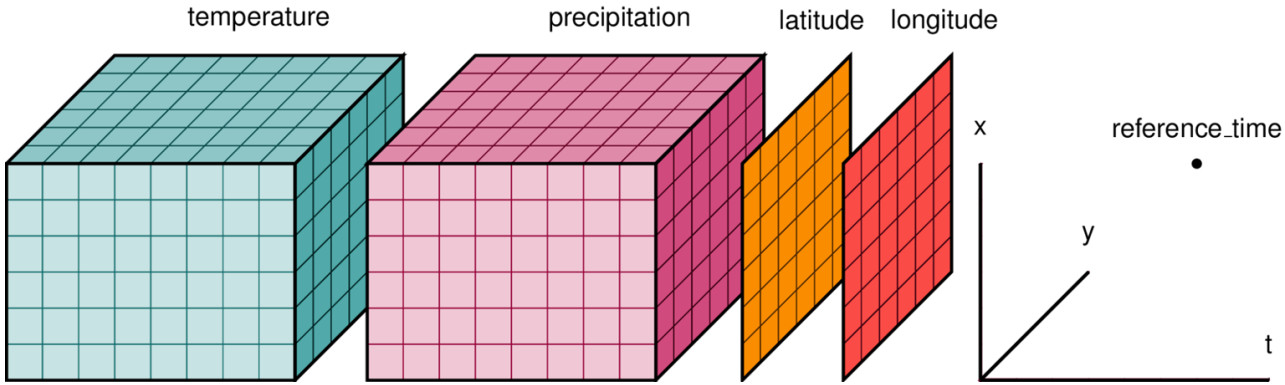
UXarray: Python package for the analysis and visualization of model output on unstructured climate grids

Rajeev Jain (Presenter), Phillip Chmielowiec, Aaron Zedwick, Hongyu Chen, Paul Ulrich, Rob Jacob, Orhan Eroglu, Rachel Tam

Development of UXarray is supported by: the Regional and Global Modeling and Analysis (RGMA) program area in the Earth Environmental System Modeling Program of the U.S. Department of Energy's (DOE) Office of Science AND the Earthcube program of the U.S. National Science Foundation (NSF).

Xarray: The Structured Scientific Data Model
What makes it so popular with geoscientists?



- **Convenience**: Select values by label, not integer location
- **Popular geoscience I/O backends**: NetCDF, GRIB, OpenDap, HDF, Zarr
- **Interoperability**: works with the scientific Python ecosystem including NumPy, Dask, Pandas, and Matplotlib
- **Performant**: Operators are vectorized, implemented in compiled code, and easily parallelized (usually) with Dask
- **Ease of use**: E.g. overloaded operators, such as `plot()`, that just “do the right thing”

image source: https://doi.org/10.5334/jors.148

UGRID Conventions

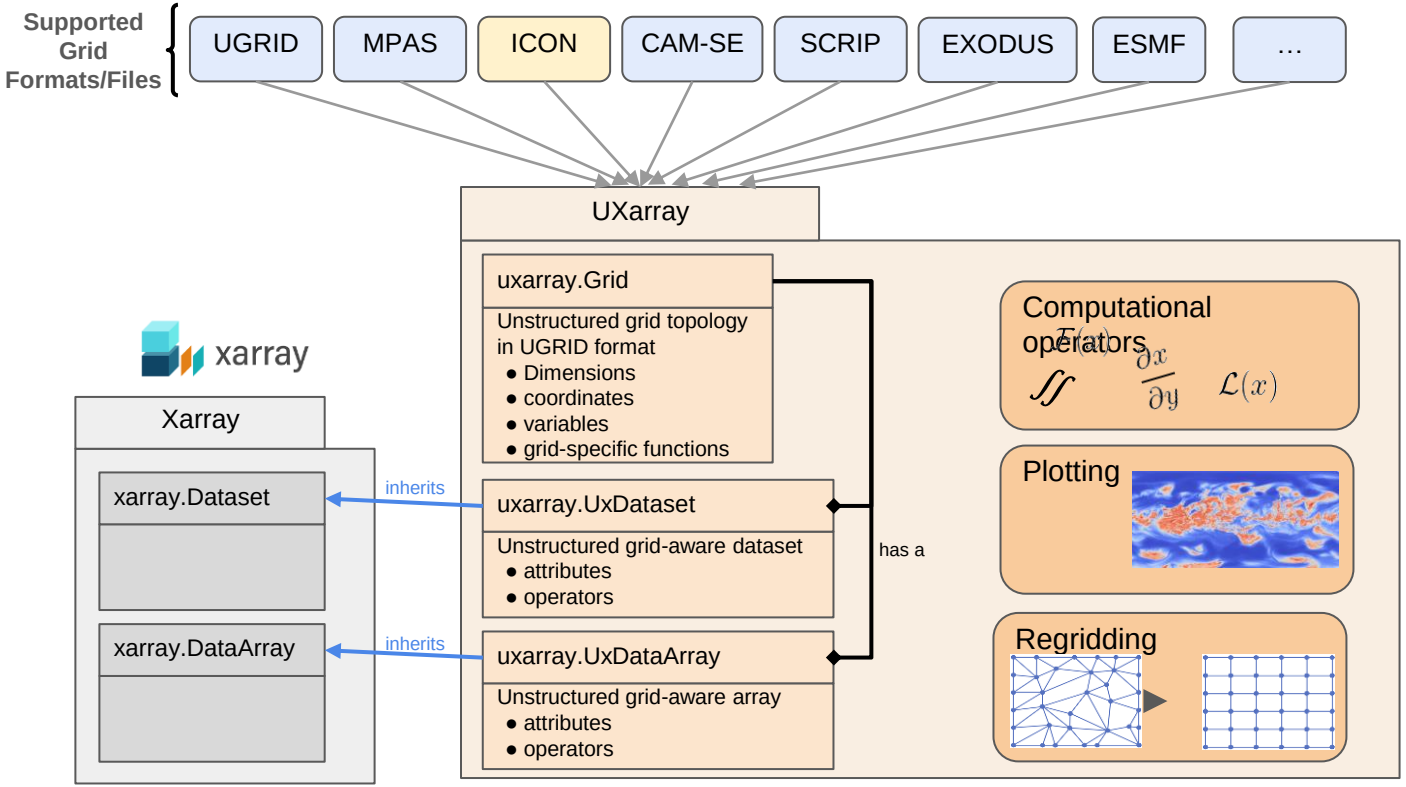
- UGRID is a convention for describing unstructured meshes in the Unidata Network Common Data Form (NetCDF) file format
- UGRID conventions are consistent, portable and extensible [1]
- Key features in the UGRID conventions include: Topology, Node attributes, Edge attributes, Face attributes, Data variables, Metadata

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	

[1]http://ugrid-conventions.github.io/ugrid-conventions/#2d-flexible-mesh-mixed-triangles-quadrilaterals-etc-topology

UXarray: Xarray + unstructured grid support

UXarray operates directly on Unstructured Grids (without regridding needed)



Key Features

- Support for reading a variety of grid formats
- Integration and other Derivative Support
- Regridding/Remapping and Subsetting
- Tree Representation and Dual Mesh Construction
- Topological Aggregation
- Lat/lon Bounding Box Calculations
- Visualization and rendering capabilities

A sampling of UXarray public class methods

```
# Open a dataset and analyze sea surface temperature
import uxarray as ux
uxds = ux.open_dataset(grid_file, data_file…)
grid = ux.open_grid(grid_file)
sst = uxds[`sst`]
```

Operators inherited as is from Xarray (grid info not needed!)

```
sst.argmin()

sst.mean()

sst.where()
```

Operators overridden from Xarray

```
sst.integrate()

sst.to_netcdf()

sst.plot()
```

New operators added by UXarray

```
grid.plot()

sst.plot.points()

grid.compute_face_areas()
```

Already using Xarray? UXarray should look pretty familiar!

Future work

Work in progress

- Derivatives (advanced capabilities)
- Zonal means (conservative and non-conservative)
- Topological aggregations
- Rendering performance optimizations

Longer term

- New computational operators
- Support for more unstructured grid formats
- Address performance and scalability issues
- More: tutorials and demonstrations

Documentation and Jupyter Notebooks