

[Edit on GitHub](#)

Here are recipes for doing some xarray-based analysis with MOM6.

- Setting up a DASK cluster using dask-jobqueue
 - Your DASK cluster at work
- Setting up a DASK cluster on your local machine
 - Sample computation:
- Getting started with MOM6
 - grid variables
 - building a xgcm grid object
 - A note on geographical coordinates
 - Plotting
- Time-based operations
 - 1. Computing climatologies for SST
 - 2. Selecting based on dates
- Spatial Operations
 - 2D horizontal averaging
 - Zonal average
 - 3D average
 - Using xgcm
- Vorticity-based diagnostics
 - Relative vorticity
 - Potential vorticity $(\zeta + f)/h$
- Computations for Potential density, buoyancy and geostrophic shear
 - Potential density
 - Buoyancy
 - Geostrophic shear
- Horizontal Remapping
 - Remapping model output to a 1x1 degree grid
 - Remapping onto the model grid
- Creating nice maps with xarray
 - Polar projections
- Comparing MOM6 data to hydrographic section



Useful doc for MOM6

<https://xgcm.rtfd.io>

 **XGCM** 0.3 Site ▾ Page ▾ » MITgcm ECCOV4 Example Generate miss... »


Getting started with xgcm for MOM6

[xgcm grid definition](#)

[A note on geographical coordinates](#)

[Vorticity computation](#)

[Plotting](#)

 [Edit on GitHub](#)

GETTING STARTED WITH XGCM FOR MOM6

- MOM6 variables are staggered according to the Arakawa C-grid
- It uses a north-east index convention
- center points are labelled (xh, yh) and corner points are labelled (xq, yq)
- important: variables xh/yh, xq/yq that are named "nominal" longitude/latitude **are not** the true geographical coordinates and are not suitable for plotting (more later)

See [indexing](#) for details.

```
[1]: import xarray as xr
from xgcm import Grid
import warnings
import matplotlib.pyplot as plt
from cartopy import crs as ccrs
import numpy as np
```

```
[2]: %matplotlib inline
warnings.filterwarnings("ignore")
```

For this tutorial, we are going to use sample data for the $\frac{1}{2}^\circ$ global model OM4p05 hosted on a GFDL thredds server:

```
[3]: dataurl = 'http://35.188.34.63:8080/thredds/dodsC/OM4p5/'

ds = xr.open_dataset(f'{dataurl}/ocean_monthly_z.200301-200712.nc4',
                    chunks={'time': 1, 'z_l': 1}, drop_variables=['average_DT',
                                                                    'average_T1',
                                                                    'average_T2'],
                    engine='pydap')
```

```
[4]: ds
```

```
[4]: <xarray.Dataset>
Dimensions:      (nv: 2, time: 60, xh: 720, xq: 720, yh: 576, yq: 576, z_i: 36, z_l
Coordinates:
  * nv           (nv) float64 1.0 2.0
  * xh           (xh) float64 -299.8 -299.2 -298.8 -298.2 ... 58.75 59.25 59.75
  * xq           (xq) float64 -299.5 -299.0 -298.5 -298.0 ... 59.0 59.5 60.0
```