Using CESM Simple Models Toolkit
for Idealized MOM6 Applications

Alper Altuntas, *Scott Bachman, Isla Simpson, Samuel Levis, Brian Dobbins, Gokhan Danabasoglu*

NCAR

2023 OMWG Winter Meeting

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

Overview

Goal:

- Streamline coupled, simple modeling within CESM.
- Enable hierarchical modeling: Provide a user-friendly modeling framework allowing users to adjust the model complexity.

Tools:

- 1. **visualCaseGen**.
  - A GUI for creating custom CESM experiments.
- 2. **mom6_bathy**
  - Python library for generating idealized MOM6 grid and bathymetry input.
- 3. **CLM tools**
  - Surface dataset and mask modification.

Funded by an NSF CSSI award. (PIs: Bachman, Simpson)

Configuring an Idealized MOM6 experiment

GUI.ipynb

Code git Python [conda env:miniconda3-visualCaseGen]

▸ **Step 1: Select Mode**

▾ **Step 2: Create Case**

Initialization Time: 1850 | 2000 | HIST

Components:

▼ ATM	▼ LND	▼ ICE	▼ OCN	▼ ROF	▼ GLC	▼ WAV
datm	clm	cice5	pop	rtm	cism	ww3
satm	dlnd	cice	mom	mosart	sglc	ww3dev
cam	slnd	dice	docn	drof		dwav
		sice	socn	srof		swav

Physics and Options:

ATM | LND | ICE | OCN | ROF | GLC | WAV

Options will be displayed here after a component selection.

GUI.ipynb

Code git Python [conda env:miniconda3-visualCaseGen]

Ocean mesh: Start from scratch | Modify existing

Grid Extent: Regional | ✗ Global

Zonally reentrant? True | False

Meridionally reentrant? True | False

Number of cells in x direction: 100

Number of cells in y direction: 50

Grid length in x direction (degrees): 10.0

Grid length in y direction (degrees): 5.0

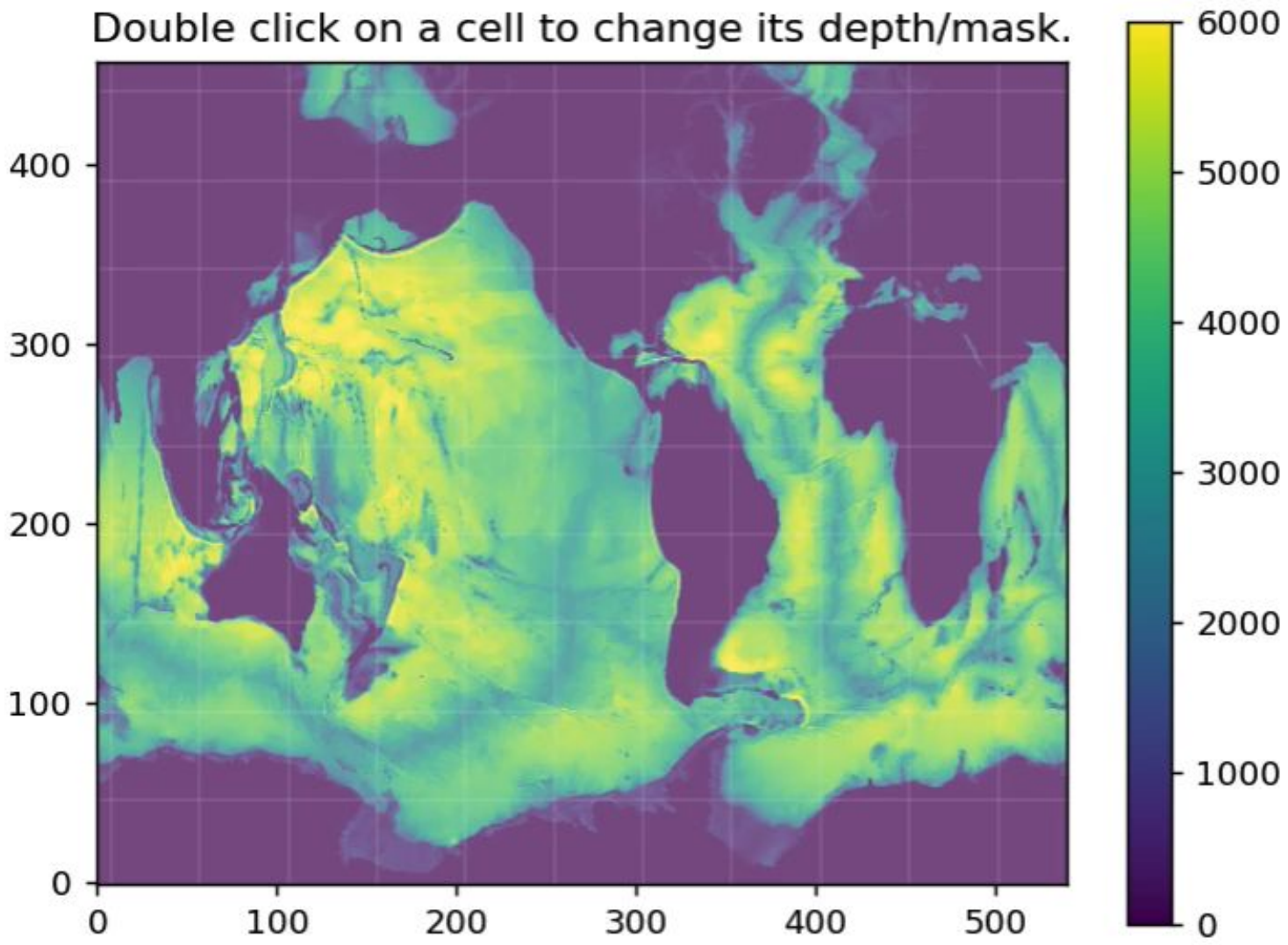
Launch mom6_bathy

Launch:

More Examples

Toggle display: Depth Mask

Figure 1

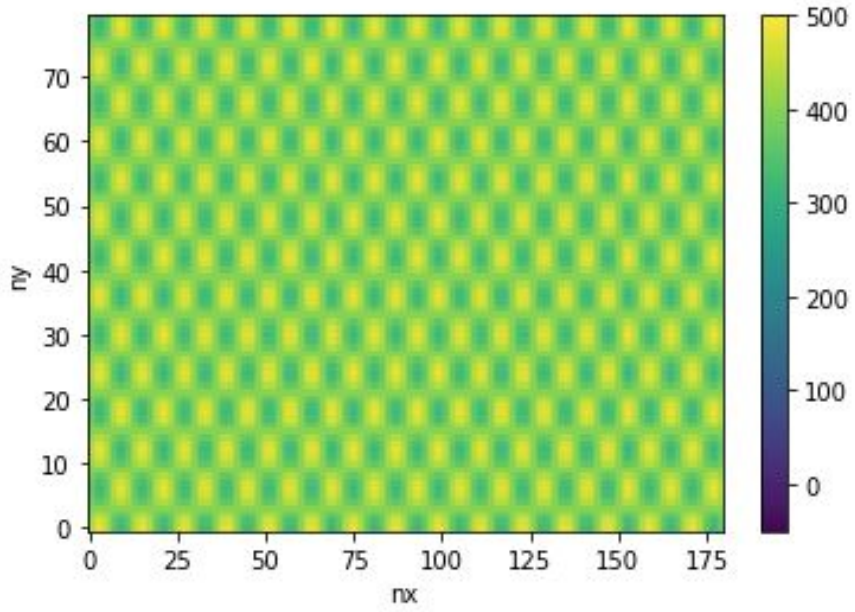


Define a custom bathymetry

In [4]:
```
# Define a sinusoidal depth array:
i = grd.tlat.nx.data                    # array of x-indices
j = grd.tlat.ny.data[:,np.newaxis]  # array of y-indices
custom_depth = 400.0 + 80.0 * np.sin(i*np.pi/6.) * np.cos(j*np.pi/6.)
bathy.set_depth_arr(custom_depth)
```

In [5]:
```
bathy.depth.plot(vmin=-50, vmax=500)
```

Out[5]: <matplotlib.collections.QuadMesh at 0x2aabedcc29b0>



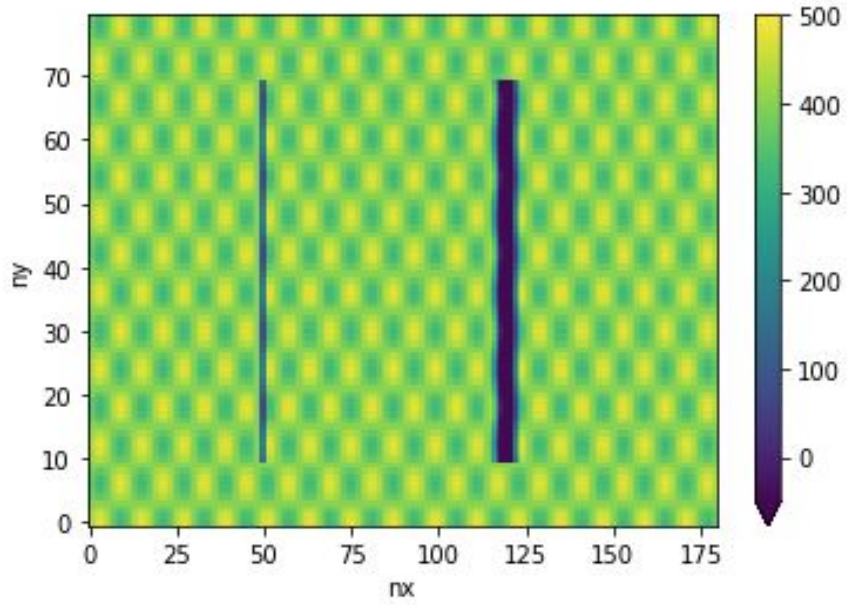
Apply ridges

In [6]:
```
# apply a submerged ridge ( height of 300 m):
bathy.apply_ridge(height=400.0, width=4.0, lon=100, ilat=(10,70) )
```

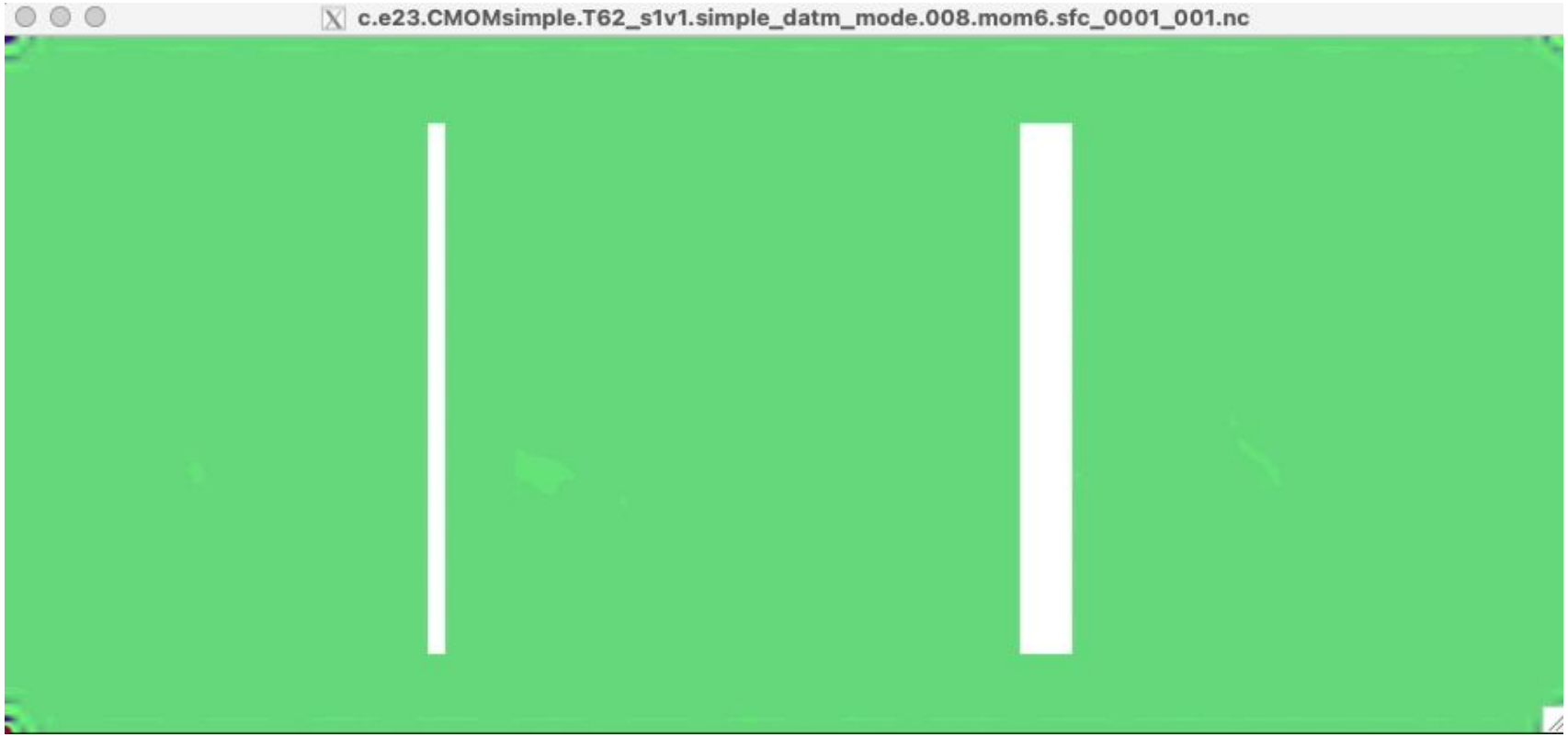
In [7]:
```
# apply a ridge reaching above MSL ( height of 600 m):
bathy.apply_ridge(height=600.0, width=16.0, lon=240, ilat=(10,70) )
```

In [8]:
```
bathy.depth.plot(vmin=-50, vmax=500)
```

Out[8]: <matplotlib.collections.QuadMesh at 0x2aabf5e5ee60>



SSV – First 5 days



LIMITATIONS

The scope of this toolchain is limited to:

- Compset
  - High level physics options
- Horizontal Grids
  - *MOM6*: grid and topography
  - *CTSM:* fsurdat and mask

The goal is to ensure consistency of high-level, inter-component settings.

Numerical stability, consistency, and physical faithfulness of individual components not guaranteed!

- It is up to the user to refine and tune:
  - IC, BC, Sponge layer, internal timesteps, etc.

How to Obtain

Github Repository:
github.com/ESMCI/visualCaseGen

Quickstart Documentation:
github.com/ESMCI/visualCaseGen/wiki/Quickstart

Next Steps

- Improve robustness and ease of use.
- CICE grid input files.
- Idealized topography inputs for CAM.
- Examples Gallery.
- Documentation.

- Wish list?

Thanks!
altuntas@ucar.edu

Appendices

GUI.ipynb

Code git Python [conda env:miniconda3-visualCaseGen]

▸ Step 1: Select Mode

▾ **Step 2: Create Case**

Initialization Time: 1850 | 2000 | HIST

Components:

▼ ATM	▼ LND	▼ ICE	▼ OCN	▼ ROF	▼ GLC	▼ WAV
datm	clm	cice5	pop	rtm	cism	ww3
satm	dlnd	cice	mom	mosart	sglc	ww3dev
cam	slnd	dice	docn	drof		dwav
		sice	socn	srof		swav

Physics and Options:

ATM | LND | ICE | OCN | ROF | GLC | WAV

Options will be displayed here after a component selection.

GUI.ipynb

Code git Python [conda env:miniconda3-visualCaseGen]

▸ Step 1: Select Mode

▾ **Step 2: Create Case**

Initialization Time: 1850 | 2000 | HIST

Components:

▼ ATM	▼ LND	▼ ICE	▼ OCN	▼ ROF	▼ GLC	▼ WAV
datm	clm	cice5	pop	rtm	cism	ww3
✗ satm	dlnd	cice	mom	mosart	sglc	ww3dev
cam	slnd	dice	docn	drof		✗ dwav
		sice	socn	srof		swav

Physics and Options:

ATM | LND | ICE | MOM | ROF | GLC | WAV

OCN physics: MOM6

GUI.ipynb

Code git Python [conda env:miniconda3-visualCaseGen]

▸ Step 1: Select Mode

▾ Step 2: Create Case

Initialization Time: 1850 | 2000 | HIST

Components:

▼ ATM	▼ LND	▼ ICE	▼ OCN	▼ ROF	▼ GLC	▼ WAV
datm	✗ clm	cice5	pop	✗ rtm	cism	ww3
✗ satm	✗ dlnd	cice	mom	✗ mosart	sglc	ww3dev
cam	slnd	dice	docn	drof		✗ dwav
		sice	socn	srof		swav

Physics and Options:

DATM | LND | ICE | MOM | ROF | GLC | WAV

ATM physics: DATM

GUI.ipynb

visualCaseGen]

ERROR: COMP_LND=clm violates assertion:"When MOM|POP is forced with DATM, LND must be stub."

OK

Components:

▼ ATM	▼ LND	▼ ICE	▼ OCN	▼ ROF	▼ GLC	▼ WAV
datm	❌ clm	cice5	pop	❌ rtm	cism	ww3
❌ satm	❌ dlnd	cice	mom	❌ mosart	sglc	ww3dev
cam	slnd	dice	docn	drof		❌ dwav
		sice	socn	srof		swav

Physics and Options:

DATM | LND | ICE | MOM | ROF | GLC | WAV

ATM physics: DATM

GUI.ipynb

Code | git | Python [conda env:miniconda3-visualCaseGen]

Components:

▼ ATM	▼ LND	▼ ICE	▼ OCN	▼ ROF	▼ GLC	▼ WAV
datm	✗ clm	cice5	pop	✗ rtm	cism	✗ ww3
✗ satm	✗ dlnd	cice	mom	✗ mosart	sglc	✗ ww3dev
cam	slnd	dice	docn	drof		✗ dwav
		sice	socn	srof		swav

Physics and Options:

DATM | SLND | SICE | MOM | SROF | SGLC | SWAV

WAV physics: SWAV

☑ (none) no modifiers

compset: **2000_DATM%QIA_SLND_SICE_MOM6_SROF_SGLC_SV**

Grids:

GUI.ipynb

Code git Python [conda env:miniconda3-visualCaseGen]

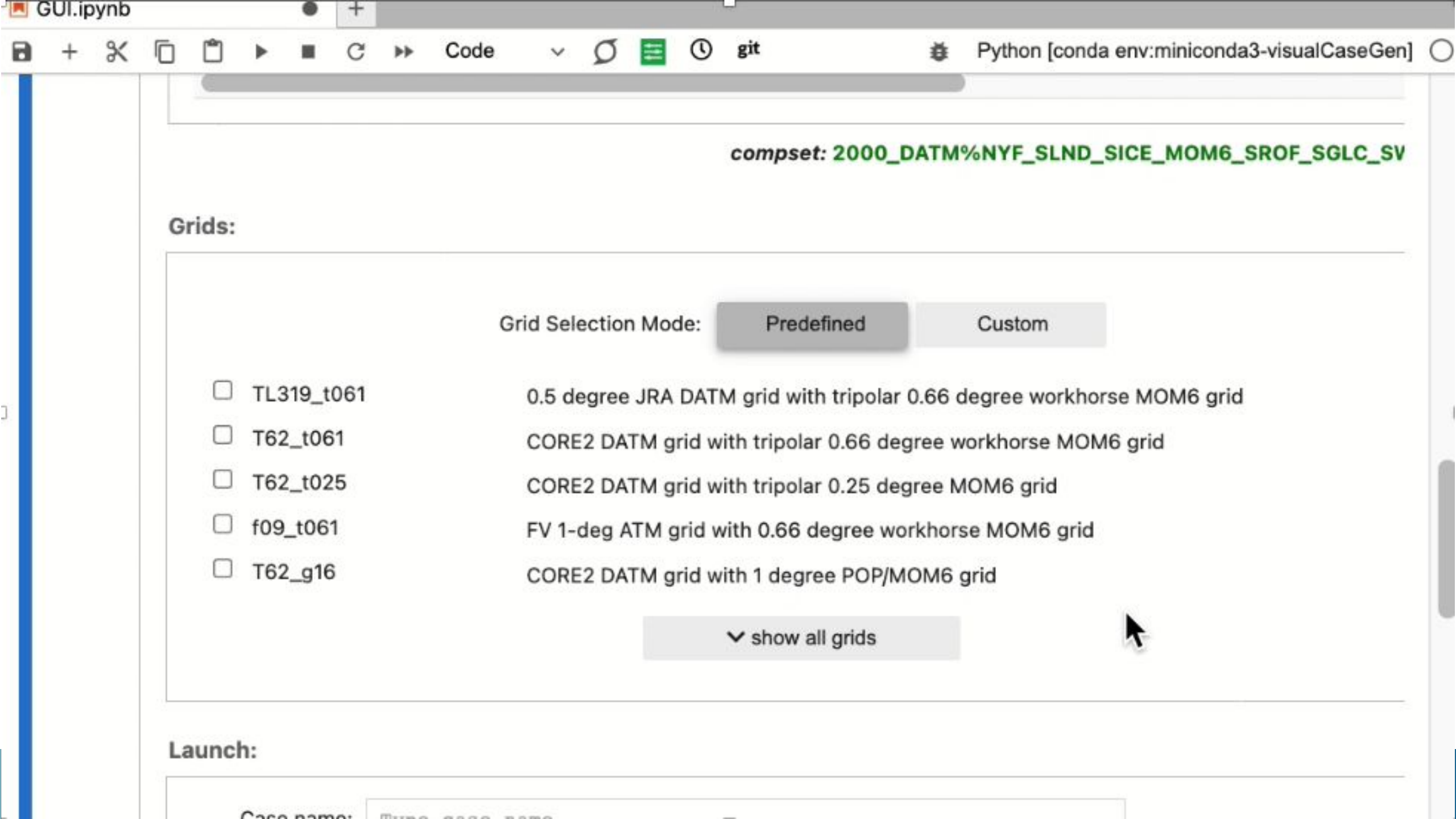
Physics and Options:

DATM | SLND | SICE | MOM | SROF | SGLC | SWAV

ATM physics: DATM

Selection: single multi

☐	❌ (none)	no modifiers
☑	QIA	QIAN data set
☐	WISOQIA	QIAN with water isotopes
☐	CRU	CRUNCEP data set
☐	CRUv7	CLM CRU NCEP v7 data set
☐	GSWP3v1	GSWP3v1 data set
☐	MOSARTTEST	no description
☐	NLDAS2	NLDAS2 regional 0.125 degree data set over the U.S. (25-53N, 235-293E). WARNING: Garba
☐	CPLHIST	Coupler hist data set (in this mode, it is strongly recommended that the model domain and th
☐	1PT	single point tower site data set


GUI.ipynb
Code
git
Python [conda env:miniconda3-visualCaseGen]
compset: 2000_DATM%NYF_SLND_SICE_MOM6_SROF_SGLC_SV
Grids:
Grid Selection Mode:
Predefined
Custom
TL319_t061
0.5 degree JRA DATM grid with tripolar 0.66 degree workhorse MOM6 grid
T62_t061
CORE2 DATM grid with tripolar 0.66 degree workhorse MOM6 grid
T62_t025
CORE2 DATM grid with tripolar 0.25 degree MOM6 grid
f09_t061
FV 1-deg ATM grid with 0.66 degree workhorse MOM6 grid
T62_g16
CORE2 DATM grid with 1 degree POP/MOM6 grid
show all grids
Launch:

GUI.ipynb

Code git Python [conda env:miniconda3-visualCaseGen]

MOM6 Grid

Ocean mesh: Start from scratch | Modify existing

Grid Extent: Regional | ✗ Global

Zonally reentrant? True | False

Meridionally reentrant? True | False

Global

Number of cells in x direction:

Number of cells in y direction:

Grid length in x direction (degrees):

Grid length in y direction (degrees):

Launch mom6_bathy

GUI.ipynb

-visualCaseGen]

ERROR: OCN_GRID_EXTENT=Global violates assertion:"LND or ICE must be present to hide Global MOM6 grid poles."

OK

Zonally reentrant?	True	False
Meridionally reentrant?	True	False

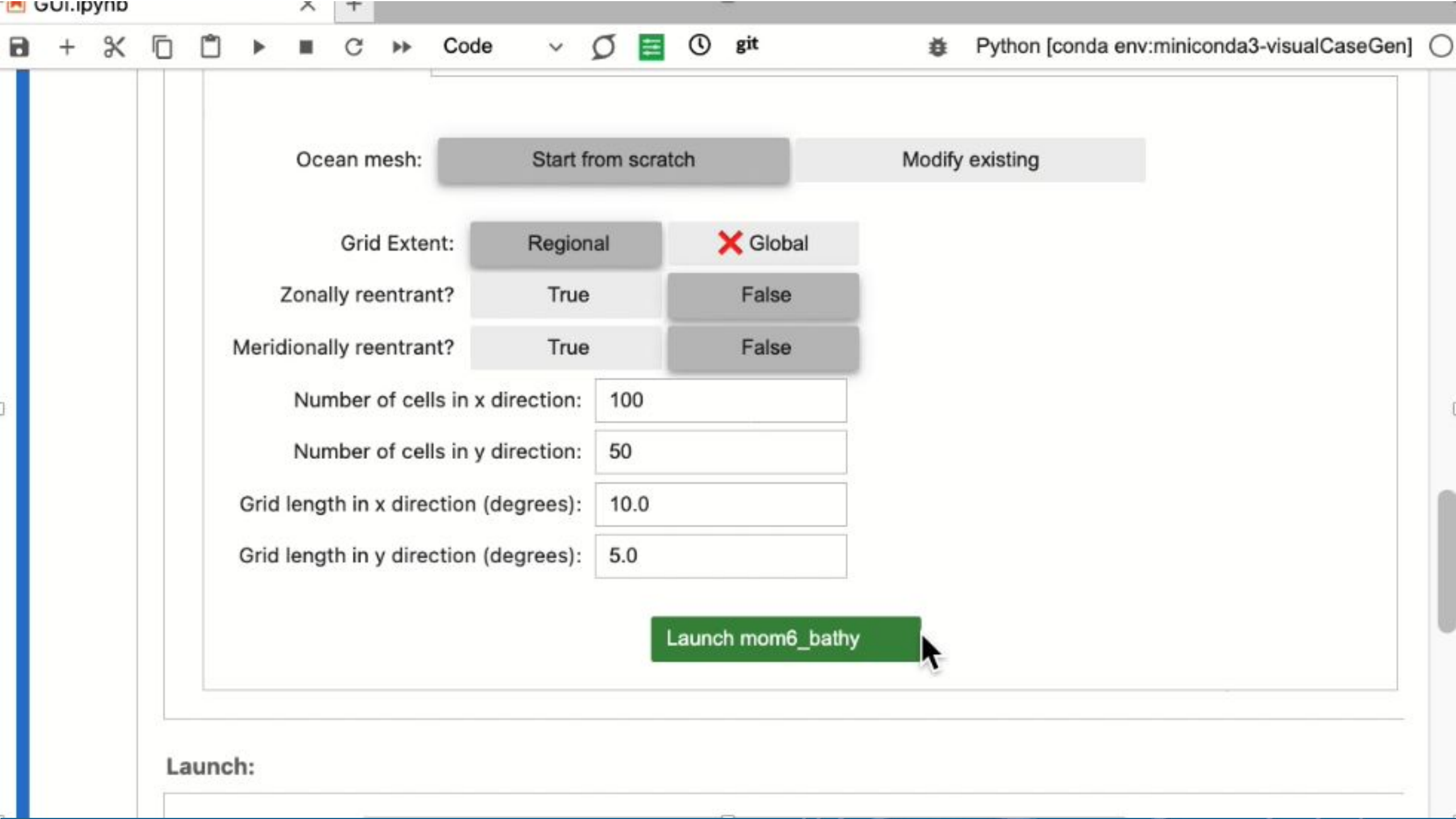
Number of cells in x direction:

Number of cells in y direction:

Grid length in x direction (degrees):

Grid length in y direction (degrees):

Launch mom6_bathy


GUI.ipynb
Code
git
Python [conda env:miniconda3-visualCaseGen]
Ocean mesh:
Start from scratch
Modify existing
Grid Extent:
Regional
Global
Zonally reentrant?
True
False
Meridionally reentrant?
True
False
Number of cells in x direction:
100
Number of cells in y direction:
50
Grid length in x direction (degrees):
10.0
Grid length in y direction (degrees):
5.0
Launch mom6_bathy
Launch:

Launcher | mom6_bathy_notebook_20

Markdown | git | Python [conda env:miniconda3-visualCaseGen]

mom6_bathy

This notebook is auto-generated by visualCaseGen GUI. Please review and execute all of the cells below to finalize your custom MOM6 grid.

1. Import mom6_bathy

[ ]:
```
%%capture
from mom6_bathy.mom6grid import mom6grid
from mom6_bathy.mom6bathy import mom6bathy
```

2. Create horizontal grid

[ ]:
```
grd = mom6grid(
                nx         = 100,         # Number of grid points in x direction
                ny         = 50,          # Number of grid points in y direction
                config     = "spherical",
                axis_units = "degrees",
                lenx       = 10.0,        # grid length in x direction, e.g., 360.0 (degrees)
                leny       = 5.0,        # grid length in y direction
```

Launcher | mom6_bathy_notebook_2C

Markdown | git | Python [conda env:miniconda3-visualCaseGen]

[1]:
```
%%capture
from mom6_bathy.mom6grid import mom6grid
from mom6_bathy.mom6bathy import mom6bathy
```

2. Create horizontal grid

[ ]:
```
grd = mom6grid(
                nx         = 100,          # Number of grid points in x direction
                ny         = 50,           # Number of grid points in y direction
                config     = "spherical",
                axis_units = "degrees",
                lenx       = 10.0,         # grid length in x direction, e.g., 360.0 (degrees)
                leny       = 5.0,         # grid length in y direction
                cyclic_x   = False,
                cyclic_y   = False,
                session_id   = "c53d9cac", # do not modify
                )
```

3. Configure bathymetry

mom6_bathy provides several idealized bathymetry options and customization methods. Below, we show how to specify the simplest bathymetry configuration, a flat bottom. Customize it as you see fit. See mom6_bathy

Launcher | mom6_bathy_notebook_20

Code | git | Python [conda env:miniconda3-visualCaseGen]

3. Configure bathymetry

mom6_bathy provides several idealized bathymetry options and customization methods. Below, we show how to specify the simplest bathymetry configuration, a flat bottom. Customize it as you see fit. See mom6_bathy documentation and example notebooks on how to create custom bathymetries.

[3]:
```
# Instantiate the bathymetry object
bathy = mom6bathy(grd, min_depth = 10.0)
```

[4]:
```
# Set the bathymetry to be a flat bottom with a uniform depth of 2000m
bathy.set_flat(D=2000.0)
```

[ ]:
```
bathy.depth.plot()
```

[ ]:
```
# Manually modify the bathymetry
%matplotlib ipympl
from mom6_bathy.depth_modifier import DepthModifier
DepthModifier(bathy)
```

4. Save the grid and bathymetry files

0:58.67

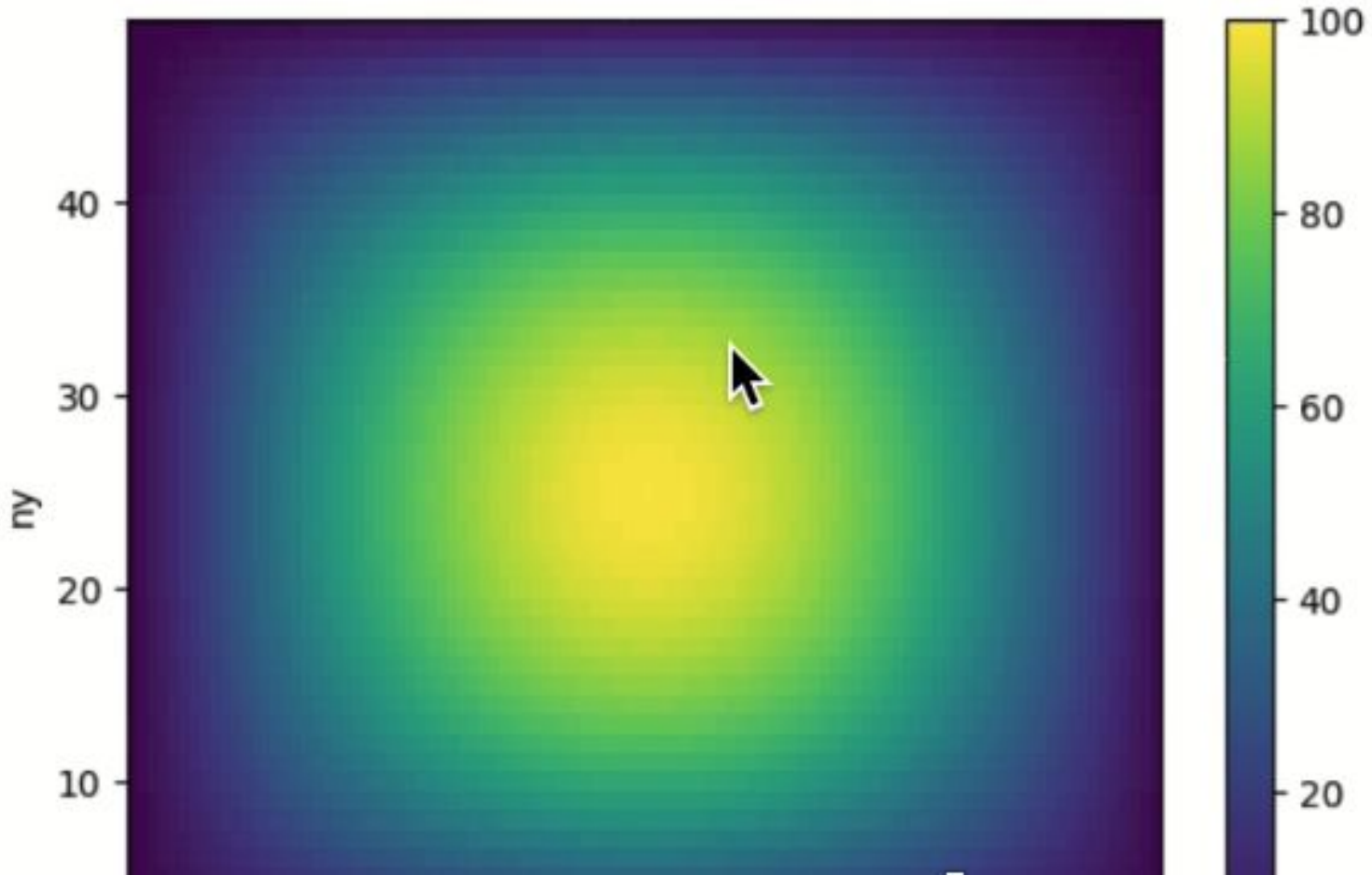
Launcher | mom6_bathy_notebook_2C

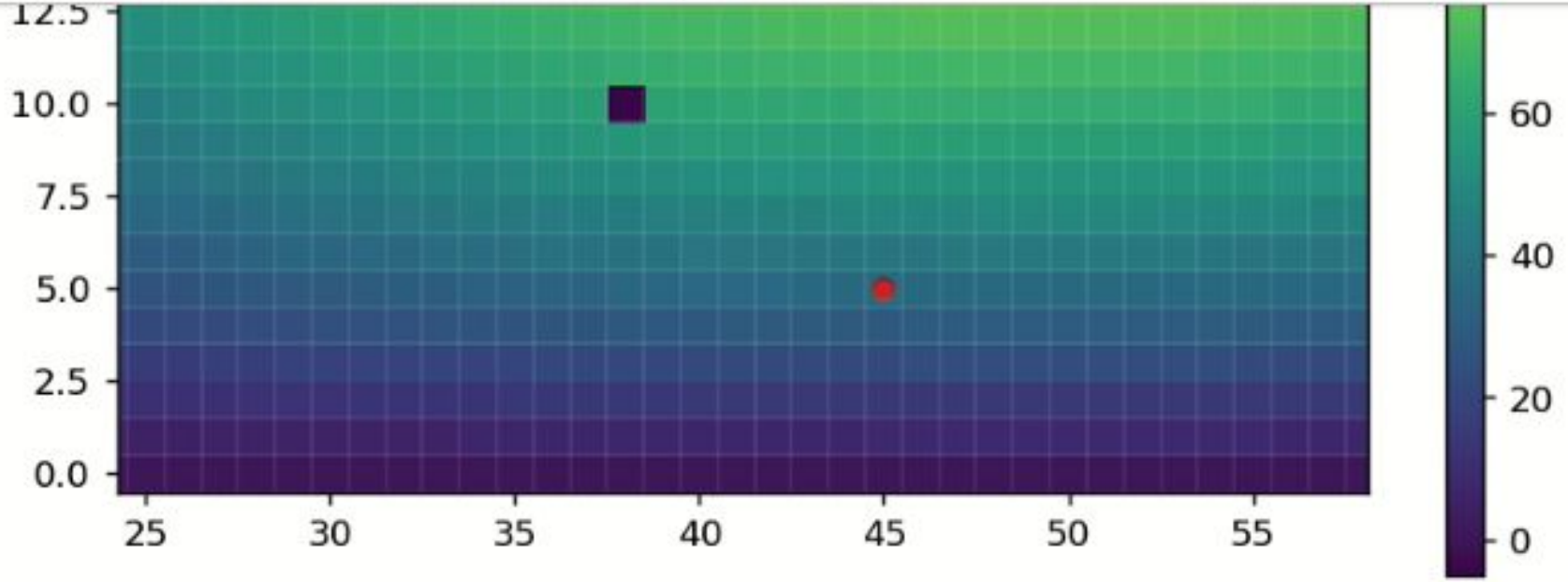
Code | git | Python [conda env:miniconda3-visualCaseGen]

[6]:
```
bathy.set_bowl(max_depth=100, dedge=0.0)
```

[8]:
```
bathy.depth.plot()
```

[8]:
```
<matplotlib.collections.QuadMesh at 0x7f69c8b04460>
```





Selected cell at i=45, j=5.

Current depth: 36

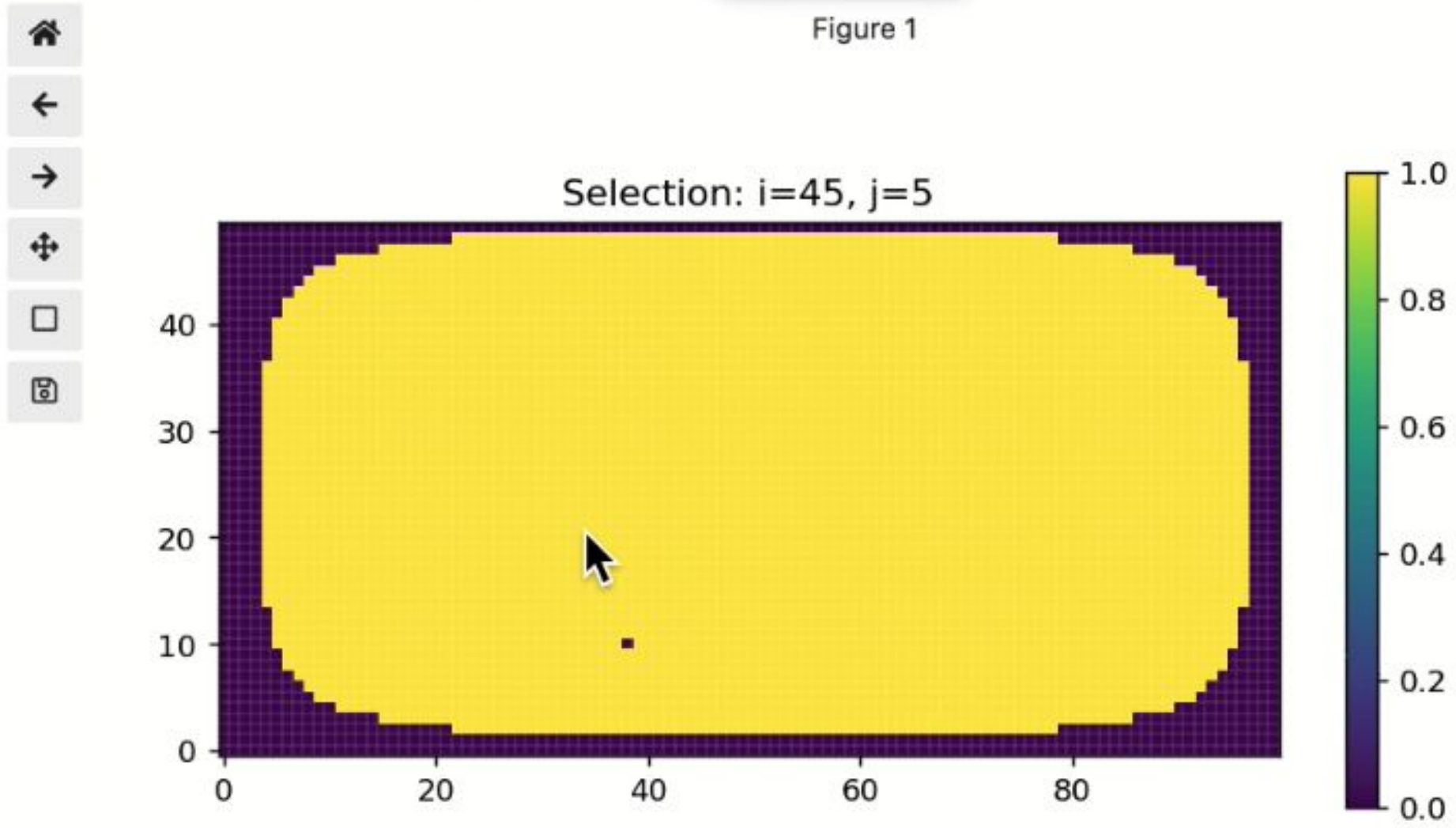
Enter new depth: 50

Apply

Save

Toggle display: Depth Mask

Figure 1



Launcher | mom6_bathy_notebook_20

Code | git | Python [conda env:miniconda3-visualCaseGen]

Current depth: 36

Enter new depth: 50 | Apply | Save | Saved!

4. Save the grid and bathymetry files

[10]:
```
# First, specify a unique name for your new grid, e.g.:
grid_name = "simple_1v1"

# Save MOM6 supergrid file:
grd.to_netcdf(supergrid_path = f"./ocean_grid_{grid_name}_20230208.nc")

# Save MOM6 topography file:
bathy.to_topog(f"./ocean_topog_{grid_name}_20230208.nc")

# Save ESMF mesh file:
bathy.to_ESMF_mesh(f"./ESMF_mesh_{grid_name}_20230208.nc")
```

```
SUCCESS! All necessary MOM6 input files are generated. You may now return to visualCaseGen to finaliz
e the case.
```

[ ]: