

In [1]:

```
#
# File:
#   NUG_rectilinear_slice_PyNGL.py
#
# Synopsis:
#   Illustrates slicing the data and creating a contour plot
#
# Categories:
#   contour plots
#   slices
#
# Author:
#   Karin Meier-Fleischer
#
# Date of initial publication:
#   June 2015
#
# Description:
#   This example shows how to slice the data at a particular
#   latitude and create contours of pressure versus longitude
#
# Effects illustrated:
#   o Drawing filled contours
#   o Reversing the Y axis
#   o Using transformation resources to log the Y axis
#
# Output:
#   One visualization is produced.
#
# Notes: The data for this example can be downloaded from
#   http://www.ncl.ucar.edu/Document/Manuals/NCL\_User\_Guide/Data/
#
"""
NCL User Guide Python Example:   NUG_rectilinear_slice_PyNGL.py
- slice filled contour plot
- colorbar
- log axis

2015-06-04   kmf
"""
from __future__ import print_function
import numpy as np
import sys, os
import Nio, Ngl
```

In [3]:

```
def nice_lon_labels(lons):
    lonstrs = []
    for l in lons:
        if l < 0:
            lonstrs.append("{}~S~o~N~W".format(np.fabs(l)))
        elif l > 0:
            lonstrs.append("{}~S~o~N~E".format(l))
        else:
            lonstrs.append("EQ")
    return lonstrs
```

In [4]:

```
#-- define variables
diri  = "./"                                #-- data directory
fname = "rectilinear_grid_3D.nc"            #-- data file name
ffile = os.path.join(diri, fname)

#---Test if file exists
if(not os.path.exists(ffile)):
    print("You do not have the necessary file ({} to run this example.".format(ffile))
    print("You can get the files from the NCL website at:")
    print("http://www.ncl.ucar.edu/Document/Manuals/NCL\_User\_Guide/Data/")
    sys.exit()
```

In [5]:

```
## open file and read variables
f      = Nio.open_file(ffile, "r")      ## open data file
t      = f.variables["t"]              ## get whole "t" variable
t26    = t[0,:,26,:]                  ## variable at lat index 26
lev     = f.variables["lev"][:]*0.01   ## all levels, convert to hPa
lat     = f.variables["lat"][:]        ## reverse latitudes
lon     = f.variables["lon"][:]        ## all longitudes

t26,lon = Ngl.add_cyclic(t26,lon)

strlat26 = lat[26]                    ## retrieve data of lat index 26

## get the minimum and maximum of the data
minval = int(np.amin(t[:]))           ## minimum value
maxval = int(np.amax(t[:]))           ## maximum value
inc     = 5                           ## contour level spacing

## values on which to place tickmarks on X and Y axis
lons = np.arange(-180,240,60)
levs = [1000,700,500,400,300,200,150,100,70,50,30,10]
```

In [6]:

```
wks_type      = "png"
wks_name      = "NUG_rectilinear_slice_PyNGL"
wks           = Ngl.open_wks(wks_type,wks_name)

## set resources
res           = Ngl.Resources

res.tiMainString      = "{} ({} ) at lat {:.2f} degrees".format(t.long_name,
                                                                t.units,
                                                                strlat26)

res.cnLevelSelectionMode = "ManualLevels" ## select manual levels
res.cnMinLevelValF      = minval          ## minimum contour value
res.cnMaxLevelValF      = maxval          ## maximum contour value
res.cnLevelSpacingF     = inc             ## contour increment

res.cnFillOn           = True             ## turn on contour fill.
res.cnLineLabelsOn     = False            ## turn off line labels.
res.cnInfoLabelOn      = False            ## turn off info label.
res.cnFillPalette       = "BlueWhiteOrangeRed" ## set color map.
res.pmLabelBarOrthogonalPosF = -0.03      ## move labelbar close to plot

res.sfXArray           = lon              ## scalar field x
res.sfYArray           = lev              ## scalar field y

res.trYReverse          = True             ## reverse the Y axis
res.nglyAxisType        = "LogAxis"        ## y axis log

res.tiYAxisString       = "{} (hPa)".format(f.variables["lev"].long_name)

res.nglPointTickmarksOutward = True        ## point tickmarks out

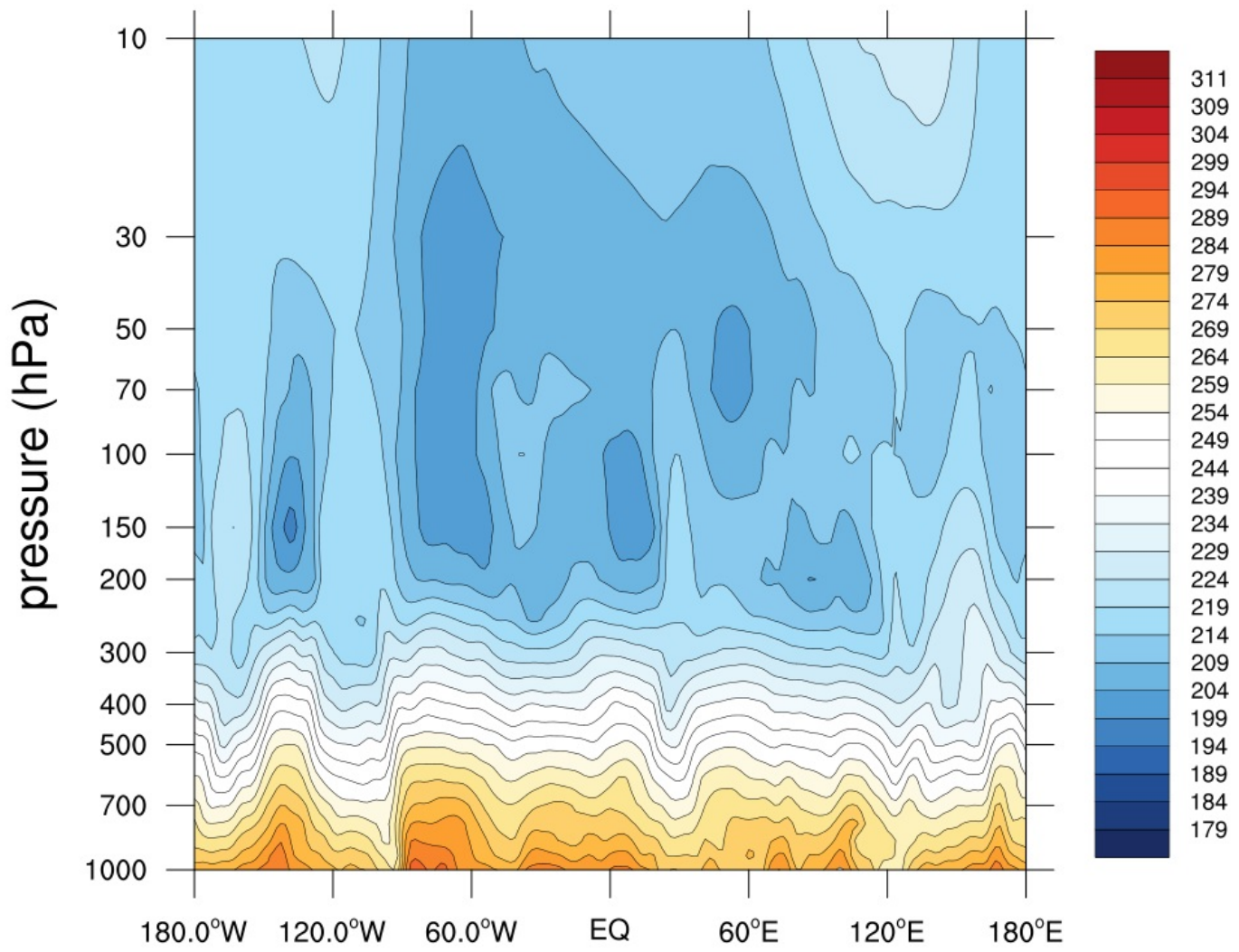
res.tmYLMode            = "Explicit"       ## set y axis tickmark labels
res.tmXBMode           = "Explicit"       ## set x axis tickmark labels
res.tmYLValues          = levs
res.tmXBValues          = lons
res.tmYLLabels          = [str(x) for x in levs]
res.tmXBLabels          = nice_lon_labels(lons)

res.tmXBLabelFontHeightF = 0.015           ## - make font smaller
res.tmYLLabelFontHeightF = 0.015

map = Ngl.contour(wks,t26,res)            ## draw contours

## end
Ngl.end()
```

temperature (K) at lat 40.10 degrees



In [ ]: