Introduction - cf-python and cf-plot

The cf in cf-python and cf-plot are to indicate that they are underpined by CF - Climate and Forecast (CF) Metadata Conventions. http://cfconventions.org (http://cfconventions.org)

cf-python - The python cf package implements the CF data model for the reading, writing and processing of data and metadata. https://cfpython.bitbucket.io (https://cfpython.bitbucket.io)

cf-plot - A set of Python routines for making the common contour, vector and line plots that climate researchers use. can also plot Numpy arrays of data. http://ajheaps.github.io/cf-plot (http://ajheaps.github.io/cf-plot)

Read, select, write example

In [1]:

```
# Inline images in Ipython Notebook - not needed in Python
%matplotlib inline
```

In [2]:

```
# Import cf-python and cf-plot
import cf, cfplot as cfp
```

In [3]:

```
# Read a data file
f=cf.read('ncas_data/data1.nc')
```

In [4]:

```
# View the contents of the file
f
```

Out[4]:

```
[<CF Field: long_name:Potential vorticity(time(1), pressure(23), latitude(160), longitude(320))
 K m**2 kg**-1 s**-1>,
 <CF Field: long_name:Ozone mass mixing ratio(time(1), pressure(23), latitude(160), longitude(3
20)) kg kg**-1>,
 <CF Field: air_temperature(time(1), pressure(23), latitude(160), longitude(320)) K>,
 <CF Field: atmosphere_relative_vorticity(time(1), pressure(23), latitude(160), longitude(320))
 m**2 s**-1>,
 <CF Field: atmosphere_relative_vorticity(time(1), pressure(23), latitude(160), longitude(320))
 s**-1>,
 <CF Field: divergence_of_wind(time(1), pressure(23), latitude(160), longitude(320)) m**2 s**-1
>,
 <CF Field: divergence_of_wind(time(1), pressure(23), latitude(160), longitude(320)) s**-1>,
 <CF Field: eastward_wind(time(1), pressure(23), latitude(160), longitude(320)) m s**-1>,
 <CF Field: geopotential(time(1), pressure(23), latitude(160), longitude(320)) m**2 s**-2>,
 <CF Field: northward_wind(time(1), pressure(23), latitude(160), longitude(320)) m s**-1>,
 <CF Field: relative_humidity(time(1), pressure(23), latitude(160), longitude(320)) %>,
 <CF Field: specific_humidity(time(1), pressure(23), latitude(160), longitude(320)) kg kg**-1>,
 <CF Field: vertical_air_velocity_expressed_as_tendency_of_pressure(time(1), pressure(23), lati
tude(160), longitude(320)) Pa s**-1>]
```

In [5]:

```
# Select the air temperature
temp=f.select('air_temperature')[0]
```

In [6]:

```
# Select by index
temp=f[2]
```

In [7]:

```
# Select by long_name
vorticity=f.select('long_name:Potential vorticity')[0]
```

In [8]:

```
# See a longer list of field contents
print vorticity
```

```
long_name:Potential vorticity field summary
-------------------------------------------
Data           : long_name:Potential vorticity(time(1), pressure(23), latitude(160), longitude(
320)) K m**2 kg**-1 s**-1
Axes           : time(1) = [1964-01-21T00:00:00Z]
               : pressure(23) = [1000.0, ..., 1.0] mbar
               : latitude(160) = [89.1415176392, ..., -89.1415176392] degrees_north
               : longitude(320) = [0.0, ..., 358.875] degrees_east
```

In [9]:

```
# Change the standard_name of the field
vorticity.standard_name='ertel_potential_vorticity'

# View properties
vorticity.properties
```

Out[9]:

```
<bound method Field.properties of <CF Field: ertel_potential_vorticity(time(1), pressure(23), l
atitude(160), longitude(320)) K m**2 kg**-1 s**-1>>
```

In [10]:

```
# Write the modified field to a netCDF file
cf.write(vorticity, 'newfile.nc')
```

Contour plots

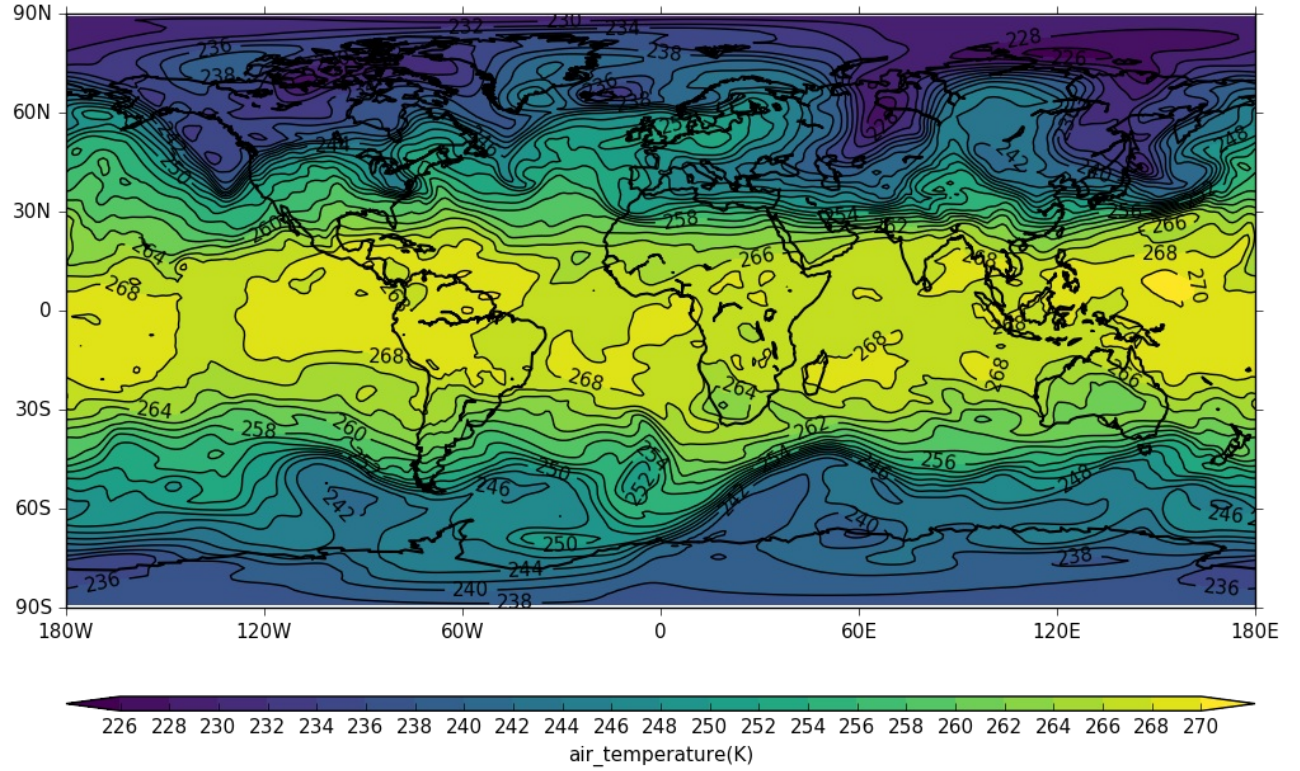
In [11]:

```
# Use subspace to select the temperature at 500mb
t_500=temp.subspace(Z=500)
print t_500
```

```
air_temperature field summary
-----------------------------
Data           : air_temperature(time(1), pressure(1), latitude(160), longitude(320)) K
Axes           : time(1) = [1964-01-21T00:00:00Z]
               : pressure(1) = [500.0] mbar
               : latitude(160) = [89.1415176392, ..., -89.1415176392] degrees_north
               : longitude(320) = [0.0, ..., 358.875] degrees_east
```

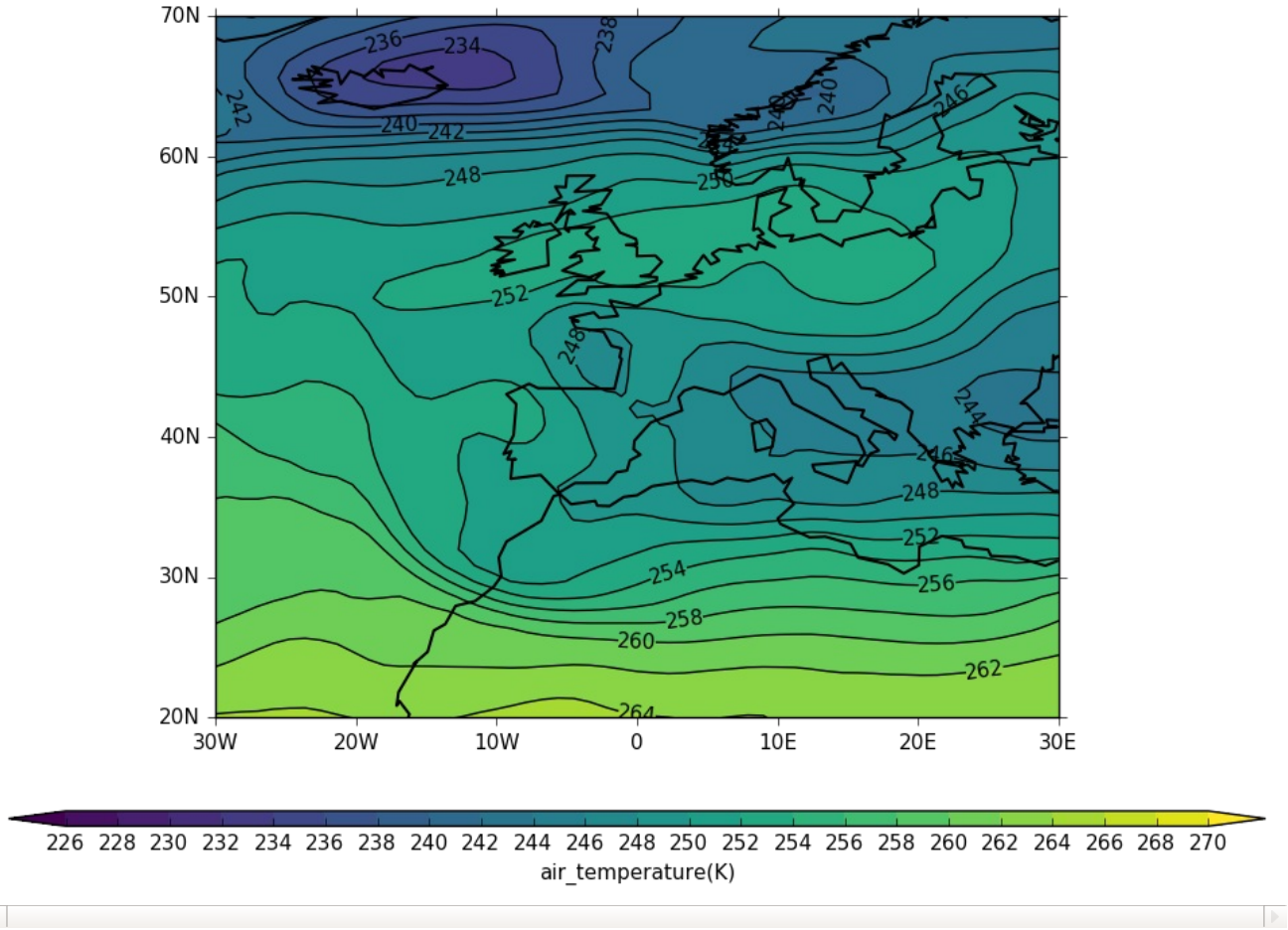
In [12]:

```
#Make a contour plot of the data
cfp.con(t_500)
```



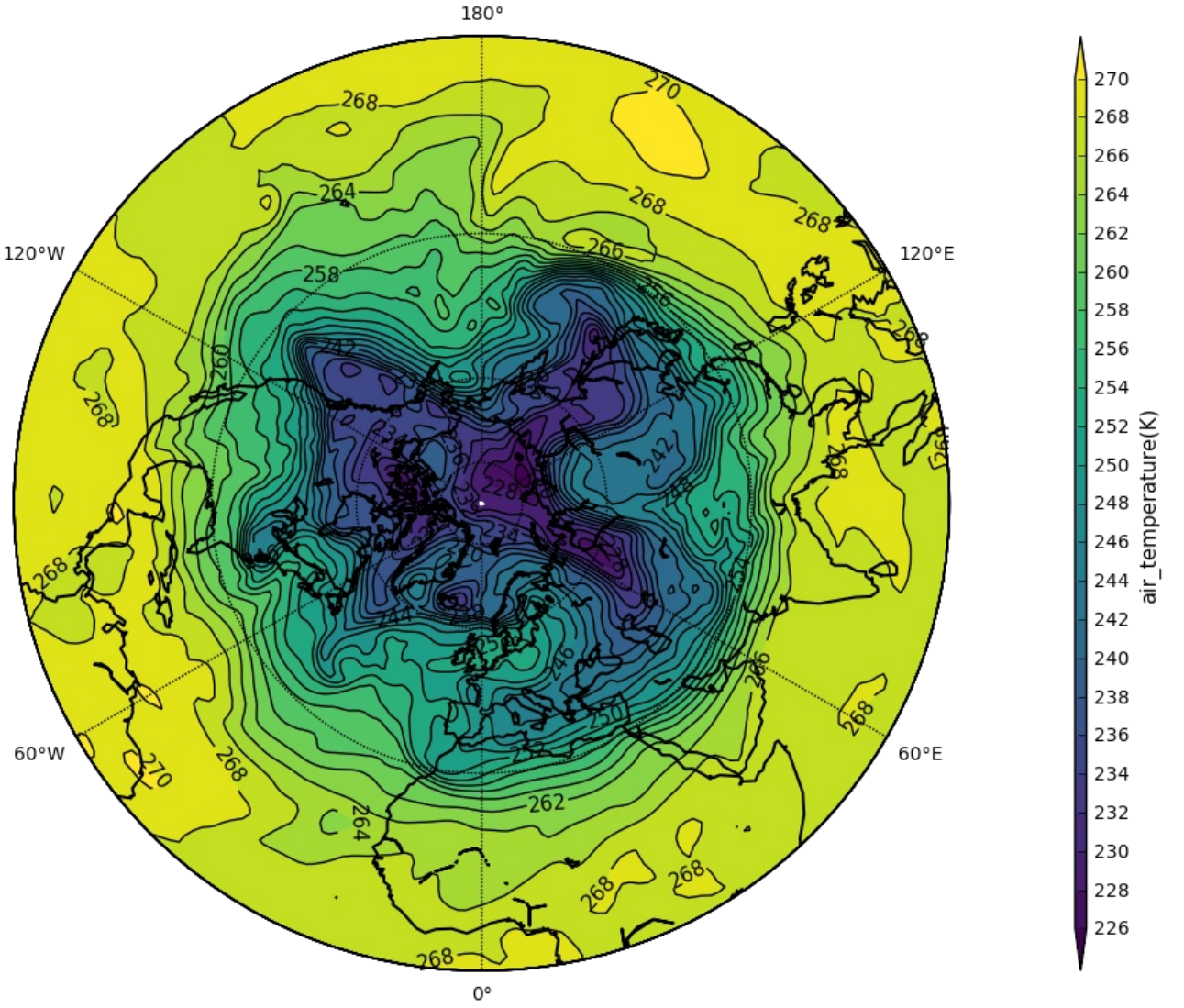
In [13]:

```
# Use mapset to select Europe and make a new contour plot
cfp.mapset(lonmin=-30, lonmax=30, latmin=20, latmax=70)
cfp.con(t_500)
```



In [14]:

```
# Make a Northern Hemiphere polar stereographic plot
cfp.mapset(proj='npstere')
cfp.con(t_500)
cfp.mapset()
```



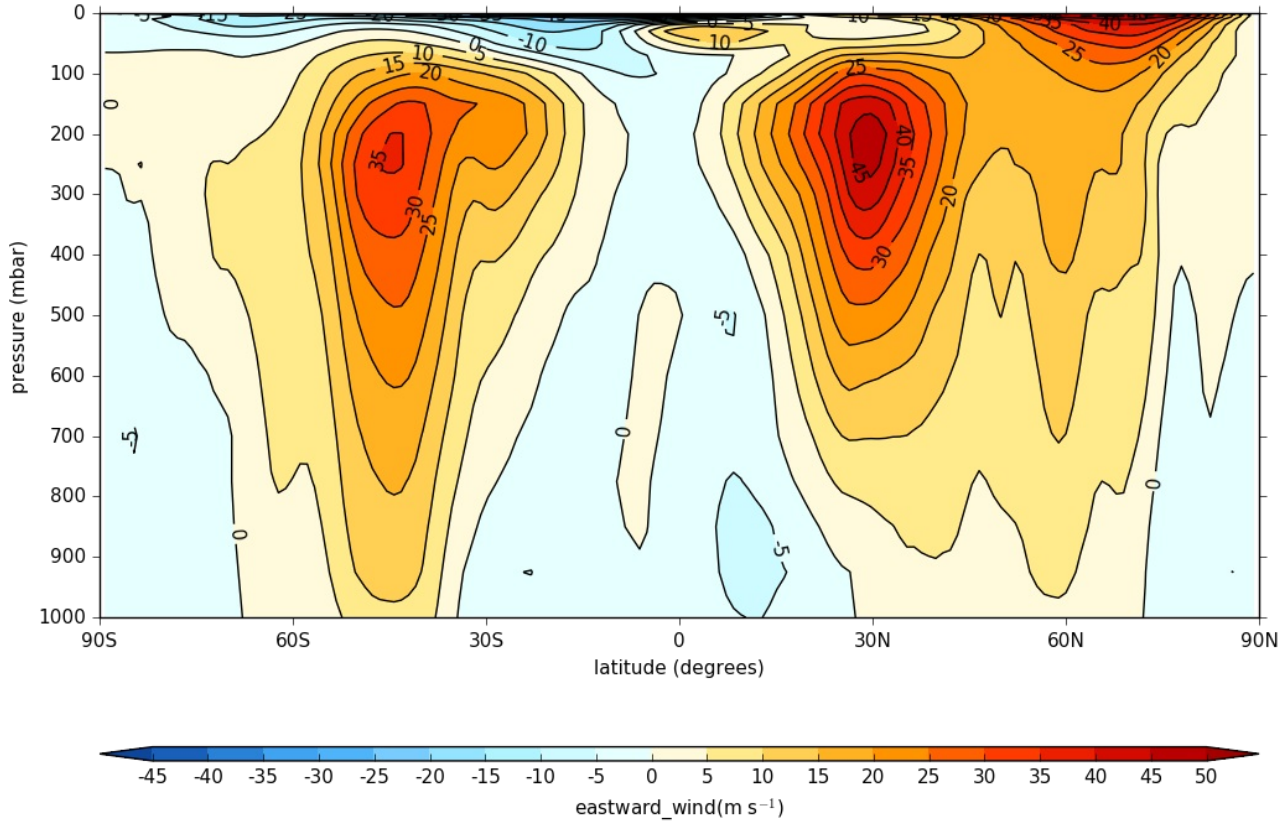
In [15]:

```
# Select the zonal wind and make a zonal mean of this using the collapse function in cf-python
u=f.select('eastward_wind')[0]
u_mean=u.collapse('mean', 'longitude')
print u_mean
```

```
eastward_wind field summary
---------------------------
Data           : eastward_wind(time(1), pressure(23), latitude(160), longitude(1)) m s**-1
Cell methods   : longitude: mean
Axes           : time(1) = [1964-01-21T00:00:00Z]
               : pressure(23) = [1000.0, ..., 1.0] mbar
               : latitude(160) = [89.1415176392, ..., -89.1415176392] degrees_north
               : longitude(1) = [179.4375] degrees_east
```

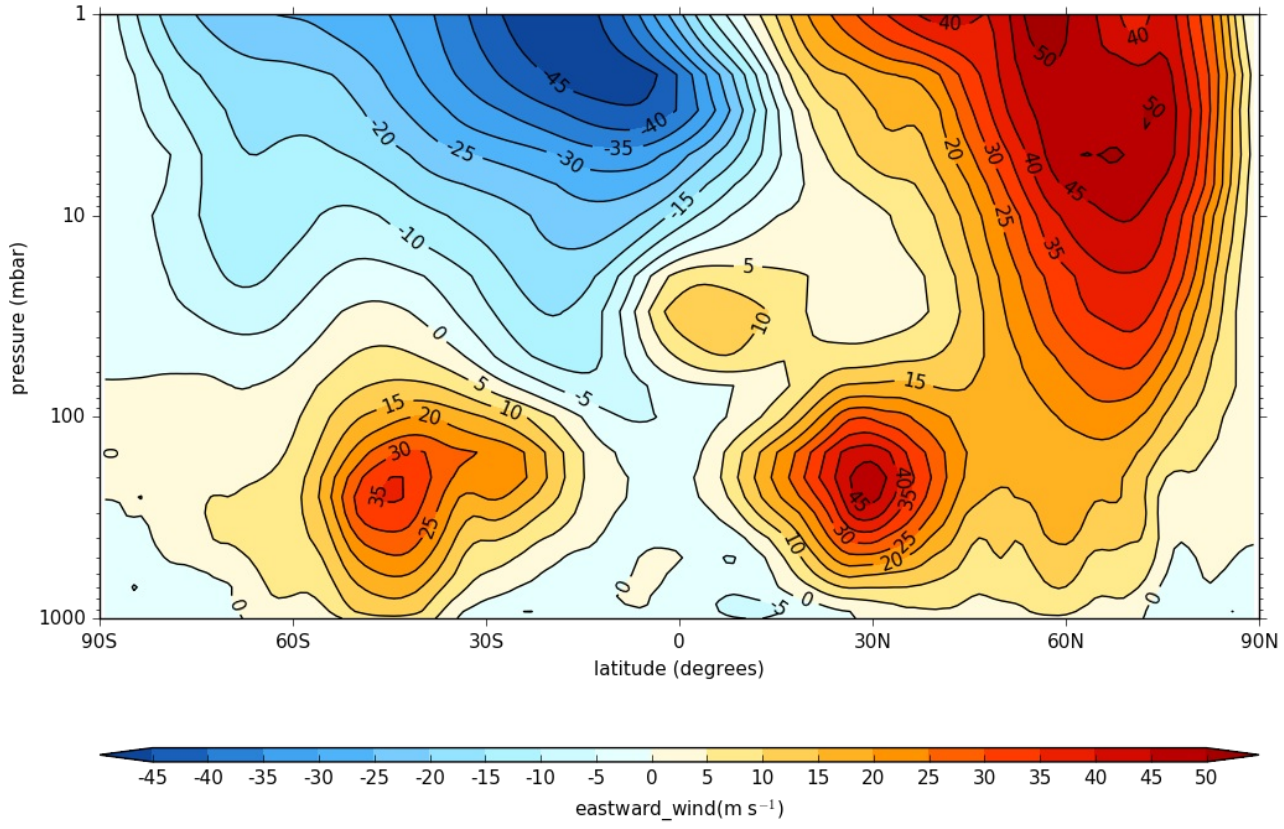
In [16]:

```
# Make a zonal mean zonal wind plot
cfp.con(u_mean)
```



In [17]:

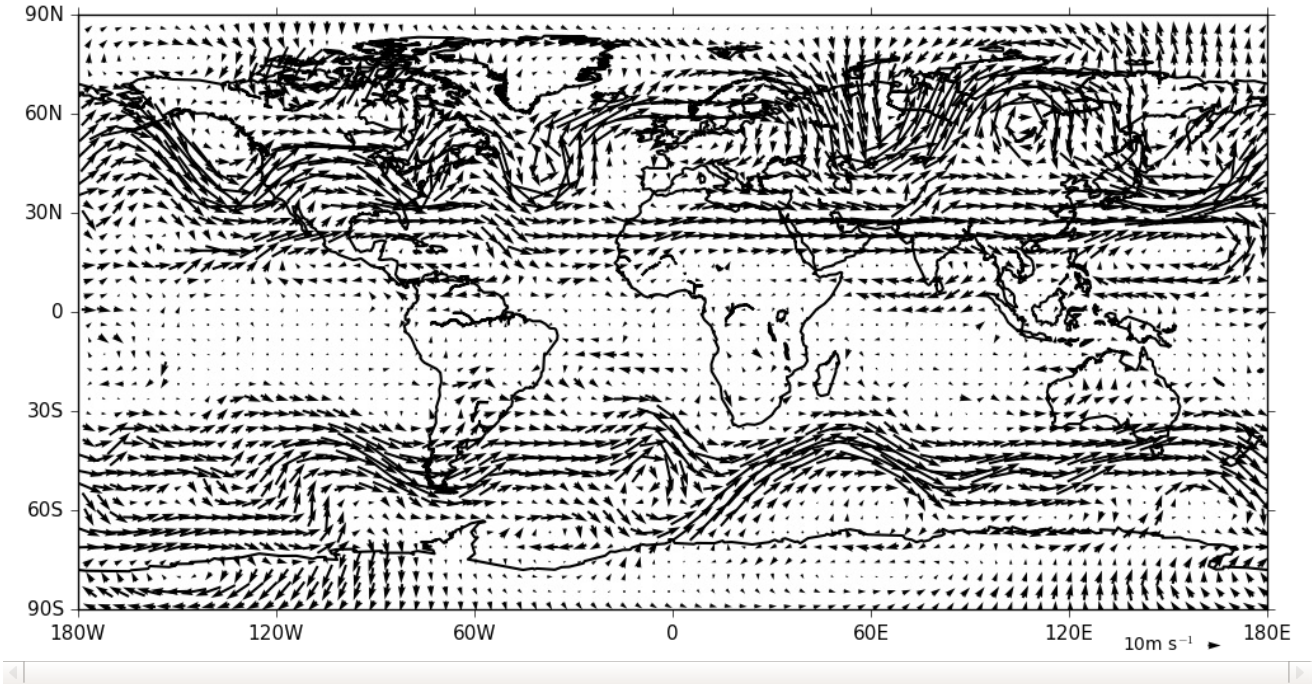
```
# Make a log y-axis plot of the zonal mean zonal wind
cfp.con(u_mean, ylog=True)
```



Vector plots

In [18]:

```
# Select u and v wind components at 500mb and make a vector plot
# We use a stride of 4 in plotting the vectors as the points are close together
u=f[7].subspace(pressure=500)
v=f[9].subspace(pressure=500)
cfp.vect(u=u, v=v, key_length=10, scale=100, stride=4)
```



Line plots

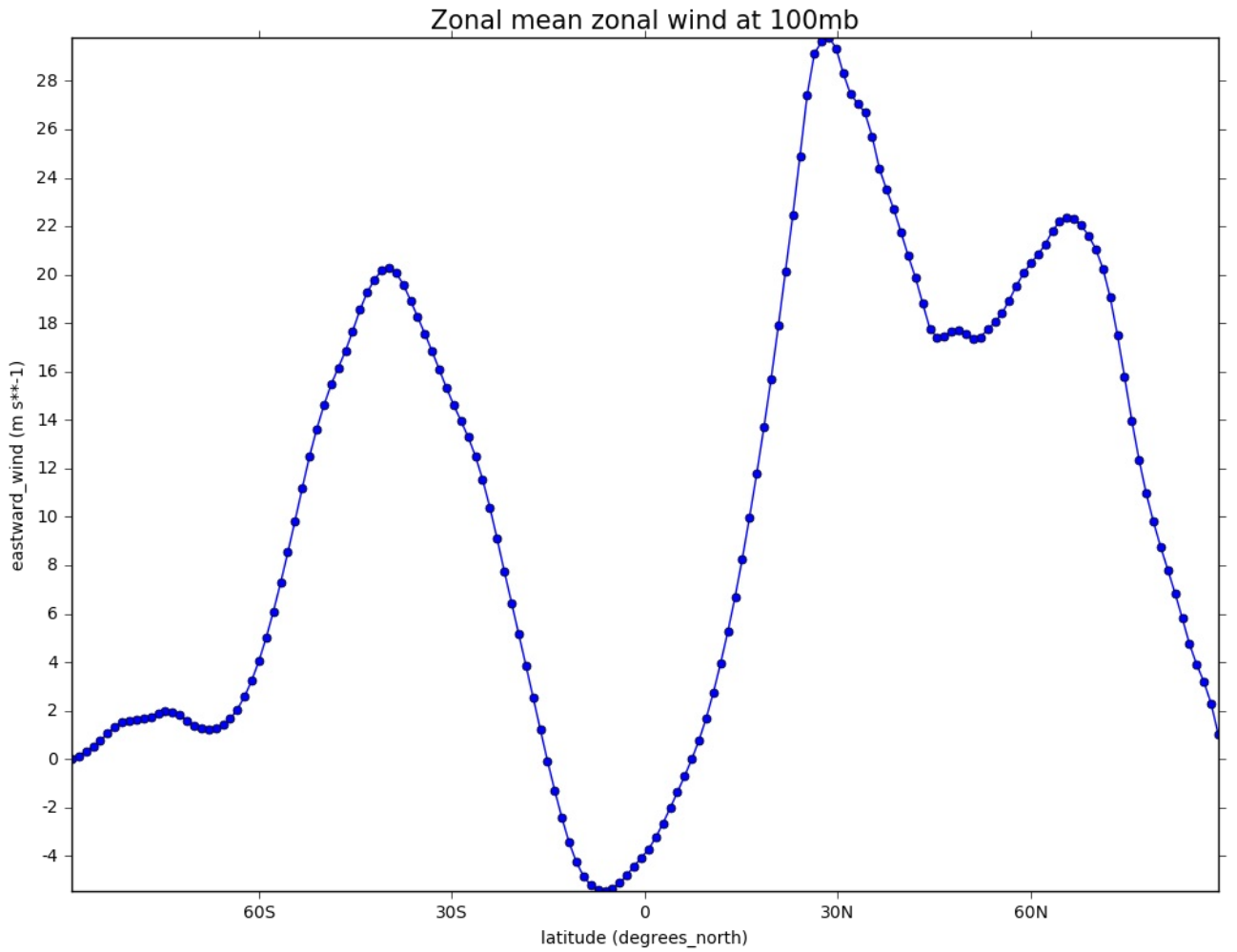
In [19]:

```
# Select the zonal mean zonal wind at 100mb
u=f[7]
u_mean=u.collapse('X: mean')
u_mean_100=u_mean.subspace(pressure=100)
print u_mean_100
```

```
eastward_wind field summary
---------------------------
Data           : eastward_wind(time(1), pressure(1), latitude(160), longitude(1)) m s**-1
Cell methods   : longitude: mean
Axes           : time(1) = [1964-01-21T00:00:00Z]
               : pressure(1) = [100.0] mbar
               : latitude(160) = [89.1415176392, ..., -89.1415176392] degrees_north
               : longitude(1) = [179.4375] degrees_east
```

In [20]:

```
cfp.lineplot(u_mean_100, marker='o', color='blue', title='Zonal mean zonal wind at 100mb')
```



Regridding

Regrid some temperature longitude-latiude data to another grid and make a plot of the difference between the two datsets.

In [21]:

```
# Read in data on two different grids
temp_era40=cf.read_field('ncas_data/data2.nc')
temp_era_in=cf.read_field('ncas_data/data3.nc')

print temp_era40, temp_era_in
```

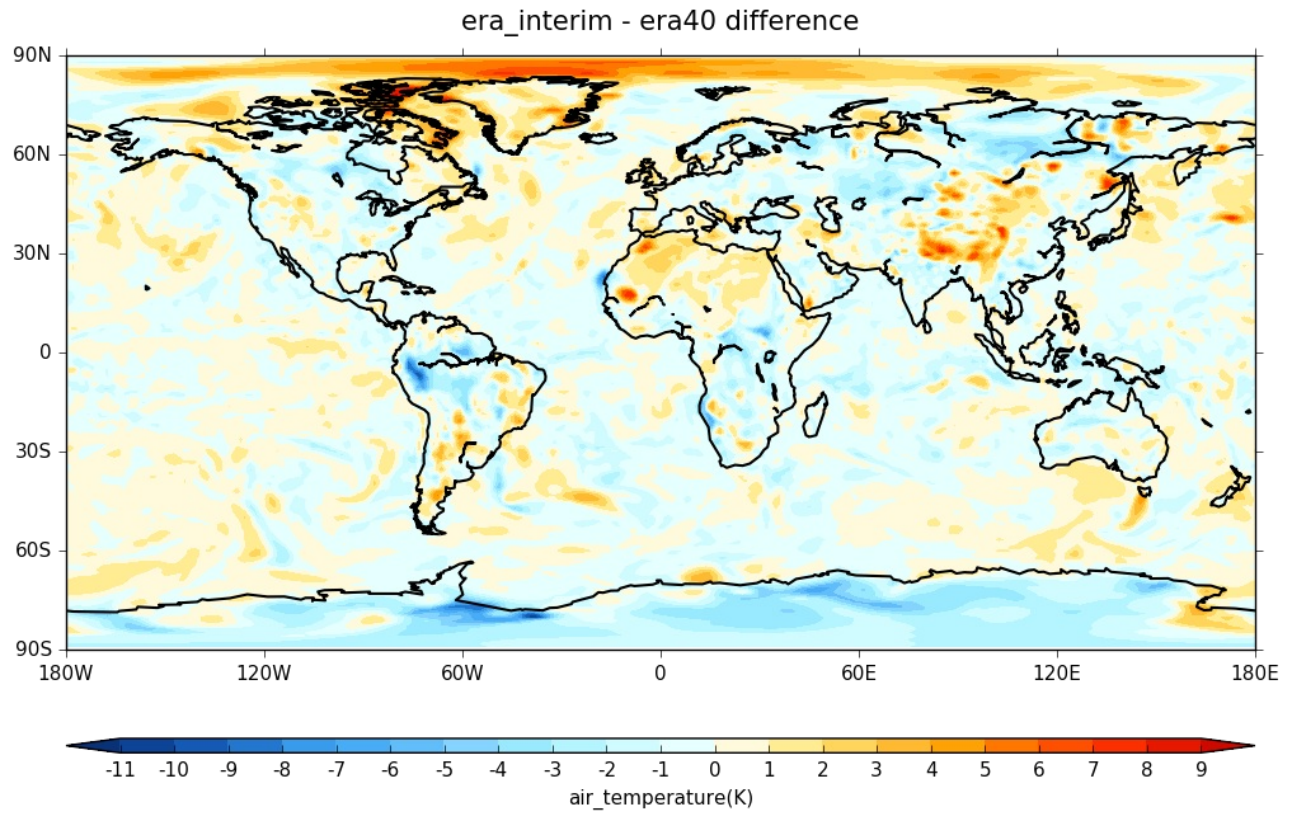
```
air_temperature field summary
-----------------------------
Data           : air_temperature(long_name:t(1), long_name:p(1), latitude(160), longitude(320))
 K
Axes           : long_name:t(1) = [1981-01-21T00:00:00Z]
               : long_name:p(1) = [1000.0] mbar
               : latitude(160) = [89.1415176392, ..., -89.1415176392] degrees_north
               : longitude(320) = [0.0, ..., 358.875] degrees_east
 air_temperature field summary
-----------------------------
Data           : air_temperature(long_name:t(1), long_name:p(1), long_name:latitude(256), long_
name:longitude(512)) K
Axes           : long_name:t(1) = [1981-01-21T00:00:00Z]
               : long_name:p(1) = [1000.0] mbar
               : long_name:latitude(256) = [89.4629440308, ..., -89.4629440308] degrees_north
               : long_name:longitude(512) = [0.0, ..., 359.296875] degrees_east
```

In [22]:

```
# Perform the regridding
temp_regrid = temp_era_in.regrids(temp_era40, method='bilinear')
```

In [23]:

```
# Make a contour plot of the difference between the two datasets
cfp.con(temp_regrid-temp_era40, lines=False, title='era_interim - era40 difference')
```



In [ ]:

```
```