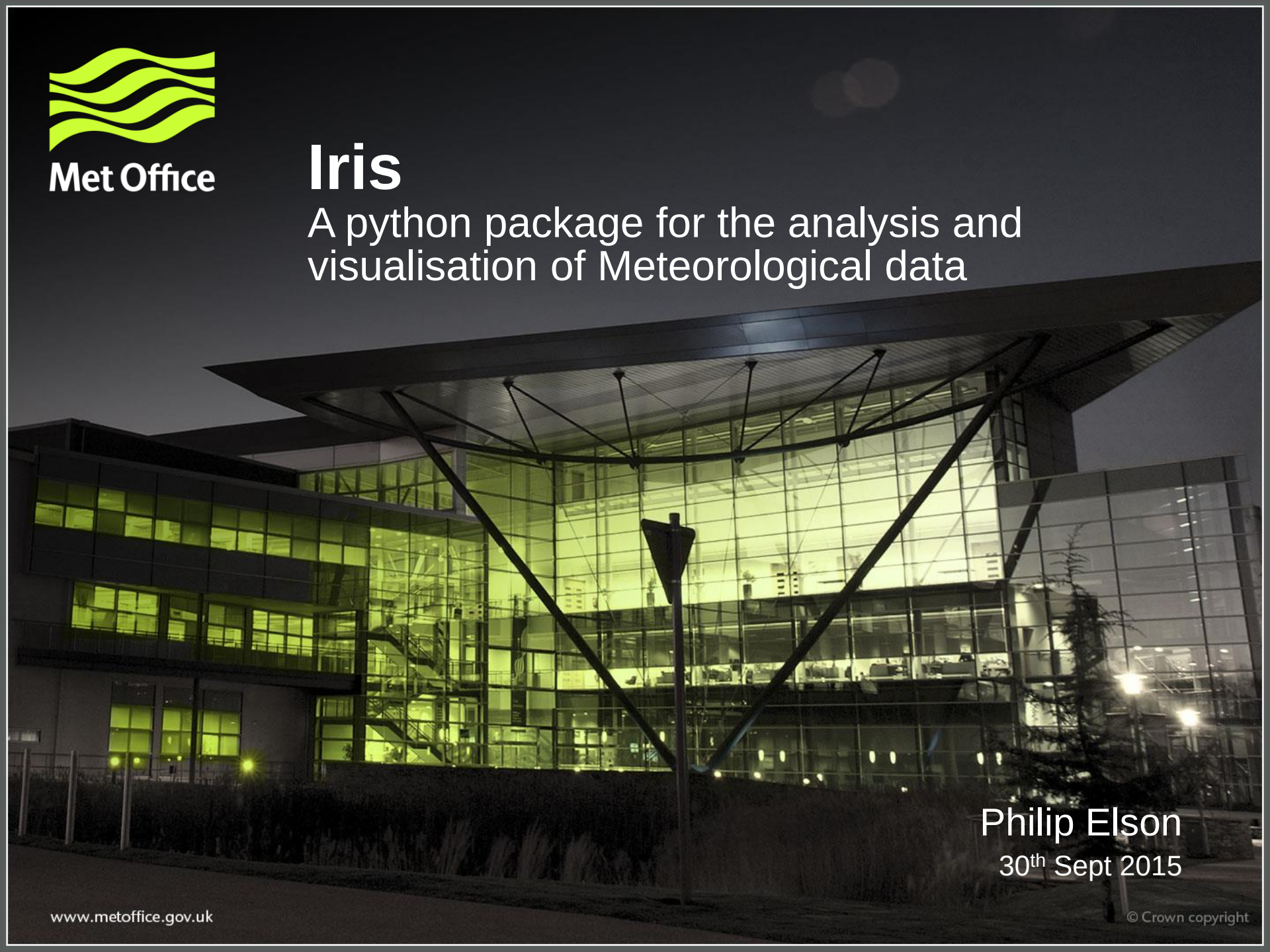


Iris

A python package for the analysis and visualisation of Meteorological data

Philip Elson

30th Sept 2015

Outline

- What is Iris?
- Iris demo
- Using Iris for novel analysis
- Opportunities for combining Iris with other tools

Audience of this talk:

- Those who write code to do data analysis and visualisation

Met Office

What is  Iris ?

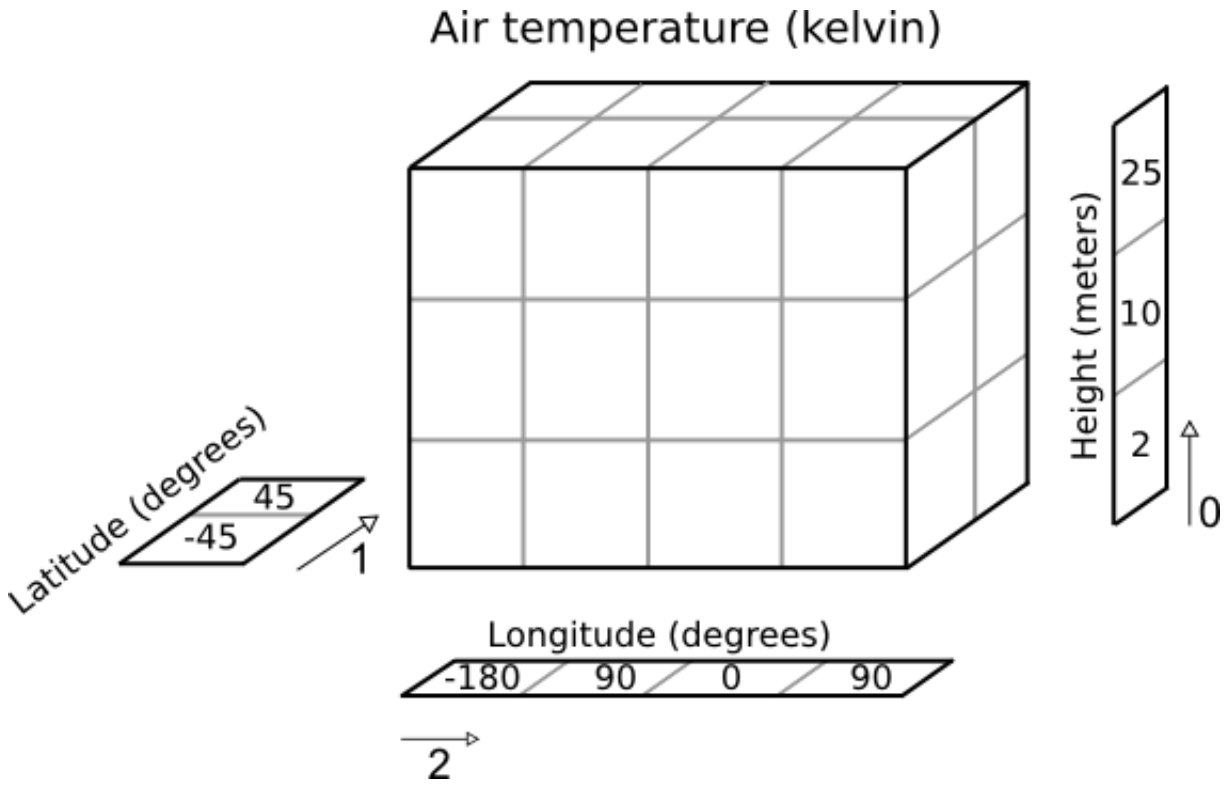
python powered

open source initiative

CF Metadata

NetCDF Climate and Forecast Metadata Convention

Met Office



Met Office

What is Iris ?

netCDF

GRIB

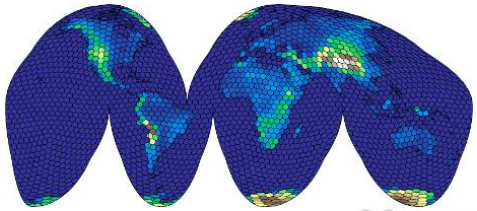
PP

...

Iris

scikit-image
image processing in python

SciPy



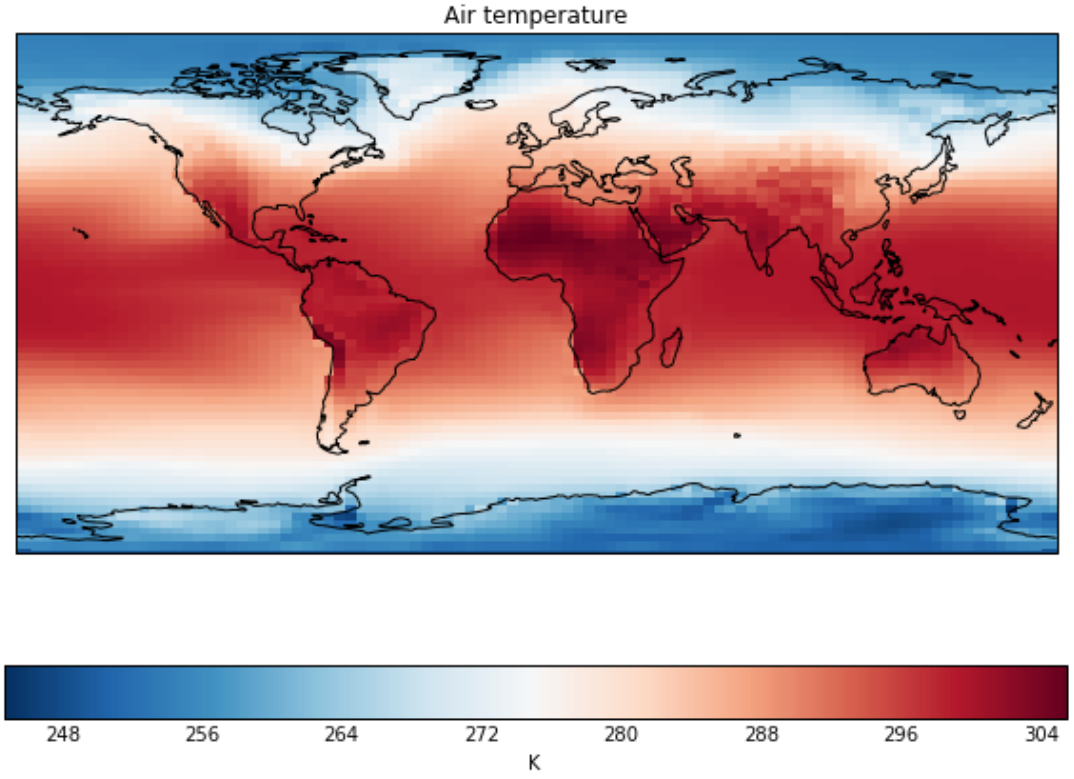
A simple example

Loading a cube

```
>>> import iris

>>> air_temp = iris.load_cube(filename,
                              'air_temperature')
>>> print(air_temp)

air_temperature / (K) (latitude: 73; longitude: 96)
Scalar coordinates:
   pressure:                1000.0 hPa
   time: 1998-12-01 00:00:00, bound=(1994-12-01 00:00:00,
                                     1998-12-01 00:00:00)
Attributes:
   STASH: m01s16i203
   source: Data from Met Office Unified Model
```

Plotting with matplotlib

```
>>> import matplotlib.pyplot as plt
>>> import iris.quickplot as qplt

>>> qplt.pcolormesh(air_temp, cmap='RdBu_r')
>>> plt.gca().coastlines()
```



Output:

- PNG
- PDF
- PS
- ...

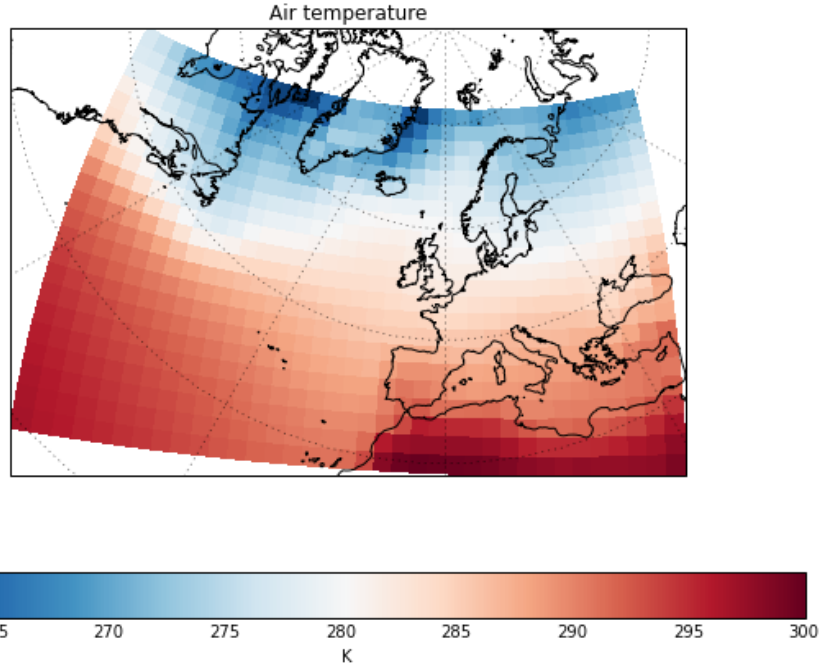
Regridding and interpolation

```
>>> from iris.analysis import Linear

>>> exeter = [('longitude', [-3.5]),
              ('latitude', [50.7])]
>>> exeter_temp = air_temp.interpolate(exeter,
                                       Linear())


>>> mslp_euro = iris.load_cube(filename2)

>>> air_temp_euro = air_temp.regrid(mslp_euro,
                                    Linear())
```

Typically, Iris takes cubes as input, and returns cubes as output.

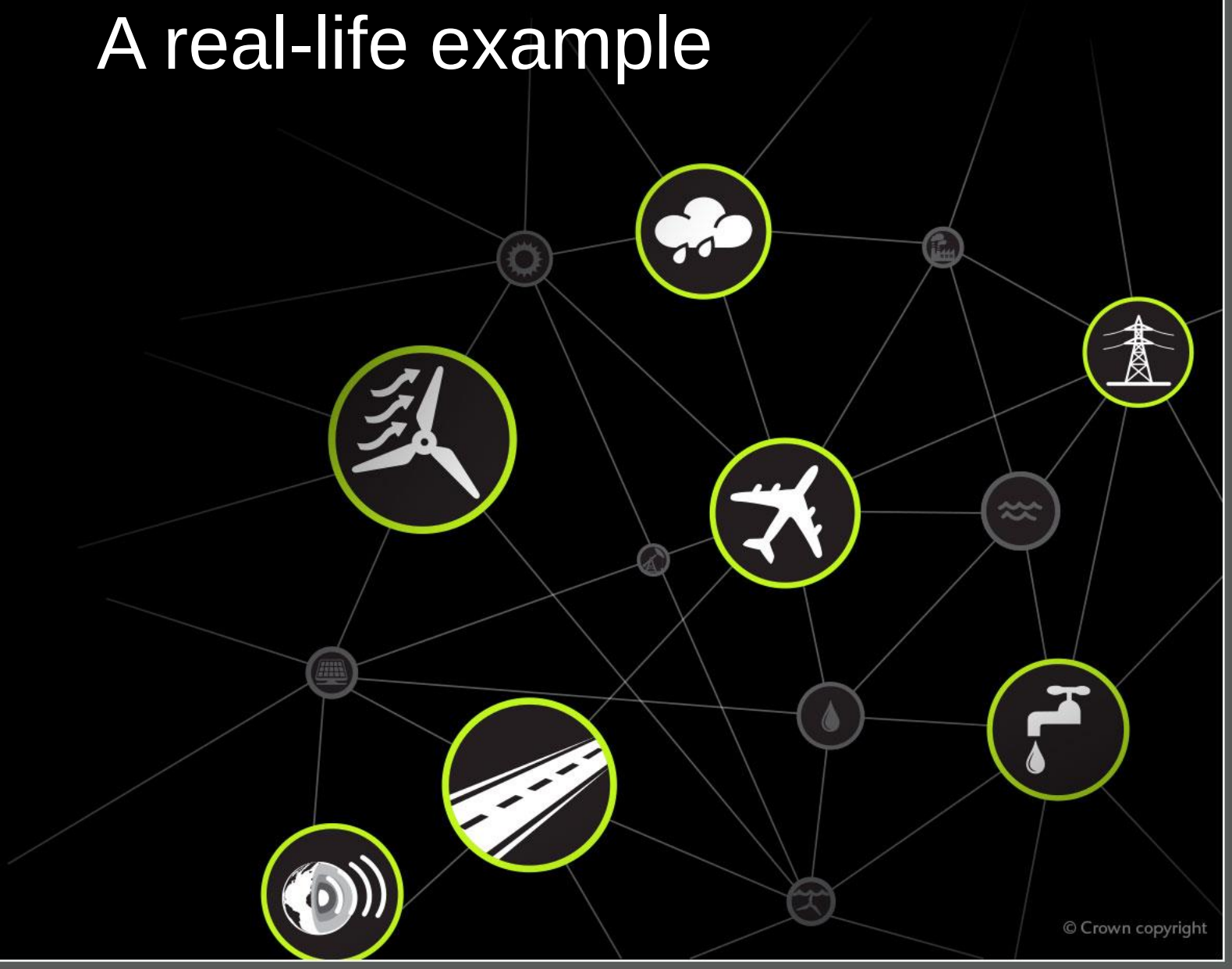
Maps with cartopy

Maps in Iris are drawn by cartopy, a python package developed to solve common dateline and pole problems seen with traditional mapping libraries.

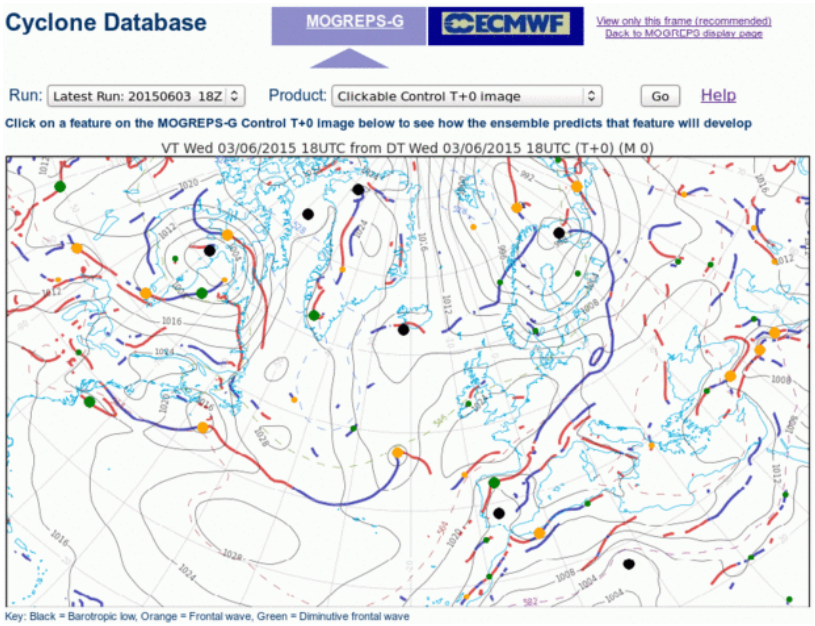



```
>>> from cartopy.crs as ccrs

>>> ax = plt.axes(projection=ccrs.NorthPolarStereo())
>>> qplt.pcolormesh(air_temp_euro, cmap='RdBu_r')

>>> ax.coastlines('50m')
>>> ax.gridlines()
```

A real-life example



MOGREPS-G Cyclone Database



An algorithm to identify and track fronts and cyclonic features, based on:

Hewson, T.D. & H.A. Titley, 2010: Objective identification, typing and tracking of the complete life-cycles of cyclonic features at high spatial resolution. Meteorol. Appl., 17, 355-381.

Implementing the algorithm

- Load the phenomenon

- Regrid and interpolate data to specific to vertical levels

- Compute isolines for locating phenomenon + isosurfaces for masking phenomenon, based on thresholds from paper

- Compute intersection of isosurfaces and isolines to identify cyclonic features

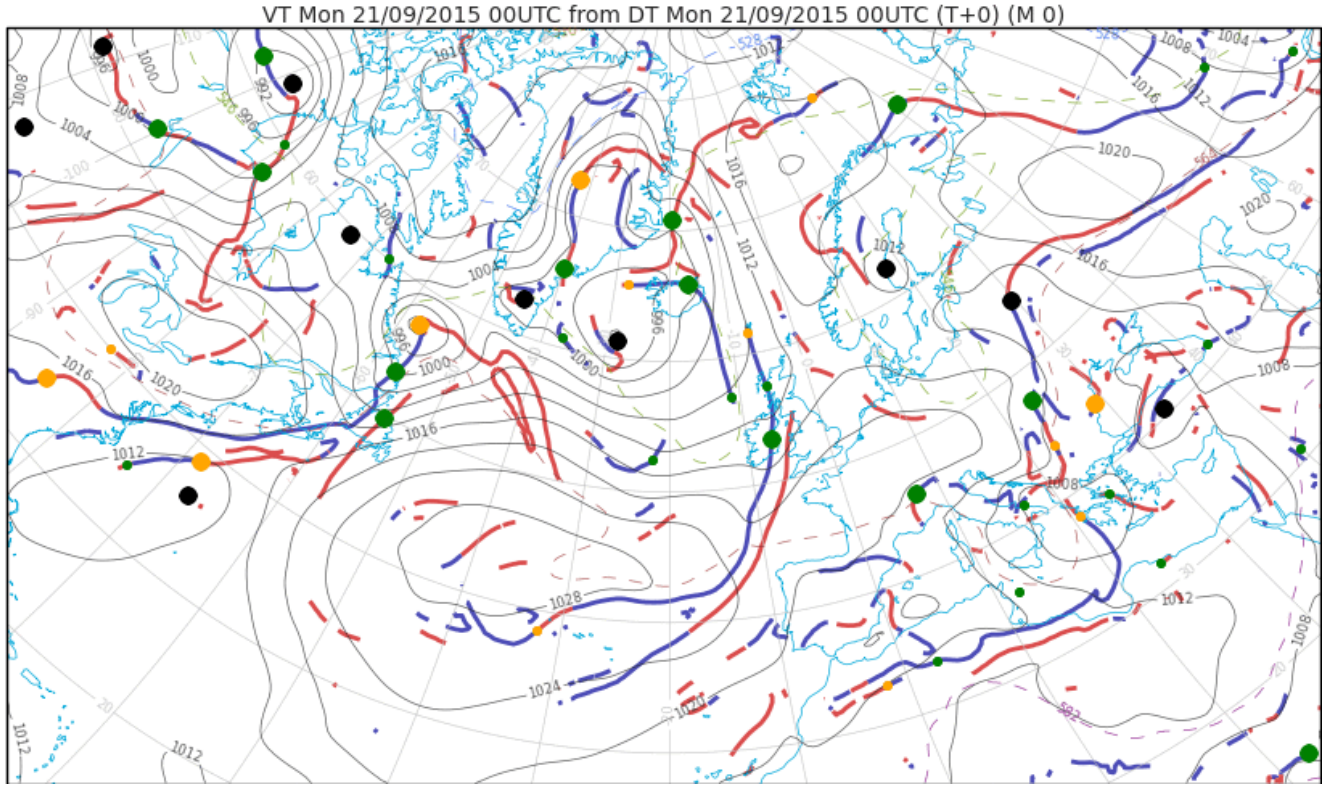
Met Office

- Classify cyclonic features based on phenomenon values

Iris

- Visualise cyclonic features and the underlying diagnostics

Iris matplotlib Cartopy



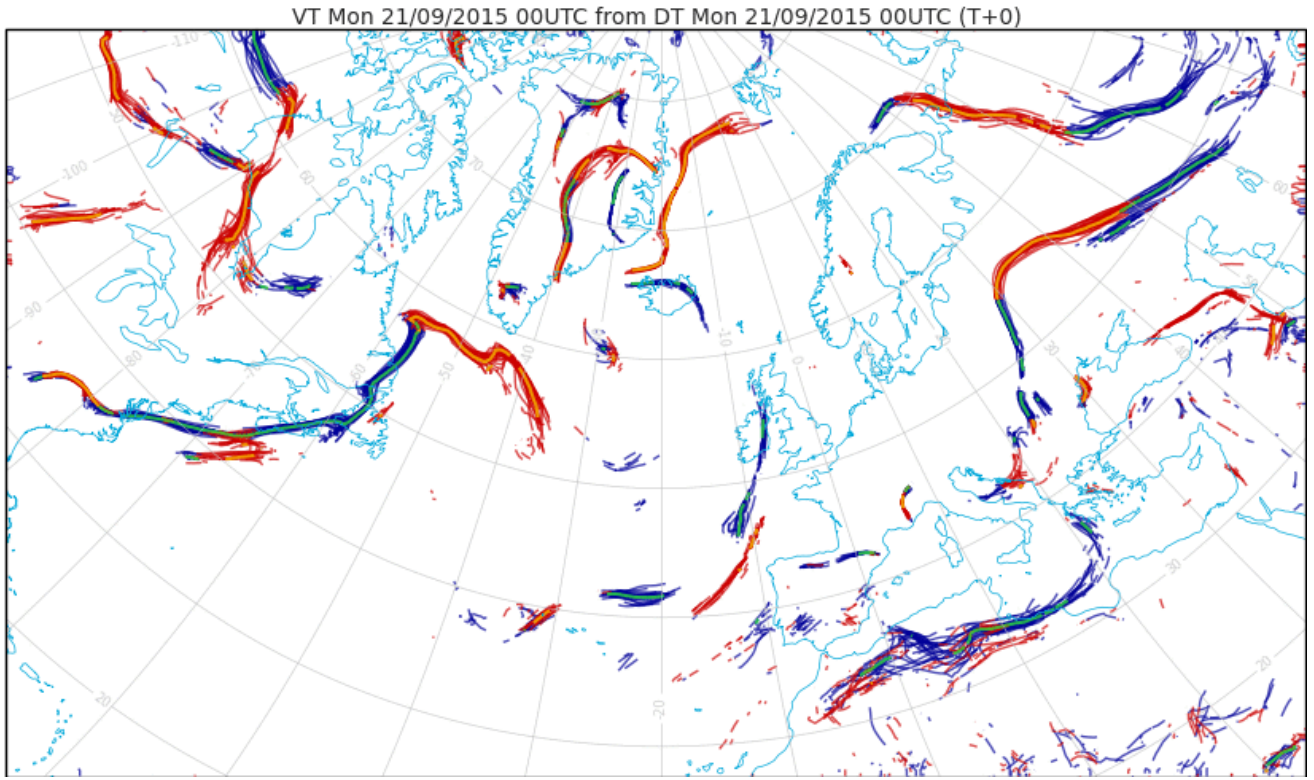
Barotropic Lows **Frontal Waves** **Diminutive Waves**

Met Office

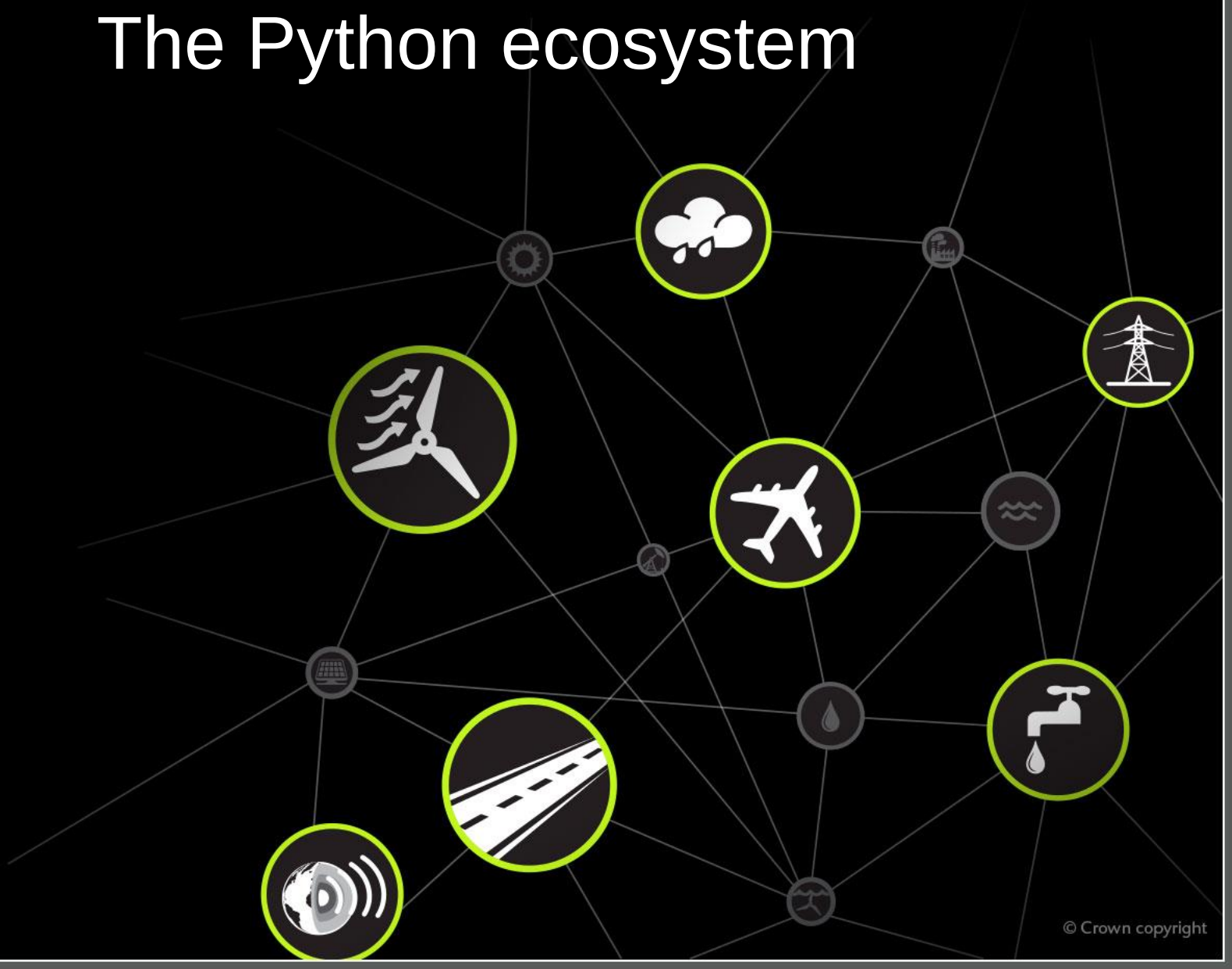
- Visualise fronts as a spaghetti plot

matplotlib

Cartopy



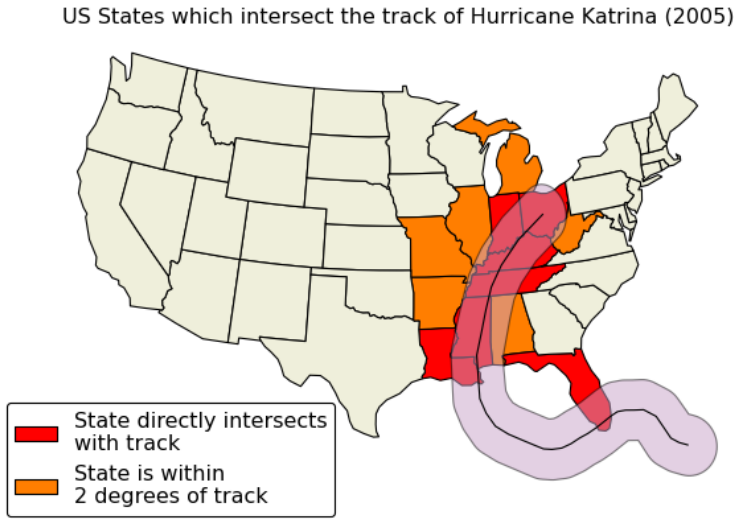
The Python ecosystem



Opportunities within Python

Recent GIS tools:

- Shapely
- Cartopy
- Fiona
- RasterIO
- QGIS



http://scitools.org.uk/cartopy/docs/latest/examples/hurricane_katrina.html

A recent publication combining shapely and Iris to assess the skill of seasonal prediction of Hurricane landfall frequencies in the North Atlantic:

Camp, J., Roberts, M., MacLachlan, C., Wallace, E., Hermanson, L., Brookshaw, A., Arribas, A., Scaife, A. A., Mar. 2015. *Seasonal forecasting of tropical storms using the Met Office GloSea5 seasonal forecast system*. Quarterly Journal of the Royal Meteorological Society

Opportunities within Python

Large data manipulation:

- Cython
- Numba

Tools to optimise slow for-loops using static typing and JIT compilation for C-like performance

- Biggus
- Dask

Opportunities within Python

Large data manipulation:

- Cython
- Numba

- Biggus
- Dask

Biggus example:

```
>>> print(data)
<Array shape=(80640, 4, 144, 192)
        dtype=dtype('float32') size=33.22 GiB>

>>> stats = [biggus.mean(data, axis=0),
             biggus.max(data, axis=0),
             biggus.min(data, axis=0)]

>>> biggus.ndarrays(stats)
```

Result in ~4m45s on an Intel Xeon E5520 with 8GiB memory, bound by I/O not CPU.

Iris is using Biggus for many of it operations. This means that we can load, analyse and save cubes way beyond the available system memory.

Installing Iris

```
conda install iris  --channel SciTools
```

Conda can be downloaded as part of “**miniconda**”: http://conda.pydata.org/miniconda.html

Met Office

Questions

Further reading:
github.com/scitools/courses

Links from presentation:
github.com/pelson/ecmwf-vis-2015

Github: github.com/pelson
Twitter: @pypelson