CTSM@UiO Documentation

Release 0.1

Anne Fouilloux

Sep 24, 2019

Contents:

Note: This documentation is not meant to be static and is a collaborative effort. If you spot anything wrong or would like to add information to the documentation, follow these guidelines

CHAPTER 1

Get CTSM

1.1 Get CTSM for running (no source changes)

If you are a novice, that's where you should start. You get the source code and run it as is.

NordicESMhub maintains a CTSM repository with all the configuration files for running on machines in the Nordics. For now we support:

- abel (UiO, Norway)

If you machine is not in the list and would like we support it, please contact us.

1.1.1 How to get CTSM

CLONE from GitHub (run only first time, or to update clm/ctsm version), in home folder

```
git clone -b release-clm5.0 https://github.com/NordicESMhub/ctsm.git
```

LOAD necessary modules in .bashrc (only first time)

```
emacs .bashrc &
```

On abel, add these three modules

```
module load python2/2.7.10
module load cesm
module load git
```

save and close

CHECK loaded modules

```
module list
```

1.1.2 How to get a specific branch

1.1.3 Which branch do I run?

1.2 Get CTSM for modifying the source code

Create your own branches before continuing, in ~/ctsm AND in ~ctsm/cime. Do this for FATES or any other modules in addition.

```
git branch <username_clm5.0.12>   # useful for remembering version, name according to␣
↪function e.g. username_cime_clm5.0.12 and username_fates_clm5.0.12
git checkout  <username_clm5.0.12>
git branch # to verify that you are on the branch you just created
```

When you need to do your own development.

CHAPTER 2

Setup CTSM

2.1 Accounting

For running CTSM, you usually need to have access to a High-Performance Computer.

If you are working in Norway and at UiO, you can use abel and need to be part of an active account:

- Notur or geofag

To check which project you can use:

```
projects
```

This will return something like:

```
geofag
nn1000k
```

In the example above, two projects can be used (geofag and nn1000k). Then make sure you choose the right project when running CTSM.

CHAPTER 3

Run CTSM

NB! this example is connected to project nn2806k (for your own project, change the project code. To see available projects and resources, use cost -p):

```
export CESM_ACCOUNT=nn2806k
```

3.1 Run your very first CTSM case

LOAD externals of CTSM (FATES and so on; only necessary first time), in folder ~/ctsm. If you are updating FATES go here first: (https://github.com/NordicESMhub/ctsm-dev/blob/master/Updating_FATES.md)

```
./manage_externals/checkout_externals
```

navigate to ~/ctsm/cime/scripts/

3.2 Inputdata

(only first time or whenever it disappears in your workdir i.e. 45 days)

```
cd ~/ctsm/cime/scripts
./link_dirtree $CESM_DATA /work/users/$USER/inputdata
```

3.3 Make a case

```
./create_newcase --case ~/cases/I2000Clm50BgcCruGs  --compset I2000Clm50BgcCruGs   --
↪res f19_g17 --machine abel --run-unsupported --project $CESM_ACCOUNT
```

navigate to ~/cases/I2000Clm50BgcCruGs

3.3.1 1) check the configuration

```
./xmlquery --l #(--l list --f file)
```

eg

```
./xmlquery STOP_OPTION
```

3.3.2 2) Change configuration

- For instance, to change the duration of a simulation to 5 days:

```
./xmlchange STOP_OPTION=ndays #(nyears, nmonths)
./xmlchange STOP_N=5 #(then 5 days)
```

or edit the xml files is another way to change these parameters (**not recommended**).

3.3.3 3) setup case

```
./case.setup  #(--reset)
```

3.3.4 4) edit user_nl_clm

add this below

```
hist_mfilt=5 #(number of output files)
hist_nhtfrq=-24 #(means daily outputs)
```

hist_mfilt allows you to specify the number of output files and *hist_nhtfrq* the frequency; here *-24* means daily outputs.

3.3.5 5) case build

```
./case.build
```

Remark: if your build fails or if you make changes and need to rebuild, make sure you clean the previous build:

```
./case.build --clean
```

3.3.6 6) run case

```
./case.submit
```

3.4 Run fates

NB! Fates is not automatically checked out with the latest version (as it is still under development), and this has to be done manually.

Follow https://github.com/NordicESMhub/ctsm-dev/blob/master/Updating_FATES (based on https://github.uio.no/huit/clm5.0_notes/issues/26 and https://github.com/ESCOMP/ctsm/wiki/Protocols-on-updating-FATES-within-CTSM)

```
./create_newcase --case ../../../ctsm_cases/fates_f19_g17 --compset 2000_DATM%GSWP3v1_
↪CLM50%FATES_SICE_SOCN_MOSART_SGLC_SWAV --res f19_g17 --machine abel --run-
↪unsupported --project $CESM_ACCOUNT
```

3.5 Run a single cell case

CLM supports running using single-point or regional datasets that are customized to a particular region.

In the section below we show you how to run ready to use single-point configurations (out of the box) and then show you how to create your own dataset for any location of your choice.

3.5.1 Out of the box

To run for the Brazil test site do the following:

```
export CESM_ACCOUNT=nn2806k

./create_newcase -case ~/cases/testSPDATASET -res 1x1_brazil -compset I2000Clm50SpGs ␣
↪--machine abel --run-unsupported --project $CESM_ACCOUNT
```

Remark: make sure you set **CESM_ACCOUNT** to your project. Customized ~~~~~~~~~~~

- Step-1:
- Step-2: For atmospheric forcing (own atm forcing and surface data) (include scripts)

```
# 0) mkdir ~/single_cell_experiment #Create a directory in your home directory calles
↪"single_cell_experiment".
# 0) cd ~/single_cell_experiment
#Then put all the scripts below in "single_cell_experiment".
# 1) MAIN SCRIPT : call it as you wish (run_clm_single.sh)
# 2) regridbatch.sh
# 3) run_locality_append.sh
# 4) stream_file_1901_to_2014
# 5) prepare_atm_forcing_data_CRUNCEP_1901-1920.ncl # this file must be duplicated␣
↪and modified to fit the dates and the forcing (gswp or cru)
```

##_____MAIN_SCRIPT___## #!/bin/sh

path1="~/single_cell_experiment" #main directory path path2="~/ctsm" #clm path path3="/work/users/marlam/inputdata" #inputdata path path4="~/cases" #directory where the case will be run

#For single cell only compsets with SGLC work!! compset0=I2000Clm50BgcCruGs #2000_DATM%CRUv7_CLM50%BGC_SICE_SOCN_MOSART_SGLC_SWAV

#If you want to use CRU as atm forcing : put these files in "single_cell_directory and adapt them. #This process must be separated in periods shorter than 50 years prepare_atm_data1="prepare_atm_forcing_data_CRUNCEP_1901-1920.ncl" prepare_atm_data2="prepare_atm_forcing_data_CRUNCEP_1921-1955.ncl" prepare_atm_data3="prepare_atm_forcing_data_CRUNCEP_1956-1990.ncl" prepare_atm_data4="prepare_atm_forcing_data_CRUNC
2016.ncl"

#If you want to use GSWP3 as atm forcing : put these files in "single_cell_experiment" and adapt them. prepare_atm_dataA="prepare_atm_forcing_data_GSWP_1901-1935.ncl" prepare_atm_dataB="prepare_atm_forcing_data_GSWP_1936-1972.ncl" prepare_atm_dataC="prepare_atm_forcing_data_GSWP_1973-2010.ncl"

regridbatch_file="regridbatch.sh" # You will find this script below–> put it in "single_cell_experiment" stream_file="stream_file_1901_to_2014" # if you use your own atm forcing you will need to modify this file, take it from below #and put it in the "single_cell_experiment" directory. imports="run_locality_append.sh" # You will find this script below–> put it in "single_cell_experiment"

#this are the stations for GRIDNAME in "1x1_Abisko_pan" #"1x1_Kytalyk_pan" "1x1_Bayelva_pan" "1x1_Zackenberg_pan" do

echo echo $GRIDNAME case $GRIDNAME in

1x1_Abisko_pan) plot_lat=68.35 plot_lon=19.05 CDATE=181219 #'date +%y%m%d' ;;

1x1_Bayelva_pan) plot_lat=78.92 plot_lon=11.93 CDATE=190103 #today YYMMDD ;;

1x1_Kytalyk_pan) plot_lat=70.83 plot_lon=147.5 CDATE=190110 #today YYMMDD ;;

1x1_Zackenberg_pan) plot_lat=74.5 plot_lon=339.4 CDATE=190103 #today YYMMDD ;;

esac

module load python2/2.7.10 module load ncl module load nco module unload netcdf.gnu/4.4.1.1

##############################„„„„„„ SURFACEDATA„„„„„„„#######################__________________________A
if false #1 Make SCRIPgrid of single cell____
then

cd ${path2}/tools/mkmapdata ./mknoocnmap.pl -p $plot_lat,$plot_lon -n $GRIDNAME mv ../mkmapgrids/SCRIPgrid_${GRIDNAME}_nomask_c${CDATE}.nc ../mkmapgrids/SCRIPgrid_${GRIDNAME}_nomask.nc mkdir -p ${path2}/tools/mkmapdata/$GRIDNAME mv ${path2}/tools/mkmapdata/map_${GRIDNAME}_noocean_to_${GRIDNAME}_nomask_aave_da_${CDATE}.nc ${path2}/tools/mkmapdata/$GRIDNAME/map_${GRIDNAME}_noocean_to_${GRIDNAME}_nomask_aave_da_${CDA

# 2)Create the mapping files needed by mksurfdata_map____A.2
#here we create a temporary file were we include GRIDNAME and then another were we include CDATE –> CDATE must be CORRECT!!!!!!!!

echo "_______________________________________Start creating mappig files" cd ${path1} tmp=$(<${regridbatch_file}) echo "${tmp//GRIDNAME/$GRIDNAME}" > regridbatch_$GRIDNAME.sh mv -f regridbatch_$GRIDNAME.sh ${path2}/tools/mkmapdata/regridbatch_$GRIDNAME.sh cd ${path2}/tools/mkmapdata chmod u+x regridbatch_$GRIDNAME.sh sbatch regridbatch_$GRIDNAME.sh sleep 300m

fi

if false #2 then

cd ${path2}/tools/mkmapdata mv -f map_*$GRIDNAME* $GRIDNAME/

# 3) Create the domain file____A.3

echo "____Start creating domain file" cd ${path1}

. ./$imports

cd ${path2}/cime/tools/mapping/gen_domain_files/src ../../../configure –macros-format Makefile –mpilib mpi-serial –machine abel –compiler intel –clean . ./.env_mach_specific.sh gmake cd .. OCNDOM=domain.ocn_noocean.nc ATMDOM=domain.lnd.{$GRIDNAME}_noocean.nccd MAPFILE="${path2}/tools/mkmapdata/${GRIDNAME}/map_${GRIDNAME}_noocean_to_${GRIDNAME}_nomask_aave_da_${ ./gen_domain -m $MAPFILE -o $OCNDOM -l $ATMDOM mkdir -p $GRIDNAME mv domain* $GRIDNAME/ mv -f $GRIDNAME/domain.lnd.*{$GRIDNAME}*.nc $GRIDNAME/domain.lnd.{$GRIDNAME}_noocean.nc

fi

if false #3 then # 4) Finally create the surface dataset____

echo "________________________________Start creating surface data" cd ${path1} . ./run_locality_append.sh cd ${path2}/tools/mksurfdata_map/src gmake clean gmake cd .. ./mksurfdata.pl -r usrspec -usr_gname $GRIDNAME -usr_gdate $CDATE -dinlc ${path3} -allownofile -usr_mapdir ../mkmapdata/$GRIDNAME -years 2000 -no-crop #not working for years 1850 :/ #-no-crop necessary because model is expecting 16 pfts, or with crop but change xml variables mv ${path2}/tools/mksurfdata_map/surfdata_${GRIDNAME}_*.nc ${path3}/lnd/clm2/surfdata_map/surfdata_${GRIDNAME}_simyr2000.nc rm -rf surfdata*.log surfdata*.namelist rm -rf ${path2}/tools/mkmapdata/${GRIDNAME}

fi

OCNDOM=domain.ocn_noocean.nc ATMDOM=domain.lnd.{$GRIDNAME}_noocean.nc GENDOM_PATH=${path2}/cime/tools/mapping/gen_domain_files/$GRIDNAME

############################,,,,,,,,,,ATMOSPHERIC FORCING,,,,,,,,,,,,###______________________________B
if false #4 then

cd ${path1} module load ncl ncl 'plot_name="'$GRIDNAME'"' plot_lat=$plot_lat plot_lon=$plot_lon ${prepare_atm_data1} ncl 'plot_name="'$GRIDNAME'"' plot_lat=$plot_lat plot_lon=$plot_lon ${prepare_atm_data2} ncl 'plot_name="'$GRIDNAME'"' plot_lat=$plot_lat plot_lon=$plot_lon ${prepare_atm_data3} ncl 'plot_name="'$GRIDNAME'"' plot_lat=$plot_lat plot_lon=$plot_lon ${prepare_atm_data4} #output is there: /work/users/marlam/inputdata/atm/datm7/CLM1PT_data/

fi if false #4 then

cd ${path1} module load ncl ncl 'plot_name="'$GRIDNAME'"' plot_lat=$plot_lat plot_lon=$plot_lon ${prepare_atm_dataA} ncl 'plot_name="'$GRIDNAME'"' plot_lat=$plot_lat plot_lon=$plot_lon ${prepare_atm_dataB} ncl 'plot_name="'$GRIDNAME'"' plot_lat=$plot_lat plot_lon=$plot_lon ${prepare_atm_dataC} #output is there: /work/users/marlam/inputdata/atm/datm7/CLM1PT_data2/

fi

#0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
#0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

############################,,,,,,,,,CASE__SETUP,,,,,,,,,,,,,,,,,,,,,,,####______________________________
if true #5 then

cd ${path2}/cime/scripts echo "________________________________Start create case" export CESM_ACCOUNT=geofag ./create_newcase –case ~/cases/${GRIDNAME}_${compset0} –compset ${compset0} –res CLM_USRDAT –machine abel –run-unsupported –project geofag

fi

if true #6 then

echo "________________________________Start changes" cd ${path4}/${GRIDNAME}_${compset0} ./xmlchange ATM_DOMAIN_PATH=$GENDOM_PATH,LND_DOMAIN_PATH=$GE ./xmlchange ATM_DOMAIN_FILE=$ATMDOM,LND_DOMAIN_FILE=$ATMDOM ./xmlchange CLM_USRDAT_NAME=$GRIDNAME ./xmlchange STOP_OPTION=nyears ./xmlchange STOP_N=114 ./xmlchange RUN_STARTDATE="1901-01-01" ./xmlchange DATM_MODE="CLM1PT" ./xmlchange RESUBMIT="0" ./xmlchange DIN_LOC_ROOT_CLMFORC="${path3}/atm/datm7/CLM1PT_data" ./xmlchange JOB_WALLCLOCK_TIME="03:59:00" ./xmlchange PROJECT="geofag" ./xmlchange DATM_CLMNCEP_YR_ALIGN="1901" ./xmlchange DATM_CLMNCEP_YR_START="1901"

./xmlchange DATM_CLMNCEP_YR_END=”2014” ./xmlchange DOUT_S=”FALSE” ./xmlchange GMAKE_J=”8” ./xmlchange CLM_ACCELERATED_SPINUP=”off” #fire_method='nofire' cat > user_nl_clm << EOF

&clm_inparm create_crop_landunit = .true. fsurdat='${path3}/lnd/clm2/surfdata_map/surfdata_${GRIDNAME}_simyr2000.nc' hist_mfilt=365 hist_nhtfrq=-24

/ &ndepdyn_nml

ndepmapalgo = ‘nn’

/ &popd_streams

popdensmapalgo = ‘nn’

/ &light_streams

lightngmapalgo = ‘nn’

/ EOF

cat> user_nl_datm <<EOF

&shr_strdata_nml vectors = ‘null’ mapmask = ‘nomask’,’nomask’,’nomask’ mapalgo = ‘nn’,’nn’,’nn’ tintalgo = ‘nearest’,’linear’,’lower’ taxmode = ‘cycle’,’extend’,’extend’ streams = ‘datm.streams.txt.CLM1PT.CLM_USRDAT 1901 1901 2014 ‘,

‘datm.streams.txt.presaero.clim_2000 1 1 1’, ‘datm.streams.txt.topo.observed 1 1 1’

/ EOF

echo “____Changes done” cd ${path1} . ./$imports cd ${path4}/${GRIDNAME}_${compset0} ./case.setup # –reset echo “____Case setup successfully”

# ./case.build –clean ./case.build echo “____Case build successfully”

fi

if false #7 then

cd ${path1} . ./$imports tmp=$(<${stream_file}) echo “${tmp//GRIDNAME/$GRIDNAME}” > user_datm.streams.txt.CLM1PT.CLM_USRDAT_${GRIDNAME} cd ${path4}/${GRIDNAME}_${compset0} ./preview_namelists mv -f ${path1}/user_datm.streams.txt.CLM1PT.CLM_USRDAT_${GRIDNAME} user_datm.streams.txt.CLM1PT.CLM_USRDAT chmod u+w user_datm.streams.txt.CLM1PT.CLM_USRDAT

# ./case.submit

fi cd ${path1} done

##_____END_OF_MAIN_SCRIPT_____##############

##__REGRIDBATCH.SH__################

#!/bin/bash # # Batch script to submit to create mapping files for all standard # resolutions. If you provide a single resolution via “$RES”, only # that resolution will be used. In that case: If it is a regional or # single point resolution, you should set ‘#PBS -n’ to 1, and be sure # that ‘-t regional’ is specified in cmdargs.

#——————————————————————————————— # Set parameters #——————————————————————————————— #SBATCH –account=geofag #SBATCH –job-name=mkmapdata #SBATCH –mem-per-cpu=256G –partition=hugemem #SBATCH –ntasks=1 #SBATCH –time=05:00:00

source /cluster/bin/jobsetup module load esmf/6.3.0rp1 #module load nco #module load ncl export ESMF_NETCDF_LIBS="-lnetcdff -lnetcdf -lnetcdf_c++" #export ESMF_DIR=/usit/abel/u1/huit/ESMF/esmf export ESMF_COMPILER=intel export ESMF_COMM=openmpi #export ESMF_NETCDF="test" export ESMF_NETCDF_LIBPATH=/cluster/software/VERSIONS/netcdf.intel-4.3.3.1/lib export ESMF_NETCDF_INCLUDE=/cluster/software/VERSIONS/netcdf.intel-4.3.3.1/include ulimit -s unlimited

export ESMFBIN_PATH=/cluster/software/VERSIONS/esmf/6_3_0rp1/bin/binO/Linux.intel.64.openmpi.default export CSMDATA=/work/users/marlam/inputdata export MPIEXEC=mpirun

phys="clm4_5" RES=GRIDNAME

GRIDFILE=../mkmapgrids/SCRIPgrid_GRIDNAME_nomask.nc regrid_num_proc=8 #———————————————————————————- # Begin main script #———————————————————————————-

if [-z "$RES"]; then echo "Run for all valid resolutions" resols=‘../../bld/queryDefaultNamelist.pl -res list -silent‘
if [! -z "$GRIDFILE"]; then

echo "When GRIDFILE set RES also needs to be set for a single resolution" exit 1

fi

else resols="$RES"

fi if [-z "$GRIDFILE"]; then

grid=""

else

if [[${#resols[@]} > 1]]; then echo "When GRIDFILE is specificed only one resolution can also be given (# resolutions ${#resols[@]})" echo "Resolutions input is: $resols" exit 1

fi grid="-f $GRIDFILE"

fi

if [-z "$MKMAPDATA_OPTIONS"]; then echo "Run with standard options" options=" "

else options="$MKMAPDATA_OPTIONS"

fi echo "Create mapping files for this list of resolutions: $resols"

#———————————————————————————-

for res in $resols; do echo "Create mapping files for: $res"

#———————————————————————————- cmdargs="-r $res $grid $options"

# For single-point and regional resolutions, tell mkmapdata that # output type is regional if [[*echo "$res"* | *grep -c "1x1_"* -gt 0 || *echo "$res"* | *grep -c "5x5_"* -gt 0]]; then

res_type="regional"

else res_type="global"

fi # Assume if you are providing a gridfile that the grid is regional if [$grid != ""];then

res_type="regional"

fi

cmdargs="$cmdargs -t $res_type"

echo "$res_type" if ["$res_type" = "regional"]; then echo “regional” # For regional and (especially) single-point grids, we can get # errors when trying to use multiple processors - so just use 1. # We also do NOT set batch mode in this case, because some # machines (e.g., yellowstone) do not listen to REGRID_PROC, so to # get a single processor, we need to run mkmapdata.sh in # interactive mode. regrid_num_proc=1

else echo “global” regrid_num_proc=8 if [! -z “$LSFUSER”]; then

echo “batch” cmdargs=”$cmdargs -b”

fi if [! -z “$PBS_O_WORKDIR”]; then

cd $PBS_O_WORKDIR cmdargs=”$cmdargs -b”

fi

fi

echo “args: $cmdargs” echo “time env REGRID_PROC=$regrid_num_proc ./mkmapdata.sh $cmdargsn” time env REGRID_PROC=$regrid_num_proc ./mkmapdata.sh $cmdargs

done ###____RUN_LOCALITY_APPEND.SH___################
##

#!/bin/sh export INC_NETCDF=/cluster/software/VERSIONS/netcdf.intel-4.3.3.1/include export LIB_NETCDF=/cluster/software/VERSIONS/netcdf.intel-4.3.3.1/lib export NETCDF_ROOT=/cluster/software/VERSIONS/netcdf.intel-4.3.3.1

###___STREAM_FILE___###########################

<dataSource> GENERIC

</dataSource> <domainInfo>

<variableNames>

time time xc lon yc lat area area mask mask

</variableNames> <filePath>

/usit/abel/u1/marlam/ctsm/cime/tools/mapping/gen_domain_files/GRIDNAME

</filePath> <fileNames>

domain.lnd.{GRIDNAME}_noocean.nc

</fileNames>

</domainInfo> <fieldInfo>

<variableNames> TBOT tbot SHUM shum WIND wind PRECTmms precn FSDS swdn PSRF pbot

</variableNames> <filePath>

/work/users/marlam/inputdata/atm/datm7/CLM1PT_pan/GRIDNAME

</filePath> <fileNames>

1901-01.nc

1901-02.nc 1901-03.nc 1901-04.nc 2014-12.nc

</fileNames> <offset>

0

</offset>

</fieldInfo>

##__PREPARE_ATMOSPHERIC_FORCING_1901_1920_##############

load “$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl” load “$NCARG_ROOT/lib/ncarg/nclscripts/csm/popRemap.ncl” load “$NCARG_ROOT/lib/ncarg/nclscripts/esmf/ESMF_regridding.ncl” load “$NCARG_ROOT/lib/ncarg/nclscripts/contrib/cd_string.

begin

clim_output = “/work/users/marlam/inputdata/atm/datm7/CLM1PT_data/”+plot_name+”/” clim_input = “/work/users/marlam/inputdata/atm/datm7/atm_forcing.datm7.cruncep_qianFill.0.5d.v7.c160715/”

system(“mkdir -p “+ clim_output)

cruCreat=True varCOMB=False varMOD=False timCOMB=False varTIMEsub=False

mon = ispan(0,11,1) mon@units = “month” month =cd_string(mon,”%N”) print(month)

if (cruCreat) then do year=1901,1920,1

do m=0,11,1

frsds=addfile(clim_input+”Solar6Hrly/clmforc.cruncep.V7.c2016.0.5d.Solr.”+year+”- “+month(m)+”.nc”,”r”) fprec=addfile(clim_input+”Precip6Hrly/clmforc.cruncep.V7.c2016.0.5d.Prec.”+year+”- “+month(m)+”.nc”,”r”) ftphw=addfile(clim_input+”TPHWL6Hrly/clmforc.cruncep.V7.c2016.0.5d.TPQWL.”+year+”- “+month(m)+”.nc”,”r”)

rsds=frsds->FSDS(:,:,:) prec=fprec->PRECTmms(:,:,:) psrf=ftphw->PSRF(:,:,:) tbot=ftphw->TBOT(:,:,:) wind=ftphw->WIND(:,:,:) qbot=ftphw->QBOT(:,:,:)

lon=frsds->LONGXY(0,:) lat=frsds->LATIXY(:,0) time=frsds->time(:) edgew =frsds->EDGEW edgee =frsds->EDGEE edges =frsds->EDGES edgen =frsds->EDGEN longxy =frsds->LONGXY latxy =frsds->LATIXY

loni=ind_nearest_coord(plot_lon, lon, 0) latj=ind_nearest_coord(plot_lat, lat, 0) print(loni) print(latj)

;###

system(“rm -f “+clim_output+year+”-“+month(m)+”.nc”) ; remove if exists fclim = addfile(clim_output+year+”-“+month(m)+”.nc”,”c”) ;”**clm1pt_1x1_**”+plot_name+”_” ntim = dimsizes(time) ; get dimension sizes nlat = 1 nlon = 1 nscalar = 1

setfileoption(fclim,”DefineMode”,True)

fAtt = True ; assign file attributes fAtt@case_title = “CRUNCEP: norway ” fAtt@conventions = “CF-1.0” fAtt@title = “CLM single point datm input data” fAtt@history = “Original data from CRUNCEP data” fAtt@creation_date = systemfunc (“date”) fileattdef(fclim, fAtt) ; copy file attributes

dimNames = (/”scalar”,”lon”,”lat”,”time”/) dimSizes = (/ nscalar, nlon, nlat, -1 /) dimUnlim = (/ False, False, False, True/) filedimdef(fclim,dimNames,dimSizes,dimUnlim)

filevardef(fclim, “EDGEW” ,typeof(edgew),getvardims(edgew)) filevardef(fclim, “EDGEE” ,typeof(edgee),getvardims(edgee)) filevardef(fclim, “EDGES” ,typeof(edges),getvardims(edges)) filevardef(fclim, “EDGEN” ,typeof(edgen),getvardims(edgen)) filevardef(fclim, “LONGXY” ,typeof(longxy) ,getvardims(longxy)) filevardef(fclim, “LATIXY” ,typeof(latxy) ,getvardims(latxy)) filevardef(fclim, "FSDS",typeof(rsds),getvardims(rsds)) filevardef(fclim, "PRECTmms",typeof(prec),getvardims(prec)) filevardef(fclim, "TBOT",typeof(tbot),getvardims(tbot)) filevardef(fclim, "WIND",typeof(wind),getvardims(wind)) filevardef(fclim, "PSRF",typeof(psrf),getvardims(psrf)) filevardef(fclim, "SHUM",typeof(qbot),getvardims(qbot)) filevardef(fclim, "time",typeof(time),getvardims(time))

filevarattdef(fclim,"EDGEW",edgew) filevarattdef(fclim,"EDGEE",edgee) filevarattdef(fclim,"EDGES",edges) filevarattdef(fclim,"EDGEN",edgen) filevarattdef(fclim,"LONGXY",longxy) filevarattdef(fclim,"LATIXY",latxy) filevarattdef(fclim,"FSDS",rsds) filevarattdef(fclim,"PRECTmms",prec) filevarattdef(fclim,"TBOT",tbot) filevarattdef(fclim,"WIND",wind) filevarattdef(fclim,"PSRF",psrf) filevarattdef(fclim,"SHUM",qbot) filevarattdef(fclim,"time",time)

setfileoption(fclim,"DefineMode",False)

print(time) fclim->time = (/time/) ; "(/", "/)" syntax tells NCL to only output the data values to the predefined locations on the file. print(fclim->time) xfloor=plot_lon-0.01 xceil =plot_lon+0.01 yfloor=plot_lat-0.01 yceil =plot_lat+0.01 print(yceil) print(xceil) fclim->EDGEW = (/xfloor/) fclim->EDGEE = (/xceil/) fclim->EDGES = (/yfloor/) fclim->EDGEN = (/yceil/)

printVarSummary(plot_lon) fclim->LONGXY = (/plot_lon/) fclim->LATIXY = (/plot_lat/) fclim->FSDS = (/rsds(:,latj,loni)/) fclim->PRECTmms = (/prec(:,latj,loni)/) ; !!!! time variable can be modified in the attributes of the variable is also copied. make sure to use (/ /) to only copy data. fclim->TBOT = (/tbot(:,latj,loni)/) fclim->WIND = (/wind(:,latj,loni)/) fclim->PSRF = (/psrf(:,latj,loni)/) fclim->SHUM = (/qbot(:,latj,loni)/)

print(fclim->time)

delete(rsds) delete(time) delete(prec) delete(tbot) delete(qbot) delete(wind) delete(psrf)

end do

end do

end if

if (varCOMB) then

do year=1996,2015,1 system("cp "+clim_output+"**clm1pt_1x1_**"+plot_name+"_FSDS_"+year+"01010130-"+year+"12312230.nc "+clim_output+"**clm1pt_1x1_**"+plot_name+"_"+year+".nc") system("ncks -h -A "+clim_output+"**clm1pt_1x1_**"+plot_name+"_PRECTmms_"+year+"01010130-"+year+"12312230.nc "+clim_output+"**clm1pt_1x1_**"+plot_name+"_"+year+".nc") system("ncks -h -A "+clim_output+"**clm1pt_1x1_**"+plot_name+"_TBOT_"+year+"01010130-"+year+"12312230.nc "+clim_output+"**clm1pt_1x1_**"+plot_name+"_"+year+".nc") system("ncks -h -A "+clim_output+"**clm1pt_1x1_**"+plot_name+"_WIND_"+year+"01010130-"+year+"12312230.nc "+clim_output+"**clm1pt_1x1_**"+plot_name+"_"+year+".nc") system("ncks -h -A "+clim_output+"**clm1pt_1x1_**"+plot_name+"_PSHUM_"+year+"01010130-"+year+"12312230.nc "+clim_output+"**clm1pt_1x1_**"+plot_name+"_"+year+".nc")

end do

end if

if (varMOD) then do year=1981,2010,1 system("ncrename -v LATXY,LATIXY "+clim_output+"**clm1pt_1x1_**"+plot_info(0,1)+"_"+year+".nc")

end do

end if

if (varTIMEsub) then do year=1981,2010,1

system("ncrcat -O -d time,0,2919,2 "+clim_output+"**clm1pt_1x1_**"+plot_info(0,1)+"_"+year+".nc "+clim_output+"**clm1pt_1x1_**"+plot_info(0,1)+"_"+year+".nc")

end do

end if

;if (varSHUMMOD) then ; do year=1981,2010,1 ; ncrcat -O clm1pt_322_1982-*****.nc clm1pt_322_1983.nc ; ncks -v PSRF clm1pt_322_1981-01.nc test.nc ; system("ncrename -v LATXY,LATIXY "+clim_output+"**clm1pt_**"+plot_info(0,1)+"_"+year+".nc") ; end do ;end if

if (timCOMB) then

do m=0,20,1 system("ncrcat -O "+clim_output+"**clm1pt_1x1_**"+plot_name+"_*.nc "+clim_output+"**clm1pt_1x1_**"+plot_name+"_1981-2010.nc")

end do

end if

end

Change configuration

```
./xmlchange NTASKS=1 #(number of CPU's, can be increased if excitation error)
```

3.6 Run a regional case

Here, the regional case procedure is explained for running the model over Fennoscandia and Svalbard region at 0.25 degree spatial resolution on Abel.

Required input files (domain and surface data) are already produced and stored under the main inputdata directory. So, start with linking the main inputdata files to your working directory.

```
cd ~/ctsm/cime/scripts
./link_dirtree $CESM_DATA /work/users/$USER/inputdata
export MYCTSMDATA=/work/users/$USER/inputdata
```

3.6.1 Definitions

As mentioned earlier in this documentation, you need to define your project account (e.g. nn2806k) before running the model. To be sure that you are using the correct version of netcdf and practical purposes, we will export the paths and some environment variables :

```
export CESM_ACCOUNT=nn2806k
export PROJECT=nn2806k
export CTSMROOT=/cluster/home/$USER/ctsm
```

(continues on next page)

(continued from previous page)

```
export INC_NETCDF=/cluster/software/VERSIONS/netcdf.intel-4.3.3.1/
export LIB_NETCDF=/cluster/software/VERSIONS/netcdf.intel-4.3.3.1/
export NETCDF_ROOT=/cluster/software/VERSIONS/netcdf.intel-4.3.3.1
```

3.6.2 Make a case

To run the model with Satellite Phenology mode, do the following

```
cd $CTSMROOT/cime/scripts
./create_newcase -case ~/cases/SCA_SpRun -res CLM_USRDAT -compset I2000Clm50SpGs --
↪machine abel --run-unsupported --project $CESM_ACCOUNT
```

Here, the compset **I2000Clm50SpGs** initialize the model from 2000 conditions with GSWP3 atmospheric forcing data. If you want to run CTSM with BGC mode and force with CRU NCEP v7 data set, use **I2000Clm50BgcCruGs** compset. To see available compset aliases and their long names, type the following :

```
$CTSMROOT/cime/scripts/query_config --compsets clm
```

Note that you can use only the compsets with **SGLC** (Stub glacier (land ice) component) for single point and regional cases. Otherwise, simulation fails.

3.6.3 Set up the case

Now, we are setting up our test simulation for three years between 2000-2002.

```
cd ~/cases/SCA_SpRun
export GRIDNAME=0.25_SCA
export SCA_DATA=$MYCTSMDATA/share/domains/
export ATMDOM=domain.0.25x0.25_SCA.nc

./xmlchange DIN_LOC_ROOT=$MYCTSMDATA
./xmlchange ATM_DOMAIN_PATH=$SCA_DATA,LND_DOMAIN_PATH=$SCA_DATA
./xmlchange ATM_DOMAIN_FILE=$ATMDOM,LND_DOMAIN_FILE=$ATMDOM
./xmlchange CLM_USRDAT_NAME=$GRIDNAME
./xmlchange STOP_OPTION=nyears
./xmlchange STOP_N=3
./xmlchange NTASKS=16
./xmlchange JOB_WALLCLOCK_TIME="03:59:00"
./xmlchange PROJECT=$CESM_ACCOUNT
./xmlchange RUN_STARTDATE="2000-01-01"
./xmlchange RESUBMIT="0"
./xmlchange DATM_CLMNCEP_YR_ALIGN="2000"
./xmlchange DATM_CLMNCEP_YR_START="2000"
./xmlchange DATM_CLMNCEP_YR_END="2002"
./xmlchange DOUT_S="FALSE"
./xmlchange GMAKE_J="8"
./case.setup
./preview_run
```

3.6.4 Customizing the CLM namelist

In order to define the location of surface data file for our Scandinavia region, we add the **fsurdat** setting to the land model's namelist.

```
cat << EOF > user_nl_clm
fsurdat="$MYCTSMDATA/lnd/clm2/surfdata_map/surfdata_0.25_SCA_hist_16pfts_Irrig_CMIP6_
↪simyr2000_c190814.nc"
EOF
```

If you run your simulation with **BGC** mode, then you need to set another surface data map as follows:

```
cat << EOF > user_nl_clm
fsurdat="$MYCTSMDATA/lnd/clm2/surfdata_map/surfdata_0.25_SCA_hist_78pfts_CMIP6_
↪simyr2000_c190819.nc"
EOF
```

If you want to print out only selected variables in hourly resolution in the model output files for each year:

```
cat << EOF > user_nl_clm
fsurdat="$MYCTSMDATA/lnd/clm2/surfdata_map/surfdata_0.25_SCA_hist_16pfts_Irrig_CMIP6_
↪simyr2000_c190814.nc"
hist_empty_htapes = .true.
hist_fincl1 = 'TSA', 'TSKIN', 'EFLX_LH_TOT', 'FSH', 'WIND', 'TWS', 'SNOWLIQ', 'SNOWICE
↪', 'SNOW_DEPTH', 'TSOI', 'H2OSOI'
hist_nhtfrq = -1
hist_mfilt = 365
EOF
```

Note that the variable names (e.g. 'TSA', 'TSKIN', 'EFLX_LH_TOT', etc.) are following the CESM name convention.

3.6.5 Build and submit the case

Final step is to build the case and submit the job to the queue.

```
./case.build
./case.submit
```

CHAPTER 4

Spin-up

CHAPTER 5

Indices and tables

- genindex
- modindex
- search