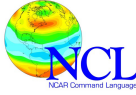


NCL



Basics IV

Grid Types

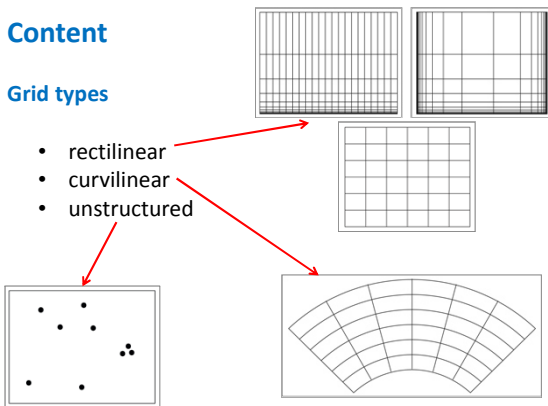
Karin Meier-Fleischer, DLR

NCL

Content

Grid types

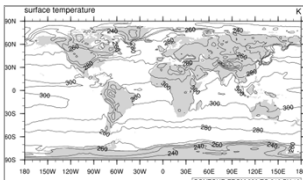
- rectilinear
- curvilinear
- unstructured



Karin Meier-Fleischer, DLR

NCL

Rectilinear grid type



```

begin
  f = addfile("$NCL_TUT/data/rectilinear_grid_2D.nc","r")
                                     ;-- open file
  t = f->tsurf(0,,:)
                                     ;-- define variable t

  wks = gsn_open_wks("x11", "plot_rectilinear");-- open workstation

  plot = gsn_csm_contour_map(wks, t, False) ; create the plot
  plot = gsn_csm_contour_map(wks, f->tsurf(0,,:), False)

end

```

Karin Meier-Fleischer, DLR

NCL

Curvilinear grid type (2D latitudes/longitudes)

```

begin
  f = addfile("$NCL_TUT/data/thetao_curvilinear_ocean.nc", "r")
  var = f->thetao(0,0,,:) ;-- define variable

  wks = gsn_open_wks("x11", "plot_curvilinear") ;-- open workstation
  plot = gsn_csm_contour_map(wks, var, res) ;-- create the plot
end

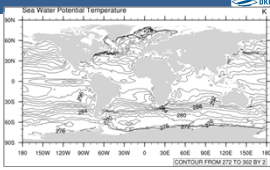
-- ERROR
(0) check_for_y_lat_coord: Warning: Data either does not contain a valid
latitude coordinate array or doesn't contain one at all.
(0) A valid latitude coordinate array should have a 'units' attribute equal to
one of the following values:
(0) 'degrees_north' 'degrees-north' 'degree_north' 'degrees north' 'degrees_N'
'Degrees_north' 'degree_N' 'degreeN' 'degreesN' 'deg north'
(0) check_for_lon_coord: Warning: Data either does not contain a valid longitude
coordinate array or doesn't contain one at all.
(0) A valid longitude coordinate array should have a 'units' attribute equal to
one of the following values:
(0) 'degrees_east' 'degrees-east' 'degree_east' 'degrees east' 'degrees_E'
'Degrees_east' 'degree_E' 'degreeE' 'degreesE' 'deg east'

```

Yuan Miao Fläschner, DLRZ 4

NCL

Curvilinear grid type



```

begin
  f = addfile("$NCL_TUT/data/thetao_curvilinear_ocean.nc", "r")
  ;-- open file

  var = f->thetao(0,0,,:) ;-- define variable
  lat2d = f->lat ;-- get 2D latitude array
  lon2d = f->lon ;-- get 2D longitude array

  wks = gsn_open_wks("x11", "plot_curvilinear") ;-- open workstation

  res = True ;-- set resource object
  res@gsnAddCyclic = False ;-- don't add cyclic point
  res@sfXArray = lon2d ;-- longitude grid cell center
  res@sfYArray = lat2d ;-- latitude grid cell center

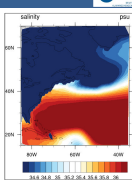
  plot = gsn_csm_contour_map(wks, var, res) ;-- create the plot
end

```

Yuan Miao Fläschner, DLRZ 5

NCL

Unstructured grid type



```

begin
  f = addfile("$NCL_TUT/data/triangular_grid_ICON.nc", "r")
  var = f->S(0,0,:) ;-- S(time,depth,ncells)

  rad2deg = 45./atan(1.) ;-- radians to degrees
  x = f->clon * rad2deg ;-- cell center, lon (degrees)
  y = f->clat * rad2deg ;-- cell center, lat (degrees)

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

  res = True ;-- don't add cyclic point
  res@gsnAddCyclic = False ;-- don't add cyclic point
  res@gsnMaximize = True ;-- maximize plot output
  res@cnFillOn = True ;-- contour fill
  res@cnLinesOn = False ;-- manual contour levels
  res@cnLevelSelectionMode = "ManualLevels" ;-- set manual contour levels
  res@cnMinLevelValF = 34.5 ;-- set min contour level
  res@cnMaxLevelValF = 36.1 ;-- set max contour level
  res@cnLevelSpacingF = 0.1 ;-- set increment

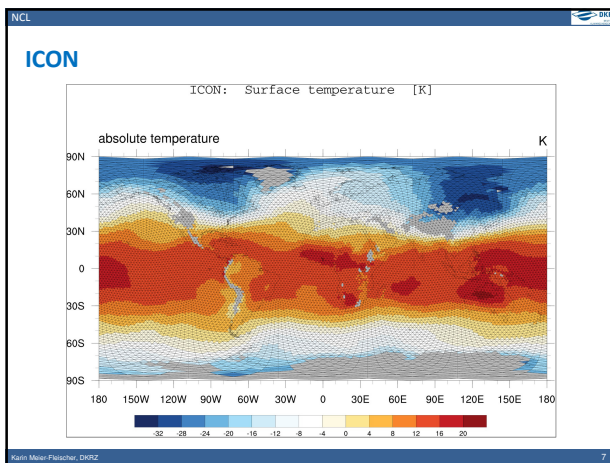
  res@mpMinLonF = -85.0 ;-- sub-region minimum longitude
  res@mpMaxLonF = -35.0 ;-- sub-region maximum longitude
  res@mpMinLatF = 15.0 ;-- sub-region minimum latitude
  res@mpMaxLatF = 70.0 ;-- sub-region maximum latitude

  res@sfXArray = x ;-- longitude grid cell center
  res@sfYArray = y ;-- latitude grid cell center

  plot = gsn_csm_contour_map(wks, var, res) ;-- create plot, but don't draw it
end

```

Yuan Miao Fläschner, DLRZ 6



NCL

ICON plot script (1/8)

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

;-- Procedure: add_lab      Create a new and add labelbar      --;
;-----
undef("add_lab")
procedure add_lab(wks,map,cmap,levels)
begin
  getvalues map
    "vpWidthF"      : vpw      ;-- retrieve the viewport width
    "vpHeightF"     : vph      ;-- retrieve the viewport height
    "vpXF"          : vpx      ;-- retrieve the viewport x pos
    "vpYF"          : vpy      ;-- retrieve the viewport y pos
end getvalues

;-- set the labelbar labels, width, height and position
labs = levels+" " ;-- labelbar labels
nlevs = dimsizes(labs) ;-- number of labels
lbwidth = 0.7 ;-- labelbar width
lbheight = 0.08 ;-- labelbar height
lbx = ((1.0-(lbwidth+vpw))/2) + vpx/2 ;-- labelbar x position
lby = ((1.0-(vpy-vph))/2) - 0.13 ;-- labelbar y position

;-- set global labelbar resources
lbres = True
lbres@gsnFrame = False ;-- don't advance frame
lbres@vpWidthF = lbwidth ;-- width of labelbar
lbres@vpHeightF = lbheight ;-- height of labelbar
lbres@vpXF = lbx ;-- labelbar x-position
lbres@vpYF = lby ;-- labelbar y-position
```

Karin Meier-Flaschner, DLRZ

NCL

ICON plot script (2/8)

```
lbres@lbPerimOn = False ;-- no label bar box
lbres@lbOrientation = "Horizontal" ;-- orientation
lbres@lbLabelFontHeightF = 0.010 ;-- label font height
lbres@lbLabelAlignment = "InteriorEdges" ;-- where to label
lbres@lbMonoFillPattern = True ;-- fill solid
lbres@lbFillColor = cmap ;-- use colors
gsn_labelbar_ndc(wks,nlevs+1,labs,lbx,lby,lbres) ;-- draw transparent labelbar

end

;=====
;-- MAIN PROGRAM
;=====
begin
  starttt = get_cpu_time()

  DataFileName = "$NCL_TUT/data/ICON/ta_gs_850.nc" ;-- data file
  gridinfoFile = "$NCL_TUT/ICON/grids/z2b4_smip.nc" ;-- grid info file
  VarName = "ta" ;-- variable name

  File = addfile(DataFileName, "r") ;-- add data file
  GridInfoFile = addfile(gridinfoFile, "r") ;-- add grid file (not contained
;-- in data file!!!)
  var = File->$VarName$(time[0,lev[0,ncells[:]) ;-- set variable
  var = var - 273.15 ;-- convert to degrees Celsius

  title = "ICON: Surface temperature [" + var@units + "]" ;-- title string
```

Karin Meier-Flaschner, DLRZ

```

NCL
ICON plot script (3/8)

varMin = 270.0 ;-- data minimum
varMax = 300.0 ;-- data maximum
varInt = 2.5 ;-- data increment
lon_min = -180.0 ;-- minimum longitude
lon_max = 180.0 ;-- maximum longitude
lat_min = -90.0 ;-- minimum latitude
lat_max = 90.0 ;-- maximum latitude

;-- define the x-, y-values and the polygon points
;-- radians to degrees
rad2deg = 45./atan(1.)

x = GridInfoFile->clon * rad2deg ;-- cell center, lon
y = GridInfoFile->clat * rad2deg ;-- cell center, lat

x10 = "lon" ;-- set named dimension lon
y10 = "lat" ;-- set named dimension lat
xunits = "degrees_east" ;-- set lon units
yunits = "degrees_north" ;-- set lat units

vlon = GridInfoFile->clon_vertices * rad2deg ;-- cell longitude vertices
vlon = where(vlon.lt.0, vlon + 360, vlon) ;-- longitude: 0-360
vlat = GridInfoFile->clat_vertices * rad2deg ;-- cell latitude vertices
nv = dimsizes(vlon(0,:)) ;-- number of points in polygon

;-- print some information to stdout
print("")
print("Data Min: " + min(var) + " Max: " + max(var))
print("Data longitude Min: " + min(vlon) + " Max: " + max(vlon))

```

```

NCL
ICON plot script (4/8)

print("Cell points: " + nv)
print("")

;-- open workstation
;-- plot output type
wtype = "png" ;-- plot output type
wtype@wkWidth = 2500 ;-- set workstation width in pixel
wtype@wkHeight = 2500 ;-- set workstation height in pixel
wks = gsn_open_wks(wtype, "plot_ICON_optimized_ts_x") ;-- open a workstation

;-- set resources
res = True
res@gsnDraw = False ;-- don't draw the plot
res@gsnFrame = False ;-- don't advance the frame
res@gsnMaximize = True ;-- maximize plot output

res@cnFillPalette = "BlueWhiteOrangeRed" ;-- choose colormap
res@cnLinesOn = False ;-- don't draw contour lines
res@cnInfoLabelOn = False ;-- switch off contour info label
res@cnFillOn = True ;-- contour fill on
res@cnFillMode = "CellFill" ;-- set fill mode
res@sfXArray = x ;-- transform x to mesh scalar field
res@sfYArray = y ;-- transform y to mesh scalar field
res@sfXCellBounds = vlon ;-- needed if set cnFillMode = "CellFill"
res@sfYCellBounds = vlat ;-- needed if set cnFillMode = "CellFill"
res@lbLabelBarOn = False ;-- don't draw a labelbar yet

```

```

NCL
ICON plot script (5/8)

res@mpFillOn = False ;-- fill map grey
res@mpDataBaseVersion = "MediumRes" ;-- map resolution
res@mpMinLonF = lon_min ;-- sub-region minimum longitude
res@mpMaxLonF = lon_max ;-- sub-region maximum longitude
res@mpMinLatF = lat_min ;-- sub-region minimum latitude
res@mpMaxLatF = lat_max ;-- sub-region maximum latitude
res@mpGreatCircleLinesOn = False ;-- important: v6.2.0 False !!

;-- create the contour plot, but don't draw it (we need to get
;-- the colors of the data values)
plot = gsn_csm_contour_map(wks, var, res)

;-- to plot the filled polygons we need the data levels and their colors
getvalues plot@contour
  "cnLevels" : levels ;-- retrieve the levels from plot@contour
  "cnFillColors" : colors ;-- retrieve the colors from plot@contour
end getvalues

nlevels = dimsizes(levels) ;-- number of levels

;-- clear the plot, but keep all the information
plot = setColorContourClear(plot, min(var), max(var))

```

NCL

ICON plot script (6/8)

```

;-- add a labelbar
add_lab(wks,plot,colors,levels) ;-- add labelbar

;-- create color array for triangles
ntri = dimsizes(y) ;-- number of triangles
gscolors = new(ntri,string) ;-- create color array (type of colors integer!)
gscolors = "gray" ;-- initialize to black (for missing colors)

;-- set resources for the triangles (polygons)
pres = True ;-- turn on edges
pres%gsEdgesOn = True ;-- turn on edges
pres%gsFillIndex = 0 ;-- solid fill

;-- set color for data less than given minimum value var_min
vlow = ind(var .lt. levels(0)) ;-- get the indices of values less levels(0)
if (.not. all(ismissing(vlow))) then
  gscolors(vlow) = colors(0) ;-- choose color
  ntri_calc = dimsizes(vlow) ;-- number of triangles
end if

;-- set colors for all cells in between var_min and var_max
do i = 1, dimsizes(levels) - 1
  vind = ind(var .ge. levels(i-1) .and. var .lt. levels(i)) ;-- get the indices of 'middle' values

```

Kurt Mearns, DMRZ

NCL

ICON plot script (7/8)

```

if (.not. all(ismissing(vind))) then
  gscolors(vind) = colors(i) ;-- choose the colors
  ntri_calc = ntri_calc + dimsizes(vind) ;-- number of triangles
end if
end do

;-- set color for data greater than given maximum var_max
nc=dimsizes(colors)-1 ;-- get the number of colors minus one
nl=dimsizes(levels)-1 ;-- get the number of levels minus one
vhigh := ind(var .gt. levels(nl)) ;-- get indices of values greater levels(nl)
if (.not. all(ismissing(vhigh))) then
  gscolors(vhigh) = colors(nc) ;-- choose color
  ntri_calc = ntri_calc + dimsizes(vhigh) ;-- number of triangles
end if
print("----> triangles calculated: " + ntri_calc)

;-- attach all the triangles using the list of colors
pres%gsColors = gscolors ;-- set colors for polygons
pres%gsSegments = ispan(0,dimsizes(var) * 3,3) ;-- assign segments array
polygon = gsn_add_polygon(wks,plot,ndtooned(vlon),ndtooned(vlat),pres) ;-- draw all triangles

;-- write title on top of the plot
tires = True ;-- text resources
tires%txFontHeightF = 0.015 ;-- text font size
tires%txJust = "CenterCenter" ;-- text justification
tires%txFont = "courier" ;-- text font

```

Kurt Mearns, DMRZ

NCL

ICON plot script (8/8)

```

xl = 0.5 ;-- text x position
yb = 0.85 ;-- text y position
gsn_text_ndc(wks,title,xl,yb, tires) ;-- plot title string on top

;-- do the complete plot
draw(plot) ;-- draw the map
frame(wks) ;-- advance the frame

;-- print elapsed CPU time
endt = get_cpu_time()
print("----> Used CPU time: " + (endt-startt) + "s")
print("----> Number of missing values: " + num(ismissing(var)))

end

```

Kurt Mearns, DMRZ