Setting up Regional MOM6

Kate Hedstrom
UAF
New Domains

• Need to make an orthogonal grid in some coordinate system
  • Rectangular
  • Curvilinear

• One way is to build in flat space using conformal map projections
  • Mercator
  • Lambert conformal conic
  • Polar stereographic
Some Options

- Pavel Sakov’s code: https://github.com/phobson/pygridgen
- Gridbuilder: https://austides.com/downloads/
- Liz and Alistair’s new thing
- WRF Domain Wizard
- My weird old stuff (from John Wilkin)
Example Domains
US West Coast

dy
Arctic 1
Southern Africa
NEP 7
Bering Sea
(WRF grid by Rob Cermak)
Fields on the Grid

• Bathymetry
  • Remap from the best you can get for your region
  • Many global products are derived from satellites and have odd pits and bumps
  • NEP, Bering, and Arctic bathymetry from IBCAO4 where it exists, ARDEM2 where it exists, SRTM30 for the rest
Land Mask

- Sasha Shchepetkin’s tools:
  http://www.atmos.ucla.edu/~alex/ROMS/tools.tar
- gshhs_to_roms_mask roms_grid.nc
- copymask mask.nc roms_grid.nc
- single_connect i0 j0 roms_grid.nc
- These are Fortran, specific to ROMS files
Land Mask

- PYROMS contains editmask (https://github.com/ESMG/pyroms) GUI for editing the land mask
- There’s a matlab version of this too: https://www.myroms.org/wiki/Land_Sea_Masking_Scripts
Land Mask

• Need to avoid little bays at the boundary:
Editing False -- click "e" to toggle
All Models Want the Same Fields

- I build ROMS grids, convert to whatever:
  - CICE/POP
  - MOM6
  - CESM
- I also have a WRF to ROMS converter
- Check angle across dateline!
After the Grid

- **OBC/IC files**
  - https://github.com/ESMG/PyCNAL_regridding
  - https://github.com/ESMG/regionalMOM6_notebooks
- **Forcing**
- **Runoff**
Runoff from JRA

year=1990

```
python regrid_runoff.py ocean_hgrid.nc \ 
  ocean_mask.nc \ 
  /import/c1/AKWATERS/kate/JRA55-do/runoff_JRA55-do-1-4-0_${year}.nc \ 
  Arctic5_runoff_${year}.nc -z \ 
  --regional_domain -r friver --progress \ 
  --fast_pickle --fms
```

```
python modify_regrid_output.py ${year}
```
MOM6 ascii inputs

- data_table
- diag_table
- MOM_input
- input.nml
<table>
<thead>
<tr>
<th>variable</th>
<th>file path</th>
<th>interpolation</th>
<th>scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM</td>
<td>/center1/AKWATERS/kate/JRA55-do/JRA55DO_1.4_Pair_2011.nc</td>
<td>bilinear</td>
<td>1.0</td>
</tr>
<tr>
<td>OCN</td>
<td>./INPUT/runoff_file.nc</td>
<td>none</td>
<td>1.0</td>
</tr>
<tr>
<td>ICE</td>
<td>/center1/AKWATERS/kate/JRA55-do/JRA55DO_1.4_lwrad_down_2011.nc</td>
<td>bilinear</td>
<td>1.0</td>
</tr>
</tbody>
</table>


MOM_input OBC Options

• All spelled out at: https://github.com/NOAA-GFDL/MOM6-examples/wiki/Open-Boundary-Conditions
• Subject to change
• Some are per segment, some apply to all segments
• Let us know if it is unclear!
Open Boundary Segments

- **Tell how many:**
  
  \[ \text{OBC\_NUMBER\_OF\_SEGMENTS} = 2 \]

- **Where and what kind:**

  \[
  \begin{align*}
  \text{OBC\_SEGMENT\_001} &= \text{"I=N, J=0:N, FLATHER"} \\
  \text{OBC\_SEGMENT\_002} &= \text{"I=0, J=N:0, SIMPLE"}
  \end{align*}
  \]

- “N” is code for “end”

- Index order (direction) matters!
Bering MOM_input

OBC_NUMBER_OF_SEGMENTS = 4
OBC_FREESLIP_VORTICITY = False
OBC_COMPUTED_VORTICITY = True
OBC_FREESLIP_STRAIN = False
OBC_COMPUTED_STRAIN = True
OBC_ZERO_BIHARMONIC = True
Bering MOM_input

OBC_SEGMENT_001 = "J=N,I=N:0,FLATHER,ORLANSKI,NUDGED,ORLANSKI_TAN,NUDGED_TAN"

OBC_SEGMENT_002 = "I=0,J=N:0,FLATHER,ORLANSKI,NUDGED,ORLANSKI_TAN,NUDGED_TAN"

OBC_SEGMENT_003 = "J=0,I=0:N,FLATHER,ORLANSKI,NUDGED,ORLANSKI_TAN,NUDGED_TAN"

OBC_SEGMENT_004 = "I=N,J=0:N,FLATHER,ORLANSKI,NUDGED,ORLANSKI_TAN,NUDGED_TAN"
Bering MOM_input

OBC_SEGMENT_001_VELOCITY_NUDGING_TIMESCALES = 3, 360.0

BRUSHCUTTER_MODE = True

OBC_SEGMENT_001_DATA =
"U=file:Bering_OBC.nc(u), V=file:Bering_OBC.nc(v), SS H=file:Bering_OBC.nc(zeta), TEMP=file:Bering_OBC.nc (temp), SALT=file:Bering_OBC.nc(salt)"

OBC_TRACER_RESERVOIR_LENGTH_SCALE_OUT = 3000.0

OBC_TRACER_RESERVOIR_LENGTH_SCALE_IN = 3.0E+04
Future Work

- Tides
- Better Python tools for pre- and post-processing
- Landfast ice parameterizations
- Algorithmic stability