Python Data Analysis and Visualization for Unstructured Grid Data

Ian Franda[1,2], Orhan Eroglu[1], Philip Chmielowiec[1], Anissa Zacharias[1]
[1]National Center for Atmospheric Research
[2]University of Wisconsin Madison

August 2, 2023

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

EarthCube

PROJECT raijin

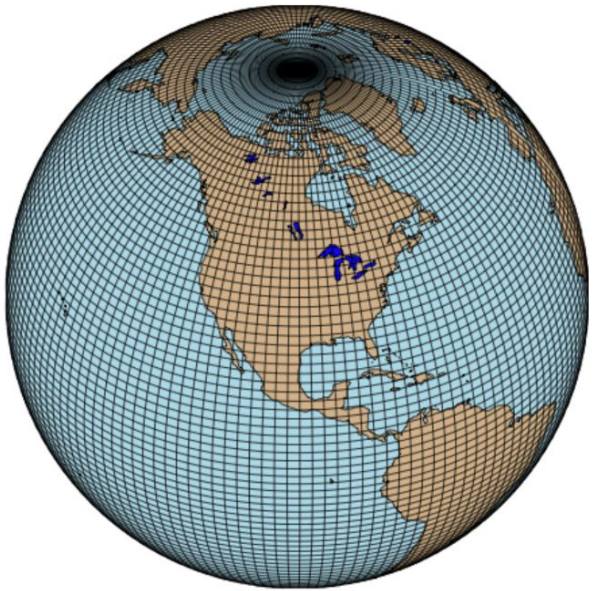
Overview

- **Geo**science **C**ommunity **A**nalysis **T**oolkit
- Develops data analysis and visualization tools for Earth System Science data
- Supported by National Science Foundation

- Enhance open-source development for unstructured grid data with community owned and developed tools
- Provide extensible and scalable tools for unstructured grid data analysis and visualization
- NSF EarthCube funded effort

Structured Grids

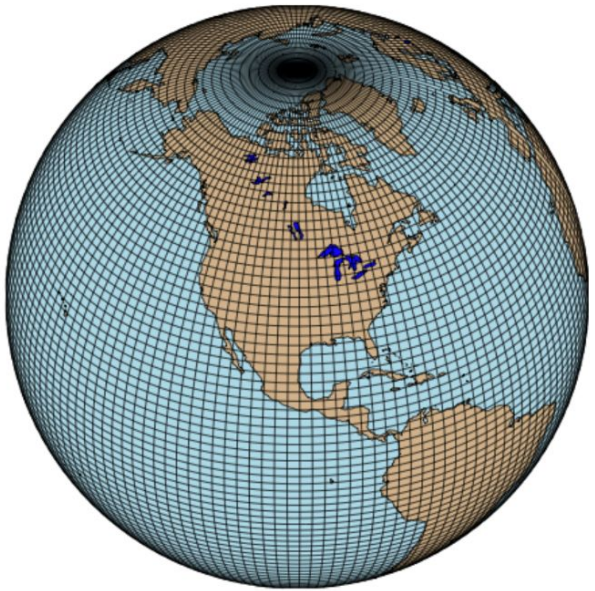
- Data stored in fixed resolution cells at regular intervals
- e.g. Cell at every degree longitude and latitude
- Extensively supported and used in the geoscience community



“Latitude-longitude” structured grid

Challenges with Structured Grids

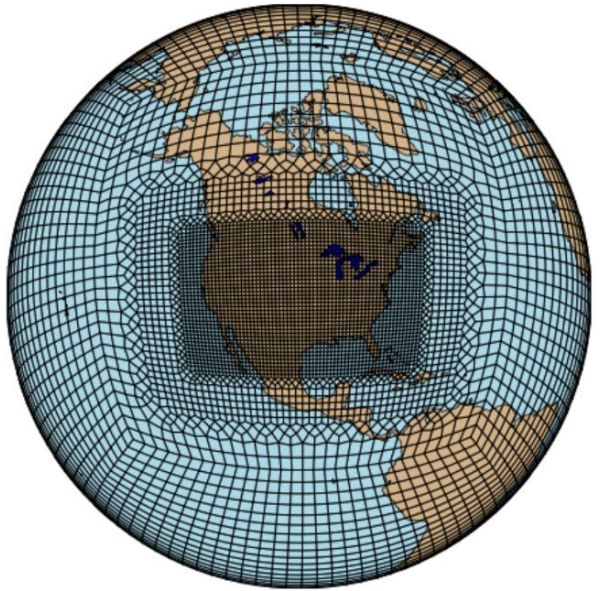
- Fixed Resolution - cells must have equal spacing
- In order to have high resolution somewhere we have to have high resolution everywhere
- Storm Resolving Resolutions (where cells are just a few square km) not practical across the entire Earth



“Latitude-longitude” structured grid

Unstructured Grids

- Allows for arbitrary cell shapes and sizes
- Can be variable resolution, which enables fine resolutions at specific regions of interest
- Helps create more precise models and simulations, such as Global Storm Resolving Models
- Poses new challenges



Variable resolution, cube sphere grid (CAM-SE)

No standard approach to using unstructured grids

- Few analysis tools exist for working directly with unstructured grids
- Common workflows involve regridding unstructured grids to structured grids

No widely used convention for storage

- Unstructured grid formats include MPAS, SCRIP, Exodus, UGRID, and more
- Each format has different ways of representing unstructured grids
- UGRID conventions have been gaining tracking as a standardized representation of unstructured grids

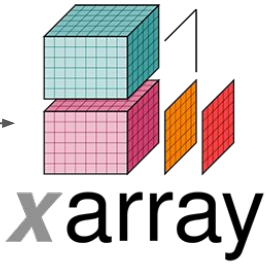
UGRID Conventions

Required topology attributes	**Value**
cf_role	mesh_topology
topology_dimension	2
node_coordinates	
face_node_connectivity	
Optionally required attributes*	
face_dimension	
edge_node_connectivity	
edge_dimension	
Optional attributes	
face_edge_connectivity	
face_face_connectivity	
edge_face_connectivity	
boundary_node_connectivity	
face_coordinates	
edge_coordinates	

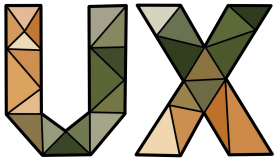
UXarray

- Collaboration between NCAR and DOE
- Supports Xarray-like data analysis techniques directly on unstructured grid data
- Represents unstructured grids in the UGRID conventions to streamline analysis and visualization functionality.

Structured Grids →



Unstructured Grids →



NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

U.S. DEPARTMENT OF ENERGY
Office of Science

PROJECT raijin

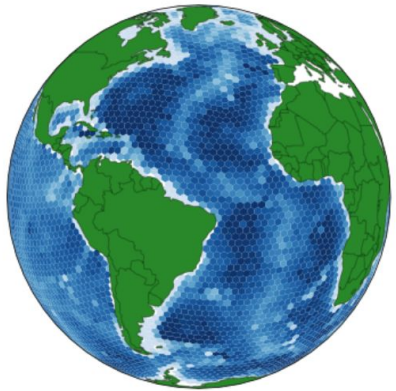
SEATS

SIParCS 2023 Project

- Worked with the GeoCAT team to develop supporting functions and jupyter notebooks for unstructured grid visualization
- Aided in the development of data visualization workflows
- Comparison of various visualization methods using UXarray

Grid Data → Visualization

```
<uxarray.Grid>
Original Grid Type: mpas
Grid Dimensions:
  * nMesh2_node: 15211
  * nMesh2_face: 7153
  * nMaxMesh2_face_nodes: 6
  * nMesh2_edge: 22403
Grid Coordinate Variables:
  * Mesh2_node_x: (15211,)
  * Mesh2_node_y: (15211,)
  * Mesh2_face_x: (7153,)
  * Mesh2_face_y: (7153,)
Grid Connectivity Variables:
  * Mesh2_face_nodes: (7153, 6)
  * Mesh2_edge_nodes: (22403, 2)
  * nNodes_per_face: (7153,)
```



Data Processing Workflow for Visualization

1. Derive Geometries from Coordinates and Connectivity Variables
2. Represent the Geometry in an appropriate format for each data structure
3. Correct Geometries that touch or cross the Antimeridian
4. Construct Data Structure

Coordinates and Connectivity Information

Longitude and Latitude of our Nodes

Lons	Lats
x_1	y_1
x_2	y_2
…	…
x_n	y_n

Connectivity array describing which nodes compose each polygon

Connectivity Information						
Polygon 1	1	9	6	4	7	8
…	…	…	…	…	…	…
Polygon n	…	…	…	…	…	…

Coordinates and Connectivity Information

Longitude and Latitude of our Nodes

Lons	Lats
x_0	y_0
x_1	y_1
…	…
x_n	y_n

Connectivity array describing which nodes compose each polygon

Connectivity Information						
Polygon 1	1	9	6	4	7	8
…	…	…	…	…	…	…
Polygon n	…	…	…	…	…	…

Each number represents the index of a node in our coordinate arrays

Coordinates and Connectivity Information

Longitude and Latitude of our Nodes

Lons	Lats
x_0	y_0
x_1	y_1
…	…
x_n	y_n

Connectivity array describing which nodes compose each polygon

Connectivity Information						
Polygon 1	1	9	6	4	7	8
…	…	…	…	…	…	…
Polygon n	…	…	…	…	…	…

Number of columns represents the maximum number of nodes a polygon can have

Coordinates and Connectivity Information

Longitude and Latitude of our Nodes

Lons	Lats
x_0	y_0
x_1	y_1
…	…
x_n	y_n

Connectivity array describing which nodes compose each polygon

Connectivity Information						
Polygon 1	1	9	6	4	X	X
…	…	…	…	…	…	…
Polygon n	…	…	…	…	…	…

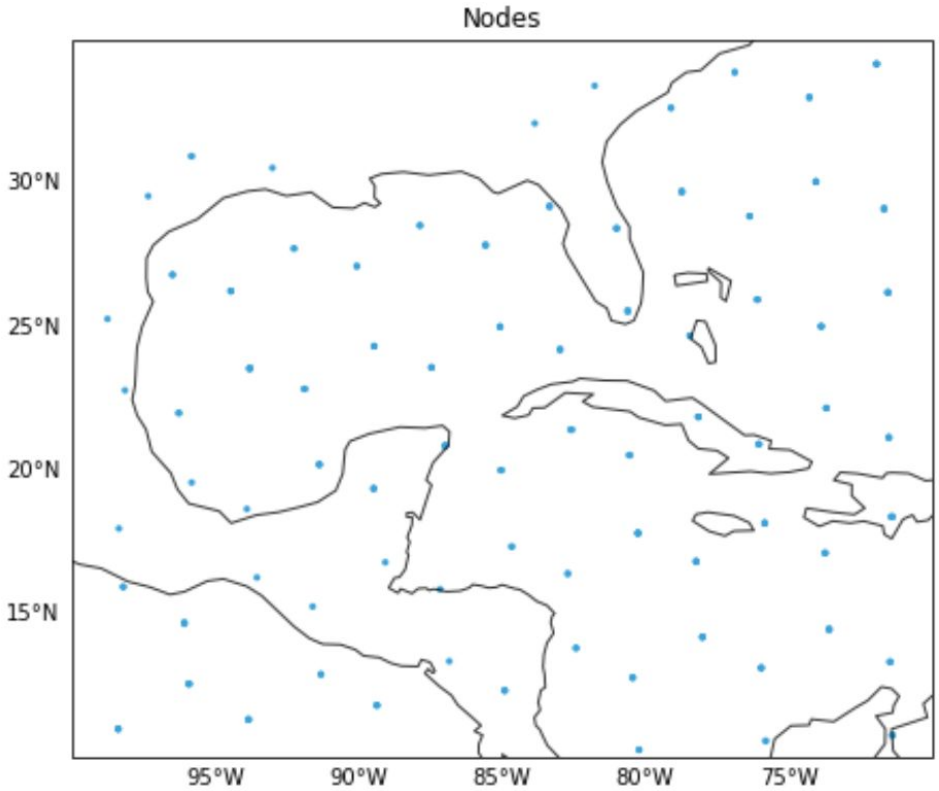
For polygons with less nodes we can pad with fill values

1) Coordinates and Connectivity to Geometry

Lons
x_0
x_1
…
x_n

Lats
y_0
y_1
…
y_n

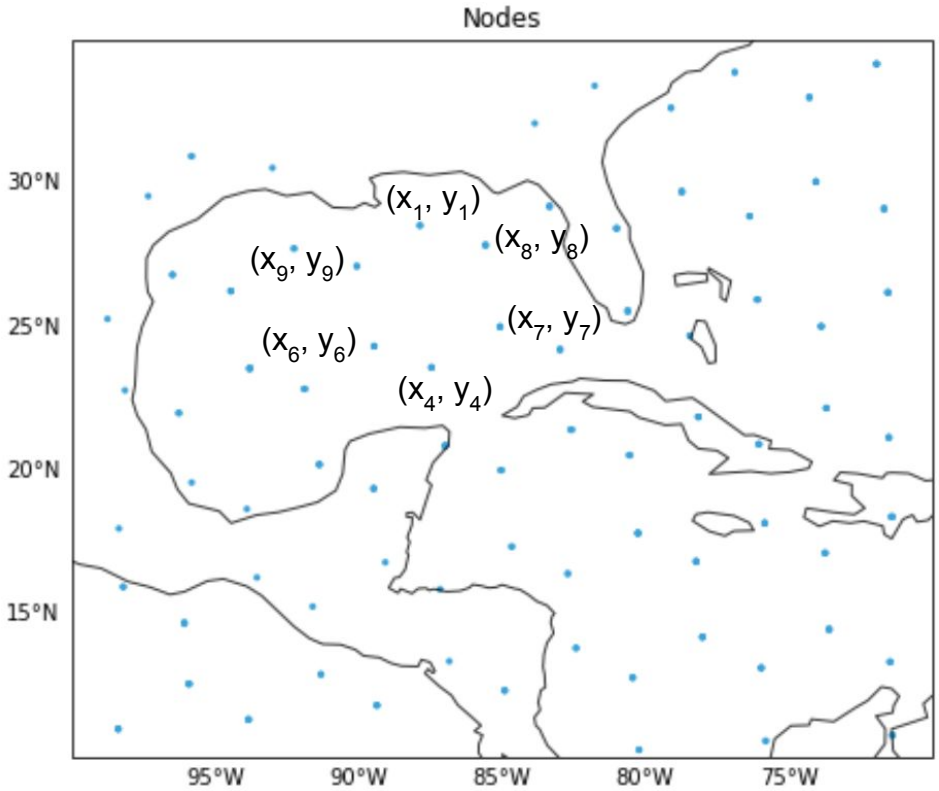
Connectivity Information						
Polygon 1	1	9	6	4	7	8
…	…	…	…	…	…	…
Polygon n	…	…	…	…	…	…



1) Coordinates and Connectivity to Geometry

Lons	Lats
x_0	y_0
x_1	y_1
…	…
x_n	y_n

Connectivity Information						
Polygon 1	1	9	6	4	7	8
…	…	…	…	…	…	…
Polygon n	…	…	…	…	…	…

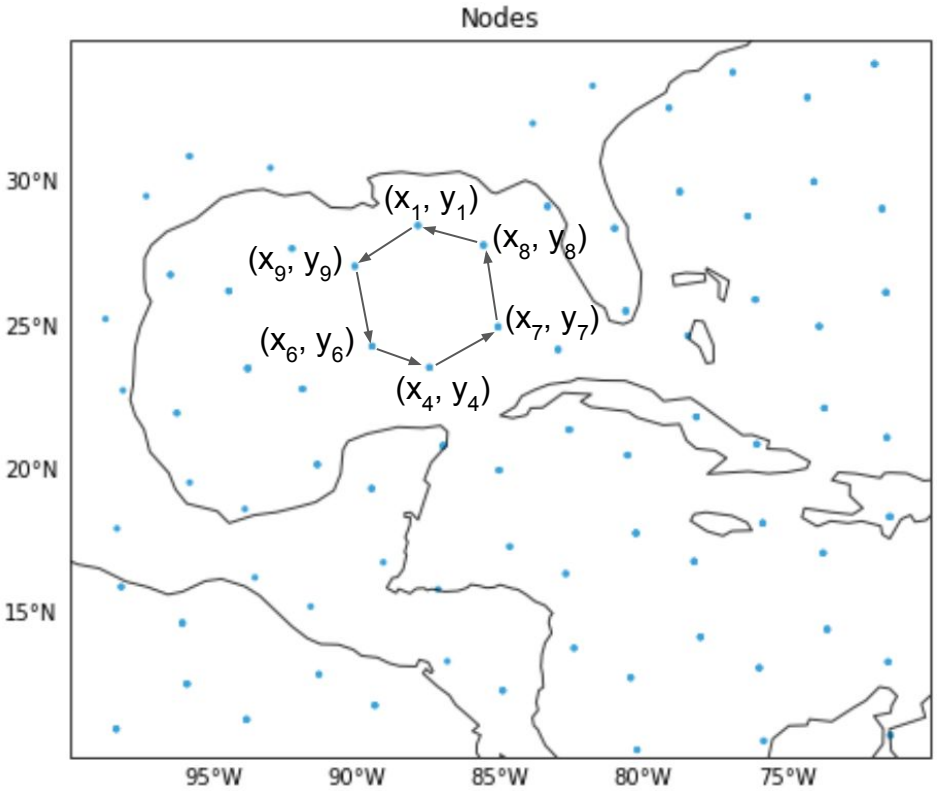


1) Coordinates and Connectivity to Geometry

Lons
x_0
x_1
…
x_n

Lats
y_0
y_1
…
y_n

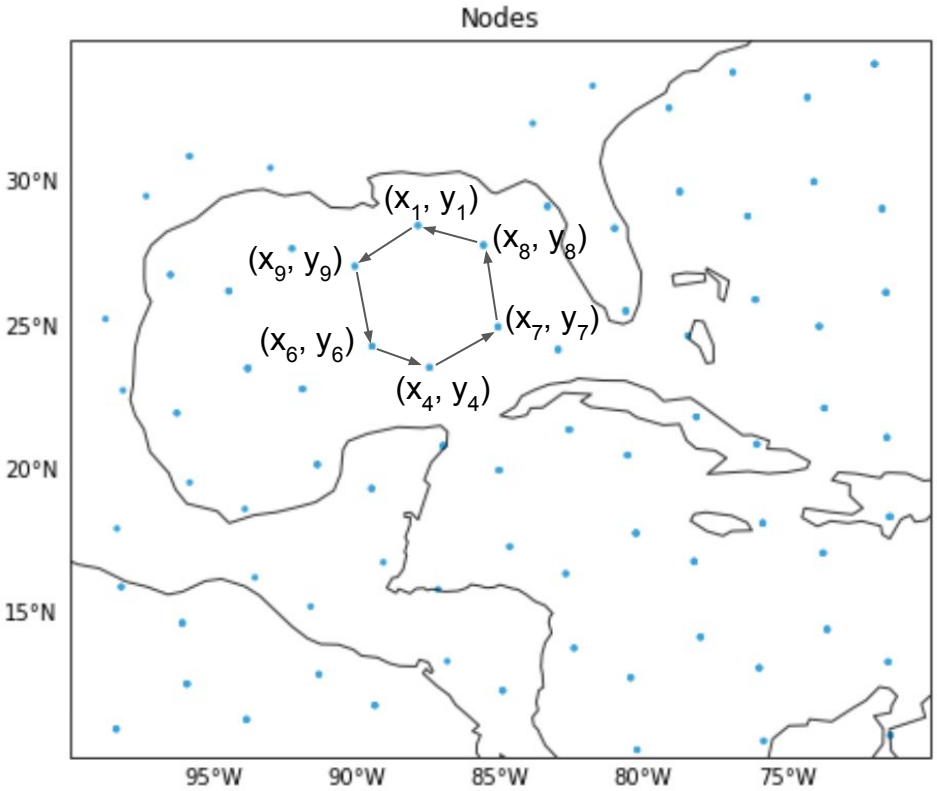
Connectivity Information						
Polygon 1	1	9	6	4	7	8
…	…	…	…	…	…	…
Polygon n	…	…	…	…	…	…



1) Coordinates and Connectivity to Geometry

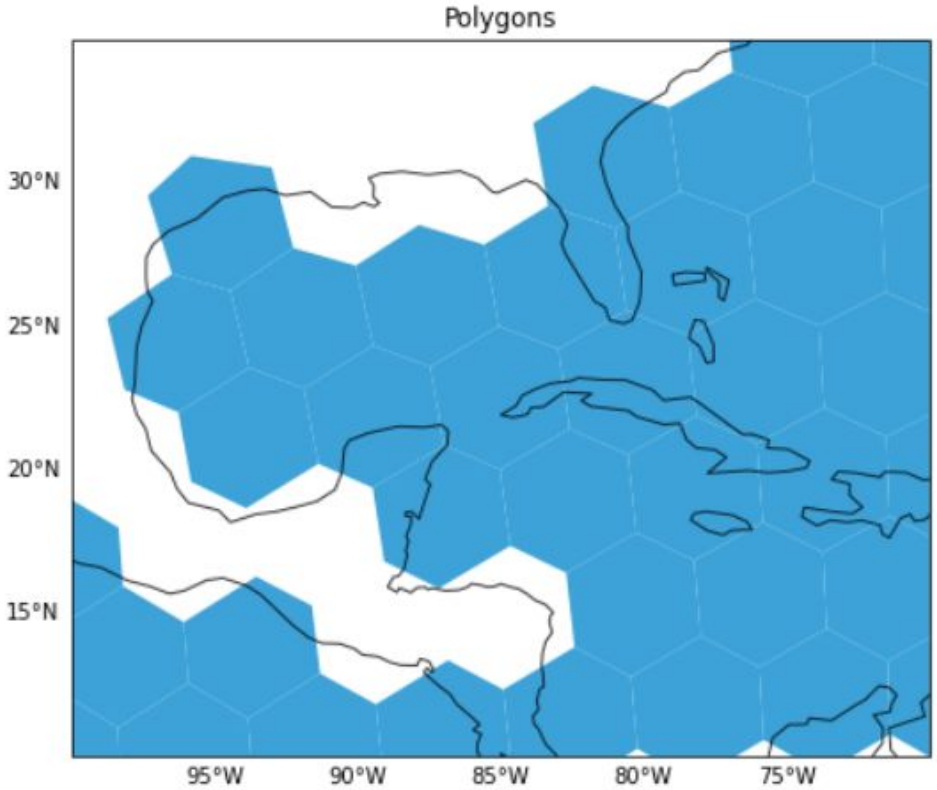
Connectivity Information						
Polygon 1	1	9	6	4	7	8
…	…	…	…	…	…	…
Polygon n	…	…	…	…	…	…

Polygon Shells				
Polygon 1	$(x_1\ y_1)$	(x_9, y_9)	…	…
…	…	…	…	…
Polygon n	…	…	…	…



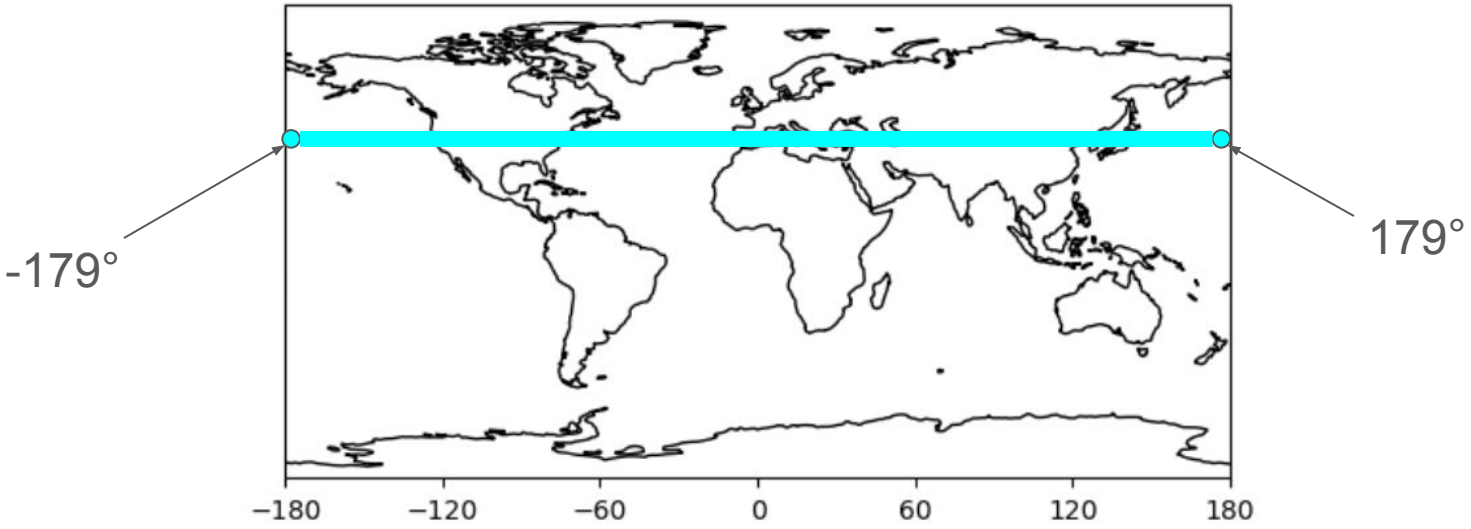
2) Convert to Shapely Polygons

- Shapely is a package that provides representations of geometric objects, such as lines and polygons
- For each face, we construct a Shapely Polygon using the shells derived in the previous step



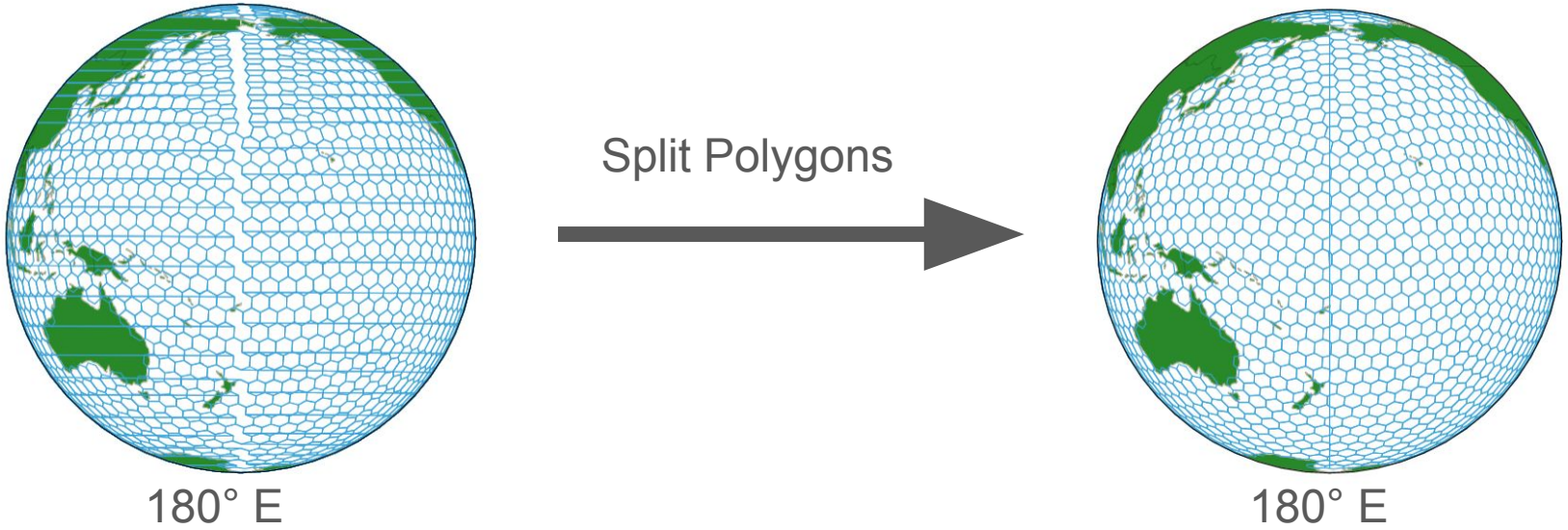
3) Correct Antimeridian Polygons

- These grids reside on a sphere: Points that seem distant based on their coordinates are actually close
- -180° E and 180° E is referred to as the Antimeridian. Polygons that cross this will connect the wrong way, and UXarray can detect these polygons.

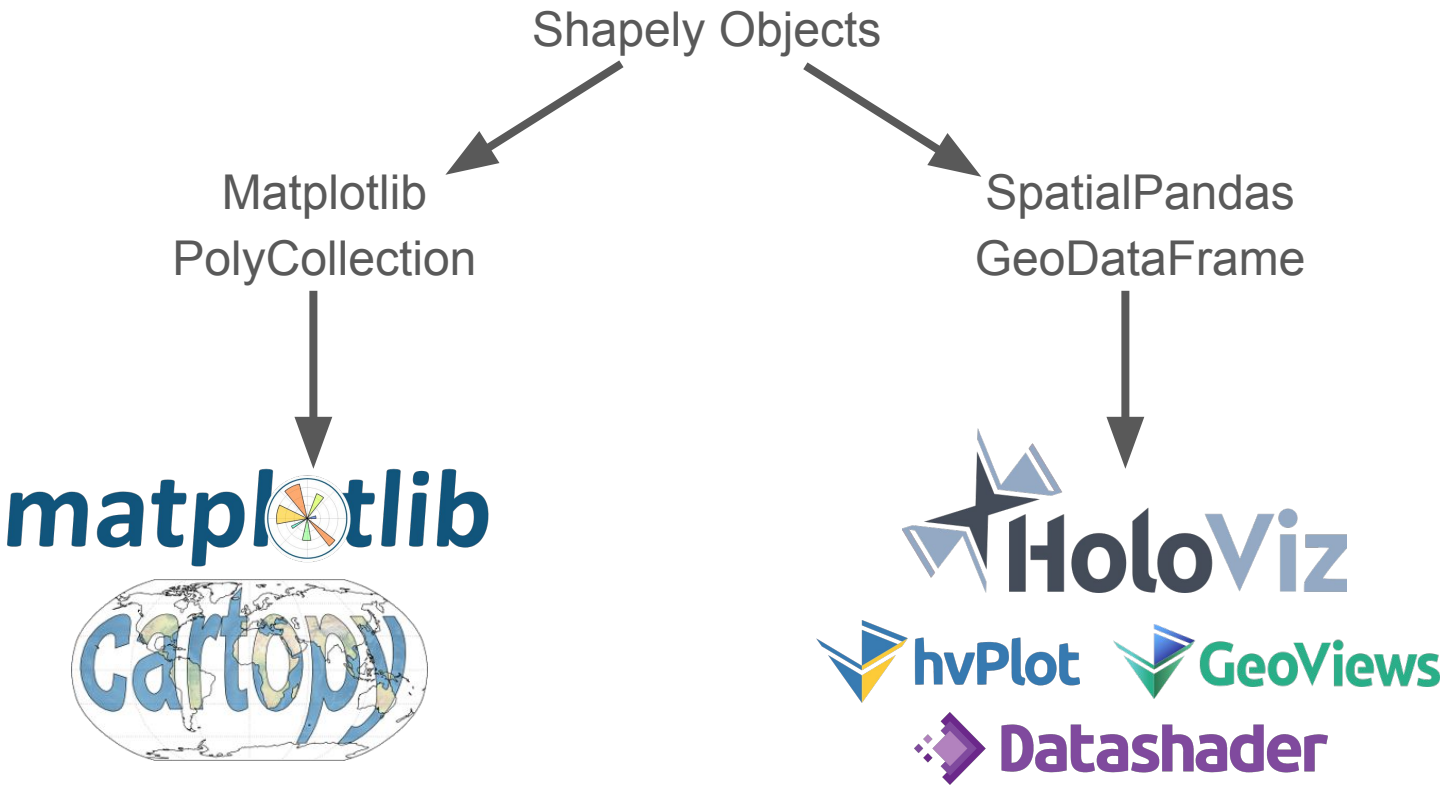


3) Correct Antimeridian Polygons

- Polygons that cross the antimeridian wrap around Earth
- **antimeridian** is a community package that splits polygons according to the GeoJSON standards



4) Create Data Structures



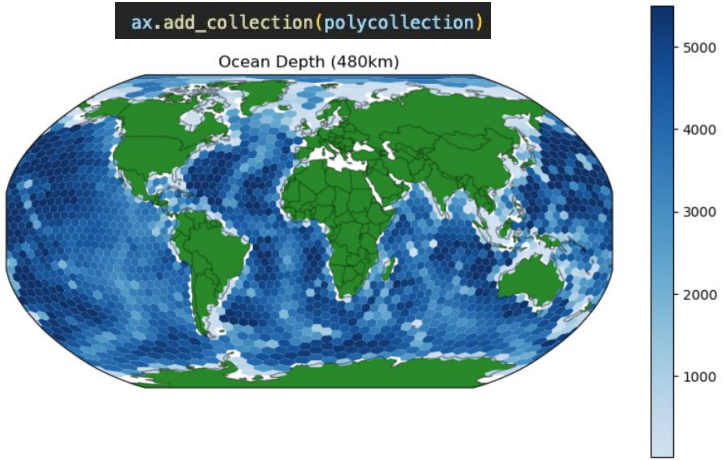
Single Line Conversion Methods

```
ds = ux.open_dataset(grid_path, data_path)
```

matplotlib

```
polycollection = ds['bottomDepth'].to_polycollection()
```

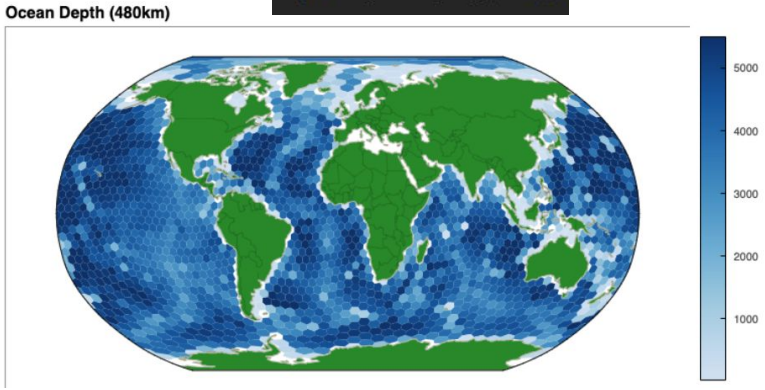
```
ax.add_collection(polycollection)
```



HoloViz

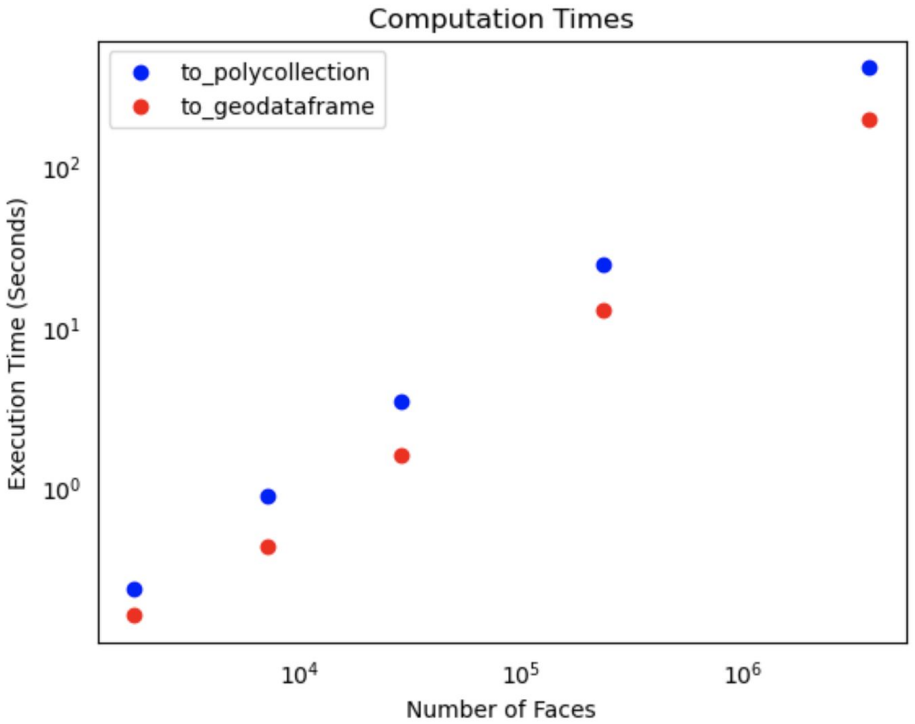
```
gdf = ds['bottomDepth'].to_geodataframe()
```

```
gdf.hvplot.polygons()
```



Performance of Functions vs Number of Faces

- These functions scale linearly with number of faces
- If we double to number of faces we would expect the execution time to double



Pythia Cookbook

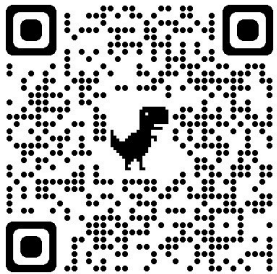
- Project Pythia is an educational resource for the Geoscience community supported by Pangeo
- Pythia Cookbooks provide example workflows for domain specific topics
- Created from Jupyter Notebooks

QR Code to Cookbook →

Cookbook Overview & Purpose

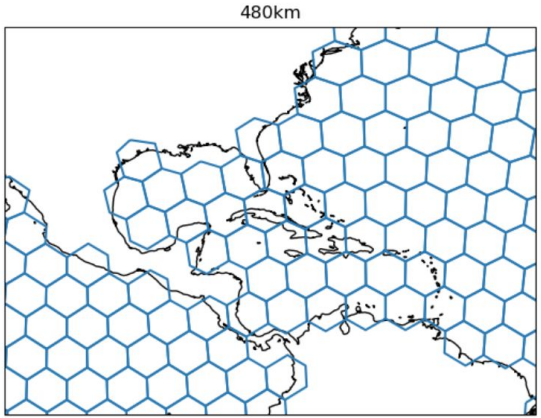
- Showcase various methods to visualize unstructured grids with UXarray
- Compare the performance of visualization methods across different grid sizes
- Matplotlib vs HoloViz
- Node, Edge, and Polygon Plots

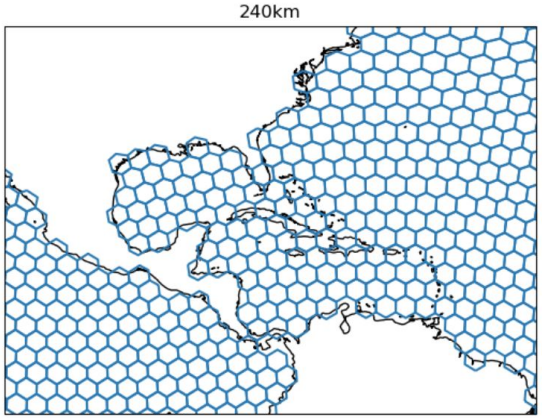
QR Code to Cookbook →

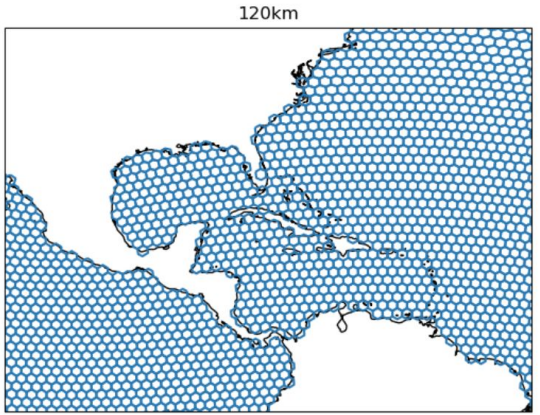


Datasets

- MPAS ocean datasets from Energy Exascale Earth System Model (E3SM)
- 3 different resolutions - 480km, 240km, 120km
- 2x the resolution means 4x the number of faces

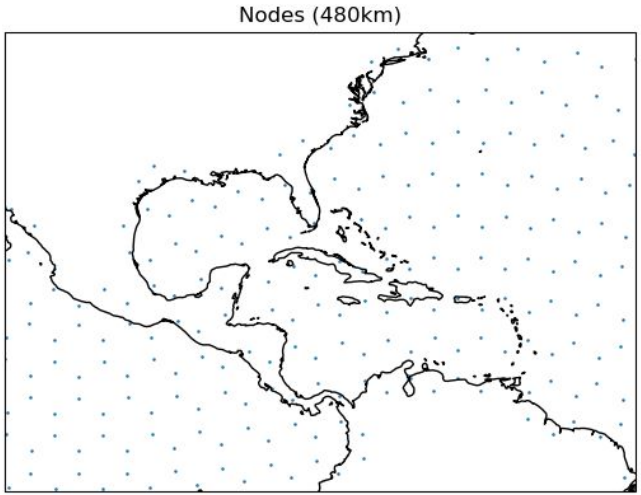


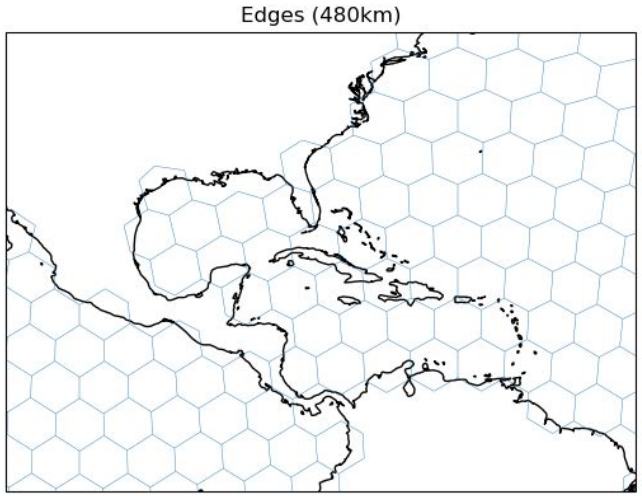




Node and Edge Plots

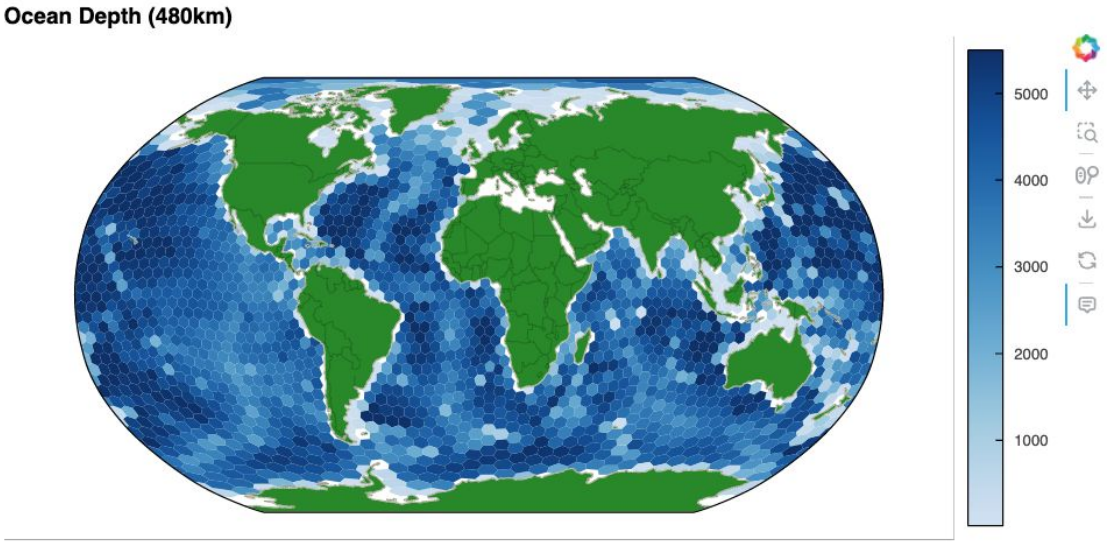
- If our goal is to view the geometries of the grid, we can create node and edge plots
- Simple and quick to compute





Polygon Plots

- To visualize data we can create shaded polygon plots
- Can make vector images or raster images

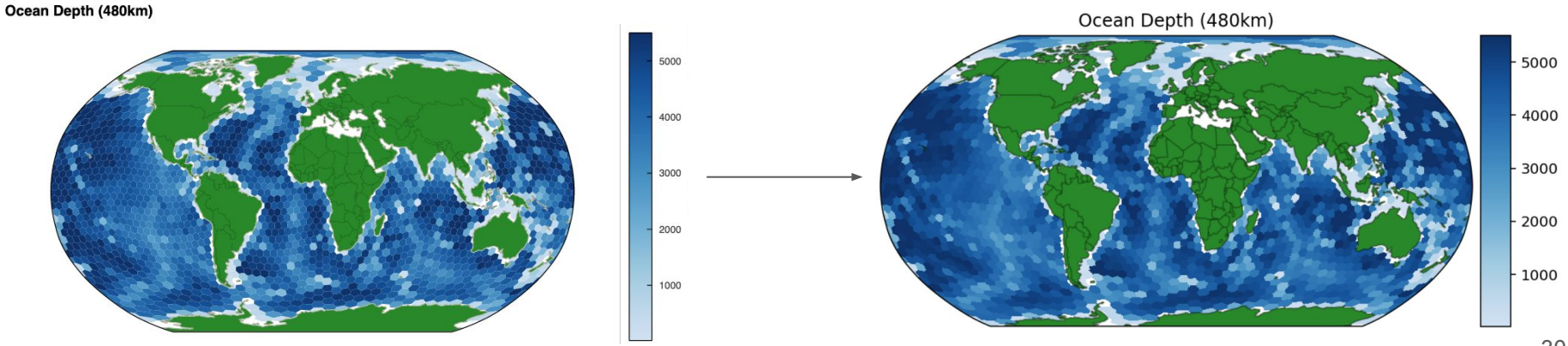


Vector

- Render each geometry individually
- Can be computationally expensive with large datasets

Raster

- Approximate geometries with a grid of pixels
- Can provide faster rendering



Vector

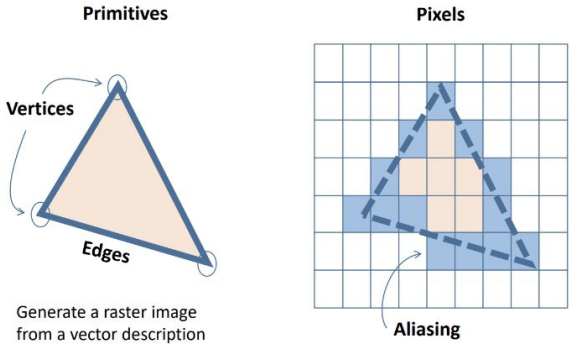
- Render each geometry individually
- Can be computationally expensive with large datasets
- Straight lines connecting nodes

Raster

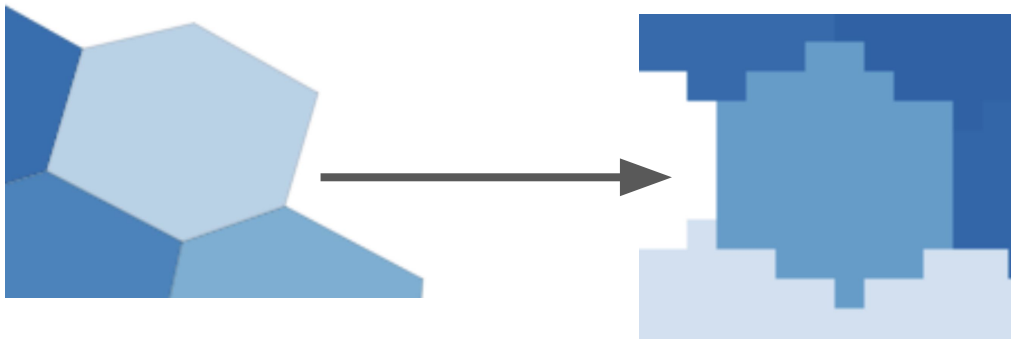
- Approximate geometries with a grid of pixels
- Can provide faster rendering
- Edges have a staircase like shape

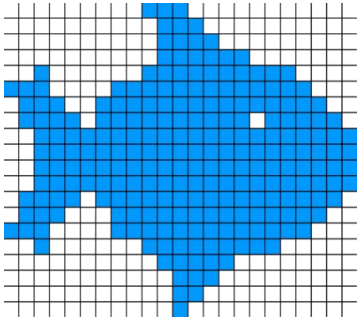
Rasterization

- Estimates shapes by shading a grid of boxes
- Raster images have lower quality, but can still provide useful information
- We are converting to a structured grid at the visualization step, without modifying the actual data



University of Illinois Urbana-Champaign



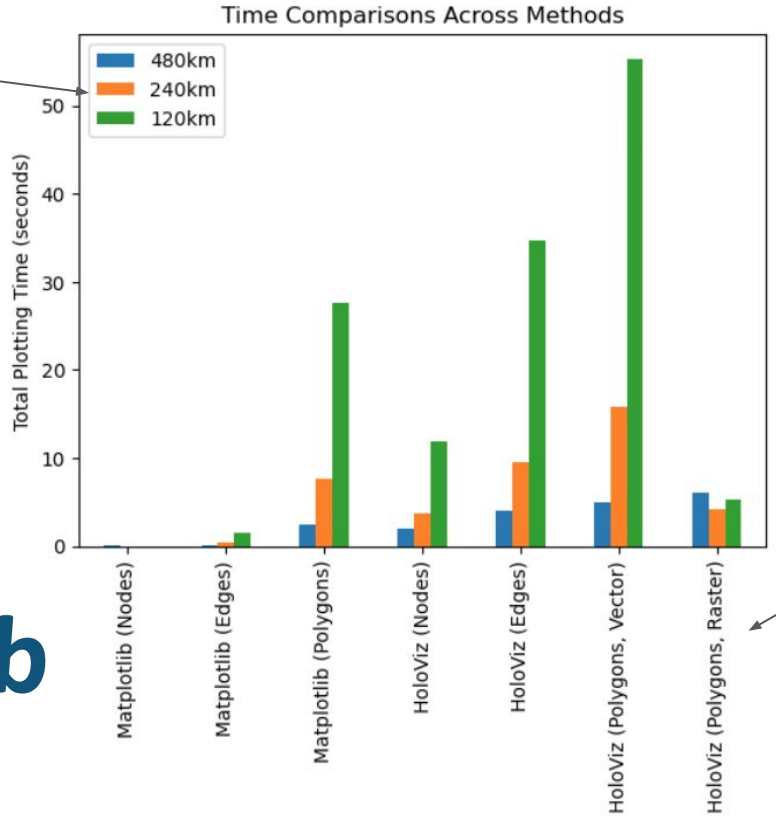


https://en.wikipedia.org/wiki/Rasterisation

Comparisons of Plotting Methods

3 resolutions, with 120km having the most cells

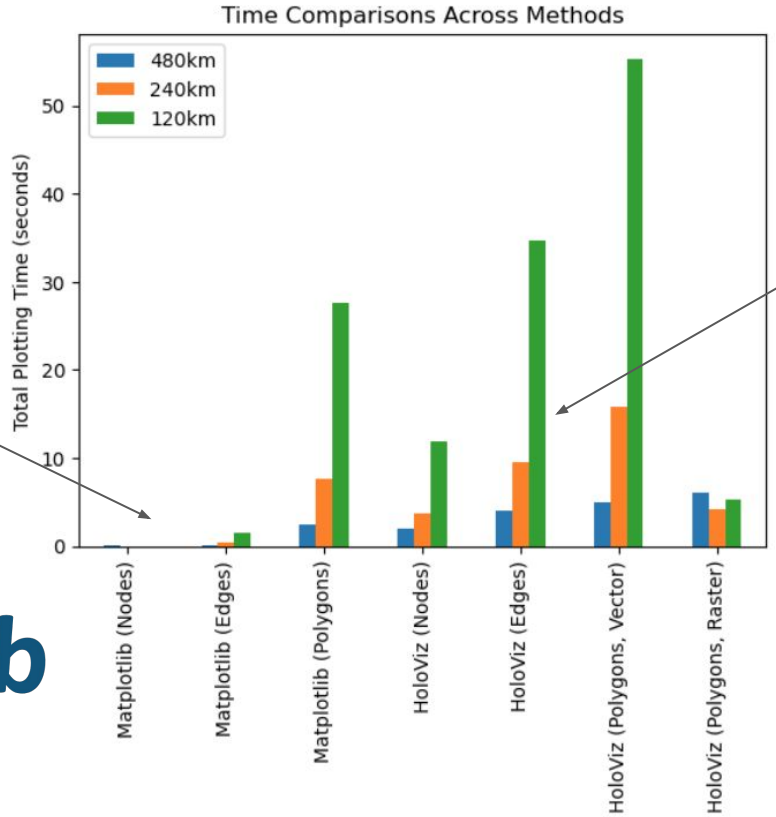
Plotting time in seconds



Plotting package and plot type

Comparisons of Plotting Methods

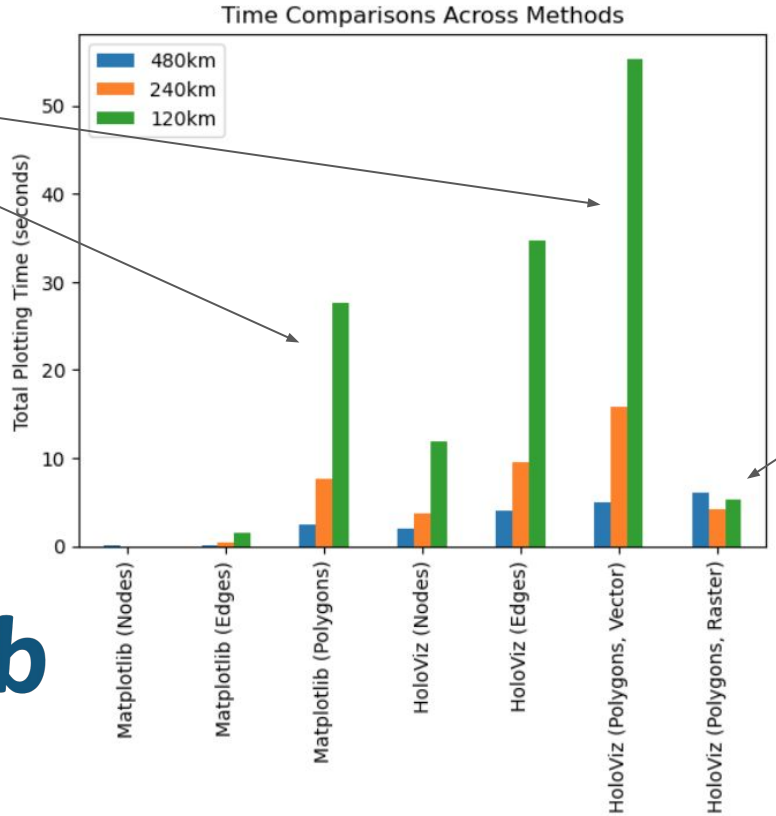
Matplotlib Nodes and Edges plot very quickly



HoloViz nodes and edges involve extracting information from polygons, so not much time is saved

Comparisons of Plotting Methods

Vector images take quite some time to display



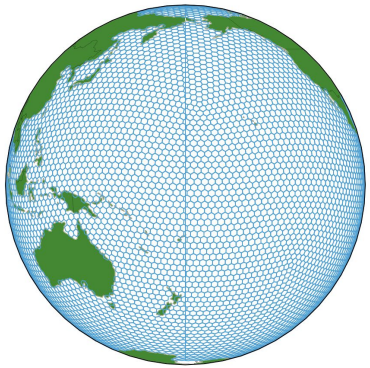
Raster images are much faster and do not vary with grid size

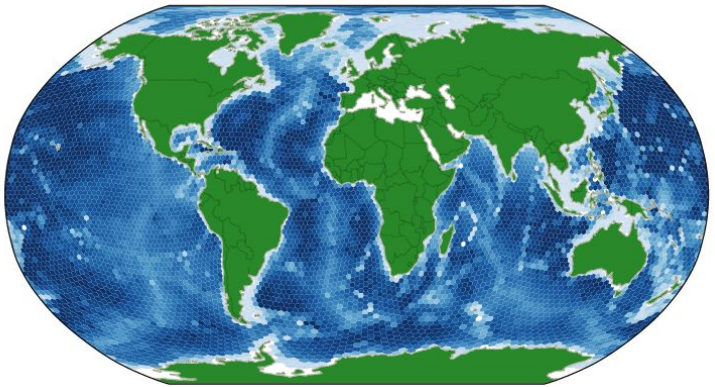
matplotlib

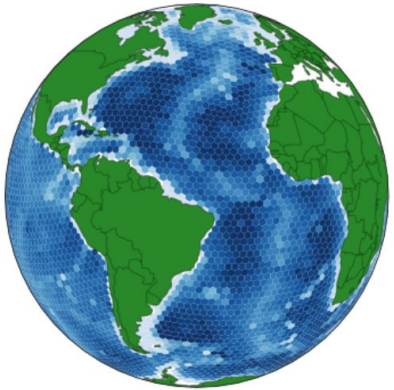
HoloViz

Future Work

- This summer's project focussed on implementing conversion methods to support visualization
- Next phase will be to implement visualization methods that internally run these conversions to allow for seamless visualization







Thank You

Visualization Cookbook

UXarray Documentation

Special thanks to the GeoCAT team: Anissa Zacharias, Julia Kent, Katelyn FitzGerald, Orhan Eroglu, Philip Chmielowiec