

Instructions for using NEMO on SISU

Yongmei Gong
8 March, 2018

The following instruction consist text instructions and a number of well commented bash files on how to launch basic nemo simulations. They are created for the environment of the super computer Sisu in CSC-IT center for science.

Contents

1	Get the code	2
2	Declare the compilers	3
3	Build NEMO for experiment - GYRE_XIOS	4
4	Build NEMO for experiment - ORCA2_LIM3	6
5	Build NEMO for experiment - ORCA1_LIM3	8
6	Clean nemo build	11
7	Check the run status and outputs	11

1 Get the code

1. Sign up for the wiki (<http://forge.ipsl.jussieu.fr/nemo/wiki/Users>) by sending an email containing your username (5 characters minimum length) to nemo@forge.ipsl.jussieu.fr;
2. Get the conformation email then reset your password.
3. Now download the source code from the distribution

```
#!/bin/bash

set -ex

5 # 2018-02-20, juha.lento@csc.fi'
# modified 2018-03-08, yongmei.gong@helsinki.fi

# NEMO is "research code", which for NEMO means that:
# - "install" in the NEMO documentation actually is what
#   is usually referred as "build"

10 # The following build instructions are based on:
# - https://forge.ipsl.jussieu.fr/nemo/wiki/Users/
#   ModelInstall#ExtracttheNEMOcode
# - https://forge.ipsl.jussieu.fr/nemo/wiki/Users/
#   ModelInterfacing/InputsOutputs#
#   ExtractingandinstallingXIOS

15 #Your user name used for registering to nemo user wiki
USER="YOUR_USER_NAME"

# Load system I/O libraries (that is how we use an
#   existing software on Sisu)
# now use xios2.0 by default

20 module load cray-hdf5-parallel cray-netcdf-hdf5parallel
    xios/

# Create the main NEMO directory (a folder) where you want
#   to store the source code

25 # Here I put in in the user application directory in Sisu,
#   $USERAPPL, where you can build (install) your own
#   software

mkdir -p $USERAPPL/nemo_test
cd $USERAPPL/nemo_test

30 # Checkout (download) the source code
```

```

# Type your password on the screen and type yes when asked
.

svn --username $USER co http://forge.ipsl.jussieu.fr/nemo/
svn/branches/2015/nemo_v3_6_STABLE/NEMOGCM

```

2 Declare the compilers

All compiler options in NEMO are controlled using files in NEMOGCM/ARCH/arch. Now we create a file to declare the compilers we use to build nemo according to what we have in Sisu.

```

#!/bin/bash

set -ex

# NEMO 3.6 STABLE + XIOS-2 build instructions for sisu.csc
# .fi
#
# 2018-02-20, juha.lento@csc.fi
# modified 2018-03-08, yongmei.gong@helsinki.fi

# NEMO is "research code", which for NEMO means that:
# - "install" in the NEMO documentation actually is what
#   is usually referred as "build"

# The following build instructions are based on:
# - https://forge.ipsl.jussieu.fr/nemo/wiki/Users/
#   ModelInstall#ExtracttheNEMOcode
# - https://forge.ipsl.jussieu.fr/nemo/wiki/Users/
#   ModelInterfacing/InputsOutputs#
#   ExtractingandinstallingXIOS

# NEMO build
# All compiler options in NEMO are controlled using files
#   in NEMOGCM/ARCH/arch-'my_arch'.fcm where 'my_arch' is
#   the name of the computing architecture.

# Now we create a file to declare the compilers we use to
#   build nemo according to what we have in Sisu

cd $USERAPPL/nemo_test/NEMOGCM
cat > ARCH/arch-XC40-SISU.fcm <<EOF
%NCDF_HOME          $NETCDF_DIR
%25 %HDF5_HOME       $HDF5_DIR
%XIOS_HOME          $(pkg-config --variable=CRAY_prefix
xios)

```

```

30 %NCDF_INC          -I%NCDF_HOME/include -I%HDF5_HOME/
    include
    %NCDF_LIB          -L%HDF5_HOME/lib -L%NCDF_HOME/lib -
        lnetcdf -lnetcdf -lhdf5_hl -lhdf5 -lz
    %XIOS_INC          -I%XIOS_HOME/inc
    %XIOS_LIB          -L%XIOS_HOME/lib -lxios
    %CPP              cpp
    %FC              ftn
    %FCFLAGS          -emf -s real64 -s integer32 -02 -
        hflex_mp=intolerant -Rb
    %FFLAGS          -emf -s real64 -s integer32 -00 -
        hflex_mp=strict -Rb
35 %LD              ftn
    %FPPFLAGS          -P -E -traditional-cpp
    %LDFLAGS          -hbyteswapio
    %AR              ar
    %ARFLAGS          -r
40 %MK              gmake
    %USER_INC          %XIOS_INC
    %USER_LIB          %XIOS_LIB
    %CC              cc
    %CFLAGS          -00
45 EOF

```

3 Build NEMO for experiment - GYRE_XIOS

Now we build an executable for the experiment GYRE_XIOS.

We use the up-to-date version of XIOS - xios2.0.990 instead the older xios1.0. This requires declaring the active cpp keys in the cpp_*.fcm files.

```

#!/bin/bash

set -ex

5 # NEMO 3.6 STABLE + XIOS-2 build instructions for sisu.csc
  .fi
#
# 2018-02-20, juha.lento@csc.fi
# modified 2018-03-08, yongmei.gong@helsinki.fi

10 # NEMO is "research code", which for NEMO means that:
# - "install" in the NEMO documentation actually is what
  is usually referred as "build"

# The following build instructions are based on:
# - https://forge.ipsl.jussieu.fr/nemo/wiki/Users/
  ModelInstall#ExtracttheNEMOcode

```

```

15 # - https://forge.ipsl.jussieu.fr/nemo/wiki/Users/
    ModelInterfacing/InputsOutputs#
    ExtractingandinstallingXIOS

    # Load system I/O libraries
    # might change to xios2.0

20 module load cray-hdf5-parallel cray-netcdf-hdf5parallel
    xios/2.0.990

    # Declare your NEMO code directory

25 NEMOBUILD="$USERAPPL/nemo_test3"

    # NEMO build

    # Declare your configuration for the simulation, e.g. the
    GYRE_XIOS experiment
30 cd $NEMOBUILD/NEMOGCM/CONFIG

    #you need to add new keys in .fcm file
    sed -i 's/$/ key_nosignedzero/' GYRE_XIOS/cpp_GYRE_XIOS.
        fcm

35 # Here you compile a executable for the experiment
    GYRE_XIOS in either $TMPDIR or $WRKDIR/DONOTREMOVE
    ./makenemo -t $TMPDIR -m XC40-SISU -r GYRE_XIOS -n
    MY_GYRE_XIOS

    # NEMO test
    #
40 # For a quick test, only!

    # cd MY_GYRE_XIOS/EXP00
    # cp $TMPDIR/MY_GYRE_XIOS/BLD/bin/nemo.exe .
    # aprun -n 4 nemo.exe

45

    # For actual experiments:
    # Copy the experiments in $WRKDIR/DONOTREMOVE
    cp GYRE_XIOS/ $WRKDIR/DONOTREMOVE/MY_GYRE_XIOS/
50 cd $WRKDIR/DONOTREMOVE/MY_GYRE_XIOS/EXP00

    # Copy the executable to the EXP00 directory
    cp $TMPDIR/MY_GYRE_XIOS/BLD/bin/nemo.exe .

55 # Creat a script for Using SLURM commands to execute batch
    jobs

```

```

in Sisu queue
# More about the SLURM commands can be found in
# - https://research.csc.fi/sisu-using-slurm-commands-to-execute-batch-jobs

60 cat > batch_job.sh <<EOF
#!/bin/bash -l
#SBATCH -t 00:29:00
#SBATCH -J gyre_xios
#SBATCH -p test
65 #SBATCH -o gyre_xios.%j
#SBATCH -e gyre_xios_err.%j
#SBATCH -N 4

aprun -n 4 nemo.exe
70 EOF

# Then submit the job in the queue
sbatch batch_job.sh

```

4 Build NEMO for experiment - ORCA2_LIM3

Now we build an executable for the experiment ORCA2_LIM3.

ORCA is the generic name given to global ocean configurations. Its specificity lies on the horizontal curvilinear mesh used to overcome the North Pole singularity found for geographical meshes. LIM (Louvain-la-Neuve sea-ice model, multi-category model LIM3 is used) is a thermodynamic-dynamic sea ice model specifically designed for climate studies.

Similarly this requires declaring the active cpp keys in the cpp_*_fcm files. And this time the experiment needs input data.

```

#!/bin/bash

set -ex

5 # NEMO 3.6 STABLE + XIOS-2 build instructions for sisu.csc
  .fi
#
# 2018-02-20, juha.lento@csc.fi
# modified 2018-03-08, yongmei.gong@helsinki.fi

10 # NEMO is "research code", which for NEMO means that:
# - "install" in the NEMO documentation actually is what
  is usually referred as "build"

```

```

# The following build instructions are based on:
# - https://forge.ipsl.jussieu.fr/nemo/wiki/Users/
  ModelInstall#ExtracttheNEMOcode
15 # - https://forge.ipsl.jussieu.fr/nemo/wiki/Users/
  ModelInterfacing/InputsOutputs#
  ExtractingandinstallingXIOS

# Load system I/O libraries
# might change to xios2.0

20 module load cray-hdf5-parallel cray-netcdf-hdf5parallel
  xios/2.0.990

# Declare your NEMO code directory

25 NEMOBUILD="$USERAPPL/nemo_test3"

# NEMO build

# Declare your configuration for the simulation, e.g. the
  ORCA2_LIM3 experiment
30 cd $NEMOBUILD/NEMOGCM/CONFIG
#you need to add new keys in .fcm file
sed -i 's/ key_nosignedzero key_xios2/' ORCA2_LIM3/
  cpp_ORCA2_LIM3.fcm

# Here you compile a executable for the experiment
  ORCA2_LIM3 in $TMPDIR
35 ./makenemo -t $TMPDIR -m XC40-SISU -r ORCA2_LIM3 -n
  MY_ORCA2_LIM3

# NEMO test
40 #
cd ORCA2_LIM3/EXP00
# If you use XIOS2.0 you need to copy all the .xml file
  from GYRE_XIOS
cp $NEMOBUILD/NEMOGCM/CONFIG/GYRE_XIOS/EXP00/*.xml .

45 # For actual experiments:
# Copy the experiments in $WRKDIR/DONOTREMOVE
cd ../..
cp ORCA2_LIM3/ $WRKDIR/DONOTREMOVE/MY_ORCA2_LIM3/
50 cd $WRKDIR/DONOTREMOVE/MY_ORCA2_LIM3/
cp $TMPDIR/MY_ORCA2_LIM3/BLD/bin/nemo.exe .

```



```

# now you need to download input data ORCA2_LIM_nemo_v3.6.
# tar from http://forge.ipsl.jussieu.fr/nemo/wiki/Users/
# ReferenceConfigurations/ORCA2_LIM3_PISCES
#
55 # Then extract them in EXP00
tar xvf ORCA2_LIM_nemo_v3.6.tar
gzip -d *gz

# Creat a script for Using SLURM commands to execute batch
# jobs
60 in Sisu queue
# More about the SLURM commands can be found in
# - https://research.csc.fi/sisu-using-slurm-commands-to-
# execute-batch-jobs
cat > batch_job.sh <<EOF
#!/bin/bash -l
65 #SBATCH -t 00:29:00
#SBATCH -J ORCA2_LIM3
#SBATCH -p test
#SBATCH -o ORCA2_LIM3.%j
#SBATCH -e ORCA2_LIM3_err.%j
70 #SBATCH -N 4

aprun -n 4 nemo.exe
EOF

75 # Then submit the job in the queue
sbatch batch_job.sh

```

5 Build NEMO for experiment - ORCA1_LIM3

The difference between ORCA1 and ORCA2 is that the spacial resolution of the former is 1 degree and the latter is 2 degree. So ORCA1 has higher resolution and also needs more input data.

You will need files and data in two folders: ORCA1_cfg and ORCA1_data to create ORCA1 experiment configuration

```

#!/bin/bash

set -ex

5 # NEMO 3.6 STABLE + XIOS-2 build instructions for sisu.csc
# .fi
#
# 2018-02-20, juha.lento@csc.fi
# modified 2018-03-06, yongmei.gong@helsinki.fi

```

```

10  # NEMO is "research code", which for NEMO means that:
    # - "install" in the NEMO documentation actually is what
        is usually referred as "build"

    # The following build instructions are based on:
    # - https://forge.ipsl.jussieu.fr/nemo/wiki/Users/
        ModelInstall#ExtracttheNEMOcode
15  # - https://forge.ipsl.jussieu.fr/nemo/wiki/Users/
        ModelInterfacing/InputsOutputs#
        ExtractingandinstallingXIOS

    # Load system I/O libraries
    # might change to xios2.0

20  module load cray-hdf5-parallel cray-netcdf-hdf5parallel
        xios/2.0.990

    # Declare your NEMO code directory

25  NEMOBUILD="$USERAPPL/nemo_test3"

    # NEMO build

    # There is no ORCA1 experiment, which means in the
        standard distribution therefore we need to creat it
        ourselves

30  cd $NEMOBUILD/NEMOGCM/CONFIG
    mkdir ORCA1
    cd ORCA1

35  # put the cpp file here
    cp /homeappl/home/ygong/appl_sisu/NEMO/NEMO_local/
        ORCA1_cfg/cpp_ORCA1_LIM3.fcm .

    # Creat a experiment dir EXP00 and put all the xml and
        namelist files and data there
    # Make sure that you have all the file_* and field_* xml
        files listed in context_nemo.xml (you don't need *
        pisces.xml files if you don't want to couple with
        biochemistry).

40  # you can put the line 'nn_msh      =      0      ! create
        (=1) a mesh file or not (=0)' under &namdom in the file
        namelist_cfg.
    # This prevents nemo to creat the mesh files, which will
        cause trouble when you re-launch the experiment in the

```

```

    same directory as they cannot be over-written

mkdir EXP00
45 cp /homeappl/home/ygong/appl_sisu/NEMO/NEMO_local/
    ORCA1_cfg/*xml .
cp /homeappl/home/ygong/appl_sisu/NEMO/NEMO_local/
    ORCA1_cfg/namelist* .

# Here you compile a executable for the experiment ORCA1
# in $TMPDIR
cd ..
50 ./makenemo -t $TMPDIR -m XC40-SISU -r ORCA1 -n MY_ORCA1


# NEMO test
55 # now the experiment gets bigger so we do it in $WRKDIR
cd $WRKDIR/DONOTREMOVE/
mkdir MY_ORCA1
cd ORCA1
cp $NEMOBUILD/NEMOGCM/CONFIG/ORCA1/EXP00 .
60 cp $TMPDIR/MY_ORCA2_LIM3/BLD/bin/nemo.exe .

# Data includes all the data describing the boundray and
# initial conditions and the climatology in NetCDF format
# And a samll *.dat file describing the humidity
cp /homeappl/home/ygong/appl_sisu/NEMO/NEMO_local/
    ORCA1_data/* .

65 # Creat a script for Using SLURM commands to execute batch
# jobs
# in Sisu queue
# More about the SLURM commands can be found in
# - https://research.csc.fi/sisu-using-slurm-commands-to-execute-batch-jobs

70 cat > batch_job.sh <<EOF
#!/bin/bash -l
#SBATCH -t 12:00:00
#SBATCH -J ORCA1
#SBATCH -p small
75 #SBATCH -o ORCA1.%j
#SBATCH -e ORCA1_err.%j
#SBATCH -N 4

aprun -n 24 nemo.exe
80 EOF

# submit the job

```

```
sbatch batch_job.sh
```

6 Clean nemo build

If something has gone wrong with or has been changed for the build of nemo (nemo.exe)

- Clean a bad configuration

```
./makenemo -C YOUR_CONFIG clean_config
```

- Uninstalling (Clean up the whole thing)

```
./makenemo -n YOUR_BUILD clean
```

(e.g. ./makenemo ./makenemo -t \$TMPDIR -m XC40-SISU -n MY_GYRE_XIOS clean)

7 Check the run status and outputs

- If run crushes, try to find 'E R R O R' section in ocean.output - Use ncview (now it is only in Petteris directory) to check the results *.nc files

```
cd /homeappl/home/puotila/bin
```

```
export HDF5_DISABLE_VERSION_CHECK=1
```

```
./ncview /YOUR_OUTPUT_DIR/_5d_00010101_00011230_grid_T.nc &
```