NEMO-PISCES model configuration in Gadi

Ming Cheng1,2, Michael Ellwood1,2

1Research School of Earth Sciences, Australian National University, Canberra, ACT, Australia; 2Australian Centre for Excellence in Antarctic Science, Research School of Earth Sciences, Australian National University, Canberra, ACT, Australia

A black background with a gold crown

Description automatically generated

A close-up of a logo

Description automatically generated

For users:

PISCES is a biogeochemical model that simulates marine biological productivity and describes the biogeochemical cycles of carbon, oxygen and main nutrients (P, N, Si, Fe) (Aumont et al., 2015). It is the marine biogeochemistry component of two ocean modelling platforms (NEMO and CROCO), three Earth System models (IPSL-CM, CNRM-CM and EC-Earth) and one operational oceanographic system (MERCATOR-Ocean).

The GADI (Garvan Data Intensive) cluster is a high-performance computing facility designed to support data-intensive research and computational workloads across various scientific disciplines, managed by the National Computational Infrastructure (NCI) in Australia.

This document will provide instructions on NEMO-PISCES configuration on GADI for people in Australia who want to work on PISCES.

NEMO: https://www.nemo-ocean.eu/

PISCES: https://www.pisces-community.org/

Instructions:

Log in to the Gadi cluster and change the directory to your work directory

|  |
| --- |
| cd $WORK |

Download xios library

|  |
| --- |
| svn co http://forge.ipsl.jussieu.fr/ioserver/svn/XIOS/trunk/  cd trunk |

Define the architectural files

**arch-X64\_GADI.env:**

|  |
| --- |
| vim arch/arch-X64\_GADI.env |

|  |
| --- |
| module load openmpi/4.1.5  module load gcc/system  module load netcdf/4.9.2p  module load hdf5/1.12.1p  module load boost/1.80.0 |

**arch-X64\_GADI.path:**

|  |
| --- |
| vim arch/arch-X64\_GADI.path |

|  |
| --- |
| NETCDF\_INCDIR=""  NETCDF\_LIBDIR=""  NETCDF\_LIB="-lnetcdff -lnetcdf"  MPI\_INCDIR=""  MPI\_LIBDIR=""  MPI\_LIB=""  HDF5\_INCDIR=""  HDF5\_LIBDIR=""  HDF5\_LIB="-lhdf5\_hl -lhdf5 -lz" |

**arch-X64\_GADI.fcm:**

|  |
| --- |
| vim arch/arch-X64\_GADI.fcm |

|  |
| --- |
| ################################################################################  ################### Projet XIOS ###################  ################################################################################  %CCOMPILER mpicc  %FCOMPILER mpif90  %LINKER mpif90  %BASE\_CFLAGS -ansi -w -D\_\_XIOS\_EXCEPTION -D\_GLIBCXX\_USE\_CXX11\_ABI=0 -std=c++11  %PROD\_CFLAGS -O3 -DBOOST\_DISABLE\_ASSERTS  %DEV\_CFLAGS -g -O2  %DEBUG\_CFLAGS -g  %BASE\_FFLAGS -D\_\_NONE\_\_  %PROD\_FFLAGS -O3  %DEV\_FFLAGS -g -O2  %DEBUG\_FFLAGS -g  %BASE\_INC -D\_\_NONE\_\_  %BASE\_LD -lstdc++  %CPP cpp  %FPP cpp -P  %MAKE gmake |

Make a change in **bld.cfg**

|  |
| --- |
| cd ..  vim bld.cfg |

**bld.cfg**: change netcdf to netcdf4 in corresponding position

|  |
| --- |
| bld::tool::cflags %CFLAGS %CBASE\_INC -I${PWD}/extern/src\_netcdf4 -I${PWD}/extern/boost/include -I{PWD}/extern/rapidxml/include -I${PWD}/extern/blitz/include |

Compile xios library

|  |
| --- |
| ./make\_xios --dev --job 4 --full --arch X64\_GADI |

Download NEMO codes

|  |
| --- |
| cd $WORK  git clone --branch 4.2.1 https://forge.nemo-ocean.eu/nemo/nemo.git nemo\_4.2.1  cd nemo\_4.2.1 |

Define architectural file

|  |
| --- |
| cp arch/arch-linux\_gfortran.fcm arch/arch-X64\_GADI.fcm  vim arch/arch-X64\_GADI.fcm |

**arch-X64\_GADI.fcm:**

|  |
| --- |
| # generic gfortran compiler options for linux  #  # NCDF\_HOME root directory containing lib and include subdirectories for netcdf4  # HDF5\_HOME root directory containing lib and include subdirectories for HDF5  # XIOS\_HOME root directory containing lib for XIOS  # OASIS\_HOME root directory containing lib for OASIS  #  # NCDF\_INC netcdf4 include file  # NCDF\_LIB netcdf4 library  # XIOS\_INC xios include file (taken into accound only if key\_iomput is activated)  # XIOS\_LIB xios library (taken into accound only if key\_iomput is activated)  # OASIS\_INC oasis include file (taken into accound only if key\_oasis3 is activated)  # OASIS\_LIB oasis library (taken into accound only if key\_oasis3 is activated)  #  # FC Fortran compiler command  # FCFLAGS Fortran compiler flags  # FFLAGS Fortran 77 compiler flags  # LD linker  # LDFLAGS linker flags, e.g. -L<lib dir> if you have libraries  # FPPFLAGS pre-processing flags  # AR assembler  # ARFLAGS assembler flags  # MK make  # USER\_INC complete list of include files  # USER\_LIB complete list of libraries to pass to the linker  # CC C compiler used to compile conv for AGRIF  # CFLAGS compiler flags used with CC  #  # Note that:  # - unix variables "$..." are accpeted and will be evaluated before calling fcm.  # - fcm variables are starting with a % (and not a $)  #  %NCDF\_HOME /apps/netcdf/4.9.2p  %HDF5\_HOME /apps/hdf5/1.12.1p  %XIOS\_HOME $WORK/trunk  %NCDF\_INC -I%NCDF\_HOME/include -I%HDF5\_HOME/include  %NCDF\_LIB -L%NCDF\_HOME/lib -lnetcdff -lnetcdf  %XIOS\_INC -I%XIOS\_HOME/inc  %XIOS\_LIB -L%XIOS\_HOME/lib -lxios -L/usr/lib/gcc/x86\_64-linux-gnu/5 -lstdc++  %CPP cpp -Dkey\_nosignedzero  %FC mpif90 -c -cpp  %FCFLAGS -fdefault-real-8 -O3 -funroll-all-loops -fcray-pointer -ffree-line-length-none  %FFLAGS %FCFLAGS  %LD mpif90  %LDFLAGS  %FPPFLAGS -P -C -traditional  %AR ar  %ARFLAGS rs  %MK make  %USER\_INC %XIOS\_INC %NCDF\_INC  %USER\_LIB %XIOS\_LIB %NCDF\_LIB  %CC cc  %CFLAGS -O0 |

Compile NEMO in ORCA2\_OFF\_PISCES configuration

|  |
| --- |
| source $WORK/trunk/arch/arch-X64\_GADI.env  ./makenemo -m X64\_GADI -r ORCA2\_OFF\_PISCES -n ORCA2\_OFF\_PISCES\_GADI -j 4 |

Compile rebuild tool

|  |
| --- |
| cd tools  ./maketools -m X64\_GADI -n REBUILD -j 4 |

Get your input files, restart file, xml files and namelists.

Inputs files (doi: 10.5281/zenodo.3767939):

https://zenodo.org/records/3767939/files/INPUTS\_PISCES\_v4.2.tar?download=1

https://zenodo.org/records/3767939/files/ORCA2\_OFF\_v4.0.tar?download=1

Run a job script

|  |
| --- |
| mkdir $WORK/test  cd $WORK/test  vim nemo\_4.2.1\_test |

**nemo\_4.2.1\_test:**

|  |
| --- |
| #!/bin/bash  #PBS -l ncpus=912  #PBS -l mem=1000GB  #PBS -l jobfs=1000GB  #PBS -q normal  #PBS -P oz91  #PBS -l walltime=13:00:00  #PBS -l storage=scratch/projectname+gdata/projectname  #PBS -l wd  # define some information about the job  ulimit -s unlimited  export NPROC=912  export XPROC=0  let RUNPROC=$NPROC-$XPROC  export ENAM=nemo4.2.1\_test  export YEAR=1  export CONT=1  export ENDYR=50  export STYR=${YEAR}  export RBLD="$WORK/nemo\_4.2.1/tools/REBUILD" # rebuild directory  export BIN="$WORK/RUNS/nemo\_4.2.1" # xml files, namelist files and nemo.exe  export XBIN="$WORK/trunk/bin" # xios directory  export OUTDIR="/scratch/projectname/usrname/NEMO\_OUT/${ENAM}" # NEMO output directory  export RUNDIR="/g/data/projectname/usrname/nemo\_inputs" # NEMO input files  export RUN="${OUTDIR}/OUT" # NEMO working directory  echo ==========================================================  echo Job submitted date = `date`  date\_start=`date +%s`  hostname  mkdir -p ${OUTDIR}  mkdir -p ${RUN}  cd ${RUN}  echo 'In directory: ' $RUN  # copy the restart file for the first year  cp $WORK/nemo\_inputs/restarts/nemo4.2.1\_restart.nc ${OUTDIR}/${ENAM}\_restart\_Y${STYR}.nc  if [ $CONT -eq 1 ]; then  # check if there is a more recent year that has already been written  cd ${OUTDIR}  restfile=`ls -t ${ENAM}\_restart\_Y\*.nc | head -1`  suffix="${restfile##\*[0-9]}"  number="${restfile%"$suffix"}"  number="${number##\*[!-0-9]}"  if [ $YEAR -eq $number ]; then  echo "Beginning from first year $YEAR"  else  export YEAR=$number  echo "Re-initialising from newer restart file '${restfile}' at year $YEAR"  fi  cd ${RUN}  else  echo "Beginning fresh new run from year ${YEAR}"  fi  # below here is specific for each year:  while [ ${YEAR} -le ${ENDYR} ] ; do  export NEXT=`expr $YEAR + 1`  echo 'YEAR='  echo $YEAR  echo 'NEXT='  echo $NEXT  ###########################################  source $WORK/trunk/arch/arch-X64\_GADI.env  # clean rundir  rm -r ${RUN}/\*  ln -sf ${RBLD}/rebuild ./  ln -sf ${RBLD}/flio\_rbld.exe ./  ln -sf ${BIN}/namelist\_pisces\_ref ./  ln -sf ${BIN}/namelist\_ref ./  ln -sf ${BIN}/namelist\_top\_ref ./  ln -sf ${BIN}/namelist\_cfg ./  ln -sf ${BIN}/namelist\_pisces\_cfg ./  ln -sf ${BIN}/namelist\_top\_cfg ./  ln -sf ${BIN}/axis\_def\_nemo.xml ./  ln -sf ${BIN}/domain\_def\_nemo.xml ./  ln -sf ${BIN}/context\_nemo.xml ./  ln -sf ${BIN}/field\_def\_nemo-oce.xml ./  ln -sf ${BIN}/field\_def\_nemo-pisces.xml ./  ln -sf ${BIN}/grid\_def\_nemo.xml ./  ln -sf ${BIN}/iodef.xml ./  ln -sf ${BIN}/nemo.exe ./nemo.exe  ln -sf ${BIN}/file\_def\_nemo-pisces.xml  # PISCES forcings  ln -sf ${RUNDIR}/\*.nc .  ln -sf ${OUTDIR}/${ENAM}\_restart\_Y${YEAR}.nc restart\_trc.nc  # check situation of RUNDIR  ls -lrt  #######################################  # run model  export OMP\_NUM\_THREADS=1  module load openmpi/4.1.5  time mpirun -np $RUNPROC ./nemo.exe  # check if the run was successful (if not, then exit and don't waste my time)  MESSAGE1=`grep " E R R O R" ocean.output`  MESSAGE2=`grep "NaN" ocean.output`  if [[ -n $MESSAGE1 ]]; then  echo " E R R O R "  echo " Abnormal run "  echo " Error in running GCM "  exit  elif [[ -n $MESSAGE2 ]]; then  echo " E R R O R "  echo " Abnormal run "  echo " NaNs in the final statistics of tracers "  exit  else  echo " YEAR $YEAR terminated normally "  fi  # run done  cd ${RUN}  # save output:  ./rebuild -o PISCES\_1y\_00010101\_00011231\_ptrc\_T.nc PISCES\_1y\_00010101\_00011231\_ptrc\_T\_0???.nc  ./rebuild -o PISCES\_1y\_00010101\_00011231\_diad\_T.nc PISCES\_1y\_00010101\_00011231\_diad\_T\_0???.nc  mv PISCES\_1y\_00010101\_00011231\_ptrc\_T.nc ${OUTDIR}/${ENAM}\_1y\_ptrc\_Y${YEAR}.nc  mv PISCES\_1y\_00010101\_00011231\_diad\_T.nc ${OUTDIR}/${ENAM}\_1y\_diad\_Y${YEAR}.nc  ./rebuild -o PISCES\_1m\_00010101\_00011231\_ptrc\_T.nc PISCES\_1m\_00010101\_00011231\_ptrc\_T\_0???.nc  ./rebuild -o PISCES\_1m\_00010101\_00011231\_diad\_T.nc PISCES\_1m\_00010101\_00011231\_diad\_T\_0???.nc  mv PISCES\_1m\_00010101\_00011231\_ptrc\_T.nc ${OUTDIR}/${ENAM}\_1m\_ptrc\_Y${YEAR}.nc  mv PISCES\_1m\_00010101\_00011231\_diad\_T.nc ${OUTDIR}/${ENAM}\_1m\_diad\_Y${YEAR}.nc  # mv \*1d\*bioscalar\* ${OUTDIR}/${ENAM}\_1d\_bioscalar\_Y${YEAR}.nc  ./rebuild -o PISCES\_00001460\_restart\_trc.nc PISCES\_00001460\_restart\_trc\_0???.nc  mv PISCES\_00001460\_restart\_trc.nc ${OUTDIR}/${ENAM}\_restart\_Y${NEXT}.nc  # increment year  export YEAR=$NEXT  echo Job finised date = `date`  done |

Contact: ming.cheng@anu.edu.au