Procedure for Rotation of POP Grid for CCSM3

Used for Geographies With No Land at the Poles

Developed by: C. Shields and S. Yeager 2008

Successfully applied to Ordovician Geography

Document author: C. Shields, June, 2011

Motivation: Applying ocean or coupled models to paleogeographies containing no polar land can be problematic with respect to numerical grid pole placement. Because all longitudes converge at the pole, numerically no computation can be done at this convergence point. (Atmospheric models solve this problem by using numerical filters at the poles). Ocean models typically resolve this issue by shifting the grid pole away from the geographical pole and locating it over a land mass. This approach is troublesome for some geologic periods, such as the late Ordovician, because of the lack of land at the polar latitudes. Historically, for the Ordovician, which has no land north of ~30^{o}N, the solution has been to create a very small artificial island at the geographic North Pole for the numerical pole to reside.

In an effort to represent the paleogeography for such periods as accurately as possible, a method has been developed for POP (CCSM3 ocean component) to locate the ocean's numerical poles on land without manufacturing islands where they did not exist in nature. The method assumes land masses do exist in the mid latitudes or equatorial regions. POP's mesh grid is rotated 90 degrees thereby placing the numerical poles over other available land regions. Further translation of the grid may be required to shift the poles E/W to ultimately hit land. This approach is unique to each geological period and requires several iterations to accomplish the ideal numerical pole placement. Some adjustment to the POP code is required.

This document describes the steps required to rotate/translate your mesh and then place your KMT (ocean bathymetry) on top of the rotated mesh. Also discussed are the required model source code modifications as well as some common runtime issues.

This document is to be used as a GUIDE and in conjunction with the standard paleoclimate documentation, NCAR TechNote, "Using CCSM3 for Paleoclimate Applications" (Rosenbloom, N., Shields, C., Brady, E., Levis, S., and Yeager, S., 2011, NCAR/TN-483+STR, 81pp.) as well as the standard setup tools. The grid rotation setup tools can be placed on the NCAR ftp server by request. Please contact C.Shields (shields@ucar.edu) for any of these materials if needed.

A. Creating your POP forcing files

1. Create your POP Mesh: *mk_nsdipole.csh*

a. Compute mesh using n/s dipole fortran code (*ns_dipole.f*). This can be found in your standard CCSM3 setup tools tarball. An example *mk_nsdipole.csh* script is supplied with the supplementary grid rotation tools to show you how to run this code separately from *mk_grid.csh*. Normally, for standard grids, the script *mk_grid.csh* is used and includes three components, 1) *ns_dipole.f*, 2) *paleotopo.f90*, and 3) *grid_bin2nc.f90*. However, for the grid rotation method, each step needs to be done separately to handle the iterative nature of creating the rotated mesh. Therefore, instead of *mk_grid.csh*, three independent scripts will be used, *mk_nsdipole.csh*, *mk_paleotopo.csh*, and *mk_lookatmesh.rotated.csh*.

i. below are some suggested settings for a generic 90 degree rotation:

```
 90N/S 0E for pole points
 nlats = 58 + 58  (nh/sh), nx = 100
dyeq = 3  (3 degrees, no equatorial enhancement)
dsiq  = de-activicated (code mod), no eq. enhancement*
 jcon = 11 (pole lines, small poles)
 topo/bath adjustments? **
```

ii. * edit *ns_dispole.f*: comment out all lines with "*fy = dsig*pi/180.d0*". You do not want to use the standard equatorial enhancement as it is defined on the normal grid. It is not appropriate for rotated grids.

iii. ** you may want to adjust your land by a few grid points once you see how the KMT lies on the mesh after all of the steps have been completed for the first time. (For the Ordovician case, even after translation, I needed to move one island 2 pop grid points east to find the best land/pole placements. Optimal distance between poles after rotation/translation is 180 degrees).

2. Rotate/Translate your POP Mesh: *meshrotate.*.ncl*

a. Run *meshrotate.ncl/meshrotate.plotda.ncl*. (Found in supplementary grid rotation tools).

b. Run *meshrotate.iter.ncl/meshrotate.plotda.iter.ncl* (Scripts for translation, or to iterate on a.).

These scripts do the following:

i. Convert dipoles from spherical to cartesian coordinates in preparation to rotate the grid.

```
;spherical(r,phi,theta) to cartesian(x,y,z)
; phi is colatitude where phi = 90. - latitude
; x = rcos(theta)sin(phi)
; y = rsin(theta)sin(phi)
; z = rcos(phi)
; r = sqrt((x2 + y2 + z2))
; phi = acos(z/r)
; theta = atan(y/x) where ncl/fortran uses atan2 (for 0 to 2pi systems)
```

ii. Rotate 90 degrees about x and/or z axis and translate as needed

```
;============ apply 90 rotation to all data in new x,y space
; use orthogonal transformation;
; x' = xcos(deg) + ysin(deg)
; y' = -xsin(deg) + y cos(deg)
; rotation (counterclockwise)
; x' = xcos(deg) - y sin(deg)
; y' = xsin(deg) + y cos(deg)
; reflection
; x' = xcos(deg) + ysin(deg)
; y' = xsin(deg) - ycos(deg)
;=== siggraph.org ref:
; xyz space, 3d... (for rotation choice counterclockwise)
;z axis rotation: (2d rotation case, as above references)
;x' = x*cos q - y*sin q
;y' = x*sin q + y*cos q
;z' = z
;x axis rotation:  x axis  w/y axis, y axis w/z, and z w/x
;y' = y*cos q - z*sin q
;z' = y*sin q + z*cos q
;x' = x
;y axis rotation:  x axis  w/zaxis, y axis w/x, and z w/y
;z' = z*cos q - x*sin q
;x' = z*sin q + x*cos q
;y' = y
```

iii. Convert back to spherical coordinates

```
;r = sqrt(x^2 + y^2 + z^2)
;phi = acos(z/r)
;theta = atan(y/x)
```

iv. Creates both pop grid and plotting grid used for step 3.

3. Place your KMT over your new rotated grid: *mk_paleotopo.csh*

Use *paleotopo.f90* as it was originally intended. See *mk_paleotopo.csh,* supplied in the grid_rotation tools, for an example way to run this code separately from *mk_grid.csh* in You are now placing your KMT on top of your rotated mesh to ultimately create your KMT file required by POP.

4. Iterate on steps 1 through 3 until you have a grid and KMT which work: *mk_lookatmeshrotated.csh*

Iterate as necessary. All tips on grid and KMT placement found in the basic documentation apply, (Rosenbloom et al, 2011). You may use *mk_lookatmesh.rotated.csh* found in your grid rotation tools to create a netcdf file suitable for viewing.

5. Edit your KMT as necessary: *kmtED*

Use *kmtED* (or alternate ncl scripts, both found in the standard setup tools) to edit your KMT as necessary according to the basic documentation (Rosenbloom et al, 2011).

6. Convert your grid and KMT files back to binary for POP and CPL mappings: *grid_nc2bin*

Use tool *grid_nc2bin* (or an alternate ncl tool, both found in the standard setup tools) to convert your final grid and kmt back to binary from netcdf. The POP model requires the binary files as does the tools to create the coupler mapping files.

Note: All of these tools are found in the setup/… directory in the grid rotation tools tarball.

B. POP/CSIM Code Adjustments

1. Discussion

For reference, please download a copy of the POP scientific description, “Reference Manual for the Parallel Ocean Program (POP): Ocean Component of the Community Climate System Model (CCSM2.0 and 3.0)”, Smith and Gent, Eds., 2004. (http://www.cesm.ucar.edu/models/ccsm3.0/pop/). Section3.1 and Figure 1 explain the POP grid.

The rotated grid method essentially redefines the grid box, (i becomes j and j becomes i), assuming a 90 degree rotation is done. Therefore, grid box length variables need to be re-defined. Note that POP and CSIM share the same grid, however, due to differences in the methodology used to compute gradients and advection, the rotated POP model requires a different definition of grid box length, dx and dy, compared to CSIM. CSIM uses a true 90 degree rotation where POP requires variable swaps (HTN/HTE) due to hardwired i and j (x and y) specifications. It is also important to note that aside from these hardwired definitions of i and j found in the grid specifications, only the anisotropic viscosity subroutine contains hardwired code for i and j. All other aspects of the model (POP and CSIM) can perform normally by simply making the adjustment at the grid level.

For further details on the grid structure, see page 8 in the POP reference manual. Original and 90^o rotated grid box length definitions are found in the below table.

Original	Definition	Definition	Adjusted	Adjusted
Variable	Length	Direction	POP	CSIM (pure 90 shift)
dxt = .5[$HTN_{i,j}$+$HTN_{i,j-1}$]	HTN	South (j-1)	HTE n (j+1)	HTN w (i-1)
dyt = .5[$HTE_{i,j}$+$HTE_{i-1,j}$]	HTE	West (i-1)	HTN w (i-1)	HTE n (j+1)
dxu = .5[$HTN_{i,j}$+$HTN_{i+1,j}$]	HTN	East (i +1)	HTE e (i+1)	HTN s (j-1)
dyu = .5[$HTN_{i,j}$+$HTN_{i,j+1}$]	HTE	North (j+1)	HTN n (j+1)	HTE e (i+1)

2. Code Modifications

The grid rotation tools tarball has a directory structure that contains pop and csim code modifications. Swap out your code for the modified subroutines. Be sure to consult the README files for further details and for the CCSM3 code base tag upon which the modifications are based. If your code base is not the same, you will find a subdirectory for each model that supplies the original, non-modified code for each subroutine so that you can isolate the necessary changes and apply them to your version. Remember to save a copy of your original subroutine before applying the changes.

Optional: Included in the code part of the tarball are POP subroutines used to reduce model solution noise by applying fixes to the GM mixing scheme. These may be necessary for most rotated grids to produce a cleaner looking simulation, but aren't necessarily required to get the model running. These modifications are recommended.

Directory structure:

code/pop/rotate_grid/…	*grid.F* and *hmxi_aniso.F*
code/pop/noise_reduce/…	*advection.F, hmix_gm.F, horizontal_mix.F, tavg.F, vmix_kpp.F,* and the input template, **_tavg_contents*
code/pop/original/…	the same pop subroutines, but the original, non-modified versions
code/csim/rotate_grid/…	*ice_grid.F* and *ice_transport_remap.F*
code/csim/original	the same ice subroutines, but the original, non-modified versions.

C. Runtime Issues

1. POP_IN changes

a. Viscosity changes

In addition to the recommended changes to the pop_in file for standard paleoclimate (see Rosenbloom et al 2011), you may need to make some changes to the viscosity parameters to further reduce the model solution noise common for rotated grids.

When setting anisotropic mixing options in the pop namelist, (*hmix_anis_nml)*, the coefficient options are:

```
vconst_1  backround
      _2 cos lat weighting
      _3 proportionality constant
      _4 length of exponential decay after grid pts
      _5 grid pts
      _6 backround
```

For the Ordovician test case, we modified the exponential decay constants to create a smoother field, i.e.

```
vconst_4   = 1.0e-9
vconst_5   = 5
```

b. GM mixing option

If the gm modifications are applied to reduce noise, the added bonus is that these new subroutines also include the addition of the bolus (eddy-induced) component to the meridional overturning and heat transport diagnostics. If you wish to include these in your output, in *hmix_gm_nml* , set

```
gm_bolus = .true.
```

2. CSIM namelist change

You may need to increase the dynamic ice step for csim. If so, in *csim.buildnml_prestage.csh,* set

```
ndyn_dt = 2
```

3. Time step changes

You will likely need to decrease your atmosphere, land, and ocean timesteps for the rotated grid. This grid is computationally more challenging than the standard paleoclimate grid, so a faster timestep may be required. See the standard paleoclimate documentation (Rosenbloom et al 2011), for instructions.

4. Machine Task Geometry

Because of the challenging grid, the default processor distribution given to you by the released code may not run your code efficiently and will most likely need to be improved. Below is an example of an ocean-heavy processor distribution that was used for the Ordovician on NCAR's IBM Bluefire. Essentially, assuming you are running at low resolution, i.e. T31_gx3, your ocean grid will now be the most expensive part of the system, not the atmosphere, as in the default T31_gx3 model. Modifications to processor distribution are made in the *env_mach.<machine>* file in your case directory.

NCAR BLUEFIRE (IBM)

Model	Task	Thread
Atmosphere	12	3
Ocean	12	1
Ice	4	1
Land	6	1
Coupler	6	1

GOOD LUCK and HAVE FUN!

Further consultation, contact C. Shields (shields@ucar.edu).