFEX/MY_RUN directory part dedicated to forcing looks like:

```
MY_RUN
|-- DATA
|    |-- Alp_for_0203
|    |-- Alqueva0206
|    |-- cdp9697
|    |-- hapex
|    |-- ma01
|    |-- me93
|    |-- vl92
|-- FORCING
|    |--  prepare_forcing.bash
|-- NAMELIST
|    |-- Alp_for_0203
|    |-- Alqueva0206
|    |-- cdp9697
|    |-- hapex
|    |-- ma01
|    |-- me93
|    |-- vl92
|-- KTEST
     |-- hapex
```

- DATA directory contains subdirectories (one per experiment) in which atmospheric time-series ascii files are stored
- NAMELIST directory contains subdirectories (one per experiment) in which at least 2 namelists are stored: the first one named MY_PARAM.nam contains information related to the forcing. For example for the "hapex" experiment, MY_PARAM.nam looks like:

```
&NAM_MY_PARAM  YEXPERIMENT_NAME   = 'HAPEX'      ,
               NUMBER_GRID_CELLS  = 1                  ,
               NUMBER_OF_TIME_STEPS_INPUT  = 17521  ,
               NUMBER_OF_TIME_STEPS_FINAL  = 17521  ,
               ZATM_FORC_STEP     = 1800.              ,
               YFORCING_FILETYPE  = 'NETCDF'
/
```

- YEXPERIMENT_NAME
  is the name associated to the experiment (12 characters)
- NUMBER_GRID_CELLS
  is the grid cell number
- NUMBER_OF_TIME_STEPS_INPUT
  number of time steps of forcing serie
- NUMBER_OF_TIME_STEPS_FINAL

number of time steps used for the simulation (should be lower or equal to NUMBER_OF_TIME_STEPS_INPUT)

- ZATM_FORC_STEP
  frequency of atmospheric forcing
- YFORCING_FILETYPE
  is the type of the forcing files asked by the user:
    - **NETCDF**:
      1 file will be created: FORCING.nc
    - **ASCII or BINARY** : Params_config.txt ascii file describing the configuration of the run will be cretaed if forcing file type is ASCII or BINARY. The content of this file is :
        - ◊ number of grid cells of the domain
        - ◊ number of atmospheric time steps
        - ◊ atmospheric time step
        - ◊ year corresponding to the begining of the simulation
        - ◊ month corresponding to the begining of the simulation
        - ◊ day corresponding to the begining of the simulation
        - ◊ seconds corresponding to the begining of the simulation
        - ◊ longitudes of grid cells
        - ◊ latitudes of grid cells
        - ◊ elevation (meters) of grid cells
        - ◊ reference height for thermodynamical variables for each grid cell
        - ◊ reference height for wind for each grid cell
    - **ASCII** : 11 ascii files, one per parameter: Forc_CO2.txt, Forc_DIR.txt, Forc_PS.txt, Forc_RAIN.txt, Forc_SNOW.txt, Forc_WIND.txt,Forc_DIR_SW.txt, Forc_LW.txt, Forc_QA.txt, Forc_SCA_SW.txt, Forc_TA.txt.
    - **BINARY** : 11 binary files, one per parameter: Forc_CO2.bin, Forc_DIR.bin, Forc_PS.bin, Forc_RAIN.bin, Forc_SNOW.bin, Forc_WIND.bin,Forc_DIR_SW.bin, Forc_LW.bin, Forc_QA.bin, Forc_SCA_SW.bin, Forc_TA.bin.

All forcing files will be placed in $SRC_SURFEX/MY_RUN/KTEST/hapex/ and a consistency test between MY_PARAM.nam and OPTIONS.nam will be done in case the forcing filetype would be different.

3.2.4. installation of an experiment

Go to $SRC_SURFEX/src and type:

export VER_USER=FORC
./configure
. ../conf/profile_surfex-LXgfortran-SURFEX-V7-FORC-MPIAUTO-DEBUG
make user

This compile the additional fortran code needed to install a predefined experiment.

Then, go to $SRC_SURFEX/MY_RUN/FORCING and launch:

./prepare_forcing.bash giving the experiment name as argument.

If you want to create a new experiment named for example 'MYTEST', you'll have to modify $SRC_SURFEX/src/FORC/my_forcing.f90 program in order to refer to the new subroutine that you'll have created and that must be named my_forc_mytest.f90 and stored in $SRC_SURFEX/src/FORC. You simply have to add few lines in my_forcing.f90 program:

```
        CASE ('MYTEST      ')
                CALL MY_FORC_MYTEST(HEXPER,KNI,KNPTS,PTSTEP_FORC,            &
                                 KYEAR,KMONTH,KDAY,PTIME,                     &
                                 PLON, PLAT, PZS, PZREF, PUREF,               &
                                 PTA, PQA, PPS, PWINDSPEED, PWINDDIR,         &
                                 PDIR_SW, PSCA_SW, PLW, PRAIN, PSNOW, PCO2  )
```

Then copy my_forc_hapex.f90 into my_forc_mytest.f90, replace HAPEX by MYTEST, refer to the correct input file and adapt the reading sequence.

Create $SRC_SURFEX/MY_RUN/NAMELIST/mytest and namelist MY_PARAM.nam and OPTIONS.nam inside this directory.

When this is done:

- go to $SRC_SURFEX/src and run *make user*
- go to $SRC_SURFEX/MY_RUN/FORCING and run *prepare_forcing.bash mytest*

3.2.5. Interpolation of forcing data at the model time step

Generally, the forcing time step is longer than the model time step.

Thatâ s why a linear interpolation is realized for nearly all atmospheric parameters, except for *RAIN* and *SNOW* that are set equal to the value at the **end of the current forcing time step** (that is the mean rate of rain during the spell between the two forcing time steps).

3.3. One example of off-line surfex application

This example is based on the situation of the 25th of October 2004 at 06UTC and covers a temporal period of 24 hours. During this day an unstable weather was observed in France, especially in the Southern part.

Here, two different file formats are used as input for the externalized surface off-line software. Both are portable: the first format is netcdf and the second is the ascii one. Netcdf format has been chosen because of several participations of PILPS intercomparison projects that requires such format, due to its portability. It follows the Alma concept (proposed by Polcher in 1998).

3.3.1. netcdf format file

3.3.1.1. FORCING.nc

For this experiment, atmospheric forcing is extracted from French database named BDAP (Base de Donnees Analysees et Prevues). Data come from the analysis of surface parameters performed by Safran analysis system devoted to hydrological applications. A constant value in space is applied for each gridbox.

3.3.1.2. list of parameters

Variable name	Dimensions	Unit	Description
time	time	days/hours/minutes/seconds since YYYY-MM-DD HH:MM:SS	time
FORC_TIME_STEP		s	forcing time step
LAT	Number_of_points	degrees	latitudes
LON	Number_of_points	degrees	longitudes
UREF	Number_of_points	m	Reference_Height_for_Wind
ZREF	Number_of_points	m	Reference_Height
ZS	Number_of_points	m	surface orography
Tair	time, Number_of_points	K	air temperature
Qair	time, Number_of_points	Kg/Kg	air specific humidity
Wind	time, Number_of_points	m/s	wind speed
DIR_SWdown	time, Number_of_points	W/m2	downward direct shortwave radiation
SCA_SWdown	time, Number_of_points	w/m2	downward diffuse shortwave radiation
LWdown	time, Number_of_points	W/m2	downward longwave radiation
PSurf	time, Number_of_points	Pa	surface pressure
Rainf	time, Number_of_points	Kg/m2/s	rainfall rate
Snowf	time, Number_of_points	Kg/m2/s	snowfall rate
CO2air	time, Number_of_points	Kg/m3	CO2 concentration
Wind_DIR	time, Number_of_points	deg	wind direction

Dimensions **Number_of_points** and **time** represent respectively the total number of gridboxes in the area of interest and the number of atmospheric time steps.

The following pictures show the time evolution of forcing quantities for the integration period, over the region of interest. Each curve correspond to a grid point.

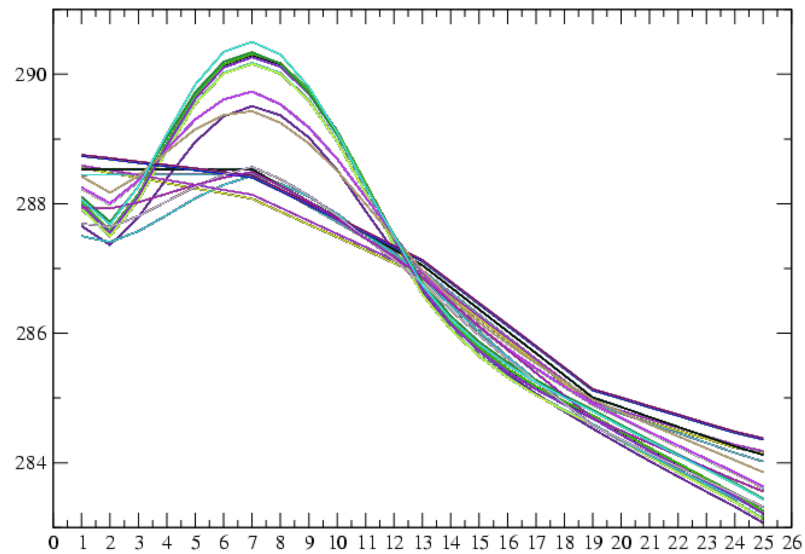


Figure 3.1: *Time evolution of temperature over the working area*

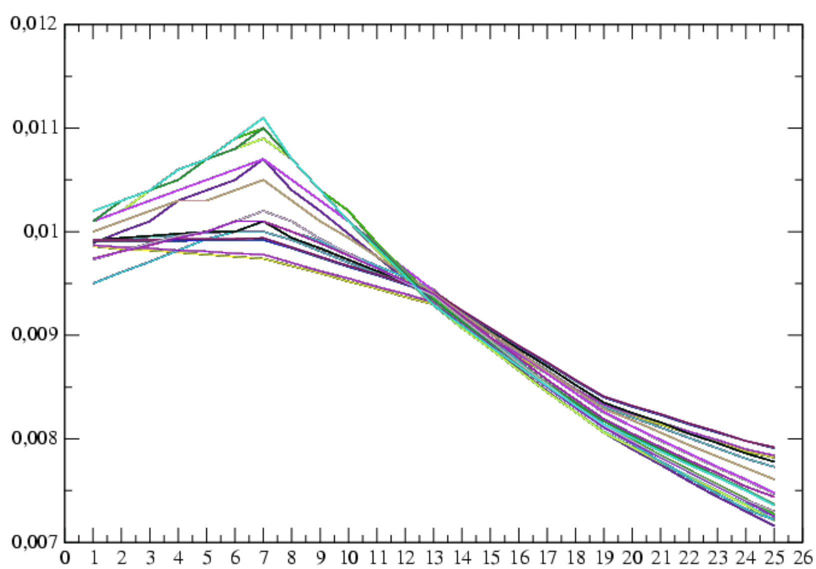


Figure 3.2: *Time evolution of specific humidity over the working area*

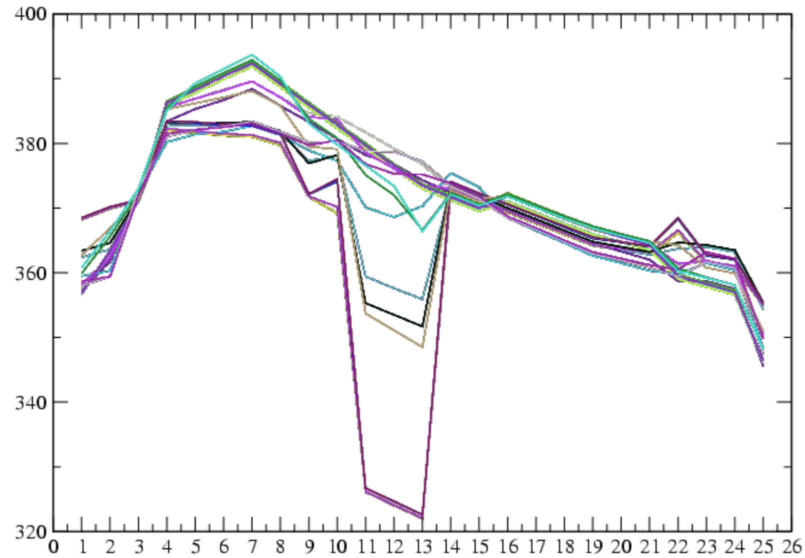


Figure 3.3: *Time evolution of longwave incoming radiation over the working area*

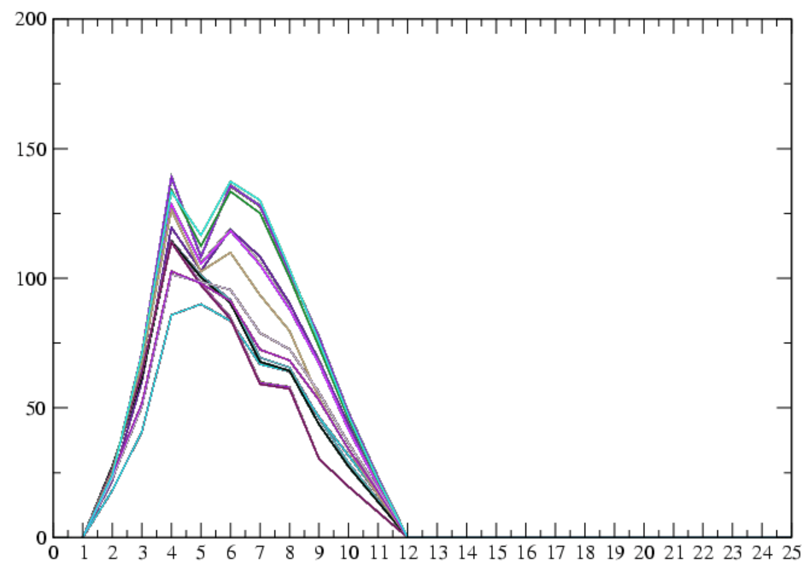


Figure 3.4: *Time evolution of shortwave incoming radiation over the working area*

3.3.2. ascii format files

This format is used in the off-line applications because it's the simpliest format that may replace more complex (in terms of file handling) like lfi format used in the meso-NH framework or FA used in the Arome framework.

3.3.2.1. PGD.txt

This file contains the information related to physiography and orography essentially. The file is splitted into several parts. The first one corresponds to the gridbox as seen as a single pixel where quantities are aggregated. The corresponding field names are prefixed with **FULL**. This gridbox may be separated into four tiles respectively associated to nature, town, sea/ocean and lake). The corresponding field names are respectively prefixed with **NATURE**, **TOWN**, **SEA** and **WATER**. The physiographic parameters written out into PGD.txt file are mainly the fraction of land covers contained in each gridbox. These fractions are computed from ECOCLIMAP database.

```
 &FULL   DIM_FULL
(-)
 1280
 &FULL   DIM_SEA
(-)
 0
 &FULL   DIM_NATURE
(-)
 1279
 &FULL   DIM_WATER
(-)
 1
 &FULL   DIM_TOWN
(-)
 240
 &FULL   ECOCLIMAP
(-)
 T
 &NATURE ISBA
ISBA
3-L
 &NATURE PHOTO
PHOTO
NON
 &NATURE GROUND_LAYER
GROUND_LAYER
 3
 &NATURE PATCH_NUMBER
PATCH_NUMBER
 1
```

Value of **DIM_FULL** indicates that this is a 2D exercise, and the gridbox contains a non-zero fraction of nature, of water (lake) and of town, but there's no fraction of sea because the number of points with a non-zero fraction of sea **DIM_SEA** is zero. 1279 gridboxes contain vegetation and 240 gridboxes contain a fraction of town, onmly one gridbow contains a fraction ok inland water(lake). Surface scheme to treat vegetation is Isba 3-L which means that soil is represented with 3 layers. The number of patches is 1, it indicates that the vegetation is not splitted into patches like it could (this should be the case for the A-gs option of Isba that treats explicitly photosynthesis).

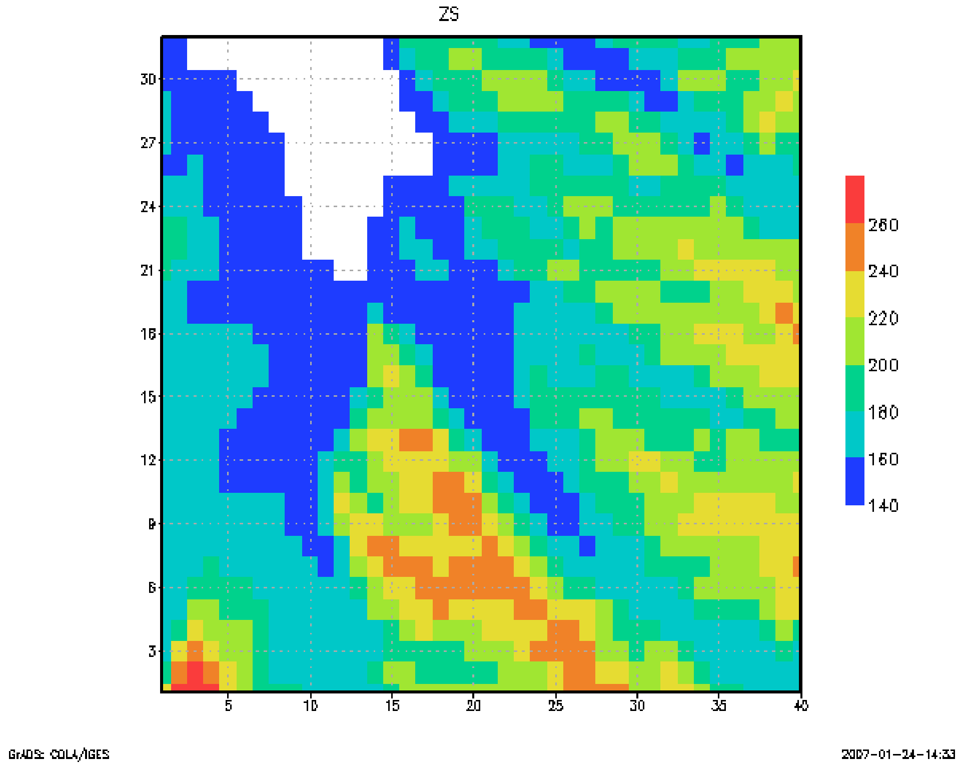


Figure 3.5: *Orography field taken from PGD.txt file*

3.3.2.2. PREP.txt

This file contains the information related to physiography and orography read from PGD.txt file and rewritten, as well as initial values of prognostic variables of the different schemes contained in SURFEX.

```
 &NATURE TG1
X_Y_TG1 (K)
   2.8576409563069382E+02   2.8548082006251650E+02   2.8540527530138650E+02   2.8546873
```

For example here are shown the first initial values of skin surface temperature over natural area for Isba surface scheme as they are written in PREP.txt file. This field is represented on figure 3.6.

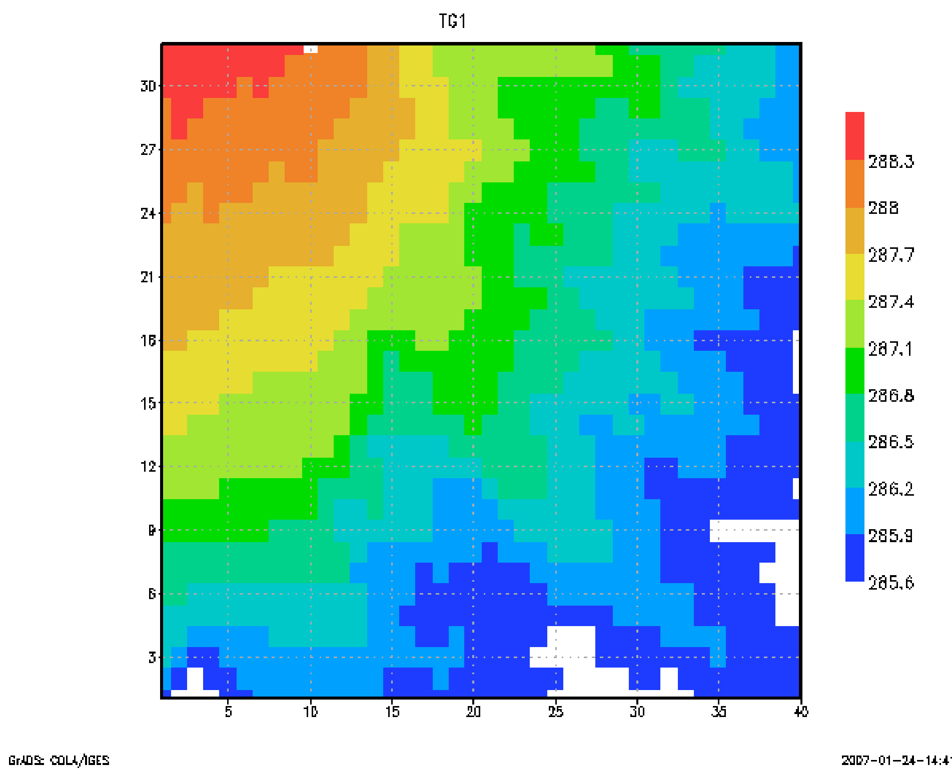


Figure 3.6: *Initial surface temperature for vegetation taken from PREP.txt file*

An other example shows the roof surface temperature over the working area (Toulouse city is located roughly at x=15, y=22)

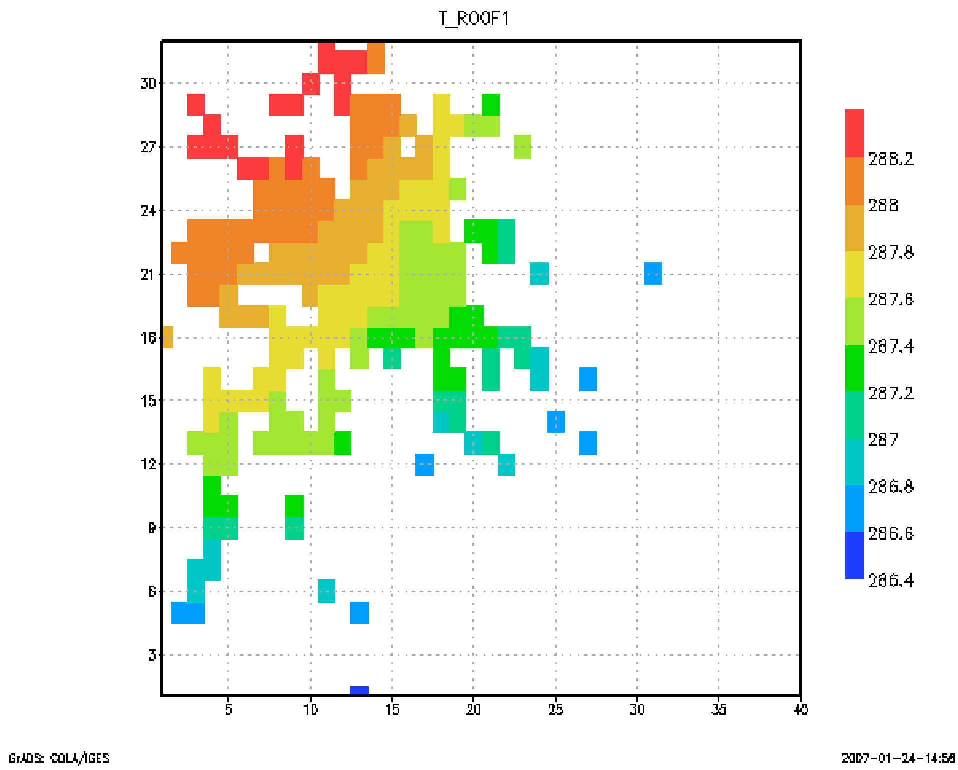


Figure 3.7: *Initial surface temperature for the roofs taken from PREP.txt file*

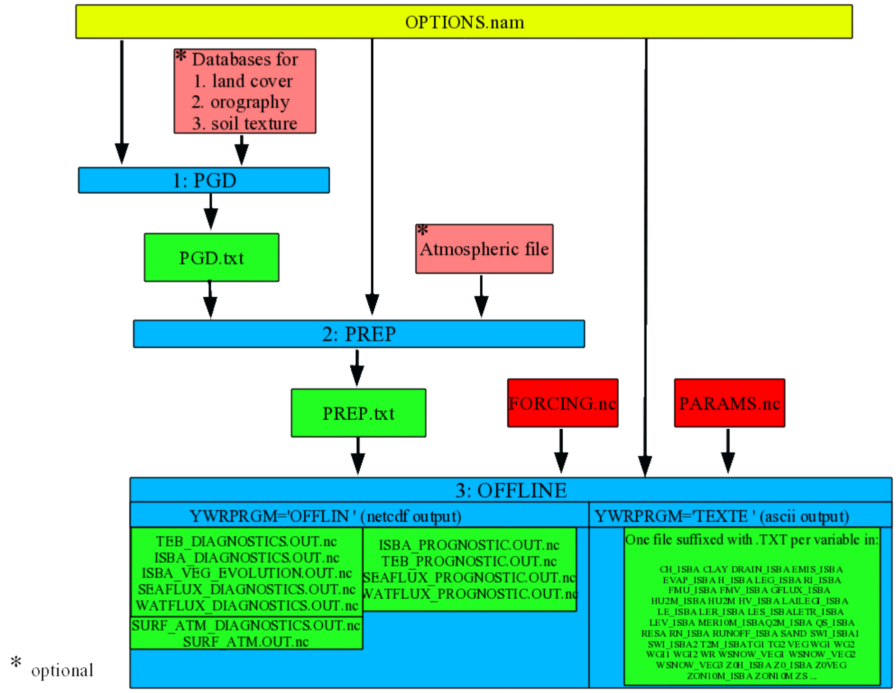
3.3.2.3. Extracting 2d fields

It's possible to extract 2d fields from PGD.txt (covers, orography, etc.) and from PREP.txt (initial prognostic variables like soil temperature profile, soil water content profile, etc.). For that purpose you need to run SXPOST tool (located at the same place as PGD, PREP and OFFLINE: $SURFEX_EXPORT/src/exe/). Input files for SXPOST are PGD.txt or PREP.txt if exists, and a namelist containing the number of fields to be extracted, the name and a flag indicating if the variable depends on patches or not. The name of a given field is the name written in PGD.txt or PREP.txt file where characters " SPMamp;" have been removed. For example, to extract orography, the name of the field is ZS in SURFEX, the mask over which it's defined can be FULL (total gridbox). To extract surface temperature over vegetation, the name will be TG1 and the mask NATURE. (grep "&" PGD.txt returns all variables of this file). To distinguish variables defined or not over patches, a flag is used: if the variable is patch dependant, the flag must be set to '+', in the contrary, it should be '-'. For example orography (ZS) doesn't depend on patches, but surface temperature (TG1) does. If the simulation uses patches and the flag is '-' then only the first patch will be treated (bare ground).
The namelist SXPOST.nam looks like:

```
2
- FULL   ZS
+ NATURE TG1
```

Running SXPOST will return a file per variable, which will contain the longitude, the latitude and the value of the field for each gridbox over which the field is defined (For example, TG1 which is known only over nature won't have a value for each gridbox of the domain).

3.3.2.4. I/O diagram



obviously not exhaustive)

Figure 3.8: *Surfex diagram showing the input/output files produced by the different tools (the list of produced variables in case YWRPRGM="TEXTE" is obviously not exhaustive)*

3.4. Some output of off-line simulation

3.4.1. Examples of prognostic variables output

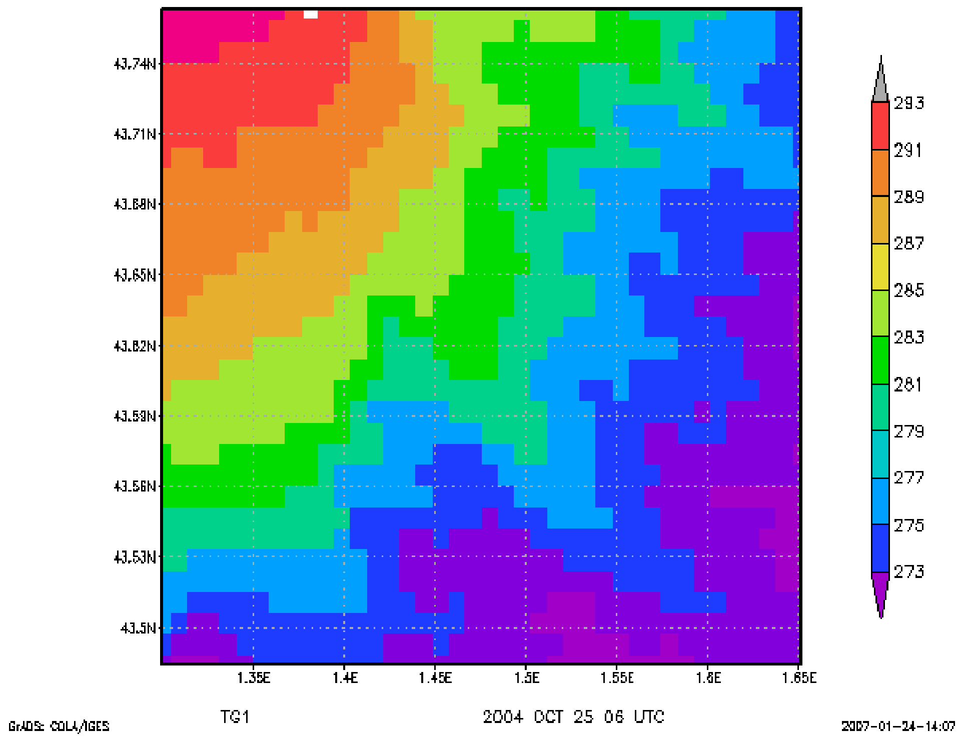


Figure 3.9: *Initial surface temperature field*

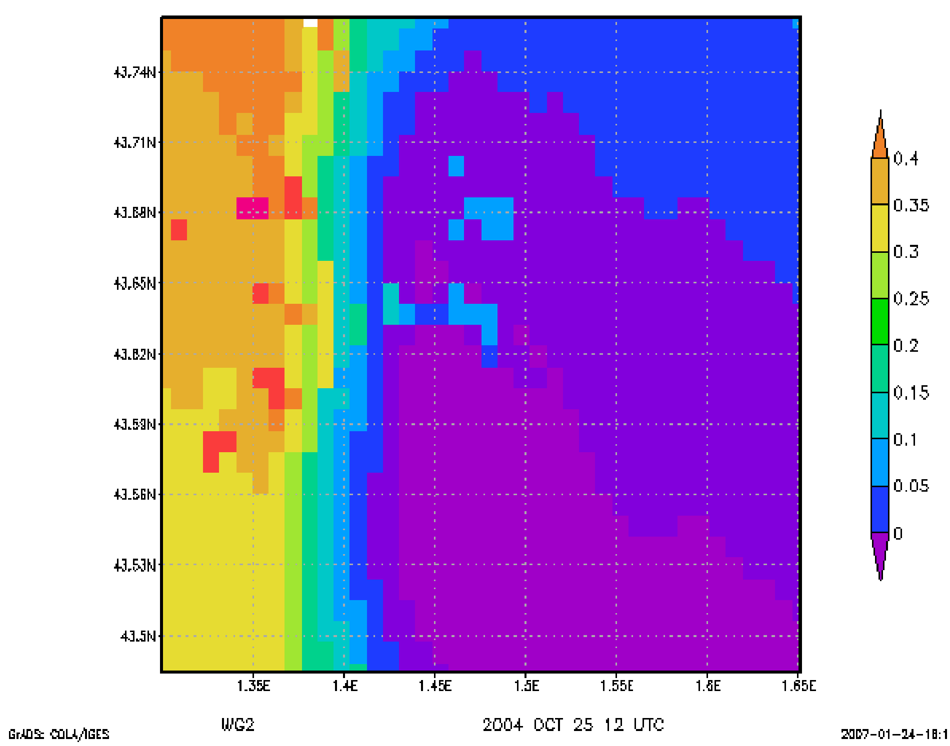


Figure 3.10: *Root layer water content after 12 hours of integration*

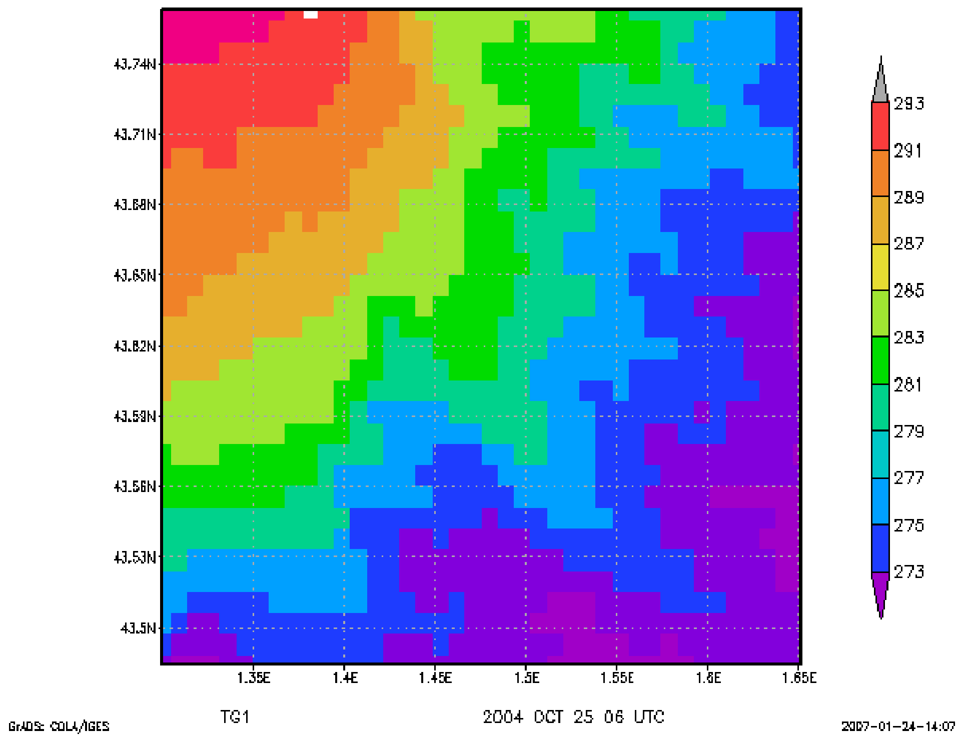


Figure 3.11: *Interception (by vegetation) water content after 12 hours of integration*

3.4.2. list of available variables

This list has been made by using vespa which is a tool that has been used to realize the plots of this document. This is not the complete possible list since only the fileds that are present in netcdf ouput files are listed.

```
********************************
*********** V E S P A ***********
********************************
***                          ***
***       V-aluable          ***
***       E-xternalized      ***
***       S-urface           ***
***       P-lot              ***
***       A-pplication       ***
***                          ***
********************************
********************************



ISBA PROGNOSTIC VARIABLES
=========================

        TG1: "Soil_temp_layer_1" ;
        WG1: "Soil_liquid_layer_1" ;
        WGI1: "Soil_ice1" ;
        TG2: "Soil_temp_layer_2" ;
        WG2: "Soil_liquid_layer_2" ;
        WGI2: "Soil_ice2" ;
        TG3: "Soil_temp_layer_3" ;
        WG3: "Soil_liquid_layer_3" ;
        WGI3: "Soil_ice3" ;
        WR: "Interception_reservoir" ;
        RESA: "Aerodynamic_resistance" ;
        WSNOW_VEG1: "Snow_Water_Equivelent_layer_1" ;
        RSNOW_VEG1: "Snow_density_layer_1" ;
        TSNOW_VEG1: "Snow_temperature_layer1" ;
        HSNOW_VEG1: "Snow_heat_layer1" ;
        WSNOW_VEG2: "Snow_Water_Equivelent_layer_2" ;
        RSNOW_VEG2: "Snow_density_layer_2" ;
        TSNOW_VEG2: "Snow_temperature_layer2" ;
        HSNOW_VEG2: "Snow_heat_layer2" ;
        WSNOW_VEG3: "Snow_Water_Equivelent_layer_3" ;
        RSNOW_VEG3: "Snow_density_layer_3" ;
        TSNOW_VEG3: "Snow_temperature_layer3" ;
        HSNOW_VEG3: "Snow_heat_layer3" ;
        ASNOW_VEG: "Snow_albedo" ;


ISBA DIAGNOSTIC VARIABLES
=========================

        RI_ISBA: "Averaged_Richardson_Number" ;
```

```
        RN_ISBA: "Averaged_Net_Radiation" ;
        H_ISBA: "Averaged_Sensible_Heat_Flux" ;
        LE_ISBA: "Averaged_Latent_Heat_Flux" ;
        GFLUX_ISBA: "Averaged_Ground_Heat_Flux" ;
        LEG: "Ground_Evaporation_Heat_Flux" ;
        LEGI: "Soil_Ice_Sublimation" ;
        LEV: "Vegetation_Evaporation_Heat_Flux" ;
        LES: "Snow_Evaporation_Heat_Flux" ;
        LER: "Canopy_Water_Interception_Evaporation" ;
        LETR: "Vegetation_Evapotranspiration" ;
        EVAP: "Evapotranspiration" ;
        DRAIN: "Soil_Drainage_Flux" ;
        RUNOFF: "Supersaturation_Runoff" ;
        LEG_ISBA: "Averaged_Ground_Evaporation_Heat_Flux" ;
        LEGI_ISBA: "Averaged_Soil_Ice_Sublimation" ;
        LEV_ISBA: "Averaged_Vegetation_Evaporation_Heat_Flux" ;
        LES_ISBA: "Averaged_Snow_Evaporation_Heat_Flux" ;
        LER_ISBA: "Averaged_Canopy_Water_Interception_Evaporation" ;
        LETR_ISBA: "Averaged_Vegetation_Evapotranspiration" ;
        EVAP_ISBA: "Averaged_Evapotranspiration" ;
        DRAIN_ISBA: "Averaged_Soil_Drainage_Flux" ;
        RUNOFF_ISBA: "Averaged_Supersaturation_Runoff" ;
        CH_ISBA: "Averaged_thermal_diffusion_coefficient" ;
        HV_ISBA: "Halstead_coefficient" ;
        Z0REL: "Output_Z0REL" ;
        VEGTYPE_PATCH_1: "fraction_of_vegetation_type_1" ;
        VEGTYPE_PATCH_2: "fraction_of_vegetation_type_2" ;
        VEGTYPE_PATCH_3: "fraction_of_vegetation_type_3" ;
        VEGTYPE_PATCH_4: "fraction_of_vegetation_type_4" ;
        VEGTYPE_PATCH_5: "fraction_of_vegetation_type_5" ;
        VEGTYPE_PATCH_6: "fraction_of_vegetation_type_6" ;
        VEGTYPE_PATCH_7: "fraction_of_vegetation_type_7" ;
        VEGTYPE_PATCH_8: "fraction_of_vegetation_type_8" ;
        VEGTYPE_PATCH_9: "fraction_of_vegetation_type_9" ;
        VEGTYPE_PATCH_10: "fraction_of_vegetation_type_10" ;
        VEGTYPE_PATCH_11: "fraction_of_vegetation_type_11" ;
        VEGTYPE_PATCH_12: "fraction_of_vegetation_type_12" ;


ISBA PHYSIOGRAPHIC VARIABLES
============================

        VEG: "Output_vegetation_fraction" ;
        Z0_ISBA: "Output_Z0_ISBA" ;
        LAI: "Output_LAI_ISBA" ;
        ALBNIR_SOIL: "Output_ALBNIR_SOIL" ;
        ALBVIS_SOIL: "Output_ALBVIS_SOIL" ;


TEB PROGNOSTIC VARIABLES
========================

        T_ROOF1: "Roof_Temperature_Layer_1" ;
        T_ROOF2: "Roof_Temperature_Layer_2" ;
        T_ROOF3: "Roof_Temperature_Layer_3" ;
        T_ROAD1: "Road_Temperature_Layer_1" ;
```

```
        T_ROAD2: "Road_Temperature_Layer_2" ;
        T_ROAD3: "Road_Temperature_Layer_3" ;
        T_WALL1: "Wall_Temperature_Layer_1" ;
        T_WALL2: "Wall_Temperature_Layer_2" ;
        T_WALL3: "Wall_Temperature_Layer_3" ;
        TI_BLD: "Internal_Building_Temperature" ;
        TI_ROAD: "Deep_Road_Temperature" ;
        WS_ROOF1: "Roof_Water_Content_Layer_1" ;
        WS_ROOF2: "Roof_Water_Content_Layer_2" ;
        WS_ROOF3: "Roof_Water_Content_Layer_3" ;
        WS_ROAD1: "Road_Water_Content_Layer_1" ;
        WS_ROAD2: "Road_Water_Content_Layer_2" ;
        WS_ROAD3: "Road_Water_Content_Layer_3" ;
        T_CANYON: "Canyon_Air_Temperature" ;
        Q_CANYON: "Canyon_Air_Humidity" ;


TEB DIAGNOSTIC VARIABLES
========================

        RI_TEB: "Averaged_Richardson_Number" ;
        CD_TEB: "Averaged_Drag_Momentum_Coef" ;
        CDN_TEB: "Averaged_Neutral_Drag_Coef" ;
        CH_TEB: "Averaged_Drag_Thermal_Coef" ;
        RESA_TEB: "Averaged_Aerodyn_Resistance" ;
        RN_TEB: "Averaged_Net_Radiation" ;
        H_TEB: "Averaged_Sensible_Heat_Flux" ;
        LE_TEB: "Averaged_Latent_Heat_Flux" ;
        GFLUX_TEB: "Averaged_Ground_Heat_Flux" ;


WATER PROGNOSTIC VARIABLES
==========================

        TS_WATER: "Averaged_Water_Temperature" ;
        Z0_WATER: "Roughness length" ;


WATER DIAGNOSTIC VARIABLES
==========================

        RI_WAT: "Averaged_Richardson_Number" ;
        CD_WAT: "Averaged_Drag_Momentum_Coef" ;
        CDN_WAT: "Averaged_Neutral_Drag_Coef" ;
        CH_WAT: "Averaged_Drag_Thermal_Coef" ;
        RESA_WAT: "Averaged_Aerodyn_Resistance" ;
        RN_WAT: "Averaged_Net_Radiation" ;
        H_WAT: "Averaged_Sensible_Heat_Flux" ;
        LE_WAT: "Averaged_Latent_Heat_Flux" ;
        GFLUX_WAT: "Averaged_Ground_Heat_Flux" ;


SURF_ATM DIAGNOSTICS VARIABLES
==============================

        RI: "Averaged_Richardson_Number" ;
```

```
        RN: "Averaged_Net_Radiation" ;
        H: "Averaged_Sensible_Heat_Flux" ;
        LE: "Averaged_Latent_Heat_Flux" ;
        GFLUX: "Averaged_Ground_Heat_Flux" ;


SURF_ATM FRACTIONS
==================

        FRAC_SEA: "Fraction_of_sea" ;
        FRAC_WATER: "Fraction_of_water" ;
        FRAC_TOWN: "Fraction_of_town" ;
        FRAC_NATURE: "Fraction_of_nature" ;


FORCING FIELDS
==============


        TA: "air temperature" ;
        QA: "air specific humidity" ;
        WIND: "wind speed" ;
        DIR_SW: "downward direct shortwave radiation" ;
        SCA_SW: "downward diffuse shortwave radiation" ;
        LW: "downward longwave radiation" ;
        PS: "surface pressure" ;
        RAIN: "rainfall rate" ;
        SNOW: "snowfall rate" ;
        CO2: "CO2 concentration" ;
        DIR_SW: "downward direct shortwave radiation" ;
        DIR: "wind direction" ;
```

4. The physiographic fields

The physiographic fields are averaged or interpolated on the specified grid by the program **PGD**. They are stored in a file, called PGD file, but only with the physiographic 2D fields, the geographic and grid data written in it.

During the PGD facility :

1. You choose the surface schemes you will use.
2. You choose and define the grid for the surface.
3. The physiographic fields are defined on this grid.

4.1. Choice of the surface schemes

You must first choose the surface schemes you will use. It is not possible, once chosen, to modify the surface schemes in the later steps (**PREP, running of the schemes, DIAG**).

Depending on the schemes you use, some additional physiographic fields will be computed if they are needed for the surface scheme chosen. For example, the ISBA scheme (used for vegetation and soil) needs the fractions of clay and sand.

4.1.1. Namelist NAM_PGD_SCHEMES

This namelist defines the four schemes that will be used, one for each type of surface (sea, inland water, town, vegetation).

Fortran name	Fortran type	values	default value
CNATURE	string of 6 characters	"NONE ", "FLUX ", "TSZ0 ", "ISBA "	"ISBA "
CSEA	string of 6 characters	"NONE ", "FLUX ", "SEAFLX"	"SEAFLX"
CWATER	string of 6 characters	"NONE ", "FLUX ", "WATFLX", "FLAKE "	"WATFLX"
CTOWN	string of 6 characters	"NONE ", "FLUX ", "TEB "	"TEB "
LGARDEN	logical		F

- CNATURE: scheme used for vegegation and natural soil covers . The different possibilities are:
    - "NONE " : no scheme used. No fluxes will be cmputed at the surface.
    - "FLUX " : ideal fluxes are prescribed. The have to be set in the fortran routine init_ideal_flux.f90.
    - "TSZ0 " : In this cheme, the fluxes are computed according to the ISBA physics, but the surface characteristics (temperature, humidity, etc...) remain constant with time.
    - "ISBA " : this is the full ISBA scheme (Noilhan and Planton 1989), with all options developped since this initial paper.
- CSEA : scheme used for sea and ocean . The different possibilities are:
    - "NONE " : no scheme used. No fluxes will be cmputed at the surface.
    - "FLUX " : ideal fluxes are prescribed. The have to be set in the fortran routine init_ideal_flux.f90.
    - "SEAFLX" : this is a relatively simple scheme, using the Charnock formula.
- CWATER : scheme used for inland water . The different possibilities are:
    - "NONE " : no scheme used. No fluxes will be cmputed at the surface.
    - "FLUX " : ideal fluxes are prescribed. The have to be set in the fortran routine init_ideal_flux.f90.
    - "WATFLX" : this is a relatively simple scheme, using the Charnock formula.
    - "FLAKE " : this is lake scheme from Mironov, 2005.
- CTOWN : scheme used for towns . The different possibilities are:
    - "NONE " : no scheme used. No fluxes will be cmputed at the surface.
    - "FLUX " : ideal fluxes are prescribed. The have to be set in the fortran routine init_ideal_flux.f90.
    - "TEB " : this is the Town Energy Balance scheme (Masson 2000), with all the susequent ameliorations of the scheme.
- LGARDEN : general flag to activate TEB_GARDEN

4.1.2. Namelist NAM_ISBA

Fortran name	Fortran type	values	default value
NPATCH	integer	between 1 and 12	1
CISBA	character (LEN=3)	'2-L', '3-L', 'DIF'	'3-L'
CPHOTO	string of 3 characters	'NON', 'AGS', 'LAI', 'AST', 'LST', 'NIT', 'NCB'	'NON'
LTR_ML	logical		F
XRM_PATCH	real	> 0. and < 1.	0.
CPEDO_FUNCTION	string of 4 characters	'CH78', 'C084', 'CP88', 'W099'	'CH78'
NGROUND_LAYER	integer	>0	1E+9
XUNIF_CLAY	real	between 0 and 1	0.33
YCLAY	character (LEN=28)		' '
YCLAYFILETYPE	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none
XUNIF_SAND	real	between 0 and 1	0.33
YSAND	character (LEN=28)		' '
YSANDFILETYPE	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none
XUNIF_RUNOFFB	real		0.5
YRUNOFFB	character (LEN=28)		' '
YRUNOFFBFILETYPE	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none
XUNIF_WDRAIN	real		0.
YWDRAIN	character (LEN=28)		' '
YWDRAINFILETYPE	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none
YCTI	character (LEN=28)		' '
YCTIFILETYPE	character (LEN=6)	'DIRECT'	none
XUNIF_SOC_TOP	real		1.E+20
XUNIF_SOC_SUB	real		1.E+20
YSOC_TOP	character (LEN=28)		' '
YSOC_SUB	character (LEN=28)		' '
YSOCFILETYPE	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none
XUNIF_PERM	real		1.E+20
YPERM	character (LEN=28)		' '
YPERMFILETYPE	character (LEN=28)		' '
XUNIF_PH	real		1.E+20
YPH	character (LEN=28)		' '
YPHFILETYPE	character (LEN=28)		' '
XUNIF_FERT	real		1.E+20
YFERT	character (LEN=28)		' '

YFERTFILETYPE	character (LEN=28)		' '
LIMP_SAND	logical		F
LIMP_CLAY	logical		F
LIMP_CTI	logical		F
LIMP_SOC	logical		F
LIMP_PERM	logical		F
XSOILGRID	real(150)		1.E+20

- NPATCH : number of patches used in ISBA. One patch corresponds to aggregated parameters. 12 patches correspond to separate energy budgets for all vegetation types present in ISBA. 3 patches correspond to bare soil types, low vegetation, trees. If CPHOTO equals 'NON' any number of patches between 1 and 12 is possible, for the other values of CPHOTO, 12 patches are required. The order and the signification of each patch is the following:
    - 1: bare ground
    - 2: rocks
    - 3: permanent snow
    - 4: deciduous forest
    - 5: conifer forest
    - 6: evergreen broadleaf trees
    - 7: C3 crops
    - 8: C4 crops
    - 9: irrigated crops
    - 10: grassland (C3)
    - 11: tropical grassland (C4)
    - 12: garden and parks
- CISBA: type of soil discretization and physics in ISBA:
    - '2-L' : force-restore method with 2 layers for hydrology
    - '3-L' : force-restore method with 3 layers for hydrology
    - 'DIF' : diffusion layer, with any number of layers
- CPHOTO:type of photosynthesis physics. The following options are currently available:
    - "NON" : none is used. Jarvis formula is used for plant transpiration.
    - "AGS" : ISBA-AGS, without evolving Leaf Area Index
    - "LAI" : ISBA-AGS, with evolving Leaf Area Index
    - "AST" : ISBA-AGS with offensive/defensive stress, without evolving Leaf Area Index
    - "LST" : ISBA-AGS with offensive/defensive stress, with evolving Leaf Area Index
    - "NIT" : ISBA-AGS with nitrogen, with evolving Leaf Area Index
- LTR_ML: to activate new radiative transfert calculation, only if CPHOTO/=NON.
- XRM_PATCH : threshol to remove little fractions of patches
- CPEDO_FUNCTION: Pedo-transfert function for DIF. The following options are currently available:
    - "CH78" : Clapp and Hornberger 1978 for BC
    - "C084" : Cosby et al. 1988 for BC
    - "CP88" : Carsel and Parrish 1988 for VG

        - W099" : Wosten et al. 1999 for VG
- NGROUND_LAYER : number of soil layer used in case of diffusion physics in the soil (CISBA = 'DIF'):
    - with CISBA = 2-L, NGROUND_LAYER default is 2
    - with CISBA = 3-L, NGROUND_LAYER default is 3
    - with CISBA= DIF and LECOCLIMAP, NGROUND_LAYER default is 14
- XUNIF_CLAY : uniform prescribed value of clay fraction.
- YCLAY: clay fraction data file name.
- YCLAYFILETYPE: type of clay data file ('DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV')
- XUNIF_SAND : uniform prescribed value of sand fraction.
- YSAND: sand fraction data file name.
- YSANDFILETYPE: type of sand data file ('DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV')
- XUNIF_RUNOFFB: uniform prescribed value of subgrid runoff coefficient.
- YRUNOFFB: subgrid runoff coefficient data file name.
- YRUNOFFBFILETYPE: type of subgrid runoff data file ('DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV')
- XUNIF_WDRAIN: uniform prescribed value of subgrid drainage.
- YWDRAIN: subgrid drainage data file name.
- YWDRAINFILETYPE: type of subgrid drainage data file ('DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV')
- YCTI: topographic indices file name.
- YCTIFILETYPE: type of topographic file ('DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV')
- XUNIF_SOC_TOP: uniform prescribed value of topsoil organic carbon (used only in CSOC=SGH in NAM_ISBAn)
- -* XUNIF_SOC_SUB: uniform prescribed value of subsoil organic carbon (used only in CSOC=SGH in NAM_ISBAn)
- YSOM_TOP: organic carbon topsoil data file name (used only in CSOC=SGH in NAM_ISBAn).
- YSOM_SUB: organic carbon subsoil data file name (used only in CSOC=SGH in NAM_ISBAn).
- YSOCFILETYPE: type of organic matter data file ('DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV') (used only in CSOC=SGH in NAM_ISBAn)
- XUNIF_PERM : uniform value of permafrost distribution (used only if CISBA=DIF)
- YPERM: file name for permafrost distribution (used only if CISBA=DIF)
- YPERMFILETYPE: permafrost distribution data file type('DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV') (used only if CISBA=DIF)
- XUNIF_PH : uniform value of soil pH (used only if LCH_NO_FLUX=T)
- YPH: file name for soil pH (used only if LCH_NO_FLUX=T)
- YPHFILETYPE: soil pH data file type ('DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV') (used only if LCH_NO_FLUX=T)
- XUNIF_FERT : uniform value of soil fertilization rate (kgN/ha/h) (used only if LCH_NO_FLUX=T)
- YFERT: file name for soil fertilisation rate (kgN/ha/h) (used only if LCH_NO_FLUX=T)
- YFERTFILETYPE: soil fertilisation rate file type (kgN/ha/h)('DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV') (used only if LCH_NO_FLUX=T)
- LIMP_SAND: reads sand fraction in an existing PGD file

- LIMP_CLAY: reads clay fraction in an existing PGD file
- LIMP_CTI: reads topographic indices in an existing PGD file
- LIMP_SOC: reads organic carbon in an existing PGD file
- LIMP_PERM: reads permafrost distribution in an existing PGD file
- XSOILGRID: uniform soil depth grid for CISBA=DIF. Default with CISBA=DIF and LECOCLIMAP is (/0.01,0.04,0.10,0.20,0.40,0.60,0.80,1.00,1.50,2.00,3.00,5.00,8.00,12.0/)

4.1.3. Namelist NAM_DATA_TSZ0

Treats TG and WG gradients as forcing variables. For that purpose, values of gradients at each time of a day are required and namelist **NAM_DATA_TSZ0** should be filled.

Fortran name	Fortran type	values	default	description	unit
NTIME	integer	1, 25	25	number of times in a day	(-)
XUNIF_DTS	real(NTIME)		-0.250	temperature gradient	K
XUNIF_DHUGRD	real		0.0		fraction

- NTIME: number of subdivisions of a day to apply forcing gradients.
- XUNIF_DTS: values of temperature gradient for each time of a day
- XUNIF_DHUGRD: values of humidity gradient for each time of a day

4.1.4. Namelist NAM_DATA_SEAFLUX

Treats SST as a forcing variable. For that purpose, several SST files at a given time are required and namelist **NAM_DATA_SEAFLUX** should be filled.

If this namelist is not given, the SST is initialized at the PREP step and remains constant during the run.

Fortran name	Fortran type	values	default	description	unit
LSST_DATA	logical		none	flag to activate this option	(-)
NTIME	integer	12	12	number of SST data	(-)
CFNAM_SST	character (LEN=28)		' '		
CFTYP_SST	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
NYEAR_SST	integer				
NMONTH_SST	integer				
NDAY_SST	integer				
XTIME_SST	real				

- LSST_DATA : flag to initialize SST from a climatology
- NTIME : number of SST input files
- CFNAM_SST: SST data file name
- CFTYP_SST: type of SST data file ('DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV')
- NYEAR_SST: year of SST data file
- NMONTH_SST: month of SST data file
- NDAY_SST: day of SST data file
- XTIME_SST: time in seconds of SST data file

How to initialise SST from external files: an example with 3 SST input files (lat, lon, value type).

```
&NAM_DATA_SEAFLUX      NTIME = 3 , LSST_DATA = T ,
                       CFNAM_SST (1)     = â  sst_1.dat                 â  , CFTYP_SS
                       CFNAM_SST (2)     = â  sst_2.dat                 â  , CFTYP_SS
                       CFNAM_SST (3)     = â  sst_3.dat                 â  , CFTYP_SS
                       NYEAR_SST(1)=1985,  NMONTH_SST(1)=12,  NDAY_SST(1)=31  , XTIME_SS
                       NYEAR_SST(2)=1986,  NMONTH_SST(2)=1 ,  NDAY_SST(2)=1   , XTIME_SS
                       NYEAR_SST(3)=1986,  NMONTH_SST(3)=1 ,  NDAY_SST(3)=2   , XTIME_SS
/
```

- XUNIF_xxx : uniform prescribed value of parameter xxx. If XUNIF_xxx is set, file CFNAM_xxx is not used.
- CFNAM_xxx: data file name associated to parameter xxx. If XUNIF_xxx is set, file

CFNAM_xxx is not used.

- CFTYP_xxx: type of sea data file ('DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV')

4.1.5. Namelist NAM_DATA_FLAKE

Over lakes, if one wants to use Flake scheme, some parameters have to be specified by the user in the namelist **NAM_DATA_FLAKE**.

Fortran name	Fortran type	values	default	advice	description	unit
				value		
XUNIF_WATER_DEPTH	real			20.	Lake depth	(*m*)
YWATER_DEPTH	character (LEN=28)		' '		filename	
YWATER_DEPTHFILETYPE	character (LEN=6)	'DIRECT', 'BINLLF', 'ASCLLV', 'BINLLV'	none			
YWATER_DEPTH_STATUS	character (LEN=28)		' '		status file name	
XUNIF_WATER_FETCH	real			1000.	wind fetch	(*m*)
YWATER_FETCH	character (LEN=28)		' '		filename	
YWATER_FETCHFILETYPE	character (LEN=6)	'DIRECT', 'BINLLF', 'ASCLLV', 'BINLLV'	none			
XUNIF_T_BS	real			286.	temperature at the outer edge of the thermally active layer of the of the bottom sediments	(*K*)
YT_BS	character (LEN=28)		' '		filename	
YT_BSFILETYPE	character (LEN=6)	'DIRECT', 'BINLLF', 'ASCLLV', 'BINLLV'	none			
XUNIF_DEPTH_BS	real			1.	depth of the sediments layer	(*m*)
YDEPTH_BS	character (LEN=28)		' '		filename	
YDEPTH_BSFILETYPE	character (LEN=6)	'DIRECT', 'BINLLF', 'ASCLLV', 'BINLLV'	none			
XUNIF_EXTCOEF_WATER	real			3.	extinction coefficient of solar radiation in water	(m^{-1})
YEXTCOEF_WATER	character (LEN=28)		' '		filename	

YEXTCOEF_WATERFILETYPE	character (LEN=6)	'DIRECT', 'BINLLF', 'ASCLLV', 'BINLLV'	none			

4.1.6. Namelist NAM_TEB

Fortran name	Fortran type	values	default value
NROAD_LAYER	integer		5
NROOF_LAYER	integer		5
NWALL_LAYER	integer		5
NFLOOR_LAYER	integer		5
NTEB_PATCH	integer		1
CBEM	character(LEN=3)	'DEF','BEM'	'DEF'
CCOOL_COIL	character(LEN=6)	'IDEAL ','DXCOIL'	'IDEAL '
CHEAT_COIL	character(LEN=6)	'IDEAL ','FINCAP'	'IDEAL '
LAUTOSIZE	logical		F
LGREENROOF	logical		F
LHYDRO	logical		F

- NROAD_LAYER : number of layers in roads
- NROOF_LAYER : number of layers in roofs
- NWALL_LAYER : number of layers in walls
- NFLOOR_LAYER :number of layers in floors
- NTEB_PATCH : number of TEB patches (corresponding to the roads orientations)
- CBEM : TEB option for the building energy model :
    - DEF : default version force-restore model from Masson et al. 2002
    - BEM : Building Energy Model Bueno et al. 2011
- CCOOL_COIL : type of cooling coil
- CHEAT_COIL : type of heating coil
- LAUTOSIZE : flag to activate autosize calculations
- LGREENROOF : logical to call ISBA from TEb for GREENROOF
- LHYDRO : urban subsoil and hydrology processes (not implemented yet)

4.2. Definition of the grid

There are 3 possibilities. 2 are always possible, one is available only if the **PGD** routine is integrated into an atmospheric model initialization facility.

1. The grid is chosen via namelists options (see below)
2. The grid is defined as a part of the grid of an already existing surface file, indicated via namelists (see below)
3. The grid is defined as being identical to the one of an atmospheric model, which is given as fortran argument in the coupling of the **PGD** surface facilities (routine PGD_SURF_ATM) into an atmospheric model initialization procedures. In this case, **all namelists that are usually used to define the surface grid are ignored**. Note that, in addition to the grid, the orography can also be given from the atmospheric file.

Note that all the namelists presented in this section are ignored if the grid is imposed, in the fortran code, from an atmospheric model. This is the case when one already have defined the atmospheric grid and one want to be sure that the surface has the same grid. For example, this is what happens in the MESONH program PREP_IDEAL_CASE (when no physiographic surface file is used).

If you are in this case, ignore all the namelists presented in this section, and only the namelists for cover and the following ones, have to be used.

4.2.1. Choice of the grid type

4.2.1.1. Namelist NAM_PGD_GRID

This namelist defines the grid type, either specified or from an existing surface file

Fortran name	Fortran type	default value
CGRID	string of 10 characters	"CONF PROJ "
YINIFILE	string of 28 characters	none
YFILETYPE	string of 6 characters	none

- CGRID: type of grid and projection . It is used **only** if a file is not prescribed (see below). The different grid possibilities are:
  - "GAUSS " : this grid is a gaussian grid (global grid, that may be stretched, rotated, ...)
  - "CONF PROJ " : this grid is a regular grid (in meters in x and y perpendicular directions) on conformal projection plan (Mercator, Lambert or polar stereographic).
  - "CARTESIAN " : this grid is a regular grid (in meters in x and y perpendicular directions), with no reference to real geographical coordinates.
  - "LONLAT REG" : this grid is defined as a regular latitude - longitude grid.
  - "LONLATVAL " : this grid is defined as a not regular latitude - longitude grid (all points and mesh sizes are defined).
  - "IGN " : this grid type contains all IGN (French National Geographical Institute) possible Lambert projections
  - "NONE " : this grid is not regular. Only the number of points and the size of each grid mesh is prescribed. There is no positioning of each point compared to any other.

- YINIFILE: name of the file used to define the grid. It is possible to define the grid as a subgrid of a previously created file. This is currently possible only for files that have a "CONF PROJ " or "CARTESIAN " grid type. The exact definition of the subgrid grid chosen is prescribed in a namelist (described below), depending on the type of grid available in the file chosen. **The use of a file has priority on the CGRID type**.
- YFILETYPE: type of the YINIFILE file, if the latter is provided. YFILETYPE must be given. The following values are currently usable:
  - "MESONH" : the file type is a MESONH file.

4.2.2. Conformal projection grids (Mercator, Lambert, Polar stereographic)

4.2.2.1. Namelist NAM_CONF_PROJ

This namelist defines the projection in case CGRID="CONF PROJ "

Fortran name	Fortran type	default value
XLAT0	real	none
XLON0	real	none
XRPK	real	none
XBETA	real	none

- XLAT0: reference latitude for conformal projection (real, decimal degrees)
- XLON0: reference longitude for conformal projection (real, decimal degrees)
- XRPK: cone factor for the projection (real):
    - XRPK=1: polar stereographic projection from south pol
    - 1>XRPK>0: Lambert projection from south pole
    - XRPK=0: Mercator projection from earth center
    - -1
    - XRPK=-1: polar stereographic projection from north pole

- XBETA: rotation angle of the simulation domain around the reference longitude (real)

4.2.2.2. Namelist NAM_CONF_PROJ_GRID

This namelists defines the horizontal domain in case CGRID="CONF PROJ ".

Fortran name	Fortran type	default
XLATCEN	real	none
XLONCEN	real	none
NIMAX	integer	none
NJMAX	integer	none
XDX	real	none
XDY	real	none

- XLATCEN: latitude of the point of the center of the domain (real, decimal degrees)
- XLONCEN: longitude of the point of the center of the domain (real, decimal degrees)
- NIMAX: number of surface points of the grid in direction *x*.
- NJMAX: number of surface points of the grid in direction *y*.
- XDX: grid mesh size on the conformal plane in *x* direction (real, meters).
- XDY: grid mesh size on the conformal plane in *y* direction (real, meters).

4.2.2.3. Namelist NAM_INIFILE_CONF_PROJ

This namelists defines the horizontal domain from an existing surface file in which grid type is "CONF PROJ ". If nothing is set in the namelist, a grid identical as the one in the file is chosen.

Fortran name	Fortran type	default value
IXOR	integer	1
IYOR	integer	1
IXSIZE	integer	YINIFILE size
IYSIZE	integer	YINIFILE size
IDXRATIO	integer	1
IDYRATIO	integer	1

- IXOR: first point I index, according to the YINIFILE grid, left to and out of the new physical domain.
- IYOR: first point J index, according to the YINIFILE grid, under and out of the new physical domain.
- IXSIZE: number of grid points in I direction, according to YINIFILE grid, recovered by the new domain. If to be used in MESONH, it must only be factor of 2,3 or 5.
- IYSIZE: number of grid points in J direction, according to YINIFILE grid, recovered by the new domain. If to be used in MESONH, it must only be factor of 2,3 or 5.
- IDXRATIO: resolution factor in I direction between the YINIFILE grid and the new grid. If to be used in MESONH, it must only be factor of 2,3 or 5.
- IDYRATIO: resolution factor in J direction between the YINIFILE grid and the new grid. If to be used in MESONH, it must only be factor of 2,3 or 5.

4.2.3. Cartesian grids

4.2.3.1. Namelist NAM_CARTESIAN

This namelist defines the projection in case CGRID="CARTESIAN "

Fortran name	Fortran type	default value
XLAT0	real	none
XLON0	real	none
NIMAX	integer	none
NJMAX	integer	none
XDX	real	none
XDY	real	none

- XLAT0: reference latitude (real, decimal degrees)
- XLON0: reference longitude (real, decimal degrees)
- NIMAX: number of surface points of the grid in direction *x*.
- NJMAX: number of surface points of the grid in direction *y*.
- XDX: grid mesh size on the conformal plane in *x* direction (real, meters).
- XDY: grid mesh size on the conformal plane in *y* direction (real, meters).

4.2.3.2. Namelist NAM_INIFILE_CARTESIAN

This namelists defines the horizontal domain from an existing surface file in which grid type is "CARTESIAN ". If nothing is set in the namelist, a grid identical as the one in the file is chosen.

Fortran name	Fortran type	default value
IXOR	integer	1
IYOR	integer	1
IXSIZE	integer	YINIFILE size
IYSIZE	integer	YINIFILE size
IDXRATIO	integer	1
IDYRATIO	integer	1

- IXOR: first point I index, according to the YINIFILE grid, left to and out of the new physical domain.
- IYOR: first point J index, according to the YINIFILE grid, under and out of the new physical domain.
- IXSIZE: number of grid points in I direction, according to YINIFILE grid, recovered by the new domain. If to be used in MESONH, it must only be factor of 2,3 or 5.
- IYSIZE: number of grid points in J direction, according to YINIFILE grid, recovered by the new domain. If to be used in MESONH, it must only be factor of 2,3 or 5.
- IDXRATIO: resolution factor in I direction between the YINIFILE grid and the new grid. If to be used in MESONH, it must only be factor of 2,3 or 5.
- IDYRATIO: resolution factor in J direction between the YINIFILE grid and the new grid. If to be used in MESONH, it must only be factor of 2,3 or 5.

4.2.4. Longitude-latitude grids

4.2.4.1. Namelist NAM_LONLAT_REG

This namelist defines the projection in case CGRID="LONLAT REG"

Fortran name	Fortran type	default value
XLONMIN	real	none
XLONMAX	real	none
XLATMIN	real	none
XLATMAX	real	none
NLON	integer	none
NLAT	integer	none

- XLONMIN: minimum longitude covered by the grid, i.e. corresponding to the west border of the domain (real, decimal degrees). XLONMIN must be smaller than XLONMAX, but no more than 360 smaller.
- XLONMAX: maximum longitude covered by the grid, i.e. corresponding to the east border of the domain (real, decimal degrees). XLONMAX must be larger than XLONMIN, but no more than 360 larger.
- XLATMIN: minimum latitude covered by the grid, i.e. corresponding to the south border of the domain (real, decimal degrees). XLATMIN must be between -90 and +90, and smaller than XLATMAX.
- XLATMAX: maximum longitude covered by the grid, i.e. corresponding to the 'right' border of the domain (real, decimal degrees). XLATMAX must be between -90 and +90, and larger than XLATMIN.
- NLON: number of surface points in the longitude direction.
- NLAT: number of surface points in the latitude direction.

4.2.4.2. Namelist NAM_LONLATVAL

This namelist defines the projection in case CGRID="LONLATVAL "

Fortran name	Fortran type	values	default value	unit
NPOINTS	integer		none	
XX	real		none	degrees East
XY	real		none	degrees North
XDX	real		none	degrees
XDY	real		none	degrees

- NPOINTS: number of grid points defining the grid
- XX: longitude of grid mesh center
- YY: latitude coordinate of grid mesh center
- XDX: grid mesh size in *x* direction (real, degrees East).
- XDY: grid mesh size in *y* direction (real, degrees North).

4.2.5. Regular Lambert grids

4.2.5.1. Namelist NAM_IGN

This namelist defines the projection in case CGRID="IGN "

Fortran name	Fortran type	values	default value
CLAMBERT	character (len=3)	'L1 ', 'L2 ', 'L3 ', 'L4 ', 'L2E', 'L93'	none
NPOINTS	integer		none
XX	real		none
XY	real		none
XDX	real		none
XDY	real		none
XCELLSIZE	real		1.E+20
XX_LLCORNER	real		1.E+20
XY_LLCORNER	real		1.E+20
NCOLS	integer		0
NROWS	integer		0

- CLAMBERT: type of Lambert prjection
    - "L1 " : Lambert I
    - "L2 " : Lambert II
    - "L3 " : Lambert III
    - "L4 " : Lambert IV
    - "L2E" : Extended Lambert II
    - "L93" : Lambert 93
- NPOINTS: number of grid points defining the grid
- XX: X coordinate of grid mesh center
- YY: Y coordinate of grid mesh center
- XDX: grid mesh size on the conformal plane in *x* direction (real, meters).
- XDY: grid mesh size on the conformal plane in *y* direction (real, meters).
- XCELLSIZE: size of the cell (equal in X and Y). Has priority on XDX and XDY.
- XX_LLCORNER: X coordinate of left side of the domain.
- XY_LLCORNER: Y coordinate of lower side of the domain.
- NCOLS : number of columns.
- NROWS: number of rows.
  The simultaneous use of XX_LLCORNER, XY_LLCORNER, NCOLS and NROWS has priority of this of NPOINTS, XX and YY (it simplifies the namelist in case of a regular grid).

4.2.6. Gaussian grids

These namelists define the projection in case CGRID="GAUSS "

4.2.6.1. Namelist NAMDIM

Fortran name	Fortran type	default value
NDGLG	integer	none

- NDGLG: number of pseudo-latitudes

4.2.6.2. Namelist NAMRGRI

Fortran name	Fortran type	default value
NRGRI	integer	none

- NRGRI: number of pseudo-longitudes on each pseudo-latitude circle starting from the rotated pole

4.2.6.3. Namelist NAMGEM

Fortran name	Fortran type	default value
RMUCEN	real	none
RLOCEN	real	none
RSTRET	real	none

- RMUCEN: sine of the latitude of the rotated pole
- RLOCEN: longitude of the rotated pole (radian)
- RSTRET: stretching factor (must be greater than or equal to 1)

4.3. Orography, subgrid orography gaussian indices and bathymetry

4.3.1. Namelist NAM_ZS

This namelist defines the orography file and orographic treatment to be done.

Fortran name	Fortran type	values	default value
XUNIF_ZS	real		none
YZS	character (LEN=28)		' ' (default orography is 0.)
YFILETYPE	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none
COROGTYPE	character (LEN=3)	'AVG', 'ENV', 'SIL ', 'MAX'	'ENV'
XENV	real		0.
LIMP_ZS	logical		F

- XUNIF_ZS : uniform value of orography imposed on all points (real,meters). If XUNIF_ZS is set, file YZS is not used.
- YZS: data file name. If XUNIF_ZS is set, file YZS is not used. If neither XUNIF_ZS and YZS is set, then orography is set to zero.
- YFILETYPE: type of data file ('DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV')
- COROGTYPE: type of orography (string of 3 characters):
    - 'AVG': mean orography $\overline{z_s}$.
    - 'ENV': envelope relief, defined from mean orography and the subgrid orography standard deviation as $\overline{z_s} + XENV * \sigma_{z_s}$.
    - 'SIL': silhouette relief, defined as the mean of the two subgrid silhouettes in directions x and y (if two main directions can be defined for the grid chosen).
    - 'MAX': maximum orography over grid box (avoid averaging in case of sea/land grid box).
- XENV: enhance factor in envelope orography definition (real).
- LIMP_ZS: reads orography from an existing PGD file

4.3.2. Namelist NAM_GAUSS_INDEX

This namelist computes the gaussian grid indices once and for all

Fortran name	Fortran type	values	default value
LINDEX_STORE	logical		F
LSTOP_PGD	logical		F
YINDEX_1KM	character (LEN=28)		' '
YINDEX_10KM	character (LEN=28)		' '
YINDEX_100KM	character (LEN=28)		' '

- LINDEX_STORE : flag to write out gaussian grid indices in a binary file
- LSTOP_PGD : flag to stop PGD execution once the binary file containing the gaussian grid indices is ready
- YINDEX_1KM : name of the file where gaussian grid indices at 1km resolution are written
- YINDEX_10KM : name of the file where gaussian grid indices at 10km resolution are written
- YINDEX_100KM : name of the file where gaussian grid indices at 100km resolution are written

4.3.3. Namelist NAM_SEABATHY

This namelist defines the bathymetry file

Fortran name	Fortran type	values	default value
XUNIF_SEABATHY	real	negative for real ocean	300
YSEABATHY	character (LEN=28)		' '
YSEABATHYFILETYPE	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV','NETCDF'	none
YNCVARNAME	character (LEN=28)		

- XUNIF_SEABATHY : uniform value of bathymetry imposed on all points (real,meters). If XUNIF_SEABATHY is set, file YSEABATHY is not used.
- YSEABATHY: data file name. If XUNIF_SEABATHY is set, file YSEABATHY is not used. If neither XUNIF_SEABATHY and YSEABATHY is set, then bathymetry is set to zero.
- YSEABATHYFILETYPE: type of data file ('NETCDF')
- YNCVARNAME: name of variable to be read in NETCDF file

4.4. Land cover data

ECOCLIMAP is a global database of surface parameters.

Itâ s composed of a global land cover map, in which each ecosystem or cover is described by, at first level, 4 fractions of the 4 main surface types or tiles (SEA, WATER, NATURE, TOWN), and then, at second level in tile NATURE, 12 fractions of the 12 vegetation types or PFTs (Plant Functional Types).

These 12 PFTs are NO (bare soil), ROCK (bare rocks), SNOW (snow and ice), TREE (decideous broadleaf trees), CONI (evergreen needleleaf trees), EVER (evergreen broadleaf trees), C3 (C3 crops), C4 (C4 crops), IRR (irrigated crops), GRAS (temperate grassland), TROG (tropical grassland), PARK (swamp areas).

Then, surface parameters, like LAI, albedo, fraction of vegetation, soil depths, roughness length, depend on the PFTs, and, for some of them (LAI, soil depths, heights of trees), also on the cover.

The user can use ECOCLIMAP or give values for the parameters fields, in the namelist.

4.4.1. Choice of the type of land cover data

If ECOCLIMAP is used, surface parameters are defined through the ECOCLIMAP global map of ecosystems, following rules of attribution of parameters.

4.4.1.1. Namelist NAM_FRAC

This namelist defines if ECOCLIMAP mechanism based on fractions of covers will be used or not.

Itâ s also where the fractions of the 4 main types of surfaces or tiles are given if ECOCLIMAP is not used to define them.

Fortran name	Fortran type	values	default value
LECOCLIMAP	Logical		.TRUE.
XUNIF_SEA	real	between 0 and 1	none
CFNAM_SEA	character (LEN=28)		' '
CFTYP_SEA	character (LEN=6)	'DIRECT', 'BINLLF'	none
		'BINLLV', 'ASCLLV'	
XUNIF_WATER	real	between 0 and 1	none
CFNAM_WATER	character (LEN=28)		' '
CFTYP_WATER	character (LEN=6)	'DIRECT', 'BINLLF'	none
		'BINLLV', 'ASCLLV'	
XUNIF_NATURE	real	between 0 and 1	none
CFNAM_NATURE	character (LEN=28)		' '
CFTYP_NATURE	character (LEN=6)	'DIRECT', 'BINLLF'	none
		'BINLLV', 'ASCLLV'	
XUNIF_TOWN	real	between 0 and 1	none
CFNAM_TOWN	character (LEN=28)		' '
CFTYP_TOWN	character (LEN=6)	'DIRECT', 'BINLLF'	none
		'BINLLV', 'ASCLLV'	

- LECOCLIMAP : flag to use ECOCLIMAP or not. From version 7.1, itâ s possible to partially use ECOCLIMAP to complete missing parameters when they are given directly in the namelist (see here)

- XUNIF_SEA : uniform prescribed value of sea fraction. If XUNIF_SEA is set, file CFNAM_SEA is not used.
- CFNAM_SEA: sea fraction data file name. If XUNIF_SEA is set, file CFNAM_SEA is not used.
- CFTYP_SEA: type of sea data file ('DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV')

- XUNIF_WATER : uniform prescribed value of water fraction. If XUNIF_WATER is set, file CFNAM_WATER is not used.
- CFNAM_WATER: water fraction data file name. If XUNIF_WATER is set, file CFNAM_WATER is not used.
- CFTYP_WATER: type of water data file ('DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV')

- XUNIF_NATURE : uniform prescribed value of nature fraction. If XUNIF_NATURE is set, file CFNAM_NATURE is not used.

- CFNAM_NATURE: nature fraction data file name. If XUNIF_NATURE is set, file CFNAM_NATURE is not used.
- CFTYP_NATURE: type of nature data file ('DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV')

- XUNIF_TOWN : uniform prescribed value of town fraction. If XUNIF_TOWN is set, file CFNAM_TOWN is not used.
- CFNAM_TOWN: town fraction data file name. If XUNIF_TOWN is set, file CFNAM_TOWN is not used.
- CFTYP_TOWN: type of town data file ('DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV')

4.4.2. ECOCLIMAP is used

Some namelists are specific to the use of ECOCLIMAP (LECOCLIMAP=T in NAM_FRAC).

4.4.2.1. Namelist NAM_COVER

This namelist gives the information to compute the surface cover fractions when ECOCLIMAP is used.

Itâ s possible to use an existing ECOCLIMAP map or the define the ECOCLIMAP covers for the userâ s domain.

Fortran name	Fortran type	values	default value
XUNIF_COVER	array of 573 reals	â ¥ 0. and â $_{i=1,573}$ XUNIF_COVER(i) = 1	none
YCOVER	character (LEN=28)		' '
YFILETYPE	character (LEN=6)	'DIRECT', 'BINLLF',	none
		'BINLLV', 'ASCLLV'	
XRM_COVER	real	â ¥ 0.	10^{-6}
XRM_COAST	real	â ¥0.	1.
XRM_LAKE	real	â ¥0.	0.
XRM_SEA	real	â ¥0.	0.
LORCA_GRID	logical		F
XLAT_ANT	real		-77.
LIMP_COVER	logical		F

- XUNIF_COVER : specified values for uniform cover fractions. For each index *i* between 1 and 573, XUNIF_COVER(*i*) is the fraction of the i^{th} ecosystem of ecoclimap. The same fraction of each ecosystem is set to all points of the grid. The sum of all ecosystem fractions must be equal to one : â $_{i=1,573}$ XUNIF_COVER(i) = 1.
  If XUNIF_COVER is set, it has priority on the use of an ecosystem file (see next item: YCOVER). In the case of grid without any reference to geographical coordinates ("CARTESIAN " or "NONE "), XUNIF_COVER **must** be set.
- YCOVER: ecoclimap data file name. It is used only if XUNIF_COVER is not set.
- YFILETYPE: type of YCOVER file ('DIRECT', 'BINLLV', 'BINLLF', 'ASCLLV').
- XRM_COVER: for each point, all fractions of ecosystems that are below XRM_COVER are removed (i.e. set to zero), and the corresponding area fractions are distributed among the remaining ecosystem fractions. Whatever the value of XRM_COVER, at least one ecosystem remains for each grid point.
- XRM_COAST: limit of coast coverage under which the coast is replaced by sea or inland water.
- XRM_LAKE: limit of inland lake coverage under which the water is removed.
- XRM_SEA: limit of sea coverage under which the sea is removed.
- LORCA_GRID: flag to ensure the compatibility between surfex and Orca grid which minimal latitude over Antarctica is 77S
- XLAT_ANT: minimum Orca grid latitude over Antarctica

- LIMP_COVER: reads the cover fractions in an existing PGD file to avoid their computation

4.4.2.2. Namelist NAM_PGD_ARRANGE_COVER

This namelist initializes change water (not lake) to nature and/or town to rock keys.

Fortran name	Fortran type	values	default value
LWATER_TO_NATURE	logical		F
LTOWN_TO_ROCK	logical		F

- LWATER_TO_NATURE: Change Wetland treated as inland water into nature
- LTOWN_TO_ROCK : Change Town into Rock

4.4.2.3. Namelist NAM_READ_DATA_COVER

Fortran name	Fortran type	values	default value
LREAD_DATA_COVER	logical		T

- LREAD_DATA_COVER: if T, covers data are read in .bin files; if F, in fortran routines.

4.4.2.4. Namelist NAM_WRITE_COVER_TEX

Fortran name	Fortran type	values	default value
CLANG	character(LEN=2)	''	'EN'

- CLANG: language used in the file class_cover_tex.tex

4.4.2.5. List of ECOCLIMAP covers names

ECOCLIMAP I

COVER 1 : Sea and ocean
COVER 2 : Inland waters
COVER 3 : Rivers
COVER 4 : Bare land
COVER 5 : Rocks
COVER 6 : Permanent snow and ice
COVER 7 : Urban and built-up

COVER 8 : Tropical undefined islands
COVER 9 : Subpolar undefined islands

ENF = Evergreen Needleleaf Forest
COVER 10 : S-America cool ENF
COVER 11 : Boreal ENF
COVER 12 : Asia subtropical ENF
COVER 13 : American Continental ENF
COVER 14 : American Subtropical ENF
COVER 15 : American Cool Marine ENF

EBF = Evergreen Broadleaf Forest
COVER 16 : Africa Equatorial EBF
COVER 17 : Africa Tr. wind EBF
COVER 18 : Oceanian Equatorial EBF
COVER 19 : Asia tropical EBF
COVER 20 : Oceania tropical EBF
COVER 21 : Amazonian EBF
COVER 22 : SH subtropical EBF
COVER 23 : Cent. America Tr. wind EBF

DNF = Deciduous Needleleaf Forest
COVER 24 : Asian boreal DNF

DBF = Deciduous Broadleaf Forest
COVER 25 : S-America tropical DBF
COVER 26 : N-America humid continental DBF
COVER 27 : Cent. America Tr. wind DBF
COVER 28 : S-America humid subtropical DBF

MF = Mixed Forest
COVER 29 : Africa dry tropical MF
COVER 30 : S-America cool MF
COVER 31 : NH Subpolar MF

COVER 32 : NH Humid subtropical MF
COVER 33 : NH Continental MF

WL = Wood Land
COVER 34 : NH Africa WL
COVER 35 : SH Africa WL
COVER 36 : Tr. wind humid and subtrop. WL
COVER 37 : Oceanian Equatorial WL
COVER 38 : Asia wet tropical WL
COVER 39 : S-America tropical WL
COVER 40 : S-America humid subtropical WL
COVER 41 : NH Subpolar WL
COVER 42 : NH Continental WL
COVER 43 : Asia humid subtropical WL
COVER 44 : N-America Semi arid WL
COVER 45 : N-America moderate polar WL
COVER 46 : S-America moderate polar WL
COVER 47 : N-America humid subtropical WL

WG = Wooded Grassland
COVER 48 : NH Africa semiarid WG
COVER 49 : NH Africa dry tropical WG
COVER 50 : Africa dry equatorial WG
COVER 51 : SH Africa dry tropical WG
COVER 52 : Oceania tropical WG
COVER 53 : Oceania semiarid WG
COVER 54 : Oceania subtrop. cool marine WG
COVER 55 : Asia humid and subtropical WG
COVER 56 : S-America trop. and subtrop. WG
COVER 57 : S-America Tr. wind WG
COVER 58 : S-America semiarid WG
COVER 59 : NH Subpolar WG
COVER 60 : NH Continental WG
COVER 61 : Asia wet and dry tropical WG
COVER 62 : N-America semi arid WG
COVER 63 : N-America humid subtropical WG
COVER 64 : S-America moderate polar WG
COVER 65 : Cent. Amer. Tr. wind & trop. WG
COVER 66 : NH Africa dry summer subtrop. WG

CS = Closed Shrubland
COVER 67 : NH Africa arid CS
COVER 68 : NH Africa semiarid CS
COVER 69 : SH Africa semiarid CS
COVER 70 : Oceania arid CS
COVER 71 : Oceania, S-America semiarid CS

COVER 72 : Oceania Tr. wind CS
COVER 73 : SH dry summer subtropical CS
COVER 74 : Asia polar CS
COVER 75 : Asia continental CS
COVER 76 : Asia tropical CS
COVER 77 : N-America polar CS
COVER 78 : N-America continental CS
COVER 79 : NH Africa dry summer subtrop. CS

OS = Open Shrubland
COVER 80 : NH arid OS
COVER 81 : NH semiarid tropical OS
COVER 82 : SH Africa and Oceania arid OS
COVER 83 : S-America semiarid tropical OS
COVER 84 : Asia dry tropical OS
COVER 85 : NH Polar OS
COVER 86 : N-America Subpolar OS
COVER 87 : N-America semiarid continental OS

G = Grassland
COVER 88 : Africa wet Tropical G
COVER 89 : NH Africa Semiarid G
COVER 90 : SH Africa Semiarid G
COVER 91 : S-America, Oceania equatorial G
COVER 92 : S-America, Oceania Semiarid G
COVER 93 : Oceania cool littoral G
COVER 94 : Asia wet and dry tropical G
COVER 95 : NH S-America wet tropical G
COVER 96 : SH S-America wet tropical G
COVER 97 : S-America semiarid G
COVER 98 : S-America moderate polar G
COVER 99 : NH semiarid Continental G
COVER 100 : Asia Subpolar G
COVER 101 : Asia humid Continental G
COVER 102 : Asia semiarid tropical G
COVER 103 : N-America continental G
COVER 104 : Asia humid subtropical G

C = Crops
COVER 105 : NH Africa arid C
COVER 106 : NH Africa, Asia wet and dry trop. C
COVER 107 : SH Africa wet and dry tropical C
COVER 108 : SH Afr. Tr. wind & semiarid trop. C
COVER 109 : Oceania dry summer subtropical C
COVER 110 : Cent. & S-Amer., Oceania Tr. wind C
COVER 111 : S-America humid subtropical C

COVER 112 : SH S-America tropical C
COVER 113 : N-Amer., Asia semiarid continental C
COVER 114 : Asia humid continental C
COVER 115 : Asia humid subtropical C
COVER 116 : Asia subpolar C
COVER 117 : Asia semiarid tropical C
COVER 118 : N-America humid continental C
COVER 119 : N-America humid subtropical C
COVER 120 : NH dry summer subtropical C
COVER 121 : NH Africa dry summer subtropical C
COVER 122 : SH Africa dry summer subtropical C

COVER 123 : Bare soil with sparse polar vegetation

COVER 124 : Warm subtropical wetlands
COVER 125 : Subpolar wetlands

COVER 151 : Dense urban
COVER 152 : Mediterranean sub-urban
COVER 153 : Temperate sub-urban
COVER 154 : Cold sub-urban
COVER 155 : Industries and commercial areas
COVER 156 : Road and rail networks
COVER 157 : Port facilities
COVER 158 : Airport
COVER 159 : Mineral extraction, construction sites
COVER 160 : Urban parks
COVER 161 : Sport facilities

COVER 162 : Spanish crops
COVER 163 : Estremadura crops
COVER 164 : Mediterranean crops
COVER 165 : Atlantic coast crops
COVER 166 : Temperate crops
COVER 167 : Po plain crops
COVER 168 : Warm temperate crops
COVER 169 : Ukrainian crops
COVER 170 : Subpolar crops
COVER 171 : Mountain crops
COVER 172 : Central Europe crops
COVER 173 : Turkish crops

COVER 174 : Mediterranean irrigated crops
COVER 175 : Irrigated crops
COVER 176 : Rice fields

COVER 177 : Mediterranean vineyards
COVER 178 : Temperate vineyards
COVER 179 : Mediterranean fruit trees
COVER 180 : Temperate fruit trees
COVER 181 : Olive groves

COVER 182 : Temperate pastures
COVER 183 : Atlantic border pastures
COVER 184 : Central and Eastern Europe pastures
COVER 185 : Ukrainian pastures
COVER 186 : Subpolar pastures

COVER 187 : Spanish complex cultivation pattern
COVER 188 : Mediter. complex cultivation pat.
COVER 189 : Temperate complex cultivation pat.
COVER 190 : French complex cultivation pat.
COVER 191 : Balkanish complex cultivation pat.

COVER 192 : Mediterranean crops and woodland
COVER 193 : Crops and woodland
COVER 194 : French crops and woodland
COVER 195 : Balkanish crops and woodland
COVER 196 : Spanish crops and woodland
COVER 197 : Baltic states crops and woodland

COVER 198 : Agro-forestry areas

COVER 199 : Spanish broad-leaved forest
COVER 200 : Estremadura broad-leaved forest
COVER 201 : Mediterranean broad-leaved forest
COVER 202 : Atlantic coast broad-leaved forest
COVER 203 : Temperate broad-leaved forest
COVER 204 : Moutain broad-leaved forest
COVER 205 : Balkanish broad-leaved forest
COVER 206 : Subpolar broad-leaved forest
COVER 207 : Black Sea broad-leaved forest

COVER 208 : Mediterranean pines
COVER 209 : Landes forest
COVER 210 : Moutain coniferous forest
COVER 211 : Temperate coniferous forest
COVER 212 : Subpolar Taiga
COVER 213 : Russian Taiga
COVER 214 : Turkish coniferous forest

COVER 215 : Mediterranean mixed forest

COVER 216 : Atlantic coast & french mixed forest
COVER 217 : Subpolar mixed forest
COVER 218 : Mountain mixed forest
COVER 219 : Eastern Europe mixed forest

COVER 220 : Mediterranean GR
COVER 221 : Atlantic coast GR
COVER 222 : Balkanish GR
COVER 223 : Estremadura GR
COVER 224 : Subpolar GR
COVER 225 : Tundra

COVER 226 : Turkish moors
COVER 227 : Mediter. moors & heath lands
COVER 228 : Moutain moors & heath lands
COVER 229 : Atlantic coast moors & heath lands

COVER 230 : Turkish shrubland
COVER 231 : Mediterranean maquis
COVER 232 : Moutain maquis

COVER 233 : Spanish woodland
COVER 234 : Mediterranean woodland
COVER 235 : Temperate woodland

COVER 236 : Sparsely vegetated areas
COVER 237 : Burnt areas
COVER 238 : Temperate wetlands
COVER 239 : Subpolar wetlands
COVER 240 : Peat bogs
COVER 241 : Salines and salt marshes

COVER 242 : Intertidal flats
COVER 243 : Coastal lagoons

ECOCLIMAP II EUROPE

forests
COVER 301 : N SCANDINAVIA TUNDRA1
COVER 302 : OURAL BF1
COVER 303 : CARELIE BF1
COVER 304 : NORTH RUSSIAN TAIGA1
COVER 305 : NORTH RUSSIAN TAIGA2
COVER 306 : CARELIE BF2
COVER 307 : RUSSIAN TAIGA3
COVER 308 : RUSSIAN BF1

COVER 309 : RUSSIAN TAIGA4
COVER 310 : S SCANDINAVIA TAIGA1
COVER 311 : SOUTH FINLANDIA MF1
COVER 312 : SOUTH NORWAY MF1
COVER 313 : BALTIC BF1
COVER 314 : BALTIC MF1
COVER 315 : SOUTH SWEDEN CF1
COVER 316 : BALTIC MF2
COVER 317 : SOUTH SWEDEN CF2
COVER 318 : SOUTH SWEDEN CF3
COVER 319 : SOUTH SWEDEN MF1
COVER 320 : MOUNTAIN MF1
COVER 321 : MOUNTAIN BF1
COVER 322 : TEMPERATE BF1
COVER 323 : TEMPERATE COMPLEX1
COVER 324 : MOUNTAIN CF1
COVER 325 : TEMP HERBACEOUS CF1
COVER 326 : ATLANTIC COAST BF1
COVER 327 : TURKISH CF1
COVER 328 : BALKAN CF1
COVER 329 : N SPAIN HERBAC MF1
COVER 330 : TEMP SW HERBAC CF1
COVER 331 : ATLANTIC COMPLEX1
COVER 332 : N SPAIN HERBAC MF2
COVER 333 : MEDITER COMPLEX1
COVER 334 : MEDITER COMPLEX2
COVER 335 : MEDITER COMPLEX3
COVER 336 : MEDITER COMPLEX4
COVER 337 : MEDITER COMPLEX5
COVER 338 : BURNT PORT HERBAC CF1
COVER 339 : BURNT PORT HERBAC BF1
COVER 340 : EGEE COAST COMPLEX1
COVER 341 : W MED COAST COMPLEX1
COVER 342 : MAGHR HERBACEOUS MF1
COVER 343 : ESTREM HERBACEOUS MF1

herbaceous / shrub covers
COVER 344 : POLAR MOUNT TUNDRA1
COVER 345 : POLAR MOUNT TUNDRA2
COVER 346 : S SCANDINAVIA TUNDRA1
COVER 347 : NORTH TUNDRA1
COVER 348 : S SCANDINAVIA TUNDRA2
COVER 349 : NORTH RUSSIA TUNDRA1
COVER 350 : ARAL CONTINENTAL GR1
COVER 351 : MOUNTAIN TAIGA MOORS1
COVER 352 : SCOTTISH SWAMP MOORS1

COVER 353 : ATLANTIC COMPLEX2
COVER 354 : ATLANTIC GR1
COVER 355 : IR SCOT SWAMP MOORS1
COVER 356 : ASIAN SPARSE GR1
COVER 357 : AS SPARSE SW COMPLEX1
COVER 358 : N CASPIAN DES OS1
COVER 359 : ATLAS AS SPARSE COMP1
COVER 360 : SPARSE SCO CEN EU GR1
COVER 361 : TEMPERATE COMPLEX2
COVER 362 : ATLANTIC COMPLEX3
COVER 363 : ATLANTIC COMPLEX4
COVER 364 : N ATLANTIC PASTURES1
COVER 365 : SPARSE SCO CEN EU GR2
COVER 366 : SPARSE MOUNT E EU GR1
COVER 367 : TUR N CASP CONT GR1
COVER 368 : N CASPIAN CONT GR1
COVER 369 : IRA N CASP CONT GR1
COVER 370 : TUR IRA MOUNT CONT GR1
COVER 371 : E CASPIAN DES OS1
COVER 372 : N CASPIAN COMPLEX1
COVER 373 : IRAN MOUNT CONT GR1
COVER 374 : ASIAN SPARSE DES OS1
COVER 375 : E CASPIAN DES OS2
COVER 376 : N MEDITER COMPLEX1
COVER 377 : N MEDITER COMPLEX2
COVER 378 : ASIAN MEDIT CONT GR1
COVER 379 : SOUTH RUSSIA CONT GR1
COVER 380 : BLSEA SPARSE CONT GR1
COVER 381 : BLSEA SPARSE CONT GR2
COVER 382 : TURK MOUNT CONT GR1
COVER 383 : TURKISH COMPLEX1
COVER 384 : CAUCASIAN COMPLEX1
COVER 385 : N CASPIAN CONT GR2
COVER 386 : VOLGA VALLEY CONT GR1
COVER 387 : VOLGA VALLEY CONT GR2
COVER 388 : W CASPIAN CONT GR1
COVER 389 : CAUCASIAN COMPLEX2
COVER 390 : CAUCASIAN COMPLEX3
COVER 391 : BLSEA SPARSE CONT GR3
COVER 392 : CENT MASSIF COMPLEX1
COVER 393 : CENT MASSIF COMPLEX2
COVER 394 : TURK COAST COMPLEX1
COVER 395 : MESOPOTAMIA GR1
COVER 396 : TURK CILICIA COMPLEX1
COVER 397 : ASIAN COMPLEX1
COVER 398 : N MED SPARSE COMPLEX1

COVER 399 : MEDITER COMPLEX6
COVER 400 : MEDIT SPARSE COMPLEX1
COVER 401 : MEDIT SPARSE COMPLEX2
COVER 402 : MEDIT SPARSE COMPLEX3
COVER 403 : MEDIT SPARSE COMPLEX4
COVER 404 : N MED HERBACEOUS CF1
COVER 405 : ESTREMADURA GR1
COVER 406 : TUNISIA COMPLEX1
COVER 407 : TUNISIA HERBACEOUS1
COVER 408 : ALGERIA HERBACEOUS1
COVER 409 : DESERTIC HERBACEOUS1
COVER 410 : DESERTIC HERBACEOUS2
COVER 411 : SPAIN DES COMPLEX1
COVER 412 : MED SPARSE COMPLEX5
COVER 413 : MED SPARSE COMPLEX6
COVER 414 : MED SPARSE COMPLEX7
COVER 415 : ME SPARSE DES COMPL1
COVER 416 : NORTH ARABIA GR1
COVER 417 : N ARABIA DES COMPLEX1
COVER 418 : N ARABIA DESERTIC GR1
COVER 419 : MOROCCO HERBACEOUS1
COVER 420 : S MED COAST HERBAC1
COVER 421 : W MEDITER WOODLAND1
COVER 422 : S MED COAST HERBAC2
COVER 423 : MESOP DES HERBACEOUS1
COVER 424 : MAG COAST DES HERBAC1
COVER 425 : TU AR SPARSE HERBAC1
COVER 426 : MEDIT SPARSE COMPLEX8
COVER 427 : MED SPARSE HERBAC1
COVER 428 : MEDIT SPARSE COMPLEX9
COVER 429 : SPAIN SPARSE COMPLEX1
COVER 430 : N MED SPARSE COMPLEX2
COVER 431 : N MED SPARSE COMPLEX3
COVER 432 : MAGHRE DES HERBAC1
COVER 433 : MAGHRE DES HERBAC2
COVER 434 : MAGHRE DES HERBAC3
COVER 435 : N ARAB DES HERBAC1
COVER 436 : MESOPO DES HERBAC2
COVER 437 : TOURAN DES HERBAC1
COVER 438 : MESOPO DES HERBAC2
COVER 439 : TOURAN DES HERBAC2
COVER 440 : NEW ZEMBLE HERBAC1
COVER 441 : NEW ZEMBLE HERBAC2

crops
COVER 442 : TRANS SIBERIAN CROPS1

COVER 443 : PO PLAIN CROPS1
COVER 444 : PO PLAIN CROPS2
COVER 445 : SPANISH FRENCH CROPS1
COVER 446 : SPANISH FR ITAL CROPS1
COVER 447 : DANUBE PLAIN CROPS1
COVER 448 : N MED SPARSE COMPLEX4
COVER 449 : BALKAN CROPS1
COVER 450 : SPAIN FR ITAL CROPS2
COVER 451 : ATLANTIC CROPS1
COVER 452 : FR MED SPARSE CROPS1
COVER 453 : FR MED SPARSE CROPS2
COVER 454 : ATL MED SPARSE CROPS1
COVER 455 : BENE BLACK SEA CROPS1
COVER 456 : FRENCH ITALIAN CROPS1
COVER 457 : FR MED SPARSE CROPS3
COVER 458 : MEDITER SPARSE CROPS1
COVER 459 : ATLANTIC CROPS2
COVER 460 : NORTH ATLANTIC CROPS1
COVER 461 : SOUTH RUSSIA CROPS1
COVER 462 : S RUSSIA BALTIC CROPS1
COVER 463 : UKRAINIAN CROPS1
COVER 464 : EAST CARPATES CROPS1
COVER 465 : E CENT EUROPE CROPS1
COVER 466 : W CENT EU SW CROPS1
COVER 467 : HUNGARIAN CROPS1
COVER 468 : N BLACK SEA CROPS1
COVER 469 : HUNG BULG CAUC CROPS1
COVER 470 : SOUTH SWEDEN CROPS1
COVER 471 : SW RUSSIA CROPS1
COVER 472 : SOUTH RUSSIA CROPS1
COVER 473 : IRAN N CASPIAN CROPS1
COVER 474 : FR TEMP SPARSE CROPS1
COVER 475 : BULGARIAN CROPS1
COVER 476 : BULGARIAN CROPS2
COVER 477 : SP TURK SPARSE CROPS1
COVER 478 : FRENCH CENT EU CROPS1
COVER 479 : N BLACK SEA CROPS2
COVER 480 : BULGARIAN CROPS3
COVER 481 : POLE CROPS1
COVER 482 : POLE CROPS2
COVER 483 : N BLACK SEA CROPS3
COVER 484 : CENT EU SPARSE CROPS1
COVER 485 : GERMAN CROPS1
COVER 486 : BEAUCE CROPS1
COVER 487 : DANE CROPS1
COVER 488 : DANE CROPS2

COVER 489 : NEU ATL SPARSE CROPS1
COVER 490 : SYRIAN CROPS1
COVER 491 : GERMAN CROPS2
COVER 492 : CHANNEL CROPS1
COVER 493 : CHANNEL CROPS2
COVER 494 : ITALIAN CROPS1
COVER 495 : TURKISH CROPS1
COVER 496 : N MEDIT SPARSE CROPS1
COVER 497 : SPAIN TUR ARAB CROPS1
COVER 498 : NORTH SPAIN CROPS1
COVER 499 : MOROCCO TUNIS CROPS1
COVER 500 : MOROCCO CROPS1
COVER 501 : MOROCCO CROPS2
COVER 502 : ALGERIAN CROPS1
COVER 503 : MOROCCO CROPS3
COVER 504 : WEST SPAIN CROPS1
COVER 505 : MOROCCO CROPS4
COVER 506 : NORTH MEDITER CROPS1
COVER 507 : SOUTH SPANISH CROPS1
COVER 508 : SICILIAN CROPS1
COVER 509 : MAGHREB SPARSE CROPS1
COVER 510 : N MEDIT SPARSE CROPS2
COVER 511 : N MEDIT SPARSE CROPS3
COVER 512 : SP IT WCOAST CROPS1
COVER 513 : ESTREMADURA CROPS1
COVER 514 : ESTREMADURA CROPS2
COVER 515 : SP IT WCOAST CROPS2
COVER 516 : ESTREMADURA CROPS3
COVER 517 : MEDIT ISLANDS CROPS1
COVER 518 : SPAIN W COAST CROPS1
COVER 519 : ESTREMADURA CROPS4
COVER 520 : MECOAST SPARSE CROPS1
COVER 521 : BRITTANY CROPS1
COVER 522 : SYRIAN CROPS2

irrigated crops
COVER 523 : NIL VALLEY CROPS1
COVER 524 : NIL VALLEY CROPS2
COVER 525 : NIL VALLEY CROPS3
COVER 526 : NIL VALLEY CROPS4
COVER 527 : SPANISH IRR CROPS1
COVER 528 : NIL VALLEY CROPS5
COVER 529 : EGEE IRR CROPS1
COVER 530 : MEDITER IRR CROPS1
COVER 531 : S SPAIN IRR CROPS1
COVER 532 : NIL VALLEY CROPS6

bare land
COVER 533 : BARE ROCK1
COVER 534 : BARE ROCK2
COVER 535 : SANDY DESERT1
COVER 536 : BARE LAND1
COVER 537 : BARE LAND2
COVER 538 : BARE LAND3
COVER 539 : BARE LAND4
COVER 540 : BARE LAND5
COVER 541 : BARE LAND6
COVER 542 : BARE LAND7
COVER 543 : BARE LAND8
COVER 544 : BARE LAND9
COVER 545 : BARE LAND10
COVER 546 : BARE LAND11
COVER 547 : BARE LAND12
COVER 548 : PERMANENT SNOW1

swamp areas and inland waters
COVER 549 : INLAND WATERS1
COVER 550 : UNDEFINED1
COVER 551 : INLAND WATERS2
COVER 552 : POLAR WETLANDS1
COVER 553 : INLAND WATERS3
COVER 554 : INLAND WATERS4
COVER 555 : INLAND WATERS5
COVER 556 : INLAND WATERS6
COVER 557 : POLAR WETLANDS2
COVER 558 : SUBPOLAR WETLANDS1
COVER 559 : SUBPOLAR WETLANDS2
COVER 560 : SUBPOLAR WETLANDS3

urban
COVER 561 : TEMPERATE SUBURBAN1
COVER 562 : TEMPERATE SUBURBAN2
COVER 563 : TEMPERATE SUBURBAN3
COVER 564 : TEMPERATE SUBURBAN4
COVER 565 : TEMPERATE SUBURBAN5
COVER 566 : COLD SUBURBAN1
COVER 567 : WARM SUBURBAN1
COVER 568 : WARM SUBURBAN2
COVER 569 : TEMPERATE SUBURBAN6
COVER 570 : TEMPERATE SUBURBAN7
COVER 571 : WARM SUBURBAN3

added classes of permanent crops

COVER 572 : SPANISH VINEYARDS1
COVER 573 : LANGUEDOC VINEYARDS1

4.4.3. some land cover data fields are defined by user

If user has available better data than ECOCLIMAP for the land cover data fields, itâ s possible to notify them in the namelist and ECOCLIMAP data will be overwritten by the userâ s data.

If the land cover data fields needed for SURFEX are not all provived by the user, the missing fields are completed with ECOCLIMAPâ s information.

4.4.3.1. ISBA scheme: NAM_DATA_ISBA

Over natural areas, all surface parameters for each vegtype at a given frequency have to be specified by the user in namelist **NAM_DATA_ISBA**.

If LECOCLIMAP = T (NAM_FRAC), only part of the surface parameters for each vegtype can be given in NAM_DATA_ISBA. They are then completed by ECOCLIMAP data.

If only data for some of the 12 vegtypes are given, other vegtypes are filled with the values of the first given vegtype placed before in the list of 12.

parameters depending on vegetation types:

Fortran name	Fortran type	values	default value	description	unit
XUNIF_VEGTYPE	real	between 0 and 1	none	vegetation type	
CFNAM_VEGTYPE	character (LEN=28)			file name	
CFTYP_VEGTYPE	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none	file type	

parameters depending on vegetation types and time:

Fortran name	Fortran type	values	default value	description	unit
NTIME	integer	1, 2, 12 or 36	36	time dimension	
XUNIF_VEG	real	between 0 and 1	none	vegetation fraction	(-)
CFNAM_VEG	character (LEN=28)		' '	file name	
CFTYP_VEG	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none	file type	
XUNIF_LAI	real		none	leaf area index	(m^2/m^2)
CFNAM_LAI	character (LEN=28)		' '	file name	
CFTYP_LAI	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none	file type	
XUNIF_Z0	real		none	roughness length	(m)
CFNAM_Z0	character (LEN=28)		' '	file name	
CFTYP_Z0	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none	file type	
XUNIF_EMIS	real		none	emissivity	(-)

CFNAM_EMIS	character (LEN=28)		' '	file name	
CFTYP_EMIS	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none	file type	

parameters depending on vegetation types and soil levels:

Fortran name	Fortran type	values	default value	description	unit
XUNIF_DG	real		none	soil layer thickness	(*m*)
CFNAM_DG	character (LEN=28)		' '		
CFTYP_DG	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_ROOTFRAC	real		none	root fraction	(-)
CFNAM_ROOTFRAC	character (LEN=28)		' '		
CFTYP_ROOTFRAC	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_ROOT_DEPTH	real		none	root depth	(-)
CFNAM_ROOT_DEPTH	character (LEN=28)		' '		
CFTYP_ROOT_DEPTH	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_GROUND_DEPTH	real		none	ground depth for hydrology	(-)
CFNAM_GROUND_DEPTH	character (LEN=28)		' '		
CFTYP_GROUND_DEPTH	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_ROOT_EXTINCTION	real		none	root extinction percentage	(-)
CFNAM_ROOT_EXTINCTION	character (LEN=28)		' '		
CFTYP_ROOT_EXTINCTION	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_ROOT_LIN	real		none	root linear parameter	(-)

CFNAM_ROOT_LIN	character (LEN=28)		' '		
CFTYP_ROOT_LIN	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		

parameters depending on vegetation types only:

Fortran name	Fortran type	values	default value	description	unit
XUNIF_RSMIN	real		none	minimal stomatal resistance	(*s/m*)
CFNAM_RSMIN	character (LEN=28)		' '		
CFTYP_RSMIN	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_GAMMA	real		none	coefficient used in the computation of RSMIN	(-)
CFNAM_GAMMA	character (LEN=28)		' '		
CFTYP_GAMMA	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_WRMAX_CF	real		none	coefficient for maximum interception water storage capacity	(-)
CFNAM_WRMAX_CF	character (LEN=28)		' '		
CFTYP_WRMAX_CF	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
XUNIF_RGL	real		none	maximum solar radiation available for photosynthesis	(W/m^2)
CFNAM_RGL	character (LEN=28)		' '		
CFTYP_RGL	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_CV	real		none	vegetation thermal inertia coefficient	($K\ m^2/J$)
CFNAM_CV	character (LEN=28)		' '		
CFTYP_CV	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_Z0_O_Z0H	real		none		(-)

				ratio of surface roughness lengths	
CFNAM_Z0_O_Z0H	character (LEN=28)		' '		
CFTYP_Z0_O_Z0H	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_ALBNIR_VEG	real		none	vegetation near-infra-red albedo	(-)
CFNAM_ALBNIR_VEG	character (LEN=28)		' '		
CFTYP_ALBNIR_VEG	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_ALBVIS_VEG	real		none	vegetation visible albedo	(-)
CFNAM_ALBVIS_VEG	character (LEN=28)		' '		
CFTYP_ALBVIS_VEG	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_ALBUV_VEG	real		none	vegetation UV albedo	(-)
CFNAM_ALBUV_VEG	character (LEN=28)		' '		
CFTYP_ALBUV_VEG	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
XUNIF_ALBNIR_SOIL	real		none	soil near-infra-red albedo	(-)
CFNAM_ALBNIR_SOIL	character (LEN=28)		' '		
CFTYP_ALBNIR_SOIL	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
XUNIF_ALBVIS_SOIL	real		none	soil visible albedo	(-)
CFNAM_ALBVIS_SOIL	character (LEN=28)		' '		
CFTYP_ALBVIS_SOIL	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_ALBUV_SOIL	real		none	soil UV albedo	(-)
CFNAM_ALBUV_SOIL	character (LEN=28)		' '		
CFTYP_ALBUV_SOIL	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		

Isba-A-gs parameters depending on vegetation types only:

Fortran name	Fortran type	values	default	description	unit
XUNIF_GMES	real		none	mesophyll conductance	(*m/s*$^{-1}$)
CFNAM_GMES	character (LEN=28)		' '		
CFTYP_GMES	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_BSLAI	real		none	ratio d(biomass)/d(lai)	(kg/m^2)
CFNAM_BSLAI	character (LEN=28)		' '		
CFTYP_BSLAI	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_LAIMIN	real		none	minimum LAI	(m^2/m^2)
CFNAM_LAIMIN	character (LEN=28)		' '		
CFTYP_LAIMIN	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_SEFOLD	real		none	e-folding time for senescence	(*s*)
CFNAM_SEFOLD	character (LEN=28)		' '		
CFTYP_SEFOLD	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_GC	real		none	cuticular conductance	(*m/s*)
CFNAM_GC	character (LEN=28)		' '		
CFTYP_GC	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
XUNIF_DMAX	real		none	maximum air saturation deficit	(*kg/kg*)
CFNAM_DMAX	character (LEN=28)		' '		
CFTYP_DMAX	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
XUNIF_F2I	real		none	critical normilized soil water content for stress parameterization	(-)
CFNAM_F2I	character (LEN=28)		' '		
CFTYP_F2I	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		

XUNIF_H_TREE	real		none	height of trees	(*m*)
CFNAM_H_TREE	character (LEN=28)		' '		
CFTYP_H_TREE	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_RE25	real		none	Ecosystem respiration parameter	($kg/kgms^{-1}$)
CFNAM_RE25	character (LEN=28)		' '		
CFTYP_RE25	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
XUNIF_CE_NITRO	real		none	leaf aera ratio sensivity to [nitrogen]	(m^2/kg)
CFNAM_CE_NITRO	character (LEN=28)		' '		
CFTYP_CE_NITRO	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
XUNIF_CF_NITRO	real		none	lethal minimum value of leaf area ratio	(m^2/kg)
CFNAM_CF_NITRO	character (LEN=28)		' '		
CFTYP_CF_NITRO	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
XUNIF_CNA_NITRO	real		none	nitrogen concentration of active biomass	(kg/kg)
CFNAM_CNA_NITRO	character (LEN=28)		' '		
CFTYP_CNA_NITRO	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
LUNIF_STRESS	logical		T		

4.4.3.2. TEB scheme: NAM_DATA_TEB

Over urban areas, all surface parameters have to be specified by the user in namelist **NAM_DATA_TEB**.

But, if LECOCLIMAP = T (NAM_FRAC), only some of them can be specified and the missing parameters are completed with ECOCLIMAP database.

Fortran name	Fortran type	values	default	description	unit
NPAR_ROOF_LAYER	integer		0	number of layers in roofs	(-)
NPAR_ROAD_LAYER	integer		0	number of layers in roads	(-)
NPAR_WALL_LAYER	integer		0	number of layers in walls	(-)
CBLD_ATYPE	character(LEN=3)	'MAJ','ARI'	'MAJ'	type of averaging for building	(-)
CCSVDATAFILE	character(LEN=28)		' '	name of the CSV data file where to read the buildings description	(-)
NUNIF_BLDTYPE	integer		1E+9	type of buidlings	(-)
CFNAM_BLDTYPE	character (LEN=28)		' '		
CFTYP_BLDTYPE	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
NUNIF_BLDAGE	integer		1E+9	date of construction of buildings	(-)
CFNAM_BLDAGE	character (LEN=28)		' '		
CFTYP_BLDAGE	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
NUNIF_USETYPE	integer		1E+9	type of use in the buildings	(-)
CFNAM_USETYPE	character (LEN=28)		' '		
CFTYP_USETYPE	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_ROAD_DIR	real		1.E+20	road direction	(Ã Â° from North, clockwise)
CFNAM_ROAD_DIR	character (LEN=28)		' '		
CFTYP_ROAD_DIR	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV',	none		

		'ASCLLV'			
Fortran name	Fortran type	values	default	description	unit
XUNIF_BLD	real		none	fraction of buildings	(-)
CFNAM_BLD	character (LEN=28)		' '		
CFTYP_BLD	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_BLD_HEIGHT	real		none	buildings height	(m)
CFNAM_BLD_HEIGHT	character (LEN=28)		' '		
CFTYP_BLD_HEIGHT	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
XUNIF_WALL_O_HOR	real		none	wall surf. / hor. surf.	(-)
CFNAM_WALL_O_HOR	character (LEN=28)		' '		
CFTYP_WALL_O_HOR	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
XUNIF_Z0_TOWN	real		none	roughness length for momentum	(m)
CFNAM_Z0_TOWN	character (LEN=28)		' '		
CFTYP_Z0_TOWN	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
XUNIF_ALB_ROOF	real		none	roof albedo	(-)
CFNAM_ALB_ROOF	character (LEN=28)		' '		
CFTYP_ALB_ROOF	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_EMIS_ROOF	real		none	roof emissivity	(-)
CFNAM_EMIS_ROOF	character (LEN=28)		' '		
CFTYP_EMIS_ROOF	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_HC_ROOF	real		none	roof layers heat capacity	($J/K/m^3$)
CFNAM_HC_ROOF	character (LEN=28)		' '		
CFTYP_HC_ROOF	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
XUNIF_TC_ROOF	real		none	roof layers thermal conductivity	($(W/K/m)$)
CFNAM_TC_ROOF	character (LEN=28)		' '		
CFTYP_TC_ROOF	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV',	none		

		'ASCLLV'			
Fortran name	Fortran type	values	default	description	unit
XUNIF_D_ROOF	real		none	roof layers depth	(*m*)
CFNAM_D_ROOF	character (LEN=28)		' '		
CFTYP_D_ROOF	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_ALB_ROAD	real		none	road albedo	(-)
CFNAM_ALB_ROAD	character (LEN=28)		' '		
CFTYP_ALB_ROAD	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_EMIS_ROAD	real		none	road emissivity	(-)
CFNAM_EMIS_ROAD	character (LEN=28)		' '		
CFTYP_EMIS_ROAD	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
XUNIF_HC_ROAD	real		none	road layers heat capacity	$(J/K/m^3)$
CFNAM_HC_ROAD	character (LEN=28)		' '		
CFTYP_HC_ROAD	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_TC_ROAD	real		none	road layers thermal conductivity	(*W/K/m*)
CFNAM_TC_ROAD	character (LEN=28)		' '		
CFTYP_TC_ROAD	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
XUNIF_D_ROAD	real		none	road layers depth	(*m*)
CFNAM_D_ROAD	character (LEN=28)		' '		
CFTYP_D_ROAD	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_ALB_WALL	real		none	wall albedo	(-)
CFNAM_ALB_WALL	character (LEN=28)		' '		
CFTYP_ALB_WALL	character (LEN=6)	'DIRECT', 'BINLLF''BINLLV', 'ASCLLV'	none		
XUNIF_EMIS_WALL	real		none	wall emissivity	(-)
CFNAM_EMIS_WALL	character (LEN=28)		' '		
CFTYP_EMIS_WALL	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
Fortran name	Fortran type	values	default	description	unit

XUNIF_HC_WALL	real		none	wall layers heat capacity	(*J/K/m^3*)
CFNAM_HC_WALL	character (LEN=28)		' '		
CFTYP_HC_WALL	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_TC_WALL	real		none	wall layers thermal conductivity	(*W/K/m*)
CFNAM_TC_WALL	character (LEN=28)		' '		
CFTYP_TC_WALL	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
XUNIF_D_WALL	real		none	wall layers depth	(*m*)
CFNAM_D_WALL	character (LEN=28)		' '		
CFTYP_D_WALL	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_H_TRAFFIC	real		none	anthropogenic sensible heat fluxes due to traffic	(*W/m^2*)
CFNAM_H_TRAFFIC	character (LEN=28)		' '		
CFTYP_H_TRAFFIC	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_LE_TRAFFIC	real		none	anthropogenic latent heat fluxes due to traffic	(*W/m^2*)
CFNAM_LE_TRAFFIC	character (LEN=28)		' '		
CFTYP_LE_TRAFFIC	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
XUNIF_H_INDUSTRIES	real		none	anthropogenic sensible heat fluxes due to factories	(*W/m^2*)
CFNAM_H_INDUSTRIES	character (LEN=28)		' '		
CFTYP_H_INDUSTRIES	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	none		
XUNIF_LE_INDUSTRIES	real		none	anthropogenic latent heat fluxes due to factories	(*W/m^2*)
CFNAM_LE_INDUSTRIES	character (LEN=28)		' '		
CFTYP_LE_INDUSTRIES	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		

Fortran name	Fortran type	values	default	description	unit
XUNIF_ROUGH_ROOF	real		none	roof roughness coefficient	
CFNAM_ROUGH_ROOF	character (LEN=28)		' '		
CFTYP_ROUGH_ROOF	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		
XUNIF_ROUGH_WALL	real		none	wall roughness coefficient	
CFNAM_ROUGH_WALL	character (LEN=28)		' '		
CFTYP_ROUGH_WALL	character (LEN=6)	'DIRECT', 'BINLLF','BINLLV', 'ASCLLV'	none		

4.4.3.3. TEB_GARDEN scheme: NAM_DATA_TEB_GARDEN

Over urban areas, if ECOCLIMAP is not used, all vegetation surface parameters have to be specified by the user in namelist **NAM_DATA_TEB_GARDEN**.

This namelist is a simplified version of NAM_DATA_ISBA, where only No, Low and High vegetation are distinguished. Moreover, only parameters LAI, Height of trees, Fraction are given, other are imposed by the scheme in the model code.

Fortran name	Fortran type	values	default value	description	unit
NTIME	integer	1, 12	12	time dimension	
CTYP_GARDEN_HVEG	character(LEN=4)	TREE, CONI, EVER	TREE	type of high vegetation	
CTYP_GARDEN_LVEG	character(LEN=4)	GRAS, TROG, PARK, C3, C4, IRR	PARK	type of low vegetation	
CTYP_GARDEN_NVEG	character(LEN=4)	NO, ROCK, SNOW	NO	type of bare soil	
XUNIF_FRAC_HVEG	real	between 0. and 1.	1.E+20	fraction of high vegetation	
XUNIF_FRAC_LVEG	real	between 0. and 1.	1.E+20	fraction of low vegetation	
XUNIF_FRAC_NVEG	real	between 0. and 1.	1.E+20	fraction of bare soil	
XUNIF_LAI_HVEG	real		1.E+20	LAI of high vegetation	m^2/m^2
XUNIF_LAI_LVEG	real		1.E+20	LAI of low vegetation	m^2/m^2
XUNIF_H_HVEG	real		1.E+20	height of trees	m
CFNAM_FRAC_HVEG	character(LEN=28)		''	file name	
CFNAM_FRAC_LVEG	character(LEN=28)		''	file name	
CFNAM_FRAC_NVEG	character(LEN=28)		''	file name	
CFNAM_LAI_HVEG	character(LEN=28)		''	file name	
CFNAM_LAI_LVEG	character(LEN=28)		''	file name	
CFNAM_H_HVEG	character(LEN=28)		''	file name	
CFTYP_FRAC_HVEG	character(LEN=28)	'DIRECT','ASCLLV', 'BINLLV','BINLLF'	''	file type	
CFTYP_FRAC_LVEG	character(LEN=28)	'DIRECT','ASCLLV', 'BINLLV','BINLLF'	''	file type	

CFTYP_FRAC_NVEG	character(LEN=28)	'DIRECT','ASCLLV', 'BINLLV','BINLLF'	''	file type	
CFTYP_LAI_HVEG	character(LEN=28)	'DIRECT','ASCLLV', 'BINLLV','BINLLF'	''	file type	
CFTYP_LAI_LVEG	character(LEN=28)	'DIRECT','ASCLLV', 'BINLLV','BINLLF'	''	file type	
CFTYP_H_HVEG	character(LEN=28)	'DIRECT','ASCLLV', 'BINLLV','BINLLF'	''	file type	

4.4.3.4. BEM scheme: NAM_DATA_BEM

Fortran name	Fortran type	values	default	description	units
NPAR_FLOOR_LAYER	integer		1	bumber of layers in roofs	
XUNIF_HC_FLOOR	real		none	heat capacity of floor layers	J m-3 K-1
CFNAM_HC_FLOOR	character (LEN=28)		''		
CFTYP_HC_FLOOR	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	''		
XUNIF_TC_FLOOR	real		none	thermal conductivity of floor layers	W m-1 K-1
CFNAM_TC_FLOOR	character (LEN=28)		''		
CFTYP_TC_FLOOR	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	''		
XUNIF_D_FLOOR	real		none	thickness of floor layers	m
CFNAM_D_FLOOR	character (LEN=28)		''		
CFTYP_D_FLOOR	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	''		
XUNIF_FLOOR_HEIGHT	real		none	building floor height	m
CFNAM_FLOOR_HEIGHT	character (LEN=28)		''		
CFTYP_FLOOR_HEIGHT	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	''		
XUNIF_TCOOL_TARGET	real		none		
CFNAM_TCOOL_TARGET	character (LEN=28)		''	cooling setpoint of indoor air	
CFTYP_TCOOL_TARGET	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	''		
XUNIF_THEAT_TARGET	real		none	heating setpoint of indoor air	
CFNAM_THEAT_TARGET	character (LEN=28)		''		
CFTYP_THEAT_TARGET			''		

	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'			
XUNIF_F_WASTE_CAN	real		none	fraction of waste heat into the canyon	
CFNAM_F_WASTE_CAN	character (LEN=28)		"		
CFTYP_F_WASTE_CAN	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	"		
XUNIF_EFF_HEAT	real		none	efficiency of the heating system	
CFNAM_EFF_HEAT	character (LEN=28)		"		
CFTYP_EFF8HEAT	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	"		
XUNIF_F_WATER_COND	real		none	fraction of evaporation for condensers	
CFNAM_F_WATER_COND	character (LEN=28)		"		
CFTYP_F_WATER_COND	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	"		
XUNIF_HR_TARGET	real		none	relative humidity setpoint	
CFNAM_HR_TARGET	character (LEN=28)		"		
CFTYP_HR_TARGET	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	"		
XUNIF_QIN	real		none	internal heat gains	W m-2 (floor)
CFNAM_QIN	character (LEN=28)		"		
CFTYP_QIN	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	"		
XUNIF_QIN_FRAD	real		none	radiant fraction of internal heat gains	
CFNAM_QIN_FRAD	character (LEN=28)		"		

CFTYP_QIN_FRAD	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	"		
XUNIF_QIN_FLAT	real		none	latent fraction of internal heat gains	
CFNAM_QIN_FLAT	character (LEN=28)		"		
CFTYP_QIN_FLAT	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	"		
XUNIF_SHGC	real		none	solar transmitance of windows	
CFNAM_SHGC	character (LEN=28)		"		
CFTYP_SHGC	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	"		
XUNIF_U_WIN	real		none	glazing thermal resistance	K m W-2
CFNAM_U_WIN	character (LEN=28)		"		
CFTYP_U_WIN	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	"		
XUNIF_GR	real		none	glazing ratio	
CFNAM_GR	character (LEN=28)		"		
CFTYP_GR	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	"		
XUNIF_SHGC_SH	real		none	solar transmitance of windows + shading	
CFNAM_SHGC_SH	character (LEN=28)		"		
CFTYP_SHGC_SH	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	"		
XUNIF_INF	real		none	infiltration/ventialtion flow rate	AC/H
CFNAM_INF	character (LEN=28)		"		
CFTYP_INF			v		

	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'			
XUNIF_V_VENT	real		none	ventilation flow rate	AC/H
CFNAM_V_VENT	character (LEN=28)		''		
CFTYP_V_VENT	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	''		
XUNIF_CAP_SYS_HEAT	real		none	capacity of the heating system	W m-2 (bld)
CFNAM_CAP_SYS_HEAT	character (LEN=28)		''		
CFTYP_CAP_SYS_HEAT	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	''		
XUNIF_CAP_SYS_RAT	real		none	rated capacity of the cooling system	W m-2 (bld)
CFNAM_CAP_SYS_RAT	character (LEN=28)		''		
CFTYP_CAP_SYS_RAT	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	''		
XUNIF_T_ADP	real		none	Apparatus dewpoint temperature of the cooling coil	K
CFNAM_T_ADP	character (LEN=28)		''		
CFTYP_T_ADP	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	''		
XUNIF_M_SYS_RAT	real		none	rated HVAC mass flow rate	kg s-1 m-2 (bld)
CFNAM_M_SYS_RAT	character (LEN=28)		''		
CFTYP_M_SYS_RAT	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	''		
XUNIF_COP_RAT	real		none	rated COP of the cooling system	
CFNAM_COP_RAT			''		

	character (LEN=28)				
CFTYP_COP_RAT	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	"		
XUNIF_T_SIZE_MAX	real		none		
CFNAM_T_SIZE_MAX	character (LEN=28)		"		
CFTYP_T_SIZE_MAX	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	"		
XUNIF_T_SIZE_MIN	real		none		
CFNAM_T_SIZE_MIN	character (LEN=28)		"		
CFTYP_T_SIZE_MIN	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	"		
XUNIF_SHADE	real	0 (inactivated), 1 (activated)	none		
CFNAM_SHADE	character (LEN=28)	flag to activate shading devices	"		
CFTYP_SHADE	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	"		
XUNIF_NATVENT	real	0 (NONE), 1 (MANU), 2 (AUTO)	none	flag to describe bld surventilation solution	
CFNAM_NATVENT	character (LEN=28)		"		
CFTYP_NATVENT	character (LEN=6)	'DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV'	"		

4.4.3.5. TEB_GREENROOF scheme: namelist NAM_DATA_TEB_GREENROOF

Fortran name	Fortran type	values	default value	description	unit
NTIME_GR	integer	1	1	time dimension	
NLAYER_GR	integer	â ¤6	6	number of layers in greenroofs	
CTYP_GR	character(len=5)	'GRASS','SEDUM'	'GRASS'	type of vegetation	
XUNIF_OM_GR	real,dimension(6)	between 0. and 1.	1.E+20	fraction of organic matter	
XUNIF_CLAY_GR	real,dimension(6)	between 0. and 1.	1.E+20	fraction of clay	
XUNIF_SAND_GR	real,dimension(6)	between 0. and 1.	1.E+20	fraction of sand	
XUNIF_LAI_GR	real,dimension(1)		1.E+20	LAI of green roof	m^2/m^2
CFNAM_OM_GR	character(LEN=28)		''	file name	
CFNAM_CLAY_GR	character(LEN=28)		''	file name	
CFNAM_SAND_GR	character(LEN=28)		''	file name	
CFNAM_LAI_GR	character(LEN=28)		''	file name	
CFTYP_OM_GR	character(LEN=28)	'DIRECT','ASCLLV', 'BINLLV','BINLLF'	''	file type	
CFTYP_CLAY_GR	character(LEN=28)	'DIRECT','ASCLLV', 'BINLLV','BINLLF'	''	file type	
CFTYP_SAND_GR	character(LEN=28)	'DIRECT','ASCLLV', 'BINLLV','BINLLF'	''	file type	
CFTYP_LAI_GR	character(LEN=28)	'DIRECT','ASCLLV', 'BINLLV','BINLLF'	''	file type	

4.5. Specificities of ECOCLIMAP II classification

4.5.1. Namelist NAM_ECOCLIMAP2

This namelist allows to choose which LAI is used: a climatological one (average over years 2002-2006) or a specific year (between 2002 and 2006). This is the place to define irrigation file.

Fortran name	Fortran type	values	default value
LCLIM_LAI	logical		.TRUE.
YIRRIG	character (LEN=28)		' '

- LCLIM_LAI: if .TRUE., climatological LAI is computed otherwise, the LAI corresponding to current year (if between 2002 and 2006) is used.
- YIRRIG: irrigation file name

4.6. Namelist to add user's own fields

4.6.1. Namelist NAM_DUMMY_PGD

This namelist allows to incorporate into the physiographic file any surface field.

You can treat up to 999 such fields. These fields will be written on all the files you will use later(after prognostic fields initialization, or during and after run, etc...). Their name in the files are 'DUMMY_GRnnn', where nnn goes from 001 to 999.

During the execution of the programs, these fields are stored in the XDUMMY_FIELDS(:,:) (first dimension: spatial dimension, second dimension: total number of fields), in the module MODD_DUMMY_SURF_FIELD$n. You must modify the fortran source, where you want to use them.

Fortran name	Fortran type	default value
NDUMMY_PGD_NBR	integer	0
CDUMMY_PGD_NAME(:)	1000 * character (LEN=20)	1000 * ' '
CDUMMY_PGD_FILE(:)	1000 * character (LEN=28)	1000 * ' '
CDUMMY_PGD_COMMENT(:)	1000 * character (LEN=40)	1000 * ' '
CDUMMY_PGD_FILETYPE(:)	1000 * character (LEN=6)	1000 *' '
CDUMMY_PGD_AREA(:)	1000 * character (LEN=3)	1000 *'ALL'
CDUMMY_PGD_ATYPE(:)	1000 * character (LEN=3)	1000 *'ARI'

Only the first NDUMMY_PGD_NBR values in these arrays are meaningfull.

- NDUMMY_PGD_NBR: number of dummy fields.
- CDUMMY_PGD_NAME(:): list of the dummy fields you want to initialize with your own data. You can give any name you want. This is a way to describe what is the field. This information is not used by the program. It is just written in the FM files.
- CDUMMY_PGD_FILE(:): list of the names of the files containing the data for the fields you have specified in CDUMMY_PGD_NAME(:).
- CDUMMY_PGD_FILETYPE(:): list of the types of the files containing the data for the fields you have specified in CDUMMY_PGD_NAME(:) ('DIRECT', 'LATLON', 'BINLLF', 'BINLLV', 'ASCLLV').
- CDUMMY_PGD_AREA(:):area of meaningfullness of the fields you have specified in CDUMMY_PGD_NAME(:) ('ALL', 'NAT', 'TWN', 'SEA', 'WAT', 'LAN', respectively for everywhere, natural areas, town areas, sea, inland waters, land = natural cover + town). For example, oceanic emission of DNS is relevant on 'SEA'.
- CDUMMY_PGD_ATYPE(:) :type of averaging (during **PGD** for the fields you have specified in CDUMMY_PGD_NAME(:) ('ARI', 'INV', 'LOG', respectively for arithmetic, inverse and logarithmic averaging).

4.7. Namelist for chemistry anthropogenic emissions

4.7.1. Namelist NAM_CH_EMIS_PGD

This namelist is used to initialize chemistry components emissions.

You can treat up to 999 such fields. These fields will be written on all the files you will use later (after prognostic fields initialization, or during and after run, etc...). Their name in the files are 'EMIS_GRnnn', where nnn goes from 001 to 999.

During the execution of the programs, these fields are stored in the XEMIS_GR_FIELDS(:,:) (first dimension: spatial dimension, second dimension: total number of fields), in the module MODD_EMIS_GR_FIELD$n. The temporal evolution, the aggregation of prescribed emissions and the link with the corresponding chemical prognostic variables are handled by the subroutine CH_EMISSION_FLUXn.f90

Fortran name	Fortran type	default value
NEMIS_PGD_NBR	integer	0
CEMIS_PGD_NAME(:)	1000 * character (LEN=20)	1000 * ' '
CEMIS_PGD_FILE(:)	1000 * character (LEN=28)	1000 * ' '
CEMIS_PGD_COMMENT(:)	1000 * character (LEN=40)	1000 * ' '
NEMIS_PGD_TIME	integer	0
CEMIS_PGD_FILETYPE(:)	1000 * character (LEN=6)	1000 *'DIRECT'
CEMIS_PGD_AREA(:)	1000 * character (LEN=3)	1000 *'ALL'
CEMIS_PGD_ATYPE(:)	1000 * character (LEN=3)	1000 *'ARI'

Only the first NEMIS_PGD_NBR values in these arrays are meaningfull.

- NEMIS_PGD_NBR: number of dummy fields.
- CEMIS_PGD_NAME(:): list of the dummy fields you want to initialize with your own data. You can give any name you want. This is a way to describe what is the field. This information is not used by the program. It is just written in the FM files.
- CEMIS_PGD_FILE(:): list of the names of the files containing the data for the fields you have specified in CEMIS_PGD_NAME(:).
- CEMIS_PGD_COMMENT(:): list of the comments associated to each emission field.
- NEMIS_PGD_TIME(:): list of the time of the files containing the data for the fields you have specified in CEMIS_PGD_NAME(:).
- CEMIS_PGD_FILETYPE(:): list of the types of the files containing the data for the fields you have specified in CEMIS_PGD_NAME(:) ('DIRECT', 'BINLLF', 'BINLLV', 'ASCLLV').
- CEMIS_PGD_AREA(:)>: area of meaningfullness of the fields you have specified in CEMIS_PGD_NAME(:) ('ALL', 'NAT', 'TWN', 'SEA', 'WAT', 'LAN', respectively for everywhere, natural areas, town areas, sea, inland waters, land = natural cover + town). For example, oceanic emission of DNS is relevant on 'SEA'.
- CEMIS_PGD_ATYPE(:): type of averaging (during PGD for the fields you have specified in CEMIS_PGD_NAME(:) ('ARI', 'INV', 'LOG', respectively for arithmetic, inverse and

logarithmic averaging). Example:

```
&NAM_CH_EMIS_PGD NEMIS_PGD_NBR  = 2,
              CEMIS_PGD_NAME(1)='COE',
              NEMIS_PGD_TIME(1)=0,
              CEMIS_PGD_COMMENT(1)='CO_00h00',
              CEMIS_PGD_AREA(1)='LAN',
              CEMIS_PGD_ATYPE(1)='ARI',
              CEMIS_PGD_FILE(1)='co_00.asc',
              CEMIS_PGD_FILETYPE(1)='ASCLLV',
              CEMIS_PGD_NAME(2)='COE',
              NEMIS_PGD_TIME(2)=43200,
              CEMIS_PGD_COMMENT(2)='CO_12h00',
              CEMIS_PGD_AREA(2)='LAN',
              CEMIS_PGD_ATYPE(2)='ARI',
              CEMIS_PGD_FILE(2)='co_12.asc',
              CEMIS_PGD_FILETYPE(2)='ASCLLV',
              CEMIS_PGD_NAME(3)='DMSE',
              NEMIS_PGD_TIME(3)=0,
              CEMIS_PGD_COMMENT(3)='dms_cte',
              CEMIS_PGD_AREA(3)='SEA',
              CEMIS_PGD_ATYPE(3)='ARI',
              CEMIS_PGD_FILE(3)='dms.asc',
              CEMIS_PGD_FILETYPE(3)='ASCLLV' /
```

4.7.2. Namelist NAM_CH_EMISSIONS

Fortran name	Fortran type	values	default value
CCH_EMIS	character(LEN=4)	'NONE','AGGR','SNAP'	'NONE'

- CCH_EMIS : option for emissions computations:
  - "NONE" : no emission
  - "AGGR" : one aggregated value for each specie and hour
  - "SNAP" : from SNAP data using potential emission & temporal profile

4.7.3. Namelist NAM_CH_SNAP_EMIS_PGD

Fortran name	Fortran type	values	default value
NEMIS_NBR	integer		0
NEMIS_SNAP	integer		0
CEMIS_NAME	character(LEN=6)		''
CEMIS_COMMENT	character(LEN=40)		''
CSNAP_MONTHLY_FILE	character(LEN=28)		''
CSNAP_DAILY_FILE	character(LEN=28)		''
CSNAP_HOURLY_FILE	character(LEN=28)		''
CSNAP _POTENTIAL_FILE	character(LEN=50)		''
CSNAP_POTENTIAL_FILETYPE	character(LEN=6)	'DIRECT', 'BINLLV', 'BINLLF', ASCLLV', 'LATLON'	''
XUNIF_SNAP	real		none
XUNIF_DELTA_LEGAL_TIME	real		none
CDELTA_LEGAL_TIME_FILE	character(LEN=50)		''
CDELTA_LEGAL_TIME_FILETYPE	character(LEN=6)	'DIRECT', 'BINLLV', 'BINLLF', 'ASCLLV', 'LATLON'	''

- NEMIS_NBR : number of chemical pgd fields chosen by user
- NEMIS_SNAP : number of snaps
- CEMIS_NAME : name of the chemical fields (emitted species)
- CEMIS_COMMENT : comment on the chemical fields (emitted species)
- CSNAP_MONTHLY_FILE : name of the snap ASCII monthly file
- CSNAP_DAILY_FILE : name of the snap ASCII daily file
- CSNAP_HOURLY_FILE : name of the snap ASCII hourly file
- CSNAP_POTENTIAL_FILE : name of the snap potential file
- CSNAP_POTENTIAL_FILETYPE : type of the snap potential file
- XUNIF_SNAP : uniform value for the snap potential (emission factore for each chemical specie and each snap)
- XUNIF_DELTA_LEGAL_TIME : uniform value for the difference (in hours) between lagal time and UTC time
- CDELTA_LEGAL_TIME_FILE : name of file for the difference between legal time and UTC time
- CDELTA_LEGAL_TIME_FILETYPE : filetype for the difference between legal time and UTC time

5. Initialization of the prognostic fields

5.1. Overview of fields computation: PREP

The prognostic fields (temperature, humidity, ice, snow, etc...) are averaged or interpolated on the specified grid by the program **PREP**. They are stored in the surface file. The computation is done separately for each surface scheme.

During the **PREP** facility :

1. You initializes the date of the surface
2. You initializes the prognostic variables of the chosen sea scheme
3. You initializes the prognostic variables of the chosen lake scheme
4. You initializes the prognostic variables of the chosen vegetation scheme
5. You initializes the prognostic variables of the chosen town scheme

Here are presented the initialization procedures for the schemes that need such information (for example, scheme "IDEAL " does not need any information here, but modificaton of the code source *init_ideal_flux.f90*).

Note that for each scheme, and for some for each variable of the scheme, it is possible to initialize the prognostic fields either form an operationnal or research model, or using prescribed (usually uniform) fields.

5.2. Date initialization and default input data file for all schemes

5.2.1. Namelist NAM_PREP_SURF_ATM

This namelist information is used to (possibly):

- initialize the date of all surface schemes. The namelist information is used only if no input data file is used, either from namelist or by fortran code (as in MESONH programs). If a file is used, the date is read in it.
- define the default file in which each scheme can read the needed data (e.g. temperature).

Note that, all the information given in this namelist can be erased for each scheme by the namelist corresponding to this scheme, as the information in the shceme namelists have priority on namelist NAM_PREP_SURF_ATM.

Fortran name	Fortran type	values	default value
CFILE	string of 28 characters		atmospheric prep file used in the program calling the surface facilities, if any - none otherwise
CFILETYPE	string of 6 characters	'MESONH', 'GRIB ', 'ASCII', 'LFI '	type of the atmospheric prep file, if any
CFILEPGD	string of 28 characters		atmospheric pgd file used in the program calling the surface facilities, if any - none otherwise
CFILEPGDTYPE	string of 6 characters	'MESONH', 'GRIB ', 'ASCII', 'LFI '	type of the atmospheric pgd file, if any
NYEAR	integer		none
NMONTH	integer		none
NDAY	integer		none
XTIME	real		none

- CFILE / CFILEPGD: name of the prep / pgd file used to define
  1. the date.
  2. the file in which to read the needed data (e.g. temperature).
     The use of a file or prescribed value in each scheme namelist has priority on the data in CFILE / CFILEPGD file of namelist NAM_PREP_SURF_ATM.
     CFILE and CFILEPGD can identify the same file.
- CFILETYPE / CFILEPGDTYPE: type of the CFILE / CFILEPGD file, if the latter is provided. CFILETYPE / CFILEPGDTYPE must then be given. The following values are currently usable:
  - "MESONH" : the file type is a MESONH file.
  - "GRIB " : the file type is a GRIB file, coming from any of these models:
    1. "ECMWF " : european center forecast model
    2. "ARPEGE" : Arpege french forecast model
    3. "ALADIN" : Aladin french forecast local model
    4. "MOCAGE" : Mocage french research chemistry model

    - "ASCII " : ASCII Surfex PREP/PGD file
    - "LFI " : LFI Surfex PREP/PGD file
- NYEAR : year of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).
- NMONTH : month of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).
- NDAY : day of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).
- XTIME : time from midnight of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read). (seconds).

5.3. Sea scheme "SEAFLX"

5.3.1. Namelist NAM_PREP_SEAFLUX

This namelist information is used to initialize the "SEAFLX" sea scheme temperature.

Fortran name	Fortran type	values	default value
XSST_UNIF	real		none
CFILE_SEAFLX	string of 28 characters		CFILE in NAM_PREP_SURF_ATM
CTYPE	string of 6 characters	'MESONH', 'GRIB ' , 'NETCDF ','ASCII ','LFI '	CFILETYPE in NAM_PREP_SURF_ATM
CFILEPGD_SEAFLX	string of 28 characters		CFILEPGD in NAM_PREP_SURF_ATM
CTYPEPGD	string of 6 characters	'MESONH', 'GRIB ' , 'ASCII ', 'LFI '	CFILEPGDTYPE in NAM_PREP_SURF_ATM
NYEAR	integer		none
NMONTH	integer		none
NDAY	integer		none
XTIME	real		none
LSEA_SBL	logical		F
LOCEAN_MERCATOR	logical		F
LOCEAN_CURRENT	logical		F
XTIME_REL	real		25920000.
LCUR_REL	logical		F
LTS_REL	logical		F
LZERO_FLUX	logical		F
LCORR_FLUX	logical		F
XCORFLX	real		0.
LDIAPYC	logical		F

- XSST_UNIF : uniform prescribed value of Sea Surface Temperature. This prescribed value, if defined, has priority on the use of CFILE_SEAFLX data.
- CFILE_SEAFLX / CFILEPGD_SEAFLX: name of the PREP/PGD files used to define the Sea surface Temperature. **The use of a file or prescribed value XSST_UNIF has priority on the data in CFILE_SEAFLX file.**
- CTYPE / CTYPEPGD: type of the CFILE_SEAFLX / CFILEPGD_SEAFLX files, if the latter is provided. CTYPE must then be given. The following values are currently usable:
    - "MESONH" : the file type is a MESONH file.
    - "GRIB " : the file type is a GRIB file, coming from any of these models:
        1. "ECMWF " : european center forecast model
        2. "ARPEGE" : Arpege french forecast model
        3. "ALADIN" : Aladin french forecast local model

4. "MOCAGE" : Mocage french research chemistry model

- ♦ "NETCDF" : the file type is a NETCDF file, coming from MERCATOR (possible only for CTYPE)
- ♦ "ASCII ": PREP/PGD Surfex ASCII file
- ♦ "LFI ": PREP/PGD Surfex LFI file

- NYEAR : year of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).
- NMONTH : month of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).
- NDAY : day of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).
- XTIME : time from midnight of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read). (seconds).
- LSEA_SBL : activates surface boundary multi layer scheme over sea.
- LOCEAN_MERCATOR : oceanic variables initialized from MERCATOR if T
- LOCEAN_CURRENT : initial ocean state with current (if F ucur=0, vcur=0)
- XTIME_REL : time of relaxation (s)
- LCUR_REL : flag for relaxation on current
- LTS_REL : flag for relaxation on ocean temperature
- LZERO_FLUX : flag for testing zero incoming flux at the ocean surface
- LCORR_FLUX : flag for flux correction
- XCORFLX : correction coefficient for surface fluxes
- LDIAPYC : flag for diapycnal mixing activation

5.4. Lake scheme "WATFLX"

5.4.1. Namelist NAM_PREP_WATFLUX

This namelist information is used to initialize the "WATFLX" sea scheme temperature.

Fortran name	Fortran type	values	default value
XTS_WATER_UNIF	real		none
CFILE_WATFLX	string of 28 characters		CFILE in NAM_PREP_SURF_ATM
CTYPE	string of 6 characters	'MESONH', 'GRIB ', 'ASCII ','LFI '	CFILETYPE in NAM_PREP_SURF_ATM
CFILEPGD_WATFLX	string of 28 characters		CFILEPGD in NAM_PREP_SURF_ATM
CTYPEPGD	string of 6 characters	'MESONH', 'GRIB ', 'ASCII ','LFI '	CFILEPGDTYPE in NAM_PREP_SURF_ATM
NYEAR	integer		none
NMONTH	integer		none
NDAY	integer		none
XTIME	real		none
LWAT_SBL	logical		F

- XTS_WATER_UNIF : uniform prescribed value of water surface temperature supposed at an **altitude of 0m** (mean sea level altitude). The temperature is then modified for each point depending on its altitude, following a uniform vertical gradient of -6.5 $K\ km^{-1}$. This prescribed value, if defined, has priority on the use of CFILE_WATFLX data.
- CFILE_WATFLX / CFILEPGD_WATFLX: name of the PREP / PGD files used to define the Sea surface Temperature. **The use of a file or prescribed value XTS_WATER_UNIF has priority on the data in CFILE_WATFLX file.**
- CTYPE / CTYPEPGD: type of the CFILE_WATFLX / CFILEPGD_WATFLX file, if the latter is provided. CTYPE / CTYPEPGD must then be given. The following values are currently usable:
    - "MESONH" : the file type is a MESONH file.
    - "GRIB " : the file type is a GRIB file, coming from any of these models:
        1. "ECMWF " : european center forecast model
        2. "ARPEGE" : Arpege french forecast model
        3. "ALADIN" : Aladin french forecast local model
        4. "MOCAGE" : Mocage french research chemistry model
    - "ASCII " : PREP / PGD Surfex ASCII file
    - "LFI ": PREP/PGD Surfex LFI file
- NYEAR : year of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).
- NMONTH : month of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).

- NDAY : day of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).
- XTIME : time from midnight of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read). (seconds).
- LWAT_SBL : activates surface boundary multi layer scheme over inland water.

5.5. Lake scheme "FLAKE"

5.5.1. Namelist NAM_PREP_FLAKE

This namelist information is used to initialize the "FLAKE" sea scheme temperature.

Fortran name	Fortran type	values	default value
XTS_UNIF	real		none
XUNIF_T_SNOW	real		min(273.15,XTS_WATER)
XUNIF_T_ICE	real		min(273.15,XTS_WATER)
XUNIF_T_WML	real		min(273.15,XTS_WATER)
XUNIF_T_BOT	real		TS_WATER or 277.15 if TS_WATER â ¤ 273.15
XUNIF_T_B1	real		TS_WATER-0.1 or 277.05 if TS_WATER â ¤ 273.15
XUNIF_CT	real		0.5
XUNIF_H_SNOW	real		0.
XUNIF_H_ICE	real		0. or 0.01 if XTS_WATER < 273.15
XUNIF_H_ML	real		XWATER_DEPTH or XWATER_DEPTH/2 if TS_WATER < 273.15
XUNIF_H_B1	real		0.
CFILE_FLAKE	string of 28 characters		CFILE in NAM_PREP_SURF_ATM
CTYPE	string of 6 characters	'MESONH', 'GRIB ' , 'ASCII ,'LFI '	CFILETYPE in NAM_PREP_SURF_ATM
CFILEPGD_FLAKE	string of 28 characters		CFILEPGD in NAM_PREP_SURF_ATM
CTYPEPGD	string of 6 characters	'MESONH', 'GRIB ' , 'ASCII ','LFI '	CFILEPGDTYPE in NAM_PREP_SURF_ATM
NYEAR	integer		none
NMONTH	integer		none
NDAY	integer		none
XTIME	real		none
LWAT_SBL	logical		F

- XTS_UNIF : uniform prescribed value of water surface temperature supposed at an **altitude of 0m** (mean sea level altitude). The temperature is then modified for each point depending on its altitude, following a uniform vertical gradient of -6.5 $K\ km^{-1}$. This prescribed value, if defined, has priority on the use of CFILE_FLAKE data.
- XUNIF_T_SNOW : surface temperature of snow (K)
- XUNIF_T_ICE : surface temperature at the ice-atmosphere or at the ice-snow interface (K)
- XUNIF_T_WML : mixed-layer temperature (K)
- XUNIF_T_BOT : water temperature at the bottom of the lake (K)

- XUNIF_T_B1 : temperature at the bottom of the upper layer of sediments (K)
- XUNIF_CT : shape factor (thermocline)
- XUNIF_H_SNOW : snow layer thickness (m)
- XUNIF_H_ICE : ice layer thickness (m)
- XUNIF_H_ML : thickness of the mixed-layer (m)
- XUNIF_H_B1 : thickness of the upper level of the active sediments (m)
- CFILE_FLAKE / CFILEPGD_FLAKE: name of the PREP and PGD files used to define the Sea surface Temperature. **The use of a file or prescribed value XTS_WATER_UNIF has priority on the data in CFILE_FLAKE file.**
- CTYPE / CTYPEPGD: type of the CFILE_FLAKE / CFILEPGD_FLAKE files, if the latter is provided. CTYPE / CTYPEPGD must then be given. The following values are currently usable:
    - ♦ "MESONH" : the file type is a MESONH file.
    - ♦ "GRIB " : the file type is a GRIB file, coming from any of these models:
        1. "ECMWF " : european center forecast model
        2. "ARPEGE" : Arpege french forecast model
        3. "ALADIN" : Aladin french forecast local model
        4. "MOCAGE" : Mocage french research chemistry model
    - ♦ "ASCII" : Surfex PREP / PGD ASCII file
    - ♦ "LFI " : Surfex PREP / PGD LFI file
- NYEAR : year of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).
- NMONTH : month of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).
- NDAY : day of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).
- XTIME : time from midnight of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read). (seconds).
- LWAT_SBL : activates surface boundary multi layer scheme over inland water.

5.6. Vegetation scheme "ISBA "

5.6.1. Namelist NAM_PREP_ISBA

This namelist information is used to initialize the "ISBA " vegetation scheme variables: soil temperature profile, soil water and ice profiles, water intercepted by leaves, snow.

Fortran name	Fortran type	values	default value
XHUG_SURF	real		none
XHUG_ROOT	real		none
XHUG_DEEP	real		none
XHUGI_SURF	real		none
XHUGI_ROOT	real		none
XHUGI_DEEP	real		none
CFILE_HUG_SURF	string of 28 characters		CFILE_HUG in this namelist
CFILE_HUG_ROOT	string of 28 characters		CFILE_HUG in this namelist
CFILE_HUG_DEEP	string of 28 characters		CFILE_HUG in this namelist
CFILE_HUG	string of 28 characters		CFILE_ISBA in this namelist
CTYPE_HUG	string of 6 characters	'MESONH', 'GRIB ','LFI ','ASCII ','ASCLLV','NETCDF'	CTYPE in this namelist
XTG_SURF	real		none
XTG_ROOT	real		none
XTG_DEEP	real		none
CFILE_TG_SURF	string of 28 characters		CFILE_TG in this namelist
CFILE_TG_ROOT	string of 28 characters		CFILE_TG in this namelist
CFILE_TG_DEEP	string of 28 characters		CFILE_TG in this namelist
CFILE_TG	string of 28 characters		CFILE_ISBA in this namelist
CTYPE_TG	string of 6 characters	'MESONH', 'GRIB ', 'LFI ', 'ASCII ', 'ASCLLV','NETCDF'	CTYPE in this namelist
CFILE_ISBA	string of 28 characters		CFILE in NAM_PREP_SURF_ATM
CTYPE	string of 6 characters	'MESONH', 'GRIB ', 'ASCII ','LFI '	CFILETYPE in NAM_PREP_SURF_ATM
CFILEPGD_ISBA	string of 28 characters		CFILEPGD in NAM_PREP_SURF_ATM
CTYPEPGD		'MESONH', 'GRIB ', 'ASCII ','LFI '	

	string of 6 characters		CFILEPGDTYPE in NAM_PREP_SURF_ATM
NYEAR	integer		none
NMONTH	integer		none
NDAY	integer		none
XTIME	real		none
LISBA_CANOPY	logical		F

- XHUG_SURF : uniform prescribed value of liquid soil water index (SWI) for the surface soil layer. This prescribed value, if defined, has priority on the use of CFILE_HUG and CFILE_ISBA data.
- XHUG_ROOT : uniform prescribed value of liquid soil water index (SWI) for the root zone soil layer(s). This prescribed value, if defined, has priority on the use of CFILE_HUG and CFILE_ISBA data.
- XHUG_DEEP : uniform prescribed value of liquid soil water index (SWI) for the deep soil layer(s). This prescribed value, if defined, has priority on the use of CFILE_HUG and CFILE_ISBA data.
- XHUGI_SURF : uniform prescribed value of ice soil water index (SWI) for the surface soil layer. This prescribed value, if defined, has priority on the use of CFILE_HUG and CFILE_ISBA data.
- XHUGI_ROOT :uniform prescribed value of ice soil water index (SWI) for the root zone soil layer(s). This prescribed value, if defined, has priority on the use of CFILE_HUG and CFILE_ISBA data.
- XHUGI_DEEP : uniform prescribed value of ice soil water index (SWI) for the deep soil layer(s). This prescribed value, if defined, has priority on the use of CFILE_HUG and CFILE_ISBA data.
- CFILE_HUG_SURF: name of the file used to define the liquid soil water index (SWI) for the surface soil layer.
- CFILE_HUG_ROOT: name of the file used to define the liquid soil water index (SWI) for the root zone soil layer(s).
- CFILE_HUG_DEEP: name of the file used to define the liquid soil water index (SWI) for the deep soil layer(s).
- CFILE_HUG: name of the file used to define the soil water profiles.
  The use of a file or prescribed value of XHUG_SURF, XHUG_ROOT and XHUG_DEEP has priority on the data in CFILE_HUG file.
- CTYPE_HUG: type of the CFILE_HUG file, if the latter is provided. CTYPE_HUG must then be given. The following values are currently usable:
    - "MESONH" : the file type is a MESONH file.
    - "GRIB " : the file type is a GRIB file, coming from any of these models:
        1. "ECMWF " : european center forecast model
        2. "ARPEGE" : Arpege french forecast model
        3. "ALADIN" : Aladin french forecast local model
        4. "MOCAGE" : Mocage french research chemistry model
    - "ASCII / LFI ": PREP file from Surfex

- ♦ "ASCLLV": ASCII latlonval file (one file for each depth)
- ♦ "NETCDF": netcdf standard file (one variable by depth)

- XTG_SURF : uniform prescribed value of temperature for the surface soil layer, supposed at an **altitude of 0m** (mean sea level altitude). The temperature is then modified for each point depending on its altitude, following a uniform vertical gradient of -6.5 $K\ km^{-1}$. This prescribed value, if defined, has priority on the use of CFILE_TG and CFILE_ISBA data.
- XTG_ROOT : uniform prescribed value of temperature for the root zone soil layer(s), supposed at an **altitude of 0m** (mean sea level altitude). The temperature is then modified for each point depending on its altitude, following a uniform vertical gradient of -6.5 $K\ km^{-1}$. This prescribed value, if defined, has priority on the use of CFILE_TG and CFILE_ISBA data.
- XTG_DEEP : uniform prescribed value of temperature for the deep soil layer(s), supposed at an **altitude of 0m** (mean sea level altitude). The temperature is then modified for each point depending on its altitude, following a uniform vertical gradient of -6.5 $K\ km^{-1}$. This prescribed value, if defined, has priority on the use of CFILE_TG and CFILE_ISBA data.
- CFILE_TG_SURF: name of the file used to define the surface soil temperature profile.
- CFILE_TG_ROOT: name of the file used to define the root zone soil temperature profile.
- CFILE_TG_DEEP: name of the file used to define the deep soil temperature profile.
- CFILE_TG: name of the file used to define the soil temperature profile.
  The use of a file or prescribed value of XTG_SURF, XTG_ROOT and XTG_DEEP has priority on the data in CFILE_TG file.
- CTYPE_TG: type of the CFILE_TG file, if the latter is provided. CTYPE_TG must then be given. The following values are currently usable:
  - ♦ "MESONH" : the file type is a MESONH file.
  - ♦ "GRIB " : the file type is a GRIB file, coming from any of these models:
    1. "ECMWF " : european center forecast model
    2. "ARPEGE" : Arpege french forecast model
    3. "ALADIN" : Aladin french forecast local model
    4. "MOCAGE" : Mocage french research chemistry model
  - ♦ "ASCII / LFI ": PREP file from Surfex
  - ♦ "ASCLLV": ASCII latlonval file (one file for each depth)
  - ♦ "NETCDF": netcdf standard file (one variable by depth)
- CFILE_ISBA / CFILEPGD_ISBA: name of the PREP / PGD files used to define any ISBA variable. **The use of a file or prescribed value XHUG_SURF, XHUG_ROOT, XHUG_DEEP, XTG_SURF, XTG_ROOT, XTG_DEEP, CFILE_WG and CFILE_TG has priority on the data in CFILE_ISBA file**.
- CTYPE / CTYPEPGD : type of the CFILE_ISBA / CFILEPGD_ISBA files, if the latter is provided. CTYPE / CTYPEPGD must then be given. The following values are currently usable:
  - ♦ "MESONH" : the file type is a MESONH file.
  - ♦ "GRIB " : the file type is a GRIB file, coming from any of these models:
    1. "ECMWF " : european center forecast model
    2. "ARPEGE" : Arpege french forecast model
    3. "ALADIN" : Aladin french forecast local model
    4. "MOCAGE" : Mocage french research chemistry model
  - ♦ "ASCII " : PREP/PGD Surfex ASCII file

    - "LFI ": PREP/PGD Surfex LFI file
- NYEAR : year of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).
- NMONTH : month of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).
- NDAY : day of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).
- XTIME : time from midnight of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read). (seconds).
- LISBA_CANOPY : activates surface boundary multi layer scheme over vegetation.

5.6.2. Namelist NAM_PREP_ISBA_SNOW

This namelist defines the type of snow scheme used in ISBA scheme.

Fortran name	Fortran type	values	default value
CSNOW	string of 3 characters	'D95', '3-L', 'EBA','CRO'	'D95'
NSNOW_LAYER	integer		1
CFILE_SNOW	string of 28 characters		CFILE_ISBA in NAM_PREP_ISBA
CTYPE_SNOW	string of 6 characters	'MESONH', 'GRIB ', 'LFI '	CTYPE in NAM_PREP_ISBA
LSNOW_IDEAL	logical		.FALSE.
LSNOW_FRAC_TOT	logical		.FALSE.
XWSNOW	real(20)		0.
XTSNOW	real(20)		273.16
XRSNOW	real(20)		300.
XASNOW	real		0.5
XSG1SNOW	real(20)		none
XSG2SNOW	real(20)		none
XHISTSNOW	real(20)		none
XAGESNOW	real(20)		none

- CSNOW : type of snow scheme. Possible snow schemes are:
    1. 'D95' : Douville et al (1995) snow scheme.
    2. '3-L' : Boone and Etchevers (2000) three layers snow scheme.
    3. 'EBA' : Bogatchev and Bazile (2005), Arpege operational snow scheme.
    4. 'CRO' : Crocus model
- NSNOW_LAYER : number of snow layers
- CFILE_SNOW : name of the file used to define the snow profiles. **The use of a file or prescribed value of XRSNOW, XTSNOW, XWSNOW and XASNOW (and XSG1SNOW, XSG2SNOW, XHISTSNOW and XAGESNOW in case of CSNOW = CROCUS) has priority on the data in CFILE_SNOW file**
- CTYPE_SNOW : type of the CFILE_SNOW file, if the latter is provided. CTYPE_SNOW must then be given. The following values are currently usable:
    - "MESONH" : the file type is a MESONH file.
    - "GRIB " : the file type is a GRIB file, coming from any of these models:
        1. "ECMWF " : european center forecast model
        2. "ARPEGE" : Arpege french forecast model
        3. "ALADIN" : Aladin french forecast local model
        4. "MOCAGE" : Mocage french research chemistry model
- LSNOW_IDEAL : if LSNOW_IDEAL = .FALSE. , only one value can be given for following snow parameters and a vertical interpolation is processed. If LSNOW_IDEAL = .TRUE., values are given for each layer and there is no vertical interpolation performed.

- LSNOW_FRAC_TOT : if LSNOW_FRAC_TOT = .TRUE., the total snow fraction *XPSN = MIN(1.0, ZSNOWSWE(:)/XWCRN_EXPL)* where ZSNOWSWE is the snow liquid water content, and XWCRN_EXPL is the critical value of the equivalent water content of the snow reservoir.
- XWSNOW : uniform value to initialize snow content, one for each layer
- XTSNOW : uniform value to initialize snow temperature, one for each layer
- XRSNOW : uniform value to initialize snow density, one for each layer
- XASNOW : uniform value to initialize snow albedo
- XSG1SNOW : uniform value to initialize snow layers grain feature 1 for Crocus, one for each layer
- XSG2SNOW : uniform value to initialize snow layers grain feature 2 for Crocus, one for each layer
- XHISTSNOW : uniform value to initialize snow layer grain historical parameter for Crocus, one for each layer
- XAGESNOW : uniform value to initialize snow grain age for Crocus, one for each layer

5.6.3. Namelist NAM_PREP_ISBA_CARBON

Fortran name	Fortran type	values	default value
CRESPSL	string of 3 characters	'DEF', 'PRM', 'CNT'	'DEF'

- CRESPSL : soil respiration option. Possible values are:
    1. 'DEF' : Norman 1992
    2. 'PRM' : Rivalland 2003
    3. 'CNT' : Century model 2007

5.7. Town scheme "TEB "

5.7.1. Namelist NAM_PREP_TEB

This namelist information is used to initialize the "TEB " urban scheme variables: road, roof and wall temperature profiles, water intercepted by roofs and roads, snow, building internal temperature.

Fortran name	Fortran type	values	default value
XWS_ROAD	real		none
XWS_ROOF	real		none
CFILE_WS	string of 28 characters		CFILE_TEB in this namelist
CTYPE_WS	string of 6 characters	'MESONH', 'GRIB ','LFI '	CTYPE in this namelist
XTS_ROAD	real		none
XTS_ROOF	real		none
XTS_WALL	real		none
XTI_BLD	real		none
XTI_ROAD	real		none
XHUI_BLD	real		none
CROAD_DIR	character(LEN=4)	'UNIF','ORIE'	'UNIF'
CWALL_OPT	character(LEN=4)	'UNIF','TWO '	'UNIF'
CFILE_TS	string of 28 characters		CFILE_TEB in this namelist
CTYPE_TS	string of 6 characters	'MESONH', 'GRIB ', 'LFI '	CTYPE in this namelist
CFILE_TEB	string of 28 characters		CFILE in NAM_PREP_SURF_ATM
CTYPE	string of 6 characters	'MESONH', 'GRIB ', 'ASCII ','LFI '	CFILETYPE in NAM_PREP_SURF_ATM
CFILEPGD_TEB	string of 28 characters		CFILEPGD in NAM_PREP_SURF_ATM
CTYPEPGD	string of 6 characters	'MESONH', 'GRIB ', 'ASCII ','LFI '	CFILEPGDTYPE in NAM_PREP_SURF_ATM
NYEAR	integer		none
NMONTH	integer		none
NDAY	integer		none
XTIME	real		none
LTEB_CANOPY	logical		F

- XWS_ROAD : uniform prescribed value of soil water interception for the road reservoir. This prescribed value, if defined, has priority on the use of CFILE_WS and CFILE_TEB data.
- XWS_ROOF : uniform prescribed value of soil water interception for the roof reservoir. This prescribed value, if defined, has priority on the use of CFILE_WS and CFILE_TEB data.
- CFILE_WS: name of the file used to define the soil water reservoirs. **The use of a file or prescribed value of XWS_ROAD and XWS_ROOF has priority on the data in CFILE_WS file**.

- CTYPE_WS: type of the CFILE_WS file, if the latter is provided. CTYPE_WS must then be given. The following values are currently usable:
    - ♦ "MESONH" : the file type is a MESONH file.
    - ♦ "GRIB " : the file type is a GRIB file, coming from any of these models:
        1. "ECMWF " : european center forecast model
        2. "ARPEGE" : Arpege french forecast model
        3. "ALADIN" : Aladin french forecast local model
        4. "MOCAGE" : Mocage french research chemistry model
- XTS_ROAD : uniform prescribed value of temperature for road, supposed at an **altitude of 0m** (mean sea level altitude). The temperature is then modified for each point depending on its altitude, following a uniform vertical gradient of -6.5 $K\ km^{-1}$. This prescribed value, if defined, has priority on the use of CFILE_TS and CFILE_TEB data.
- XTS_ROOF : uniform prescribed value of temperature for roof, supposed at an **altitude of 0m** (mean sea level altitude). The temperature is then modified for each point depending on its altitude, following a uniform vertical gradient of -6.5 $K\ km^{-1}$. This prescribed value, if defined, has priority on the use of CFILE_TS and CFILE_TEB data.
- XTS_WALL : uniform prescribed value of temperature for wall, supposed at an **altitude of 0m** (mean sea level altitude). The temperature is then modified for each point depending on its altitude, following a uniform vertical gradient of -6.5 $K\ km^{-1}$. This prescribed value, if defined, has priority on the use of CFILE_TS and CFILE_TEB data.
- XTI_BLD : uniform prescribed value of internal building temperature. This temperature is not dependent on altitude. This prescribed value, if defined, has priority on the use of CFILE_TS and CFILE_TEB data.
- XTI_ROAD : uniform prescribed value of deep road temperature, supposed at an **altitude of 0m** (mean sea level altitude). The temperature is then modified for each point depending on its altitude, following a uniform vertical gradient of -6.5 $K\ km^{-1}$. This prescribed value, if defined, has priority on the use of CFILE_TS and CFILE_TEB data.
- XHUI_BLD : uniform bulding relative hum (between 0-1)
- CROAD_DIR : TEB option for road direction:
    - ♦ UNIF : no specific direction
    - ♦ ORIE : many road ORIEntations (linked to NTEB_PATCH)
- CWALL_OPT : TEB option for walls:
    - ♦ UNIF : uniform walls
    - ♦ TWO : two separated walls
- CFILE_TS: name of the file used to define the soil temperature profile. **The use of a file or prescribed value of XTS_ROAD, XTS_ROOF, XTS_WALL, XTI_BLD or XTI_ROAD has priority on the data in CFILE_TS file**.
- CTYPE_TS: type of the CFILE_TS file, if the latter is provided. CTYPE_TS must then be given. The following values are currently usable:
    - ♦ "MESONH" : the file type is a MESONH file.
    - ♦ "GRIB " : the file type is a GRIB file, coming from any of these models:
        1. "ECMWF " : european center forecast model
        2. "ARPEGE" : Arpege french forecast model
        3. "ALADIN" : Aladin french forecast local model
        4. "MOCAGE" : Mocage french research chemistry model
- CFILE_TEB / CFILEPGD_TEB: name of the PREP/PGD files used to define any TEB

variable. **The use of a file or prescribed value XWS_ROAD, XWS_ROOF, XTS_ROAD, XTS_ROOF, XTS_WALL, XTI_BLD, XTI_ROAD, CFILE_WS or CFILE_TS has priority on the data in CFILE_TEB file**.

- CTYPE / CTYPEPGD: type of the CFILE_TEB / CFILEPGD_TEB file, if the latter is provided. CTYPE must then be given. The following values are currently usable:
    - "MESONH" : the file type is a MESONH file.
    - "GRIB " : the file type is a GRIB file, coming from any of these models:
        1. "ECMWF " : european center forecast model
        2. "ARPEGE" : Arpege french forecast model
        3. "ALADIN" : Aladin french forecast local model
        4. "MOCAGE" : Mocage french research chemistry model
    - "ASCII ": PREP/PGD Surfex ASCII file
    - "LFI ": PREP/PGD Surfex LFI file
- NYEAR : year of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).
- NMONTH : month of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).
- NDAY : day of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read).
- XTIME : time from midnight of surface UTC time. It is used only if no atmospheric file or no surface file is given (in those the date can be read). (seconds).
- LTEB_CANOPY : activates surface boundary multi layer scheme over town.

5.7.2. Namelist NAM_PREP_TEB_SNOW

Fortran name	Fortran type	values	default value
CSNOW_ROOF	string of 6 characters	'1-L'	'1-L'
CSNOW_ROAD	string of 6 characters	'1-L'	'1-L'
XWSNOW_ROOF	real		none
XWSNOW_ROAD	real		none
XTSNOW_ROOF	real		none
XTSNOW_ROAD	real		none
XASNOW_ROOF	real		none
XASNOW_ROAD	real		none
XRSNOW_ROOF	real		none
XRSNOW_ROAD	real		none

- CSNOW_ROAD : snow scheme used over roads
- CSNOW_ROOF : snow scheme used over roofs
- XWSNOW_ROOF : roof snow content
- XWSNOW_ROAD : road snow content
- XTSNOW_ROAD : road temperature
- XTSNOW_ROOF : roof temperature
- XRSNOW_ROOF : roof density
- XRSNOW_ROAD : road density
- XASNOW_ROAD : road albedo
- XASNOW_ROOF : road albedo

5.7.3. Namelist NAM_PREP_TEB_GARDEN

duplication of NAM_PREP_ISBA to initialize vegetation in urban areas.

5.7.4. Namelist NAM_PREP_GARDEN_SNOW

duplication of NAM_PREP_ISBA_SNOW to initialize vegetation in urban areas (without specific CROCUS options).

5.7.5. Namelist NAM_PREP_GREENROOF

duplication of NAM_PREP_ISBA to initialize vegetation on greenroofs in urban areas.

5.7.6. Namelist NAM_PREP_GREENROOF_SNOW

duplication of NAM_PREP_ISBA_SNOW to initialize vegetation on greenroofs in urban areas (without specific CROCUS options).

6. How to run the externalized surface physical schemes

Here are described the options available during the run of the several surface schemes.

6.1. "SURF_ATM" general options available over all tiles

6.1.1. Namelist NAM_SURF_CSTS

Fortran name	Fortran type	values	default value
XEMISSN	real		1.0
XANSMIN	real		0.5
XANSMAX	real		0.85
XAGLAMIN	real		0.8
XAGLAMAX	real		0.85
XALBWAT	real		0.135
XALBCOEF_TA96	real		0.037
XALBSCA_WAT	real		0.06
XEMISWAT	real		0.98
XALBWATICE	real		0.85
XEMISWATICE	real		1.0
XHGLA	real		33.3
XWSNV	real		5.0
XCFFV	real		3.0
XZ0SN	real		0.001
XZ0HSN	real		0.0001

- XEMISSN : snow emissivity
- XANSMIN : minimum value for snow albedo
- XANSMAX : maximum value for snow albedo
- XAGLAMIN : minimum value for permanent snow/ice albedo
- XAGLAMAX : maximum value for permanent snow/ice albedo
- XALBCOEF_TA96 : coefficient used in th computation of albedo if 'TA96' option selected
- XALBSCA_WAT : water diffuse albedo
- XEMISWAT: water emissivity
- XALBWATICE : sea ice albedo
- XEMISWATICE : sea ice emissivity
- XHGLA : Height of aged snow in glacier case (allows Pn=1)
- XWSNV : Coefficient for calculation of snow fraction over vegetation
- XCFFV : Coefficient for calculation of floodplain fraction over vegetation
- XZ0SN : roughness length of pure snow surface (m)
- XZ0HSN:roughness length for heat of pure snow surface (m)

6.1.2. Namelist NAM_SURF_ATM

Fortran name	Fortran type	values	default value
XCISMIN	real		6.7E-5
XVMODMIN	real		0.
LALDTHRES	logical		F
LDRAG_COEF_ARP	logical		F
LALDZ0H	logical		F
LNOSOF	logical		F
LRW_PRECIP	logical		F
XEDB	real		5.
XEDC	real		5.
XEDD	real		5.
XEDK	real		1.
XUSURIC	real		1.
XUSURID	real		0.035
XUSURICL	real		4.
XVCHRNK	real		0.015
XVZ0CM	real		0.0
XRIMAX	real		0.0
XDELTA_MAX	real		1.0
LVZIUSTAR0_ARP	logical		F
LRRGUST_ARP	logical		F
XVZIUSTAR0	real		0.
XRZHZ0M	real		1.
XRRSCALE	real		1.15E-4
XRRGAMMA	real		0.8
XUTILGUST	real		0.125
LCPL_ARP	logical		F
LQVNPLUS	logical		F
LVERTSHIFT	logical		T
CIMPLICIT_WIND	character(LEN=3)	'NEW','OLD'	'NEW'

- LALDTHRES: flag to set a minimum wind and shear like done in Aladin model.
- XCISMIN: minimum wind shear to compute turbulent exchange coefficient (used only if LALDTHRES)
- XVMODMIN: minimum wind speed to compute turbulent exchange coefficient (used only if LALDTHRES)

- LDRAG_COEF_ARP: to use drag coefficient computed like in Arpege/Aladin models
- LALDZ0H: to take into account orography in heat roughness length
- LNOSOF: no parameterization of subgrid orography effects on atmospheric forcing
- LRW_PRECIP: flag used to red/write precipitation forcing from/into the erstart file for ARPEGE/ALADIN run
- XEDB, XEDC, XEDD, XEDK: coefficients used in Richardson critical numbers computation
- XUSURIC, XUSURID, XUSURICL: Richardson critical numbers
- XVCHRNK, XVZ0CM: Charnock's constant and minimal neutral roughness length over sea (formulation of roughness length over sea)
- XRIMAX: limitation of Richardson number in drag computation
- XDELTA_MAX: maximum fraction of the foliage covered by intercepted water for high vegetation
- LVZIUSTAR0_ARP: flag to activate aladin formulation for zoh over sea
- LRRGUST_ARP: flag to activate the correction of CD, CH, CDN due to moist gustiness
- XVZIUSTAR0: aladin formulation for zoh over sea
- XRZHZ0M: aladin formulation for zoh over sea
- XRRSCALE: aladin formulation for zoh over sea
- XRRGAMMA: aladin formulation for zoh over sea
- XUTILGUST: correction of CD, CH, CDN due to moist gustiness
- LCPL_ARP: activate aladin formulation for Cp and L
- LQVNPLUS: An option for the resolution of the surface temperature equation (Arpege)
- LVERTSHIFT : vertical shifth from atmospheric orography to surface orography
- CIMPLICIT_WIND : wind implicitation option:
    - OLD: direct
    - NEW: Tayler serie, order 1

6.1.3. Namelist NAM_WRITE_SURF_ATM

Fortran name	Fortran type	values	default value
LNOWRITE_CANOPY	logical		F
LNOWRITE_TEXFILE	logical		F

- LNOWRITE_CANOPY: if T, do not write canopy prognostic variables in initial/restart or LBC files
- LNOWRITE_TEXFILE: if T, do not fill class_cover_data.tex file during the model setup

6.1.4. Namelist NAM_SSOn

The namelist NAM_SSOn concerns the roughness parameterization for orography.

Fortran name	Fortran type	values	default value
CROUGH	string of 4 characters	'Z01D', 'Z04D', 'NONE', 'BE04'	BE04
XFRACZ0	real		2.
XCOEFBE	real		2.

- CROUGH: type of orographic roughness length. The following options are currently available:
    - "Z01D" : orographic roughness length does not depend on wind direction
    - "Z04D" : orographic roughness length depends on wind direction
    - "BE04" : Beljaars 2004 orographic drag
    - "NONE" : no orographic treatment
- XFRACZ0 : Z0=Min(Z0, Href/XFRACZ0)
- XCOEFBE : coefficient for Beljaars calculation of SSO drag.

6.2. "SEAFLX" sea scheme options

6.2.1. Namelist NAM_SEAFLUXn

Fortran name	Fortran type	values	default value
CSEA_FLUX	string of 6 characters	'DIRECT', 'ITERAT', 'COARE3', 'ECUME '	'ECUME '
CSEA_ALB	string of 4 characters	'UNIF', 'TA96','MK10'	'TA96'
LPWG	logical		F
LPRECIP	logical		F
LPWEBB	logical		F
LPROGSST	logical		F
NTIME_COUPLING	integer		
CINTERPOL_SST	string of 6 characters	'ANNUAL', 'MONTH ', 'NONE '	'NONE '
XICHCE	real		0.

- CSEA_FLUX: type of flux computation physics. The following option is currently available:
  - "DIRECT" : direct Charnock computation. No effect of convection in the the boundary layer on the fluxes formulae.
  - "ITERAT" : iterative method proposed by Fairall et al (1996) from TOGA-COARE experiment, amended by Mondon and Redelsperger (1998) to take into account effect of atmospheric convection on fluxes.
  - "COARE3" : iterative method proposed by Fairall et al (1996) from TOGA-COARE experiment, amended by cnrm/memo to take into account effect of atmospheric convection, precipitation and gustiness on fluxes.
  - "ECUME " : iterative method proposed by Fairall et al (1996) from TOGA-COARE experiment, amended by cnrm/memo to take into account effect of atmospheric convection, precipitation and gustiness on fluxes: improvement of surface exchange coefficients representation.
- LPWG: correction of fluxes due to gustiness
- LPRECIP: correction of fluxes due to precipitation
- LPWEBB:correction of fluxes due to convection (Webb effect)
- CSEA_ALB: type of albedo formula. The following options are currently available:
  - "UNIF" : a uniform value of 0.135 is used for water albedo
  - "TA96" : Taylor et al (1996) formula for water direct albedo, depending on solar zenith angle θ:
    $$\alpha_{dir} = 0.037/\left(1.1\cos(\theta)^{1.4} + 0.15\right)$$
  - "MK10": albedo from Marat Khairoutdinov
- LPROGSST: set it to .TRUE. to make SST evolve with tendency when using the 1d oceanic model
- NTIME_COUPLING: coupling time frequency between surface and the 1d oceanic model
- CINTERPOL_SST: interpolate monthly or annual SST to daily SST (function interpol_quadra)
- XICHCE: coefficient used in the Ecume formulation (computation of exchange coefficients over sea)

6.2.2. Namelist NAM_SURF_SLT

Fortran name	Fortran type	values	default value
CEMISPARAM_SLT	string of 5 characters	'Vig01','Sch04'	'Sch04"

- "CEMISPARAM_SLT" : One-line sea salt emission parameterization type. This namelist gives the distribution of emitted sea salt of SURFEX. For Each paramterization type, a geometric standard deviation and a median radius is given. See the code init_sltn.f90 (MesoNH) or init_sltn.mnh (AROME, ALADIN) for values associated to these parameterizations. Note that if the defaut value is change, it is necessary to uses the same modes in the sea initialisation in the atmospheric model. It concerns the value of XINIRADIUS_SLT (initial radius), XINISIG_SLT (standard deviation) and CRGUNITS (mean radius definition) to have the same aerosol size distribution emitted and in the atmosphere. It is possible to do it directly in the fortran code (modd_salt.mnh in case of aladin/arome, modd_salt.f90 for MesoNH) or for MesoNH only, change the values of these variables in NAM_AERO_CONF (prep_real_case or prep_ideal_case).

6.3. "WATFLX" inland water scheme options

6.3.1. Namelist NAM_WATFLUXn

Fortran name	Fortran type	values	default value
CINTERPOL_TS	string of 6 characters	'ANNUAL', 'MONTH ', 'NONE '	'NONE '
CWAT_ALB	string of 4 characters	'UNIF', 'TA96'	'UNIF'

- CWAT_ALB: type of formulation used to set albedo over water
- CINTERPOL_TS: interpolate monthly or annul TS to daily TS (function interpol_quadra)

6.4. "FLAKE" lake scheme options

6.4.1. Namelist NAM_FLAKEn

Fortran name	Fortran type	values	default value
LSEDIMENTS	logical		T
CSNOW_FLK	string of 3 characters	'DEF '	'DEF'
CFLK_FLUX	string of 3 characters	'FLAKE', 'DEF ','ECUME'	'DEF '
CFLK_ALB	string of 4 characters	'UNIF', 'TA96 '	'UNIF'
XICHCE	real		0
LPRECIP	logical		F
LPWEBB	logical		F

- LSEDIMENTS: to use the bottom sediments scheme of Flake (default)
- CSNOW_FLK: snow scheme to be used. For the time being only option 'DEF' is active
- CFLK_FLUX: scheme to be used to compute surface fluxes of moment, energy and water vapor:
    - "DEF " to activate the classic watflux
    - "FLAKE " to use flake parameterization
    - "ECUME" to use the ECUME parameterization
- CFLK_ALB:type of albedo for Flake.
- XICHCE : CE coefficient if CFLK_FLUX=ECUME
- LPRECIP : flag for precipitation correction if CFLK_FLUX=ECUME
- LPWEBB : flag for heat flux correction if CFLK_FLUX=ECUME

6.5. "ISBA " vegetation scheme options

6.5.1. Namelist NAM_SGH_ISBAn

Fortran name	Fortran type	values	default value
CRUNOFF	string of 4 characters	'WSAT', 'DT92' , 'SGH', 'TOPD'	'WSAT'
CTOPREG	string of 4 characters	'DEF ', 'NON'	'DEF '
CKSAT	string of 4 characters	'DEF', 'SGH','EXP'	'DEF'
CSOC	string of 3 characters	'DEF', 'SGH'	'DEF'
CRAIN	string of 3 characters	'DEF', 'SGH'	'DEF'
CHORT	string of 4 characters	'DEF ', 'SGH'	'DEF '
LTRIP	logical		F
LFLOOD	logical		F

- CRUNOFF: type of subgrid runoff. The following options are currently available:
    - "WSAT" : runoff occurs only when saturation is reached
    - "DT92" : Dumenill and Todini (1992) subgrid runoff formula
    - "SGH" : Decharme et al. (2006) Topmodel like subgrid runoff
    - "TOPD" : if LCOUPL_TOPD=T, allows that DUNNE runoff contains only saturated pixels on meshes so only catchments
- CTOPREG: kind of regression. Option activated only if CRUNOFF = 'SGH'. The following options are currently available:
    - "DEF" : Wolock and MacCabe regression between topographic indices computed at 1km and 100m resolution (recommended)
    - "NON" : no regression
- CKSAT: Activates the exponential profile for Ksat. The following options are currently available:
    - "DEF" : homogeneous profile
    - "SGH" : exponential decreasing profile with depth (due to compaction of soil)
    - "EXP" : with CISBA="3-L" and LCOUPL_TOPD=T, allows to read a file containing values for the F parameter, computed by topmodel during PGD.
- CSOC: soil organic carbon effect:
    - "DEF": default value
    - "SGH" soil SOC profile
- CRAIN: Activates the spatial distribution of rainfall intensity. The following options are currently available:
    - "DEF" : homogeneous distribution
    - "SGH" : exponential distribution which depends on the fraction of the mesh where it rains. This fraction depends on the mesh resolution and the intensity of hourly precipitation. (If the horizontal mesh is lower than 10km then the fraction equals 1).
- CHORT: Activates the Horton runoff due to water infiltration excess. The following options are currently available:
    - "DEF" : no Horton runoff
    - "SGH" : Horton runoff computed

- LTRIP: Activates TRIP river routing model (RRM) scheme
- LFLOOD: Activates the flooding scheme

6.5.2. Namelist NAM_ISBAn

Fortran name	Fortran type	values	default value
CC1DRY	string of 4 characters	'DEF ', 'GB93'	'DEF '
CSCOND	string of 4 characters	'NP89', 'PL98'	'PL98'
CSOILFRZ	string of 3 characters	'DEF', 'LWT'	'DEF'
CDIFSFCOND	string of 4 characters	'DEF ', 'MLCH'	'DEF '
CSNOWRES	string of 3 characters	'DEF', 'RIL'	'DEF'
CALBEDO	string of 4 characters	'MEAN', 'DRY ', 'WET ', 'EVOL','CM13'	'DRY '
CROUGH	string of 4 characters	'Z01D', 'Z04D', 'NONE', 'BE04', 'UNDE'	'UNDE'
CCPSURF	string of 3 characters	'DRY ', 'HUM'	'DRY '
LCANOPY_DRAG	logical		.FALSE.
LGLACIER	logical		.FALSE.
XTSTEP	real		none
XCGMAX	real		2.E-5
XCDRAG	real		0.15
LVEGUPD	logical		T

- CC1DRY: type of C1 formulation for dry soils. The following options are currently available:
    - "DEF " : Giard-Bazile formulation
    - "GB93" : Giordani 1993, Braud 1993
- CSCOND: type of thermal conductivity. The following options are currently available:
    - "NP89" : Noilhan and Planton (1989) formula
    - "PL98" : Peters-Lidard et al. (1998) formula
- CSOILFRZ: type of soil freezing-physics option. The following options are currently available:
    - "DEF" : Boone et al. 2000; Giard and Bazile 2000
    - "LWT" : Phase changes as above, but relation between unfrozen water and temperature considered
- CDIFSFCOND: type of Mulch effects. The following options are currently available:
    - "DEF " : no mulch effect
    - "MLCH" : include the insulating effect of leaf litter/mulch on the surf. thermal cond.
- CSNOWRES: type of turbulent exchanges over snow. The following options are currently available:
    - "DEF" : Louis
    - "RIL" : Maximum Richardson number limit for stable conditions ISBA-SNOW3L turbulent exchange option
- CALBEDO: type of bare soil albedo. The following options are currently available:
    - "DRY " : dry bare soil albedo
    - "WET " : wet bare soil albedo

- ♦ "MEAN" : albedo for bare soil half wet, half dry
- ♦ "EVOL" : albedo of bare soil evolving with soil humidity
- ♦ "CM13" : albedo by cover and vegetation type processed from satellite data

- CROUGH: type of orographic roughness length. The following options are currently available:
  - ♦ "Z01D" : orographic roughness length does not depend on wind direction
  - ♦ "Z04D" : orographic roughness length depends on wind direction
  - ♦ "BE04" : Beljaars 2004 orographic drag
  - ♦ "NONE" : no orographic treatment

    By default, CROUGH is set to "UNDE" (undefined) since its initialization depends if "LISBA_CANOPY" is activated or not, finally default value of CROUGH is updated as follows:
  - ♦ "CROUGH = Z04D" if "LISBA_CANOPY" = F
  - ♦ "CROUGH = BE04" if "LISBA_CANOPY" = T
- CCPSURF: type of specific heat at surface. The following options are currently available:
  - ♦ "DRY" : specific heat does not depend on humidity at surface
  - ♦ "HUM" : specific heat depends on humidity at surface.
- LCANOPY_DRAG: drag activated in SBL scheme within the canopy
- LGLACIER: If activated, specific treatment (as in Arpege) over permanent snow/ice regions. Snow depth initialised to 10m and soil ice to porosity. During the run, snow albedo ranges from 0.8 to 0.85
- XTSTEP: time step for ISBA. Default is to use the time-step given by the atmospheric coupling (seconds).
- XCGMAX: maximum value for soil heat capacity.
- XCDRAG: drag coefficient in canopy.
- LVEGUPD: True = update vegetation parameters every decade

6.5.3. Namelist NAM_SURF_DST

Fortran name	Fortran type	values	default value
CEMISPARAM_DST	string of 5 characters	'AMMA','Dal87','EXPLI','alf98','EXPLI'	'AMMA'
CVERMOD	string of 6 characters	'CMDVER'	'NONE'
XFLX_MSS_FDG_FCT	real		8.0e-4

- "CEMISPARAM_DST" : One-line dust emission parameterization type. This namelist gives the distribution of emitted dust of SURFEX. For Each paramterization type, a geometric standard deviation and a median radius is given. Moreover , the repatition of mass flux could be derive from the friction velocity (case of "AMMA" or "EXPLI") or imposed (case of "Dal87", "alf98", "She84" or "PaG77". See the code init_dstn.f90 (MesoNH) or init_dstn.mnh (AROME, ALADIN) for values associated to these parameterizations. Note that if the defaut value is change, it is necessary to uses the same modes in the dust initialisation in the atmospheric model. It concerns the value of XINIRADIUS (initial radius), XINISIG (standard deviation) and CRGUNITD (mean radius definition) to have the same aerosol size distribution emitted and in the atmosphere. It is possible to do it directly in the fortran code (modd_dust.mnh in case of aladin/arome, modd_dust.f90 for MesoNH) or for MesoNH only, change the values of these variables in NAM_AERO_CONF (prep_real_case or prep_ideal_case).
- "XFLX_MSS_FDG_FCT" : Value of the Î± factor representing the ratio of the vertical mass flux over the horizontal mass flux in the saltation layer (use only If CVERMOD="NONE"). This Î± factor depend on the size distribution of the aerosol consider in the model.
- "CVERMOD" New parameterization of the dust emission formulation. In development, not recommended to uses it in this version.

6.5.4. Namelist NAM_ASSIM

Declaration of keys for ISBA assimilation scheme (2DVAR, Bouyssel et al.).

Fortran name	Fortran type	values	default value
LASSIM	logical		F
CASSIM	character(LEN=5)	"PLUS", "AVERA", "2DVAR"	"PLUS "

- LASSIM: Assimilation or not
- CASSIM: type of correction

6.5.5. Namelist NAM_AGRI

Agricultural Practices

Fortran name	Fortran type	values	default value
LAGRIP	logical		F

- LAGRIP : General switch for agricultural practices (seeding and irrigation)

6.5.6. Namelist NAM_DEEPSOIL

deep soil characteristics

Fortran name	Fortran type	values	default value
LDEEPSOIL	logical		F
LPHYSDOMC	logical		F

- LDEEPSOIL:General switch for deep soil fields (temperature and relaxation time). XTDEEP_CLI =(/236.,236.,220.,209.,206.,211.,214.,210.,207.,212.,220.,229./) XGAMMAT_CLI=(/4.,4.,4.,3.,1.,2.,3.,1.,1.,1.,1.,2./)
- LPHYSDOMC : General switch to impose CT and soil water/ice contents CT(:) = 9.427757E-6

6.5.7. Namelist NAM_TREEDRAG

Declaration to take into account tree drag in the atmospheric model instead of SURFEX. The Z0 forest is therefore reduced to the Z0 grass.

Fortran name	Fortran type	values	default value
LTREEDRAG	logical		F

- LTREEDRAG:flag used to take into account tree drag in the atmospheric model instead of SURFEX.

6.5.8. Namelist NAM_SPINUP_CARBn

Soil and wood carbon spinup

Fortran name	Fortran type	values	default value
LSPINUPCARBS	logical		F
LSPINUPCARBW	logical		F
XSPINMAXS	real		0.
XSPINMAXW	real		0.
NNBYEARSPINS	integer		0
NNBYEARSPINW	integer		0

- LSPINUPCARBS : true to do the soil carbon spinup
- LSPINUPCARBW : true to do the wood carbon spinup
- XSPINMAXS : maximum number of times CARBON_SOIL subroutine is called for each timestep in simulation during acceleration procedure
- XSPINMAXW : maximum number of times the wood is accelerated
- NNBYEARSPINS : number of years needed to reach soil equilibrium
- NNBYEARSPINW : number of years needed to reach wood equilibrium

6.5.9. Namelist NAM_CROCUSn

CROCUS snow scheme.

Fortran name	Fortran type	values	default value
LSNOWDRIFT	logical		T
LSNOWDRIFT_SUBLIM	logical		F
XZ0ICEZ0SNOW	real		10.
XRHOTHRESHOLD_ICE	real		850.
XALBICE1	real		0.38
XALBICE2	real		0.23
XALBICE3	real		0.08
XVAGING_NOGLACIER	real		60.
XVAGING_GLACIER	real		900.

- LSNOWDRIFT : logical for snowdrift
- LSNOWDRIFT_SUBLIM : logical for snowdrift subliation
- XZ0ICEZ0SNOW : roughness length ratio between ice and snow
- XRHOTHRESHOLD_ICE : density threshold for ice detection in CROCUS scheme
- XALBICE1, XALBICE2, XALBICE3 : prescribed ice albedo in 3 spectral bands for glacier simulation with CROCUS scheme
- XVAGING_NOGLACIER, XVAGING_GLACIER : for ageing effects

6.6. "TEB" town scheme options

6.6.1. Namelist NAM_TEBn

Fortran name	Fortran type	values	default value
CZ0H	character(LEN=6)	'MASC95','BRUT82','KAND07'	'MASC95'
CCH_BEM	character(LEN=5)	'','DOE-2'	''

- CZ0H: TEB option for z0h roof & road:
    - 'MASC95' : Mascart et al 1995
    - 'BRUT82' : Brustaert 1982
    - 'KAND07' : Kanda 2007
- CCH_BEM : BEM option for roof / wall outside convective coefficient :
    - 'DOE-2': DOE-2 model from EnergyPlus Engineering reference, p65

6.7. "IDEAL" ideal flux scheme options

6.6.1. Namelist NAM_IDEAL_FLUX

Fortran name	Fortran type	values	default value
NFORCF	integer <48		2
NFORCT	integer <48		2
XTIMEF	real(NFORCF)		0
XTIMET	real(NFORCT)		0
XSFTH	real(NFORCF)		0.
CSFTQ	character(LEN=7)	'kg/m2/s', 'W/m2 '	'kg/m2/s'
XSFTQ	real(NFORF)		0.
XSFCO2	real(NFORF)		
CUSTARTYPE	character(LEN=5)	'Z0 ', 'USTAR'	'Z0 '
XUSTAR	real(NFORCF)		0.
XZ0	real		0.01
XALB	real		
XEMIS	real		1.
XTSRAD	real(NFORT)		XTT=273.15K

- NFORCF : number of surface forcing instants for fluxes. The default value is NFORC=2.
- NFORCT : number of surface forcing instants for radiative temperature. The default value is NFORC=2.
- XTIMEF : times of forcing for fluxes (from beginning of run)
- XTIMET : times of forcing for temperature (from beginning of run)
- XSFTH: hourly data of heat surface flux (W/m2)
- CSFTQ: Unit for the evaporation flux (kg/m2/s) or (W/m2)
- XSFTQ: hourly data of water vapor surface flux
- XSFCO2: hourly data of CO2 surface flux (kg/m2/s)
- CUSTARTYPE: type of computation for friction
- XUSTAR: hourly data of friction (m2/s2)
- XZ0: roughness length (m)
- XALB: albedo (-)
- XEMIS:emissivity (-)
- XTSRAD: radiative temperature (K)

6.8. Coupling with TRIP model

6.8.1. NAM_GRID_TRIP

Fortran name	Fortran type	values	default value
TLONMIN	real		none
TLONMAX	real		none
TLATMIN	real		none
TLATMAX	real		none
TRES	real	1.,0.5	none

- TLONMIN : minimum longitude (degrees)
- TLONMAX : maximum longitude (degrees)
- TLATMIN : minimum latitude (degrees)
- TLATMAX : maximum latitude (degrees)
- TRES : resolution

6.8.2. NAM_TRIPn

Fortran name	Fortran type	values	default value
CVIT	character(LEN=3)	'DEF','VAR'	'DEF'
CGROUNDW	character(LEN=3)	'DEF','CST','VAR'	'DEF'
LFLOODT	logical		F
LTRIP_DIAG_MISC	logical		F
LDIAG_CPL	logical		F
LNCPRINT	logical		F
LPRINT_TRIP	logical		F
XTSTEP_COUPLING	real		86400.
XTRIP_TSTEP	real		3600.
XDATA_TAUG	real		30.
XRATMED	real		1.4
XCVEL	real		0.5

- CVIT : type of stream flow velocity:
    - 'DEF': constant velocity = 0.5m/s
    - 'VAR' : variable velocity
- CGROUNDW :use groundwater scheme
    - 'DEF' : no groundwater scheme
    - 'CST' : constant transfert time
    - 'VAR' : textural dependence of transfert time
- LFLOODT : to use TRIP-FLOOD
- LTRIP_DIAG_MISC : more diagnostics for model testing
- LDIAG_CPL : daily output diagnostics
- LNCPRINT : netcdf read/write messages
- LPRINT_TRIP : prints water budget messages
- XTSTEP_COUPLING : timestep for the coupling
- XTRIP_TSTEP : timestep for trip
- XDATA_TAUG : constant transfert time value
- XRATMED : meandering ratio
- XCVEL : constant velocity value

6.9. Coupling with TOPMODEL

6.9.1. NAM_PGD_TOPD

Fortran name	Fortran type	values	default value
CCAT	character(LEN=15), dimension(10)		''
LCOUPL_TOPD	logical		F
XF_PARAM_BV	real		2.5
XC_DEPTH_RATIO_BV	real		1.

- CCAT : base name for topographic files
- LCOUPL_TOPD : to perform the coupling with TOPMODEL
- XF_PARAM_BV : F parameter for the exponential profile, values for each catchment
- XC_DEPTH_RATIO_BV : depth ratio for the exponential profile, values for each catchment

6.9.2. NAM_TOPD

Fortran name	Fortran type	values	default value
LBUDGET_TOPD	logical		F
LSTOCK_TOPD	logical		F
NNB_TOPD	integer		1
NFREQ_MAPS_WG	integer		0
NFREQ_MAPS_ASAT	integer		0
NNB_STP_STOCK	integer		1
NNB_STEP_RESTART	integer		1
XSPEEDR	real		3.
XSPEEDG	real		0.3
XQINIT	real		0.
XRTOP_D2	real		1.

- LBUDGET_TOPD : to compute budget
- LSTOCK_TOPD : to stock runoff and drainage values (for another simulation)
- NNB_TOPD : ratio beween time steps of Topmodel and ISBA
- NFREQ_MAPS_WG : frequency of ouput WG maps
- NFREQ_MAPS_ASAT : frequency of output ASAT maps
- NNB_STP_STOCK : number of time steps to write for the next simulation
- NNB_STP_RESTART : number of time steps to restart from a previous simulation
- XSPEEDR : river speed
- XSPEEDG : ground speed
- XQINIT : initial discharge at the outlet of the catchments
- XRTOP_D2 : depth used by topodyn for lateral transfers (expressed in ratio of isba d2)

7. How to run the externalized surface chemical schemes

Here are described the options available during the run of the several schemes for emission and deposition of chemical species. Note that all the schemes for deposition and emission of chemical species do activate only if chemical species are present (i.e. if the coupling between atmosphere and surface include the chemical species concentrations and fluxes).

7.1. Chemical settings control

7.1.1. Namelist NAM_CH_CONTROLn

Fortran name	Fortran type	values	default value
CCHEM_SURF_FILE	string of 28 characters		' '

- CCHEM_SURF_FILE: name of general (chemical) purpose ASCII input file.

7.2. Chemical anthropogenic emissions

7.2.1. Namelist NAM_CH_SURFn

Fortran name	Fortran type	values	default value
LCH_SURF_EMIS	logical		.FALSE.

- LCH_SURF_EMIS: flag to use anthropogenic emissions or not.

7.3. Chemical deposition over ocean

7.3.1. Namelist NAM_CH_SEAFLUXn

Fortran name	Fortran type	values	default value
CCH_DRY_DEP	string of 6 characters	'NONE ', 'WES89 '	'WES89 '

- CCH_DRY_DEP: type of deposition scheme.
  - " NONE " : no chemical deposition scheme.
  - "WES89 " : Wesley (1989) deposition scheme.

7.4. Chemical deposition over lakes

7.4.1. Namelist NAM_CH_WATFLUXn

Fortran name	Fortran type	values	default value
CCH_DRY_DEP	string of 6 characters	'NONE ', 'WES89 '	'WES89 '

- CCH_DRY_DEP: type of deposition scheme.
    - " NONE " : no chemical deposition scheme.
    - "WES89 " : Wesley (1989) deposition scheme.

7.5. Chemical deposition over towns

7.5.1. Namelist NAM_CH_TEBn

Fortran name	Fortran type	values	default value
CCH_DRY_DEP	string of 6 characters	'NONE ', 'WES89 '	'WES89 '

- CCH_DRY_DEP: type of deposition scheme.
    - " NONE " : no chemical deposition scheme.
    - "WES89 " : Wesley (1989) deposition scheme.

7.6. Chemical deposition and biogenic emissions over vegetation

7.6.1. Namelist NAM_CH_ISBAn

Fortran name	Fortran type	values	default value
CCH_DRY_DEP	string of 6 characters	'NONE ', 'WES89 '	'WES89 '
LCH_BIO_FLUX	logical		.FALSE.
LCH_NO_FLUX	logical		.FALSE.

- CCH_DRY_DEP: type of deposition scheme.
    - " NONE " : no chemical deposition scheme.
    - "WES89 " : Wesley (1989) deposition scheme.
- LCH_BIO_FLUX: flag to activate the biogenic emissions.
- LCH_NO_FLUX: flag to calculate the NO emissions.

7.7. Chemical aerosol scheme (ORILAM)

7.7.1. Namelist NAM_CHS_ORILAM

Fortran name	Fortran type	values	default value
LCH_AERO_FLUX	logical		.FALSE.
LCO2PM	logical		.FALSE.
XEMISRADIUSI	real		0.05
XEMISRADIUSJ	real		0.2
XEMISSIGI	real		1.80
XEMISSIGJ	real		2.00
CRGUNIT	character	"MASS", "NUMB"	"NUMB"

- LCH_AERO_FLUX: switch to active aerosol surface flux for ORILAM
- "LCO2PM" : switch to activate emission of primary aerosol (Black and Organic carbon) compute from CO emssion. Uses only if CO emission is defined in the surface field (see PREP_PGD) and if there is no data for primary aerosol emissison.
- "XEMISRADIUSI" : Aerosol flux, mean radius of aitken mode in *Î¼m* (only if LCH_AERO_FLUX=.TRUE.).
- "XEMISRADIUSJ" : Aerosol flux, mean radius of accumulation mode in *Î¼m* (only if LCH_AERO_FLUX=.TRUE.).
- "XEMISSIGI" : Aerosol flux, standard deviation of aitken mode in *Î¼m* (only if LCH_AERO_FLUX=.TRUE.).
- "XEMISSIGJ" : Aerosol flux, standard deviation of accumulation mode in *Î¼m* (only if LCH_AERO_FLUX=.TRUE.).
- "CRGUNIT" : Aerosol flux, Definition of XEMISRADIUSI or XEMISRADIUSJ: mean radius can be define in mass ("MASS") or in number ("NUMB").

8. Externalized surface diagnostics

The diagnostics for the surface require the call to the complete physics of the surface. Therefore, they can be computed either during the run of the schemes (in order to have for example continuous time series of these diagnostics), or can be computed at a given instant only, if atmospheric forcing is given at this instant for the surface scheme to do one time step. The cumulated diagnostics are of course significant only when computed during a run.

8.1. Diagnostics relative to the general surface monitor

8.1.1. Namelist NAM_DIAG_SURF_ATMn

Fortran name	Fortran type	default value
LFRAC	logical	.FALSE.
LDIAG_GRID	logical	.FALSE.

- LFRAC: flag to save in the output file the sea, inland water, natural covers and town fractions.
- LDIAG_GRID: flag for mean grid diagnostics

8.2. Diagnostics relative to the general surface monitor and to each surface scheme

8.2.1. Namelist NAM_DIAG_SURFn

Fortran name	Fortran type	values	default value
N2M	integer	0, 1, 2	0
LSURF_BUDGET	logical		.FALSE.
LSURF_BUDGETC	logical		.FALSE.
LRESET_BUDGETC	logical		.FALSE.
LRAD_BUDGET	logical		.FALSE.
LCOEF	logical		.FALSE.
LSURF_VARS	logical		.FALSE.
L2M_MIN_ZS	logical		.FALSE.

- N2M : flag to compute surface boundary layer characteristics:
    - N2M=1 : computes temperature at 2 m, specific humidity at 2 m, relative humidity, zonal and meridian wind at 10 m, and Richardson number. 2m and 10m quantities are calculated extrapolating atmospheric forcing variables with Paulson laws using surface heat, water and momentum fluxes.
    - N2M=2 : computes temperature at 2 m, specific humidity at 2 m, relative humidity, zonal and meridian wind at 10 m, and Richardson number. 2m and 10m quantities are calculated interpolating between atmospheric forcing variables and surface temperature and humidity.
- LSURF_BUDGET: flag to save in the output file the terms of the surface energy balance (net radiation, sensible heat flux, latent heat flux, ground flux), for each scheme (on the four separate tiles), on each patch of the vegetation scheme if existing, and aggregated for the whole surface. The diagnosed fields are (* stands for the scheme considered (*=nothing : field aggregated on the whole surface;*=name of a scheme : field for this scheme):
    - RN_* : net radiation
    - H_* : turbulent sensible heat flux
    - LE_* : turbulent latent heat flux
    - GFLUX_*: ground or storage heat flux
    - FMU_* : zonal wind stress
    - FMV_* : meridian wind stress

      If both LSURF_BUDGET and LRAD_BUDGET are T then downward and upward shortwave radiation per spectral band will be written into output file (theyâ re computed even if LRAD_BUDGET is false). The following output fields are then available:
    - SWD_* : downward short wave radiation
    - SWU_*: upward short wave radiation
    - SWBD_* : downward short wave radiation for each spectral band
    - SWBU_* : upward short wave radiation for each spectral band
    - LWD_*:downward long wave radiation
    - LWU_* : upward long wave radiation

- LSURF_BUDGETC: flag to save in the output file the time integrated values of all budget terms that have been activated
- LRESET_BUDGETC: flag to reset cumulatives variables at the beginning of a run
- LCOEF: flag to save in the output file the transfer coefficients used in the computation of the surface energy fluxes, for each scheme (on the four separate tiles) and aggregated for the whole surface. The diagnosed fields are (* stands for the scheme considered (*=nothing : field aggregated on the whole surface; *=name of a scheme : field for this scheme):
    - CD_* : drag coefficient for momentum
    - CH_* : drag coefficient for heat
    - CE_* : drag coefficient for evaporation (differs from CH only over sea)
    - Z0_* : roughness length
    - Z0H_* : thermal roughness length
- LSURF_VARS: flag to save in the output file the surface specific humidity for each scheme (on the four separate tiles), on each patch of the vegetation scheme if existing. The diagnosed fields are (* stands for the scheme considered (*=nothing : field aggregated on the whole surface; *=name of a scheme :
- QS_* : specific humidity

- L2M_MIN_ZS: flag for 2 meters quantities evaluated on the minimum orographyy of the grid

8.2.2. Namelist NAM_WRITE_DIAG_SURFn

Fortran name	Fortran type	values	default value
LSELECT	logical		.FALSE.
CSELECT	array of string of characters		
LPROVAR_TO_DIAG	logical		.FALSE.

- LSELECT: if true it indicates that a selection will be used as output .
- CSELECT: array containing the list of output fields .
- LPROVAR_TO_DIAG: used to write out prognostic variables like diagnostic one, on average over all patches.

8.3. Diagnostics relative to the ISBA vegetation scheme

8.3.1. Namelist NAM_DIAG_ISBAn

Fortran name	Fortran type	default value
LPGD	logical	.FALSE.
LSURF_EVAP_BUDGET	logical	.FALSE.
LSURF_MISC_BUDGET	logical	.FALSE.
LPATCH_BUDGET	logical	.TRUE.
LSURF_MISC_DIF	logical	.FALSE.
LWATER_BUDGET	logical	.FALSE.

- LPGD : flag to save in the output file the physiographic fields of ISBA scheme that are computed from ecoclimap data from the ecosystem fractions.
- LSURF_EVAP_BUDGET: flag to save in the output file the detailed terms of the water vapor fluxes, on each patch of the vegetation scheme if existing, and aggregated for the natural surface. The diagnosed fields are:
- LSURF_MISC_BUDGET: flag to save in the output file miscelleaneous fields. The diagnosed fields are:
    - ♦ HV : Halstead coefficient
    - ♦ SNG : snow fraction over bare ground
    - ♦ SNV : snow fraction over vegetation
    - ♦ SN : total snow fraction
    - ♦ SWI : soil wetness index for each ground layer $(w_g - w_{wilt})/(w_{fc} - w_{wilt})$ where w_g is the volumic water content, w_{fc} is the porosity and w_{wilt} corresponds to the plant wilting point.
    - ♦ GPP : Gross primary production
    - ♦ RDK : Dark respiration
- LPATCH_BUDGET: flag to save in the output file the diagnostics for each patch (default is .T.)
- LSURF_MISC_DIF: to calculate and write specific DIF diagnostics
- LWATER_BUDGET : to calculate and write the water budget

8.4. Diagnostics relative to the TEB town scheme

8.4.1. Namelist NAM_DIAG_TEBn

Fortran name	Fortran type	default value
LSURF_MISC_BUDGET	logical	.FALSE.
LPGD	logical	.FALSE.
LPGD_FIX	logical	.FALSE.
LUTCI	logical	.FALSE.

- LSURF_MISC_BUDGET: flag to save in the output file miscelleaneous fields. The diagnosed fields are:
  - ♦ Z0_TOWN : roughness length for town
  - ♦ QF_BLD : domestic heating
  - ♦ QF_BLDWFR : domestic heating
  - ♦ FLX_BLD : heat flux from bld
  - ♦ TI_BLD_EQ : internal temperature without heating
  - ♦ TI_BLDWFR : internal temperature without heating
  - ♦ QF_TOWN : total anthropogenic heat
  - ♦ DQS_TOWN : storage inside building
  - ♦ H_WALL : wall sensible heat flux
  - ♦ H_ROOF: roof sensible heat flux
  - ♦ H_ROAD: road sensible heat flux
  - ♦ RN_WALL : net radiation at wall
  - ♦ RN_ROOF : net radiation at roof
  - ♦ RN_ROAD: net radiation at road
  - ♦ GFLUX_WALL : net wall conduction flux
  - ♦ GFLUX_ROOF : net roof conduction flux
  - ♦ GFLUX_ROAD : net road conduction flux
  - ♦ LE_ROOF : roof latent heat flux
  - ♦ LE_ROAD : road latent heat flux
- LPGD : flag to save PGD fields if TEB garden is activated
- LPGD_FIX : flag to save fixed PGD fields if TEB garden is activated
- LUTCI : to calculate and write UTCI diagnostics

8.5. Diagnostics relative to the FLAKE scheme

8.5.1. Namelist NAM_DIAG_FLAKEn

Fortran name	Fortran type	default value
LWATER_PROFILE	logical	.FALSE.
XZWAT_PROFILE	real	

- LWATER_PROFILE: flag to save in the output file miscelleaneous fields. The diagnostic is temperature at the depths defined by:
    - XZWAT_PROFILE : depth of output levels (m) in namelist

8.6. Diagnostics relative to the 1D oceanic scheme

8.6.1. Namelist NAM_DIAG_OCEANn

Fortran name	Fortran type	default value
LDIAG_OCEAN	logical	.FALSE.

- LDIAG_OCEAN: flag for ocean variables

9. Externalized surface model output fields

Model output fields depend on the tile and on the configuration of run.

In case of NETCDF outputs files:

- Fields colored in orange are not written
- Fields colored in blue are never written for any tile
- Fields colored in green are not written for FLAKE and TEB.

9.1. Prognostic model output fields

9.1.1. ISBA / GARDEN / GREENROOF

For garden, add GD_ at beginning of fields names.
For greenroof, add GR_ at beginning of fields names. In case of the use of TEB patches, still add Tn_ at beginning of fields names, where n is the number of the current TEB patch.

- *TGl, WGl, WGIl*: temperature, liquid water and ice water contents for each layer
- *WR*: water intercepted on leaves
- **CPHOTO/='NON' and CPHOTO/='AGS' and CPHOTO/='AST'**: *LAI*

- **CPHOTO='NIT' or CPHOTO='NCB'**:
  - ♦ *BIOMASl*: biomass of previous day
  - ♦ *RESPIR2,nnbiomass-2*: daily cumulated respiration of biomass
  - ♦ **CPHOTO='NIT'**: *RESPIRnnbiomass-1,nnbiomass*

- *RESA*: aerodynamical resistance

9.1.2. ISBA

- **LFLOOD**: *Z0_FLOOD*: roughness length of flood water
- **LGLACIER**: *ICE_STO*: glacier ice storage

- if **CISBA=="DIF" .AND. CSOC=="SGH"**: *XPATCH*: for external prep with SOC

- *TSRAD_NAT*: radiative temperature

- **LLAND_USE**:
    - ♦ *OLD_PATCH, OLD_DG*

- **CPHOTO/='NON'**:
    - ♦ *AN, ANDAY, ANFM, LE_AGS*: net C02 assimilation, daily net CO2 assimilation, maximum leaf assimilation, evapotranspiration

- **CRESPSL='CNT'**:
    - ♦ *LITTERl*: litter pools
    - ♦ *SOILCARBl*: soil carbon pools
    - ♦ *LIGNIN_STRl*: ratio lignin/carbon in structural litter

- **CDSTYN='Y'**:
    - ♦ *FLX_DSTMd*: dust variables

9.1.3. TEB

In case of the use of several TEB patches, each of these fields names begin by *Tn_* where *n* is the number of the current TEB patch.

- *TROOFl, WS_ROOF*: roof temperatures, roof water content
- *TROADl, WS_ROAD*: road temperatures, road water content
- if **CWALL_OPT=="UNIF"**: *TWALLl*: wall temperatures
- if **CWALL_OPT=="TWO "**: *TWALLAl, TWALLBl*: A and B walls temperatures
- *T_WIN1*: outdoor window temperature. Used only with CBEM="BEM", but must be initialized in all cases
- *TI_BLD, TI_ROAD*: deep road temperature
- *T_CANYON, Q_CANYON*: temperature and humidity of canyon air
- if **CBEM=="BEM"**:
  - *QI_BLD*: internal building specific humidity
  - *T_WIN2*: indoor windoor temperature
  - *TFLOOl*: floor temperatures
  - *TMASSl* internal th. mass temperatures
  - *TI_ROAD*: deep road temperature
  - *TCANYON*: temperature of canyon air
  - *QCANYON*: humidity of canyon air

9.1.4. SNOW

Applied for ISBA, ROAD, ROOF, GARDEN and GREENROOF (SURF = VEG, RF, RD, GD, GR). With TEB patches, for town surfaces, add Tn_ at beginning of the fields, where n is the number of the patch.

- *WSN_SURF_l, RSN_SURFl, ASN_SURF*
- **CSNOW='1-L'**: *TSN_SURFl*
- **CSNOW='3-L' or CSNOW='CRO'**: *HSN_SURFl*
- **CSNOW='CRO'**: *SG1_SURFl, SG2_SURFl, SHI_SURFl, SAG_SURFl*

9.1.5. SEAFLUX

- **LMERCATOR**:
  - ♦ *DTFNSOLl, DTFSOLl, T_OC_RELl, S_OC_RELl, U_OC_RELl, V_OC_RELl*: tendency surface linked to non solar flux, budget term linked to solar flux, relaxation profiles for oceanic temperature, salinity, zonal current, meridian current
  - ♦ *TEMP_OCl, SALT_OCl, UCUR_OCl, VCUR_OCl, TKE_OCl, KMEL_OCl*: temperature, salinity, zonal current, meridian current, turbulent kinetic energy, oceanic mixing coefficient
  - ♦ *SEAINBATHl*: bathymetry index
  - ♦ *SSS, SEA_HMO*: sea surface salinity, oceanic mixing length

- **LINTERPOL_SST**:
  - ♦ *SST_MTHm*: SST month t

- *SST*: sea surface temperature
- *Z0SEA*: sea roughness length

9.1.6. WATFLUX

- **LINTERPOL_TS**:
    - ♦ *TS_WATERm*: TS_WATER month t

- *TS_WATER*: surface temperature
- *Z0WATER*: water roughness length

9.1.7. FLAKE

- *TS_WATER*: surface temperature
- *T_SNOW, T_ICE, T_MNW, T_WML, T_BOT, T_B1*: temperatures for snow, ice, mean, mixed layer, bottom, bottom of the upper layer of the sediments.
- *CT*: shape factor (thermocline)
- *H_SNOW, H_ICE, H_ML, H_B1*: snow, ice, mixed layer and upper layer of bottom sediments thickness
- *Z0WATER, USTAR_WATER*: roughness length and friction velocity

9.1.8. Canopy

For flake and watflux, replace SSO_CAN by WAT_SBL; for isba, by ISBA_CAN; for teb, by TEB_CAN; for seaflux, by SEA_SBL.

- **Common fields**:
  - ♦ *SSO_CAN_Zl*: altitudes of canopy levels (m)
  - ♦ **Not in PREP step**:
    - ◊ *SSO_CAN_Ul*: wind at canopy levels (m/s)
    - ◊ *SSO_CAN_El*: Tke at canopy levels (m2/s2)

- **Flake, Isba, Watflux, Seaflux, Teb (not in PREP step)**:
  - ♦ *WAT_SBL_Tl, WAT_SBL_Ql, WAT_SBL_Pl*: temperature, humidity and pression
  - ♦ *WAT_SBL_LMO*: Monin-Obhukov length

- **Teb (not in PREP step)**:
  - ♦ *TEB_CAN_LM, TEB_CAN_LE*: mixing and dissipative lengths

9.2. Diagnostic model output fields

9.2.1. All tiles aggregated

- **LFRAC (NAM_DIAG_SURF_ATMn)**:
  - ♦ *FRAC_SEA, FRAC_NATURE, FRAC_WATER, FRAC_TOWN*: fractions for each tile

- **N2M>=1 + L2M_MIN_ZS=T (NAM_DIAG_SURFn)**:
  - ♦ *T2M_MIN_ZS, Q2M_MIN_ZS, HU2M_MIN_ZS*: temperature, air humidity and air relative humidity at 2 meters brought back to min zs.

- **LSURF_BUDGET**:
  - ♦ *FMUNOSSO, FMVNOSSO*: zonal and meridian frictions, without and with SSO. (Pa)

- **N2M>=1 or LSURF_BUDGET or LSURF_BUDGETC**:
  - ♦ *TS, TSRAD, EMIS*: surface temperature, radiative temperature, surface emissivity

- **LCOEF**:
  - ♦ *UREF, ZREF*: reference heights for momentum and heat

9.2.2. For each tile and all tiles aggregated

Replace TILE by ISBA, WAT, SEA or TEB to get names of diagnostic fields. For the tiles aggregated, remove _TILE from the names of the fields.

All the following flags are in namelist **NAM_DIAG_SURFn**.

- **N2M>=1 (T2MMIN, T2MMAX, HU2MMIN, HU2MMAX, W10M and W10MMAX not for FLAKE nor TEB)**:
    - ♦ *RI_TILE*: Bulk-Richardson number
    - ♦ *T2M_TILE, T2MMIN_TILE, T2MMAX_TILE*: air temperature at 2 meters
    - ♦ *Q2M_TILE*: air humidity at 2 meters
    - ♦ *HU2M_TILE, HU2MMIN_TILE, HU2MMAX_TILE*: air relative humidity at 2 meters
    - ♦ *ZON10M_TILE, MER10M_TILE, W10M_TILE, W10MMAX_TILE*: zonal, meridian, wind at 10 meter (only if 1^{st} level of atmospheric model is upper than 10 m)
    - ♦ *SFCO2_TILE*: CO2 flux (KG/M2/S) (not for sea and water for the moment)

- **LSURF_BUDGET**:
    - ♦ *RN_TILE, H_TILE, LE_TILE, LEI_TILE, GFLUX_TILE*: net radiation at surface, sensible heat flux, total latent heat flux, sublimation latent heat flux, net soil-vegetation flux
    - ♦ **LRAD_BUDGET**:
        - ◊ *SWD_TILE, SWU_TILE, LWD_TILE, LWU_TILE, SWD_TILE_b, SWU_TILE_b*: short wave downward and upward radiation, long wave downward and upward radiation, short wave downward and upward radiation by spectral band
    - ♦ *FMU, FMV*: zonal and meridian frictions, without and with SSO. (Pa)

- **LSURF_BUDGETC (not for FLAKE nor for TEB):**
    - ♦ *RNC_TILE, HC_TILE, LEC_TILE, LEIC_TILE, GFLUXC_TILE*:cumulated fluxes at surface
    - ♦ **LRAD_BUDGET**:
        - ◊ *SWDC_TILE, SWUC_TILE, LWDC_TILE, LWUC_TILE*: cumulated radiations
    - ♦ *FMUC_TILE, FMVC_TILE*: cumulated frictions with SSO

- **LCOEF**:
    - ♦ *CD_TILE, CH_TILE, CE_TILE*: drag coefficients for wind, heat, vapor
    - ♦ *Z0_TILE, Z0H_TILE*: roughness lengths for momentum and heat

- **LSURF_VARS (not for aggregated tiles)**:
    - ♦ *QS_TILE*: specific humidity

9.2.3. ISBA & GARDEN

To get fields names for GARDEN, add GD_ at beginning of following names. (For albedos _ISBA, replace _ISBA by _T).

- **LPGD (values by patch) (and LGARDEN for garden)**:
  - ♦ **CPHOTO='NON' or CPHOTO='AGS' or CPHOTO='AST'**: *LAI*
  - ♦ *VEG, Z0VEG, PATCH, DGl*: vegetation fraction, surface roughness length, fraction for each patch, soil depth
  - ♦ *Z0REL*: orography roughness length
  - ♦ **CHORT='SGH'**: *DICE*: soil ice depth for runoff
  - ♦ *VEGTY_Pp*: fraction of each vegetation type for each patch
  - ♦ *RSMIN, GAMMA, CV, RGL, EMIS_ISBA, WRMAX_CF*: minimum stomatal resistance, coefficient for RSMIN calculation, vegetation thermal inertia coefficient, maximum solar radiation usable in photosynthesis, surface emissivity, coefficient for maximum water interception
  - ♦ **LSURF_DIAG_ALBEDO**: *ALBNIR_S, ALBVIS_S, ALBUV_S, ALBNIR_ISBA, ALBVIS_ISBA, ALBUV_ISBA*: near infrared, visible, uv soil and total albedos

LPGD is in namelists **NAM_DIAG_ISBAn** and **NAM_DIAG_TEBn**.
LGARDEN is in namelist **NAM_PGD_SCHEMES**.
CHORT is in namelist **NAM_SGH_ISBAn**.
LSURF_DIAG_ALBEDO is T for the run step.

9.2.4. ISBA

- **LSURF_EVAP_BUDGET (with LSURF_BUDGETC, can be cumulated in ...C_ISBA) (NAM_DIAG_ISBAn)**:
  - ♦ *LEG_ISBA, LEGI_ISBA, LEV_ISBA, LES_ISBA, LER_ISBA, LETR_ISBA*: bare ground evaporation and sublimation, vegetation evaporation and sublimation, evaporation due to interception for tile nature, vegetation evapotranspiration
  - ♦ if **CSNOW=="3-L" or CSNOW=="CRO"**: *LESL_ISBA*: liquide water evaporation over snow
  - ♦ *EVAP_ISBA, DRAIN_ISBA, RUNOFF_ISBA*: total evaporation flux, drainage, runoff
  - ♦ *DRIVEG_ISBA, RRVEG_ISBA, SNOMLT_ISBA*: dripping from the vegetation reservoir, precipitation intercepted by vegetation, snow melt
  - ♦ **CHORT='SGH' (NAM_SGH_ISBAn) or CISBA='DIF' (NAM_ISBA)**:
    - ◊ *HORTON_ISBA*: horton runoff
  - ♦ **LFLOOD (NAM_SGH_ISBAn)**:
    - ◊ *IFLOOD, PFLOOD*: floodplains infiltration, precipitation intercepted by the floodplains
    - ◊ *LEF_ISBA, LEIF_ISBA*: floodplains evaporation and sublimation
  - ♦ **CPHOTO/='NON' (NAM_ISBA)**: *GPP_ISBA, R_AUTO_ISBA, R_ECO_ISBA*: gross primary production, autotrophic respiration, ecosystem respiration
  - ♦ **LWATER_BUDGET (NAM_DIAG_ISBAn)**:
    - ◊ *RAINF_ISBA, SNOWF_ISBA, DWG_ISBA, DWGI_ISBA, DWR_ISBA, DSWE_ISBA, WATBUD_ISBA*: input rainfall rate, input snowfall rate, change in liquid soil moisture, change in solide soil moisture, change in water on canopy, change in snow water equivalent, isba water budget as residue

- **LSURF_BUDGETC (NAM_DIAG_SURFn) and LGLACIER (NAM_ISBAn)**:
  - ♦ *ICE_FC_ISBA*: ice flux

- **LPATCH_BUDGET (NAM_DIAG_ISBAn) and NPATCH > 1** :
  - ♦ upper and common fields can be given by patch by replacing _ISBA by _P in their names.

- **LSURF_MISC_BUDGET (NAM_DIAG_ISBAn)**:
  - ♦ *HV_ISBA*: Halstead coefficient
  - ♦ *PSNG_ISBA, PSNV_ISBA, PSN_ISBA*: snow fraction over ground, vegetation and total
  - ♦ *TALB_ISBA*: total albedo
  - ♦ **CSNOW='3-L' or CSNOW='CRO' (NAM_PREP_ISBA_SNOW)**:
    - ◊ *TS_ISBA, TSRAD_ISBA*: total surface and radiative temperature
  - ♦ *SWIl_ISBA, TSWIl_ISBA*: soil wetness index and total swi by layer
  - ♦ *TSWI_T_ISBA*: total soil wetness index over the soil column
  - ♦ *WGTOT_T_ISBA, WGI_T_ISBA*: total water and ice content over the soil column
  - ♦ **CISBA='DIF' (NAM_ISBA) AND LSURF_MISC_DIF (NAM_DIAG_ISBAn)**:
    - ◊ *TSWI_R_ISBA, WGTOT_R_ISBA, WGI_R_ISBA*: total soil wetness index, water and ice contents over the root zone

- ◊ *TSWI_S_ISBA, WG_S_ISBA, WGI_S_ISBA*: total soil wetness index, water and ice contents over the surface
- ◊ *TSWI_D2_ISBA, WG_D2_ISBA, WGI_D2_ISBA*: total soil wetness index, water and ice contents over comparable FR-DG2 reservoir
- ◊ *TSWI_D3_ISBA, WG_D3_ISBA, WGI_D3_ISBA*: total soil wetness index, water and ice contents over comparable FR-DG3 reservoir

♦ **CISBA='DIF' (NAM_ISBA)** :
- ◊ *ALT_ISBA, FLT_ISBA*: active layer thickness over permafrost, frozen layer thickness over non-permafrost

♦ *WSNOW_T_ISBA, DSNOW_T_ISBA, TSNOW_T_ISBA*: total snow reservoir, depth and temperature

♦ **CRUNOFF='SGH' or CRUNOFF='DT92' (NAM_SGH_ISBAn)**:
- ◊ *FSAT_ISBA*: soil saturated fraction

♦ **CRAIN='SGH' (NAM_SGH_ISBAn)**:
- ◊ *MUF_ISBA*: fraction of the grid cell reached by the rainfall

♦ **LFLOOD (NAM_SGH_ISBAn)**:
- ◊ *FFG_ISBA, FFV_ISBA, FF_ISBA*: flood fraction over ground, vegetation and total
- ◊ *FFLOOD_ISBA, PIFLOOD_ISBA*: grid-cell potential flood fraction and floodplain inflitration

♦ **LPATCH_BUDGET: previously defined fields for each patch**:
- ◊ *SWIl, TSWIl*
- ◊ if **CISBA==â DIFâ** : *ALT_PATCH, FLT_PATCH*
- ◊ *HV, PSNG, PSNV, PSN, TALB*
- ◊ **CSNOW='3-L' or CSNOW='CRO'**: *TS_PATCH, TSRAD_PATCH, SNOWLIQl, SNOWTEMPl*
- ◊ *WSNOW_VEGT, DSNOW_VEGT, TSNOW_VEGT*
- ◊ **CRUNOFF='DT92'**: *FSAT_PATCH*
- ◊ **LFLOOD**: *FFG_PATCH, FFV_PATCH, FF_PATCH*

♦ **LAGRIP (NAM_AGRI)**: *IRRISEUIL*: irrigation threshold

♦ **LTR_ML (NAM_ISBA)**:
- ◊ *FAPAR, FAPIR, FAPAR_BS, FAPIR_BS, DFAPARC, DFAPIRC*: fapar and fapir of vegetation, of bare soil, and cumulated from 0 UTC
- ◊ *DLAI_EFFC*: cumulated effective LAI

• **LPGD (values by patch)**:

♦ **CISBA='DIF'**:
- ◊ *DROOT_DIF, DG2_DIF, RUNOFFD, DTOT_DIF, ROOTFRACl*: root depth, dg2 depth, runoff depth, total soil depth for moisture, root fraction by layer
- ◊ *W33*: moisture threshold for bare soil evaporation (not by patch)

♦ *WSATl, WFCl, WWILTl*: soil porosity, field capacity, wilting point, by layer (not by patch)

♦ **LAGRIP AND (CPHOTO=LAI' or CPHOTO='LST' or CPHOTO='NIT' or CPHOTO='NCB')**:
- ◊ *TSEED, TREAP*: dates of seeding and reaping
- ◊ *IRRIG, WATSUP*: flag for irrigation, water supply during irrigation process

- **LPROVAR_TO_DIAG (NAM_WRITE_DIAG_SURFn)**:
  - ♦ *TGl_ISBA, WGl_ISBA, WGIl_ISBA, WR_ISBA*
  - ♦ **LGLACIER**: *ICE_STO_ISBA*
  - ♦ *ASNOW_ISBA*
  - ♦ **CSNOW='3-L' or CSNOW='CRO'**: *WSNOW_l_ISBA, DSNOW_l_ISBA, TSNOW_l_ISBA*
  - ♦ **CPHOTO/='NON' and CPHOTO/='AGS' and CPHOTO/='AST'**: *LAI_ISBA*

9.2.5. TEB

- **LSURF_MISC_BUDGET (NAM_DIAG_TEBn)**:

In case of the use of TEB patches, add Tn_ at beginning of fields name, where n is the number of the current TEB patch.

-
  - *D_RD*: road fraction
  - *Z0_TOWN*: town roughness length
  - *XQF_BLD: domestic heating*
  - *XFLX_BLD*: heat flux from bld
  - *XQF_TOWN*: total anthropogenic heat
  - *XDQS_TOWN*: heat storage inside building
  - *RN_RD, H_RD, LE_RD, GFLUX_RD*: fluxes for road
  - if **CWALL_OPT=="UNIF"**: *RN_WL, H_WL, GFLUX_WL*: fluxes for wall
  - if **CWALL_OPT=="TWO "**: *RN_WLA, RN_WLB,H_WALLA, H_WALLB, GFLUX_WLA,GFLUX_WLB*: fluxes for wall
  - *RN_RF, H_ROOF, LE_ROOF, GFLUX_RF*: fluxes for roof
  - *RUNOFF_TWN*: runoff for town
  - if **LGARDEN**: *RN_GD, H_GD, LE_GD, GFLUX_GD*: fluxes for garden
  - *RN_BLT, H_BLT, LE_BLT, GFLUX_BLT*: fluxes for built surfaces
  - *SWA_RF, SWA_SN_RF, LWA_RF, LWA_SN_RF*: sdown and ldown absorbed by roof and by snow on roof
  - *SWA_RD, SWA_SN_RD, LWA_RD, LWA_SN_RD*: sdown and ldown absorbed by road and by snow on road
  - IF **CWALL_OPT=="UNIF"**: *SWA_WL, LWA_WL*: sdown and ldown absorbed by wall
  - IF **CWALL_OPT=="TWO "**: *SWA_WLA, LWA_WLA, SWA_WLB, LWA_WLB*: sdown and ldown absorbed by wall A and B
  - IF **LGARDEN**: *SWA_GD, LWA_GD* : sdown and ldown absorbed by garden
  - *REF_SW_GO, LWE_GO*: total solar rad reflected by ground, LW emitted by ground (w/m2)
  - REF_SW_FA, LWE_FA: total solar rad reflected by facade, LW emitted by facade (W/m2)
  - IF **CBEM=="BEM"**:
  - *H_BLD_CL*: sensible cooling demand (W/m2)
  - *T_BLD_CL*: total cooling demand (W/m2)
  - *H_BLD_HT*: sensible heating demand (W/m2)
  - *LE_BLD_CL*: latent cooling demand (W/m2)
  - *LE_BLD_HT*: latent heating demand (W/m2)
  - *H_WASTE*: sensible waste heat from HVAC (W/m2)
  - *LE_WASTE*: latent waste heat from HVAC (W/m2)
  - *HVAC_CL*: cooling energy consumption (W/m2)
  - *HVAC_HT*: heating energy consumption (W/m2)
  - *CAP_SYS*: actual capacity of the cooling system (W m-2(bld))
  - *M_SYS*: actual HVAC mass flow rate (kg s-1 m-2(bld)

- ♦ *COP*: actual COP of the cooling system ()
- ♦ *Q_SYS*: supply air specific humidity (kg kg-1)
- ♦ *T_SYS*: supply air temperature (K)
- ♦ *TR_SW_WIN*: solar radiation transmitted through windows (W m-2(bld))
- ♦ *FAN_POWER*: HVAC fan power (W m-2(bld))
- ♦ *T_RAD_IND*: indoor mean radiant temperature (K)
- ♦ *HU_BLD*: indoor relative humidity (-)
- ♦ *SWA_WIN*: Sdown absorbed by windows (W/m2)
- ♦ *LWA_WIN*: Ldown absorbed by windows (W/m2)
- ♦ IF **LGREENROOF**:
- ♦ *RN_GR, H_GR, LE_GR, GFLUX_GR*: fluxes for greenroofs
- ♦ *SWA_GR*: Sdown absorbed by greenroofs (W/m2)
- ♦ *LWA_GR*: Ldown absorbed by greenroofs (W/m2)
- ♦ *G_GR_ROOF*: heatflux between greenroof and roof (W/m2)
- ♦ *RUNOFF_GR*: greenroof soil surface runoff (kg/m2/s)
- ♦ *DRAIN_GR*: greenroof total vertical drainage (kg/m2/s)

- **LPGD**:
  - ♦ *BLD*: building fraction
  - ♦ *WALL_O_HOR*: wall surface over plan area surface
  - ♦ *BLD_HEIGHT*: building height (m)
  - ♦ *Z0_TOWN*: town roughness length (m)
  - ♦ *ROAD_DIR*: road direction
  - ♦ *GD_FRAC*: garden fraction
  - ♦ *GR_FRAC*: greenroof fraction
  - ♦ *ALB_ROOF*: roof albedo
  - ♦ *EMIS_ROOF*: roof emissivity
  - ♦ *HC_ROOFl*: roof heat capacity for each layer
  - ♦ *TC_ROOFl*: roof thermal conductivity for each layer
  - ♦ *D_ROOFl*: roof layer thickness for each layer
  - ♦ *ROUGH_ROOF*: roof roughness
  - ♦ *ALB_WALL*: wall albedo
  - ♦ *EMIS_WALL*: wall emissivity
  - ♦ *HC_WALLl*: wall heat capacity for each layer
  - ♦ *TC_WALLl*: wall thermal conductivity for each layer
  - ♦ *D_WALLl*: wall layer thickness for each layer
  - ♦ *ROUGH_WALL*: wall roughness
  - ♦ *ALB_ROAD*: road albedo
  - ♦ *EMIS_ROAD*: road emissivity
  - ♦ *HC_ROADl*: road heat capacity for each layer
  - ♦ *TC_ROADl*: road thermal conductivity for each layer
  - ♦ *D_ROADl*: road layer thickness for each layer
  - ♦ *H_TRAFFIC*: traffic heat flux
  - ♦ *LE_TRAFFIC*: traffic latent flux
  - ♦ *H_INDUSTRY*: industry heat flux
  - ♦ *LE_INDUSTRY*: industry latent flux

- **LPGD AND CBEM=="BEM"**:
  - ♦ *HC_FLOORl*: floor heat capacity for each layer
  - ♦ *TC_FLOORl*: floor thermal conductivity for each layer
  - ♦ *D_FLOORl*: floor layer thickness for each layer

- **N2M>0 AND LUTCI**:
  - ♦ *UTCI_IN*: UTCI for person indoor (Â°C)
  - ♦ *UTCI_OUTSUN*: UTCI for person at sun (Â°C)
  - ♦ *UTCI_OUTSHAD*:UTCI for person in shade (Â°C)
  - ♦ *TRAD_SUN*: mean radiant temperature seen by person at sun (K)
  - ♦ *TRAD_SHADE*: mean radiant temperature seen by person in shade (K)

9.2.6. SEAFLUX

- **LPROVAR_TO_DIAG (NAM_WRITE_DIAG_SURFn) or LINTERPOL_SST (NAM_SEAFLUXn)**:
    - ♦ *SST*: sea surface temperature

- **LDIAG_OCEAN (NAM_DIAG_OCEANn)**:
    - ♦ *TOML, SOML, UOML, VOML, DOML*: mean temperature, salinity, zonal current, meridian current, density

9.2.7. WATFLUX

- **LPROVAR_TO_DIAG (NAM_WRITE_DIAG_SURFn) or LINTERPOL_TS (NAM_WATFLUXn)**:
    - ♦ *TS_WATER*: surface temperature

- **LWATER_PROFILE (for FLAKE) (NAM_DIAG_FLAKEn)**:
    - ♦ *TW_d*: water temperature in output levels

9.2.8. Chemical diagnostics

- **NBEQ>0**:
  - ♦ **CCH_DRY_DEP='WES89' (NAM_CH_SEAFLUXn, NAM_CH_WATFLUXn, NAM_CH_TEBn, NAM_CH_ISBAn)**: *DV_NAT_name, DV_TEB_name, DV_WAT_name, DV_SEA_name*
  - ♦ **LCH_BIO_FLUX (NAM_CH_ISBAn)**: *FISO, FMONO*: isoprene and monoterpenes emission fluxes

- **CDSTYN='Y'**: *FLX_DSTd*: dust variables to be send to output

- if **CCH_EMIS==â SNAPâ** : *EMIS_name*: emission data at time t (ppm*m/s)

9.3. Physiographic fields

9.3.1. Common fields

- **GRID**:
    - *CONF PROJ, CARTESIAN, LONLATVAL*: XX, YY, DX, DY
    - *LONLAT REG*: REG_LON, REG_LAT
    - *GAUSS*: LATGAUSS, LONGAUSS, LAT_G_XY, LON_G_XY, MESHGAUSS

- *COVER FIELDS, ZS*

- *AVG_ZS, SIL_ZS*: orography
- *SSO_STDEV, MIN_ZS, MAX_ZS, SSO_ANIS, SSO_DIR, SSO_SLOPE*: subgrid orography
- *HO2IP, HO2JP, HO2IM, HO2JM, AOSIP, AOSJP, AOSIM, AOSJM*: subgrid orography roughness

- *DUMMY_GRd*: dummy fields

- if **CCH_EMIS==â AGGRâ** : *EMIS_name*: emission data (x, y, t)

- if **CCH_EMIS==â SNAPâ** : *EMIS_name_M, EMIS_name_D, EMIS_name_H, SNAPi_name*: temporal profiles of snap, potential emissions of species for each snap

9.3.2. ISBA / GARDEN / GREENROOF

For GARDEN, add GD_ at beginning of fields names.
For GREENROOF, add GR_ at beginning of fields names.

- *CLAY, SAND* : (not for GREENROOF)

- *RUNOFFB*: orographic runoff coefficient
- *WDRAIN*: subgrid drainage coefficient

9.3.3. ISBA

- **LSOCP**:
  - ♦ *SOC_TOP, SOC_SUB*: clay, sand, organic carbon

- **LPERM**:
  - ♦ *PERM*: permafrost distribution

- **LNOF**:
  - ♦ *PH, FERT*: pH and fertilization rate for NO2 fluxes

- **LCTI**: topographic index statistics
  - ♦ *TI_MIN, TI_MAX, TI_MEAN, TI_STD, TI_SKEW*

9.3.4. ISBA_PAR

- *D_VEGTYPE*
- *D_VEG_Ttime*
- *D_LAI_Ttime*
- *D_Z0_Ttime*
- *D_EMIS_Ttime*
- *D_RSMIN*
- *D_GAMMA*
- *D_WRMAX_CF*
- *D_RGL*
- *D_CV*
- *D_Z0_O_Z0H*
- *D_DGlayer*
- *D_ROOTFRAClayer*
- *D_GROUND_DPT*
- *D_ROOT_DEPTH*
- *D_ROOT_EXT*
- *D_ROOT_LIN*
- *D_DICE*
- *D_ALBNIR_VEG*
- *D_ALBVIS_VEG*
- *D_ALBUV_VEG*
- *D_ALBNIR_SOI*
- *D_ALBVIS_SOI*
- *D_ALBUV_SOI*
- *D_GMES*
- *D_BSLAI*
- *D_LAIMIN*
- *D_SEFOLD*
- *D_GC*
- *D_DMAX*
- *D_STRESS*
- *D_H_TREE*
- *D_RE25*
- *D_CE_NITRO*
- *D_CF_NITRO*
- *D_CNA_NITRO*
- *D_IRRIG_Ttime*
- *D_WATSUP_Ttime*

9.3.5. TSZ0_PAR

- *D_DTS*
- *D_DHUGRD*

9.3.6. TEB_PAR

- *D_BLDTYPE*
- *D_BLD_AGE*
- *D_USETYPE*
- *D_Z0_TOWN*
- *D_BLD*
- *D_GARDEN*
- *D_ROAD_DIR*
- *D_ALB_ROOF*
- *D_EMI_ROOF*
- *D_HC_ROOFl*
- *D_TC_ROOFl*
- *D_D_ROOFl*
- *D_ALB_ROAD*
- *D_EMIS_ROAD*
- *D_HC_ROADl*
- *D_TC_ROADl*
- *D_D_ROADl*
- *D_ALB_WALL*
- *D_EMI_WALL*
- *D_HC_WALLl*
- *D_TC_WALLl*
- *D_D_WALLl*
- *D_BLD_HEIG*
- *D_WALL_O_H*
- *D_H_TRAF*
- *D_LE_TRAF*
- *D_H_IND*
- *D_LE_IND*
- *D_ROUGH_ROOF*
- *D_ROUGH_WALL*
- if **CBEM=="BEM"**:
    - *D_HC_FLOORl*
    - *D_TC_FLOORl*
    - *D_D_FLOORl*
    - *D_TCOOL_TARG*
    - *D_THEAT_TARG*
    - *D_F_WAST_CAN*
    - *D_EFF_HEAT*
    - *D_QIN*
    - *D_QIN_FRAD*
    - *D_SHGC*
    - *D_U_WIN*
    - *D_GR*
    - *D_SHGC_SH*

- ♦ *D_FLOOR_HEIG*
- ♦ *D_INF*
- ♦ *D_QIN_FLAT*
- ♦ *D_HR_TARGET*
- ♦ *D_V_VENT*
- ♦ *D_CAP_SYS_HE*
- ♦ *D_CAP_SYS_RA*
- ♦ *D_T_ADP*
- ♦ *D_M_SYS_RAT*
- ♦ *D_COP_RAT*
- ♦ *D_T_SIZE_MAX*
- ♦ *D_T_SIZE_MIN*
- ♦ *D_FWAT_COND*
- ♦ *D_SHADE*
- ♦ *D_NATVENT*

- if **LGARDEN**:
  - ♦ *D_FRAC_LVEG*
  - ♦ *D_FRAC_NVEG*
  - ♦ *D_LAI_HVEGtime*
  - ♦ *D_LAI_LVEGtime*
  - ♦ *D_H_HVEG*
- if **LGREENROOF**:
  - ♦ *D_FRAC_GR*
  - ♦ *D_OM_GR*
  - ♦ *D_CLAY_GR*
  - ♦ *D_SAND_GR*
  - ♦ *D_LAI_GRtime*

9.3.7. SEAFLUX

- *BATHY*: bathymetry

9.3.8. SEAF_PAR

- *D_SST_Ttime*
- *TDATA_SST*

9.3.9. FLAKE

- *WATER_DEPTH, WATER_FETCH, T_BS, DEPTH_BS, EXTCOEF_WAT*

10. Scanning of main SURFEX features through the namelist options

10.1. PGD step

10.1.1. Different types of grids

10.1.1.1. CONF PROJ grid

```
&NAM_PGD_GRID           CGRID = 'CONF PROJ '
/
&NAM_CONF_PROJ_GRID
  XLATCEN=43.,
  XLONCEN=0.
  NIMAX=12,
  NJMAX=8,
  XDX=60000.,
  XDY=60000.
/
&NAM_CONF_PROJ
  XLAT0=43., XLON0=0., XRPK=1.0, XBETA=0.
/
```

10.1.1.2. Cartesian grid

```
&NAM_PGD_GRID          CGRID = 'CARTESIAN'
/
&NAM_CARTESIAN         XLAT0 = 45.,
                       XLON0 = 2.,
                       NIMAX = 10,
                       NJMAX = 10,
                       XDX = 500,
                       XDY = 500
/
```

10.1.1.3. Gauss grid

```
&NAM_PGD_GRID           CGRID = 'GAUSS '
/
&NAMDIM                 NDGLG = 4
/
&NAMRGRI                NRGRI = 4,4,4,4
/
```

10.1.1.4. LONLATREG grid

```
&NAM_PGD_GRID          CGRID = 'LONLAT REG'
/
&NAM_LONLAT_REG
                       XLONMIN = -1.,
                       XLONMAX = 10,
                       XLATMIN = 40,
                       XLATMAX = 50,
                       NLON = 110,
                       NLAT = 100
/
```

10.1.1.5. IGN grid

```
&NAM_PGD_GRID           CGRID = 'IGN '
/
&NAM_IGN                    CLAMBERT  = 'L2E'   ,
                            NPOINTS   = 5    ,
                            XX        =
                                         876000.        ,
                                         892000.        ,
                                         860000.        ,
                                         868000.        ,
                                         876000.        ,
                            XY        =
                                         2361000.        ,
                                         2361000.        ,
                                         2353000.        ,
                                         2353000.        ,
                                         2353000.        ,
                            XCELLSIZE = 8000.
/
```

10.1.1.6. LONLATVAL grid

```
&NAM_PGD_GRID           CGRID = 'LONLATVAL'
/
&NAM_LONLATVAL             NPOINTS   = 5    ,
                            XX        = 0.,0.2,0.45,0.6,0.9,
                            XY        = 45.1,47.2,47.8,48.,48.5 ,
                            XDX = 0.2,0.01,0.7,0.05,0.15,
                            XDY = 0.05,0.07,0.3,0.1,0.5
/
```

10.1.2. Options for ISBA scheme

10.1.2.1. To run with ISBA 2-L

```
&NAM_PGD_SCHEMES CNATURE = 'ISBA',
                 CSEA = 'NONE',
                 CTOWN = 'NONE',
                 CWATER = 'NONE,
                 LGARDEN = F
/
&NAM_ISBA CISBA = '2-L',
                   CPHOTO = 'NON',
                   NPATCH = 1,
                   NGROUND_LAYER = 2
/
```

10.1.2.2. To run with ISBA 3-L

```
&NAM_PGD_SCHEMES CNATURE = 'ISBA',
                 CSEA = 'NONE',
                 CTOWN = 'NONE',
                 CWATER = 'NONE,
                 LGARDEN = F
/
&NAM_ISBA CISBA = '3-L',
                   CPHOTO = 'NON',
                   NPATCH = 1,
                   NGROUND_LAYER = 3
/
```

10.1.2.3. To run with ISBA DIF (idealized grid)

```
&NAM_PGD_SCHEMES CNATURE = 'ISBA',
                 CSEA = 'NONE',
                 CTOWN = 'NONE',
                 CWATER = 'NONE,
                 LGARDEN = F
/
&NAM_ISBA CISBA = 'DIF',
                   CPHOTO = 'NON',
                   NPATCH = 1,
                   NGROUND_LAYER = 14
/
```

10.1.2.4. To run with 12 patchs and photosynthesis options

```
&NAM_PGD_SCHEMES CNATURE = 'ISBA',
                 CSEA = 'NONE',
                 CTOWN = 'NONE',
                 CWATER = 'NONE,
                 LGARDEN = F
/
&NAM_ISBA  CPHOTO = 'NON/AGS/LST/AST/LAI/NIT/NCB',
                     NPATCH = 12
/
```

10.1.2.5. To activate the new radiative transfer

```
&NAM_PGD_SCHEMES CNATURE = 'ISBA',
                 CSEA = 'NONE',
                 CTOWN = 'NONE',
                 CWATER = 'NONE,
                 LGARDEN = F
/
&NAM_ISBA LTR_ML = T
/
```

10.1.2.6. To change the pedotransfert function

```
&NAM_PGD_SCHEMES CNATURE = 'ISBA',
                 CSEA = 'NONE',
                 CTOWN = 'NONE',
                 CWATER = 'NONE,
                 LGARDEN = F
/
&NAM_ISBA CISBA = 'DIF',
                   CPEDO_FUNCTION = 'CH78'/'CO84'/'CP88'/'WO99'
/
```

10.1.2.7. To define the physiographic fields

```
&NAM_PGD_SCHEMES CNATURE = 'ISBA',
                 CSEA = 'NONE',
                 CTOWN = 'NONE',
                 CWATER = 'NONE,
                 LGARDEN = F
/
```

General (all configurations):

```
&NAM_ISBA       YCLAY = 'CLAY_HWSD_MOY',
                YCLAYFILETYPE = 'DIRECT',
                YSAND = 'SAND_HWSD_MOY',
                YSANDFILETYPE = 'DIRECT',
                YRUNOFFB = 'RUNOFFB.SIM',
                YRUNOFFBFILETYPE = 'ASCLLV',
                YWDRAIN = 'WDRAIN.SIM',
                YWDRAINFILETYPE = 'ASCLLV'
/
```

With :

```
&NAM_ISBAn   CSOC = 'SGH'
/

&NAM_ISBA      CISBA = 'DIF'
               YSOC_TOP = 'soc_top',
               YSOC_SUB = 'soc_sub',
               YSOCFILETYPE = 'DIRECT',
/
```

For the permafrost depth:

```
&NAM_ISBA      CISBA='DIF',
               YPERM = 'perm_glo_10km',
               YPERMFILETYPE = 'DIRECT'
/
```

With:

```
&NAM_ISBAn   CRUNOFF = 'SGH'
/

&NAM_ISBA      YCTI           = 'topo_index'  ,
               YCTIFILETYPE   = 'DIRECT'
/
```

With :

```
&NAM_SGH_ISBAn   LCH_NO_FLUX = T
/
```

```
&NAM_ISBA      XUNIF_PH = 6,
               XUNIF_FERT = 200.
/
```

10.1.3. Options for the TEB scheme

10.1.3.1. General options

```
&NAM_PGD_SCHEMES CNATURE = 'NONE',
                 CSEA = 'NONE',
                 CTOWN = 'TEB',
                 CWATER = 'NONE,
                 LGARDEN = F
/
&NAM_TEB    NROAD_LAYER = 3,
            NROOF_LAYER = 3,
            NWALL_LAYER = 3
/

With :

&NAM_PGD_SCHEMES  CNATURE = 'ISBA',
                  LGARDEN = T
/

&NAM_TEB   LGREENROOF = T
/
```

10.1.3.2. Options for patches

```
&NAM_PGD_SCHEMES CNATURE = 'NONE',
                 CSEA = 'NONE',
                 CTOWN = 'TEB',
                 CWATER = 'NONE,
                 LGARDEN = F
/

With :

&NAM_PREP_TEB   CROAD_DIR = 'ORIE'
/

&NAM_TEB  NTEB_PATCH = 5
/
```

10.1.3.3. Options for BEM

```
&NAM_PGD_SCHEMES CNATURE = 'NONE',
                 CSEA = 'NONE',
                 CTOWN = 'TEB',
                 CWATER = 'NONE,
                 LGARDEN = F
/
&NAM_TEB     CBEM = 'BEM',
             CCOOL_COIL = 'DXCOIL',
             CHEAT_COIL = 'FINCAP',
             LAUTOSIZE = T
/
```

10.1.4. Options for the physiographic fields

10.1.4.1. Use of ECOCLIMAP

```
&NAM_FRAC  LECOCLIMAP=T
/

&NAM_COVER  YCOVER = 'ECOCLIMAP_II_EUROP',
            YCOVERFILETYPE = 'DIRECT'
/

&NAM_ECOCLIMAP2  LCLIM_LAI =F,
                 YIRRIG = 'NAME'
/
```

10.1.4.2. Physiographic fields given by user not interfering with the use of ECOCLIMAP

NAM_ZS

- With :

```
&NAM_PGD_SCHEMES  CSEA='SEAFLX'
/
```

NAM_SEABATHY

NAM_DATA_SEAFLUX

- With :

```
&NAM_PGD_SCHEMES  CNATURE='TSZ0'
/
```

NAM_DATA_TSZ0

- With :

```
&NAM_PGD_SCHEMES  CWATFLX='FLAKE'
/
```

NAM_DATA_FLAKE

10.1.4.3. Physiographic fields given by user interfering with the use of ECOCLIMAP

See:

- NAM_FRAC to define the fractions of tiles.
- NAM_DATA_ISBA to define the ISBA parameters.
- NAM_DATA_TEB to define the general TEB parameters.
- NAM_DATA_BEM to define the TEB-BEM parameters.
- NAM_DATA_TEB_GARDEN to define the TEB-VEG parameters.
- NAM_DATA_TEB_GREENROOF to define the TEB-VEG-GR parameters.

10.2. PREP step

10.2.1. Types of input PREP files

The uppest level to define a PREP input file is in namelist:

```
&NAM_PREP_SURF_ATM   CFILE='NAME'
                     CFILETYPE = 'MESONH'/'GRIB'/'ASCII'/'LFI'
                     CFILEPGD = 'NAME2'
                     CFILEPGDTYPE = 'MESONH'/'GRIB'/'ASCII'/'LFI'
/
```

Then, a specific file can be defined for each scheme :

```
&NAM_PREP_ISBA/WATFLX/FLAKE/TEB/SEAFLX   CFILE_ISBA/WATFLX/FLAKE/TEB/SEAFLX ='NAME'
                                         CTYPE = 'MESONH'/'GRIB'/'ASCII'/'LFI'
                                         CFILEPGD_ISBA/WATFLX/FLAKE/TEB/SEAFLX = 'NAME2
                                         CTYPEPGD = 'MESONH'/'GRIB'/'ASCII'/'LFI'
/
```

And still after, a specific file can be defined for each variable. Other types are possible: 'ASCLLV' for all and 'NETCDF' for TG in ISBA case.

10.2.2. Canopy scheme

Canopy is activated at PREP level, with a logical for each scheme:

```
&NAM_PREP_ISBA LISBA_CANOPY=T
/
```

&NAM_PREP_TEB LTEB_CANOPY=T /

&NAM_PREP_SEAFLUX LSEA_SBL=T /

&NAM_PREP_WATFLUX LWAT_SBL=T /

&NAM_PREP_FLAKE LWAT_SBL=T /

To be noted that there are rules for the choice of CROUGH functions of the use of CANOPY or not:

```
&NAM_SSOn   CROUGH=Z01D/Z04D/BE04/UNDE/NONE
/
```

&NAM_ISBAn CROUGH=Z01D/Z04D/BE04/UNDE/NONE /

CROUGH from NAM_SSOn is prioritary on this from NAM_ISBAn.

With CANOPY, only BE04 can be used and reciprocally, BE04 cannot be used without CANOPY.

If 'BE04' is defined at NAM_SSOn level, a general canopy (not depending on schemes) is activated.

10.2.3. Schemes options defined at PREP level

- For TEB:

```
&NAM_PREP_TEB  CROAD_DIR = 'UNIF'/'ORIE'
               CWALL_OPT='UNIF'/'TWO'
/
```

- For SEAFLUX:

```
&NAM_PREP_SEAFLUX  LOCEAN_MERCATOR =T,
                   LOCEAN_CURRENT=T,
                   XTIME_REL=100000.,
                   LCUR_REL=T,
                   LTS_REL=T,
                   LZERO_FLUX=T,
                   LCORR_FLUX=T,
                   XCORFLX=1.2,
                   LDIAPYC=T
/
```

Relaxation on TS occurs only if LCORR_FLUX=F.

10.2.4. Snow and carbon options

- SNOW:

For ISBA, GARDEN and GREENROOF, possible snow models are :

```
&NAM_PREP_ISBA/GARDEN/GREENROOF_SNOW CSNOW = 'D95','3-L','EBA','CRO'
/
```

For ROOF and ROAD, only CSNOW_ROOF='1-L' and CSNOW_ROAD='1-L' are allowed (NAM_PREP_TEB_SNOW). With CISBA='DIF', CSNOW cannot be 'D95' nor 'EBA'.

- CARBON:

```
&NAM_PREP_ISBA_CARBON CRESPSL='DEF'/'PRM'/'CNT'
/
```

CRESPSL is used if CPHOTO/='NON'. CRESPSL='CNT' is possible only with CPHOTO='NCB'. In GARDEN and GREENROOF, CRESPSL='DEF'.

10.3. OFFLINE step

10.3.1. Options for SEAFLUX and/or WATFLUX and/or FLAKE schemes

For SEAFLUX and FLAKE, the type of parameterization for sea fluxes can be:

```
&NAM_SEAFLUXn    CSEA_FLUX='DIRECT'/'ITERAT'/'COARE3'/'ECUME'
/
```

```
&NAM_FLAKEn  CFLK_FLUX = 'FLAKE'/'DEF '/'ECUME '
/
```

For SEAFLUX, WAFLUX and FLAKE, he type of albedo calculation can be:

```
&NAM_SEAFLUXn  CSEA_ALB = 'UNIF'/'TA96'/'MK10'
/
```

```
&NAM_WATFLUX/FLAKEn  CWAT/FLK_ALB = 'UNIF'/'TA96'
/
```

For SEAFLUX and FLAKE, if ECUME or COARE3 (only for SEAFLUX) is used, fluxes can be corrected according to precipitation, and if ECUME is used, according to heat flux :

```
&NAM_SEAFLUX/FLAKEn   LPRECIP=T (precipitation)
                      LPWEBB=T (heat flux)
/
```

For SEAFLUX, if ECUME of COEAR3 is used, fluxes can be corrected according to gustiness:

```
&NAM_SEAFLUXn  LPWG = T
/
```

For SEAFLUX and WATFLUX, monthly or annual SST can be interpolated to daily SST:

```
&NAM_SEAFLUX/WATFLUXn  CINTERPOL_SST/TS='ANNUAL'/'MONTH '/'NONE '
/
```

For the 1D ocean model, the SST can evolve following the tendency:

```
&NAM_SEAFLUXn  LPROGSST = T
/
```

10.3.2. Options for FLAKE scheme

The bottom sediments scheme of Flake can be used :

```
&NAM_FLAKEn   LSEDIMENTS = T
/
```

10.3.3. Options for TEB scheme

The z0h for roof & road can be:

```
&NAM_TEBn  CZ0H = 'MASC95'/'BRUT82'/'KAND07'
/
```

The BEM option for roof/wall outside convective coefficient can be

```
&NAM_TEBn  CCH_BEM = ''/'DOE-2'
/
```

10.3.4. Options for ISBA scheme

10.3.4.1. Main options

- C1 formulation for dry soils can be:

```
&NAM_ISBA   CISBA='2-L'/'3-L'
/
&NAM_ISBAn  CC1DRY='DEF'/GB93'
/
```

(impact in *soil.F90*)

- Type of thermal conductivity can be:

```
&NAM_ISBAn  CSCOND = 'NP89'/'PL98'
/
```

(impact in *soil.F90* and *soildif.F90*)

- Type of soil freezing-physics option can be:

```
&NAM_ISBA  CISBA= = '2-L'/'3-L'
/
&NAM_ISBAn  CSOILFRZ = 'DEF'/'LWT'
/
```

(impact in *isba_fluxes.F90*)

- Type of Mulch effects can be:

```
&NAM_ISBA  CISBA='DIF'
/
&NAM_ISBAn  CDIFSFCOND = 'DEF'/'MLCH'
/
```

(impact in *soildif.F90*)

- Type of turbulent exchanges over snow can be:

```
&NAM_PREP_ISBA_SNOW  CSNOW='3-L'/'CRO'
/
&NAM_ISBAn   CSNOWRES = 'DEF'/'RIL'
/
```

(impact in *snow3l.F90* and *snowcro.F90*)

- The type of albedo can be:

```
&NAM_ISBAn  CALBEDO = 'DRY'/'WET'/'MEAN'/'EVOL'/'CM13'
/
```

(impact in *soil_albedo.F90*, *albedo.F90* and *convert_patch_isba.F90*)

- The type of specific heat at surface can be:

```
&NAM_ISBAn  CCPSURF = 'DRY'/'HUM'
/
```

(impact in *drag.F90*)

- The drag can be activated in SBL scheme within the canopy:

```
&NAM_PREP_ISBA  LISBA_CANOPY = T
/
&NAM_ISBAn LCANOPY_DRAG = T
/
```

(impact in *coupling_isba_canopyn.F90*)

- Specific treatment can be applied over permanent snow/ice regions:

```
&NAM_ISBAn  LGLACIER = T
/
```

(impact in *hydro_snow.F90* is CISBA/='DIF' or CSNOW='D95', in *snow3l.F90* and *snowcro.F90* otherwise)

10.3.4.2. Hydrological options

- The type of subgrid runoff can be:

```
&NAM_SGH_ISBAn  CRUNOFF='WSAT'/'DT92'/'SGH'/'TOPD'
/
```

TOPD is the same as DT92 expect if LCOUPL_TOPD is activated. In this case, the fraction of saturated mesh from TOPMODEL is considered to calculate the ISBA runoff.

(impact in *hydro_sgh.F90*, *isba_sgh_update.F90*)

- The kind of regression for CRUNOFF=â SGHâ can be:

```
&NAM_SGH_ISBAn  CRUNOFF = 'SGH',
                CTOPREG = 'DEF'/'NON'
/
```

(impact in *init_top.F90*)

- The profile for Ksat can be:

```
&NAM_SGH_ISBAn  CKSAT = 'DEF'/'SGH'/'EXP'
/
```

(impact in *soil.F90*, *hydro_soil.F90*, *hydro_soildif.F90* and *compute_isba_parameters*)
CKSAT='EXP' only runs with CISBA='3-L' and allows to read a F parameter file coming from the coupling with TOPMODEL.

- The spatial distribution of rainfall intensity can be:

```
&NAM_SGH_ISBAn  CRAIN = 'DEF'/'SGH'
/
```

(impact in *hydro_sgh.F90* and *isba_sgh_update.F90*)

- The Horton runff due to water infiltration excess can be :

```
&NAM_SGH_ISBAn  CHORT='DEF'/'SGH'
/
```

(impact in *hydro_sgh.F90* and *hydro_soildif.F90*)

10.3.4.3. Other options

- To run the soil and wood carbon spinup:

see NAM_SPINUP_CARBn

(soil carbon spinup runs with CRESPSL='CNT' and wood carbon spinup with CPHOTO='NCB')

- To calculate z0 for trees from the LAI:

see NAM_TREEDRAG

- To activate soil deep soil characteristics:

see NAM_DEEPSOIL

(impact in *soil_heatdif.F90* if CISBA='DIF' and in {soil_temp_arp.F90} elsewhere is LTEMP_ARP).

- To compute irrigation on agricultural areas:

see NAM_AGRI

(impact in *vegetation_evol.F90*; runs only if CPHOTO='LAI'/'LST'/'NIT'/'NCB')

10.3.5. Options for CROCUS snow scheme

Can be activated:

```
&NAM_CROCUSn  LSNOWDRIFT = T
              LSNOWDRIFT_SUBLIM = T
```

(impact in *snowcro.F90*)

10.3.6. Options for SURF_ATM scheme

- The vertical shift can be unactivated:

```
&NAM_SURF_ATM  LVERTSHIFT=F
/
```

- The wind implicitation can be:

```
&NAM_SURF_ATM  CIMPLICIT_WIND = 'NEW'/'OLD'
/
```

- To unactivate subgrid orography effects on atmospheric forcing:

```
&NAM_SURF_ATM   LNOSOF = T
/
```

(impact in *drag.F90*)

- Coming from ALADIN or ARPEGE:

```
&NAM_SURF_ATM  LALDTHRES = T, (impact in wind_threshold.F90)
               LDRAG_COEF_ARP = T,  (impact in drag.F90, ice_sea_flux.F90, water_flux.F
               LALDZ0H = T, (impact in z0eff.F90)
               LRW_PRECIP = T,  (impact in coupling_surf_atmn.F90)
               LVZIUSTAR0 = T, (impact in water_flux.F90)
               LRRGUST_ARP = T, (ipact in drag.F90, ice_sea_flux.F90, water_flux.F90)
               LCPL_ARP = T, (impact in drag.F90, e_budget.F90)
               LQVNPLUS = T (impact in  e_budget.F90)
/
```

- To write a restart file:

```
&NAM_IO_OFFLINE   LRESTART = T
/
```

- To run the inquiry mode:

```
&NAM_IO_OFFLINE  LINQUIRE = T
/
```

- Not to compress FA diagnostic files:

```
&NAM_IO_OFFLINE  LDIAG_FA_NOCOMPACT = T
/
```

- Not to get the orography coming from the forcing files:

```
&NAM_IO_OFFLINE  LSET_FORC_ZS = T
/
```

- To change the names of output files at the end of the day:

```
&NAM_IO_OFFLINE  LOUT_TIMENAME = T
/
```

- For coherence between forcing Qair and calculated Qsat (Tair):

```
&NAM_IO_OFFLINE LLIMIT_QAIR = T
/
```

- To read forcing files in N times:

```
&NAM_IO_OFFLINE NB_READ_FORC = N
/
```

- To run the land use mechanism:

```
&NAM_IO_OFFLINE LLAND_USE = T
/
```

- To change algorithm of distribution for MPI parallelization:

```
&NAM_IO_OFFLINE  YALG_MPI = 'LIN'/'ADJ'/'TILL'/'TILA'
/
```

- To lower the number of points affected to the I/O processor in case of MPI parallelization:

```
&NAM_IO_OFFLINE  XIO_FRAC = value between 0 and 1
/
```

10.3.7. Options for the format files

Possible format files are:

```
&NAM_IO_OFFLINE   CFORCING_FILETYPE = 'NETCDF'/'ASCII'/'BINARY',
                  CSURF_FILETYPE = 'ASCII'/'LFI'/'FA',
                  CTIMESERIES_FILETYPE = 'NETCDF'/'ASCII'/'LFI'/'FA'/'BINARY'/'TEXTE'
/
```