



## Guide

# d'installation, de configuration et de compilation du modèle MOLOCH

Version 1.0

02 février 2026

Réalisé par :

Younoussa Adamou Sayri  
Rakiswende Thomas Bere  
Francesco Pasi  
Valerio Capecchi  
Vieri Tarchiani



Politecnico  
di Torino



Consiglio Nazionale delle Ricerche  
Istituto per la BioEconomia



Direction de la Météorologie  
Nationale  
Direction des Ressources en Eau  
Niger



## Table des matières

Introduction .....	3
I. Objectifs du guide.....	3
I. Etape1 installation du model .....	4
1) Configuration des répertoires : .....	4
2) Téléchargement des sources .....	5
3) Compilation de libaec,.....	5
4) Compilation d'ecCodes.....	5
5) Préparation de dimensions.inc.....	6
6) Compilation globale (PRE, MODEL, POST)	6
7) Déploiement .....	7
II. Etape 2 : paramétrisation et compilation: .....	8
1) Structure du Répertoire du Domaine .....	8
2) Configuration du Domaine (dimensions.inc & premoloch.inp) .....	8
A. Paramétrage de la grille (dimensions.inc).....	8
B. Configuration Géographique (premoloch.inp).....	9
3) Paramètres de Simulation (moloch.inp) .....	9
4) Lancement du premoloch .....	11
A. Recompilation après changement de domaine.....	11
Étape A : Création du lien symbolique.....	11
Étape B : Configuration du Makefile .....	11
Étape C : Lancez la séquence de nettoyage et de compilation toujours dans le même répertoire ou se trouve le make file: .....	12
5) Lancement du model .....	12
A. Prémoloch : .....	13
B. Moloch : .....	13
C. Postmoloch.....	14

## Introduction

Le projet **SLAPIS Sahel**, financé par l'Agence Italienne pour la Coopération au Développement (AICS), vise à renforcer la résilience du Niger et du Burkina Faso face aux risques hydrologiques. Sous l'égide du Polytechnique de Turin et de l'IBE-CNR, l'un des axes majeurs du projet consiste à améliorer les capacités de **prévision numérique du temps (PNT)**.

À cet effet, la DNM (Niger) et l'ANAM (Burkina Faso) ont déployé une chaîne de prévision articulée autour de deux modèles complémentaires :

- **WRF (Weather Research and Forecasting)** : un standard mondial pour la recherche et l'exploitation opérationnelle.
- **MOLOCH (MOdel LOCal H-coordinates)** : un modèle de pointe à haute résolution développé par l'ISAC-CNR.

Pour garantir la maîtrise de ces outils, deux experts nationaux ont bénéficié d'une bourse de recherche d'un an à Florence (IBE-CNR/LaMMA). Cette immersion vise à optimiser le l'exploitation opérationnelle des modèles et le paramétrage. À titre de soutien technique, des chaînes miroirs ont été configurées sur les serveurs du LaMMA pour servir de plateforme de formation et de facilité de back-up. De plus un guide d'utilisation a été préparée tant pour WRF que pour MOLOCH.

## I. Objectifs du guide

Ce manuel détaille le protocole standard d'installation et de configuration du modèle **MOLOCH** en environnement Linux. Pour simplifier ce processus complexe, un script d'automatisation nommé `compile_moloch.sh` a été développé. Ce document sert de support technique pour l'installation du modèle **MOLOCH**. Le code source et les instructions de déploiement sont accessibles sur :

- **Dépôt GitHub** : <https://github.com/slapis-git/install-moloch-home>
- **Documentation** : des fichiers `compile_moloch.sh` `posproc.sh` `postmoloch.sh` accompagnent le script pour guider l'utilisateur dans le mode automatisé.

## I. Etape 1 installation du model

### Prérequis

Vérifie que les compilateurs et outils sont installés :

Outil	Vérification	Installation
<b>gfortran</b>	<b>gfortran --version</b>	<b>sudo apt install gfortran</b>
<b>ifort</b>	<b>ifort --version</b>	<b>Intel Compiler</b>
<b>MPI (openmpi)</b>	<b>mpif90 --version</b>	<b>sudo apt install libopenmpi-dev openmpi-bin</b>
<b>CMake</b>	<b>cmake --version</b>	<b>sudo apt install cmake</b>
<b>Make</b>	<b>make --version</b>	<b>sudo apt install make</b>
<b>pip3</b>	<b>pip3 --version</b>	<b>sudo apt install python3-pip</b>

**sudo apt update && sudo apt install libcurl4-openssl-dev**

NB

Installer le cmake s'il est n'est pas installer car:

Pour la compilation de eccodes on a besoin de cmake.

Pour la compilation de rad\_ecmwf on a besoin de make. on a aussi besoin de la librairie netCDF (versions 3 or 4) et éventuellement HDF5.

### 1) Configuration des répertoires :

Le script crée une structure propre. Au lieu d'avoir des fichiers partout, il sépare : **src (Source) et bin (Binaire)**

- export BASE\_DIR="\$HOME/Models"
- export SRC\_DIR="\$BASE\_DIR/src"
- export BIN\_DIR="\$BASE\_DIR/bin"
- export DOMAIN\_DIR="\$HOME/moloch/domain/sahel\_d01"
- mkdir -p \$SRC\_DIR \$BIN\_DIR \$DOMAIN\_DIR
- cd \$SRC\_DIR
- echo "--- Début de l'installation automatique de MOLOCH ---"

## 2) Téléchargement des sources

- # 2. Téléchargement des sources (si absent)
- [ ! -d "libaec" ] && git clone https://github.com/erget/libaec.git
- [ ! -d "eccodes" ] && git clone https://github.com/ecmwf/eccodes.git
- [ ! -d "globo-bolam-moloch" ] && git clone https://gitlab.com/isac-meteo/globo-bolam-moloch.git
- echo "Téléchargement des librairies et du modèle..."
- git clone https://github.com/erget/libaec.git || true
- git clone https://github.com/ecmwf/eccodes.git || true
- git clone https://github.com/valcap/MOLOCH-LAMMA.git || true
- git clone https://gitlab.com/isac-meteo/globo-bolam-moloch.git || true
- **baec & ecCodes** : Ce sont les outils pour lire les fichiers météo mondiaux (format GRIB2). Sans eux, MOLOCH est "aveugle", il ne peut pas lire les données de départ (GFS ou IFS).
- **geo\_dataset** : Ce sont les cartes du monde (montagnes, lacs, forêts). C'est ce qui permet au modèle de savoir que le Sahel n'est pas l'Himalaya.
- **MOLOCH** : Le noyau qui calcule le vent, la pluie et la température.

## 3) Compilation de libaec,

C'est l'étape la plus longue. Le script transforme le texte (C++ et Fortran) en outils mathématiques rapides.

**Pourquoi libaec en premier ?** Parce que c'est un compresseur de données. ecCodes en a besoin pour décompresser les fichiers GRIB2 très lourds.

- echo "--- Étape 2 : Compilation de libaec ---"
- cd \$SRC\_DIR/libaec && mkdir -p build && cd build
- cmake -DCMAKE\_INSTALL\_PREFIX=\$BIN\_DIR/libaec ..
- make -j\$(nproc) install

## 4) Compilation d'ecCodes

```
echo "--- Étape 3 : Compilation d'ecCodes avec support CURL ---"
cd $SRC_DIR/eccodes && rm -rf build && mkdir -p build && cd build
cmake .. -DCMAKE_INSTALL_PREFIX=$BIN_DIR/eccodes \
-ENABLE_AEC=ON \
-DAEC_INCLUDE_DIR=$BIN_DIR/libaec/include \
-DAEC_LIBRARY=$BIN_DIR/libaec/lib/libaec.so \
-ENABLE_NETCDF=ON \
-ENABLE_FORTRAN=ON \
-DNetCDF_C_LIBRARY=/usr/lib/x86_64-linux-gnu/libnetcdf.so \
```

---

```
-DNetCDF_FORTRAN_LIBRARY=/usr/lib/x86_64-linux-gnu/libnetcdff.so \
-DCMAKE_EXE_LINKER_FLAGS="-lcurl" \
-DCMAKE_SHARED_LINKER_FLAGS="-lcurl"
make -j$(nproc) install
```

## 5) Préparation de dimensions.inc

- echo "--- Étape 4 : Configuration des dimensions ---"
- cd \$SRC\_DIR/globo-bolam-moloch/moloch
- cat <<EOF > dimensions.inc
- INTEGER, PARAMETER :: gnlon = 102
- INTEGER, PARAMETER :: gnlat = 102
- INTEGER, PARAMETER :: nlev = 50
- INTEGER, PARAMETER :: nlevg = 50
- INTEGER, PARAMETER :: nsoil = 7
- INTEGER, PARAMETER :: nprocsx = 1
- INTEGER, PARAMETER :: nprocsy = 1
- EOF
- cp dimensions.inc \$SRC\_DIR/globo-bolam-moloch/sources/moloch/
- cp dimensions.inc \$SRC\_DIR/globo-bolam-moloch/sources/common

## 6) Compilation globale (PRE, MODEL, POST)

- echo "--- Étape 5 : Compilation de la chaîne MOLOCH ---"
- export FC=gfortran
- export FC\_MPI=mpif90
- export USE\_MPI=YES
- export DIR\_MOL=\$SRC\_DIR/globo-bolam-moloch/sources/moloch
- export DIR\_COM=\$SRC\_DIR/globo-bolam-moloch/sources/common
- export LIB\_ECC="-L\$BIN\_DIR/eccodes/lib -leccodes -leccodes\_f90"
- export INC\_ECC="-I\$BIN\_DIR/eccodes/include"
- # On définit les drapeaux de compilation pour corriger les erreurs de lignes trop longues
- export FLAGS\_FIX="-O2 -I. \$INC\_ECC -ffree-line-length-none -fallow-argument-mismatch -w"
- 
- for COMP in executable\_premodel executable\_model executable\_postmodel
- do
- echo ">> Compilation de \$COMP"
- cd \$SRC\_DIR/globo-bolam-moloch/moloch/\$COMP
- 
- # Correction des Makefiles
- sed -i "s|\\$HOME|sources/moloch|\$DIR\_MOL|g" Makefile

- sed -i "s|\$(HOME)/sources/common|\$DIR\_COM|g" Makefile
- 
- cp \$SRC\_DIR/globo-bolam-moloch/moloch/dimensions.inc .
- 
- make clean || true
- 
- make FC=\$FC FC\_MPI=\$FC\_MPI USE\_MPI=\$USE\_MPI \
 DIR\_SOURCES\_MOLOCH=\$DIR\_MOL \
 DIR\_SOURCES\_COMMON=\$DIR\_COM \
 LDFLAGS="\$LIB\_ECC" \
 FCFLAGS="\$FLAGS\_FIX" \
 FCFLAGS\_MPI="\$FLAGS\_FIX -Dmpi" -j\$(nproc)
- done

## 7) Déploiement

- echo "--- Étape 6 : Mise en service du domaine ---"
- cd \$DOMAIN\_DIR
- ln -sf \$SRC\_DIR/globo-bolam-moloch/moloch/executable\_premodel/premoloch .
- ln -sf \$SRC\_DIR/globo-bolam-moloch/moloch/executable\_model/moloch .
- ln -sf \$SRC\_DIR/globo-bolam-moloch/moloch/executable\_postmodel/postmoloch .
- cp \$DIR\_MOL/premoloch\_an\_example.inp ./premoloch.inp
- cp \$DIR\_MOL/moloch\_an\_example.inp ./moloch.inp
- cp \$DIR\_MOL/postmoloch\_an\_example.inp ./postmoloch.inp
- # AJOUT : Copie de sauvegarde du fichier dimensions pour référence
- cp \$SRC\_DIR/globo-bolam-moloch/moloch/dimensions.inc ./dimensions.inc
- echo "=====
- echo " INSTALLATION TERMINÉE AVEC SUCCÈS ! "
- echo " Dossier de travail : \$DOMAIN\_DIR "
- echo "=====

**Préparer les données statiques: pour télécharger les données statiques on utilise le lien ci-dessous :**

# Exemple pour geo\_dataset

cd \$HOME/src

- wget -r -np [https://www.isac.cnr.it/dinamica/geo\\_dataset/](https://www.isac.cnr.it/dinamica/geo_dataset/)
- tar -xvf geo\_dataset.tar.gz # ou le fichier tar que tu as téléchargé ###pour décomposé

on obtient par exemple: \$HOME/src/geo\_dataset/

```

├── orography/
├── landuse/
├── soil/
├── rivers/
└── masks/

```

| ...

## II. Etape 2 : paramétrisation et compilation:

### 1) Structure du Répertoire du Domaine

Dans votre dossier de domaine (ex: ~/moloch/domain/sahel\_d01), vous devez impérativement retrouver les fichiers suivants pour assurer le bon fonctionnement des pré-traitements et de la simulation :

- **dimensions.inc** : Paramètres de la grille et parallélisation.
- **premoloch.inp** : Configuration géographique et format d'entrée.
- **moloch.inp** : Paramètres de temps et physique du modèle.
- **postmoloch.inp** : Configuration des sorties.
- **grib\_sample.inp** : Fichier contenant les liens vers la bibliothèque ecCodes.

```
-rw-rw-r-- 1 slapis slapis 108 Jan 15 14:08 dimensions.inc
-rw-rw-r-- 1 slapis slapis 115 Jan 15 14:08 grib_sample.inp
-rw-rw-r-- 1 slapis slapis 2485 Jan 15 14:08 moloch.inp
-rw-rw-r-- 1 slapis slapis 1512 Jan 15 14:08 mpd.hosts
-rw-rw-r-- 1 slapis slapis 7509 Jan 15 14:08 postmoloch.inp
-rw-rw-r-- 1 slapis slapis 3588 Jan 15 14:08 premoloch.inp
```

**Le mpd.hosts** : elle est spécifique au serveur

### 2) Configuration du Domaine (dimensions.inc & premoloch.inp)

#### A. Paramétrage de la grille (dimensions.inc)

Ce fichier définit la résolution et la puissance de calcul allouée. Les valeurs doivent être cohérentes avec votre capacité machine :

- **gnlon, gnlat** : Nombre total de points en longitude et latitude.
- **nprocx, nprocy** : Nombre de processeurs. *Exemple : pour 120 processeurs, utilisez 10 x 12.*
- **nlev** : Nombre de niveaux verticaux (ex: 50).
- **nlevg** : Nombre de niveaux de sol (ex: 13).

**Formule de vérification** : Pour que le calcul soit valide, le nombre de points moins 2 doit être divisible par le nombre de processeurs. *Exemple : (gnlon - 2) / nprocx doit être un nombre pair (ex: 800/10 = 80), (gnlat - 2) / nprocy doit être un nombre pair .*

```
integer, parameter :: gnlon=802, gnlat=482, nlev=50, nlevg=13
integer, parameter :: nprocxsx=10, nprocpsy=12
```

## B. Configuration Géographique (premoloch.inp)

Vous devez spécifier ici le format des données et la résolution réelle :

- **INPUT\_FORMAT** : Utilisez GRIB2 pour les données modernes.
- **DLON, DLAT** : Résolution en degrés.
  - *Exemple pour 9km* : \$9 / 110,54 \approxeq
  - x 0,082.
  - *Exemple pour 3km* : \$3 / 110 \approx 0,027.
- **ALON0, ALAT0** : Coordonnées du coin inférieur gauche du domaine (expl -2, -3)

```
&PARAM_PREMOLOCH
  INPUT_FORMAT=GRIB2, INPUT_FILE_STD=2,
  NLON=802, NLAT=482, NLEV=50,
  DLON=0.082, DLAT=0.08,
  X0=0.0, Y0= 0.0,
  ALON0=-22.4, ALAT0=-4.8,
```

! ----- MODEL GRID PARAMETERS -----

- ! NLON, NLAT, NLEV: dimensions of the Moloch grid
- ! NLEVG: number of Moloch soil levels
- ! DLON, DLAT: Moloch grid resolution in degrees
- ! X0, Y0: geographical coordinates in deg of the centre of rotation (point "T" of the Arakawa grid)
- ! For non rotated grid, define X0=Y0=0.
- ! If X0<-180 or X0>360 or Y0<-90 or Y0>90, then X0 and Y0 are redefined equal to input values.
- ! ALON0, ALAT0: rotated coord. in deg of the SW corner (point "V") of the model grid.
- ! B0: stretching parameter ( $0.2 > B0 \leq 1$ ), defining the Moloch hybrid vertical coordinate:
- ! reducing B0 from 1. decreases the height of the lowest model level above the surface.
- ! In case of non rotated grid, ALON0, ALAT0 must define the true coordinates of the SW corner
- ! and DLON should be set to DLAT/COS(ALAT0) where ALAT0 is a mean latitude of the model grid.
- ! SLT(10): thickness of soil layers (m).
- ! If any SLT(:)<0., then NLEVG and SLT() are redefined equal to input values.

## 3) Paramètres de Simulation (moloch.inp)

Ce fichier gère le "temps" et la physique. Voici les paramètres clés extraits de votre configuration :

Paramètre	Description	Conseil de réglage
<b>dtstep</b>	Pas de temps d'intégration (secondes)	Environ 10 à 20 fois la résolution (km).
<b>hrun</b>	Durée totale de la simulation (heures)	À modifier selon vos besoins de prévision.
<b>nradm</b>	Schéma de radiation	<b>2</b> pour combiner Geleyn et ECMWF.
<b>hbound</b>	Fréquence des conditions aux limites	<b>3</b> si vous utilisez des fichiers GFS toutes les 3h.
<b>nlconv</b>	Paramétrage de la convection	.true. pour l'activer.
<b>nlana</b>	Sauvegarde de l'état initial	.true. pour inclure l'état initial en sortie.

```
.....
model
!tstep = 120.,
!adv = 3,
!sound = 6,
!nbl = 8,
!nadm = 2,
!hrun = 24,
!hist = 1,
!hist_full_res = -24,
!swshf = 60.,
!hbound = 3,
!diag = 1.,
!hfr=1,
!srad = 720.,
!filter = 5.,
!top = 0.92,
!mic2=.false.,
!lana=.true.,
!lconv=.true.,
!lradar=.false.,
!ltwice=.false.,
!ldiff=.false.,
!climate=.false.,
!rst=.false.,
!rst= 240.,
!back=-0.1, hspray=-0.50, nxspray= 70, nyspray= 70, nzspray=45, xsorg=15.200, ysorg=37.24 ! sicilia
end
```

- **dtstep** integration time step (in seconds)
- **! adv** number of advection steps in dtstep sec.
- **! sound** number of steps for sound waves
- **! nbl** no. of lateral boundary lines
- **! nradm** radiation scheme flag: 0 - no radiation,
- **!** 1 - Geleyn only every srad seconds
- **!** 2 - Geleyn and ECMWF combined
- **! hrun** forecast duration in hours
- **! hist** interval (in hours) between two writings of model output (mhf)
- **! hist\_full\_res** interval (in hours) between two writings of model output (mhf) in full horizontal resolution
- **! hbound** interval (in hours) between two boundary conditions

- ! hdiag interval (in hours) between two model diagnostics printouts
- ! srad interval (in seconds) between two calls of radiation parameterization
- ! mswshf interval (in minutes) between two writings of surface fields (shf)
- ! mslfilter period (in minutes) of filtering of turbulent coefficient of the surface layer exchanging
- ! htop fraction of vertical column where microphys. and turbul. are active (<= 1.)
- ! nlmic2 if true, 2-moment microphysics
- ! nlbfixed if true, fixed boundary conditions
- ! nlana if true, saves initial condition in output (mhf)
- ! nltwice if true, double computation of vertical advection per timestep
- ! nlhdiff if true, computes horizontal turbulent diffusion
- ! nlconv if true, parameterization of convection is switched on
- ! nlord if true, orographic drag (block+GW)
- ! nlradar if true, simulation of radar reflectivity
- ! nlwshf if true, writes surface fields (shf)
- ! nlclimate if true, very long run options are active (name of input/output files)
- ! nlrst, if true, restart input files be read at the run initial
- ! mhfr 1: full mhf resolution; 2: half mhf resolution
- ! hrst interval (in hours) between two restart files writing
- ! hback interval (in hours) between two outputs of trajectory file (if negative, file is not written)
- ! hspray interval (in hours) between two outputs of SPRAY file (if negative, file is not written)

## 4) Lancement du premoloch

### A. Recompilation après changement de domaine

Chaque modification de dimensions.inc nécessite un lien symbolique vers le code source avant de compiler :

#### Étape A : Cr éation du lien symbolique

Le code source situ é dans le r épertoire de calcul doit pointer vers les param ètres de votre domaine (ex: Sahel).

- cd /home/slapis/moloch/src/v25.2.1/moloch
- # On remplace le fichier par d éfaut par un lien vers votre domaine
- ln -sf ../../domain/sahel\_d01/dimensions.inc .

#### Étape B : Configuration du Makefile

Revenez dans le r épertoire racine des sources pour configurer les options du mod èle :

- cd /home/slapis/moloch/src/v25.2.1
- vi Makefile,

**Modifiez le MAkefile comme suit si ça n'a pas été faite:**

- **BOLAM = NO**
- **MOLOCH = YES**
- **GLOBO = NO**
- **GRAPHICS\_PLGRIBFA = NO**

**Étape C : Lancez la séquence de nettoyage et de compilation toujours dans le même répertoire ou se trouve le make file:**

- make clean
- make

En résumé :

Fichier	Utilité
<b>premodel</b>	Exécutable pour préparer les données géographiques et initiales.
<b>model</b>	L'exécutable principal du simulateur MOLOCH.
<b>postmodel</b>	Exécutable pour générer les fichiers GRIB finaux.
<b>convert_shf_to_grib2</b>	Exécutable pour générer les fichiers GRIB finaux en surface
<b>premoloch.inp</b>	Paramètres de la grille et de la résolution.
<b>moloch.inp</b>	Paramètres physiques et pas de temps (dtstep).
<b>postmoloch.inp</b>	Options de sortie et variables à extraire.

Ensuite copier tous les 4 exécutable ( premodel, model, postmodel, convert\_shf\_to\_grib) dans le domaines.

## 5) Lancement du model

Pour lancer le model, il faut créer un répertoire run ou vas tourner le modèle, dans ce répertoire run tu crée le différent lien vers ce répertoire :

Par exemple :

- **cd ~/moloch/**
- **mkdir -p run\_sahel**
- **cd run\_sahel**

- a. crée le lien du répertoire run vers les données statiques c'est-à-dire On va lier tout ce dont moloch a besoin pour tourner sans copier les fichiers,

b. crée le lien du répertoire run vers les données gfs par exemple :

- ln -sf ../../data/input/gfs/2025072400/gfs.00z.grb2.f000 grib\_001,
- ln -sf ../../data/input/gfs/2025072400/gfs.00z.grb2.f001 grib\_002
- ln -sf ../../data/input/gfs/2025072400/gfs.00z.grb2.f002 grib\_003

Ainsi de suite sauf si tu crées une boucle.

c. crée le lien du répertoire run vers le Domain (**dimensions.inc**, **premoloch.inp**, **moloch.inp**, **postmoloch.inp**, **grib\_sample.inp**)

d. crée le lien du répertoire run vers les exécutable ou faire une copier (executable\_convert\_shf\_to\_grib2, executable\_meteograms, executable\_model, executable\_postmodel, executable\_premodel).

### A. Prémoloch :

Dans le répertoire run, lancez **prémoloch** avec la commande

- mpirun -np 1 ./premoloch

Cette étape lit tous les fichiers GRIB et génère les fichiers nécessaires pour préparer le lancement du modèle. Par exemple :

- Output mhf file input\_atm\_09.mhf written,
- Output mhf file input\_soil\_09.mhf written

### B. Moloch :

Une fois le prémoloch exécuté, on peut lancer le modèle principal en parallèle selon le nombre de processeurs disponibles sur le HPC. Par exemple :

- mpirun -np 120 ./moloch, Ici, 120 correspond au nombre de processeurs (Nproc).

Il arrive parfois qu'au moment du lancement, le système demande Open MPI. Dans ce cas :

- Vérifiez les modules disponibles :
- module avail openmpi
- Chargez le module correspondant :
- module load NOM\_DU\_MODULE

Après cela, relancez : mpirun -np 120 ./moloch

Le modèle commencera alors à générer les fichiers de sortie, par exemple :

- Output written on file moloch\_atm\_008.mhf
- Output written on file moloch\_soil\_008.mhf

## C. Postmoloch

Pour comprendre le fonctionnement du post-Moloch, il faut savoir qu'après la compilation du modèle, on obtient deux types de fichiers :

- Les fichiers .shf, qui contiennent uniquement les données de surface 2D.
- Les fichiers .mhf, qui contiennent les données verticales 3D.

Afin de traiter ces deux types de fichiers, nous avons mis en place **deux scripts distincts** pour faciliter la compréhension du post-Moloch. Avant d'exécuter ces scripts, il est nécessaire de modifier le fichier **postmoloch.inp** afin d'activer ou désactiver les paramètres souhaités. Dans ce fichier, la valeur **1** signifie que le paramètre est inclus, tandis que **0** indique qu'il est exclu.

Ci-dessous figurent les paramètres relatifs aux variables de surface.

ISFLAG(01) = 0, ! orography	ISFLAG(32) = 0, ! Temper at 0.5 m above surface
ISFLAG(02) = 1, ! M.S.T. pressure	ISFLAG(33) = 0, ! Rel. hum. at 0.5 m above surface
ISFLAG(03) = 1, ! Total precipitation	ISFLAG(34) = 0, ! Temper. at 0.05 m above surface
ISFLAG(04) = 0, ! Snow fall	ISFLAG(35) = 0, ! Ground temp. at 5 cm depth
ISFLAG(05) = 1, ! Wind at 10 m	ISFLAG(36) = 0, ! Ground temp. at 10 cm depth
ISFLAG(06) = 1, ! Temper. at 2 m	ISFLAG(37) = 0, ! Ground temp. at 20 cm depth
ISFLAG(07) = 1, ! Specific hum. at 2 m	ISFLAG(38) = 0, ! Ground temp. at 50 cm depth
ISFLAG(08) = 1, ! Relative hum. at 2 m	ISFLAG(39) = 0, ! Ground temp. at 100 cm depth
ISFLAG(09) = 0, ! Ground water lev. 1	ISFLAG(40) = 0, ! Albedo
ISFLAG(10) = 0, ! Ground water lev. 3	ISFLAG(41) = 0, ! Emissivity (broadband)
ISFLAG(11) = 0, ! Ground water lev. 5	ISFLAG(42) = 0, ! Emissivity (window)
ISFLAG(12) = 0, ! Ground water lev. 7	ISFLAG(43) = 0, ! Cumulated short wave radiative flux
ISFLAG(13) = 1, ! Skin temperature	ISFLAG(44) = 0, ! Cumulated long wave radiative flux
ISFLAG(14) = 0, ! Ground temp. lev. 1	ISFLAG(45) = 0, ! cumulated sensible heat flux
ISFLAG(15) = 0, ! Ground temp. lev. 3	ISFLAG(46) = 0, ! Cumulated latent heat flux
ISFLAG(16) = 0, ! Ground temp. lev. 5	ISFLAG(47) = 1, ! T min 2 m
ISFLAG(17) = 0, ! Ground temp. lev. 7	ISFLAG(48) = 1, ! T max 2 m
ISFLAG(18) = 0, ! Flux of sensible heat	ISFLAG(49) = 1, ! Wind speed max 10 m
ISFLAG(19) = 0, ! Flux of latent heat	ISFLAG(50) = 1, ! Lifted index
ISFLAG(20) = 0, ! Temperature at lowest atm. level	ISFLAG(51) = 0, ! Sea ice fraction
ISFLAG(21) = 0, ! Equiv. pot. temperature at lowest atm. level	ISFLAG(52) = 0, ! Sea ice thickness
ISFLAG(22) = 0, ! Wind at lowest atm. level	ISFLAG(53) = 0, ! Dew point temperature at 2 m
ISFLAG(23) = 0, ! Spec. hum. at lowest atm. level	ISFLAG(54) = 1, ! Integrated Water Vapour
ISFLAG(24) = 1, ! Total cloud cover	ISFLAG(55) = 1, ! CIN
ISFLAG(25) = 0, ! High cloud cover	ISFLAG(56) = 0, ! PBL height
ISFLAG(26) = 0, ! Middle cloud cover	ISFLAG(57) = 0, ! Wind gust
ISFLAG(27) = 0, ! Low cloud cover	ISFLAG(58) = 0, ! Level of 0 c
ISFLAG(28) = 0, ! Snow cover	ISFLAG(59) = 0, ! Snow cover products
ISFLAG(29) = 0, ! Run-off	ISFLAG(60) = 0, ! Lapse rate in lower troposphere
ISFLAG(30) = 1, ! CAPE	ISFLAG(61) = 0, ! Height of snow fall limit (m)
ISFLAG(31) = 0, ! Skin specific humidity	ISFLAG(62) = 0, ! Height of snow fall limit (m), fist guess
	ISFLAG(63) = 1, ! Surface Pressure
	ISFLAG(64:70) = 0, 0, 0, 0, 0, 0, 0,
	ISFLAG(71:80) = 0, 0, 0, 0, 0, 0, 0, 0, 0,

Ci-dessous les paramètres de niveaux :

	GPH	T	U, V	Q	RH	W	THE	CWI/PV	
IPFLAG0(001:008)	= 1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1000 HPA
IPFLAG0(009:016)	= 1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	950
IPFLAG0(017:024)	= 1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	900
IPFLAG0(025:032)	= 1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	850
IPFLAG0(033:040)	= 0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	800
IPFLAG0(041:048)	= 0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	750
IPFLAG0(049:056)	= 1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	700
IPFLAG0(057:064)	= 0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	650
IPFLAG0(065:072)	= 1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	600
IPFLAG0(073:080)	= 0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	550
IPFLAG0(081:088)	= 1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	500
IPFLAG0(089:096)	= 0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	450
IPFLAG0(097:104)	= 0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	400
IPFLAG0(105:112)	= 0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	350
IPFLAG0(113:120)	= 1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	1, 1, 1, 0,	300
IPFLAG0(121:128)	= 0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	250
IPFLAG0(129:136)	= 0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	200
IPFLAG0(137:144)	= 0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	150
IPFLAG0(145:152)	= 0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	100
IPFLAG0(153:160)	= 0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	0, 0, 0, 0,	50

Après avoir modifié le fichier postmoloch.inp, nous sommes prêts à lancer le post-traitement Moloch. Pour cela, nous avons développé un premier script, postmoloch.sh, dédié au traitement des fichiers .mhf, lesquels contiennent les données verticales 3D ainsi que certaines variables de surface :

### Le script automatise postmoloch.sh

- `#!/bin/bash`
- `# Post-traitement MHF → GRIB2, nettoyage complet des valeurs invalides`
- `OUT_DIR="post_results"`
- `mkdir -p "$OUT_DIR"`
- `for atm_file in moloch_atm_*.mhf; do`
- `num=$(echo "$atm_file" | sed -E 's/.*_(\d{1,2})\.mhf/\1/')`
- `soil_file="moloch_soil_${num}.mhf"`
- `if [ ! -f "$soil_file" ]; then`
- `echo "⚠️ Fichier SOIL manquant pour $atm_file, passage au suivant."`
- `continue`
- `fi`
- `echo "-----"`
- `echo "Traitement MHF n°$num :"`
- `echo "ATM : $atm_file"`
- `echo "SOIL: $soil_file"`
- `echo "-----"`
- `# Créer les liens pour postmoloch`
- `ln -svf $atm_file moloch_atm.mhf`
- `ln -svf $soil_file moloch_soil.mhf`
- `# Lancer postmoloch`
- `./postmoloch`
- `wait`
- `sleep 1`

---

```

• # docer les GRIB2 générés
• grib_files=$(ls moloch_*.grib2 2>/dev/null)
• for gf in $grib_files; do
•   mv "$gf" "$OUT_DIR/${num}_${gf}"
• done
• echo "Fichiers GRIB2 déplacés dans $OUT_DIR."
• # Supprimer les temporaires
• rm -f tmp_atm.mhf tmp_soil.mhf
• done
• echo "-----"
• echo "Traitement terminé. Tous les GRIB2 sont dans : $OUT_DIR"

```

Ensuite, nous avons utilisé un programme secondaire appelé **postproc.sh**, qui lit les fichiers .shf (moloch\_\*.shf), c'est-à-dire les données historiques de surface (variables comme : température à 2 m, vent 10 m U/V, humidité relative à 2 m, précipitations, pression de surface, etc.) et les convertit en **GRIB2**, le format standard des fichiers météorologiques.

Les fichiers GRIB2 générés sont stockés dans le répertoire post\_results.

Le traitement se déroule en plusieurs étapes :

1. Configurer **ECCODES** et ses templates.
2. Créer le répertoire de sortie post\_results.
3. Parcourir tous les fichiers .shf de Moloch.
4. Pour chaque fichier :
  - Créer un lien symbolique input.shf.
  - Lancer la conversion en GRIB2.
  - Déplacer les fichiers GRIB2 générés dans post\_results avec un nom clair et structuré.
5. Afficher des messages clairs pour suivre l'avancement du traitement.

Le script automatisé du posproc.sh est le suivant :

```

• #script de post-traitement Moloch : conversion SHF → GRIB2
• #
• # Répertoire ECCODES
• export DIR_GRIB_LIB=/cluster/sources/eccodes/2.14.1/eccodes-2.14.1-Source
• CURDIR=`pwd`
• # Créer le répertoire et le lien vers les samples ECCODES si nécessaire
• mkdir -p "$DIR_GRIB_LIB/share/eccodes"
• if [ ! -e "$CURDIR/samples" ]; then
•   ln -s "$DIR_GRIB_LIB/samples" .
• fi
• # Répertoire de sortie pour les GRIB2
• OUT_DIR="post_results"
• mkdir -p "$OUT_DIR"
• # Boucle sur tous les fichiers moloch_*.shf
• for shf_file in moloch_*.shf; do

```

- # Nom de base du fichier
- **base\_name=\$(basename "\$shf\_file" .shf)**
- echo "-----"
- echo "Traitement du fichier : \$shf\_file"
- echo "-----"
- # Créer un lien symbolique vers input.shf pour le convertisseur
- ln -sf "\$shf\_file" input.shf
- # Lancer la conversion SHF → GRIB2
- ./convert\_shf\_to\_grib2
- # Vérifier si des fichiers GRIB2 ont été créés
- grib\_files=\$(ls moloch\_\*.grib2 2>/dev/null)
- if [ -n "\$grib\_files" ]; then
- for gf in \$grib\_files; do
- mv "\$gf" "\$OUT\_DIR/\${base\_name}\_\$gf"
- done
- echo "Fichiers GRIB2 déplacés dans \$OUT\_DIR."
- else
- echo "⚠ Aucun fichier GRIB2 créé pour \$shf\_file"
- fi
- done
- echo "-----"
- echo "Traitement terminé. Tous les GRIB2 sont dans : \$OUT\_DIR"

Enfin les résultats (fichier grib 2) sont enregistrés dans un dossier post\_resultats.