National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

Centre for Environmental Data Analysis
SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL

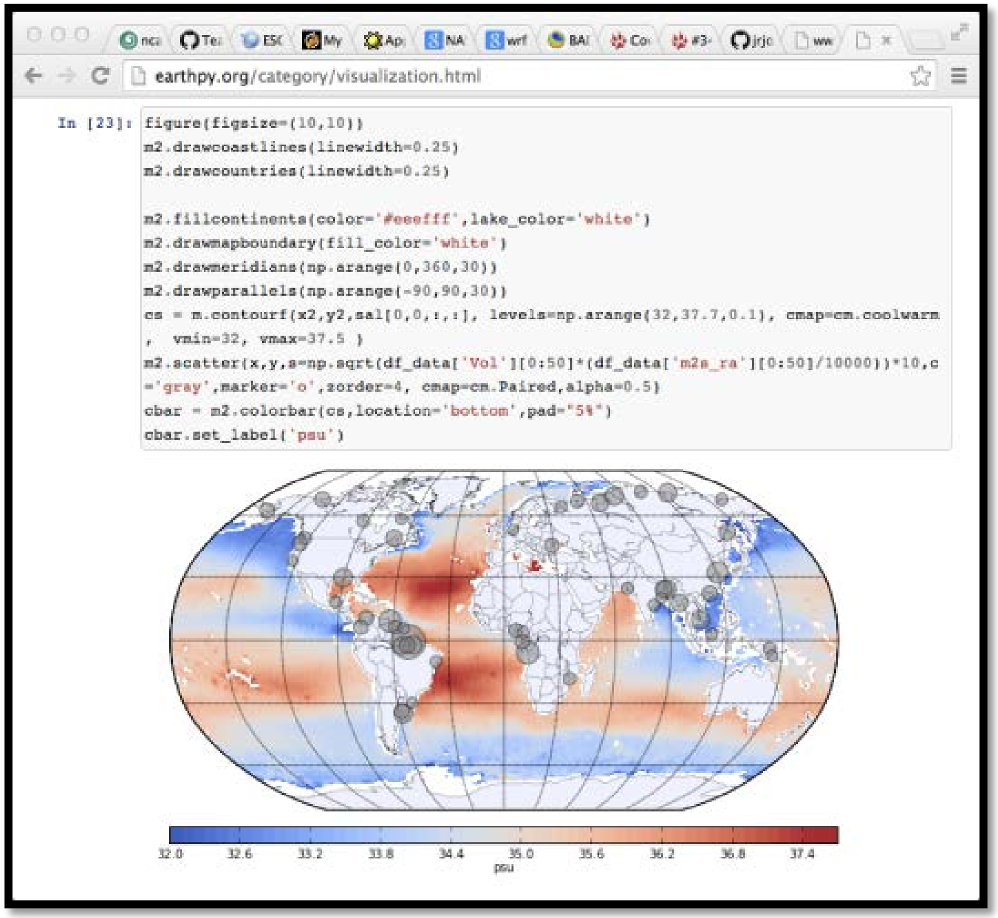
Python tools for analysing data

There are lots of tools

Many (Atmospheric) Science libraries are available for Python:

- `netCDF4-python`
- `cdat`
- `cf-python`
- `Iris`
- `pyNGL`

- Many others
  - `OpenClimateGIS`
  - `pyTroll`
  - …



earthpy.org/category/visualization.html

In [23]:
```
figure(figsize=(10,10))
m2.drawcoastlines(linewidth=0.25)
m2.drawcountries(linewidth=0.25)

m2.fillcontinents(color='#eeefff',lake_color='white')
m2.drawmapboundary(fill_color='white')
m2.drawmeridians(np.arange(0,360,30))
m2.drawparallels(np.arange(-90,90,30))
cs = m.contourf(x2,y2,sal[0,0,:,:], levels=np.arange(32,37.7,0.1), cmap=cm.coolwarm
,  vmin=32, vmax=37.5 )
m2.scatter(x,y,s=np.sqrt(df_data['Vol'][0:50]*(df_data['m2s_ra'][0:50]/10000))*10,c
='gray',marker='o',zorder=4, cmap=cm.Paired,alpha=0.5)
cbar = m2.colorbar(cs,location='bottom',pad="5%")
cbar.set_label('psu')
```

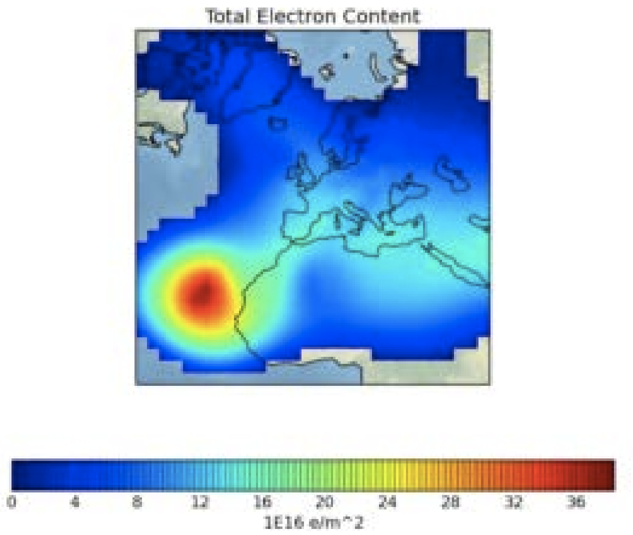
Higher-level data tools

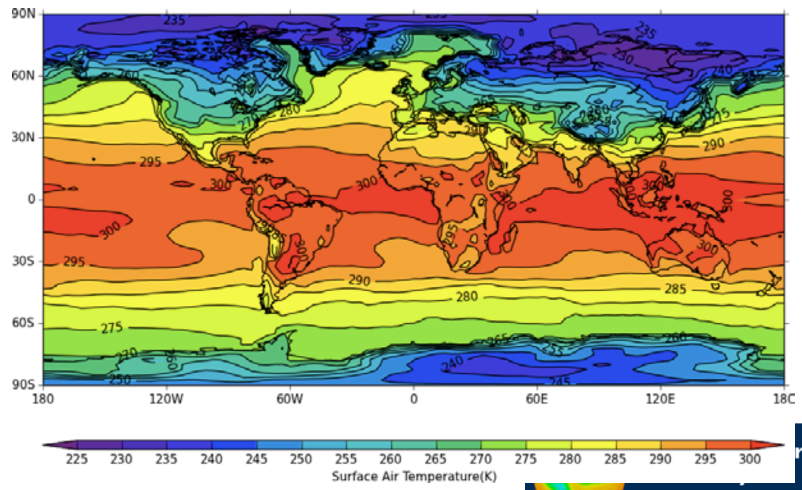
Iris

- Developed by the Met Office
- Reads NetCDF, PP and Grib
- Supports CF-conventions via the "Cube"
- Plotting via cartopy or matplotlib + basemap

cf-python

- Developed by Uni. Reading
- Reads NetCDF and PP
- Strict interpretation of the CF-conventions
- Plotting via cfplot or matplotlib + basemap





What makes these tools useful?

- The biggest gain from using these tools is that you can work with higher-level objects that know about real-world coordinate systems.

- Hence you can subset a variable based on temporal, spatial and other constraints rather than using slicing in index space.

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

Centre for Environmental Data Archival
SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL

Iris - a Python package for data analysis and visualisation

Iris

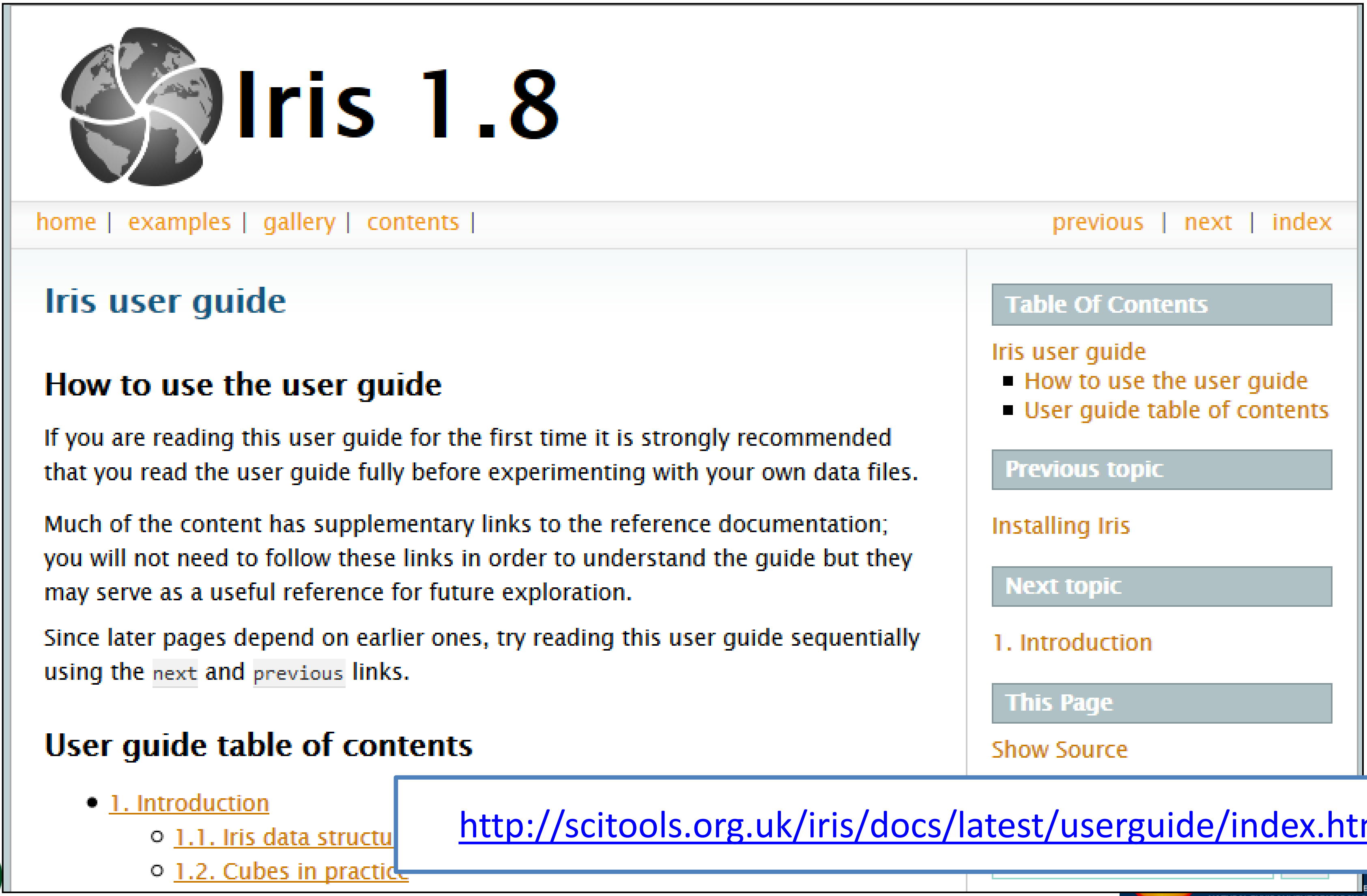
What is Iris?

Iris is publicised as:

"A Python library for Meteorology and Climatology"

- Implements the CF-netCDF Data Model in its "cube" design.
- Supports read/write access to a range of data formats (including CF-netCDF, GRIB, and PP).
- Fundamental data manipulation operations, such as arithmetic, interpolation, and statistics;
- A range of integrated plotting options.

Documentation



Iris 1.8

home | examples | gallery | contents | previous | next | index

Iris user guide

How to use the user guide

If you are reading this user guide for the first time it is strongly recommended that you read the user guide fully before experimenting with your own data files.

Much of the content has supplementary links to the reference documentation; you will not need to follow these links in order to understand the guide but they may serve as a useful reference for future exploration.

Since later pages depend on earlier ones, try reading this user guide sequentially using the `next` and `previous` links.

User guide table of contents

Table Of Contents

Previous topic

Installing Iris

Next topic

1. Introduction

This Page

Show Source

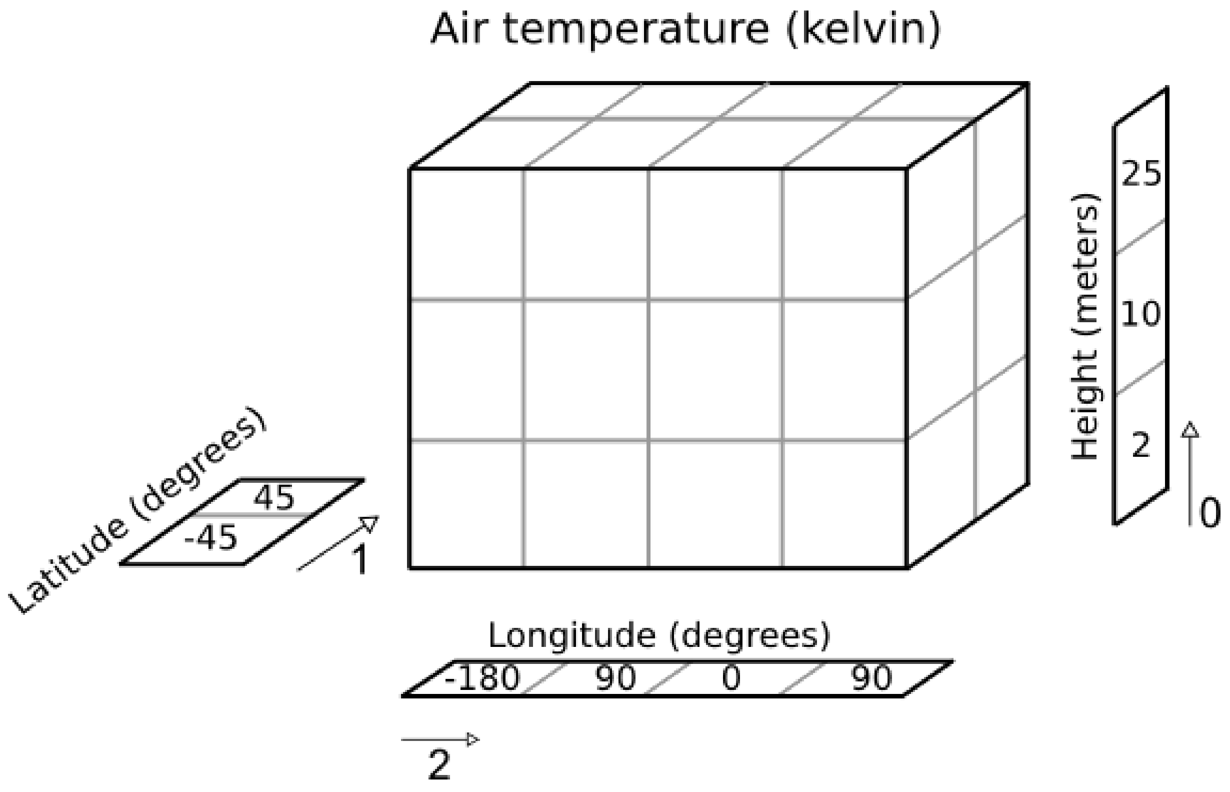
http://scitools.org.uk/iris/docs/latest/userguide/index.html

Main concept - the "cube"

A cube consists of:

- a standard name and/or a long name and unit;
- a data array;
- a collection of coordinates and associated data dimensions on the cube’s data array;
- an attributes dictionary for metadata;
- a list of cell methods (e.g. "mean over time")
- a list of coordinate “factories” used to derive coordinates from the values of other coordinates in the cube;

The "cube" - in a picture



So what can Iris do?

There are too many features to describe in detail. Here are some things that extend functionality we have seen in lower level libraries:

Loading data from multiple files:

```
import iris
filename = iris.sample_data_path('GloSea4', '*.nc')
cubes = iris.load(filename)
```

Constrained loading

Constrained by CF standard name:

```
filename = iris.sample_data_path('uk_hires.nc')
cubes = iris.load(filename,
                  ['air_potential_temperature',
                   'specific_humidity'])
```

Constrained by coordinate selection:

```
filename = iris.sample_data_path('uk_hires.nc')
level_10_or_12_fp_6 = iris.Constraint(
                 model_level_number=[10, 16],
                 forecast_period=6)
cubes = iris.load(filename, level_10_or_16_fp_6)
```

Cube slicing/indexing - like numpy

Cubes can be sliced and indexed like numpy arrays:

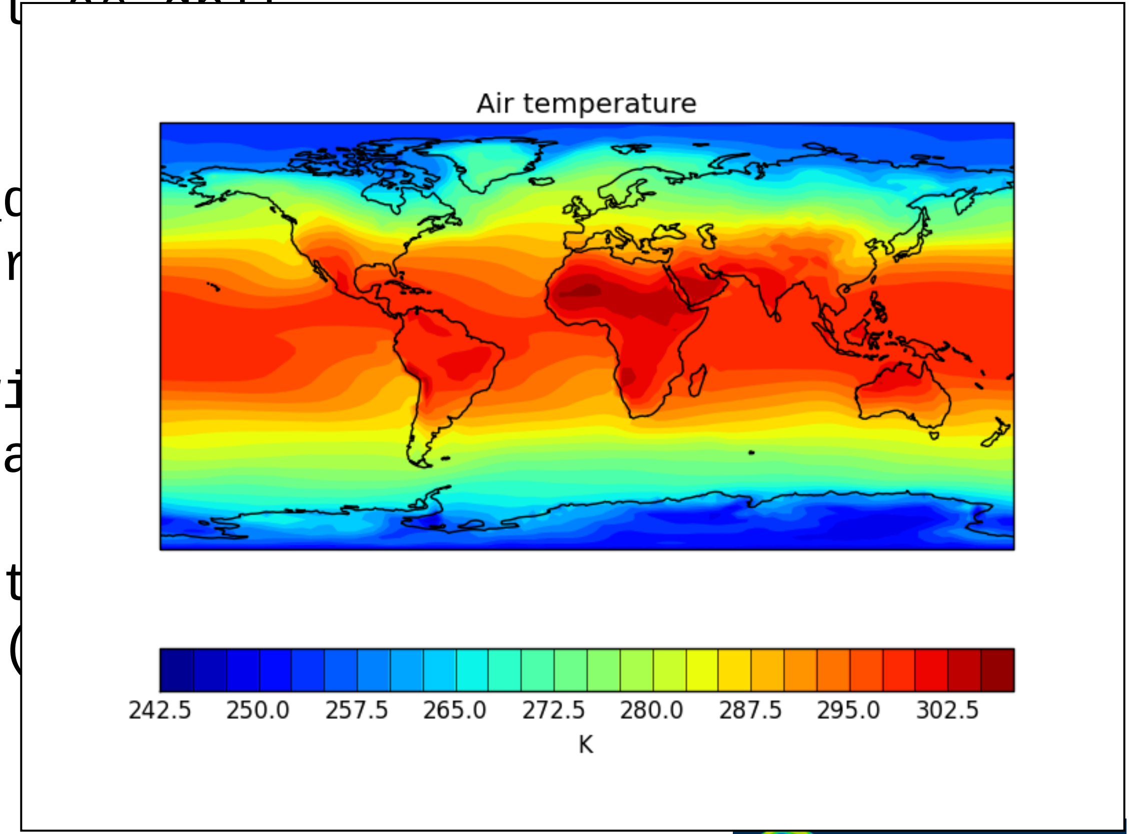
```
# get the first element of the first dimension
# (+ every other dimension)
print cube[0]

# get the first 4 elements of the first dimension
# (+ every other dimension)
print cube[0:4]

# Get the first element of the first and third
dimension (+ every other dimension)
print cube[0, :, 0]
```

Plotting a cube

```
import matplotlib.pyplot as plt

import iris
import iris.quickplot as qplt

# Load the data
fname = iris.sample_data_path('air_temp.pp')
temperature_cube = iris.load_cube(fname)

# Draw the contour with 25 levels.
qplt.contourf(temperature_cube, 25)

# Add coastlines to the map created by contourf.
plt.gca().coastlines()

plt.show()
```

Plotting a cube

```
import matplotlib.pyplot as plt

import iris
import iris.quickplot as qplt

# Load the data
fname = iris.sample_d
temperature_cube = ir

# Draw the contour wi
qplt.contourf(tempera

# Add coastlines to t
plt.gca().coastlines(

plt.show()
```



National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

Centre for Environmental Data Analysis
SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL

Collapsing cubes

Cubes can be collapsed using various statistical/mathematical operations.

Calculate a time-series mean:

```
air_temp_mean = air_temp.collapsed('time',
                                   iris.analysis.MEAN)
```

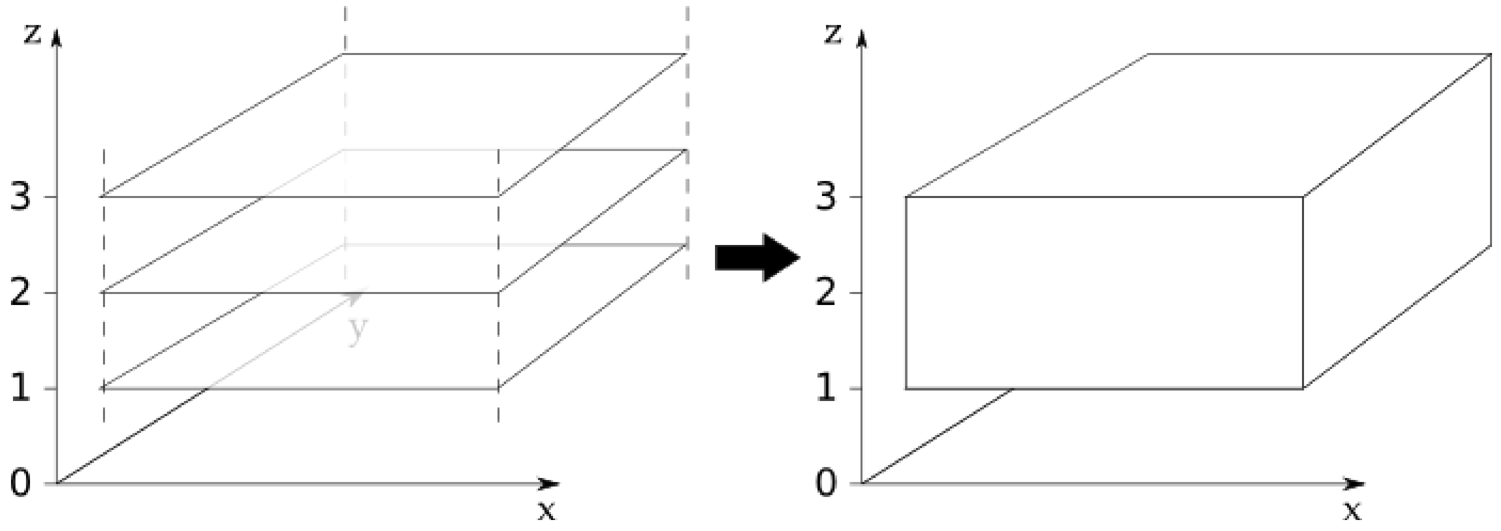
Merging cubes

```
>>> print cubes
0: air_temperature / (kelvin)          (y: 4; x: 5)
1: air_temperature / (kelvin)          (y: 4; x: 5)
2: air_temperature / (kelvin)          (y: 4; x: 5)

>>> print cubes[0]
air_temperature / (kelvin)             (y: 4; x: 5)
 ...      z: 1 meters
>>> print cubes[1]
air_temperature / (kelvin)             (y: 4; x: 5)
 ...      z: 2 meters
>>> print cubes[2]
air_temperature / (kelvin)             (y: 4; x: 5)
 ...      z: 3 meters

>>> print cubes.merge()
0: air_temperature / (kelvin)          (z: 3; y: 4; x: 5)
```

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

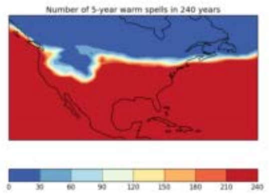
Centre for Environmental Data Analysis
SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL

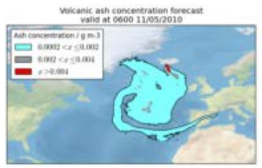
Merging cubes

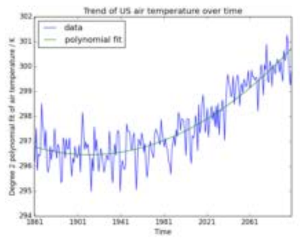


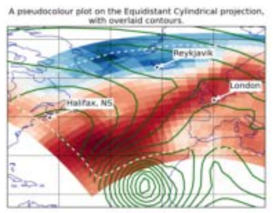
And plotting

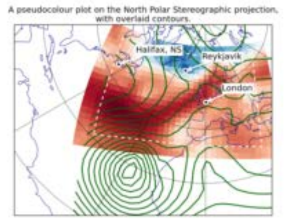
See: http://scitools.org.uk/iris/docs/latest/gallery.html

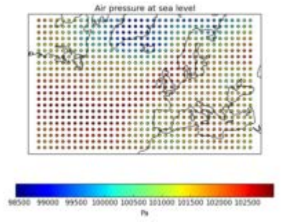


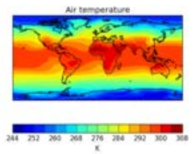

Volcanic ash concentration forecast
valid at 0600 11/05/2010
Ash concentration / g m-3

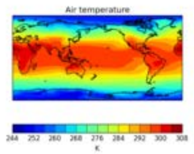


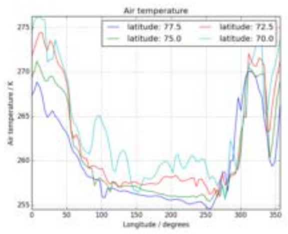

A pseudocolour plot on the Equidistant Cylindrical projection, with overlaid contours.
Reykjavik
London
Halifax, NS

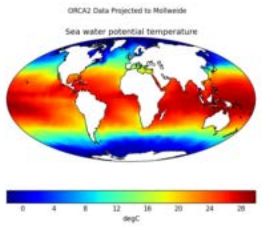

A pseudocolour plot on the North Polar Stereographic projection, with overlaid contours.
Halifax, NS
Reykjavik
London

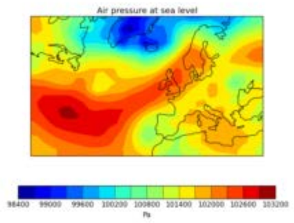


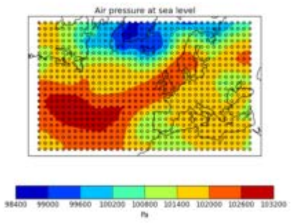


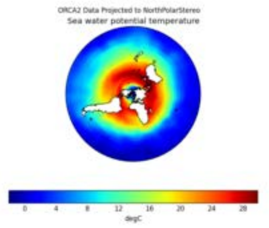


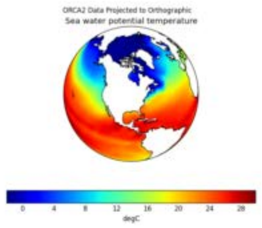


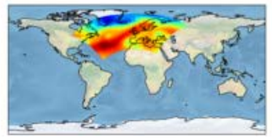

ORCA2 Data Projected to Mollweide
Sea water potential temperature
degC






ORCA2 Data Projected to NorthPolarStereo
Sea water potential temperature
degC


ORCA2 Data Projected to Orthographic
Sea water potential temperature
degC



National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

Environmental Data Analysis
SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL

Further reading

Iris documentation:
http://scitools.org.uk/iris/docs/latest

Iris image gallery:
http://scitools.org.uk/iris/docs/latest/gallery.html