

# Exercises and Tasks

NCL Workshop

## Contents:

- Simple examples for a better understanding
- User exercises and tasks to get more practice

# Lecture Parts

1. XY-Plots
2. Maps and Contours
3. Vectors
4. Slices and Overlays
5. Panel Plots
6. Projections, Map Resolutions, Shapefiles
7. Captions and Annotations
8. Animations

# Base NCL script

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"  
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
```

```
begin
```

- *create data or open data file*
- *set variable/variables*
- *set plotting resources*
- *create plot*

```
end
```

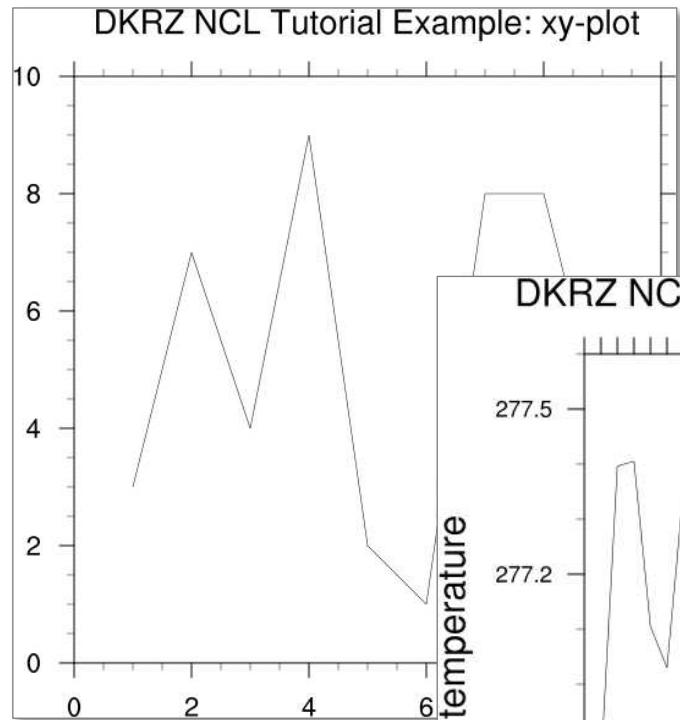
This base script could be copied from [/work/kv0653/NCL\\_Tutorial/scripts/basescript.ncl](/work/kv0653/NCL_Tutorial/scripts/basescript.ncl)

# Part I

## XY-Plots

### Exercises and Tasks

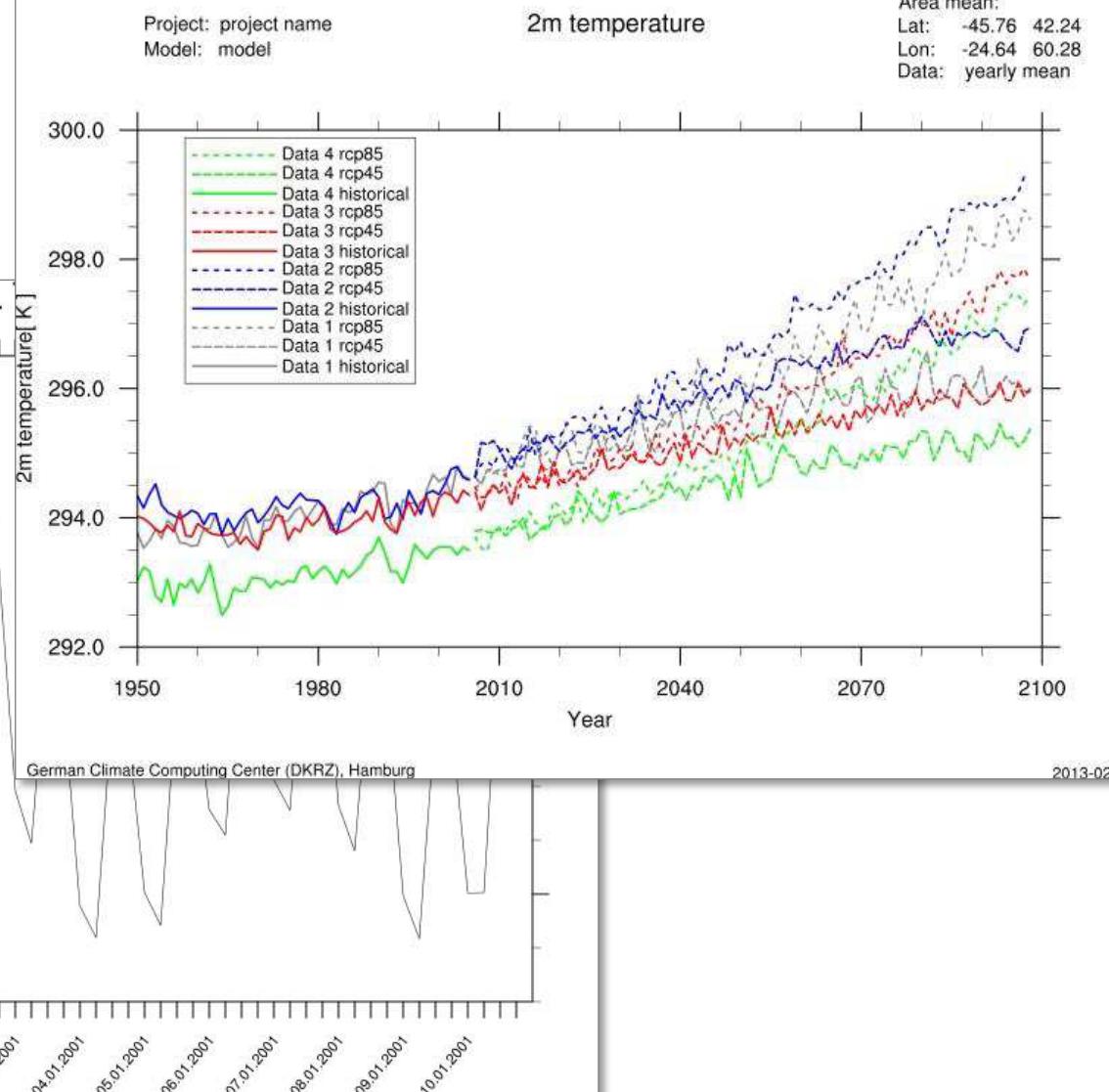
# XY-Plots



DKRZ NCL

surface temperature

## DKRZ NCL Tutorial Example: multiple timeseries



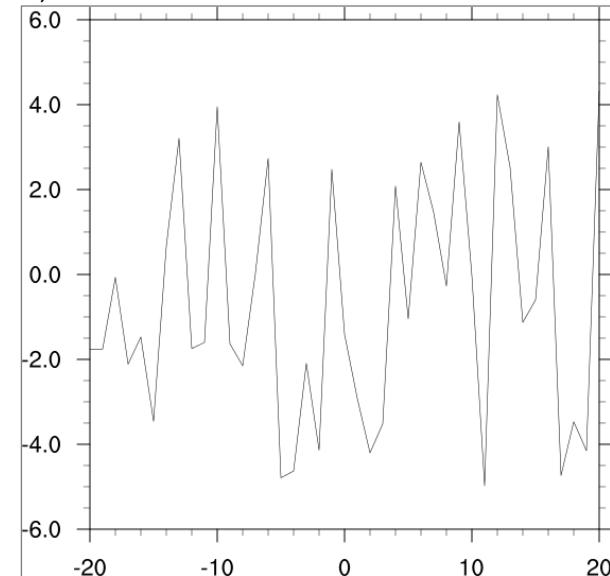
# Structure of a simple XY-plot script

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
    x = ispan(-20,20,1)
    y = random_uniform(-5,5,dimsizes(x))

    wks = gsn_open_wks("png", "part_I_simple_xy")
    plot = gsn_csm_xy(wks, x, y, False)
end
```

Plot formats: X11, PNG, PDF, PS, EPS, EPSI, NCGM



# XY-plot script reading data from ASCII file

```

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  data = asciiread("ASCII_data.asc", (/12,2/), "integer")
    ;-- read ASCII data to 2D array variable data

  x = data(:,0)          ;-- get all data from column 1
  y = data(:,1)          ;-- get all data from column 2

  wks = gsn_open_wks("png", "part_I_read_ASCII_xy")

  res           = True      ;-- create plot resource object
  res@tiMainString = "counts per day" ;-- draw a title on top of plot

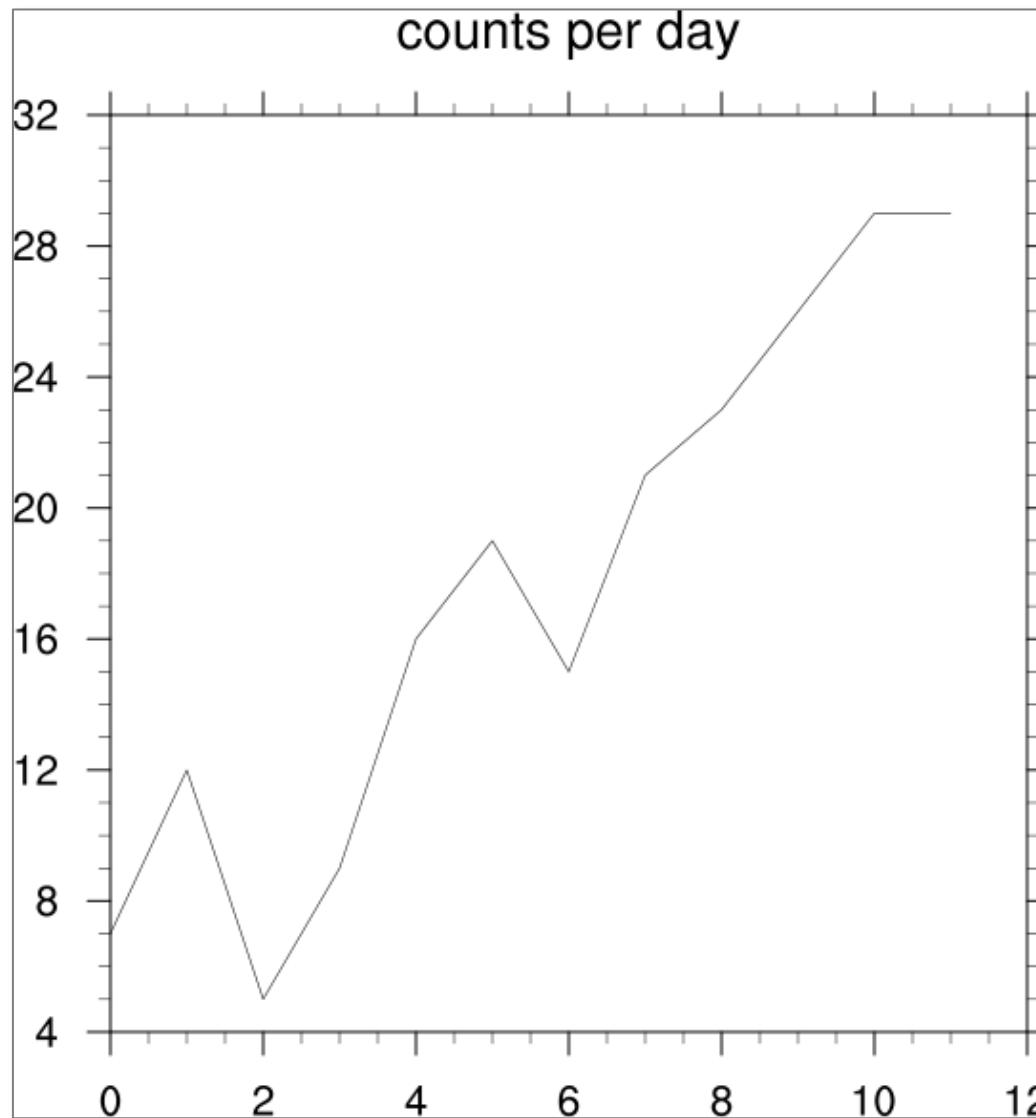
  plot = gsn_csm_xy(wks, x, y, res)

end

```

0	7
1	12
2	5
3	9
4	16
5	19
6	15
7	21
8	23
9	26
10	29
11	29
12	31

# XY-plot script reading data from ASCII file



# XY-plot script using geographical data

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
    ;-- open data file
    f = addfile("$NCL_TUT/data/EH_OM_A1B_3Dvectors_t1-24.nc", "r")
    x = f->u&lat                                ;-- latitude values
    y = f->u(0,0,:,:{10})                         ;-- values at longitude 10 deg.

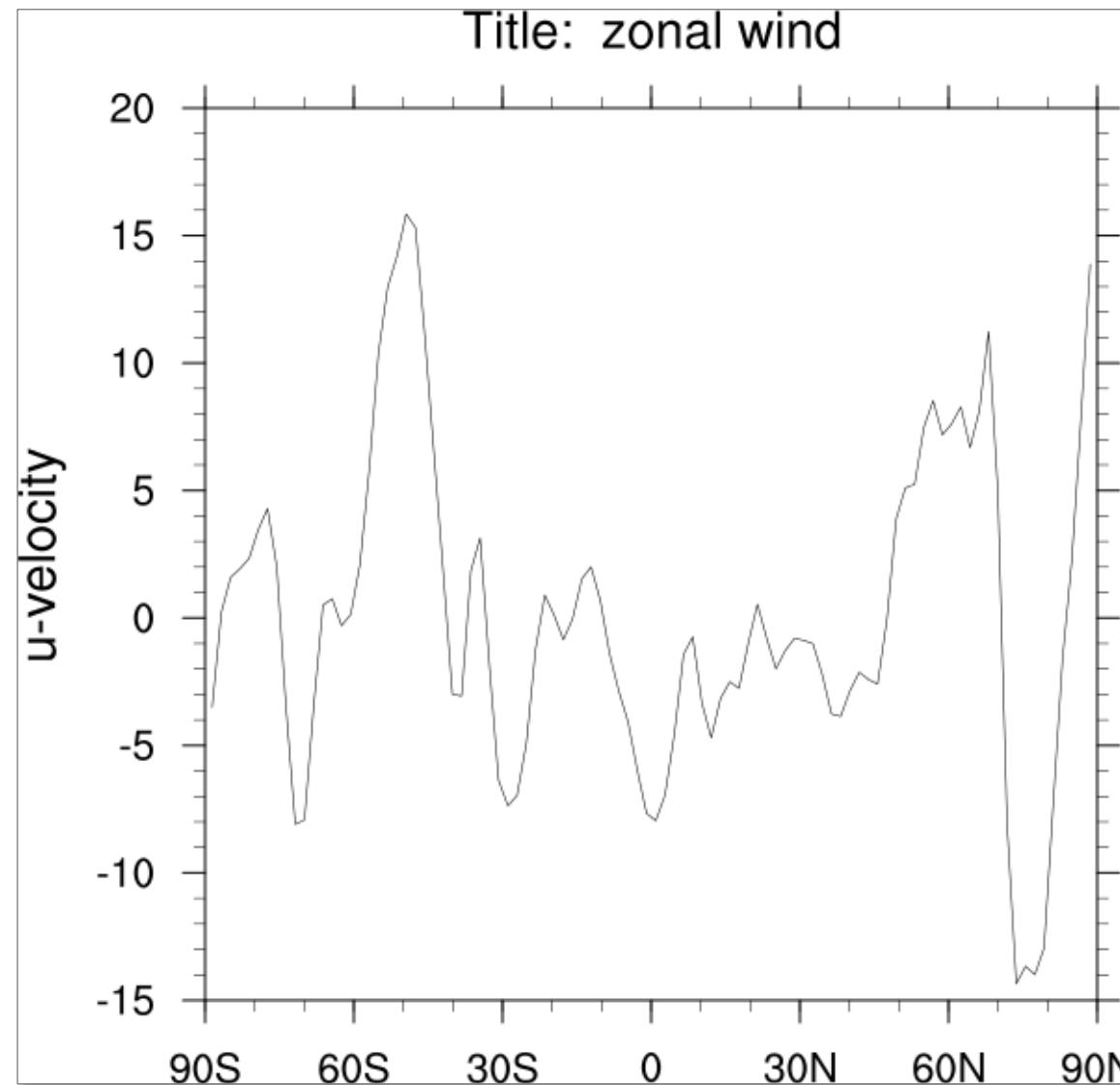
    wks = gsn_open_wks("png", "part_I_read_data_xy")

    res
    res@tiMainString = True                      ;-- create plot resource object
    res@tiMainString = "Title: zonal wind" ;-- draw a title

    plot = gsn_csm_xy(wks, x, y, res)
end
```

Dimension: time, level, lat, lon

# XY-plot script using geographical data



# XY-plot multiple curves

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin

f = addfile("$NCL_TUT/data/EH_OM_A1B_3Dvectors_t1-24.nc", "r")
u = f->u                                     ;-- set variable u

;-- assign multidimensional array
data      = new(/2,dimsizes(u&lat)/), float)
data(0,:) = u(0,0,:,:,{10})                   ;-- values at longitude 10 deg.
data(1,:) = u(0,0,:,:,{-105})                ;-- values at longitude -105 deg.

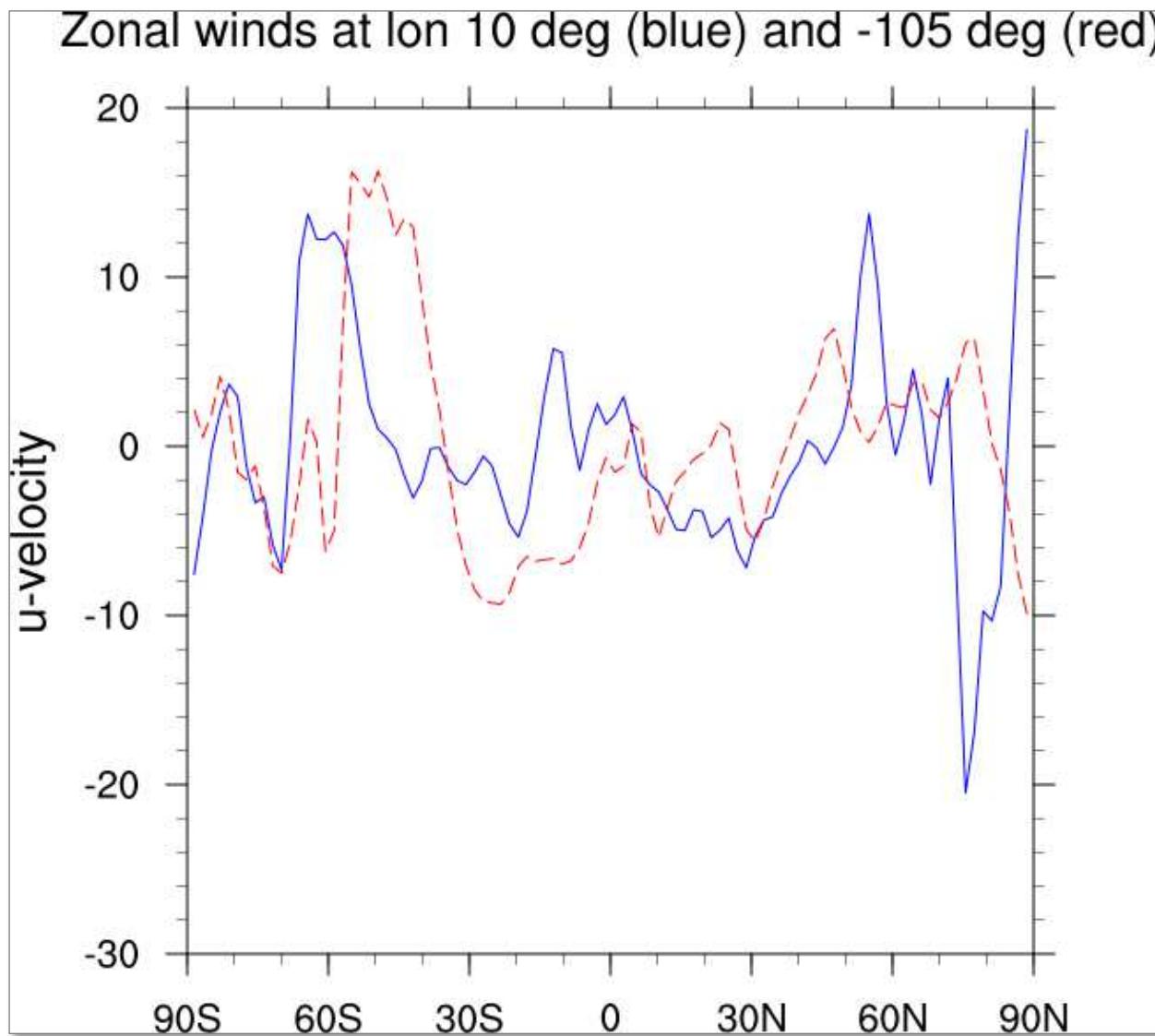
wks = gsn_open_wks("png", "part_I_multiple_curves_xy")

res          = True                         ;-- create plot resource object
res@xyLineColors = ("blue", "red")          ;-- line colors
res@xyLineThicknesses = (2.0,2.0)           ;-- set line thickness
res@tiMainString = "Zonal winds at lon 10 deg (blue) and -105 deg (red)"

plot = gsn_csm_xy(wks, u&lat, data, res) ;-- create the plot

end
```

# XY-plot multiple curves



# XY-plot change dash pattern

```

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f = addfile("$NCL_TUT/data/EH_OM_A1B_3Dvectors_t1-24.nc", "r")
  u = f->u                                     ;-- set variable u

  ;-- assign multidimensional array
  data      = new((/3,dimsizes(u&lat)/), float)
  data(0,:) = u(0,0,:,:,{10})                   ;-- values at longitude 10 deg.
  data(1,:) = u(0,0,:,:,{-105})                ;-- values at longitude -105 deg.
  data(2,:) = u(0,0,:,:,{-120})                ;-- values at longitude -120 deg.

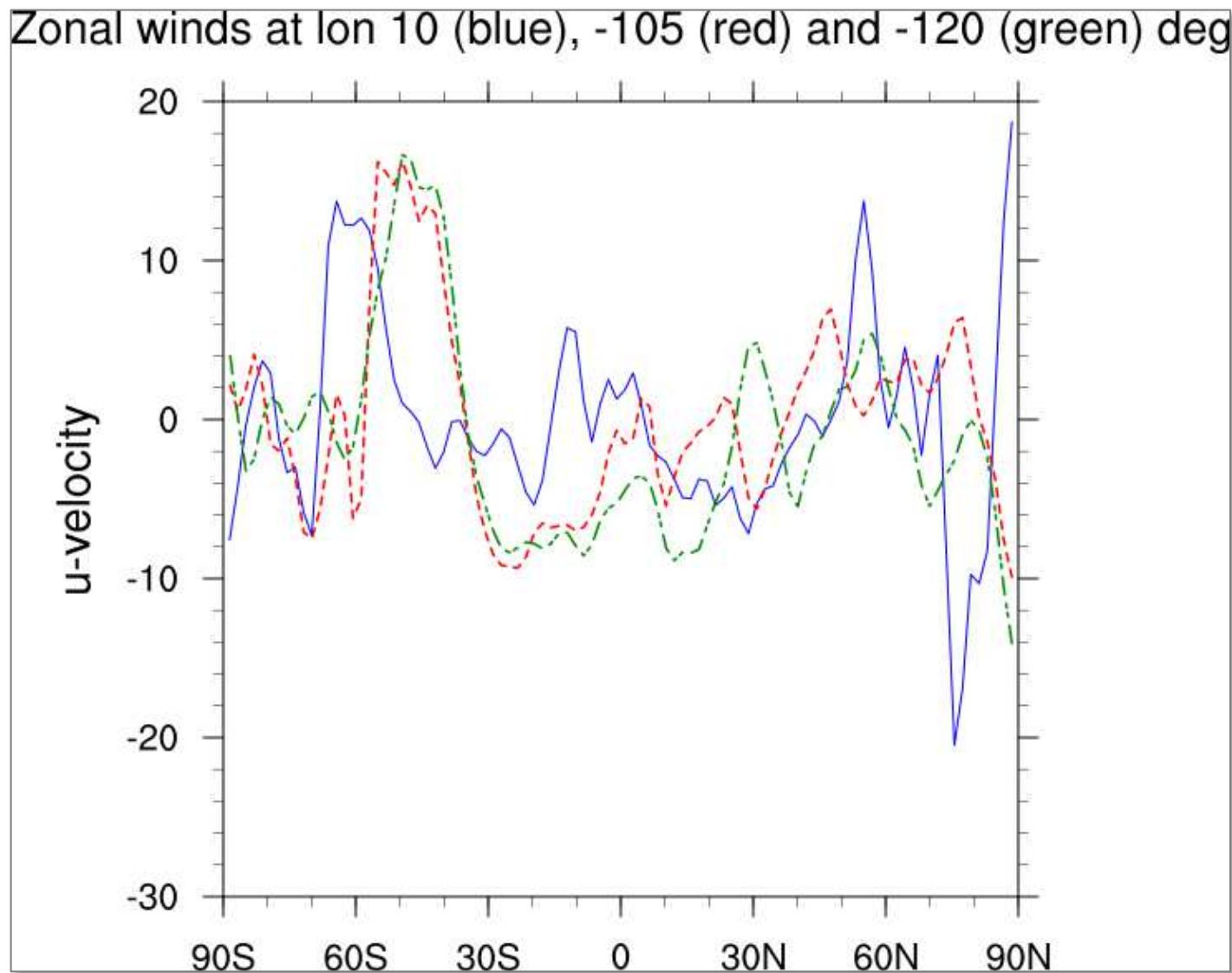
  wks = gsn_open_wks("png", "part_I_dash_pattern_xy")

  res           = True                         ;-- create plot resource object
  res@xyLineColors = (/ "blue", "red", "green4" /) ;-- line colors
  res@xyLineThicknesses = (/ 2.0, 3.0, 3.0 /) ;-- set line thickness
  res@xyDashPatterns = (/ 0, 11, 4 /)    ;-- 0=solid,11=dashed,4=line dotted
  res@tiMainString = "Zonal winds at lon 10(blue), -105(red) and \
                      -120(green) deg"
  plot = gsn_csm_xy(wks, u&lat, data, res) ;-- create the plot

end

```

# XY-plot change dash pattern



# XY-plot fill between lines

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin

f = addfile("$NCL_TUT/data/EH_OM_A1B_3Dvectors_t1-24.nc", "r")
u = f->u                                     ;-- set variable u

;-- assign multidimensional array
data      = new((/2,dimsizes(u&lat)/), float)
data(0,:) = u(0,0,:,:,{10})                  ;-- values at longitude 10 deg.
data(1,:) = u(0,0,:,:,{-105})                ;-- values at longitude -105 deg.

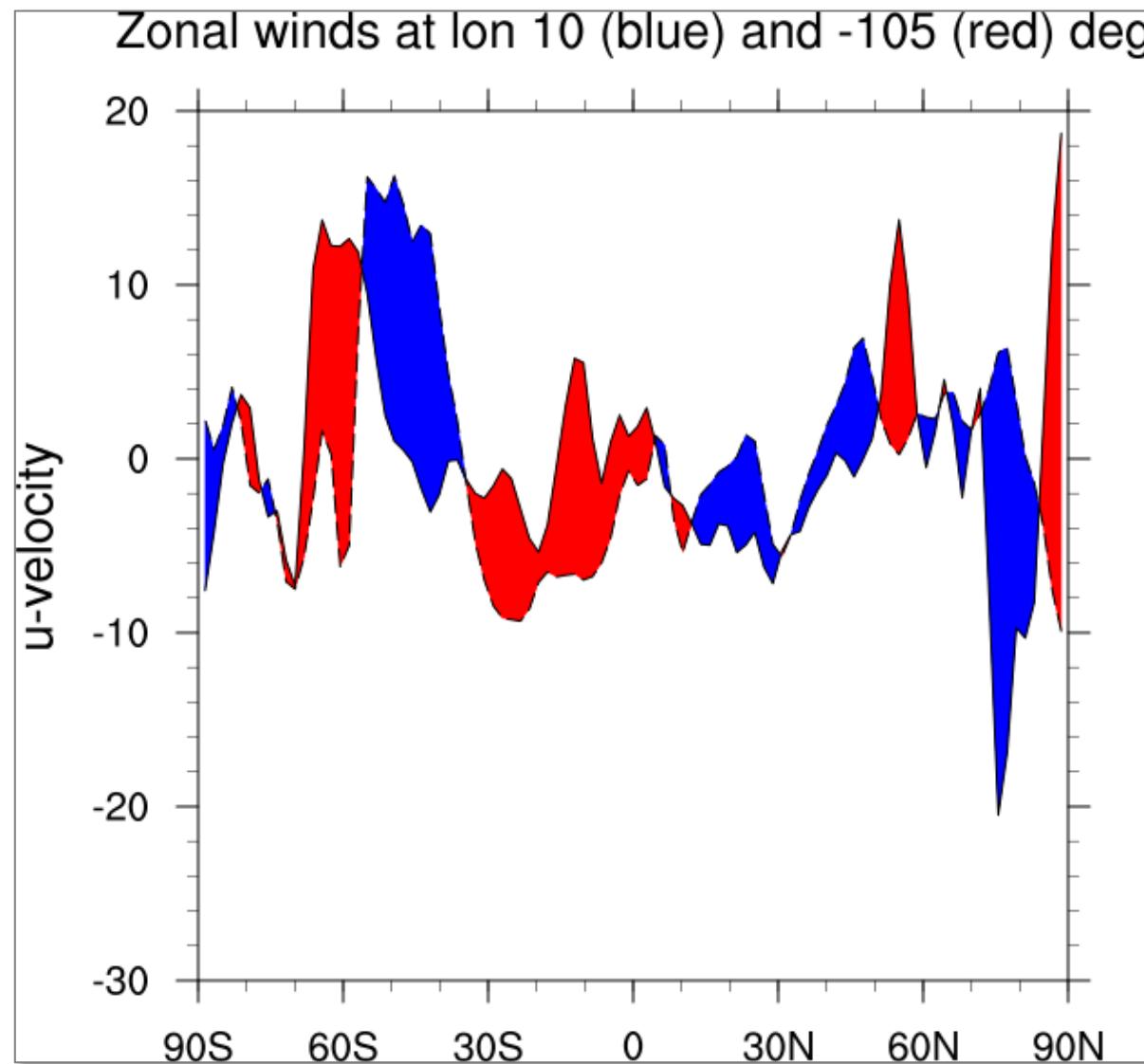
wks = gsn_open_wks("png", "part_I_fill_between_lines_xy")

res          = True                         ;-- create plot resource object
res@xyLineThicknessF = 2.0                   ;-- set line thickness
res@gsnXYAboveFillColors = "red"
res@gsnXYBelowFillColors = "blue"
res@tiMainString = "Zonal winds at lon 10 deg (blue) and -105 deg (red)"

plot = gsn_csm_xy(wks, u&lat, data, res) ;-- create the plot

end
```

# XY-plot fill between lines



# XY-plot with a reference line

```

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  x = fspan(1.,12.,24)                                ;-- generate x values
  y = random_uniform(-1.,1.,24)                        ;-- generate y values

  wks = gsn_open_wks("png","part_I_colored_above_below_xy")

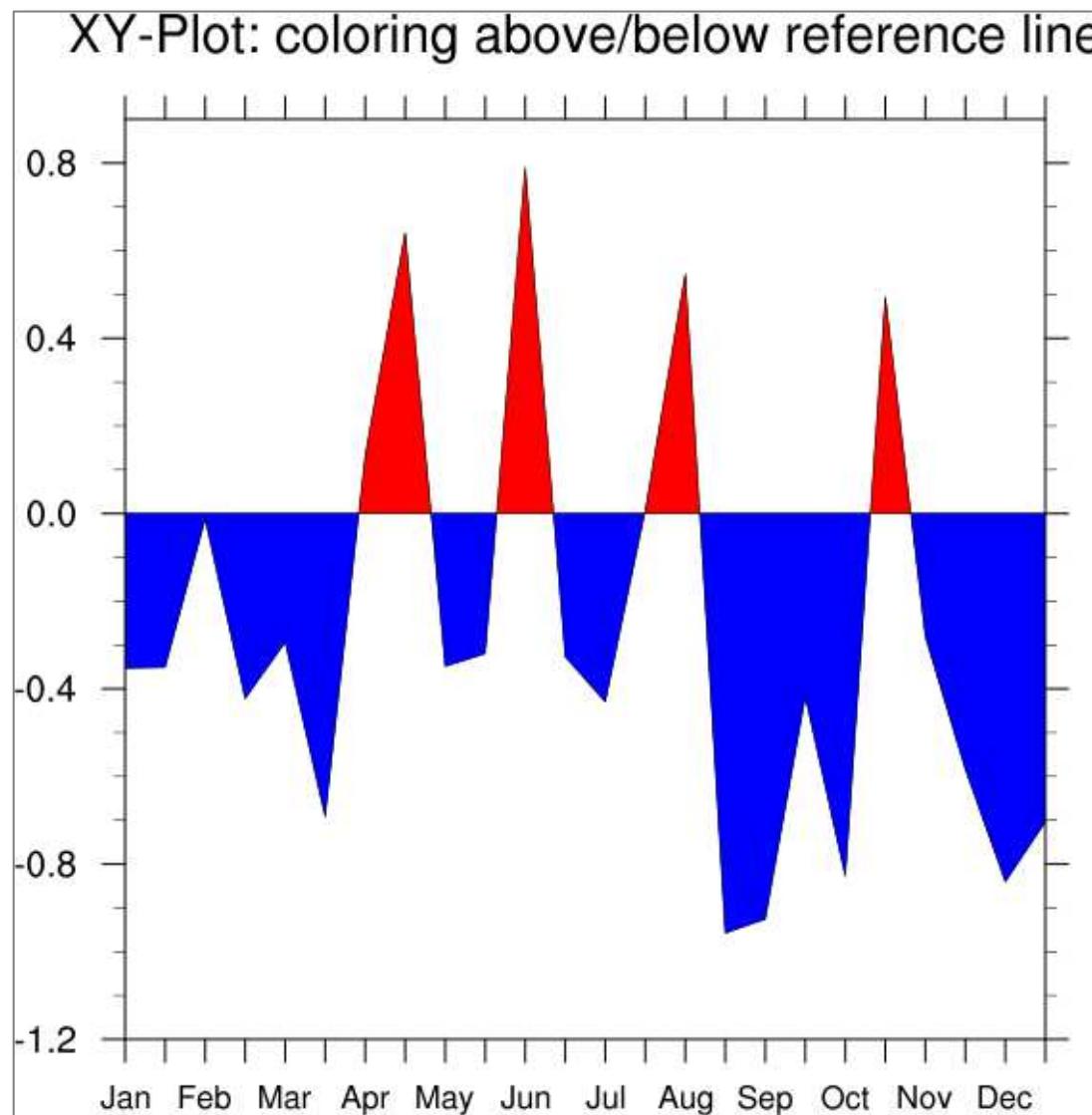
  res          = True        ;-- create plot resource object
  res@tmXBMode = "Explicit" ;-- use explicit labels
  res@tmXBValues = x         ;-- set x-axis values
  res@tmXBLABELFontHeightF = 0.015      ;-- x-axis label font size
  res@tmXBLabels = (/ "Jan", "", "Feb", "", "Mar", "", "Apr", "", \
                     "May", "", "Jun", "", "Jul", "", "Aug", "", \
                     "Sep", "", "Oct", "", "Nov", "", "Dec", "")/
  res@gsnYRefLine = 0.          ;-- reference line at 0.
  res@gsnAboveYRefLineColor = "red"    ;-- color red above reference line
  res@gsnBelowYRefLineColor = "blue"   ;-- color blue below reference line

  res@tiMainString = "XY-Plot: coloring above/below reference line"

  plot = gsn_csm_xy(wks,x,y,res)                  ;-- create the plot
end

```

# XY-plot with a reference line



# XY-plot auxiliary grid lines

```

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f = addfile("$NCL_TUT/data/EH_OM_A1B_3Dvectors_t1-24.nc", "r")
  u = f->u          -- set variable u

  data      = new((/2,dimsizes(u&lat)/), float)
  data(0,:) = u(0,0,:,:{10})                      -- values at longitude 10 deg.
  data(1,:) = u(0,0,:,:{-105})                     -- values at longitude -105 deg.

  wks = gsn_open_wks("png", "part_I_fill_between_lines_xy")

  res           = True                         -- create plot resource object
  res@xyLineColors = (/ "blue", "red" /)        -- set line colors
  res@xyLineThicknessF = 2.0                   -- set line thickness
  res@tiMainString = "Zonal winds at lon 10 deg (blue) and -105 deg (red)"

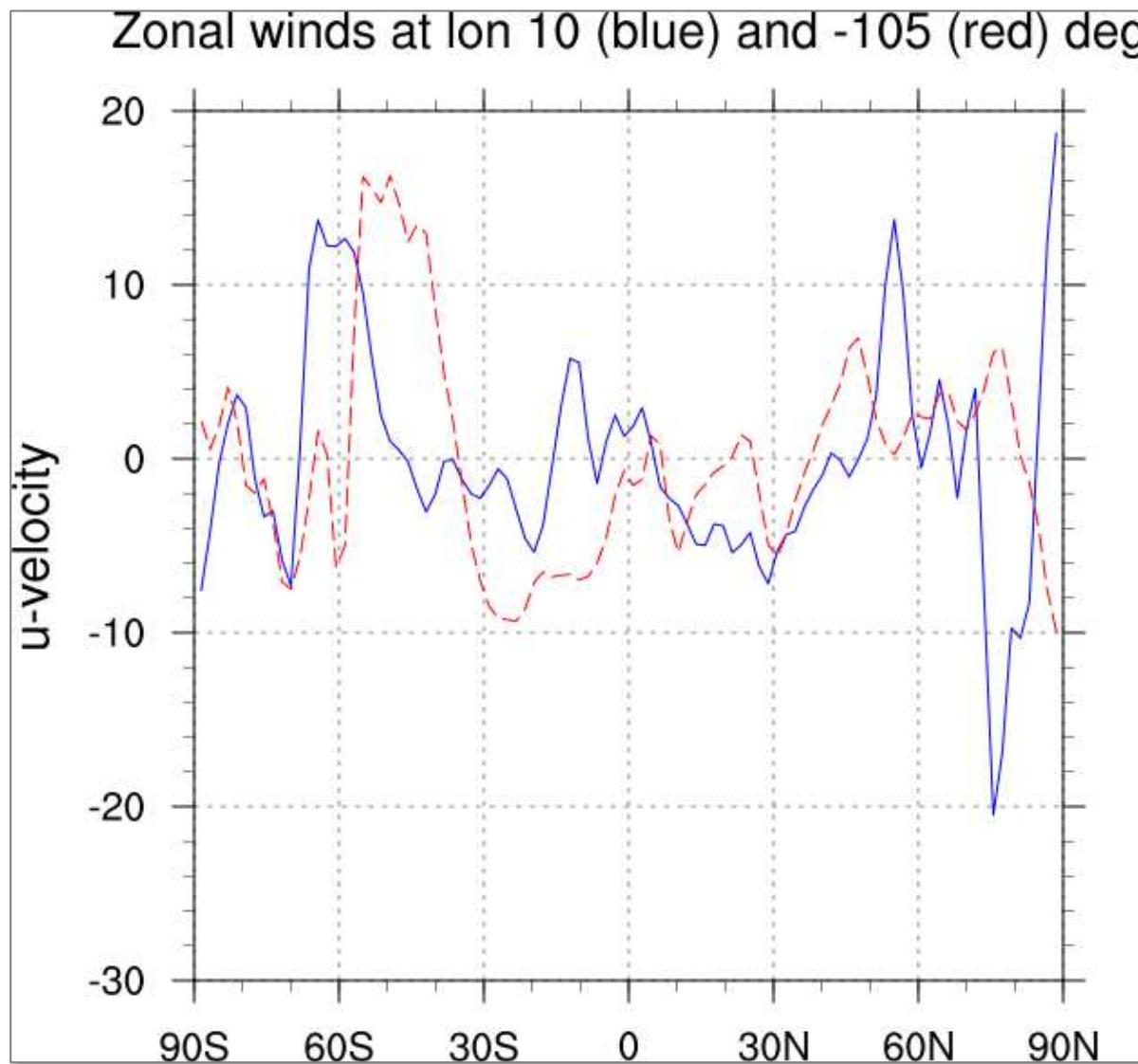
  res@tmXMajorGrid           = True            -- turn on x-grid
  res@tmXMajorGridThicknessF = 1.0             -- thinner the lines
  res@tmXMajorGridDashPattern = 2               -- use dashed lines
  res@tmYMajorGrid           = True            -- turn on y-grid
  res@tmYMajorGridThicknessF = 1.0             -- thinner the lines
  res@tmYMajorGridDashPattern = 2               -- use dashed lines

  plot = gsn_csm_xy(wks, u&lat, data, res)    -- create the plot

end

```

# XY-plot auxiliary grid lines



# XY-plot two different y-axis

```

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f = addfile("$NCL_TUT/data/EH_OM_A1B_3Dvectors_t1-24.nc", "r")
  t  = f->t(0,0,:,{10})           ;-- set variable t;      left y-axis
  slp = f->slp(0,:,10)           ;-- set variable slp;   right y-axis
  x  = f->lat                   ;-- x-axis values

  wks = gsn_open_wks("png", "part_I_two_different_y_axis_xy")

  resT          = True           ;-- create plot resource object
  resT@xyLineColor = "red"        ;-- set line colors
  resT@xyLineThicknessF = 3.0     ;-- set line thickness
  resT@xyDashPatterns = 0         ;-- solid line
  resT@tiYAxisString = t@long_name +" "+"[solid]" ;-- axis string

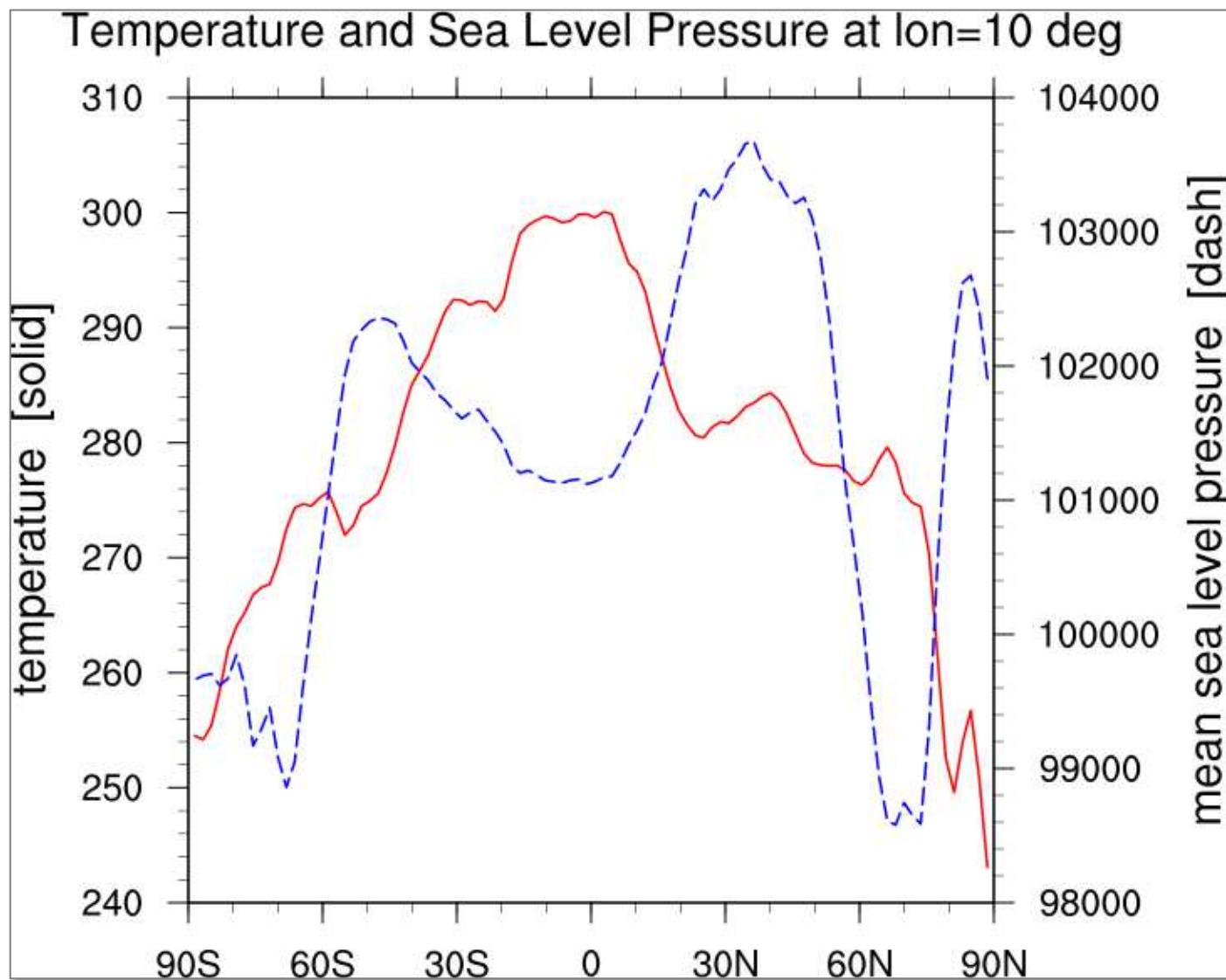
  resSLP         = True
  resSLP@xyLineColor = "blue"      ;-- set line colors
  resSLP@xyLineThicknessF = 3.0     ;-- set line thickness
  resSLP@xyDashPatterns = 1         ;-- dashed line
  resSLP@tiYAxisString = slp@long_name +" "+"[dash]" ;-- axis string
  resSLP@tiMainString = "Temperature and Sea Level Pressure at lon=10 deg"

  plot = gsn_csm_xy2(wks, x, t, slp, resT, resSLP) ;-- create the plot

end

```

## XY-plot two different y-axis



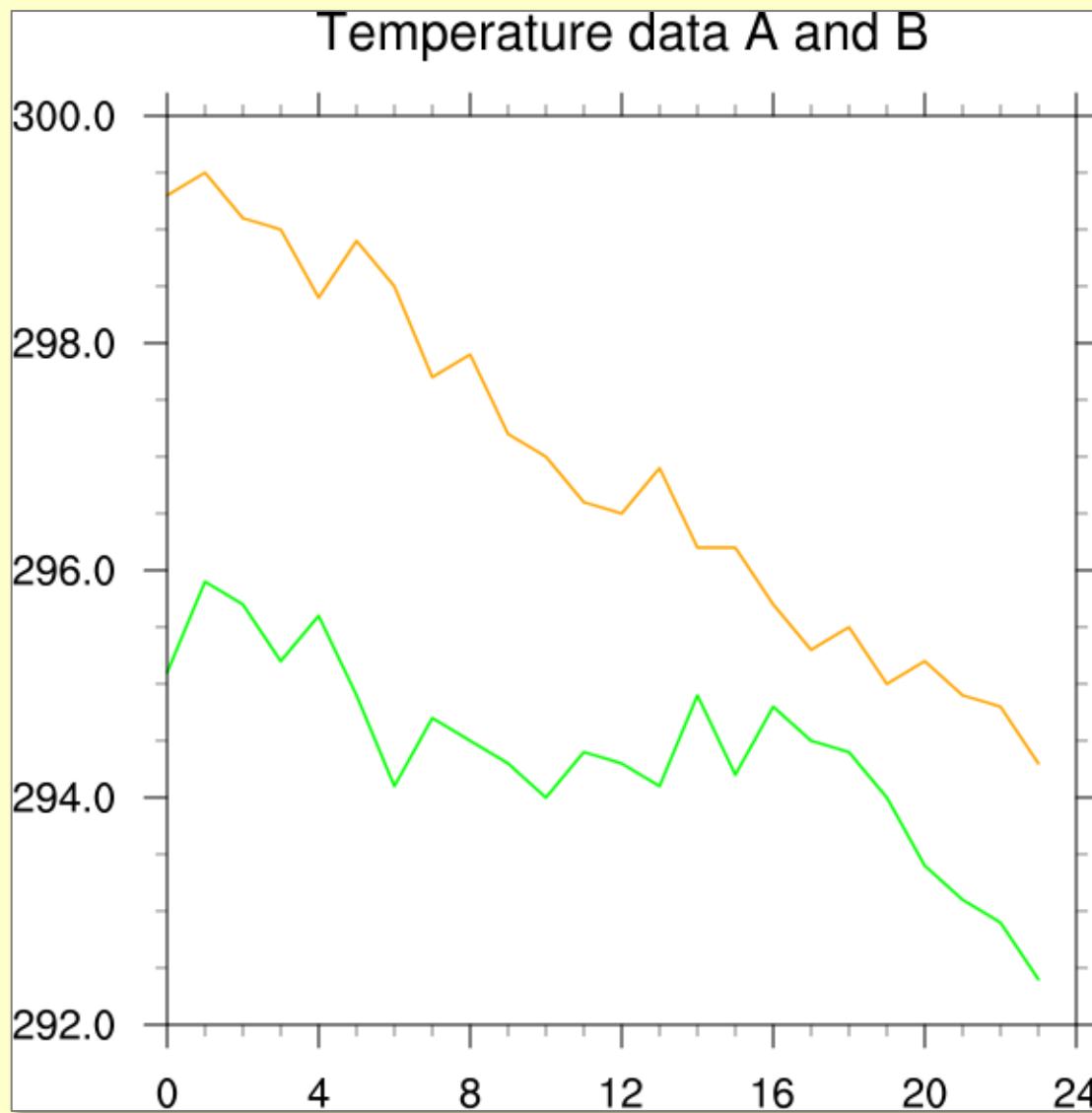
# Task: XY-plot script reading data from ASCII file

- Use the script `part_I_read_ASCII_data_xy.ncl`
- Copy it to your own directory and modify it
  - Read data file `ASCII_data_II.asc` column **1** and **2**
  - Define x-axis as `time` with **24** one hour integer values
  - Plot data with different line colors (**green** and **orange**)
  - Make the **lines thicker** (=3)

## Hints:

- `use ispan to create time x-axis`
- `xyMonoDashPattern`
- `xyLineThicknessF`
- `xyLineColors`

## Task: XY-plot script reading data from ASCII file



# Task: XY-plot script reading data from ASCII file

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  data_col = asciread("ASCII_data_II.asc", (/24,3/), "float")

  date    = ispan(0,23,1)           ;-- generate time array of 24 one hour values

  mod_a   = data_col(:,1)          ;-- get all data from column 2
  mod_b   = data_col(:,2)          ;-- get all data from column 3

  wks = gsn_open_wks("png", "task_I_read_ASCII_data_2_xy") ;-- open a workstation

  res           = True            ;-- create plot resource object
  res@tiMainString = "Temperature data A and B" ;-- draw a title

  res@xyMonoDashPattern = True      ;-- use solid lines pattern for all
  res@xyLineThicknessF  = 3          ;-- set line thickness
  res@xyLineColors     = (/"green","orange"/) ;-- line colors

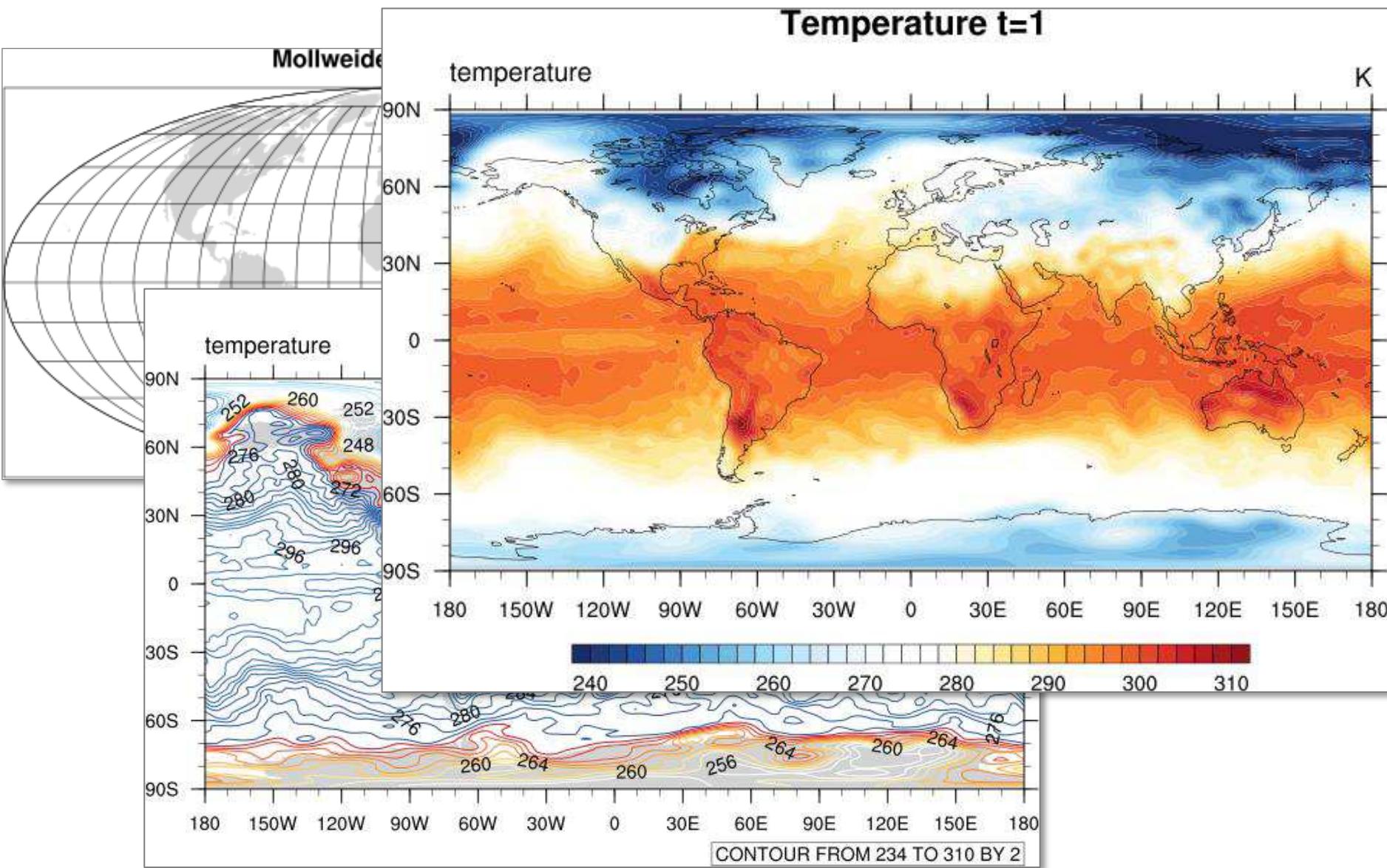
  plot = gsn_csm_xy(wks, date, (/mod_a,mod_b/), res) ;-- create the plot
end
```

## Part II

# Maps and Contours

### Exercises and Tasks

# Maps and Contours



# Simple map plot

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

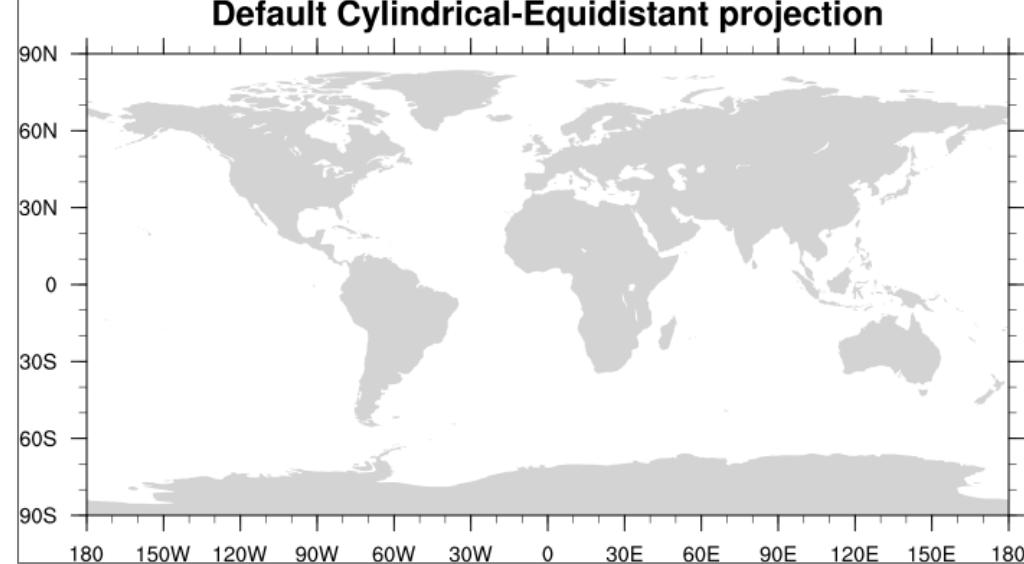
begin

wks = gsn_open_wks("png", "part_II_simple_map")

res           = True
res@tiMainString = "Default Cylindrical-Equidistant projection"

plot = gsn_csm_map(wks, res)           ;-- create the plot

end
```



# Map plot changing the projection

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

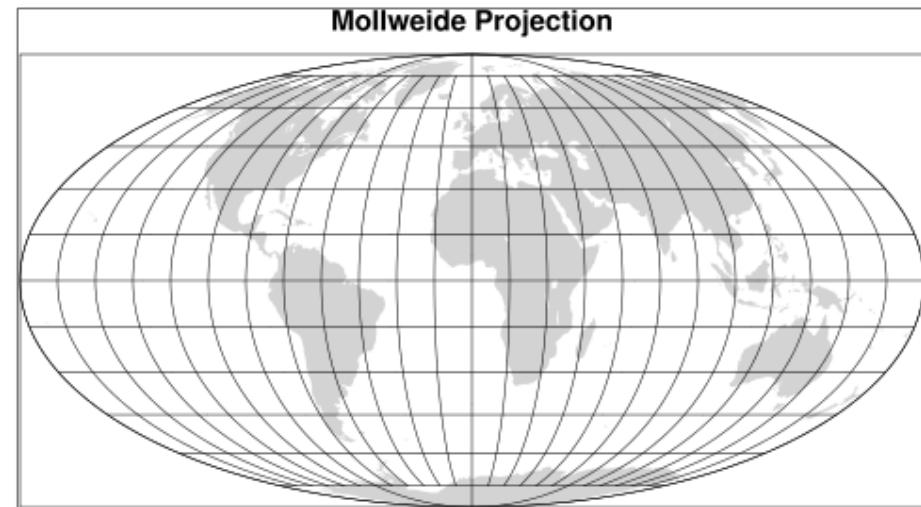
begin

wks = gsn_open_wks ("png", "part_II_projection_map")

res           = True
res@mpProjection = "Mollweide"          ;-- set projection
res@mpGridAndLimbOn = True               ;-- turn on grid lines
res@tiMainString = "Mollweide Projection" ;-- draw a title

plot = gsn_csm_map(wks, res)            ;-- create the plot

end
```



# Map plot selecting a subregion

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin

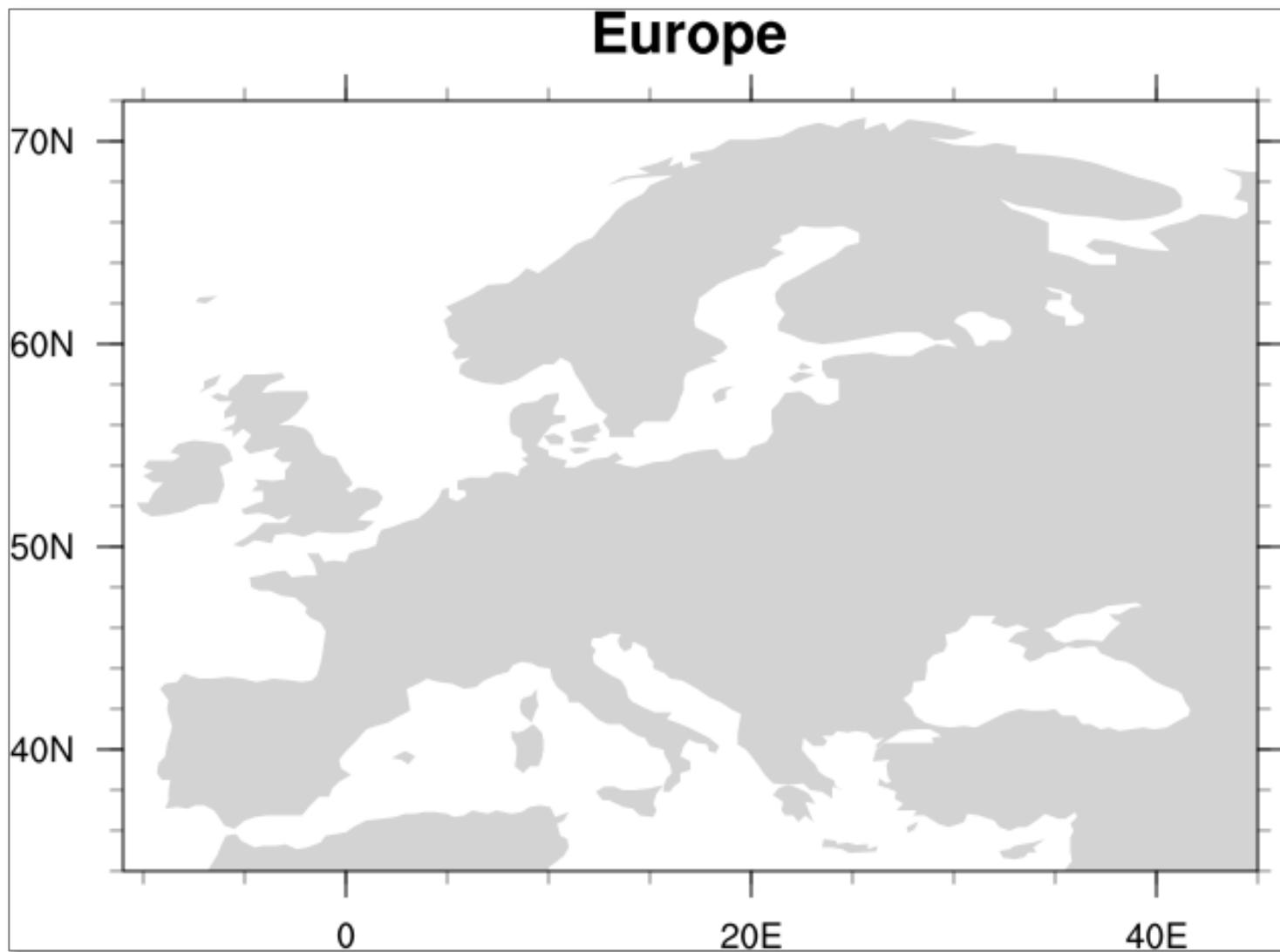
wks = gsn_open_wks("png", "part_II_subregion_map")

res           = True
res@tiMainString = "Europe"          ;-- draw a title
res@mpMinLonF   = -11                ;-- min longitude
res@mpMaxLonF   = 45                 ;-- max longitude
res@mpMinLatF   = 34                 ;-- min latitude
res@mpMaxLatF   = 72                 ;-- max latitude

plot = gsn_csm_map(wks, res)         ;-- create the plot

end
```

# Map plot selecting a subregion



# Map plot changing the resolution

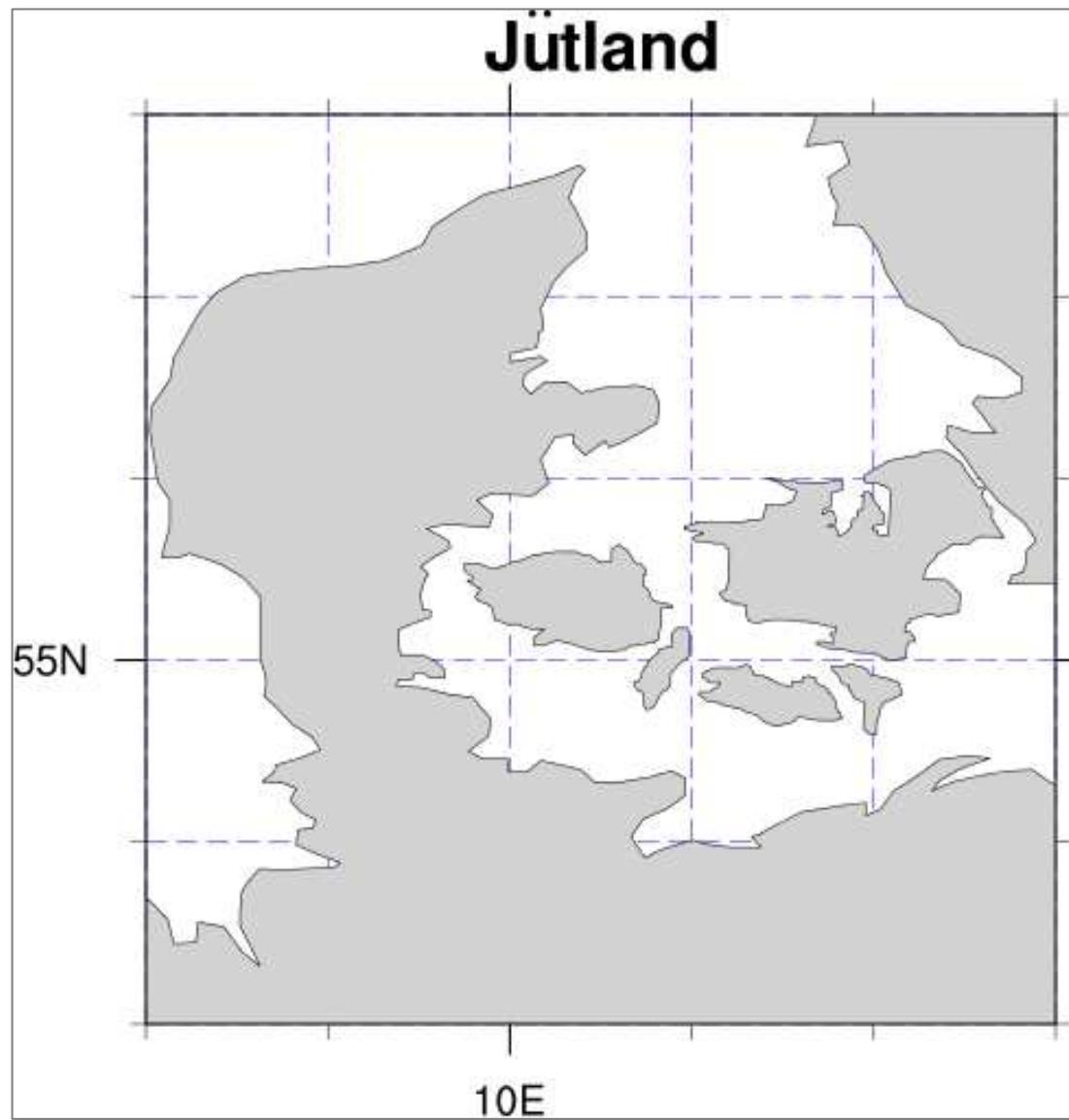
```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  wks = gsn_open_wks("png", "part_II_resolution_map")

  res             = True
  res@tiMainString = "Ju~H-15V6F35~H~FV-6H3~tland" ;-- draw a title
  res@mpLimitMode   = "LatLon"                      ;-- mode for limits
  res@mpMinLonF     = 8                            ;-- min longitude
  res@mpMaxLonF     = 13                           ;-- max longitude
  res@mpMinLatF     = 53                           ;-- min latitude
  res@mpMaxLatF     = 58                            ;-- max latitude
  res@mpDataBaseVersion = "MediumRes"            ;-- set map resolution
  res@mpGridAndLimbOn = True                         ;-- draw grid lines
  res@mpGridLineColor = "blue"                      ;-- grid line color
  res@mpOutlineOn    = True                         ;-- draw coastlines
  res@mpGridSpacingF = 1                            ;-- grid line spacing
  res@mpGridLineThicknessF = 1                     ;-- grid line thickness
  res@mpGridLineDashPattern = 1                   ;-- grid line dash pattern
  res@mpGridAndLimbDrawOrder = "PreDraw"           ;-- first draw grid then map

  plot = gsn_csm_map(wks, res)                    ;-- create the plot
end
```

# Map plot changing the resolution



# Contour plot using defaults

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin

  f = addfile("$NCL_TUT/data/EH_OM_A1B_3Dvectors.nc", "r")
  t = f->t(0,0,:,:)

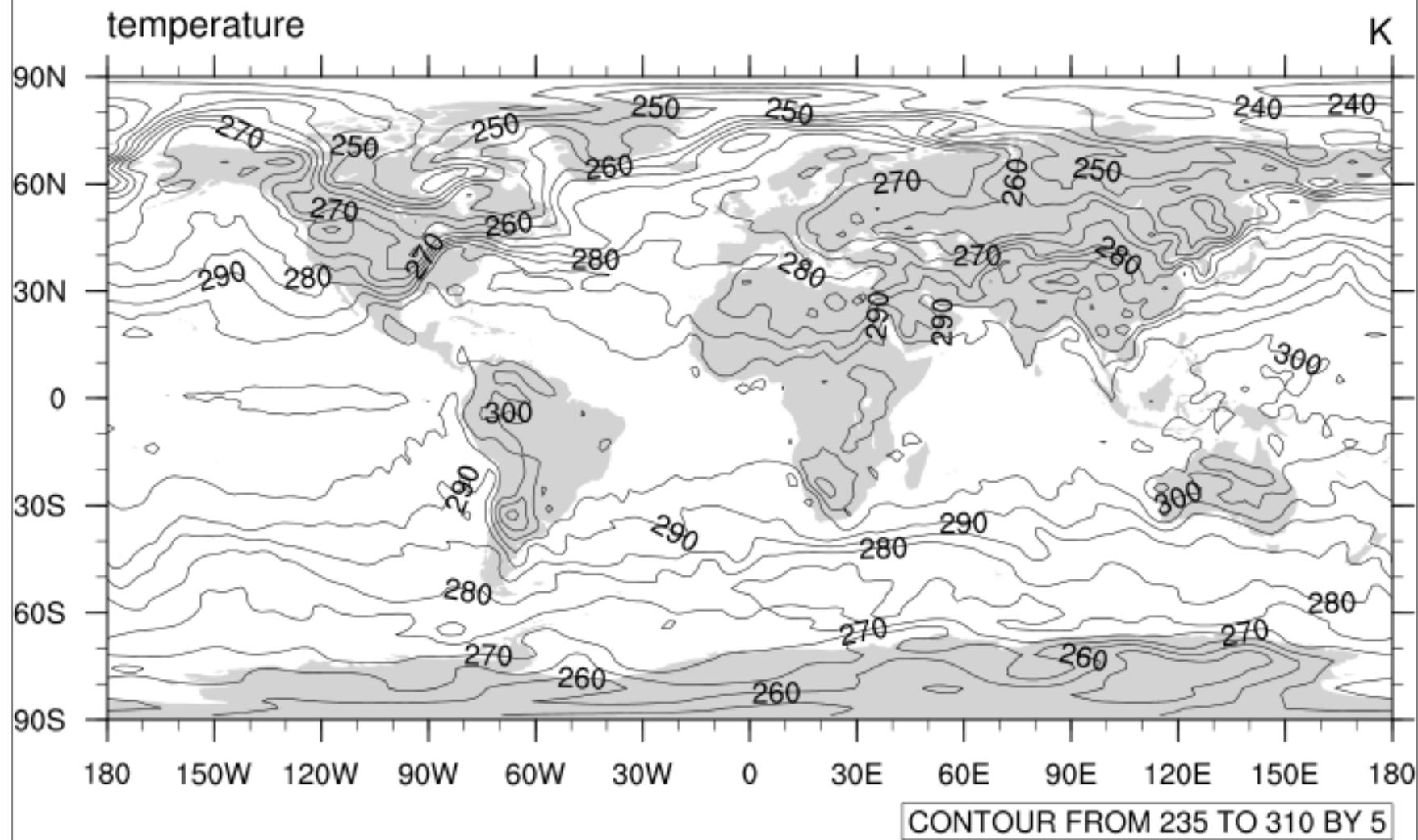
  wks = gsn_open_wks("png", "part_II_contour_def_map")

  res           = True
  res@tiMainString = "Temperature t=1"          ;-- draw a title

  plot = gsn_csm_contour_map(wks, t, res)        ;-- create the plot

end
```

## Temperature t=1



# Contour plot with colored lines

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin

f = addfile("$NCL_TUT/data/EH_OM_A1B_3Dvectors.nc", "r")
t = f->t(0,0,:,:)

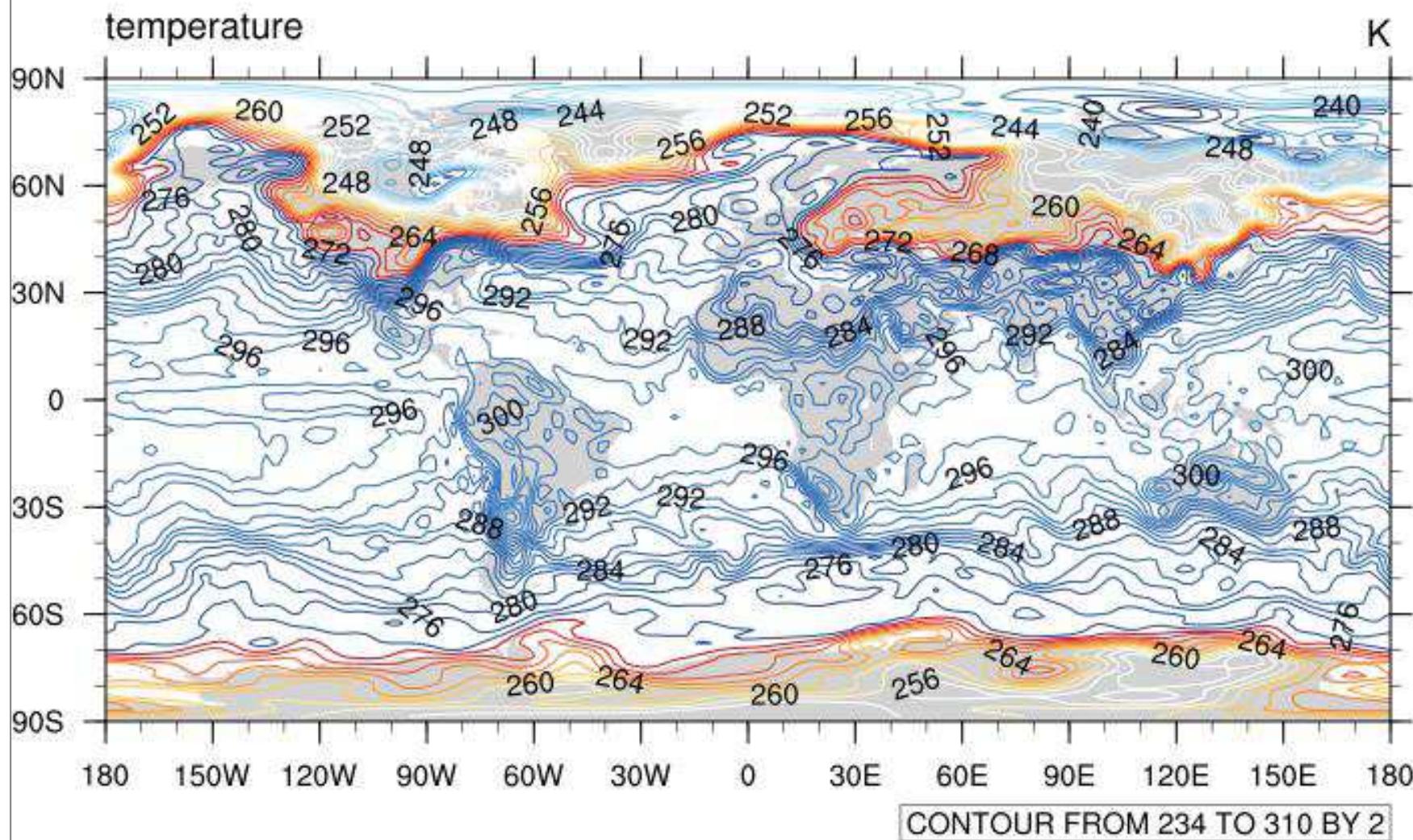
wks = gsn_open_wks("png", "part_II_contour_lines_col_map")

res
      = True
res@tiMainString
      = "Temperature t=1"           ;-- draw a title
res@cnLineThicknessF
      = 2                           ;-- make lines thicker
res@cnMonoLineColor
      = False                         ;-- use multiple line colors
res@cnLineColors
      = toint(fspan(2,240,20)) ;-- define the colors
res@cnLevelSpacingF
      = 2                           ;-- contour level spacing

plot = gsn_csm_contour_map(wks, t, res)          ;-- create the plot

end
```

# Temperature t=1



# Contour plot with color fill

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin

f = addfile("$NCL_TUT/data/EH_OM_A1B_3Dvectors_t1-24.nc", "r")
t = f->t(0,0,:,:)

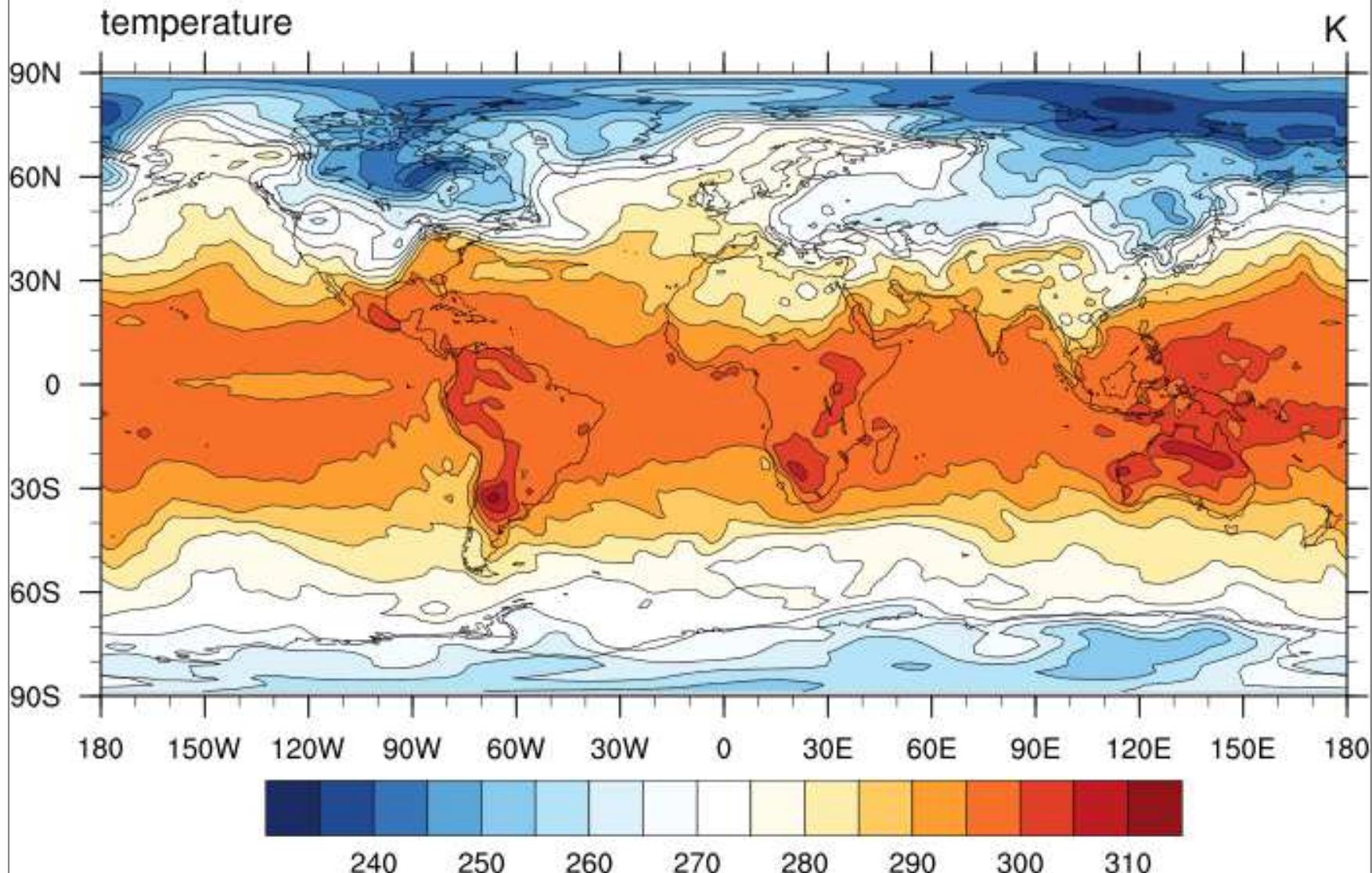
wks = gsn_open_wks("png", "part_II_contour_fill_map")

res           = True
res@tiMainString = "Temperature t=1"      ;-- draw a title
res@cnFillOn   = True                      ;-- color fill between contour levels

plot = gsn_csm_contour_map(wks, t, res)    ;-- create the plot

end
```

# Temperature t=1



# Contour plot with line settings

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f = addfile("$NCL_TUT/data/EH_OM_A1B_3Dvectors_t1-24.nc","r")
  t = f->t(0,0,:,:)

  wks = gsn_open_wks("png", "part_II_contour_fill_line_settings_map")

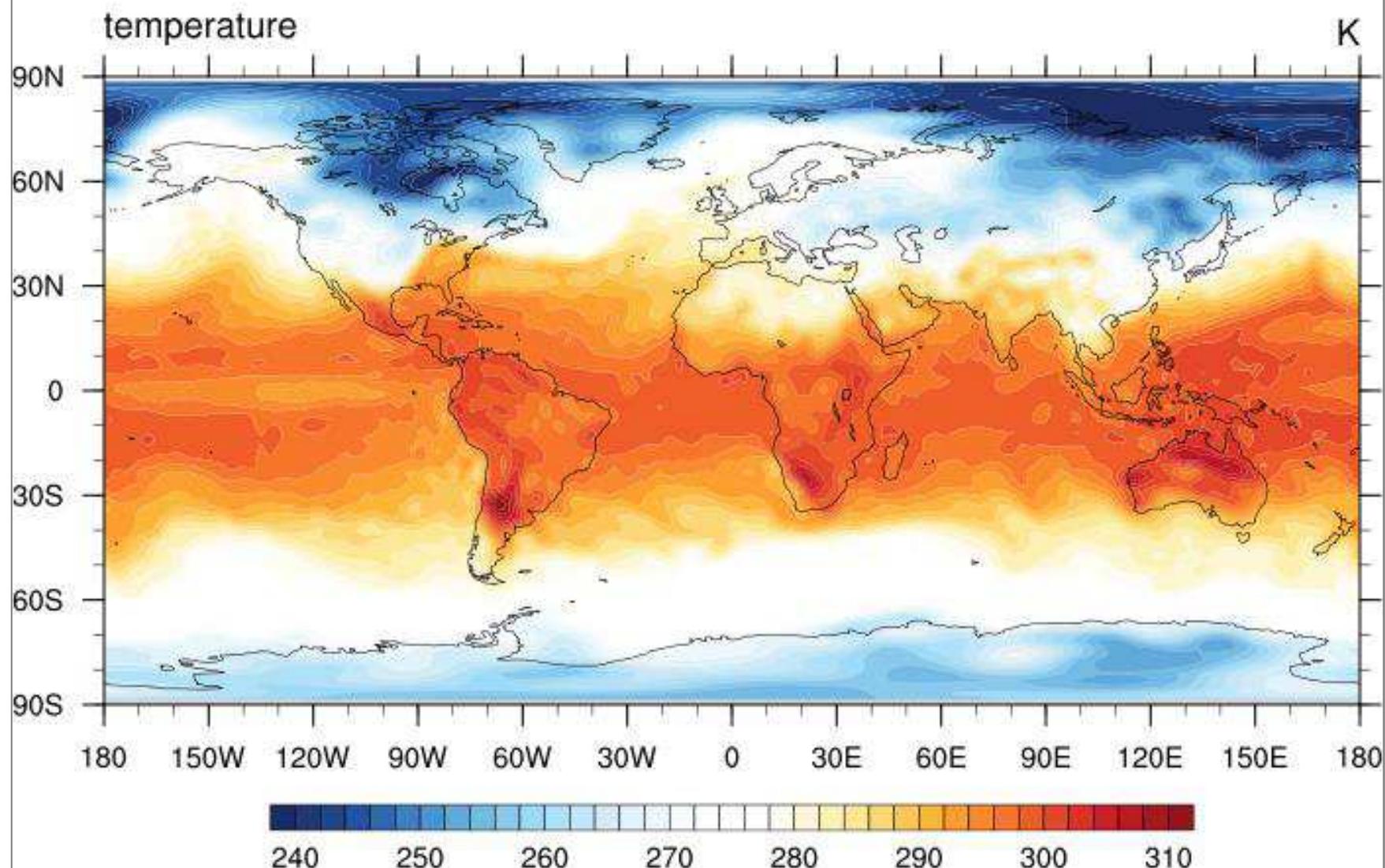
  res
    = True
  res@tiMainString
    = "Temperature t=1" ;-- draw a title
  res@cnFillOn
    = True ;-- color fill between contour levels
  res@cnLinesOn
    = False ;-- don't draw contour lines
  res@cnLevelSelectionMode
    = "ManualLevels" ;-- set contour levels manually
  res@cnMinLevelValF
    = 240. ;-- minimum contour level
  res@cnMaxLevelValF
    = 310. ;-- maximum contour level
  res@cnLevelSpacingF
    = 2 ;-- contour level spacing

  res@lbLabelStride
    = 5
  res@lbBoxMinorExtentF
    = 0.15 ;-- decrease the height of the labelbar

  plot = gsn_csm_contour_map(wks, t, res) ;-- create the plot

end
```

# Temperature t=1



# Contour plot with grid lines (1/3)

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f = addfile("$NCL_TUT/data/EH_OM_A1B_3Dvectors_t1-24.nc","r")
  t = f->t(0,0,:,:)

  wks = gsn_open_wks("png", "part_II_contour_fill_line_settings_map")
  gsn_define_colormap(wks,"rainbow")           ;-- set colormap

  res          = True
  res@tiMainString = "Temperature t=1" ;-- draw a title
  res@cnFillOn      = True           ;-- color fill between contour levels
  res@cnLinesOn     = False          ;-- don't draw contour lines
  res@cnLevelSelectionMode = "ManualLevels" ;-- set contour levels manually
  res@cnMinLevelValF = 240.          ;-- minimum contour level
  res@cnMaxLevelValF = 310.          ;-- maximum contour level
  res@cnLevelSpacingF = 2            ;-- contour level spacing

  res@lbLabelStride = 5
  res@lbBoxMinorExtentF = 0.15       ;-- decrease the height of the labelbar
```

# Contour plot with grid lines (2/3)

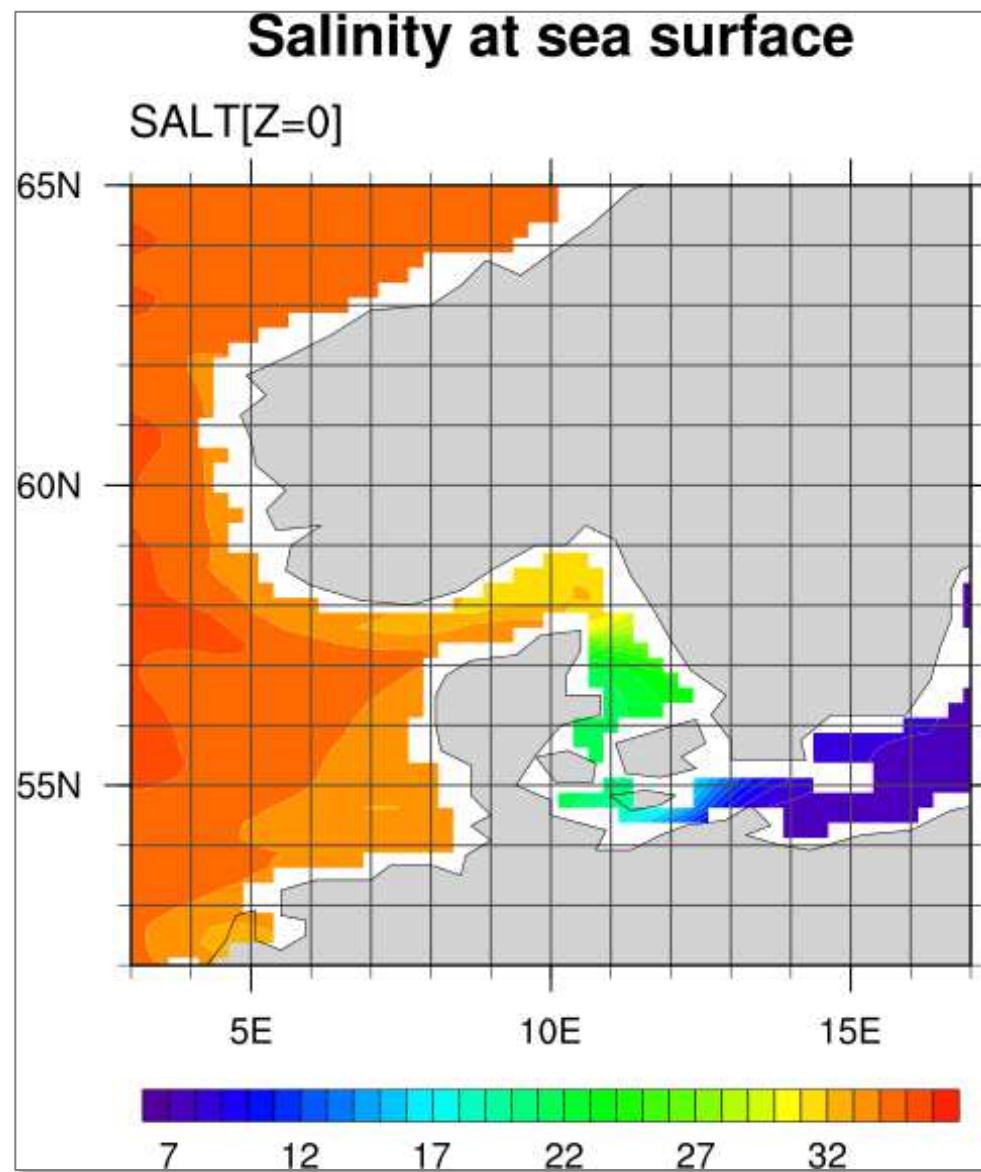
```
res@mpMinLonF          = 3                      ;-- min longitude
res@mpMaxLonF          = 17                     ;-- max longitude
res@mpMinLatF          = 52                     ;-- min latitude
res@mpMaxLatF          = 65                     ;-- max latitude

res@mpGridAndLimbOn    = True                   ;-- plot grid lines

res@mpGridSpacingF     = 1                      ;-- grid line spacing 1 degree
res@mpGridLineThicknessF = 2.0                  ;-- make grid lines thicker
res@mpGridLineColor     = "Gray30"              ;-- grid line color

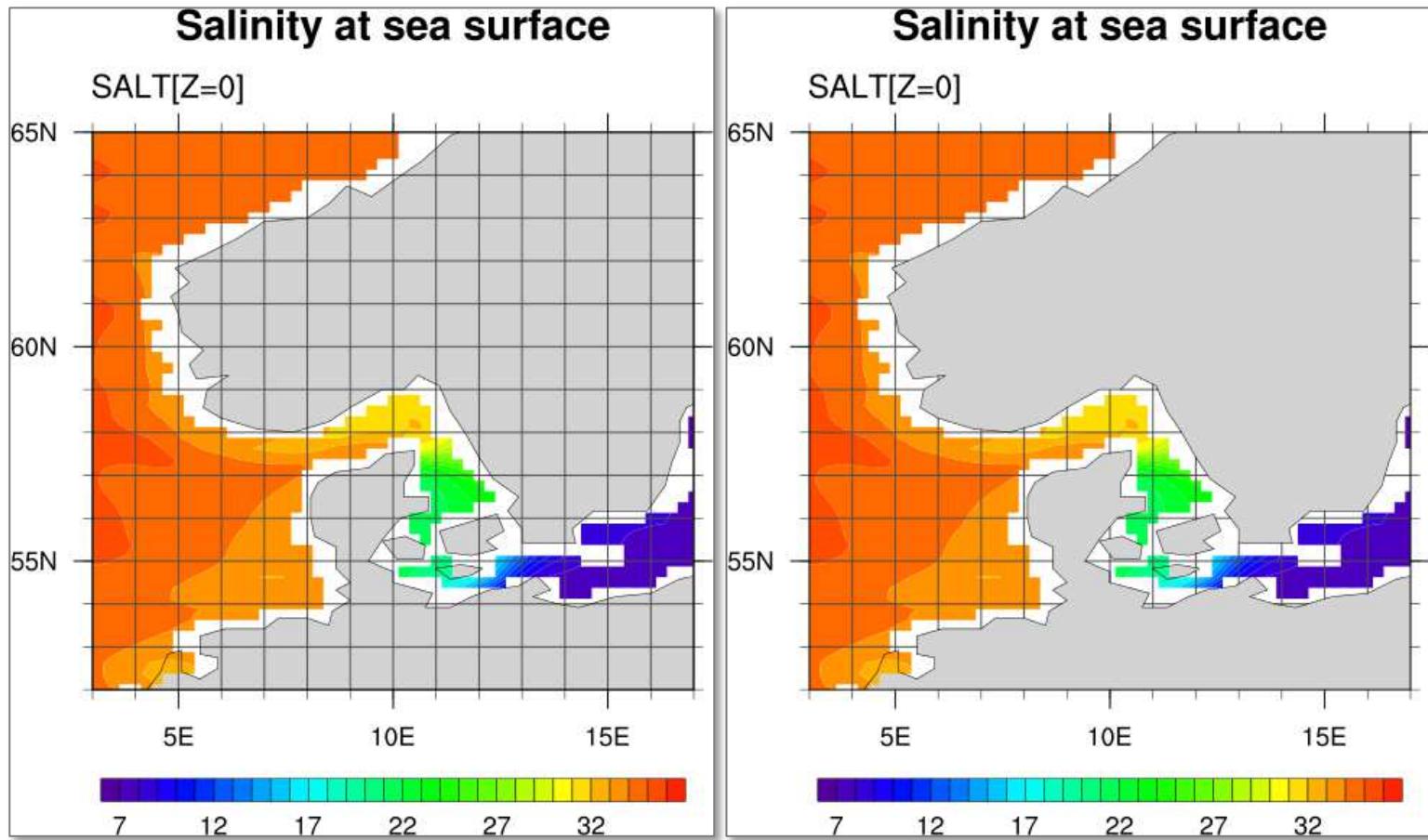
res@gsnSpreadColors    = True                  ;-- spread color map
res@gsnSpreadColorStart = 2                     ;-- start color index
res@gsnSpreadColorEnd   = -6                   ;-- end color index

;-- draw the contour map
plot = gsn_csm_contour_map(wks, var, res)
```



# Contour plot with grid lines (3/3)

```
res@mpGridMaskMode      = "MaskLand"          ;-- mask grid over land  
plot = gsn_csm_contour_map(wks, var, res)    ;-- create a second plot  
  
end
```



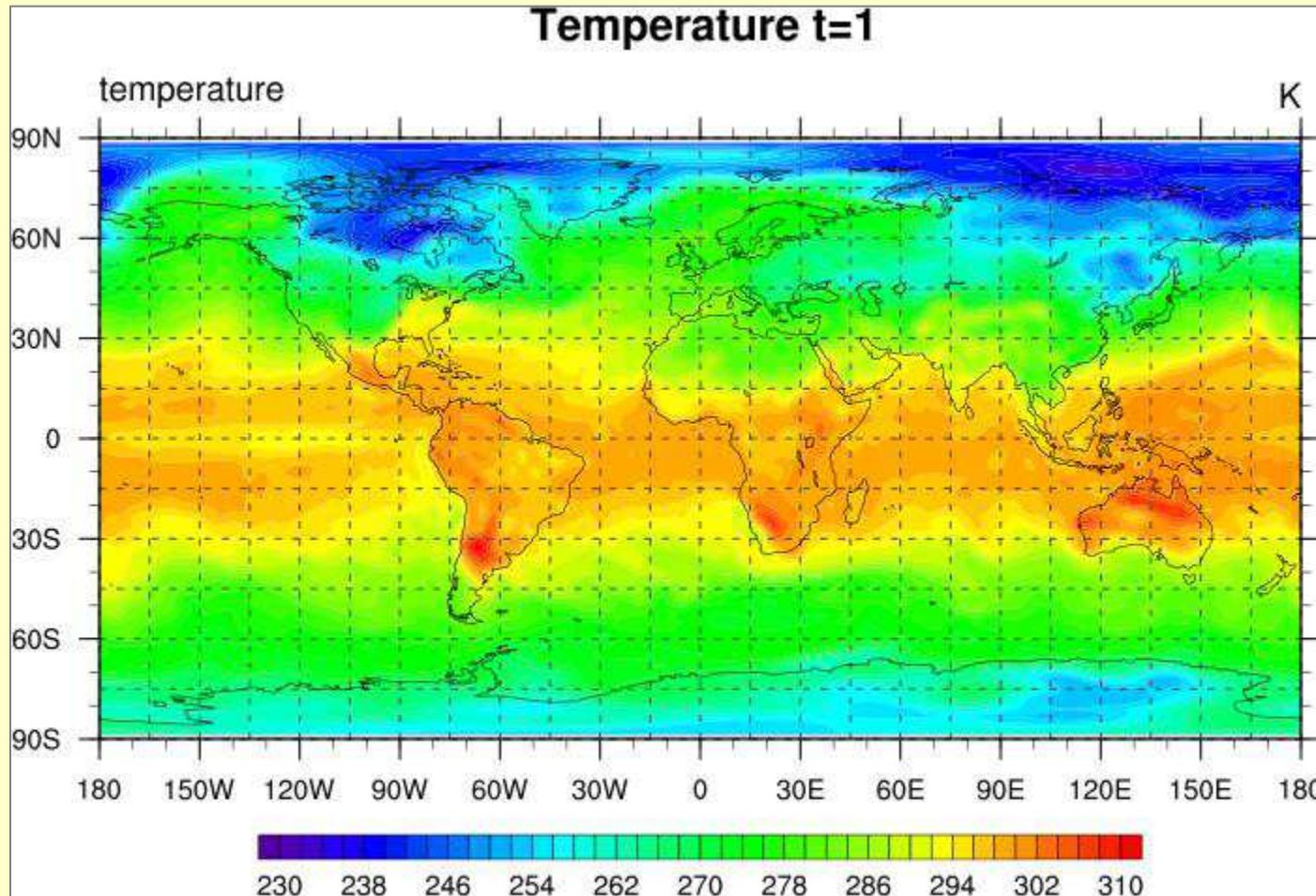
## Task: Contour plot

- Use the script `part_II_contour_fill_map.ncl`
- Copy it to your own directory and modify it
  - Switch **off** the black contour lines
  - Set the contour levels manually to **min=230** and **max=310**
  - Change the contour level spacing to **2 degrees**
  - Draw grid lines with **color=Gray30**, **thickness=2** and dash **pattern=2**
  - Change the colormap to **rainbow**

Hints:

- `cnLevelSelectionMode = "ManualLevels"`
- `cnMinLevelValF, cnMaxLevelValF`
- `cnLevelSpacingF`
- `mpGridAndLimbOn, mpGridLines*`
- `cnFillPalette = "rainbow"`

# Task: Contour plot



# Task: Contour plot with grid lines (1/2)

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f = addfile("../../EH_OM_A1B_3Dvectors_t1-24.nc","r")
  t = f->t(0,0,:,:)

  wks = gsn_open_wks("png", "task_II_contour_map")

  res           = True                      ;-- create plot resource object
  res@tiMainString = "Temperature t=1"      ;-- draw a title on top of the plot
  res@cnFillOn   = True                      ;-- color fill between contour levels
  res@cnFillPalette = "rainbow"              ;-- change the colormap
  res@cnLinesOn   = False                     ;-- switch off contour lines
  res@cnLevelSelectionMode = "ManualLevels" ;-- set contour levels manually
  res@cnMinLevelValF = 230.                   ;-- minimum contour level
  res@cnMaxLevelValF = 310.                   ;-- maximum contour level
  res@cnLevelSpacingF = 2.                    ;-- contour level spacing
```

# Task: Contour plot with grid lines (2/2)

```
res@lbLabelStride      = 4           ;-- labelbar label increment
res@lbBoxMinorExtentF = 0.15        ;-- decrease the height of the labelbar

res@mpGridAndLimbOn   = True         ;-- plot grid lines
res@mpGridLineThicknessF = 2.         ;-- grid line thickness
res@mpGridLineColor    = "Gray30"     ;-- grid line color
res@mpGridLineDashPattern = 2         ;-- grid line dash pattern

;-- create the plot
plot = gsn_csm_contour_map(wks, t, res)

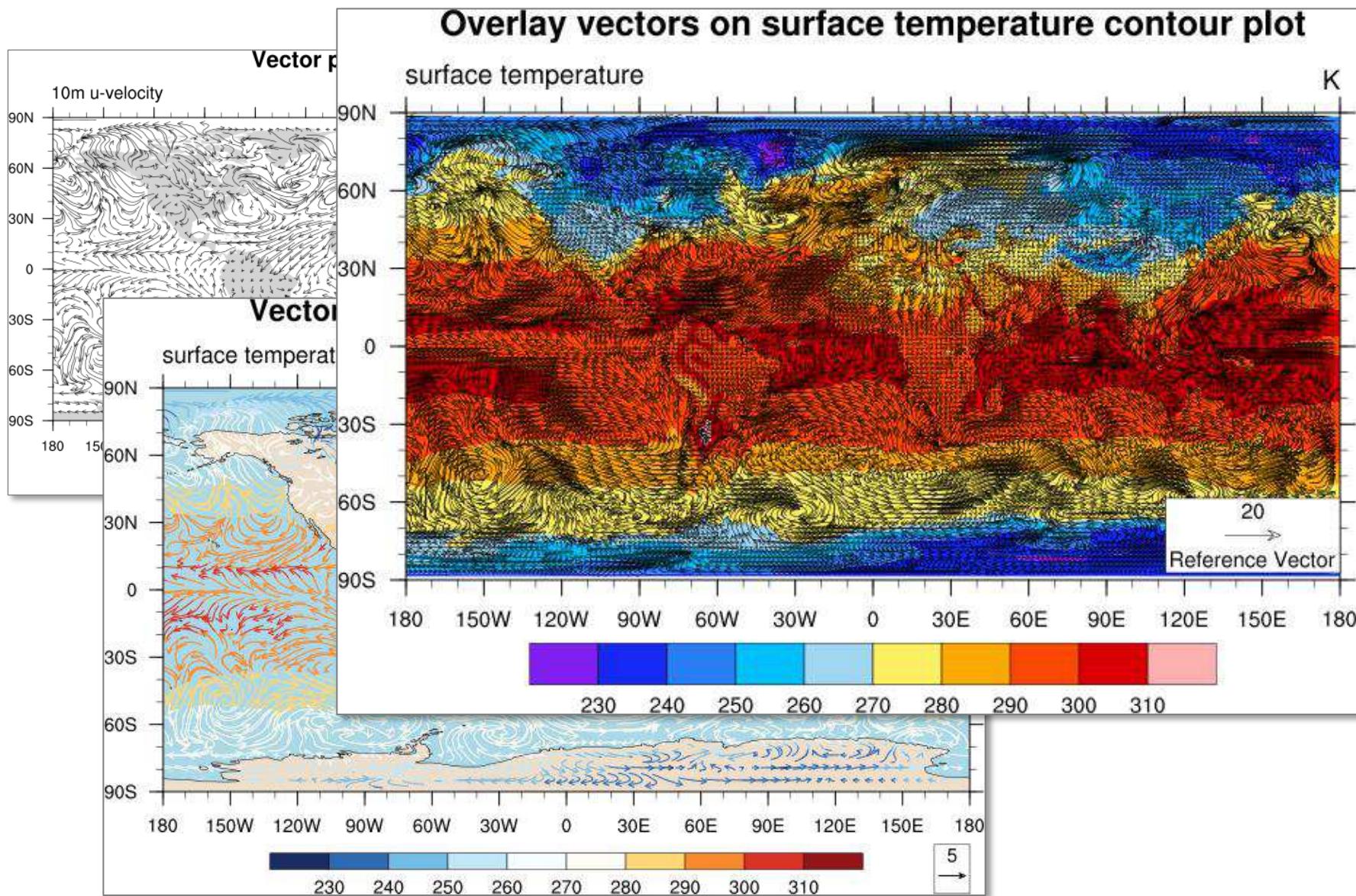
end
```

# Part III

## Vectors

### Exercises and Tasks

# Vectors



# Simple vector plot

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin

f      = addfile("../ECHAM5_OM_A1B_2001_0101-1001_2D.nc","r")
u      = f->u10(0,:,:)
v      = f->v10(0,:,:)
                                ;-- u-velocity, first time step
                                ;-- v-velocity, first time step

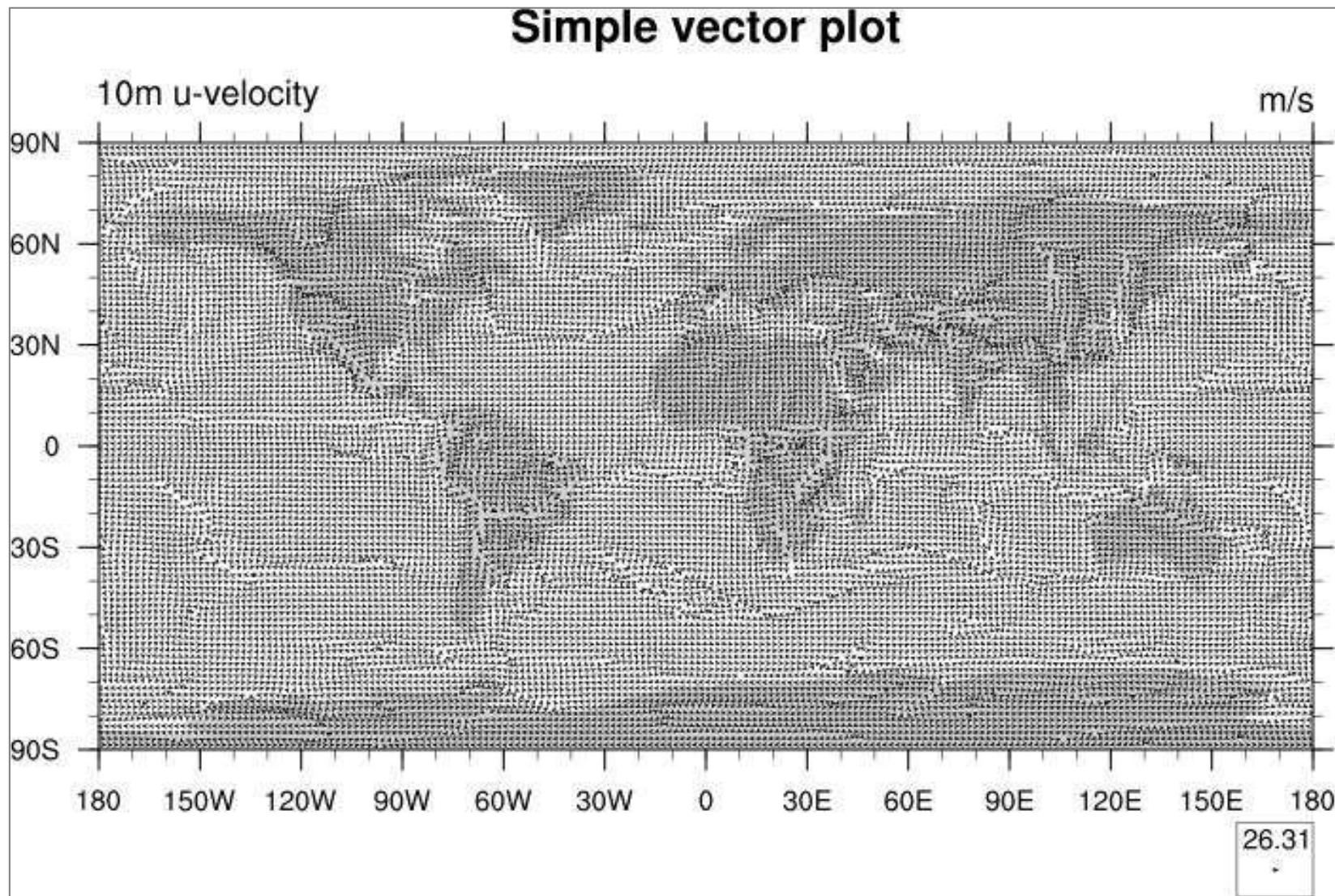
wks   = gsn_open_wks("png","part_III_simple_vectors")

res           = True
                ;-- create plot resource object
res@tiMainString = "Simple vector plot" ;-- draw a title on top of the plot

plot = gsn_csm_vector_map_ce(wks,u,v,res)

end
```

# Simple vector plot



# Vector plot settings

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f      = addfile("../ECHAM5_OM_A1B_2001_0101-1001_2D.nc", "r")
  u      = f->u10(0,:,:)
  v      = f->v10(0,:,:)
  ;-- u-velocity, first time step
  ;-- v-velocity, first time step

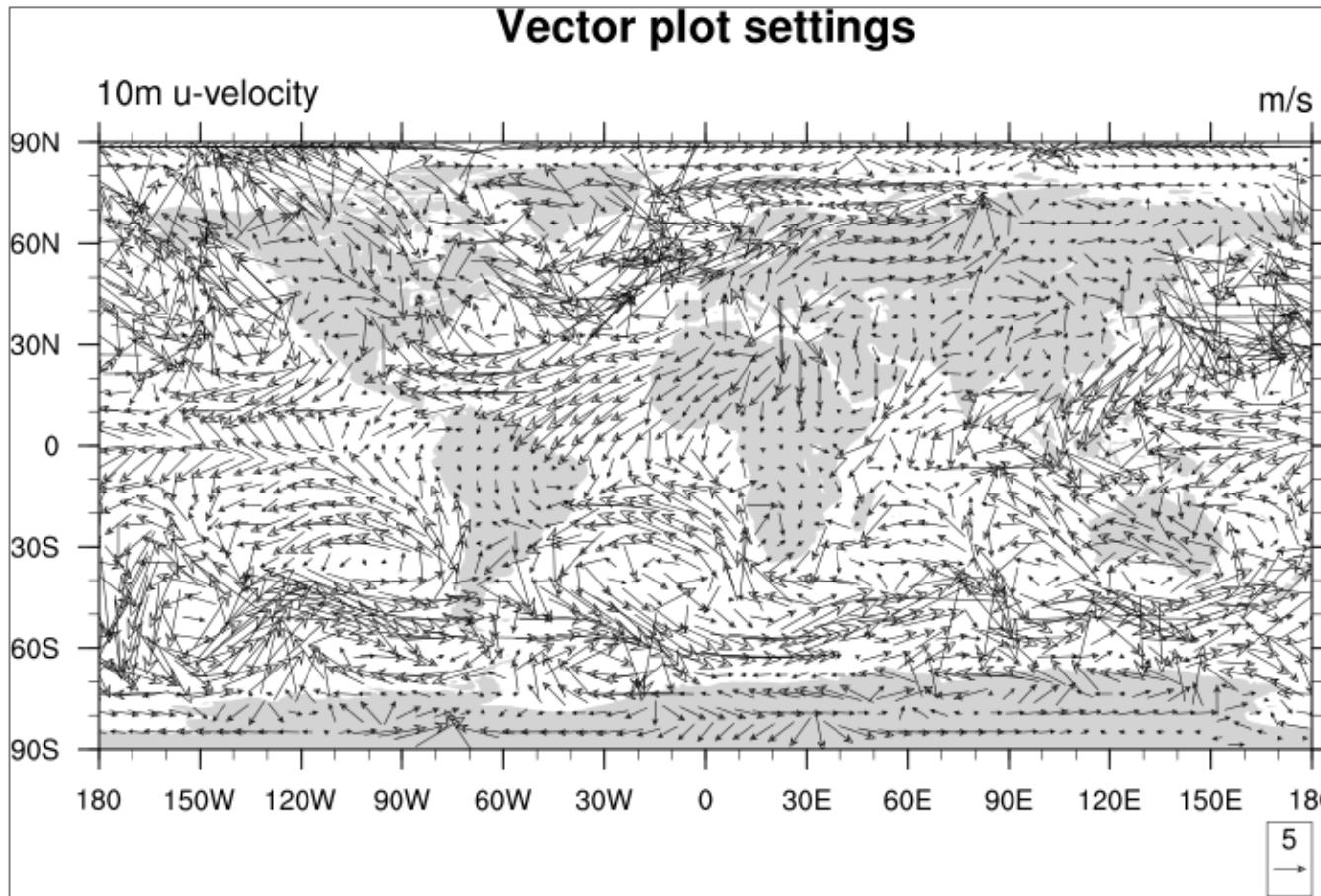
  wks = gsn_open_wks("png", "part_III_vector_settings")

  res           = True          ;-- create plot resource object
  res@tiMainString = "Vector plot settings" ;-- draw a title

  res@vcRefMagnitudeF = 5.        ;-- make vectors larger
  res@vcRefLengthF    = 0.02       ;-- reference vector length
  res@vcMinDistanceF = 0.01       ;-- thin out vectors

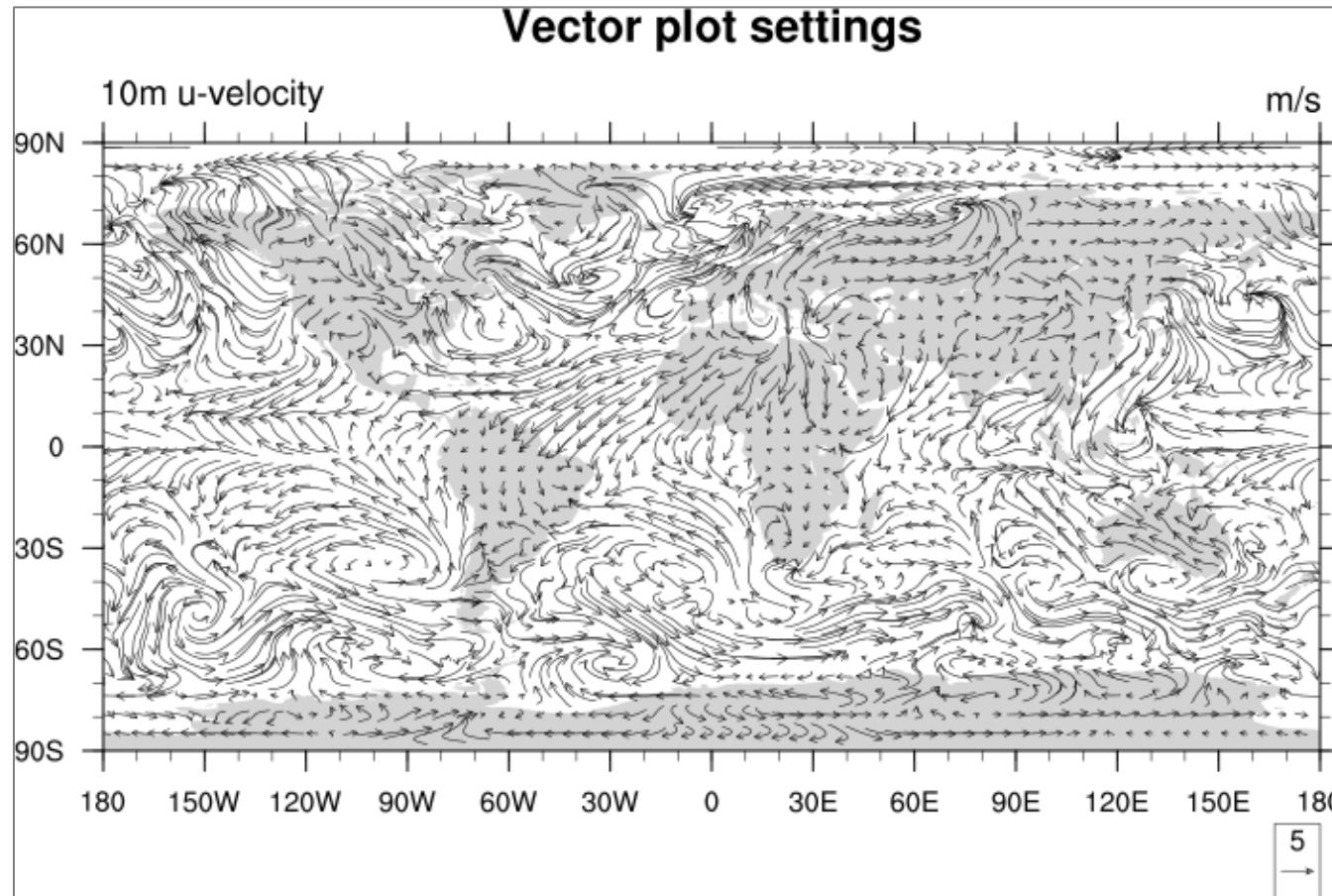
  plot = gsn_csm_vector_map_ce(wks, u, v, res)
```

# Vector plot settings



# Vector plot settings

```
res@vcGlyphStyle      = "CurlyVector"          ;-- turn on curly vectors  
plot = gsn_csm_vector_map_ce(wks,u,v,res)  
end
```



# Vector plot colorize

```

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
    f      = addfile("../ECHAM5_OM_A1B_2001_0101-1001_2D.nc","r")
    u      = f->u10(0,:,:)
                           ;-- u-velocity, first time step
    v      = f->v10(0,:,:)
                           ;-- v-velocity, first time step

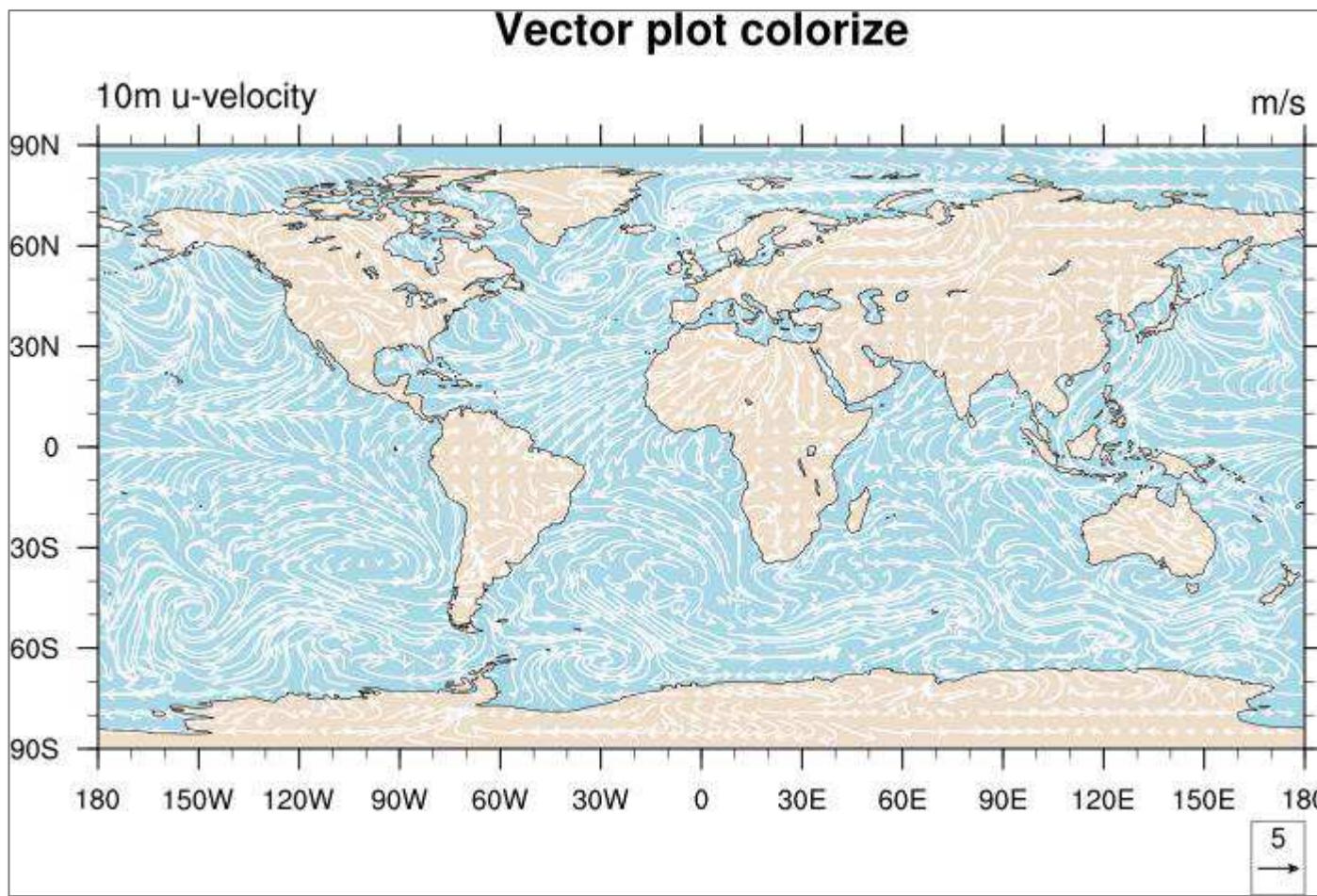
    wks   = gsn_open_wks("png","part_III_vector_colorize")

    res          = True           ;-- create plot resource object
    res@tiMainString = "Vector plot colorize" ;-- draw a title
    res@vcMinDistanceF = 0.01        ;-- thin out vectors
    res@vcGlyphStyle = "CurlyVector" ;-- turn on curly vectors
    res@vcLineArrowColor = "white"    ;-- change vector color to white
    res@vcLineArrowThicknessF = 2.0    ;-- change vector thickness
    res@vcRefMagnitudeF = 5.         ;-- make vectors larger
    res@vcRefLengthF = 0.025        ;-- ref vector length
    res@vcRefAnnoArrowUseVecColor = False ;-- don't use vcLineArrowColor for ref
    res@mpOutlineOn = True          ;-- draw coastlines
    res@mpLandFillColor = "AntiqueWhite2" ;-- fill land areas with color
    res@mpOceanFillColor = "lightblue" ;-- fill ocean areas with color
    res@mpInlandWaterFillColor = "lightblue" ;-- fill inland water areas with color

    plot = gsn_csm_vector_map_ce(wks,u,v,res)
end

```

# Vector plot colorize



# Vector plot colorized by surface temperature (1/2)

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f = addfile("../ECHAM5_OM_A1B_2001_0101-1001_2D.nc","r")
  u = f->u10(0,:,:)
  v = f->v10(0,:,:)
  t = f->tsurf(0,:,:)
  ;-- u-velocity, first time step
  ;-- v-velocity, first time step
  ;-- surface temperature

  wks = gsn_open_wks("png","part_III_vector_colorize_temp")

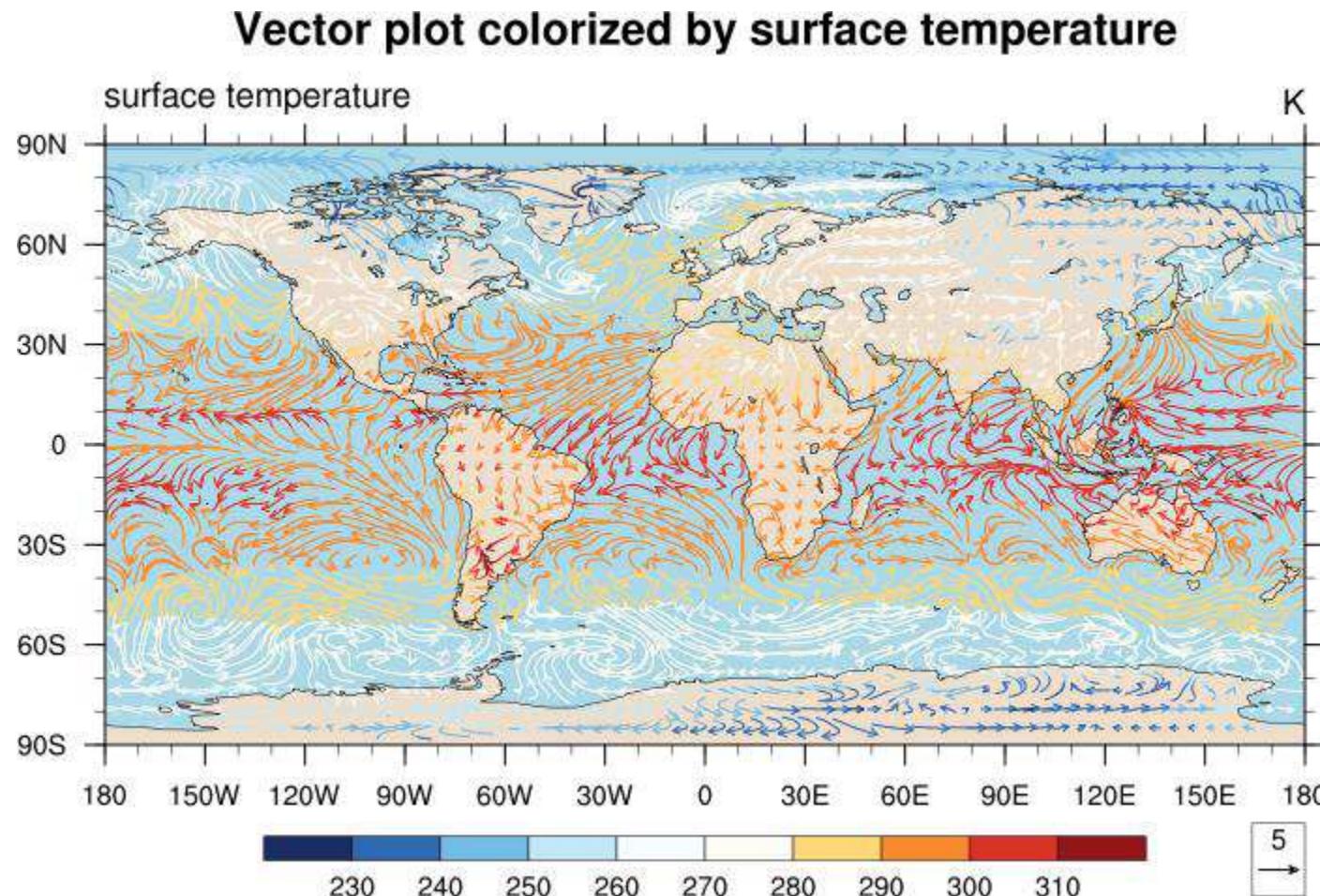
  res          = True
  ;-- create plot resource object
  res@tiMainString = "Vector plot colorized by surface temperature"

  res@vcGlyphStyle      = "CurlyVector"
  ;-- turn on curly vectors
  res@vcMinDistanceF   = 0.01
  ;-- thin out vectors
  res@vcLineArrowThicknessF = 2.0
  ;-- change vector thickness
  res@vcRefMagnitudeF  = 5.
  ;-- make vectors larger
  res@vcRefLengthF     = 0.025
  ;-- ref vector length
  res@vcRefAnnoArrowUseVecColor = False
  ;-- don't use vcLineArrowColor for ref
```

# Vector plot colorized by surface temperature (2/2)

```
res@mpOutlineOn          = True           ;-- draw coastlines  
  
res@mpLandFillColor     = "AntiqueWhite2" ;-- fill land areas with color  
res@mpOceanFillColor    = "lightblue"      ;-- fill ocean areas with color  
res@mpInlandWaterFillColor = "lightblue" ;-- fill inland water areas with color  
  
;-- create the plot  
plot = gsn_csm_vector_scalar_map_ce(wks,u,v,t,res)  
  
end
```

# Vector plot colorized by surface temperature



# Vector plot overlay on contour plot (1/2)

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f      = addfile("../ECHAM5_OM_A1B_2001_0101-1001_2D.nc","r")
  u      = f->u10(0,:,:)
  v      = f->v10(0,:,:)
  t      = f->tsurf(0,:,:)

  wks = gsn_open_wks("png","part_III_vector_colorize_overlay")
  gsn_define_colormap(wks,"ncl_default")           ;-- choose color map

  cnres          = True                         ;-- contour plot resources
  cnres@gsnDraw = False                        ;-- don't draw plot, yet
  cnres@gsnFrame = False                       ;-- don't advance frame
  cnres@gsnLeftString = t@long_name           ;-- change left string
  cnres@gsnRightString = t@units              ;-- change right string
  cnres@tiMainString = "Overlay vectors on surface temperature contour plot"
  cnres@cnFillOn = True                         ;-- draw filled contours
  cnres@cnLinesOn = False                       ;-- don't draw contour lines
  cnres@mpFillOn = False                        ;-- no map fill
```

# Vector plot overlay on contour plot (2/2)

```

vcres          =  True           ;-- vector resources
vcres@gsnDraw      =  False        ;-- don't draw vector plot, yet
vcres@gsnFrame     =  False        ;-- don't advance frame
vcres@gsnRightString =  " "
vcres@gsnLeftString =  " "
vcres@tiXAxisString =  " "
vcres@vcGlyphStyle   =  "CurlyVector" ;-- use curly vectors
vcres@vcRefMagnitudeF =  20          ;-- define vector ref magnitude
vcres@vcRefLengthF    =  0.045       ;-- define length of ref vector
vcres@vcRefAnnoOrthogonalPosF = -1.36 ;-- move reference vector into plot
vcres@vcRefAnnoString2On  =  False      ;-- don't draw string below ref vector

cont_plot      =  gsn_csm_contour_map_ce(wks,t,cnres)
vec_plot       =  gsn_csm_vector(wks,u,v,vcres)

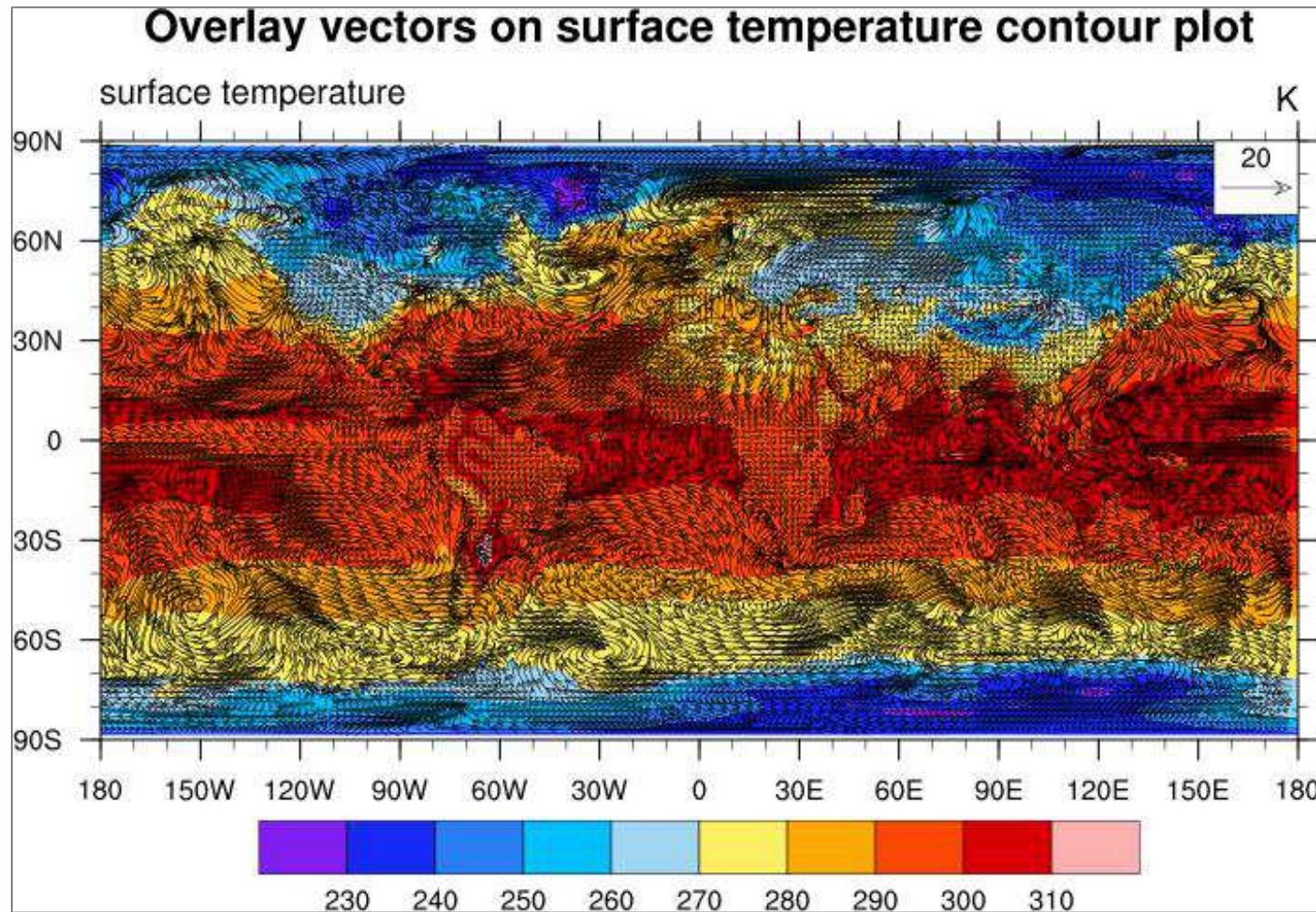
overlay(cont_plot, vec_plot)

draw(cont_plot)
frame(wks)

end

```

# Vector plot overlay on contour plot



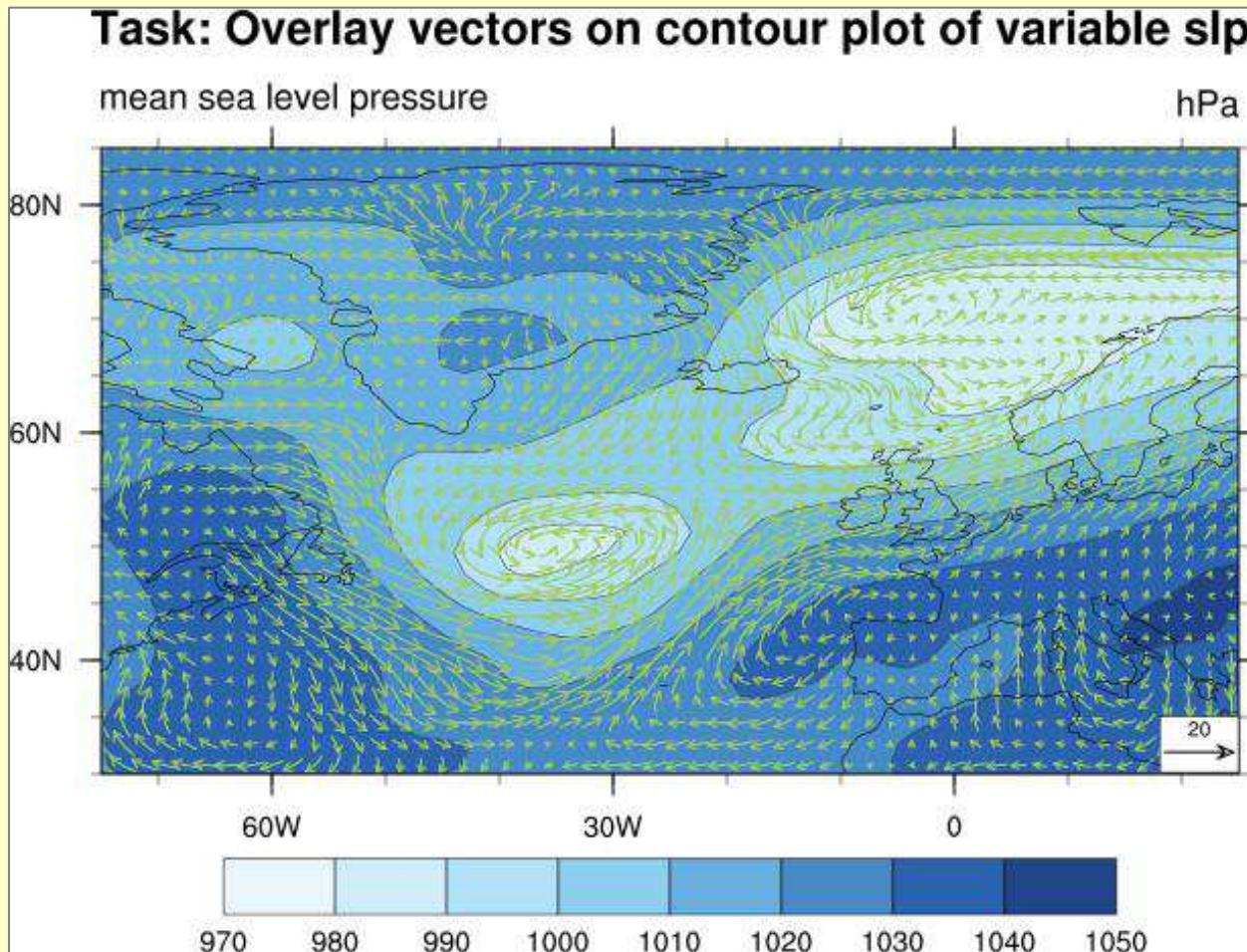
# Task: colorized vectors on contour plot

- Use the script **part\_III\_vector\_colorize\_overlay.ncl**
- Copy it to your own directory and modify it
  - Load **variable slp** instead of **tsurf**; devide it by **100** to get „**hPa**“ and change the **units** attribute
  - Use **colormap „WhiteBlue“** for contouring the **slp** variable
  - Select sub-region **lon** from **-75** to **25** and **lat** from **30** to **85** degrees
  - Don't draw the outer boxes of the labelbar
  - Set the **vector color** to „**darkolivegreen3**“
  - Set reference vector **magnitude** to **20** and **length** to **0.045**
  - Make **vectors thicker (=2.0)**
  - Move the reference vector annotation into the **lower right corner** of the plot
  - Plot the reference arrow in **black**
  - Don't draw the string below the reference vector annotation

## Hints:

- **gsn\_define\_colormap**
- **mpLimitMode, mpMinLonF,...**
- **cnLabelBarEndStyle**
- **vcRefAnnoOrthogonalPosF**
- **vcRefAnnoString2On**
- **vcRefAnnoArrowUseVecColor**
- **vcRefAnnoFontHeightF**

# Task: colorized vectors on contour plot



# Task: colorized vectors on contour plot (1/2)

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin

f      = addfile("../ECHAM5_OM_A1B_2001_0101-1001_2D.nc","r")
u      = f->u10(0,:,:)
v      = f->v10(0,:,:)
p      = f->slp(0,:,:)
p      = p/100
p@units = "hPa"
;-- set new units attribute

wks = gsn_open_wks("png","task_III_vectors_1")
gsn_define_colormap(wks,"WhiteBlue")      ;-- choose predefined colormap

;-- common resources for contour and vector plot
res          = True
res@gsnDraw   = False
;-- don't draw plot, yet
res@gsnFrame  = False
;-- don't advance frame
res@tiMainString = "Task: Overlay vectors on contour plot of variable slp"
```

# Task: colorized vectors on contour plot (2/2)

```
;-- set contour plot resources

cnres = res ;-- contour resources

cnres@cnFillOn = True ;-- turn on fill colors
cnres@cnLineThicknessF = 0.7 ;-- draw the contour lines thinner
cnres@cnLabelBarEndStyle = "ExcludeOuterBoxes" ;-- don't draw outer labelbar
;-- boxes

cnres@mpFillOn = False ;-- turn off map fill

cnres@mpLimitMode = "LatLon" ;-- limit mode
cnres@mpMinLonF = -75 ;-- minimum longitude
cnres@mpMaxLonF = 25 ;-- maximum longitude
cnres@mpMinLatF = 30 ;-- minimum latitude
cnres@mpMaxLatF = 85 ;-- maximum latitude
```

# Task: colorized vectors on contour plot (3/2)

```
;-- set the vector resources

vcres = res ;-- vector resources

vcres@gsnRightString = " " ;-- no right string
vcres@gsnLeftString = " " ;-- no left string

vcres@vcGlyphStyle = "CurlyVector" ;-- use curly vectors
vcres@vcLineArrowColor = "darkolivegreen3" ;-- change vector color to white
vcres@vcLineArrowThicknessF = 2.0 ;-- change vector thickness

vcres@vcRefMagnitudeF = 20 ;-- define vector reference magnitude
vcres@vcRefLengthF = 0.045 ;-- define length of reference vector
vcres@vcRefAnnoOrthogonalPosF = -0.452 ;-- move reference vector into plot
vcres@vcRefAnnoString2On = False ;-- don't draw string below ref vector
vcres@vcRefAnnoArrowUseVecColor = False ;-- don't use vcLineArrowColor for ref
vcres@vcRefAnnoFontHeightF = 0.008 ;-- font size of reference vector
;-- annotation
```

## Task: colorized vectors on contour plot (4/2)

```
;-- create the plots, but don't draw them

cont_plot = gsn_csm_contour_map_ce(wks,p,cnres)
vec_plot = gsn_csm_vector(wks,u,v,vcres)

;-- overlay vec_plot on cont_plot, but don't draw them

overlay(cont_plot, vec_plot)

;-- now, draw the plot and advance frame

draw(cont_plot)
frame(wks)

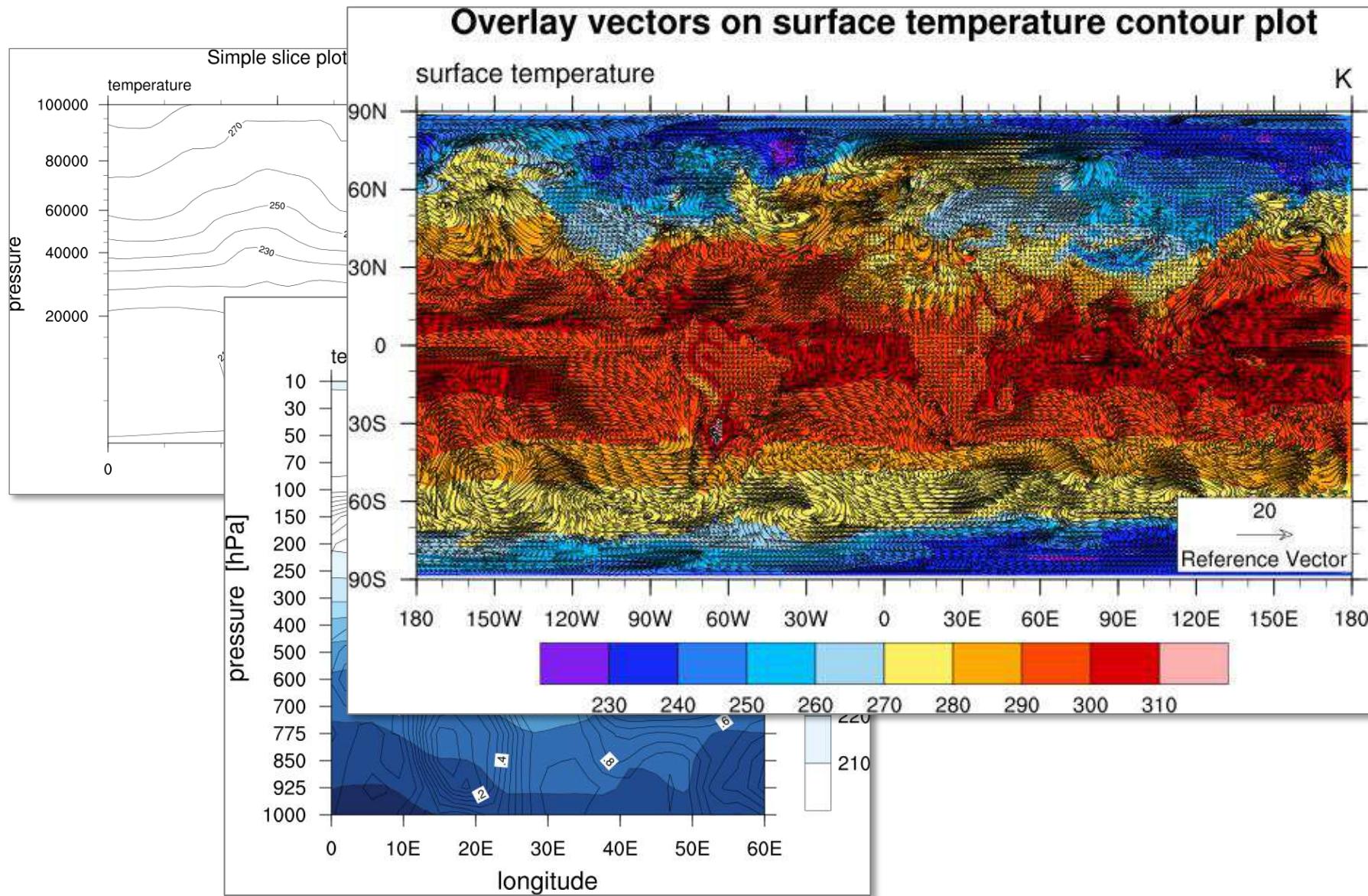
end
```

Part IV

## Slices and Overlays

Exercises and Tasks

# Slices and Overlays



# Simple slice plot

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin

f      = addfile("../ECHAM5_OM_A1B_2001_timestep1_3D.nc", "r")
var   = f->t(0,:,{40},{0:60})          ;-- first time step, latitude=40N,
                                              ;-- longitude=0-60E.
lon_t = f->lon({0:60})                ;-- longitude=0-60E
lev_t = f->lev                         ;-- currently 17 levels

wks = gsn_open_wks("png","part_IV_simple_slice")

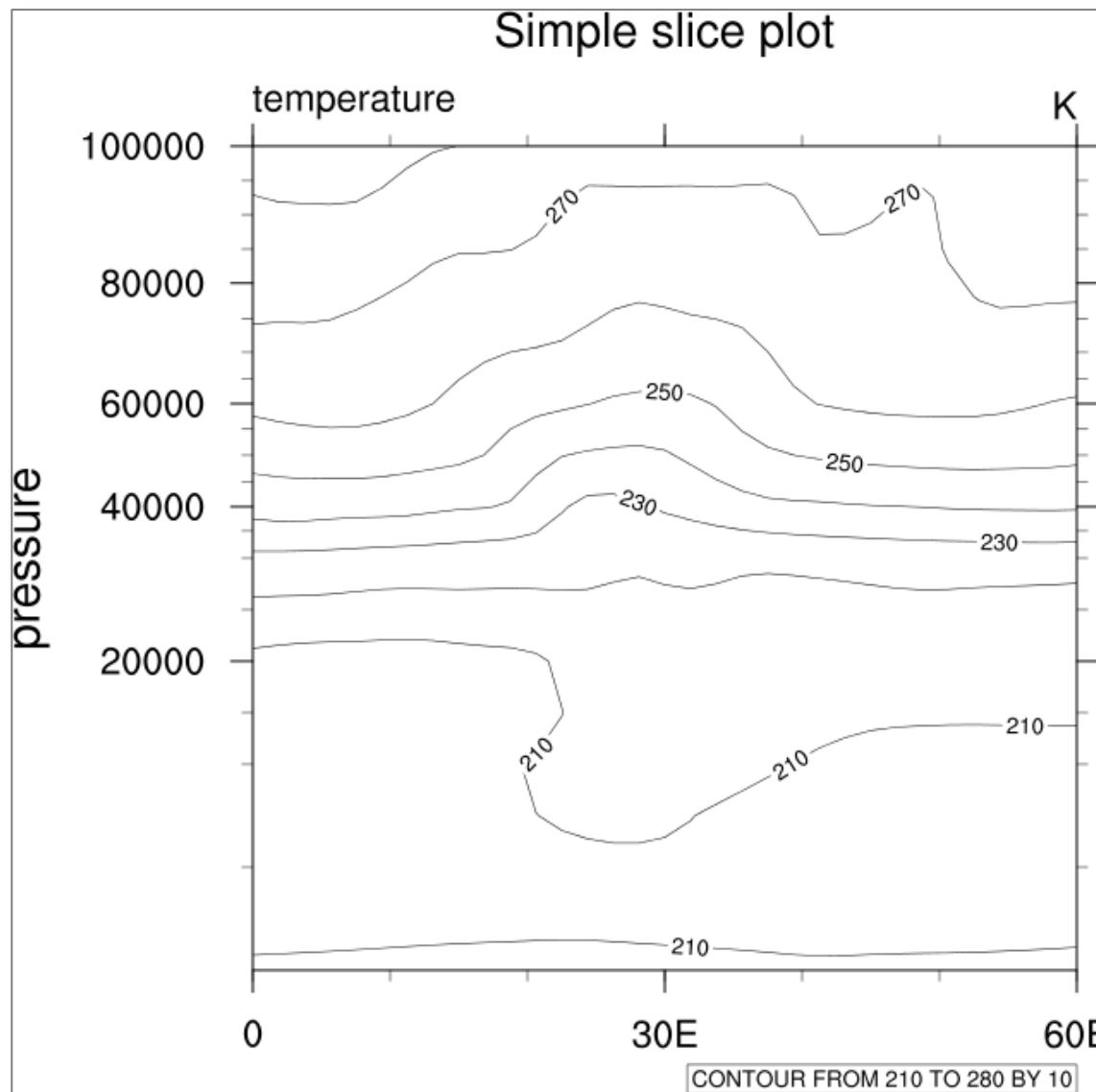
res           = True
res@tiMainString = "Simple slice plot" ;-- draw title

res@sfxArray    = lon_t      ;-- uses lon_t as plot x-axis
res@sfyArray    = lev_t      ;-- uses lev_t in hPa as plot y-axis

plot = gsn_csm_contour(wks,var,res)

end
```

# Simple slice plot



# Slice plot with contour line overlay (1/3)

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f      = addfile("../..//ECHAM5_OM_A1B_2001_timestep1_3D.nc", "r")

;-- variables: first time step, latitude=40N, longitude=0-60E
  t      = f->t(0,:,:{40},{0:60})
  rhum   = f->rhumidity(0,:,:{40},{0:60})
  lon_t  = f->lon({0:60})                      ;-- select longitude=0-60E
  lev_t  = f->lev/100                           ;-- convert to hPa units

  llables = new(dimsizes(lev_t),string)          ;-- assign array for level labels
  llables = lev_t                                ;-- define level labels

  wks = gsn_open_wks("png","part_IV_slice_rhumidity_overlay")
  gsn_define_colormap(wks,"WhiteBlue")           ;-- set the colormap to be used

  res           = True
  res@gsnDraw    = False                         ;-- don't draw the plot
  res@gsnFrame   = False                         ;-- don't advance the plot
  res@gsnSpreadColors = True                     ;-- use full colormap
```

# Slice plot with contour line overlay (2/3)

```
res@cnFillOn      =  True           ;-- turn on color fill
res@cnLineLabelsOn =  False          ;-- turns off contour line labels
res@cnInfoLabelOn =  False          ;-- turns off contour info label

res@lbOrientation   = "vertical"    ;-- vertical label bar

res@sfxArray        =  lon_t          ;-- uses lon_t as plot x-axis
res@sfyArray        =  lev_t          ;-- uses lev_t in hPa as plot y-axis

res@tryReverse     =  True           ;-- reverses y-axis

res@tmXBTickSpacingF =  10.          ;-- label x-axis every 10 degrees
res@tmYILMode       = "Explicit"    ;-- set y-axis labeling to explicit
res@tmYLValues      =  lev_t          ;-- values for y-axis tickmarks
res@tmYILLabels     =  llabels         ;-- set labels equal to values (type string)

res@tiXAxisString   =  f->lon@long_name      ;-- draw y-axis title
res@tiYAxisString   =  f->lev@long_name + " [hPa]" ;-- draw y-axis title

res@tiMainString    = "Slice at latitude = 40~S~o~N~N~C~~z70~contour lines: "+\
                      rhum@long_name
```

# Slice plot with contour line overlay (3/3)

```
plot1 = gsn_csm_contour(wks,t,res)

res@cnFillOn      = False           ;-- turn off color fill
res@cnLineLabelsOn = True            ;-- turns on contour line labels
res@gsnLeftString = ""              ;-- don't draw left string

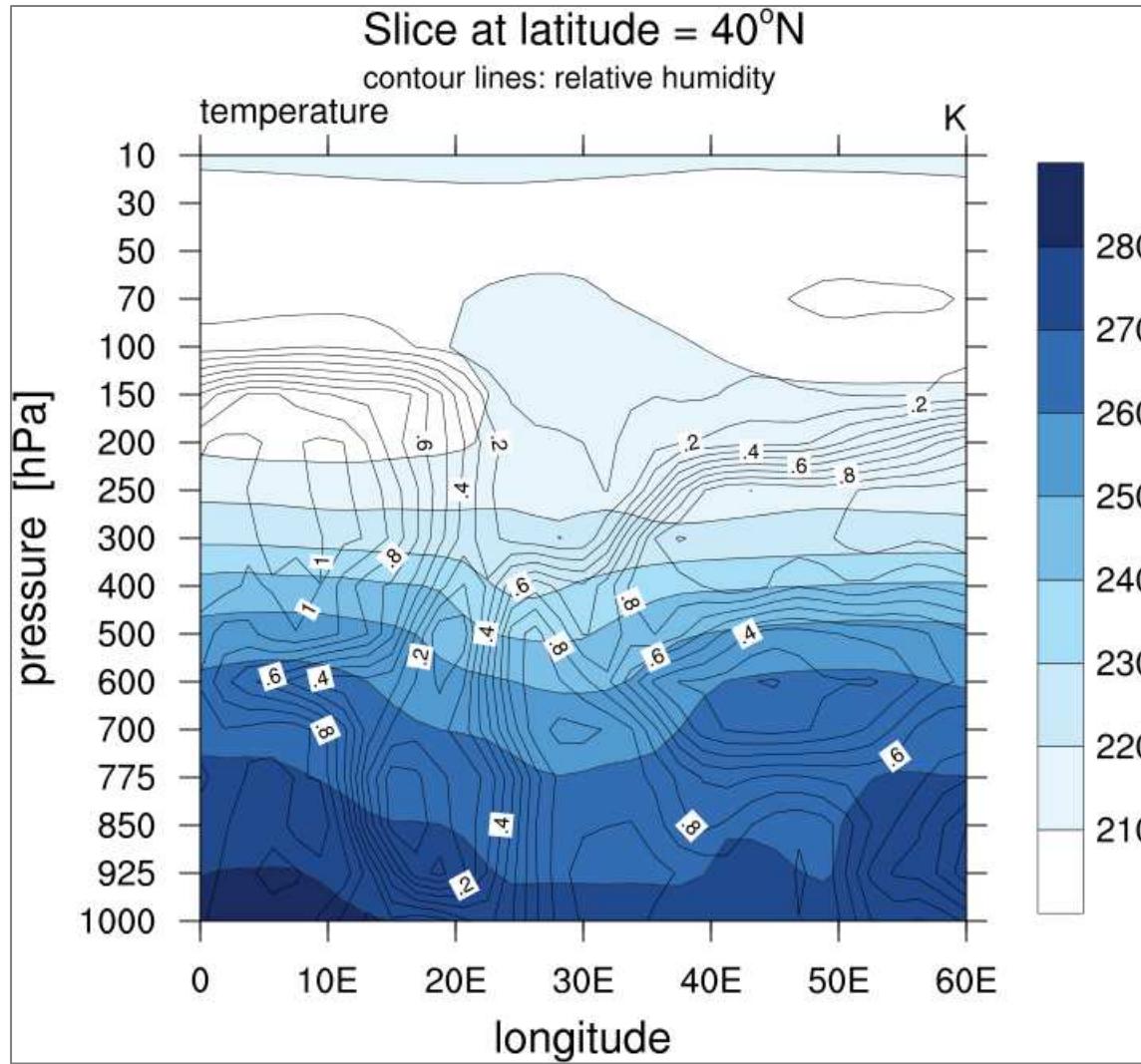
plot2 = gsn_csm_contour(wks,rhum,res)

overlay(plot1, plot2)

draw(plot1)
frame(wks)

end
```

## Slice plot with contour line overlay (3/3)



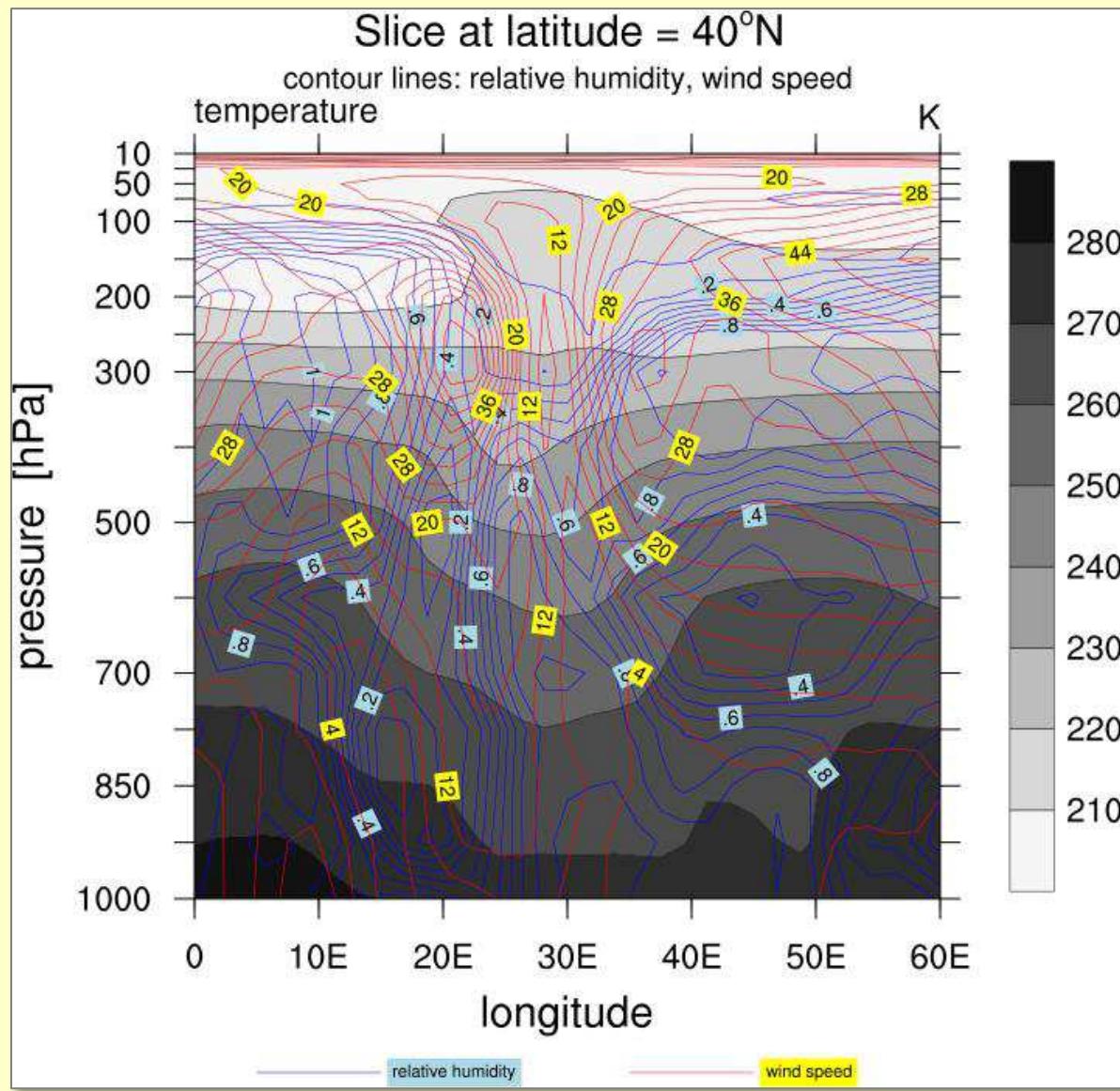
# Task: overlay contour lines of rhum and var3

- Use the script **part\_IV\_slice\_rhumidity\_overlay.ncl**
- Copy it to your own directory and modify it
  - Load **variable var3** (wind speed)
  - Use **colormap „PalGrey“** and **reverse** it for contouring the **t** variable
  - Draw the variable **var3** with color **„red“**
  - Draw the **line label** of **var3** with **background color „yellow“**
  - Draw the variable **rhum** with the color **„blue“**
  - Draw the **line label** of **rhum** with **background color „lightblue“**
  - Draw a **legend** for the contour lines at the bottom

## Hints:

- **gsn\_reverse\_colormap**
- **cnLineColor**
- **cnLineLabelBackgroundColor**
- **gsLineColor**
- **txBackgroundColor**
- **overlay**

# Task: overlay contour lines of rhum and var3



# Task: overlay contour lines of rhum and var3 (1/5)

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f = addfile("../ECHAM5_OM_A1B_2001_timestep1_3D.nc", "r")
  t = f->t(0,:,:,{40},{0:60})      ;-- first time step, latitude=40N, longitude=0-60E
  rhum           = f->rhumidity(0,:,:,{40},{0:60})
  rhum@long_name = f->rhumidity@long_name          ;-- set long_name attribute
  var3           = f->var3(0,:,:,{40},{0:60})
  var3@long_name = "wind speed"                   ;-- set long_name attribute
  lon_t          = f->lon({0:60})                  ;-- longitude=0-60E
  lev_t          = f->lev/100                     ;-- convert to hPa units

  ;-- create levels array for y-axis labeling
  llables = new(dimsizes(lev_t),string)           ;-- assign new array for labels
  llables = lev_t                                 ;-- llables of type string

  ;-- define workstation
  wks = gsn_open_wks("png","task_IV_slice_wind_speed_overlay")
  gsn_define_colormap(wks,"NCL_PalGrey")          ;-- set the colormap to be used
  gsn_reverse_colormap(wks)
```

# Task: overlay contour lines of rhum and var3 (2/5)

```

res                      = True
res@gsnDraw              = False      ;-- don't draw the plot
res@gsnFrame              = False      ;-- don't advance the frame
res@gsnSpreadColors       = True       ;-- use full colormap
res@gsnSpreadColorStart   = 4          ;-- start color index
                                ;--(! reversed colormap don't reverse Start)
res@gsnSpreadColorEnd     = 80         ;-- end color index
                                ;--(! reversed colormap don't reverse End)

res@cnFillOn              = True       ;-- turn on color fill
res@cnLineLabelsOn        = False      ;-- turns off contour line labels
res@cnInfoLabelOn         = False      ;-- turns off contour info label

res@lbOrientation          = "vertical" ;-- vertical label bar

res@sfxArray               = lon_t      ;-- uses lon_t as plot x-axis
res@sfyArray               = lev_t      ;-- uses lev_t in hPa as plot y-axis

res@gsnYAxisIrregular2Linear = True      ;-- converts y-axis irreg. to linear
res@trYReverse              = True      ;-- reverses y-axis

```

# Task: overlay contour lines of rhum and var3 (3/5)

```

res@tmXBTickSpacingF      = 10.                      ;-- label x-axis every 10 degrees
res@tmYILMode              = "Explicit"            ;-- set y-axis labeling to explicit
res@tmYLValues             = lev_t                  ;-- values for y-axis tickmarks
res@tmYLLabels              = llables                ;-- set labels equal to values
res@tmYLLabelStride         = 2                     ;-- draw every 5th label

res@tiXAxisString          = f->lon@long_name     ;-- draw y-axis title
res@tiYAxisString          = f->lev@long_name+" [hPa]" ;-- draw y-axis title

res@tiMainString = "           Slice at latitude = 40~S~o~N~N~C~~Z70~contour lines:
"+rhum@long_name+", "+var3@long_name

plot1 = gsn_csm_contour(wks,t,res)      ;-- generate plot1, but don't draw it yet

res@cnFillOn                = False                 ;-- turn off color fill
res@cnLineLabelsOn           = True                 ;-- turns on contour line labels
res@gsnLeftString            = ""                   ;-- don't draw left string
res@cnLineColor               = "blue"                ;-- contour line color
res@cnLineLabelBackgroundColor = "lightblue"        ;-- cont. line label backgr. Color

plot2 = gsn_csm_contour(wks,rhum,res)    ;-- generate plot2, but don't draw it yet

```

# Task: overlay contour lines of rhum and var3 (4/5)

```

res@cnLineColor          = "red"           ;-- contour line color
res@cnLineLabelBackgroundColor = "yellow" ;-- cont. line label backgr. Color

plot3 = gsn_csm_contour(wks,var3,res)    ;-- generate plot2, but don't draw it yet

;-- set legend x,y-position   x1: line and string 1      x2: line and string 2
x  = 0.25                           ;-- legend start x-position
x1 = (/x,x+0.1/)                   ;-- legend line 1 x-position
x2 = (/x1(1)+0.2,x1(1)+0.3/)     ;-- legend line 2 x-position
y  = 0.06                           ;-- legend y-position at bottom
y1 = (/y,y/)

;-- legend text and polyline resources
txres                      = True        ;-- text resource object
txres@txFontHeightF        = 0.010      ;-- set legend text font size
txres@txJust                = "CenterLeft" ;-- text justification
txres@txBackgroundFillColor = "lightblue";-- legend string 2 background color

plres                      = True        ;-- polyline resource object
plres@gSLineColor           = "blue"     ;-- polyline color

```

## Task: overlay contour lines of rhum and var3 (5/5)

```

gsn_polyline_ndc(wks,x1,y1,plres) ;-- draw legend line 1
gsn_text_ndc(wks,rhum@long_name,x1(1)+0.01, y, txres) ;-- draw legend string 1

txres@txBackgroundColor = "yellow" ;-- legend string 2 background color
plres@gsLineColor = "red" ;-- polyline color

gsn_polyline_ndc(wks,x2,y1,plres) ;-- draw legend line 2
gsn_text_ndc(wks,var3@long_name,x2(1)+0.01, y, txres) ;-- draw legend string 2

overlay(plot1, plot2) ;-- overlay plot2 on plot1
overlay(plot1, plot3) ;-- overlay plot3 on plot1

draw(plot1) ;-- draw the complete plot
frame(wks) ;-- advance the frame

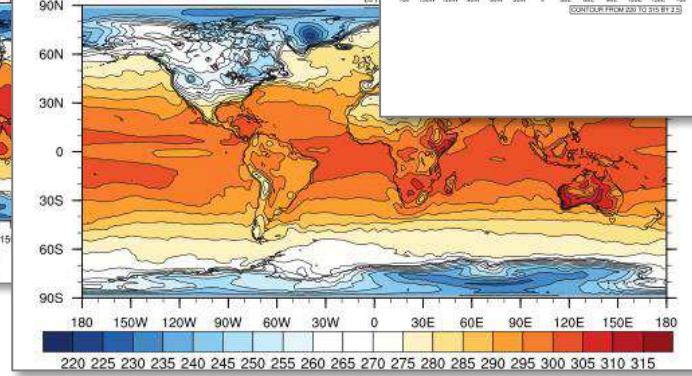
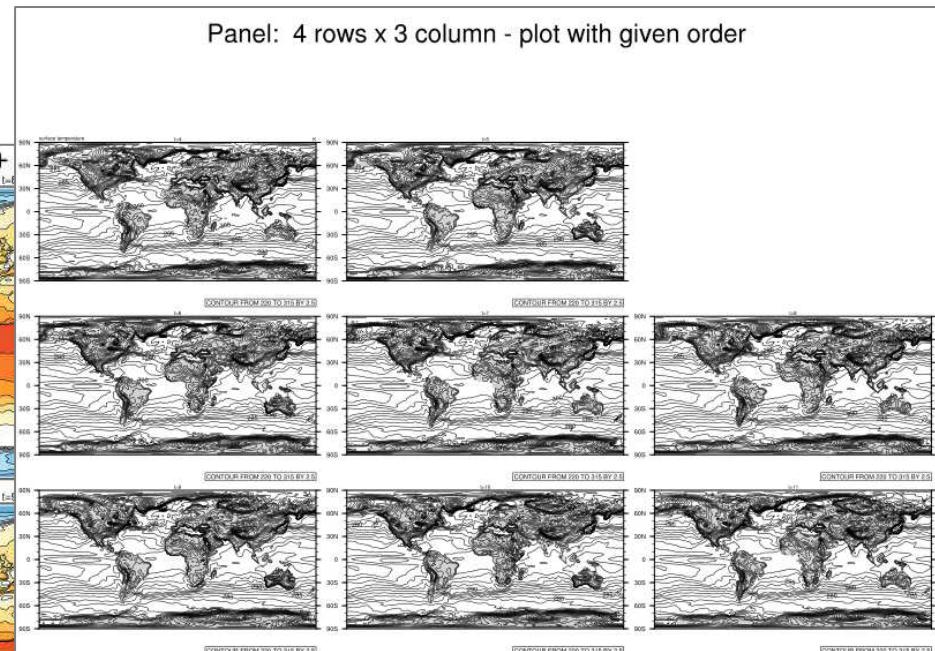
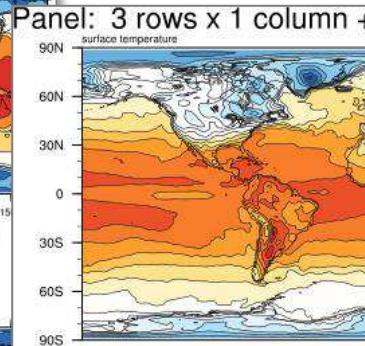
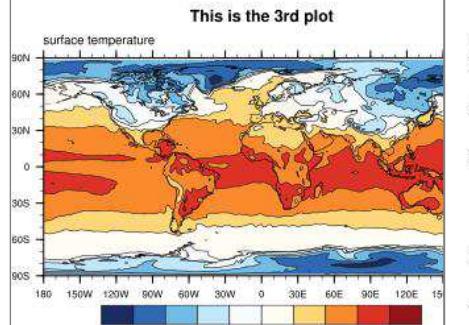
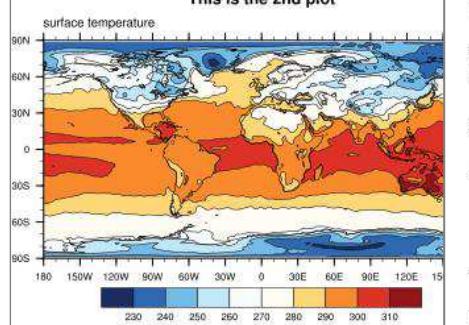
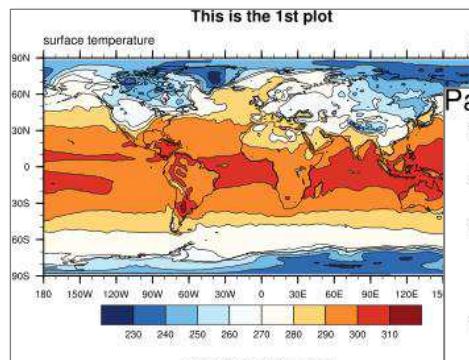
```

Part V

## Panel Plots

Exercises and Tasks

# Panel Plots



# Simple panel plot (1/2)

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin

f    = addfile("$NCL_TUT/data/ECHAM5_OM_A1B_2001_0101-1001_2D.nc", "r")
var = f->tsurf

wks = gsn_open_wks("png" , "part_V_simple_panel")

res           = True
res@gsnDraw   = False      ;-- don't draw the plot - yet
res@gsnFrame  = False      ;-- don't advance the frame
res@gsnAddCyclic = True     ;-- add cyclic point
res@cnFillOn  = True       ;-- enable color filled contours
res@cnLineLabelsOn = False  ;-- don't draw labels on contour lines

plot = new(3, "graphic")          ;-- assign empty plot array with 2 elements

res@tiMainString = "This is the 1st plot" ;-- draw a title on top of 1. plot

plot(0) = gsn_csm_contour_map_ce(wks,var(0,:,:),res) ;-- create the plot 0
```

# Simple panel plot (2/2)

```
res@tiMainString = "This is the 2nd plot"      ;-- draw a title on top of 2. plot

plot(1) = gsn_csm_contour_map_ce(wks,var(9,:,:),res)    ;-- create the plot

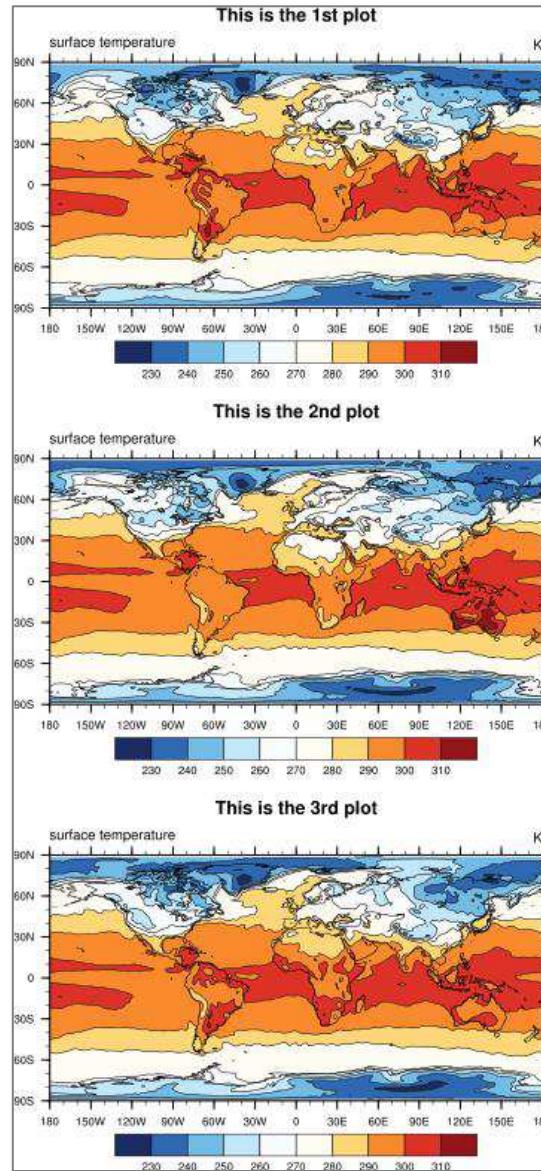
res@tiMainString = "This is the 3rd plot"      ;-- draw a title on top of 3rd plot

plot(2) = gsn_csm_contour_map_ce(wks,var(19,:,:),res)    ;-- create the plot

;-- create and plot the panel plot:      rows = 3 ; columns = 1
gsn_panel(wks,plot,(/3,1/),False)

end
```

# Simple panel plot (2/2)



# Panel plot with common labelbar (1/2)

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin

f    = addfile("$NCL_TUT/data/ECHAM5_OM_A1B_2001_0101-1001_2D.nc", "r")
var = f->tsurf

wks = gsn_open_wks("png", "part_V_panel_attachment_one_labelbar")

res                         = True
res@gsnDraw                  = False           ;-- don't draw the plot - yet
res@gsnFrame                 = False           ;-- don't advance the frame
res@gsnAddCyclic             = True            ;-- add cyclic point

res@cnFillOn                 = True            ;-- enable color filled contours
res@cnLineLabelsOn           = False           ;-- don't draw labels on contour lines
res@cnMinLevelValF           = 220             ;-- contour level minimum
res@cnMaxLevelValF           = 315             ;-- contour level maximum
res@cnLevelSpacingF          = 5                ;-- contour level interval
res@cnLineLabelsOn           = False           ;-- don't draw labels on contour lines
res@cnInfoLabelOn            = False           ;-- don't draw labels on contour lines
```

# Panel plot with common labelbar (2/2)

```
res@lbLabelBarOn          = False      ;-- don't draw a labelbar below each plot

;-- create empty graphic plot array
nplots = 3
plot = new(nplots,"graphic")

;-- create the plots
do i=0,nplots-1
  res@gsnCenterString    = "t=" + (i+8)  ;-- draw center string
  plot(i) = gsn_csm_contour_map_ce(wks,var((i+8),:,:),res)  ;-- create plots
end do

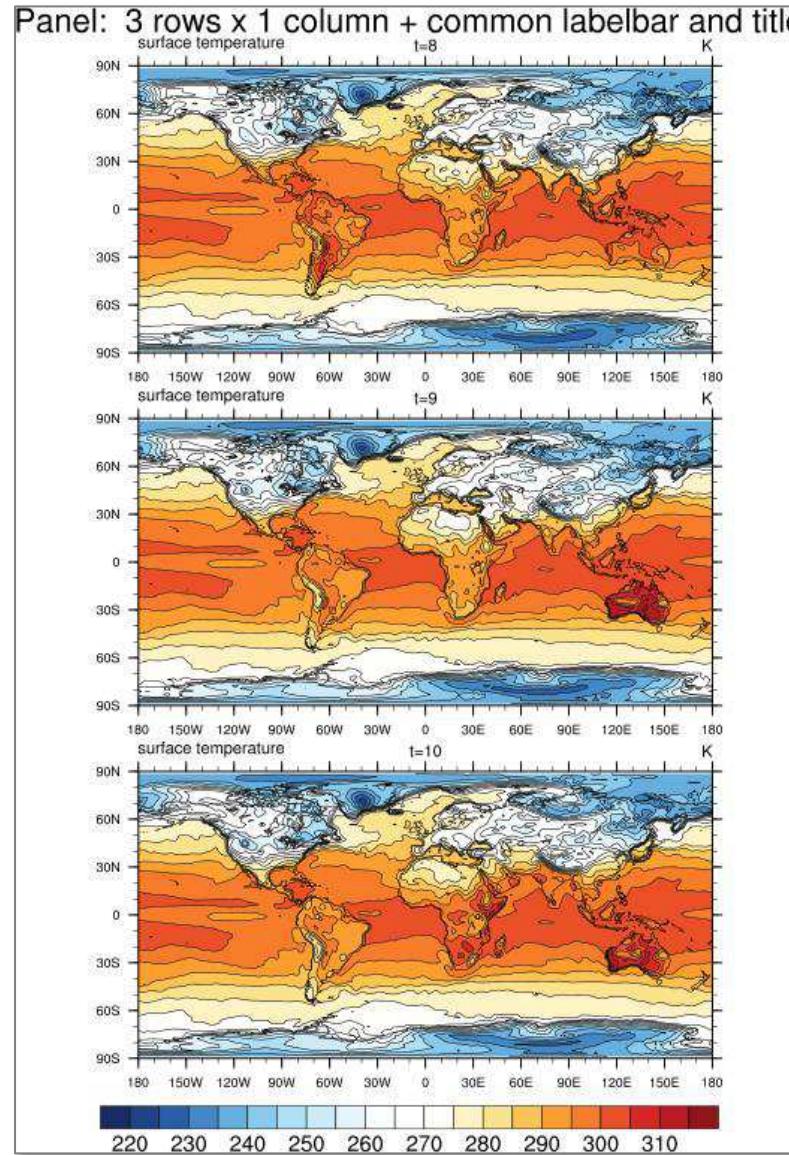
;-- panel resources
pres                      = True
pres@gsnPanelTop          = 0.96       ;-- panel top position (y-value)
pres@gsnPanelBottom        = 0.012      ;-- panel bottom position (y-value)
pres@gsnPanelLabelBar      = True       ;-- draw a common labelbar

pres@pmLabelBarOrthogonalPosF = 0.0      ;-- don't move labelbar
```

# Panel plot with common labelbar (3/2)

```
pres@txString          = "Panel:  3 rows x 1 column + common legend and title"  
                           ;-- draw a common title on top  
pres@txFontHeightF    = 0.020      ;-- text font size  
pres@txPosXF           = 0.5        ;-- text x-position  
pres@txPosYF           = 0.97       ;-- text y-position  
pres@txJust             = "CenterCenter" ;-- text justification  
  
gsn_panel(wks,plot,/3,1/),False)      ;-- draw the panel  
  
end
```

# Panel plot with common labelbar (3/2)



# Task: panel plot 3 rows x 3 columns

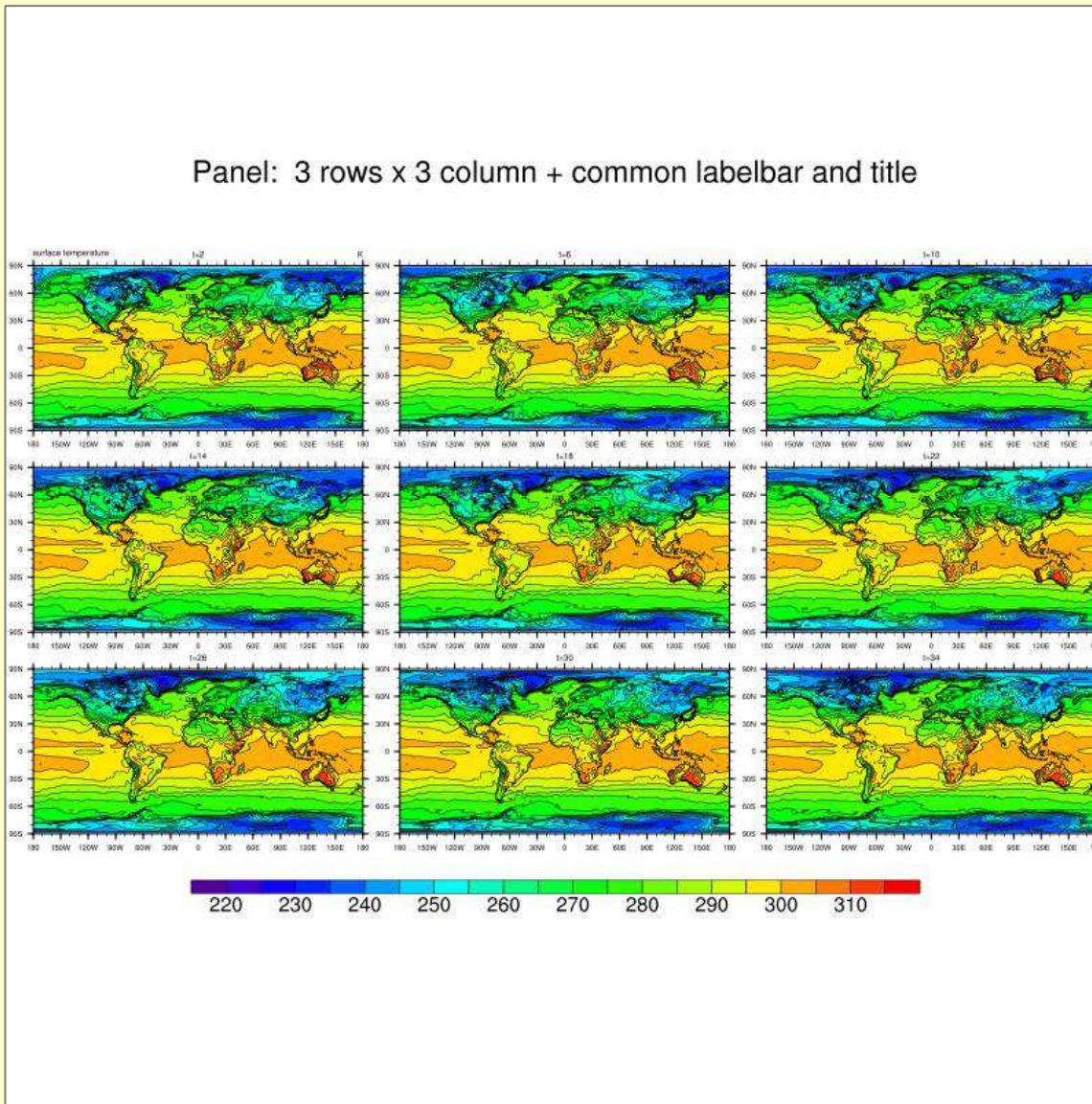
- Use the script **part\_V\_simple\_panel.ncl**
- Copy it to your own directory and modify it
  - Use colormap "rainbow" all plots have contour levels from 220 to 315 K in 5 degrees steps
  - Create plots of time indexes **2,6,10,14,18,22,26,30,34**
  - Create panel plot with **3 rows and 3 columns**
  - Draw a **common labelbar** and move it downward

## Hints:

*See also part\_V\_panel\_attachment\_one\_labelbar.ncl*

- **cnLevelSelectionMode**
- **cnMinLevelValF / cnMaxLevelValF**
- **cnLevelSpacingF**
- **lbLabelBarOn**
- **gsnPanelLabelBar**
- **gsnPanelTop / gsnPanelBottom**

# Task: panel plot 3 rows x 3 columns



# Task: overlay contour lines of rhum and var3 (1/5)

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f    = addfile("$NCL_TUT/data/ECHAM5_OM_A1B_2001_0101-1001_2D.nc","r")
  var = f->tsurf

  wks = gsn_open_wks("png","tasks_V_panel_3x3_one_labelbar")

  res           = True
  res@gsnDraw   = False          ;-- don't draw the plot - yet
  res@gsnFrame  = False          ;-- don't advance the frame
  res@gsnAddCyclic = True        ;-- add cyclic point
  res@gsnStringFontHeightF = 0.015 ;-- set string font size

  res@cnFillOn      = True        ;-- enable color filled contours
  res@cnFillPalette = "rainbow"   ;-- choose a colormap
  res@cnLevelSelectionMode = "ManualLevels" ;-- set manual contour levels, so that
                                              ;-- all plots have the same labelbar colors/values
  res@cnMinLevelValF = 220         ;-- contour level minimum
  res@cnMaxLevelValF = 315         ;-- contour level maximum
  res@cnLevelSpacingF = 5           ;-- contour level interval
```

# Task: overlay contour lines of rhum and var3 (1/5)

```

res@cnLineLabelsOn      = False           ;-- don't draw labels on contour lines
res@cnInfoLabelOn       = False           ;-- don't draw labels on contour lines

res@lbLabelBarOn        = False           ;-- don't draw a labelbar below each plot

;-- create empty graphic plot array
nplots = 9
plot = new(nplots, "graphic")

;-- create the plots
do i=0,nplots-1
  if (i.ne.0) then
    res@gsnLeftString     = ""            ;-- don't draw string tas@long_name on
                                         ;-- the top left except for plot 0
    res@gsnRightString    = ""            ;-- don't draw string tas@units on
                                         ;-- top right except for plot 0
  end if
  m=2+(i*4)                      ;-- select every 4th time step, start t=2
  res@gsnCenterString         = "t=" + m
  plot(i) = gsn_csm_contour_map_ce(wks,var(m,:,:,:),res) ;-- create the plots and
                                         ;-- write it to the plot array
end do

```

# Task: overlay contour lines of rhum and var3 (1/5)

```
;-- panel resources
pres          = True
pres@gsnPanelTop      = 0.96           ;-- panel top position (y-value)
pres@gsnPanelBottom    = 0.012          ;-- panel bottom position (y-value)
pres@gsnPanelLabelBar   = True           ;-- draw a common labelbar

pres@pmLabelBarOrthogonalPosF = -0.02      ;-- move labelbar downward

pres@txString          = "Panel: 3 rows x 3 column + common labelbar and title"
                                         ;-- draw a common title on top
pres@txFontHeightF      = 0.020          ;-- text font size
pres@txPosXF            = 0.5             ;-- text x-position
pres@txPosYF            = 0.85            ;-- text y-position
pres@txJust              = "CenterCenter" ;-- text justification

gsn_panel(wks,plot,(/3,3/),pres)

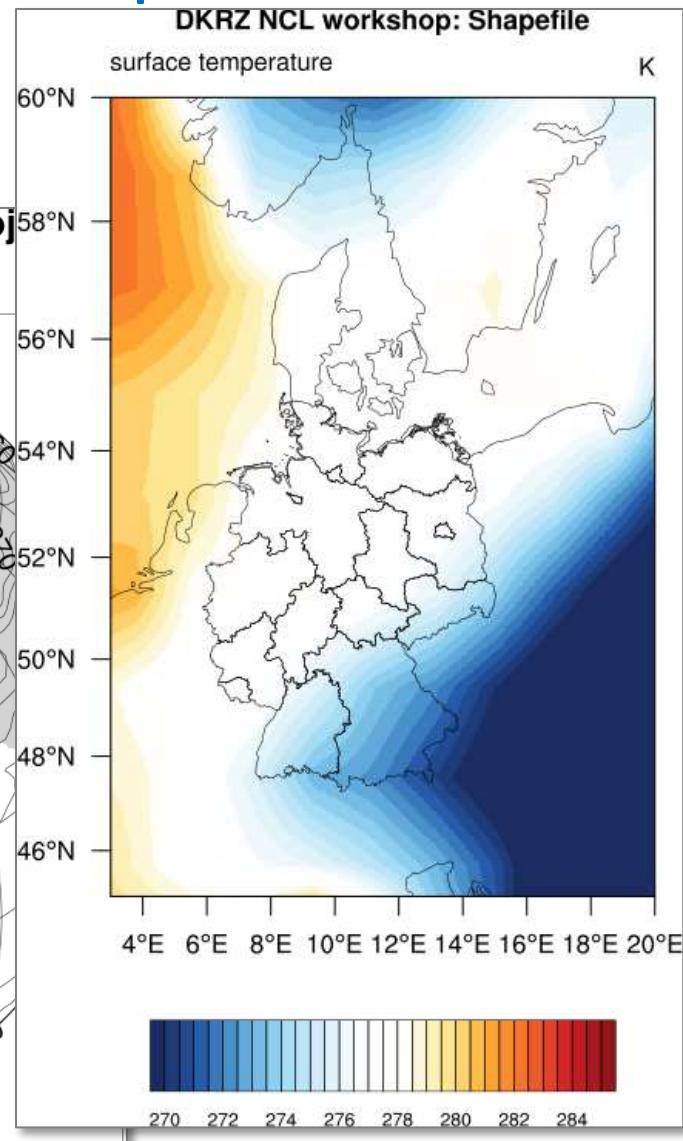
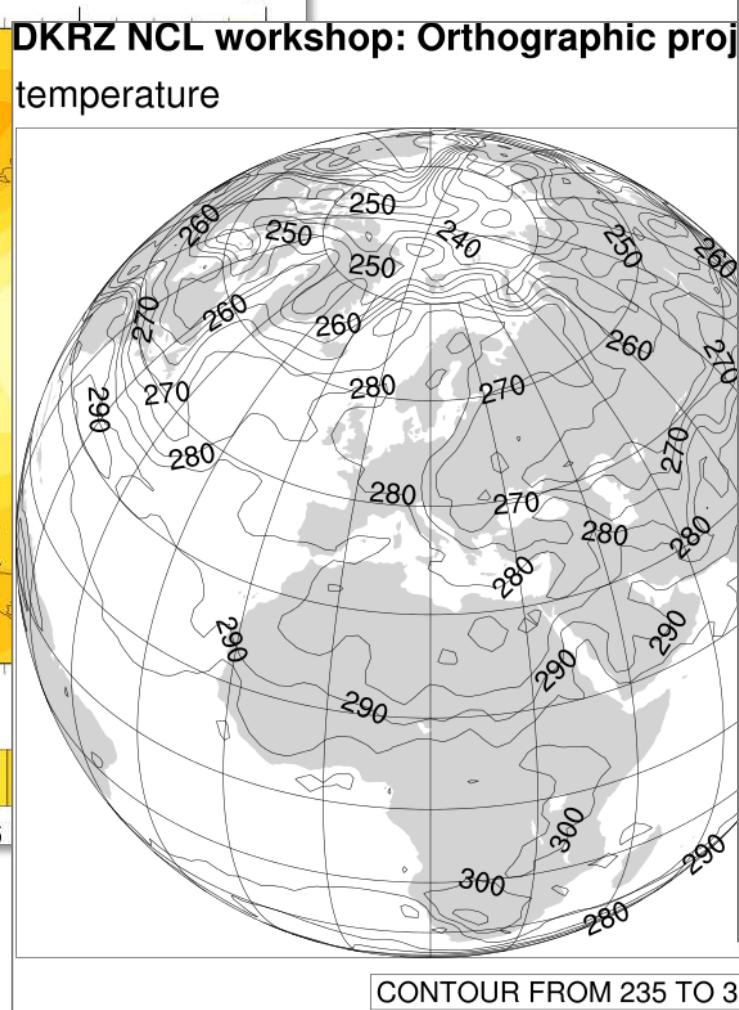
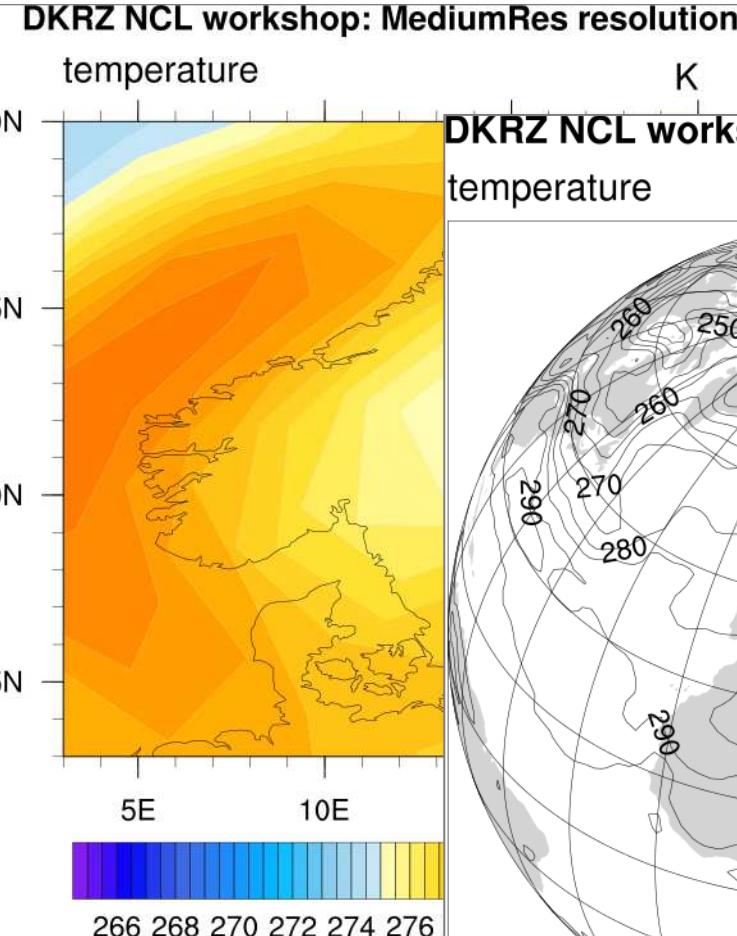
end
```

# Part VI

## Map Resolutions, Projections and Shapefiles

Exercises and Tasks

# Map resolution, projection and shapefiles



# Map Resolution (1/2)

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  file1 = addfile("$NCL_TUT/data/ECHAM5_OM_A1B_2001_timestep1_3D.nc","r")
  var   = file1->t(0,0,:,:)

  wks = gsn_open_wks("png","part_VI_map_resolution")

  res           = True
  res@gsnMaximize = True          ;-- maximize plot output

  res@cnFillOn      = True          ;-- turn on contour fill
  res@cnFillMode    = "RasterFill"  ;-- contour cell fill modes
  res@cnRasterSmoothingOn = True    ;-- contour smoothing on
  res@cnFillPalette = "ncl_default";-- choose a colormap
  res@cnLinesOn     = False         ;-- turn off contour lines
  res@cnLineLabelsOn = False        ;-- turn off line labels
  res@cnLevelSelectionMode = "ManualLevels";-- set contour levels manually
  res@cnMinLevelValF = 265.         ;-- minimum contour level
  res@cnMaxLevelValF = 285.         ;-- maximum contour level
  res@cnLevelSpacingF = 0.5          ;-- contour level spacing
```

# Map Resolution (2/2)

```
res@mpMinLonF          = 3.           ;-- min lon
res@mpMaxLonF          = 20.          ;-- max lon
res@mpMinLatF          = 53.          ;-- min lat
res@mpMaxLatF          = 70.          ;-- max lat

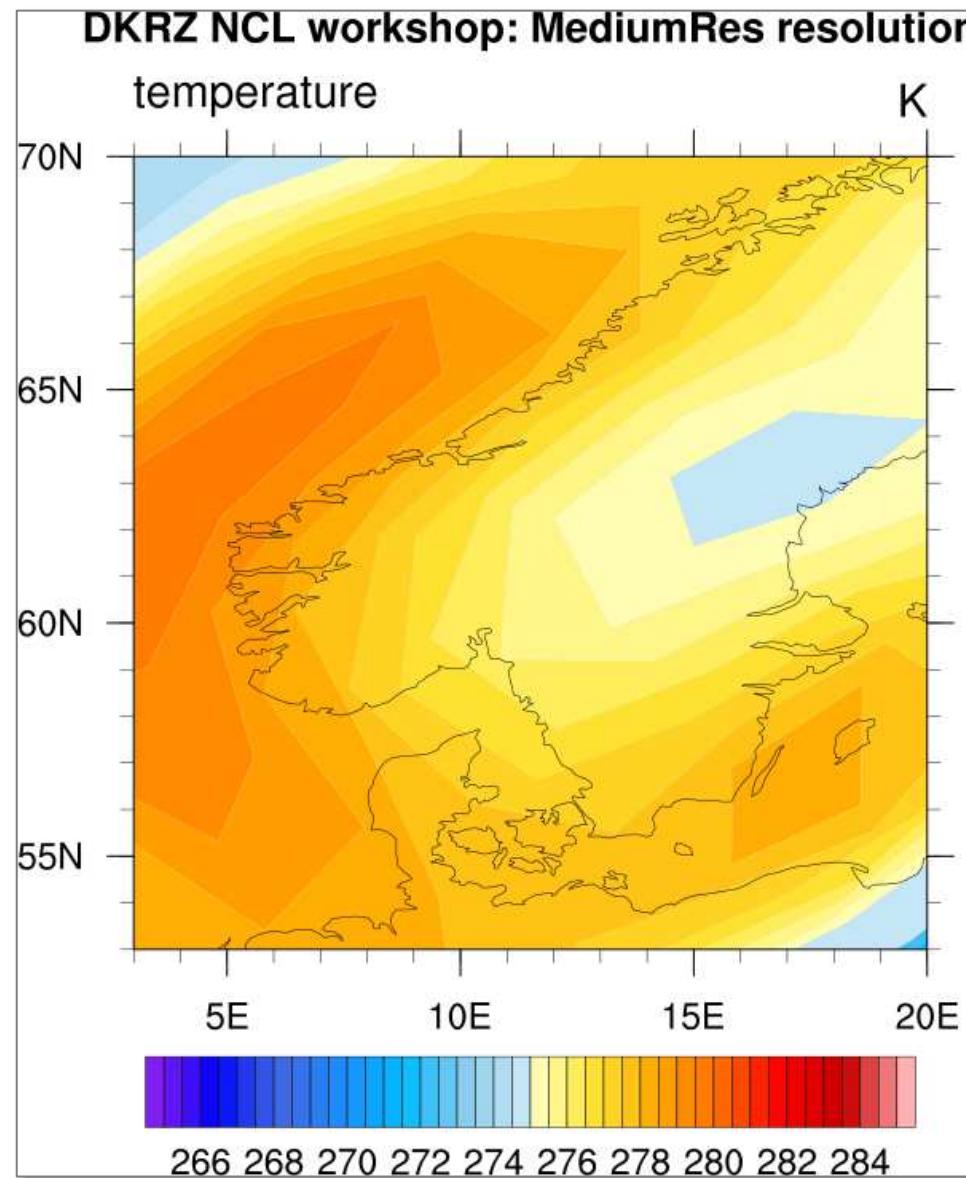
res@mpDataBaseVersion   = "MediumRes" ;-- choose map data base version

res@tiMainString        = "DKRZ NCL workshop: MediumRes resolution"
res@tiMainFontHeightF   = 0.02

plot = gsn_csm_contour_map(wks, var, res)

end
```

# Map Resolution (2/2)



# Map Projection

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f      = addfile("$NCL_TUT/data/ECHAM5_OM_A1B_2001_timestep1_3D.nc", "r")
  var   = f->t(0,0,:,:)

  wks = gsn_open_wks("png", "part_VI_map_projection")

  res           = True
  res@gsnMaximize = True          ;-- maximize plot output

  res@mpProjection     = "Orthographic" ;-- choose map projection
  res@mpCenterLonF     = 15           ;-- center plot on lon value
  res@mpCenterLatF     = 40           ;-- center plot on lat value
  res@mpGridAndLimbOn  = True          ;-- draw grid and limb lines

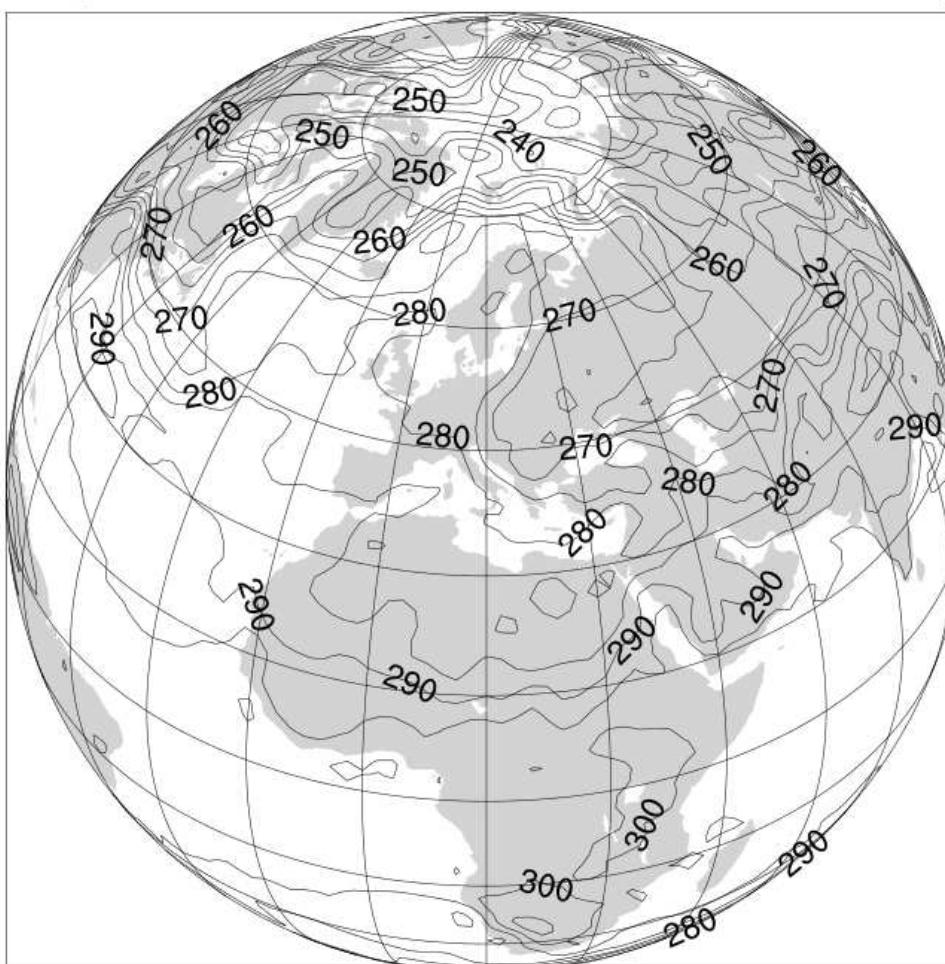
  res@tiMainString     = "DKRZ NCL workshop: Orthographic projection"
  res@tiMainFontHeightF = 0.02

  plot = gsn_csm_contour_map(wks, var, res)
end
```

# Map Projection

## DKRZ NCL workshop: Orthographic projection

temperature



CONTOUR FROM 235 TO 310 BY 5

# Shapefiles – outline counties of Germany (3/3)

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f      = addfile("$NCL_TUT/data/ECHAM5_OM_A1B_2001_0101-1001_2D.nc", "r")
  var   = f->tsurf(0,:,:)

  wks = gsn_open_wks("png", "part_VI_shapefile")

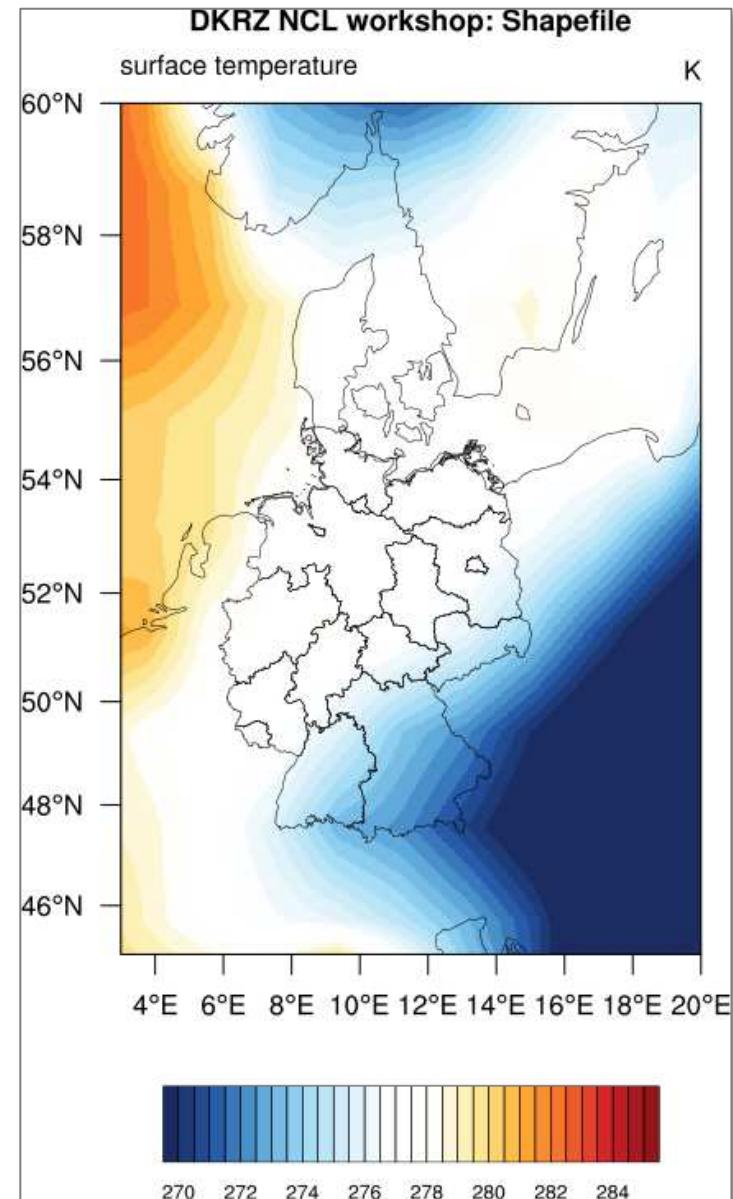
  res           = True
  res@gsnDraw    = False          ;-- don't draw the plot
  res@gsnFrame   = False          ;-- don't advance frame yet
  res@gsnMaximize = True          ;-- maximize plot in frame

  res@cnFillOn    = True          ;-- turn on countour fill
  res@cnFillMode   = "RasterFill" ;-- contour cell fill mode
  res@cnRasterSmoothingOn = True ;-- contour smoothing on
  res@cnLinesOn    = False         ;-- don't draw contour lines
  res@cnLevelSelectionMode = "ManualLevels" ;-- set manual contour levels
  res@cnMinLevelValF  = 270.0      ;-- minimum contour value
  res@cnMaxLevelValF  = 285.0      ;-- maximum contour value
  res@cnLevelSpacingF = 0.5        ;-- contour value increment
```

# Shapefiles – outline counties of Germany (2/3)

# Shapefiles – outline counties of Germany (3/3)

```
draw(plot)           ;-- draw the plot  
frame(wks)          ;-- advance the frame  
  
end
```



# Task: Map of Germany with Mercator Projection

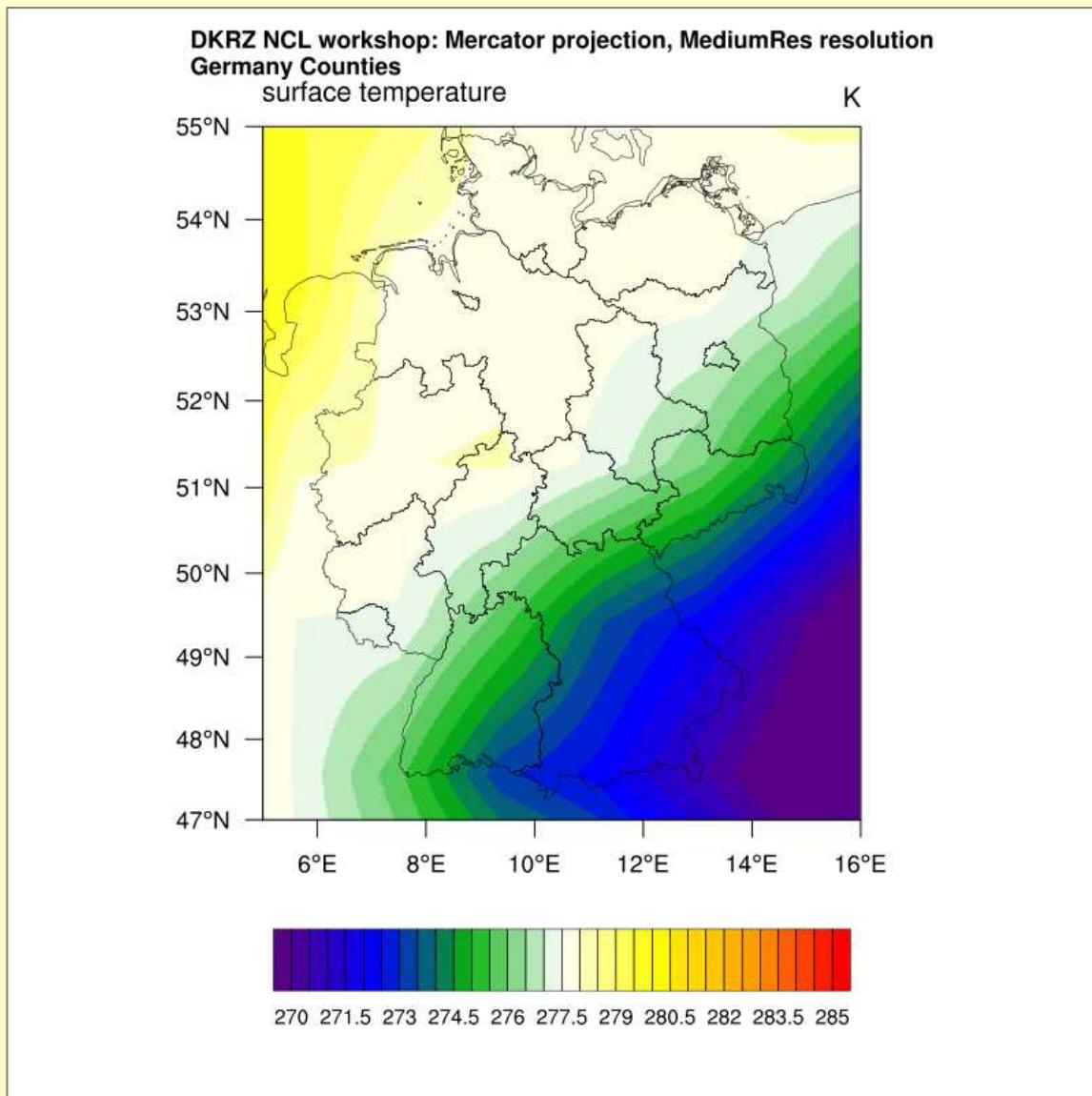
- Use the script **part\_VI\_map\_projection.ncl**
- Copy it to your own directory and modify it
  - Use **Mercator** projection and choose the **region of Germany**  
**lon: 5-16 deg. and lat: 47-55 deg.**
  - Center the plot on **lon=11 deg.** and **lat=51 deg.**
  - Use medium resolution **MediumRes** for the map and outline it
  - Choose a nice colormap
  - Read the shapefile as learned before to plot **Germanys county lines**

## Hints:

*See also part\_VI\_map\_resolution.ncl*

- **mpProjection**
- **mpDataBaseVersion**
- **mpMinLonF/mpMaxLonF/mpMinLatF/mpMaxLatF**
- **mpCenterLonF/mpCenterLatF**
- **cnFillPalette**
- **gsn\_add\_shapefile**

# Task: Map of Germany with Mercator Projection



# Task: Map of Germany with Mercator Projection (1/3)

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f      = addfile("$NCL_TUT/data/ECHAM5_OM_A1B_2001_0101-1001_2D.nc", "r")
  var   = f->tsurf(0,:,:) 

  wks = gsn_open_wks("png","task_VI_Mercator_MediumRes_Germany_Counties")

  res           = True
  res@gsnDraw    = False          ;-- don't draw the plot
  res@gsnFrame   = False          ;-- don't advance frame yet
  res@gsnMaximize = True          ;-- maximize plot in frame

  res@cnFillOn     = True          ;-- turn on countour fill
  res@cnFillMode    = "RasterFill" ;-- contour cell fill mode
  res@cnRasterSmoothingOn = True   ;-- contour smoothing on
  res@cnFillPalette  = "testcmap"  ;-- choose a colormap
  res@cnLinesOn     = False         ;-- don't draw contour lines
```

# Task: Map of Germany with Mercator Projection (1/3)

```
res@cnLevelSelectionMode = "ManualLevels"      ;-- set manual contour levels
res@cnMinLevelValF     = 270.0                  ;-- minimum contour value
res@cnMaxLevelValF     = 285.0                  ;-- maximum contour value
res@cnLevelSpacingF    = 0.5                     ;-- contour value increment

res@mpFillOn            = False                 ;-- turn off map fill
res@mpLimitMode          = "LatLon"              ;-- must be set to LatLon for
                                                ;-- Mercator sub-region selection

res@mpMinLatF           = 47.                    ;-- min lat
res@mpMaxLatF           = 55.                    ;-- max lat
res@mpMinLonF           = 5.                     ;-- min lon
res@mpMaxLonF           = 16.                    ;-- max lon
res@mpOutlineOn          = True                 ;-- draw map outlines
res@mpDataBaseVersion    = "MediumRes"           ;-- choose map resolution
res@mpProjection         = "Mercator"            ;-- set map projection
res@pmTickMarkDisplayMode = "Always"             ;-- draw tickmarks

res@tiMainString = "DKRZ NCL workshop: Mercator projection, MediumRes
resolution~C~Germany Counties" ;-- draw title
res@tiMainFontHeightF   = 0.015
```

# Task: Map of Germany with Mercator Projection (1/3)

```
;-- generate map, but don't draw it
plot = gsn_csm_contour_map(wks,var,res)

;-- add polylines from the shapefile to the plot
shp_filename = ("$NCL_TUT/data/Shapefiles/DEU_adm/DEU_adm1.shp")
poly = gsn_add_shapefile_polylines(wks,plot,shp_filename,False)

;-- draw the plot and the attached shapefile outlines and advance the frame
draw(plot)
frame(wks)

end
```

## Part VII

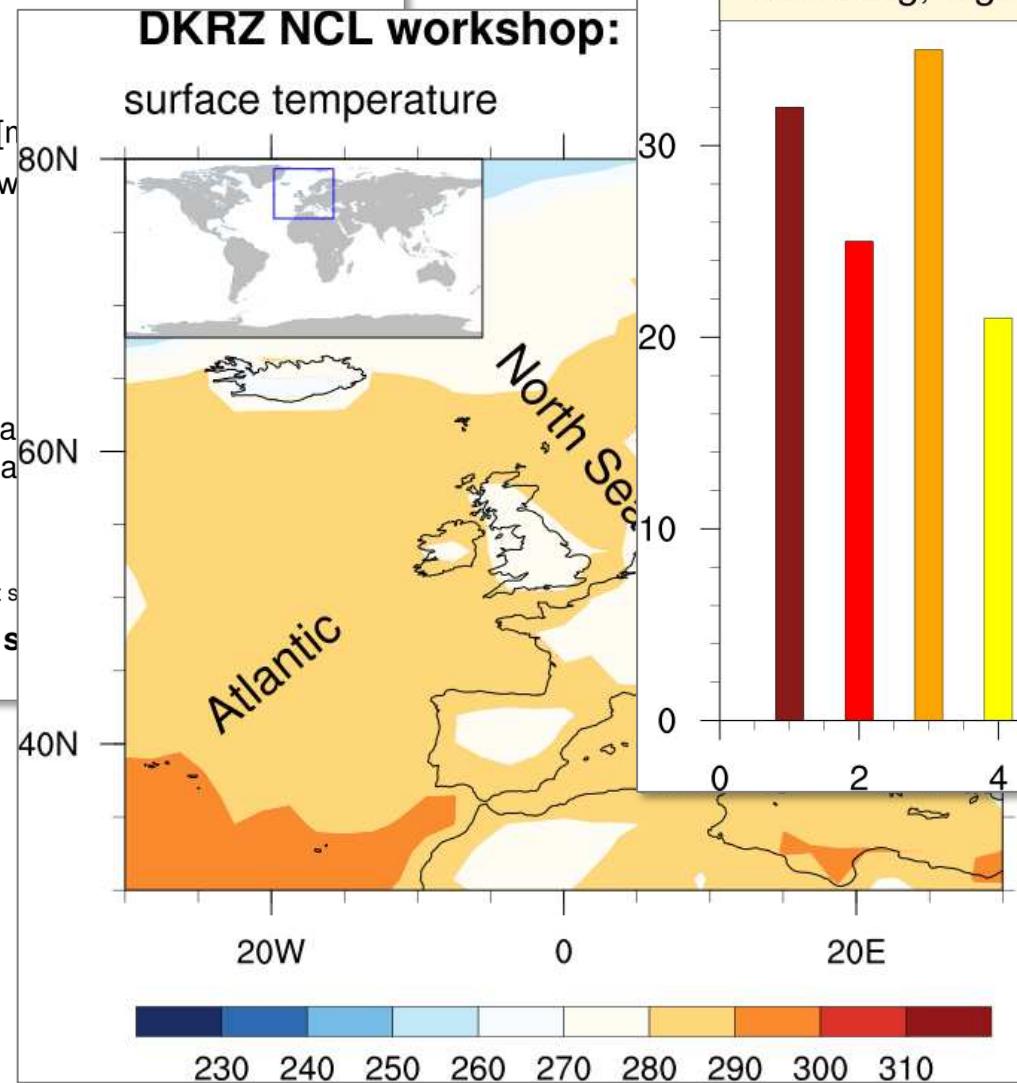
# Caption and Annotations

Exercises and Tasks

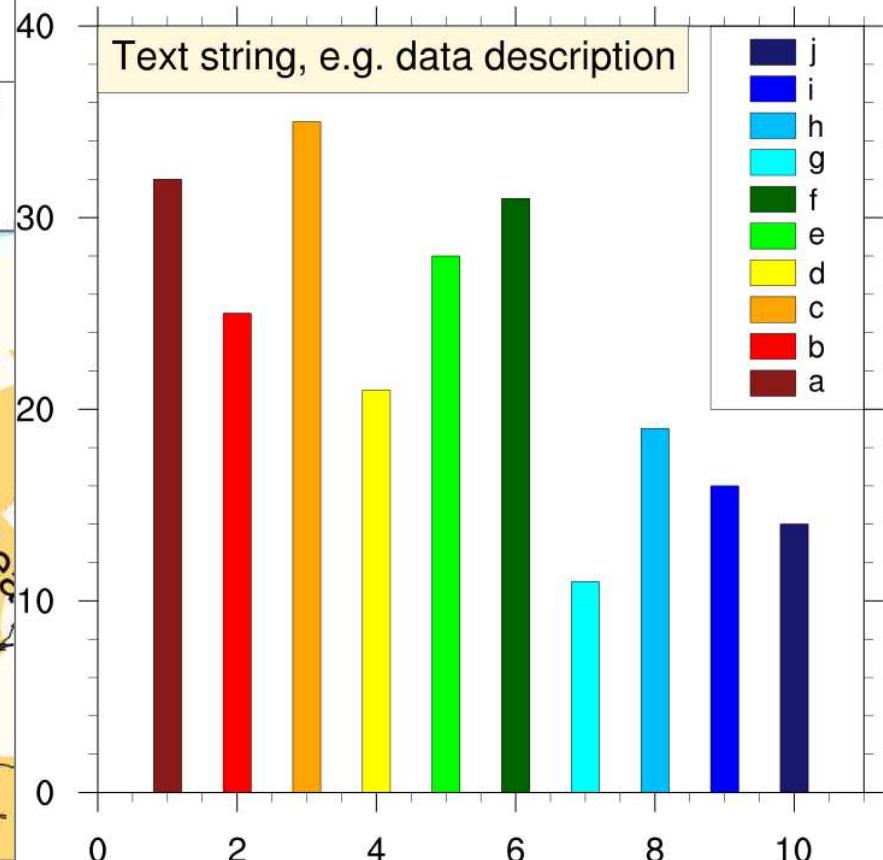
# Caption and Annotations

## DKRZ NCL workshop: Text settings

Umlaute: Ä ä  
 Superscript: m<sup>3</sup> [n]  
 Subscript: Schw  
 Format:  
 1.  
 2.  
 3.  
 4.  
 Greek font: alpha sigma  
 Font size 100%  
**Helvetica-bold font s**



## DKRZ NCL workshop: Labelbar Annotation



# Map annotations (1/4)

```

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f    = addfile("$NCL_TUT/data/ECHAM5_OM_A1B_2001_0101-1001_2D.nc", "r")
  var = f->tsurf(0,:,:)

  wks = gsn_open_wks("png", "part_VII_map_annotation")

  mres1          = True
  mres1@gsnDraw   = False           ;-- don't draw the plot yet
  mres1@gsnFrame  = False           ;-- don't advance the frame
  mres1@gsnMaximize = True          ;-- maximize the plot output
  mres1@cnFillOn   = True           ;-- filled contours
  mres1@cnLinesOn  = False          ;-- don't draw contour lines
  mres1@mpDataBaseVersion = "MediumRes" ;-- set map data base
  mres1@mpOutlineOn   = True          ;-- turn map outline on
  mres1@mpGeophysicalLineThicknessF = 2 ;-- increase coastline thickness
  mres1@mpMinLonF     = -30.         ;-- min lon
  mres1@mpMaxLonF     = 30.          ;-- max lon
  mres1@mpMinLatF     = 30.          ;-- min lat
  mres1@mpMaxLatF     = 80.          ;-- max lat

```

# Map annotations (2/4)

```

mres1@lbBoxMinorExtentF      =  0.15          ;-- decrease height of labelbar

mres1@tiMainString          = "DKRZ NCL workshop:" map annotation" ;-- title string
mres1@tiMainFontHeightF     =  0.02          ;-- main title font size

map1 = gsn_csm_contour_map(wks, var, mres1)      ;-- draw the contour map

mres2                         = True           ;-- resources smaller world map
mres2@gsnDraw                 = False          ;-- don't draw the plot yet
mres2@gsnFrame                = False          ;-- don't advance the frame
mres2@gsnLeftString          = ""             ;-- don't draw left string
mres2@gsnRightString         = ""             ;-- don't draw right string
mres2@vpWidthF                = 0.3            ;-- set viewport width
mres2@vpHeightF               = 0.3            ;-- set viewport height
mres2@mpLandFillColor         = "Gray75"       ;-- land fill color
mres2@mpOceanFillColor        = "White"         ;-- ocean fill color
mres2@mpInlandWaterFillColor = "White"         ;-- inland water color
mres2@tmXBOn                  = False          ;-- turn off tickmarks at bottom
mres2@tmXTOn                  = False          ;-- turn off tickmarks at top
mres2@tmYLOn                  = False          ;-- turn off tickmarks at left side
mres2@tmYROn                  = False          ;-- turn off tickmarks at right side

```

# Map annotations (3/4)

```

map2 = gsn_csm_map(wks, mres2)                                ;-- create the smaller world map,
                                                               ;-- but don't draw it yet

pres                  = True                                     ;-- set polyline resources
pres@gsLineThicknessF = 2.0                                     ;-- polyline thickness
pres@gsLineColor      = "blue"                                    ;-- polyline color

;-- attach polyline to map2
pbox = gsn_add_polyline(wks, map2, (-30,30,30,-30,-30), (30,30,80,80,30), pres)

amres                = True                                     ;-- set annotation resources
amres@amJust          = "TopLeft"                                ;-- labelbar justification
amres@amParallelPosF = -0.5                                    ;-- move labelbar to the right of plot
amres@amOrthogonalPosF = -0.5                                  ;-- move labelbar to the top of plot

annoid = gsn_add_annotation(map1, map2, amres)    ;-- attach the labelbar to the plot

txres1               = True                                     ;-- set text resources
txres1@txFontHeightF = 0.03                                    ;-- smaller text font size
txres1@txAngleF       = 40.                                     ;-- rotate text string 40 degrees

text1 = gsn_add_text(wks, map1, "Atlantic", -20, 45, txres1) ;-- attach text to map1

```

# Map annotations (4/4)

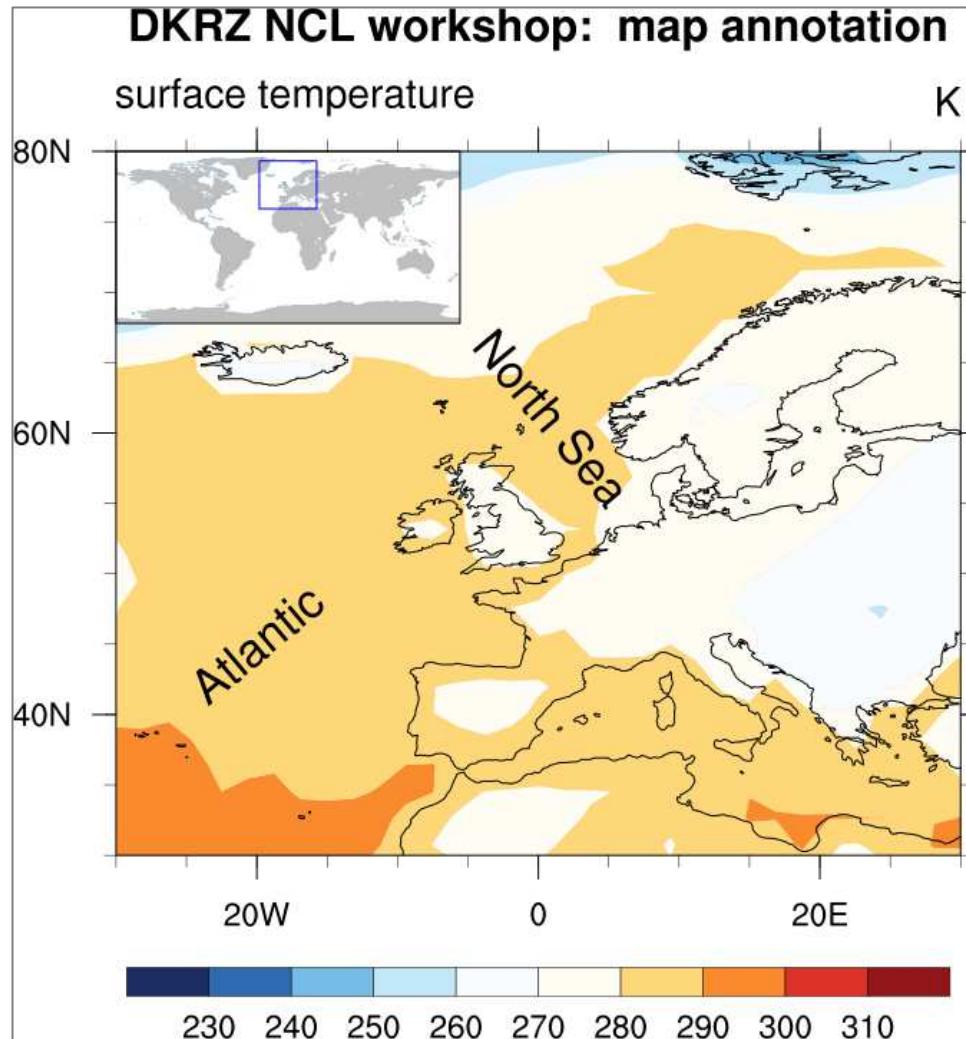
```
txres2          = True           ;-- set text resources
txres2@txFontHeightF = 0.03       ;-- smaller text font size
txres2@txAngleF   = -50.         ;-- rotate text string 40 degrees

text2 = gsn_add_text(wks, map1, "North Sea", 1.0, 61, txres2) ;-- attach text to map1

;-- draw the map and advance the frame
draw(map1)
frame(wks)

end
```

# Map annotations



# Labelbar/legend annotation (1/4)

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin

    llab = (/"a","b","c","d","e","f","g","h","i","j"/) ;-- define labels
    cols = (/"firebrick4","red","orange","yellow","green","darkgreen", \
              "cyan","DeepSkyBlue","blue","midnightblue"/) ;-- define colors
    nlb = dimsizes(cols) ;-- number of labelbar boxes

    x = (/1,2,3,4,5,6,7,8,9,10/) ;-- define x-values
    y = (/32,25,35,21,28,31,11,19,16,14/) ;-- define y-values

    wks = gsn_open_wks("png", "part_VII_legend_annotation")

    res = True
    res@gsnMaximize = True ;-- maximize plot output
    res@gsnDraw = False ;-- don't draw plot yet
    res@gsnFrame = False ;-- don't advance frame yet
    res@gsnXYBarChart = True ;-- turn on bar chart
    res@gsnXYBarChartBarWidth = 0.4 ;-- set width of bins
    res@gsnXYBarChartColors = cols ;-- set the colors for the bins
```

# Labelbar/legend annotation (2/4)

```

res@trYMinF          = 0                      ;-- start bins from bottom
res@trYMaxF          = max(y)+5                ;-- start bins from bottom
res@trXMinF          = 0                      ;-- start at x-value 0
res@trXMaxF          = 11                     ;-- end at x-value 11

res@tiMainString     = "DKRZ NCL workshop: Labelbar Annotation," ;-- title string

plot = gsn_csm_xy (wks,x,y,res)           ;-- create the plot, but don't draw it yet

getvalues plot
    "vpWidthF"   : vpw                      ;-- get viewport width of plot
    "vpHeightF"  : vph                      ;-- get viewport height of plot
end getvalues

lres                  = True                 ;-- set labelbar resources
lres@lbAutoManage     = True                 ;-- necessary to control sizes
lres@lbFillColors     = cols                 ;-- labelbar colors
lres@vpWidthF         = 0.2 * vpw            ;-- labelbar width
lres@vpHeightF        = 0.5 * vph            ;-- labelbar height
lres@lbBoxMajorExtentF = 0.7                 ;-- insert white space between boxes
lres@lbMonoFillPattern = True                ;-- solid fill pattern

```

# Labelbar/legend annotation (3/4)

```

lres@lbLabelFontHeightF = 0.08           ;-- font height
lres@lbLabelJust       = "CenterLeft"    ;-- label justification

lbid = gsn_create_labelbar(wks,nlb,llab,lres) ;-- create the labelbar,
                                                ;-- but don't draw it yet

amres                  = True            ;-- set annotation resources
amres@amJust            = "TopRight"      ;-- labelbar justification
amres@amParallelPosF   = 0.5             ;-- move labelbar to the right of plot
amres@amOrthogonalPosF = -0.5            ;-- move labelbar to the top of plot

annoid = gsn_add_annotation(plot,lbid,amres) ;-- attach the labelbar to the plot

txres                  = True            ;-- set text resources
txres@txPerimOn         = True            ;-- draw a box around the text
txres@txBackgroundColor = "cornsilk,"     ;-- box fill color
txres@txFontHeightF     = 0.03            ;-- smaller text font size

txid = gsn_create_text(wks, "Text string, e.g. data description", txres)
      ;-- create text, but don't draw it yet

```

# Labelbar/legend annotation (4/4)

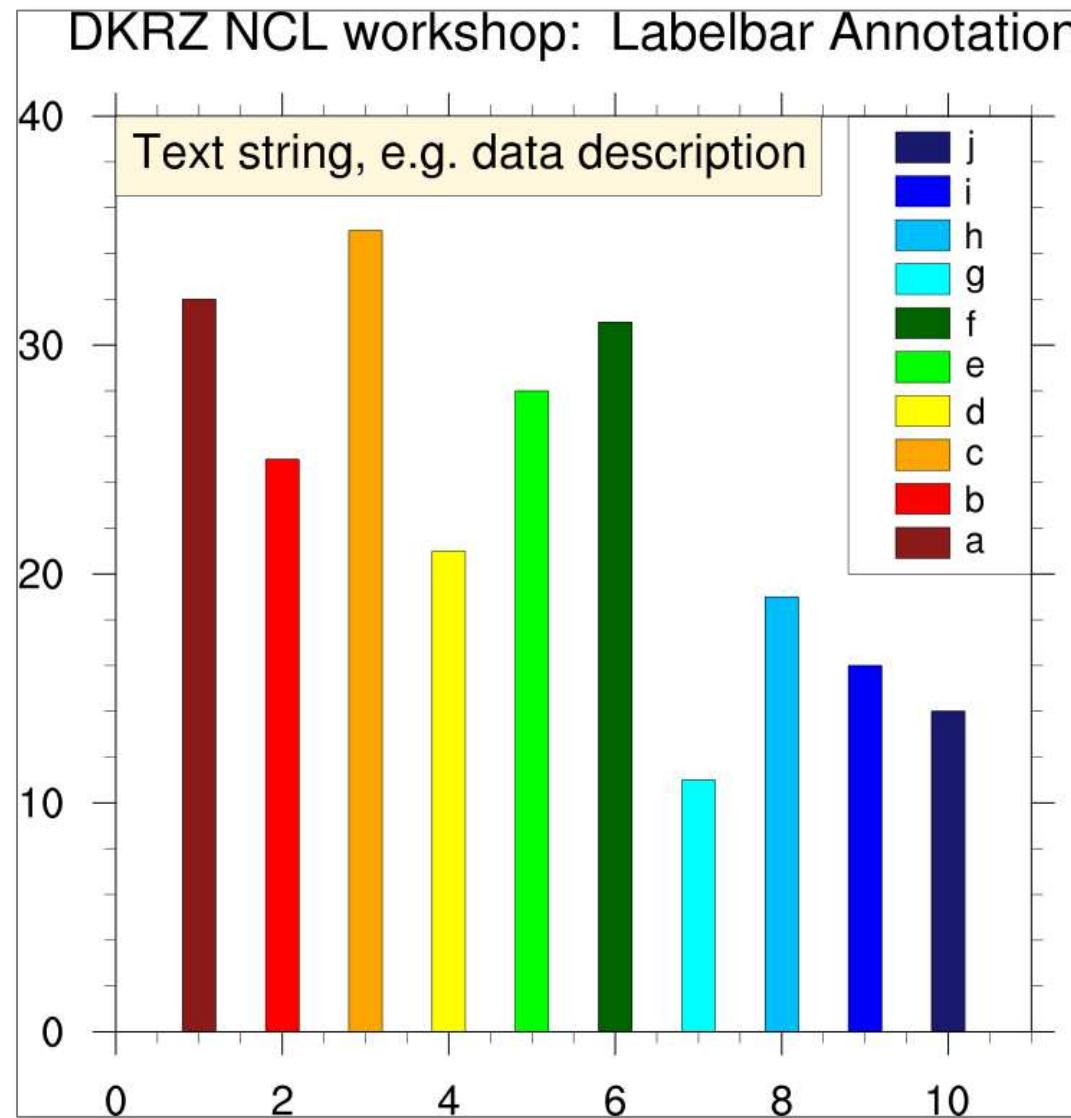
```
amres@amJust          = "TopLeft"      ;-- labelbar justification
amres@amParallelPosF = -0.5           ;-- move labelbar to the right of plot
amres@amOrthogonalPosF = -0.5         ;-- move labelbar to the top of plot

annoid = gsn_add_annotation(plot, txid, amres) ;-- attach text to the plot

;-- draw second plot and advance the frame
draw(plot)
frame(wks)

end
```

# Labelbar/legend annotation



# Text settings (1/6)

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"

begin

;-- define german "Umlaute"
Auml = "A~H-15V6F35~H~FV-6H3~"
auml = "a~H-13V2F35~H~FV-2H3~"
Ouml = "O~H-16V6F35~H~FV-6H3~"
ouml = "o~H-14V2F35~H~FV-2H3~"
Uuml = "U~H-15V6F35~H~FV-6H3~"
uuml = "u~H-13V2F35~H~FV-2H3~"

;-- define super- and subscript variable
super = "m~S~3~N~ [m s~S~-1~N~] [kg m~S~-2~N~] 30~S~o~N~C"
sub   = "Schwefels"+auml+"ure: H~B~2~N~SO~B~4~N~"

data  = (/ "1.",      "2.",      "3.",      "4."/ )
diff  = (/ 16.25,    -0.93,    0.43,    3.5/)
var   = (/ 0.06,     0.02,     0.04,     0.05/)
ratio = (/        2,      2.4,      1.1,      0.9/)

ntext = dimsizes(data)
```

## Text settings (2/6)

```
wks = gsn_open_wks("png", "part_VII_text_settings")

;-- x, y start point for writing
x = 0.1
y = 0.95
inc = 0.06

;-- text resources
txres           = True
txres@txFont    = "helvetica-bold"      ;-- change font
txres@txFontHeightF = 0.03              ;-- set font size
txres@txJust    = "CenterCenter"        ;-- set text justification
str = "DKRZ NCL workshop: Text settings" ;-- title string
gsn_text_ndc(wks,str,0.5,y,txres)       ;-- draw title string

txres@txFont    = "helvetica"            ;-- change font
txres@txJust    = "CenterLeft"          ;-- change text justification
str1 = "Umlaute:"
gsn_text_ndc(wks,str1,x,y-2*inc,txres) ;-- draw string

str2 = Auml+" "+auml+" "+Ouml+" "+ouml+" "+Uuml+" "+uuml
gsn_text_ndc(wks,str2,x+0.3,y-2*inc,txres) ;-- draw string
```

## Text settings (3/6)

```
str1 = "Superscript:"  
gsn_text_ndc(wks,str1,x,y-3*inc,txres)  
str2 = super  
gsn_text_ndc(wks,str2,x+0.3,y-3*inc,txres)  
  
str1 = "Subscript:"  
gsn_text_ndc(wks,str1,x,y-4*inc,txres)  
str2 = sub  
gsn_text_ndc(wks,str2,x+0.3,y-4*inc,txres)  
  
;-- nice formated text output using sprintf  
str = "Format:"  
gsn_text_ndc(wks,str,x,y-5*inc,txres)  
  
xpos = 0.4  
do i=0,ntext-1  
    ypos = y-5*inc-i*0.05  
    gsn_text_ndc(wks,data(i),xpos,ypos,txres)  
end do  
  
txres@txJust = "CenterRight"
```

## Text settings (4/6)

```
do i=0,ntext-1
    xpos = 0.65                                ;-- column 1
    ypos = y-5*inc-i*0.05
    str = sprintf("%6.2f",diff(i))
    gsn_text_ndc(wks,str,xpos,ypos,txres)

    xpos = xpos + 0.12                         ;-- column 2
    str = sprintf("%5.2f",var(i))
    gsn_text_ndc(wks,str,xpos,ypos,txres)

    xpos = xpos + 0.12                         ;-- column 3
    str = sprintf("%3.1f",ratio(i))
    gsn_text_ndc(wks,str,xpos,ypos,txres)
end do

;-- greek characters
xpos = 0.3
ypos = 0.35

str1 = "Greek font:"
gsn_text_ndc(wks,str1,xpos,ypos,txres)
```

## Text settings (5/6)

```
str2 = "alpha = ~F33~a~N~"  
gsn_text_ndc(wks,str2,xpos+0.27,ypos,txres)  
  
str2 = "beta    = ~F33~b~N~"  
gsn_text_ndc(wks,str2,xpos+0.55,ypos,txres)  
  
str2 = "sigma = ~F33~s~N~"  
gsn_text_ndc(wks,str2,xpos+0.27,ypos-0.05,txres)  
  
;-- decrease the font  
str1 = "Font size 100%"  
gsn_text_ndc(wks,str1,xpos+0.08,ypos-3*inc,txres)  
  
str2 = "~Z70~Font size 70%~N~"  
gsn_text_ndc(wks,str2,xpos+0.3,ypos-3*inc,txres)  
  
str3 = "~Z40~Font size 40%~N~"  
gsn_text_ndc(wks,str3,xpos+0.45,ypos-3*inc,txres)
```

# Text settings (6/6)

```
;-- change Helvetica font to Helvetica-bold
txres1          = True
txres1@txFont      = "helvetica-bold"
txres1@txFontHeightF = 0.03
txres1@txJust       = "CenterLeft"

bold = "Helvetica-bold font size 0.03"
gsn_text_ndc(wks,bold,x,0.08,txres1)

frame(wks)

end
```

# Text settings

## DKRZ NCL workshop: Text settings

Umlaute: Ä ä Ö ö Ü ü

Superscript:  $m^3$  [ $m\ s^{-1}$ ] [ $kg\ m^{-2}$ ]  $30^\circ C$

Subscript: Schwefelsäure:  $H_2SO_4$

Format:

1.	16.25	0.06	2.0
2.	-0.93	0.02	2.4
3.	0.43	0.04	1.1
4.	3.50	0.05	0.9

Greek font: alpha =  $\alpha$  beta =  $\beta$   
sigma =  $\sigma$

Font size 100% Font size 70% Font size 40%

**Helvetica-bold font size 0.03**

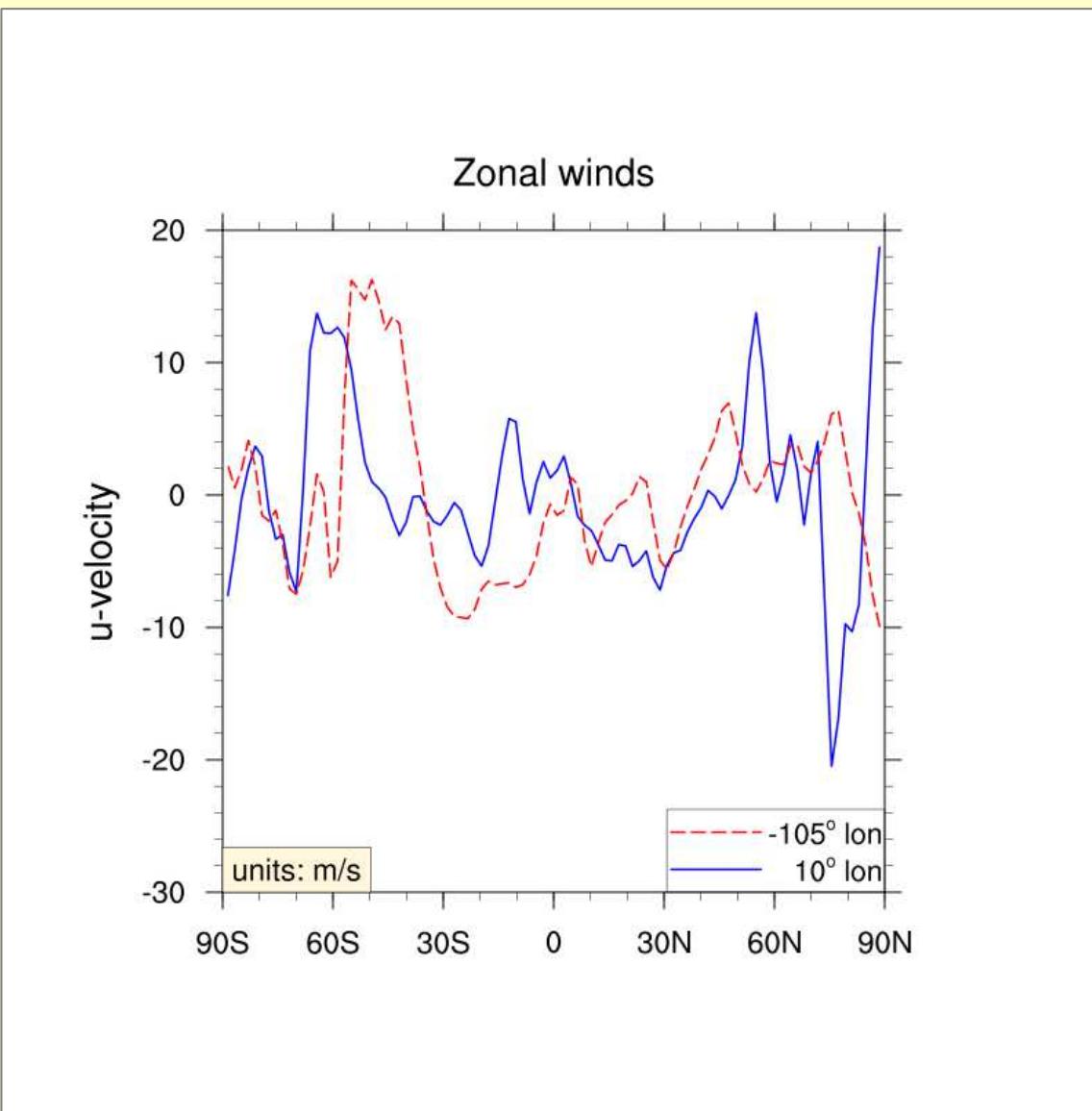
# Task: Annotate map with text and legend

- Use the script **part\_I\_multiple\_curves\_xy.ncl**
- Copy it to your own directory and modify it
  - Change the title string to "**Zonal winds**"
  - Set the **line thicknesses** to **3.0**
  - Add a **legend** for the two line graphs in the **lower right corner** with the labels "**-105 degrees lon**" for the red line and "**10 degrees lon**" for the blue line
  - Add a **text box** at the **lower left corner** of the plot containing the **units** of the data variable. The text box **background color** is **cornsilk**.

## Hints:

- **xyLineThicknesses/IgLineThicknessF**
- **gsn\_create\_legend**
- **u@units**
- **txBackgroundFillColor**
- **gsn\_create\_text**
- **gsn\_add\_annotation**

# Task: Annotate map with text and legend



# Task: Annotate map with text and legend (1/3)

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
  f = addfile("$NCL_TUT/data/EH_OM_A1B_3Dvectors_t1-24.nc","r")
  u = f->u                                     ;-- set variable u

  data      = new((/2,dimsizes(u&lat)/), float) ;-- assign multidimensional array
  data(0,:) = u(0,0,:,:,{10})                   ;-- values at longitude 10 deg.
  data(1,:) = u(0,0,:,:,{-105})                ;-- values at longitude -105 deg.

  wks = gsn_open_wks("png", "task_VII_xy_plot_annotations")

  res          = True                         ;-- create plot resource object
  res@gsnDraw  = False                        ;-- don't draw the plot yet
  res@gsnFrame = False                        ;-- don't advance the frame
  res@xyLineColors = ("blue", "red")          ;-- line colors
  res@xyLineThicknesses = (/3.0,3.0/)        ;-- set line thicknesses
  res@tiMainString = "Zonal winds"           ;-- draw title

  plot = gsn_csm_xy(wks, u&lat, data, res) ;-- create the plot, but don't draw it
```

## Task: Annotate map with text and legend (2/3)

```
;-- set legend resources
lres           = True
lres@lgLineColors = (/"blue","red"/)
lres@lgLineThicknessF = 3.0
lres@lgLabelFontHeightF = .08          ;-- set the legend label font thickness
lres@lgLabelJust      = "CenterRight" ;-- set text justification
lres@vpWidthF         = 0.2            ;-- width of legend in NDC coordinates
lres@vpHeightF        = 0.075          ;-- height of legend in NDC coordinates

lbid = gsn_create_legend(wks,2,(/" 10~S~o~N~ lon"," -105~S~o~N~ lon"/),lres)
;-- create the legend

;-- set annotation resources
amres          = True
amres@amJust    = "BottomRight" ;-- labelbar justification
amres@amParallelPosF = 0.5          ;-- move legend to the right
amres@amOrthogonalPosF = 0.5        ;-- move the legend down

annoid1 = gsn_add_annotation(plot,lbid,amres) ;-- attach legend to plot
```

# Task: Annotate map with text and legend (3/3)

```
;-- set text resources
txres          = True
txres@txPerimOn = True      ;-- draw a box around the text
txres@txBackgroundFillColor = "cornsilk" ;-- box fill color
txres@txFontHeightF = 0.02    ;-- smaller text font size

txid = gsn_create_text(wks, "units: "+u@units, txres) ;-- create text, but
                                                       ;-- don't draw it yet

amres@amJust      = "BottomLeft" ;-- labelbar justification
amres@amParallelPosF = -0.5     ;-- move labelbar to the right of plot
amres@amOrthogonalPosF = 0.5     ;-- move labelbar to the top of plot

annoid = gsn_add_annotation(plot, txid, amres) ;-- attach text to the plot

;-- draw the plot and advance the frame
draw(plot)
frame(wks)

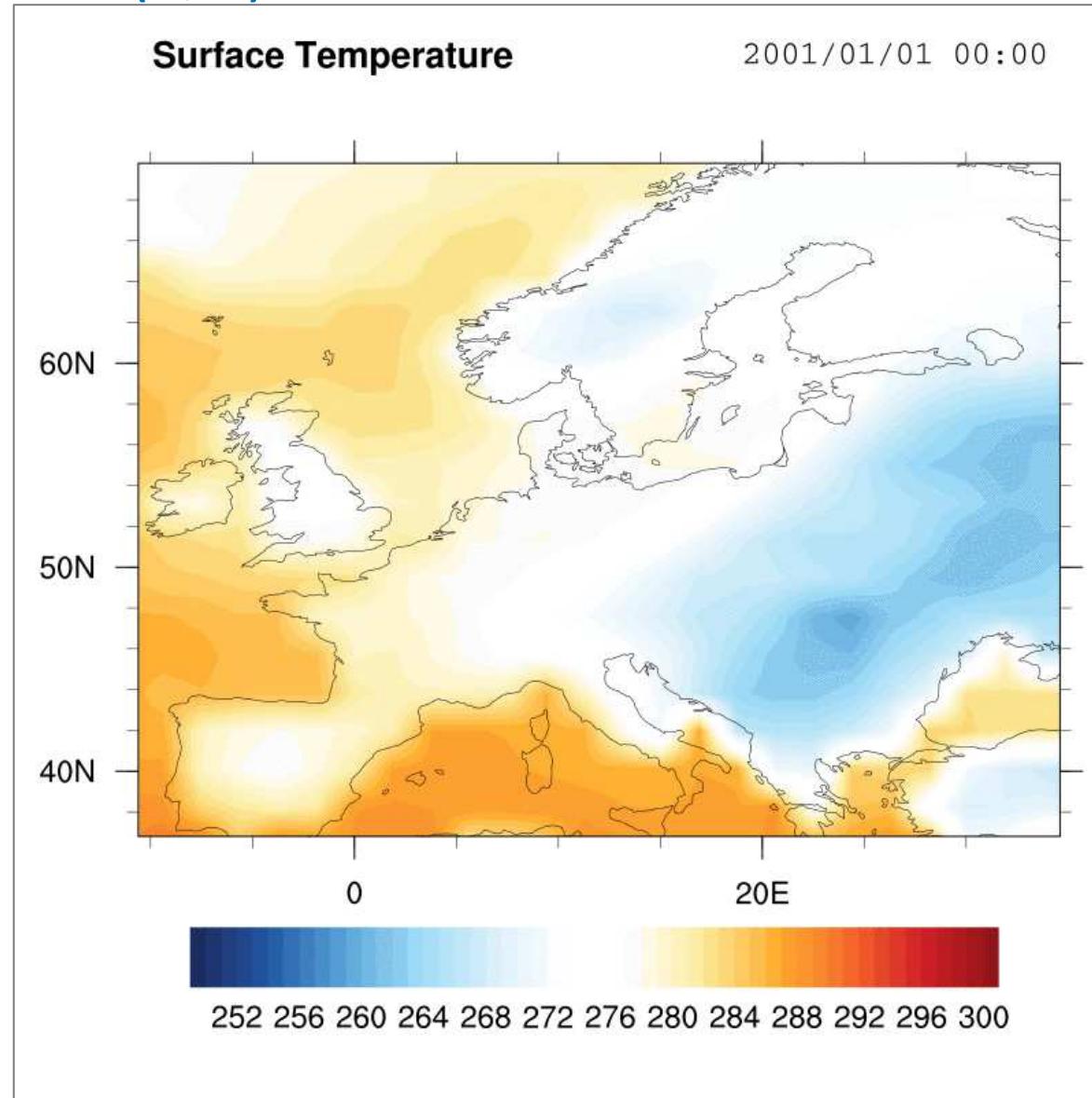
end
```

Part VIII

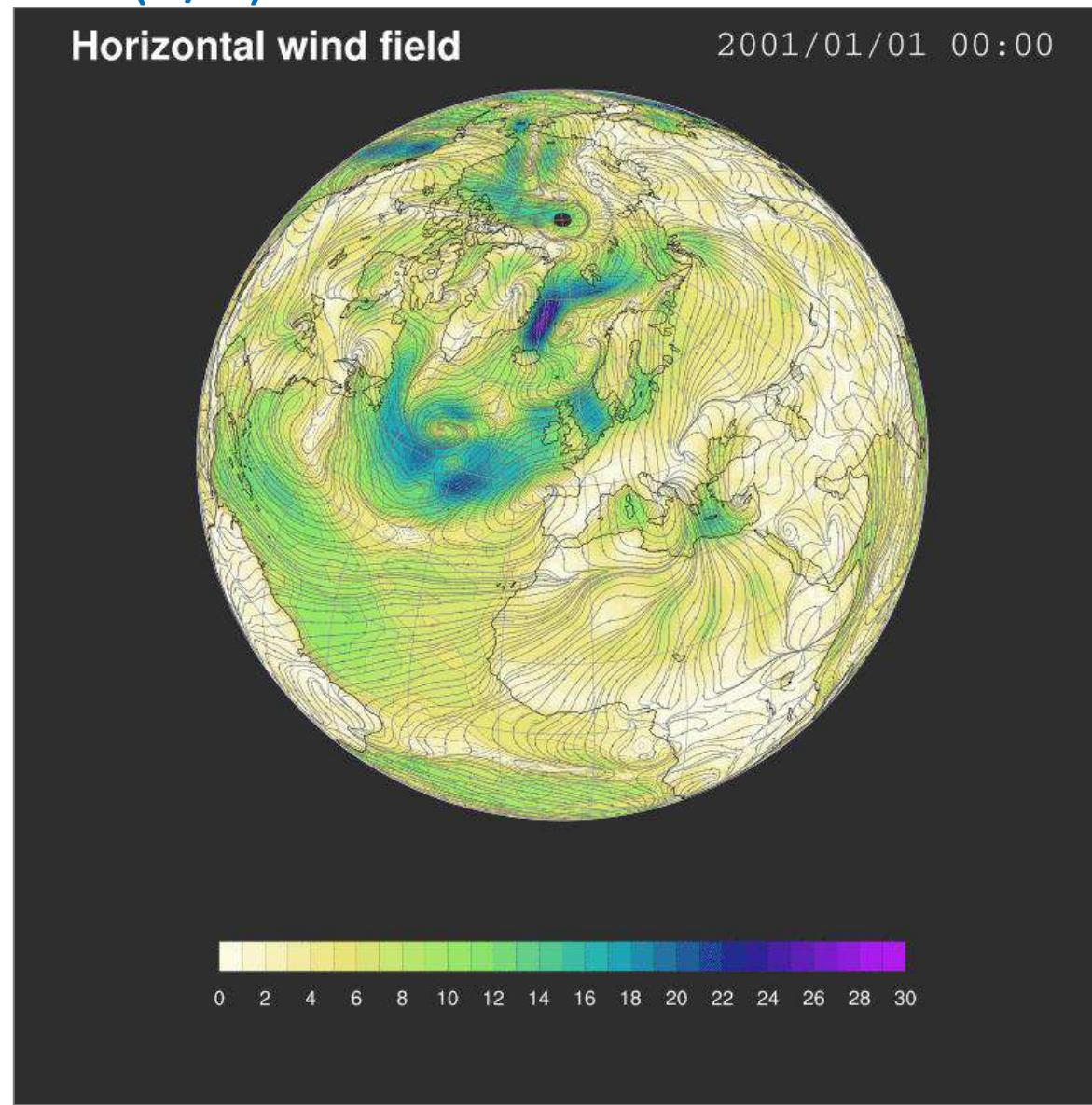
## Animations

Exercises and Tasks

# Animations (1/2)



## Animations (2/2)



# Time animation of a contour plot (1/4)

```
undef ("getDate")
function getDate(time)
local utc_date, year, mon, day, hours, mins, str_date
begin
    utc_date = cd_calendar(time, 0)           ;-- convert date to UT-referenced date
    year     = sprinti("%0.4i",tointeger(utc_date(:,0))) ;-- get year as integer
    mon      = sprinti("%0.2i",tointeger(utc_date(:,1))) ;-- get month as int
    day      = sprinti("%0.2i",tointeger(utc_date(:,2))) ;-- get day as integer
    hours    = sprinti("%0.2i",tointeger(utc_date(:,3))) ;-- get day as integer
    mins     = sprinti("%0.2i",tointeger(utc_date(:,4))) ;-- get day as integer
    str_date = year+ "/" +mon+ "/" +day+ " " +hours+ ":" +mins ;-- yyyy/mm/dd HH:MM
    return(str_date)
end

;-----
;-- main script
;-----

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
```

# Time animation of a contour plot (2/4)

```
f      = addfile("$NCL_TUT/data/ECHAM5_OM_A1B_2001_0101-1001_2D.nc", "r")
temp  = f->tsurf                                ;-- temperature data 1st time step
time  = f->time                                 ;-- time values

wks_type           = "png"                      ;-- plot output type
wks = gsn_open_wks(wks_type,"part_VIII_animation_contour_map")

res               = True
res@gsnMaximize   = True                        ;-- maximize plot output
res@gsnAddCyclic  = False                       ;-- don't add cyclic point
res@gsnRightString = ""                         ;-- don't draw right string
res@gsnLeftString = ""                         ;-- don't draw left string

res@cnFillPalette = "BlueWhiteOrangeRed" ;-- choose color map
res@cnLevelSelectionMode = "ManualLevels"       ;-- use manual cont. line levels
res@cnMinLevelValF = 250.0                      ;-- contour min. value
res@cnMaxLevelValF = 300.0                      ;-- contour max. value
res@cnLevelSpacingF = 1.0                        ;-- contour interval
res@cnFillOn       = True                        ;-- enable color fill
res@cnLinesOn     = False                       ;-- disable contour lines
```

# Time animation of a contour plot (3/4)

```
res@cnFillMode          = "RasterFill"           ;-- set fill mode
res@cnRasterSmoothingOn = True                 ;-- smooth contouring

res@lbBoxLinesOn         = False                ;-- no lines around labelbar boxes

res@mpDataBaseVersion   = "MediumRes"           ;-- set map data base
res@mpMinLonF            = -10.6               ;-- lon min. value
res@mpMinLatF            = 36.8                ;-- lat min. value
res@mpMaxLonF            = 34.6                ;-- lon max. value
res@mpMaxLatF            = 69.8                ;-- lat max. value

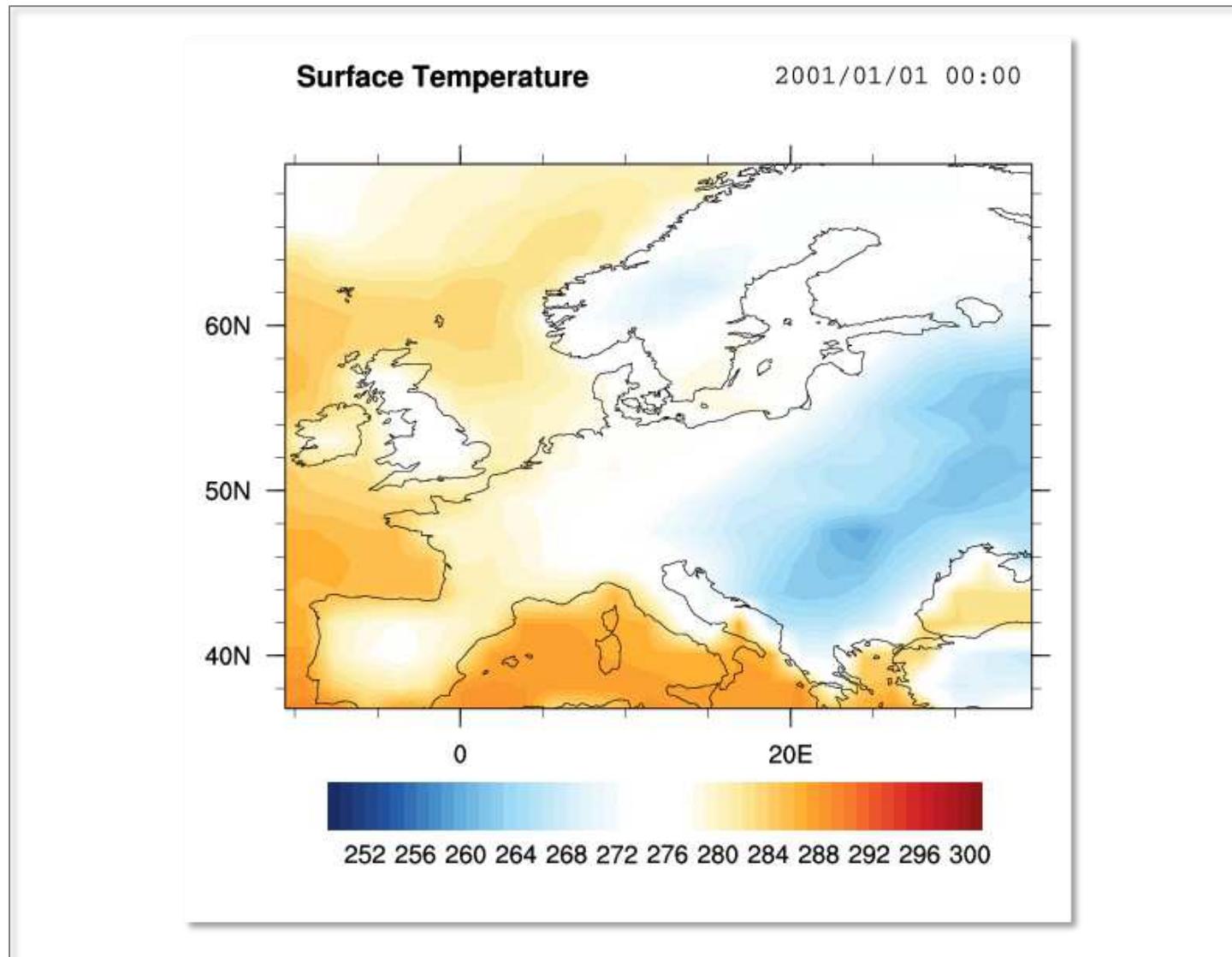
res@tiMainFontHeightF   = 0.02                ;-- main title font size
res@tiMainOffsetYF       = 0.06                ;-- move title upward

;-- create plots, loop along time
ntimes = dimsizes(time)
do i=0,ntimes-1
    res@tiMainString = "Surface Temperature~F29~~Z70~" +
getdate(time(i)) + "~N~" ;-- draw title
    plot = gsn_csm_contour_map(wks,temp(i,:,:),res)    ;-- create the plots
end do
```

# Time animation of a contour plot (4/4)

```
;-- if wks_type = "ps": make sure that the PS file is closed  
  delete(wks)  
  
;-- convert the plot output from PNG to Animated GIF  
  system("convert -rotate -90 -delay 75 part_VIII_animation_contour_map*.png  
part_VIII_animation_contour_map.gif")  
  
  system("rm -rf part_VIII_animation_contour_map*.png")  
  
end
```

# Time animation of a contour plot



# Time animation – vectors on contour (1/5)

```
undef ("getDate")
function getDate(time)
local utc_date, year, mon, day, hours, mins, str_date
begin
    utc_date = cd_calendar(time, 0)           ;-- convert date to UT-referenced date
    year     = sprinti("%0.4i",tointeger(utc_date(:,0))) ;-- get year as integer
    mon      = sprinti("%0.2i",tointeger(utc_date(:,1))) ;-- get month as int
    day      = sprinti("%0.2i",tointeger(utc_date(:,2))) ;-- get day as integer
    hours    = sprinti("%0.2i",tointeger(utc_date(:,3))) ;-- get day as integer
    mins     = sprinti("%0.2i",tointeger(utc_date(:,4))) ;-- get day as integer
    str_date = year+ "/" +mon+ "/" +day+ " " +hours+ ":" +mins ;-- yyyy/mm/dd HH:MM
    return(str_date)
end

;-----
;-- main script
;-----

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
```

# Time animation – vectors on contour (2/5)

```

f      = addfile("$NCL_TUT/data/ECHAM5_OM_A1B_2001_0101-1001_2D.nc", "r")
temp   = f->tsurf                                     ;-- temperature data
u      = f->u10                                       ;-- 10m u-velocity
v      = f->v10                                       ;-- 10m v-velocity
time   = f->time                                      ;-- time values

wks_type = "png"                                         ;-- plot output type
wks = gsn_open_wks(wks_type, "part_VIII_animation_simple_vector_contour_map")

res          = True
res@gsnMaximize = True                                ;-- maximize plot output
res@gsnAddCyclic = False                             ;-- don't add cyclic point
res@gsnRightString = ""                                ;-- don't draw right string
res@gsnLeftString = ""                                ;-- don't draw left string
res@gsnScalarContour = True                            ;-- must be set for overlay
                                                ;-- vectors on contour map

res@cnFillPalette = "BlueWhiteOrangeRed"           ;-- choose color map
res@cnLevelSelectionMode = "ManualLevels"          ;-- use manual contour levels
res@cnMinLevelValF = 250.0                            ;-- contour min. value
res@cnMaxLevelValF = 300.0                            ;-- contour max. value
res@cnLevelSpacingF = 1.0                             ;-- contour interval

```

# Time animation – vectors on contour (3/5)

```

res@cnFillOn          = True                      ;-- enable color fill
res@cnLinesOn         = False                     ;-- disable contour lines
res@cnFillMode        = "RasterFill"           ;-- set fill mode
res@cnRasterSmoothingOn = True                      ;-- smooth contouring
res@cnLabelBarEndStyle = "ExcludeOuterBoxes";-- don't draw outer boxes

res@lbBoxLinesOn      = False                     ;-- no lines around labelbar boxes
res@lbBoxMinorExtentF = 0.2                        ;-- decrease height of labelbar boxes
res@lbTitleString     = temp@long_name+" ["+temp@units+"];-- labelbar title
res@lbTitlePosition   = "Bottom"                 ;-- labelbar title position
res@lbTitleFontHeightF = 0.015                     ;-- labelbar title font size
res@lbBottomMarginF   = 0.1                        ;-- move the labelbar title up

res@mpDataBaseVersion = "MediumRes"            ;-- set map data base
res@mpMinLonF          = -10.6                    ;-- lon min. value
res@mpMinLatF          = 36.8                     ;-- lat min. value
res@mpMaxLonF          = 34.6                     ;-- lon max. value
res@mpMaxLatF          = 69.8                     ;-- lat max. value

```

# Time animation – vectors on contour (4/5)

```

res@vcLineArrowColor      = "grey30"                      ;-- streamline color
res@vcGlyphStyle          = "CurlyVector"                ;-- curly vectors
res@vcRefMagnitudeF       = 10                           ;-- define vector ref mag
res@vcRefLengthF          = 0.045                        ;-- define length of vec ref
res@vcRefAnnoOrthogonalPosF = -0.124                      ;-- move ref vec into plot
res@vcRefAnnoParallelPosF = 0.999                        ;-- move ref vec to the right into plot
res@vcLineArrowThicknessF = 2.0                          ;-- make vector lines thicker

res@tiMainFontHeightF     = 0.02                         ;-- main title font size
res@tiMainOffsetYF         = 0.06                         ;-- move title upward

;-- create plots, loop along time
ntimes = dimsizes(time)
do i=0,ntimes-1
  res@tiMainString = "Horizontal wind field~F29~~Z70~" +
getdate(time(i)) + "~N~~C~~Z60~[m/s] ~N~"           ;-- draw title
  plot  = gsn_csm_vector_scalar_map(wks,u(i,:,:),v(i,:,:),temp(i,:,:),res)
          ;-- vectors on contours map
  print("Plot time: "+i+" "+getdate((time(i))))
end do

```

# Time animation – vectors on contour (5/5)

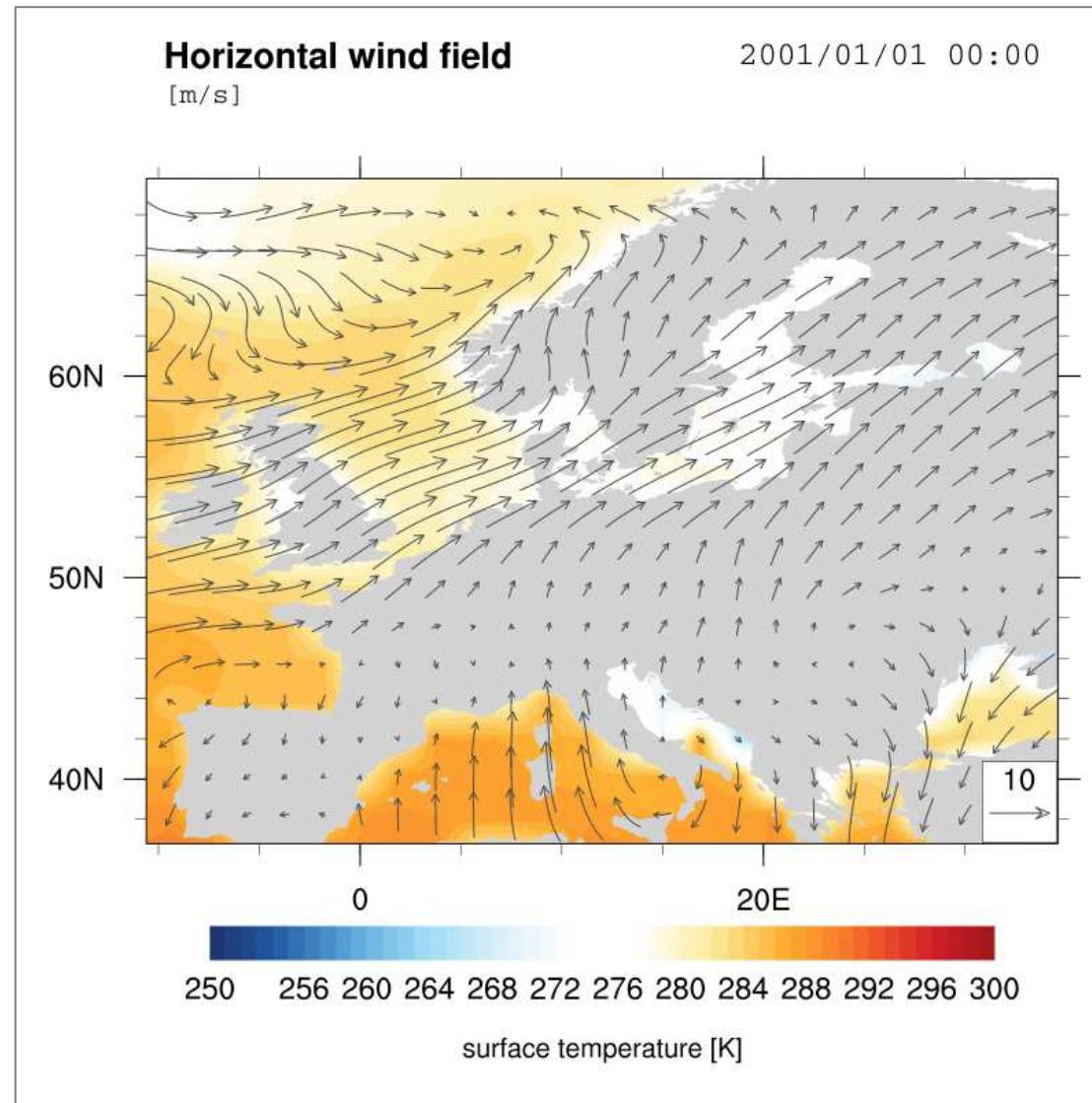
```
;-- convert the plot output from PNG to Animated GIF
print("Converting to GIF....")

system("convert -delay 50 part_VIII_animation_simple_vector_contour_map*.png
part_VIII_animation_simple_vector_contour_map.gif")

system("rm -rf part_VIII_animation_simple_vector_contour_map.*.png")

end
```

# Time animation of a contour plot



# Task: Animation – spinning globe

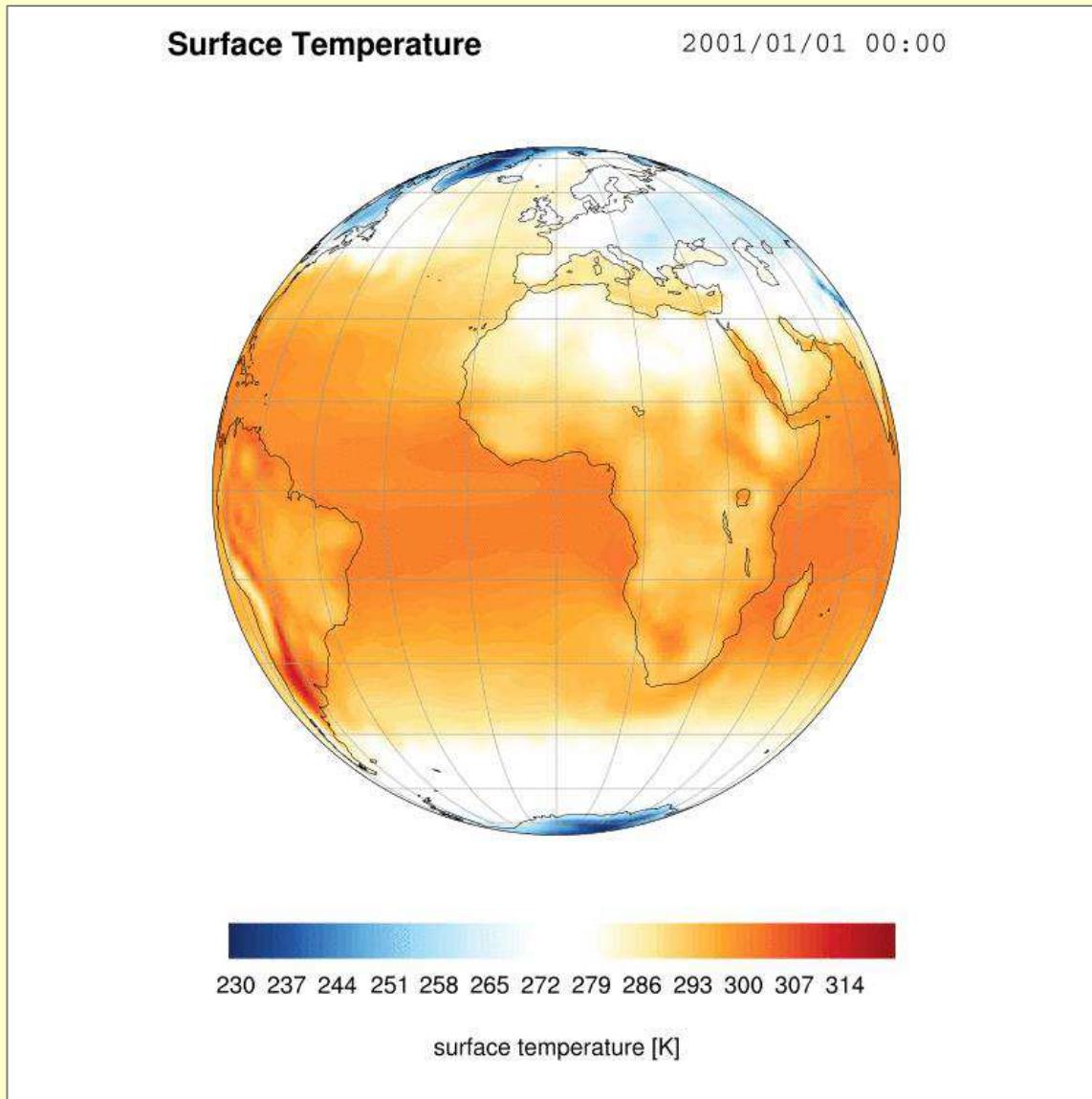
- Use the script **part\_VIII\_animation\_contour\_map.ncl**
- Copy it to your own directory and modify it
  - Interpolate the data to **half an hour time steps** (use CDOs)
  - Choose **Orthographic** projection.
  - Don't draw a box around the globe. Draw grid lines.
  - Set contour levels to **230-320 K**.
  - Rotate the globe **westward** in **360./ntimes steps**. Stop the rotation of the globe at **longitude=90°W**.
  - Change the delay of the frames to 25 (convert –delay 25 .....)

➤ **Running the time interpolation takes some time!**

## Hints:

- **cdo intntime,12 .....**
- **mpProjection**
- **mpPerimOn / mpGridAndLimbOn**
- **cn MinLevelValF / cnMaxLevelsValF**
- **mpCenterLonF**

# Task: Animation – spinning globe



# Task: Animation – spinning globe (1/5)

```
undef ("getDate")
function getDate(time)
local utc_date, year, mon, day, hours, mins, str_date
begin
    utc_date = cd_calendar(time, 0)          ;-- convert date to UT-referenced date
    year     = sprinti("%0.4i",tointeger(utc_date(:,0))) ;-- get year as integer
    mon      = sprinti("%0.2i",tointeger(utc_date(:,1))) ;-- get month as int
    day      = sprinti("%0.2i",tointeger(utc_date(:,2))) ;-- get day as integer
    hours    = sprinti("%0.2i",tointeger(utc_date(:,3))) ;-- get day as integer
    mins     = sprinti("%0.2i",tointeger(utc_date(:,4))) ;-- get day as integer
    str_date = year+"/"+mon+"/"+day+" "+hours+":"+mins ;-- yyyy/mm/dd HH:MM
    return(str_date)
end

;-----
;-- main script
;-----

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"

begin
```

# Task: Animation – spinning globe (2/5)

```
;-- interpolate to half hour time steps to prevent leaps
system("cdo intntime,12 $NCL_TUT/data/ECHAM5_OM_A1B_2001_0101-1001_2D.nc
tmp/tinterpolate_12.nc")

f      = addfile("tmp/tinterpolate_12.nc","r")
temp   = f->tsurf                                ;-- temperature data 1st time step
time   = f->time                                 ;-- time values

wks_type          = "png"                         ;-- plot output type
wks = gsn_open_wks(wks_type,"task_VIII_animation_spinning_globe")

res             = True
res@gsnMaximize = True                           ;-- maximize plot output
res@gsnAddCyclic = True                          ;-- don't add cyclic point
res@gsnRightString = ""                          ;-- don't draw right string
res@gsnLeftString = ""                           ;-- don't draw left string

res@cnFillPalette = "BlueWhiteOrangeRed"        ;-- choose color map
res@cnLevelSelectionMode = "ManualLevels"       ;-- use manual contour line levels
res@cnMinLevelValF = 230.0                        ;-- contour min. value
res@cnMaxLevelValF = 320.0                        ;-- contour max. value
res@cnLevelSpacingF = 1.0                         ;-- contour interval
```

# Task: Animation – spinning globe (3/5)

# Task: Animation – spinning globe (4/5)

```
res@tiMainFontHeightF      =  0.02                      ;-- main title font size
res@tiMainOffsetYF         =  0.06                      ;-- move title upward

;-- create plots, loop along time
ntimes = dimsizes(time)
incr = 360./ntimes
do i=0,ntimes-1
  rlon = 0. - (i*incr)
  if (rlon .gt. -90) then
    res@mpCenterLonF = rlon                            ;-- center lon position
  end if
  res@tiMainString        = "Surface Temperature~F29~~Z70~" +
getDate(time(i)) + "~N~" ;-- draw title
  plot = gsn_csm_contour_map(wks,temp(i,:,:,:),res)   ;-- create the plots
  print("Plot time: "+i+" "+getDate((time(i))))
end do
```

# Task: Animation – spinning globe (5/5)

```
;-- if wks_type = "ps": make sure that the PS file is closed  
  delete(wks)  
  
;-- convert the plot output from PNG to Animated GIF  
  print("Converting to GIF....")  
  
  system("convert -delay 75 task_VIII_animation_spinning_globe*.png  
task_VIII_animation_spinning_globe.gif")  
  
  system("rm -rf task_VIII_animation_spinning_globe.*.png")  
  
end
```