Bilinear Interpolation

Python Reimplementation of Fortran Subroutine

ALINA GUHA ,
Grinnell College, SIParCS/GeoCAT Intern

JULY 26, 2022

GeoCAT (Geoscience Community Analysis Toolkit)

"Pivot to Python"

- Replicate NCL functionality
- Open development model

GeoCAT (Geoscience Community Analysis Toolkit)

“Pivot to Python”

- Replicate NCL functionality
- Open development model

GeoCAT-comp (and GeoCAT-f2py)

- Geosciences computational functions in Scientific Python Ecosystem

GeoCAT-examples (and GeoCAT-viz)

- Geosciences data plotting gallery in Scientific Python Ecosystem

WRF-Python

- Diagnostic and interpolation routines of WRF -ARW model outputs

Project Raijin

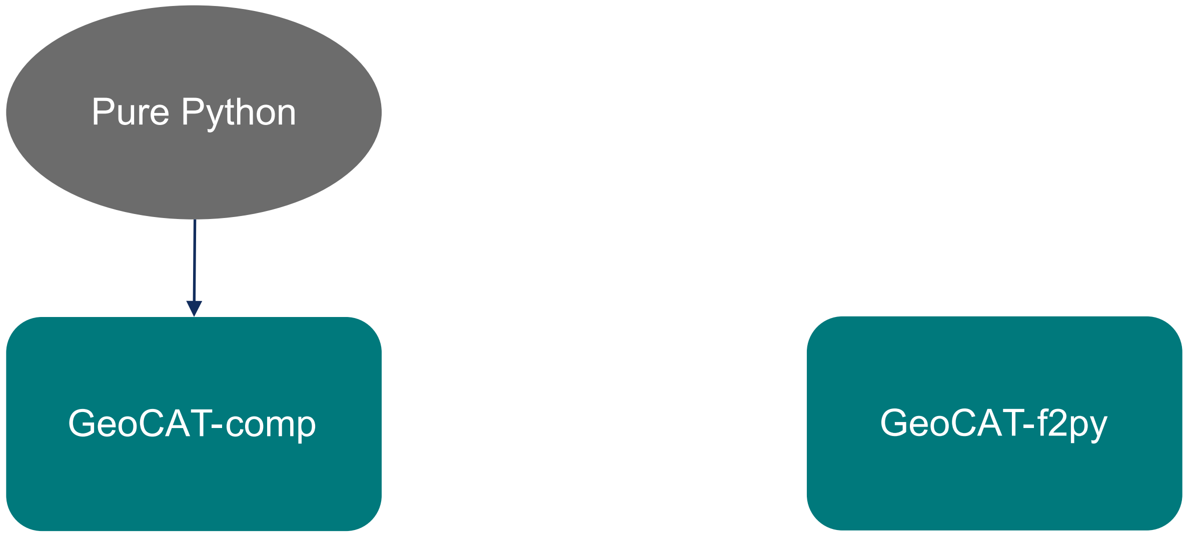
- NSF EarthCube-funded award
- Support analysis and visualization on unstructured data

Fortran to Python

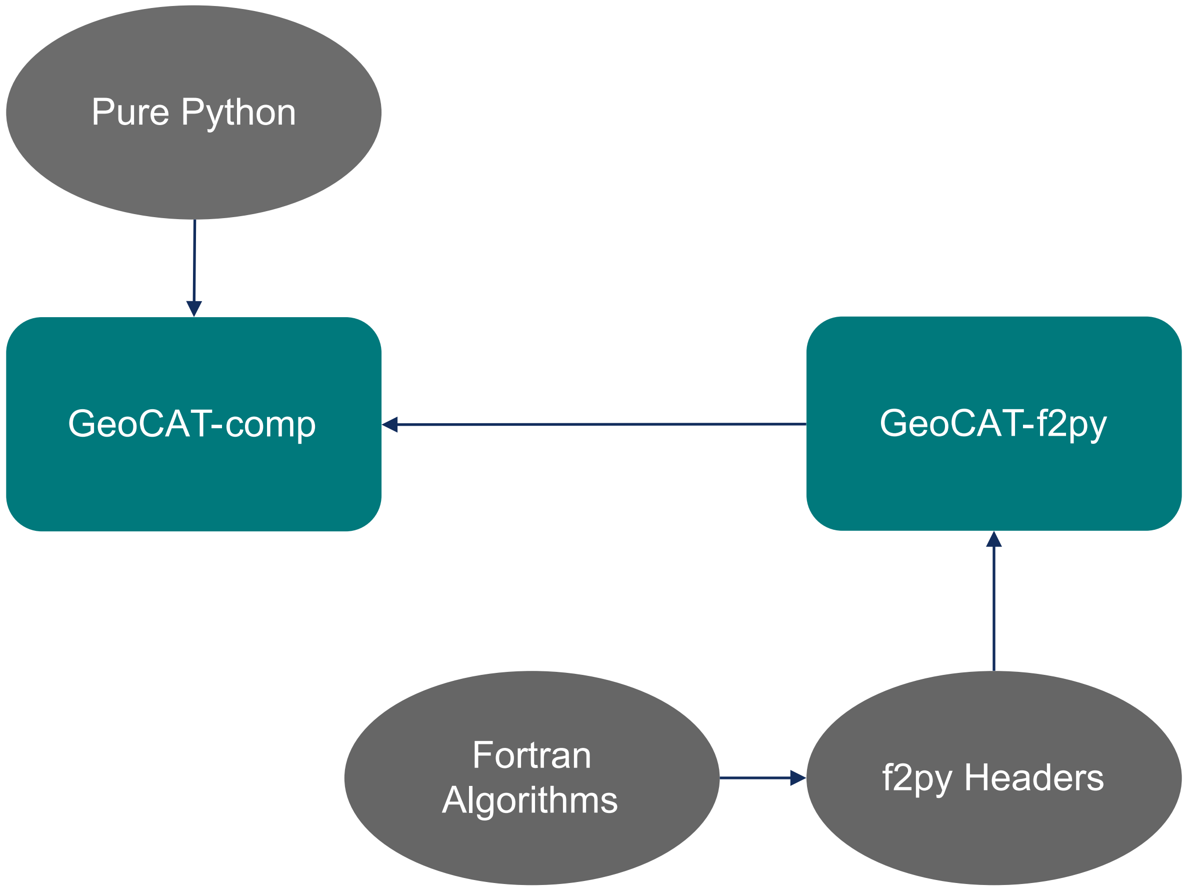
GeoCAT-comp

GeoCAT-f2py

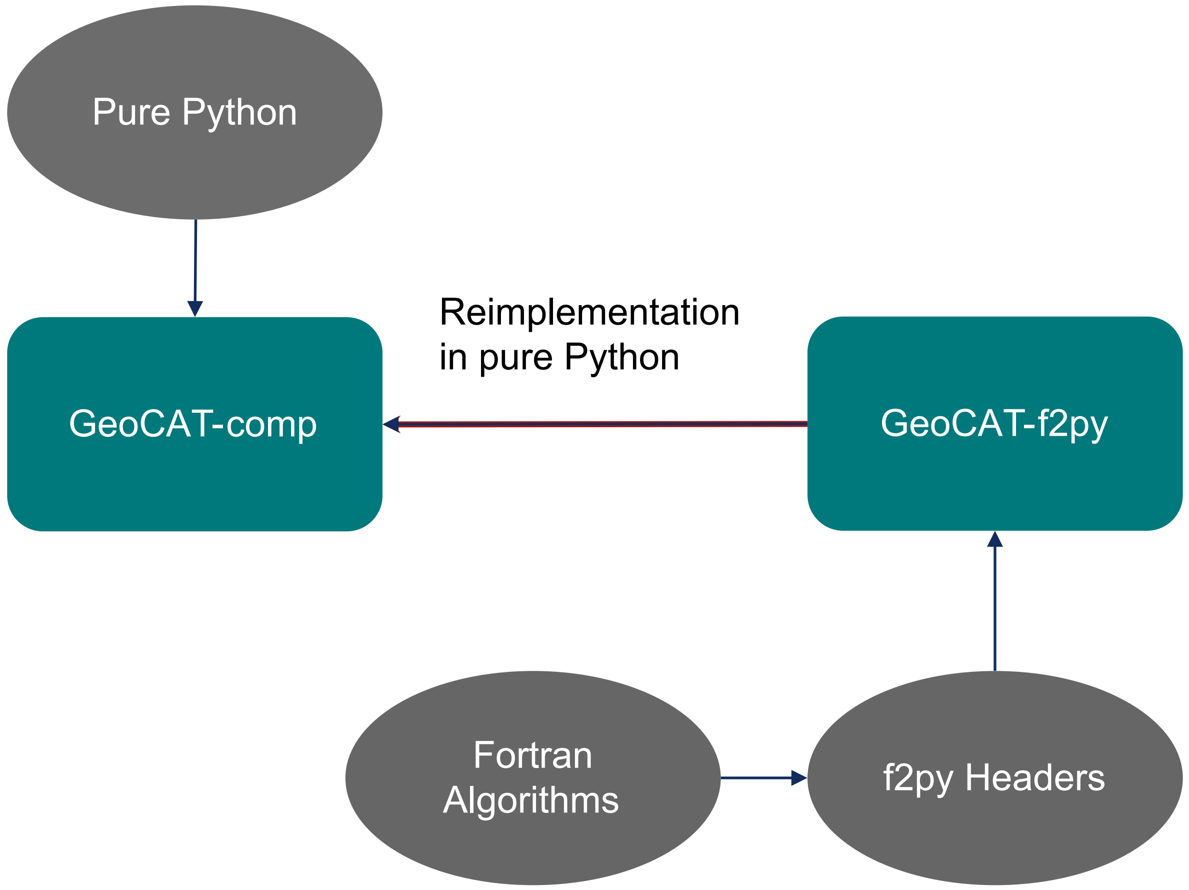
Fortran to Python



Fortran to Python



Fortran to Python



dlinint2 -> interp_multidim

```fortran
SUBROUTINE DLININT2(NXI,XI,NYI,YI,FI,ICYCX,NXO,XO,NYO,YO,FO,XIW,
    +                    FXIW,NXI2,XMSG,IOPT,IER)
```

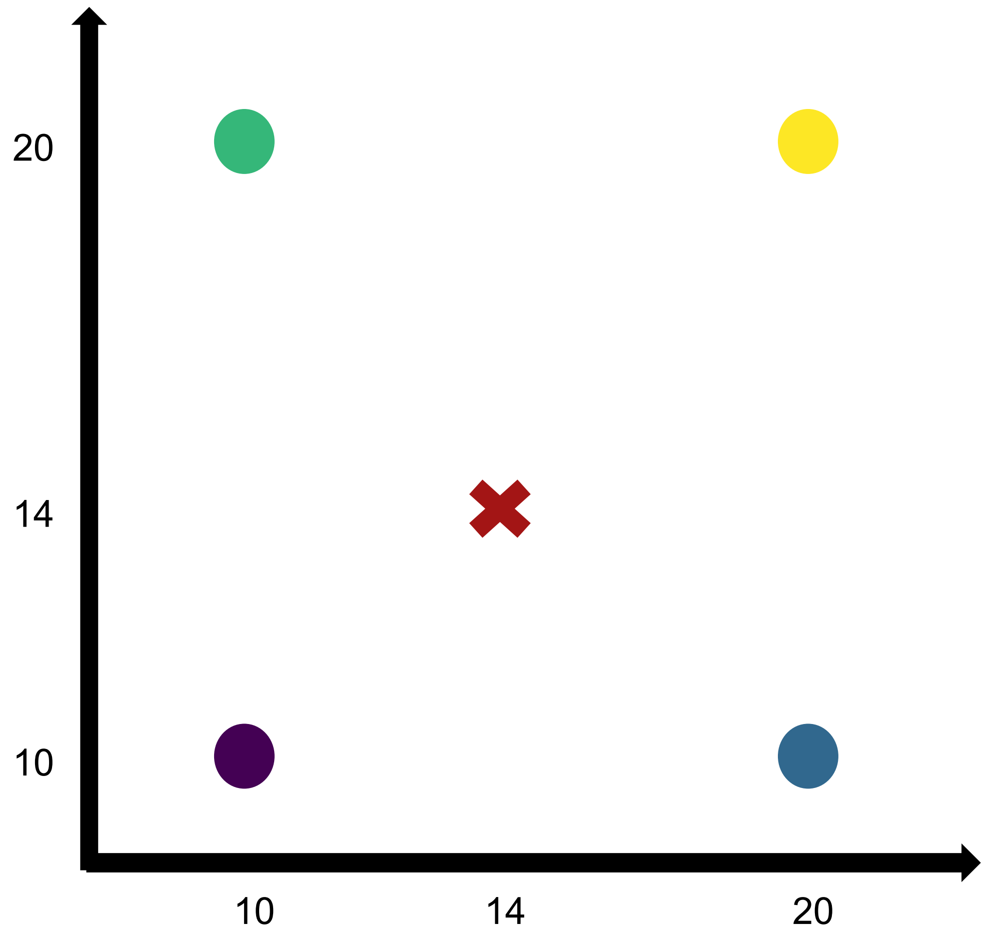
NCAR | UCAR

dlinint2 -> interp_multidim

```
SUBROUTINE DLININT2(NXI,XI,NYI,YI,FI,ICYCX,NXO,XO,NYO,YO,FO,XIW,
    +                    FXIW,NXI2,XMSG,IOPT,IER)
```

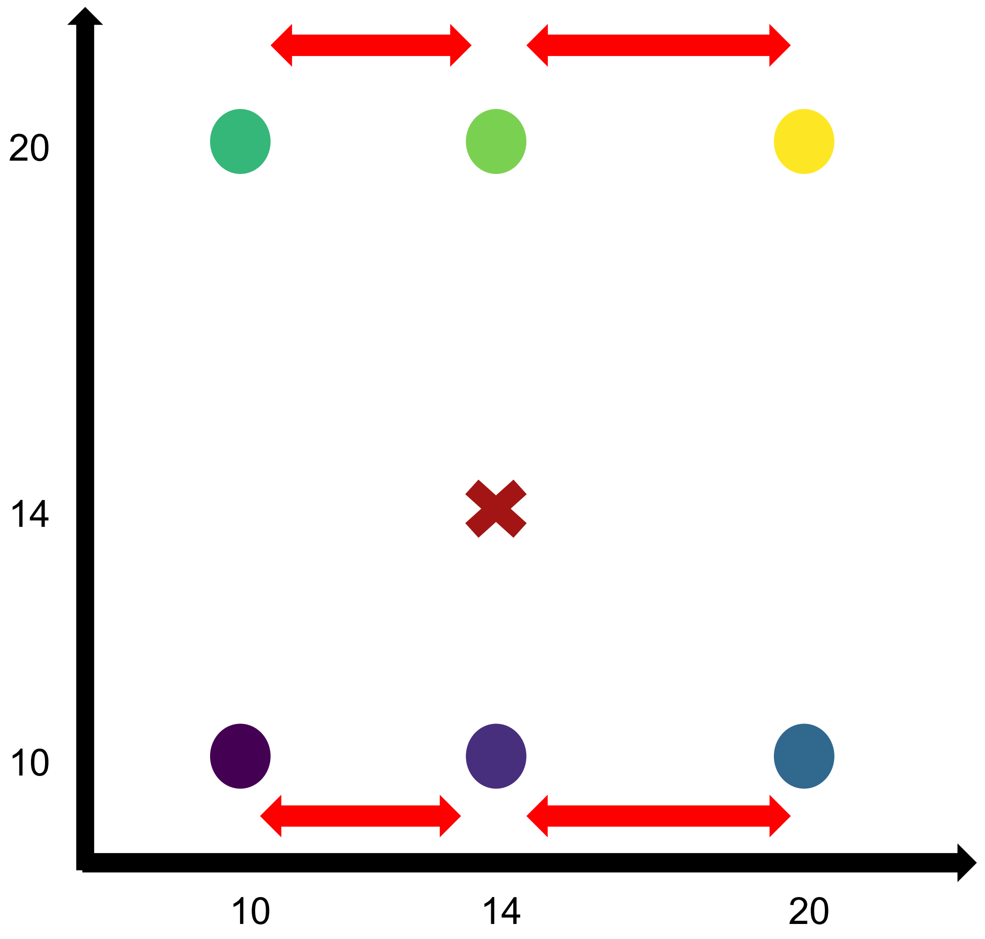
```
def interp_multidim(data_in:
        typing.Union[xr.DataArray, np.ndarray],
              lat_out: np.ndarray,
              lon_out: np.ndarray,
              lat_in: np.ndarray = None,
              lon_in: np.ndarray = None,
              cyclic: bool = False,
              missing_val: np.number = None,
              method: str = "linear",
              fill_value:
        typing.Union[str, np.number] = np.nan)
```

Bilinear Interpolation



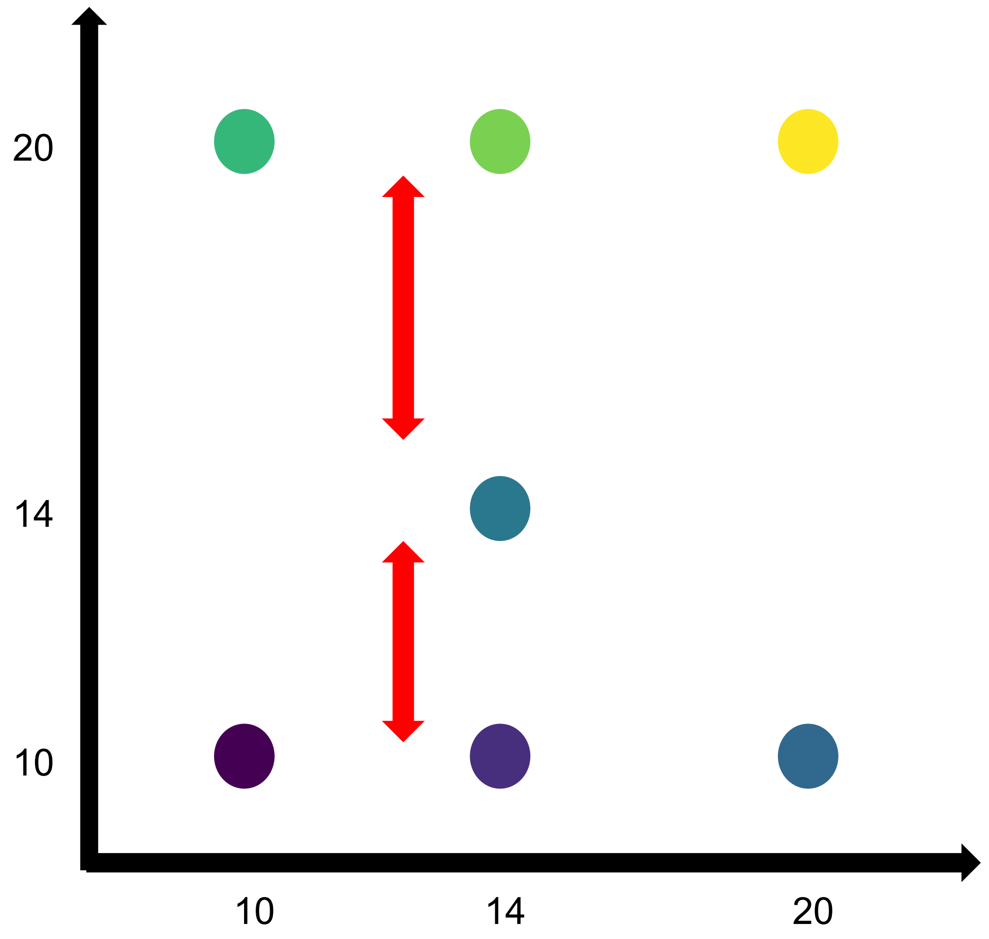
Approximate values at unknown data points using input data

Bilinear Interpolation



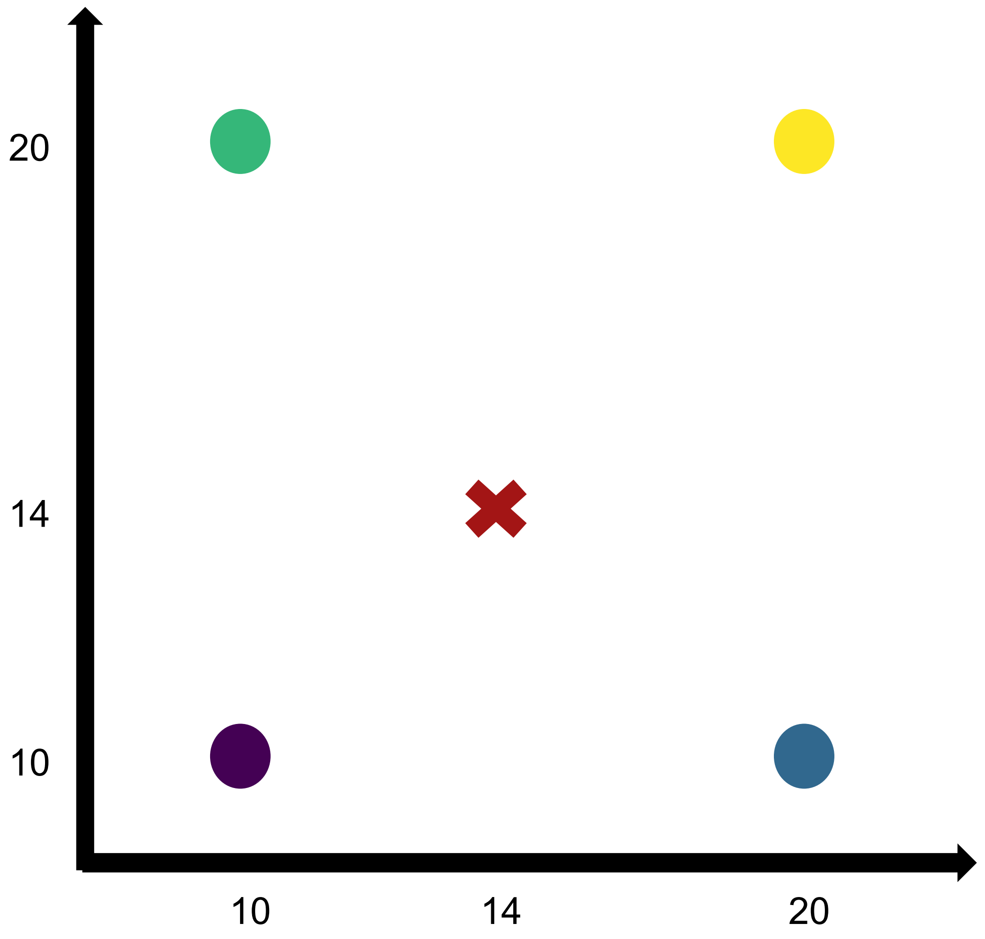
2 linear interpolations along the latitudes

Bilinear Interpolation

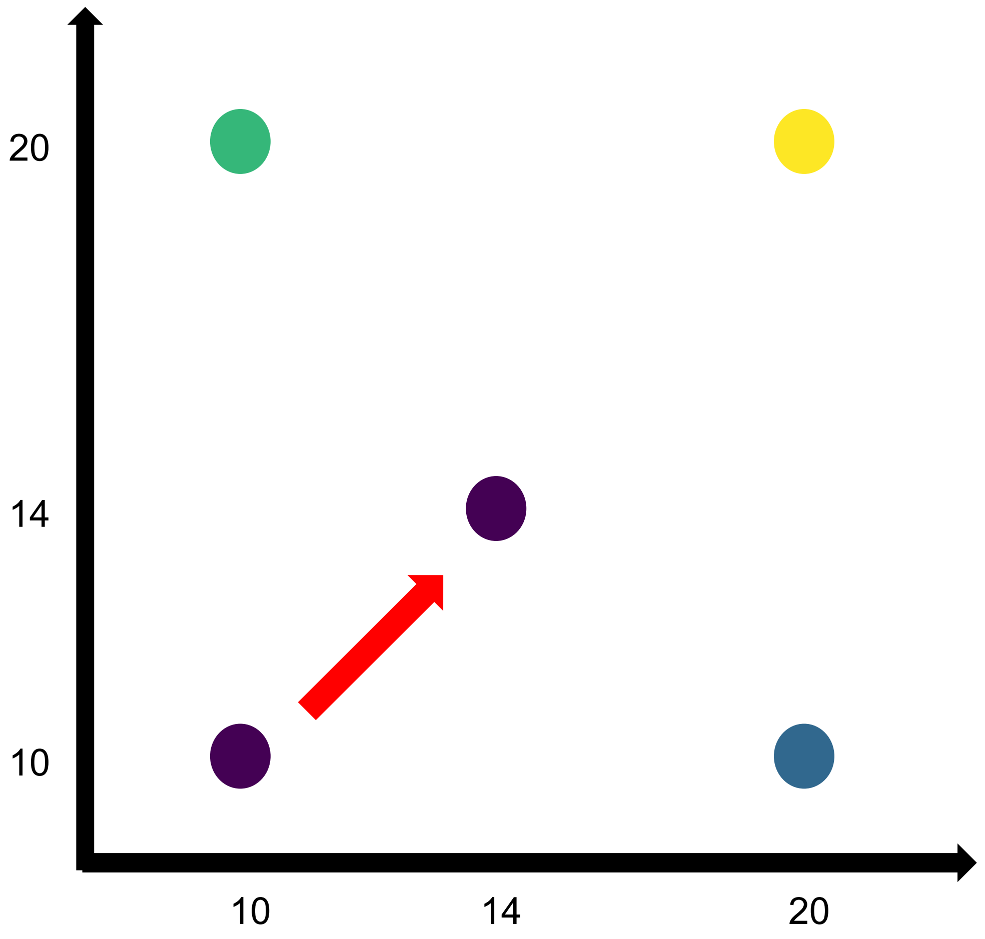


1 linear interpolation along the longitude

Nearest Neighbor Interpolation

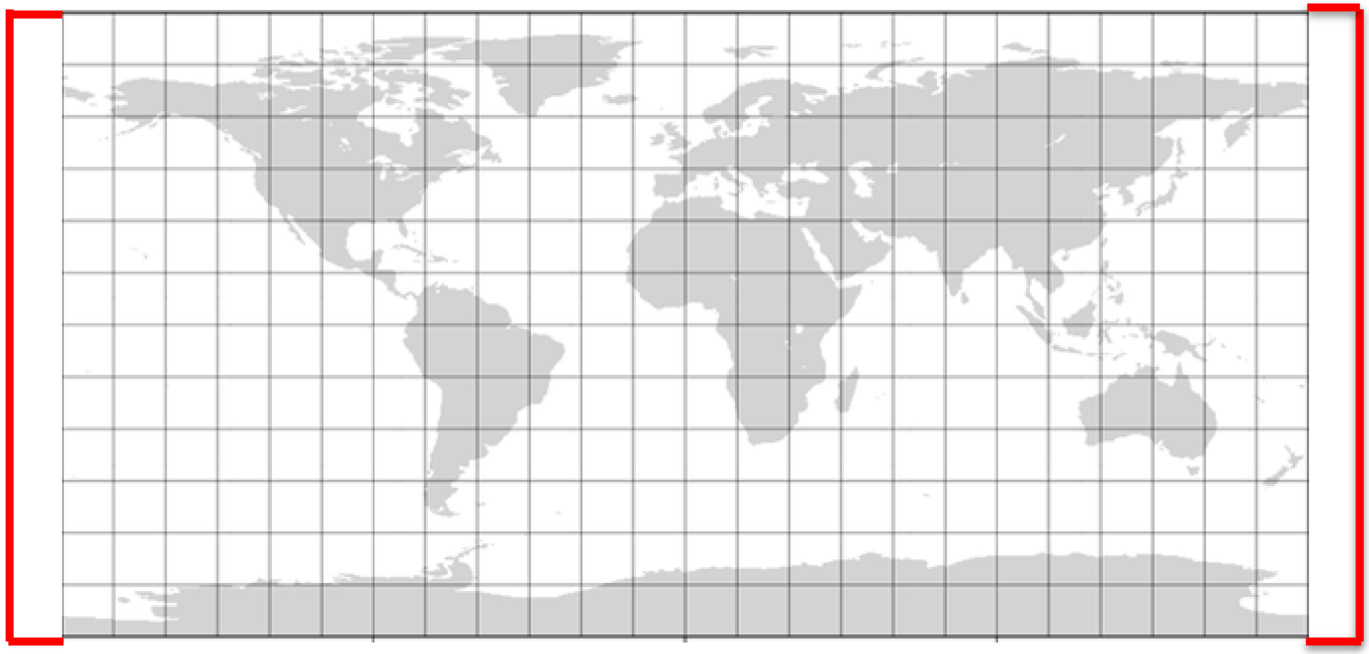


Nearest Neighbor Interpolation



Assign value based on nearest data point

Adding Cyclic Points



Wrapping longitudes to handle geospatial data

dlinint2 -> interp_multidim

```
SUBROUTINE DLININT2(NXI,XI,NYI,YI,FI,ICYCX,NXO,XO,NYO,YO,FO,XIW,
     +                   FXIW,NXI2,XMSG,IOPT,IER)
```

```
def interp_multidim(data_in: supported_types,
                    lat_out: np.ndarray,
                    lon_out: np.ndarray,
                    lat_in: np.ndarray = None,
                    lon_in: np.ndarray = None,
                    cyclic: bool = False,
                    missing_val: np.number = None,
                    method: str = "linear",
                    fill-value: str = "none")
```

xarray

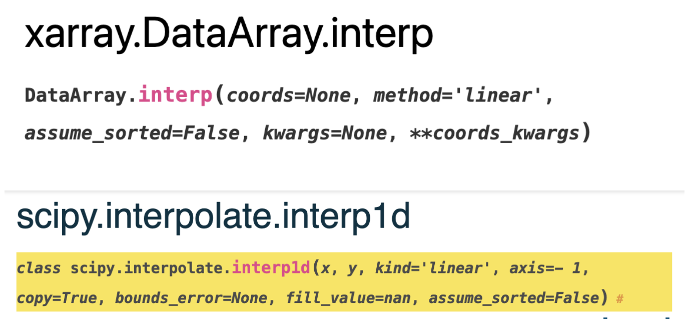
SciPy

NumPy

Changes

`interp_multidim`

- Increased capabilities
  - Method - Bilinear and Nearest Neighbor
  - Extrapolation
- Scalable
  - Eliminate need for auto-chunking
  - Dask compatible
- Improving algorithm flow
  - $O(n^2)$ -> $O(n)$ solution



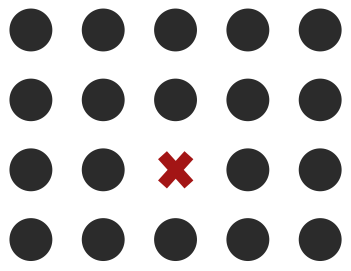
xarray.DataArray.interp

```
DataArray.interp(coords=None, method='linear', assume_sorted=False, kwargs=None, **coords_kwargs)
```

scipy.interpolate.interp1d

```
class scipy.interpolate.interp1d(x, y, kind='linear', axis=- 1, copy=True, bounds_error=None, fill_value=nan, assume_sorted=False) #
```

Room for Improvement

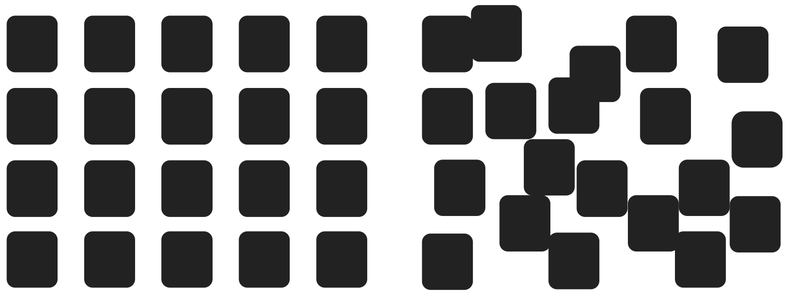


Missing Data

Returns NaN if any of the four surrounding points are NaN

Might create more gaps than desired

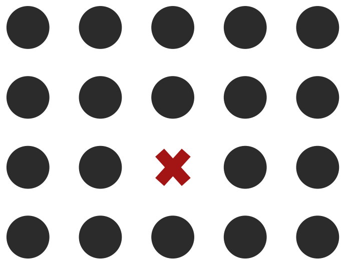
Room for Improvement



Only for regular/rectilinear to regular/rectilinear

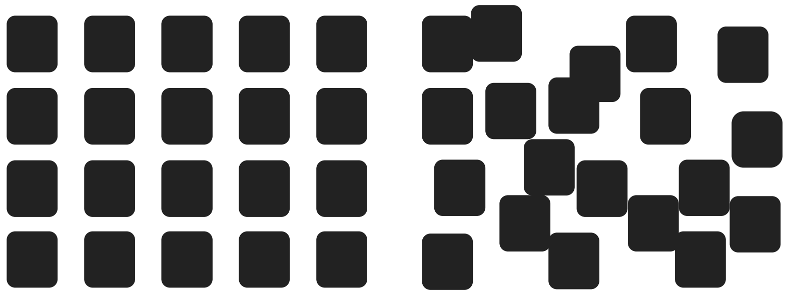
Limited use cases

Structured vs Unstructured Data

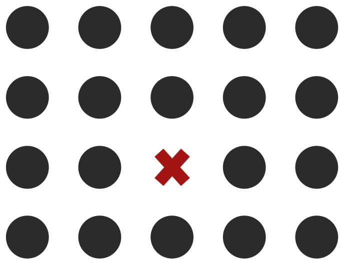


Missing Data

Room for Improvement



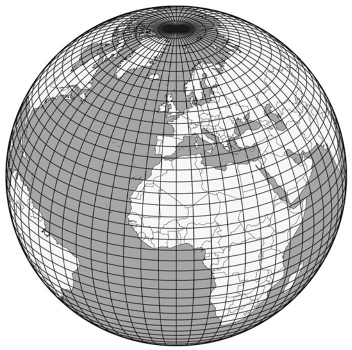
Structured vs Unstructured Data



Missing Data

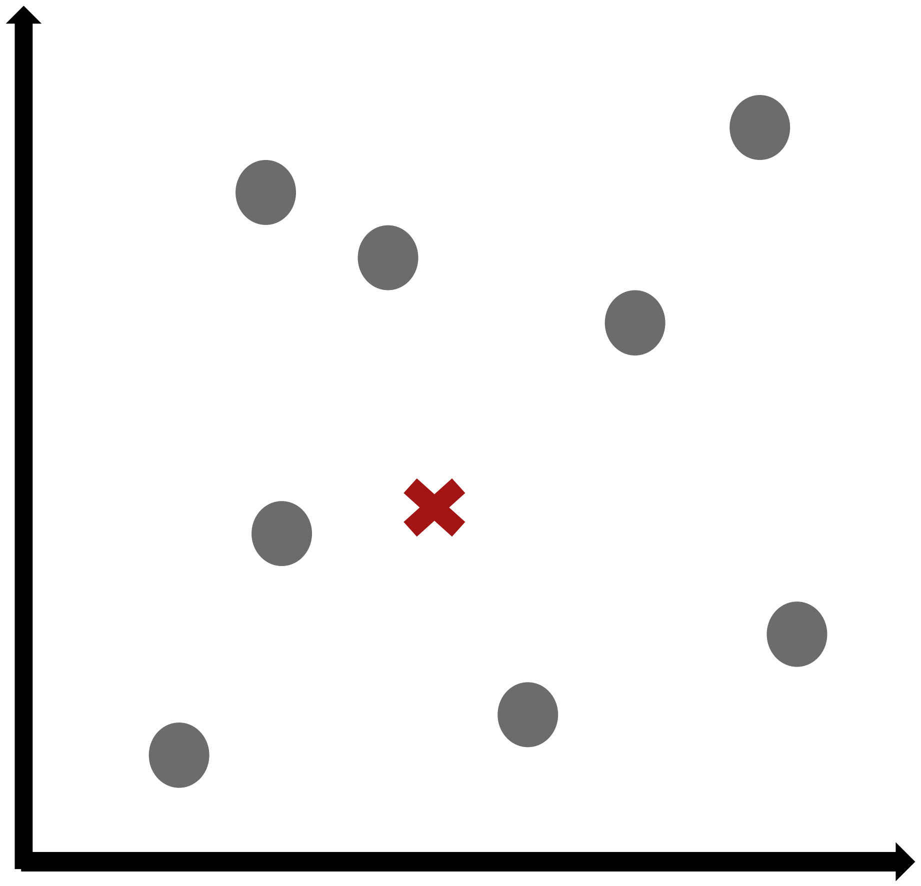
Does not take into account earth's spherical nature

Distance at poles vs equator

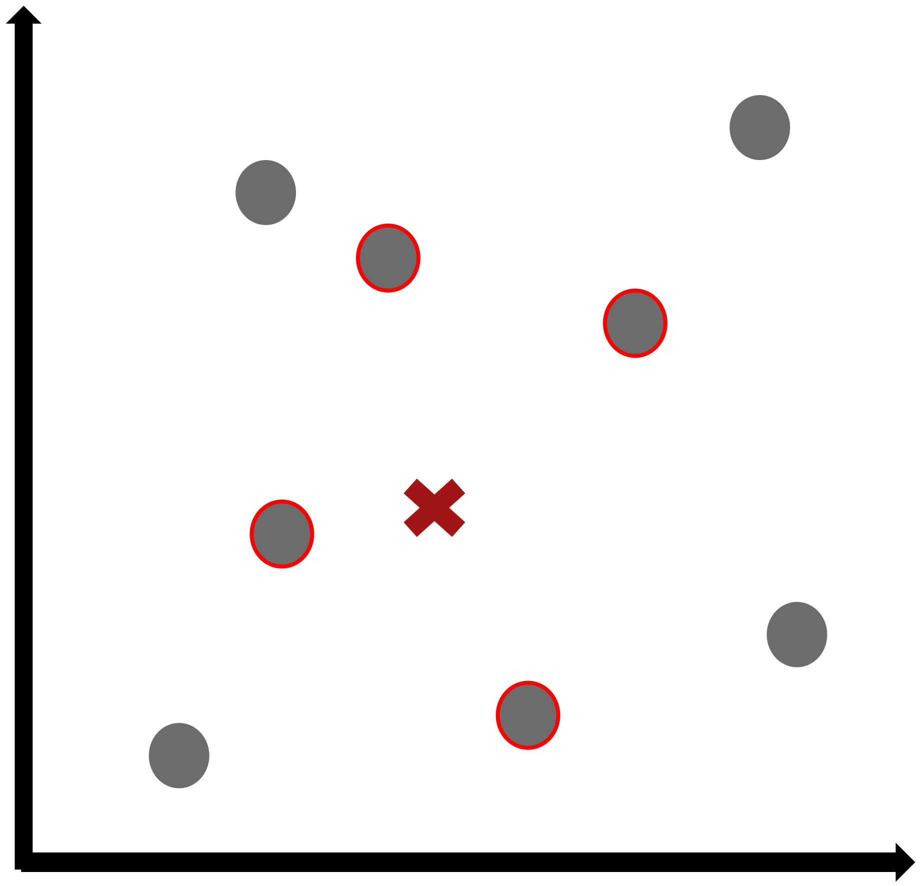


Distance Calculation

Inverse Distance Weighted Interpolation

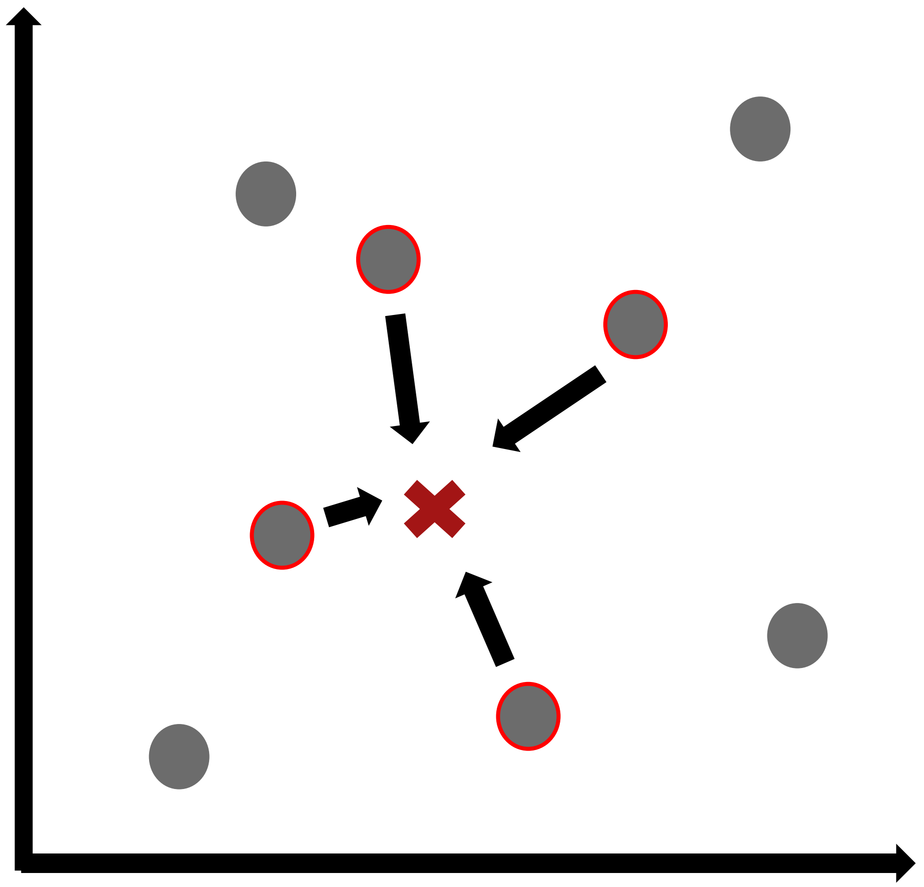


Inverse Distance Weighted Interpolation



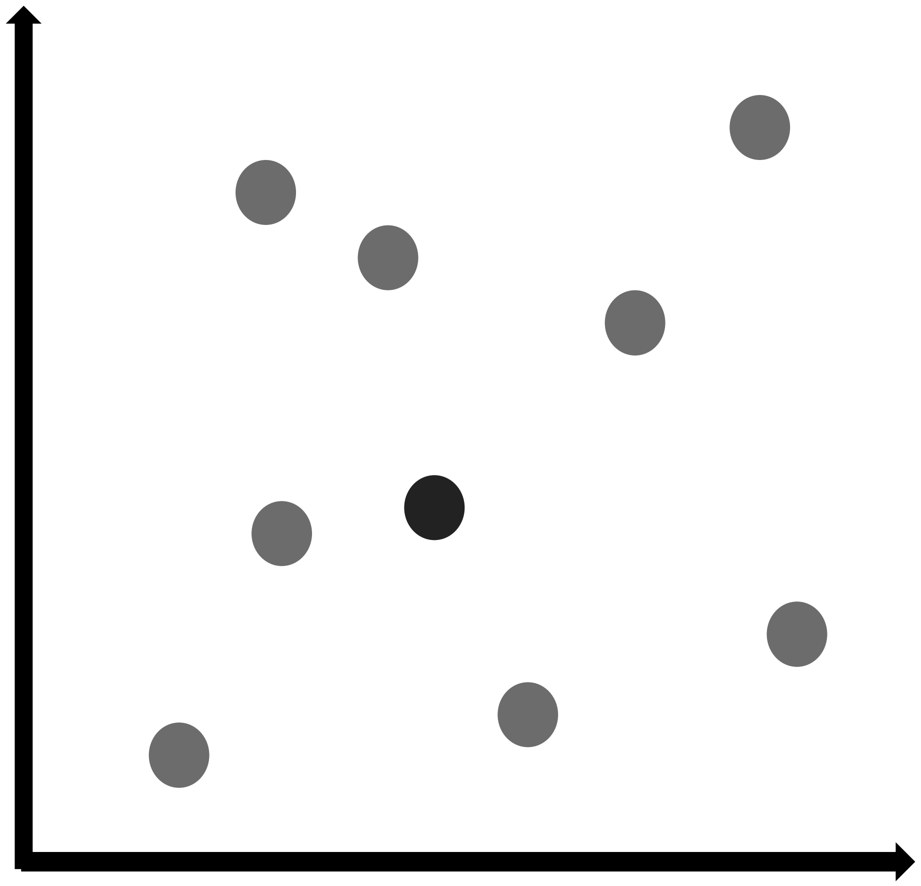
Find four surrounding points

Inverse Distance Weighted Interpolation

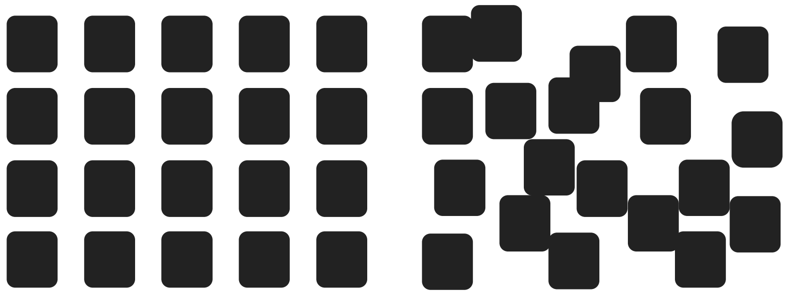


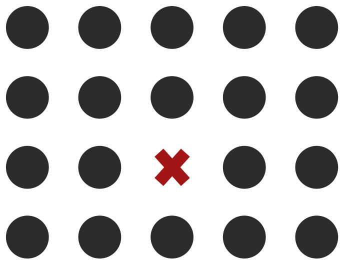
Calculate distance, invert result and raise to a power

Inverse Distance Weighted Interpolation



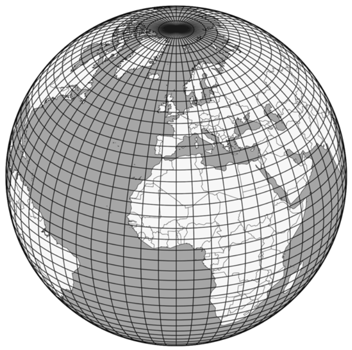
Room for Improvement


Structured vs Unstructured Data



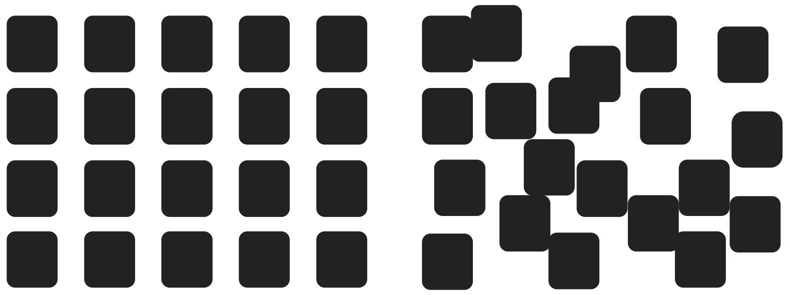
Can use 3 surrounding points instead of 4

Missing Data

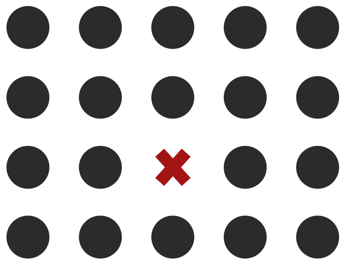

Distance Calculation

Room for Improvement

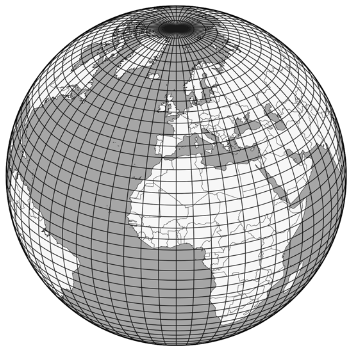
Can be applied to all kinds of grid structures



Structured vs Unstructured Data

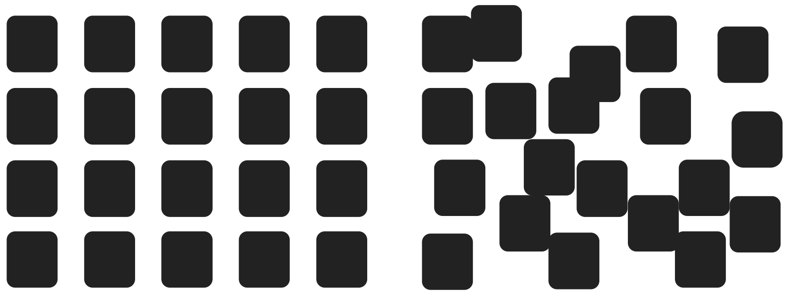


Missing Data

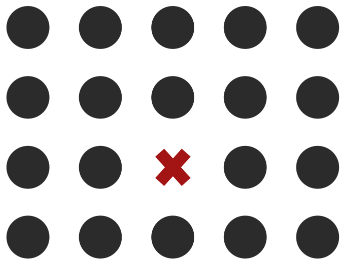


Distance Calculation

Room for Improvement



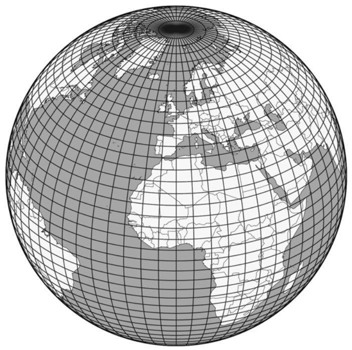
Structured vs Unstructured Data



Missing Data

Spherical distance is calculated

Computation heavy



Distance Calculation

Why Fortran to Python?

Easier to maintain and distribute

Increased Scalability

Increased speed with vectorized functions and updated algorithms

Thank you!

Mentors
Orhan Eroglu and Alea Kootz
SIParCS Team
Virginia Doo, Francesgladys Pulido, Jerry Cyccone, AJ Lauer

Reach out with questions: alinaguha@gmail.com