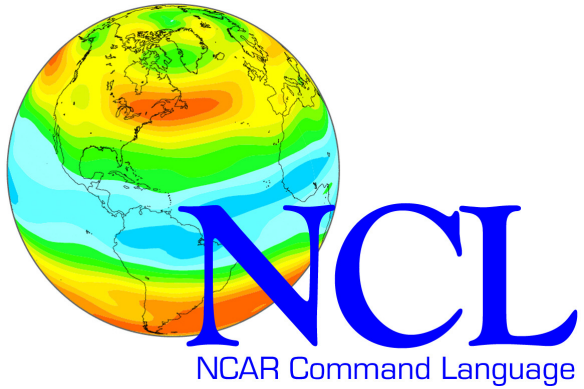
Introduction to NCL Graphics

Vectors and Overlays

Part III in the series

November 10, 2014

Mary Haley



CISL NESL NCAR NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

Sponsored by the National Science Foundation

Notes

- Third in a series of five (?) intro lectures on NCL Graphics
- Two more webinars: *panels* & *primitives* – *Nov 12*
- Lectures loosely based ones used in NCL Workshops
- Geared towards new users of NCL. . .but with tips for advanced users

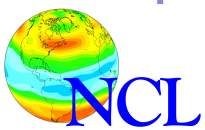
Goals for this series of lectures

- Get comfortable with creating NCL graphics
- Learn common things people do with NCL graphics
- Get tips for editing, debugging, creating publication-quality graphics
- Get questions answered

Assumptions

You have familiarity with:

- basic NCL language features
- NCL array syntax
- reading data off a NetCDF file using NCL
- basic structure of an NCL graphics script
- NCL/NetCDF data model
- UNIX

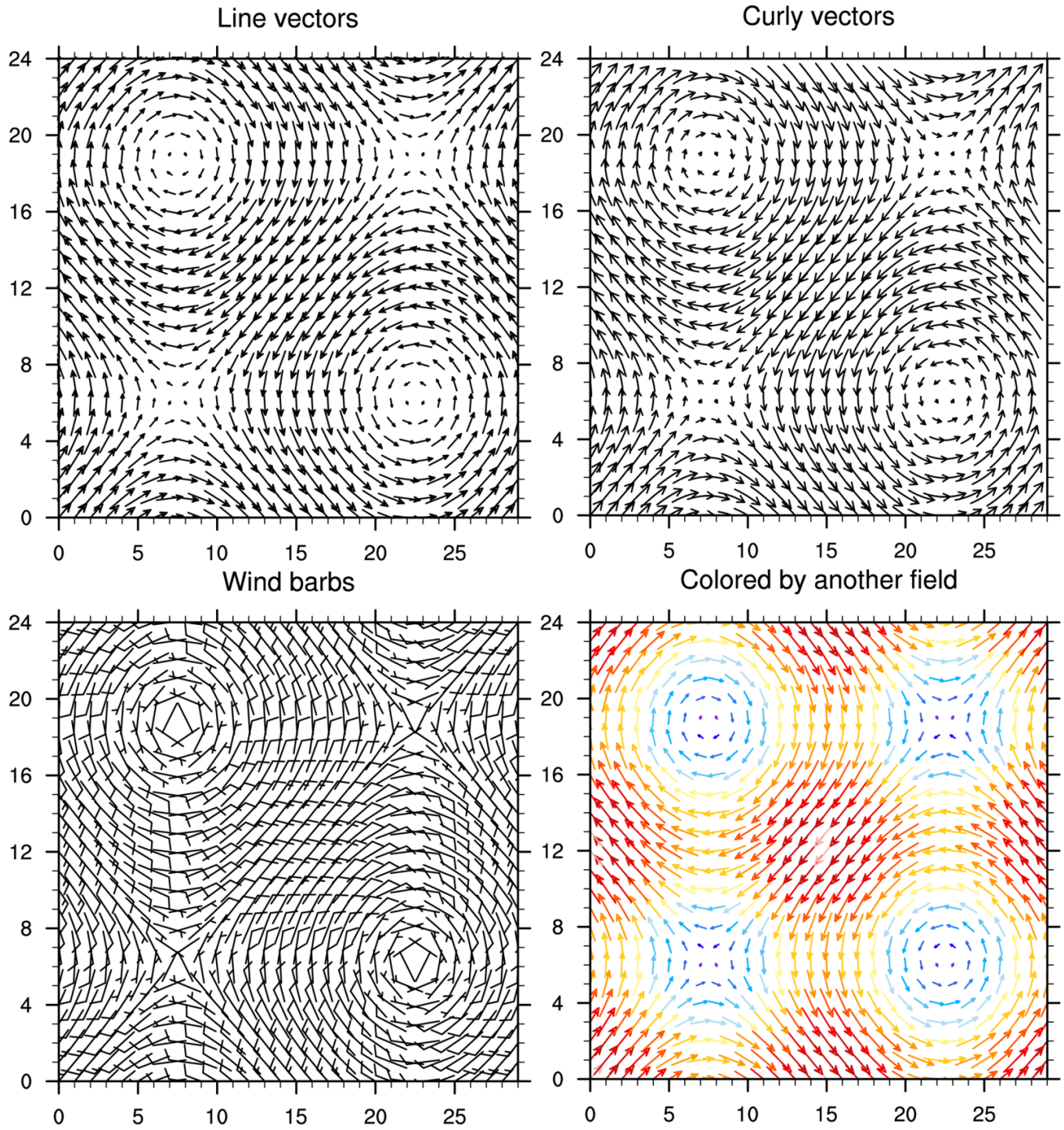


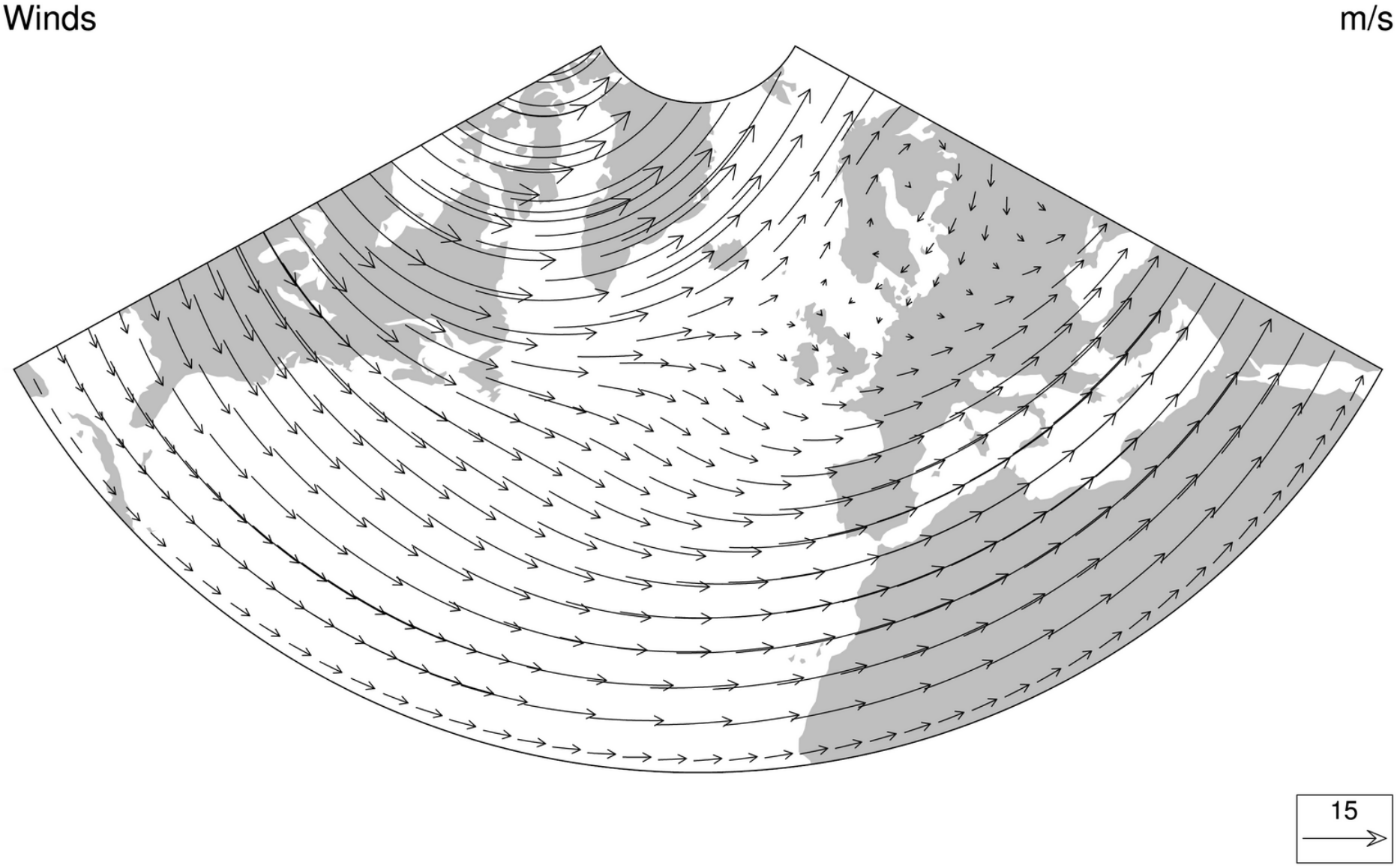
NCL Graphics topics for this lecture

- Gallery
- Description of NCL vector plots
- Line-by-line vector plot example
- Demo
- Description of overlay plots
- Line-by-line overlay plot example
- Demo

Vector types

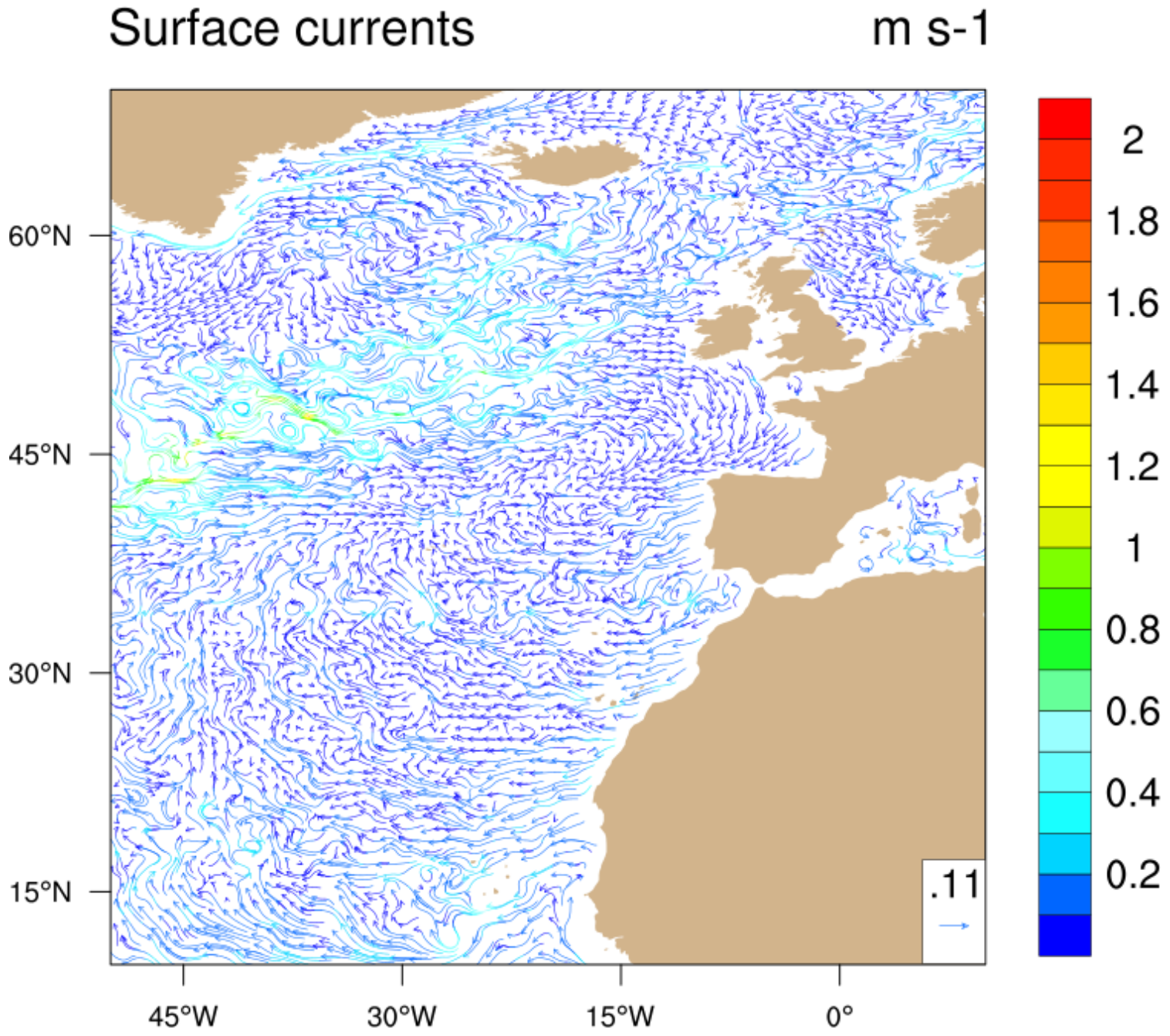
1. Line
2. Curly
3. Wind barb



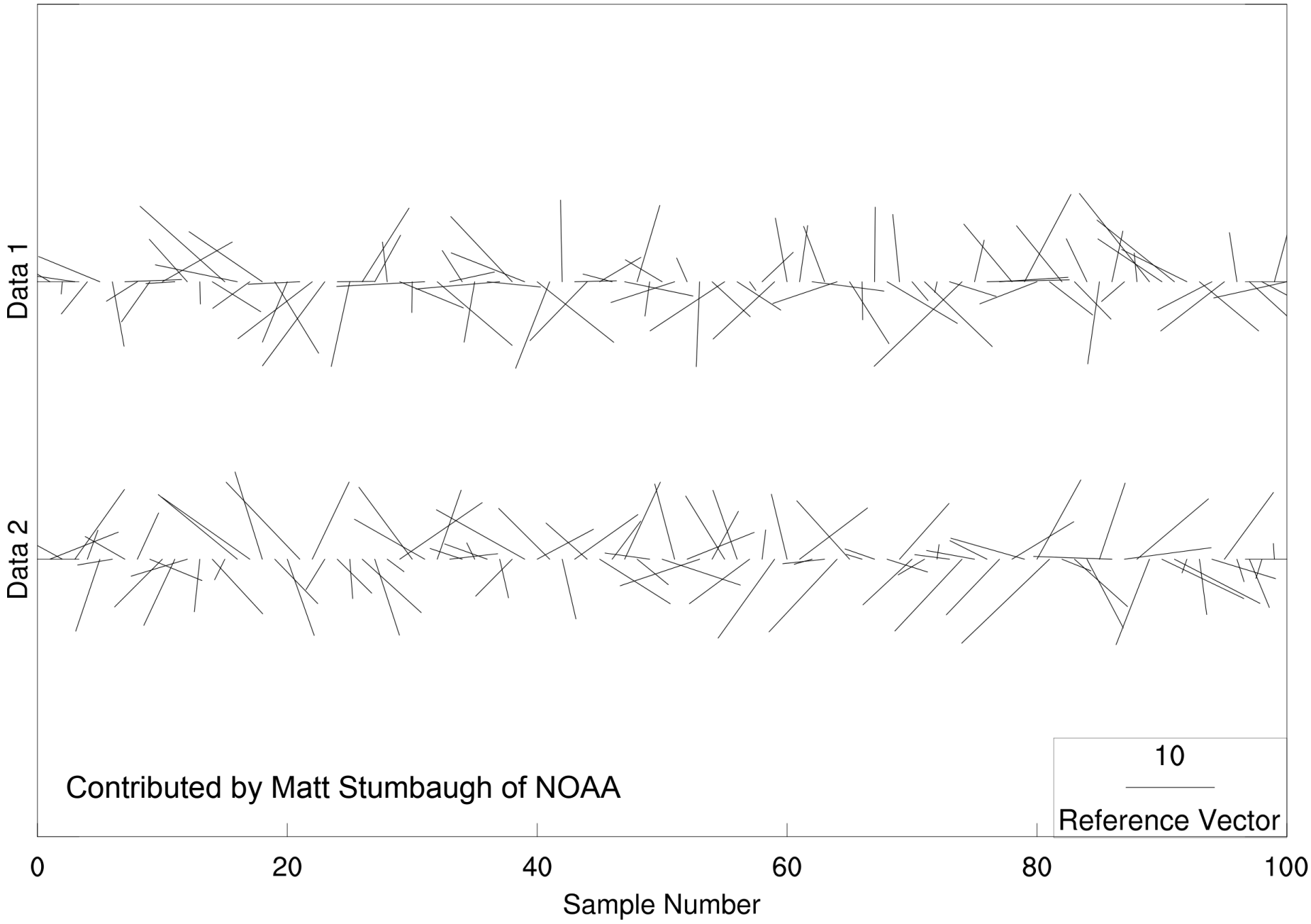


“masked” lambert conformal plot

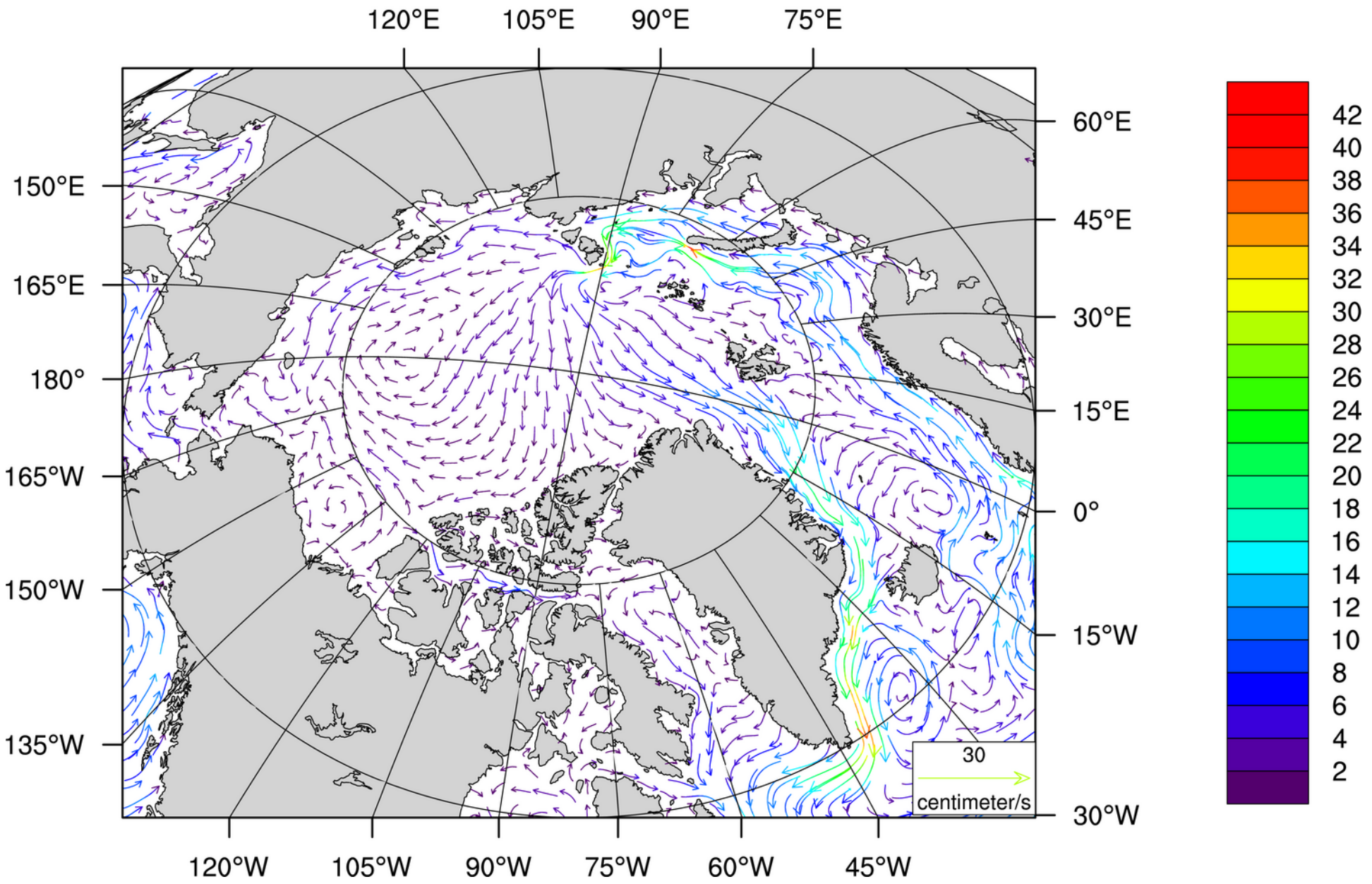
vozocrtx / vomecrty



Stick Plot

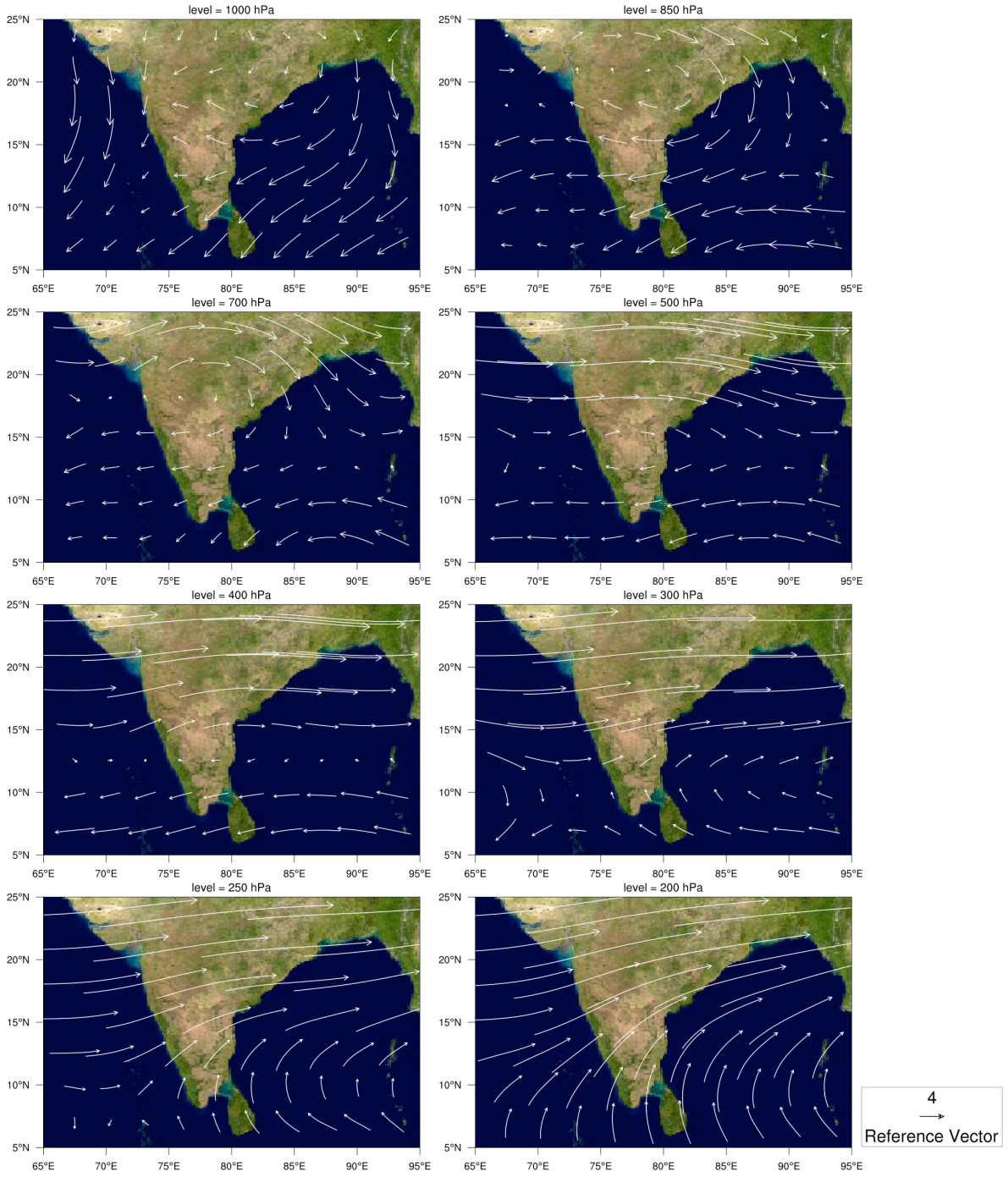

Data 1
Data 2
Contributed by Matt Stumbaugh of NOAA
10
Reference Vector
0
20
40
60
80
100
Sample Number

Curly vectors colored by magnitude

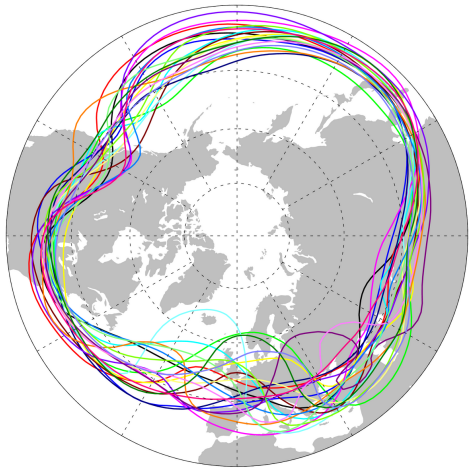


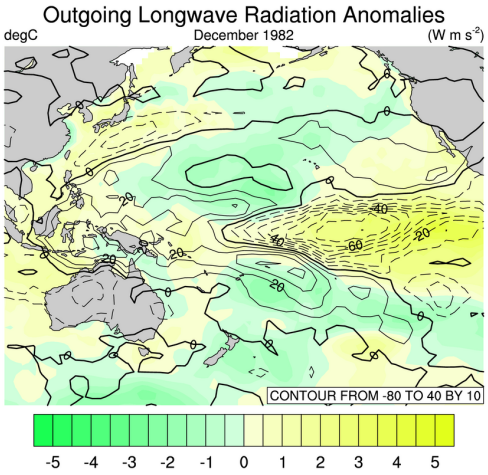
Courtesy Dave Brown, NCAR/CISL

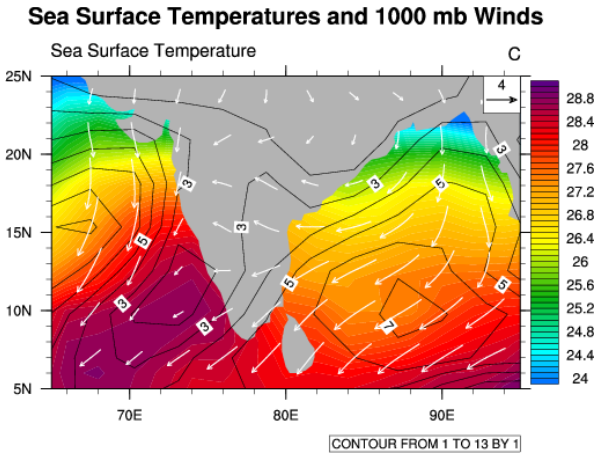
Zonal wind (m/2)

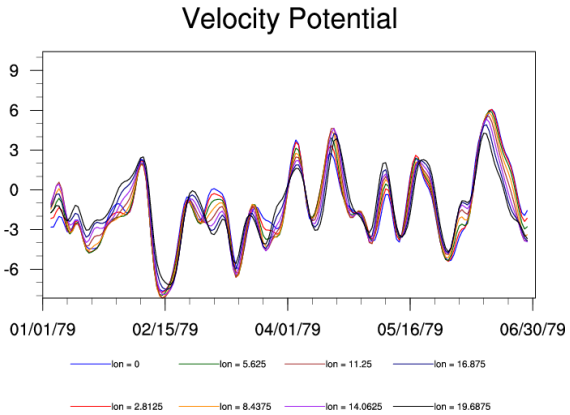


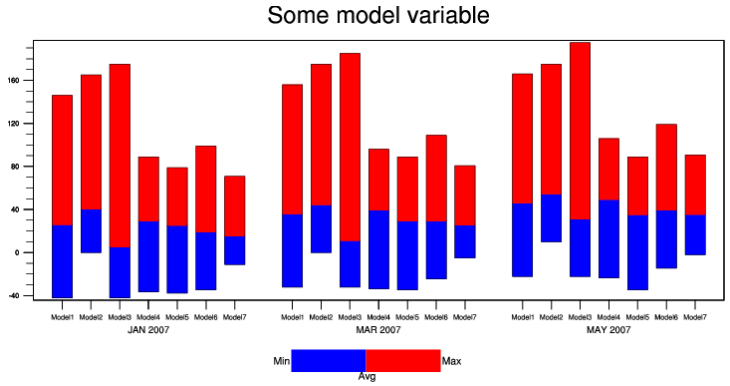
Overlay plots are plots that are drawn on top of other plots.

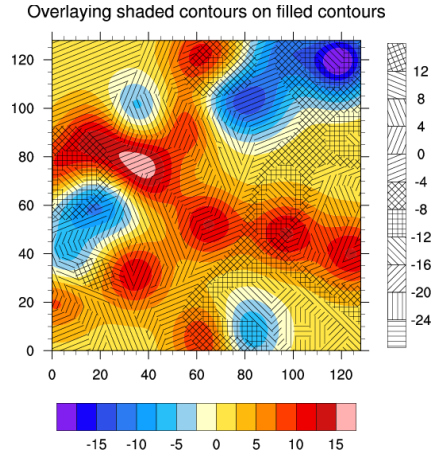


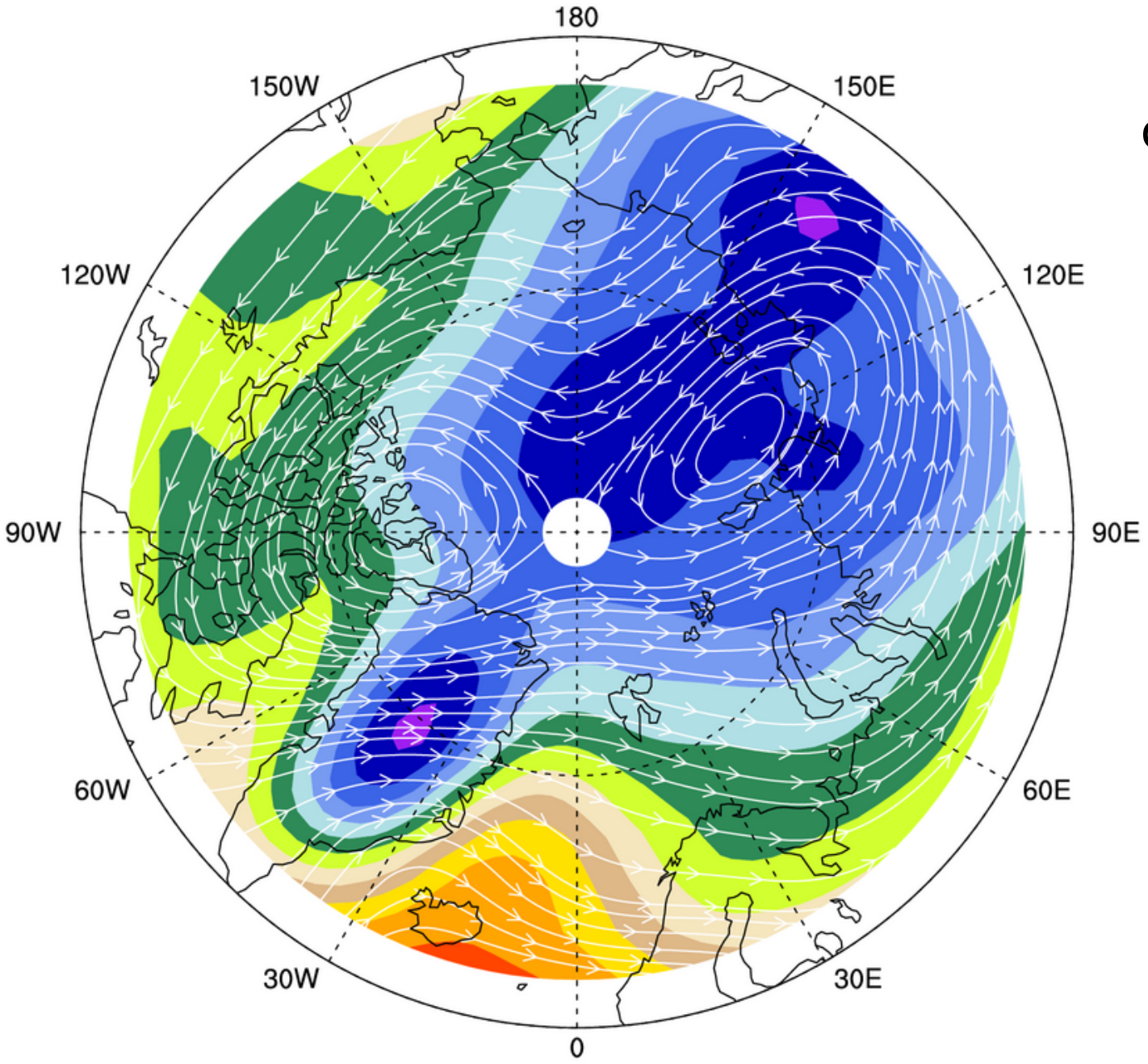




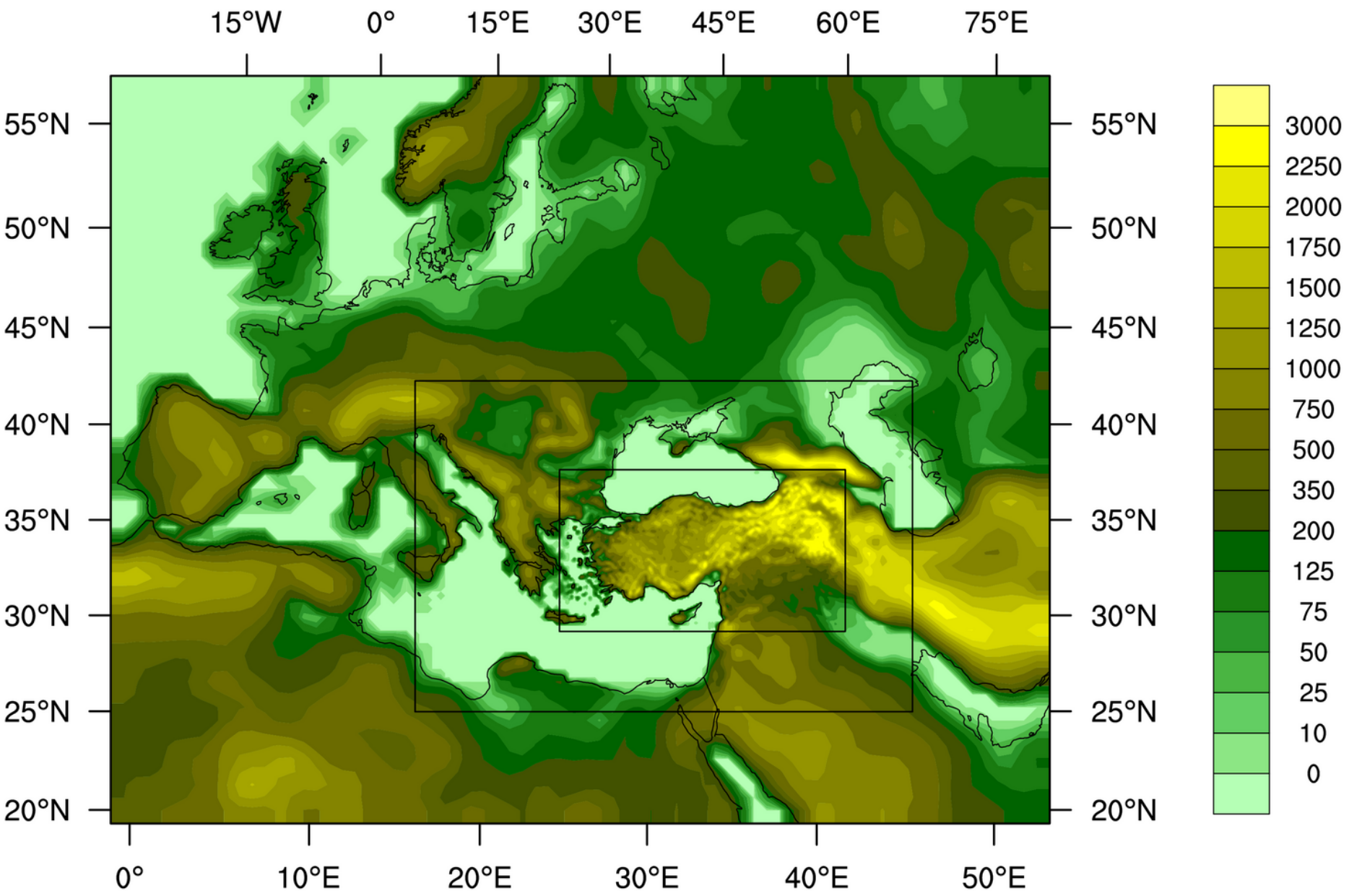






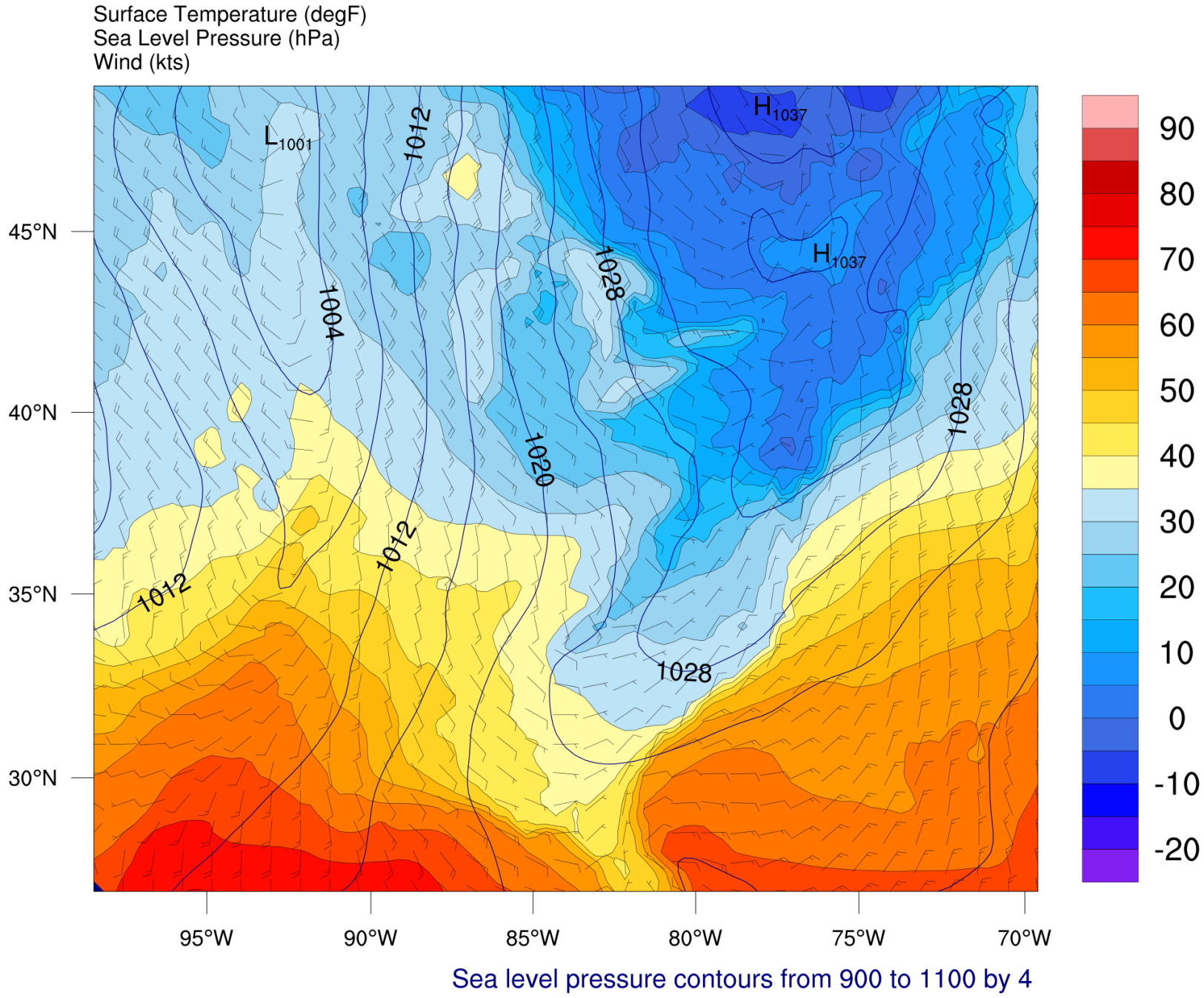


Streamlines
over contours

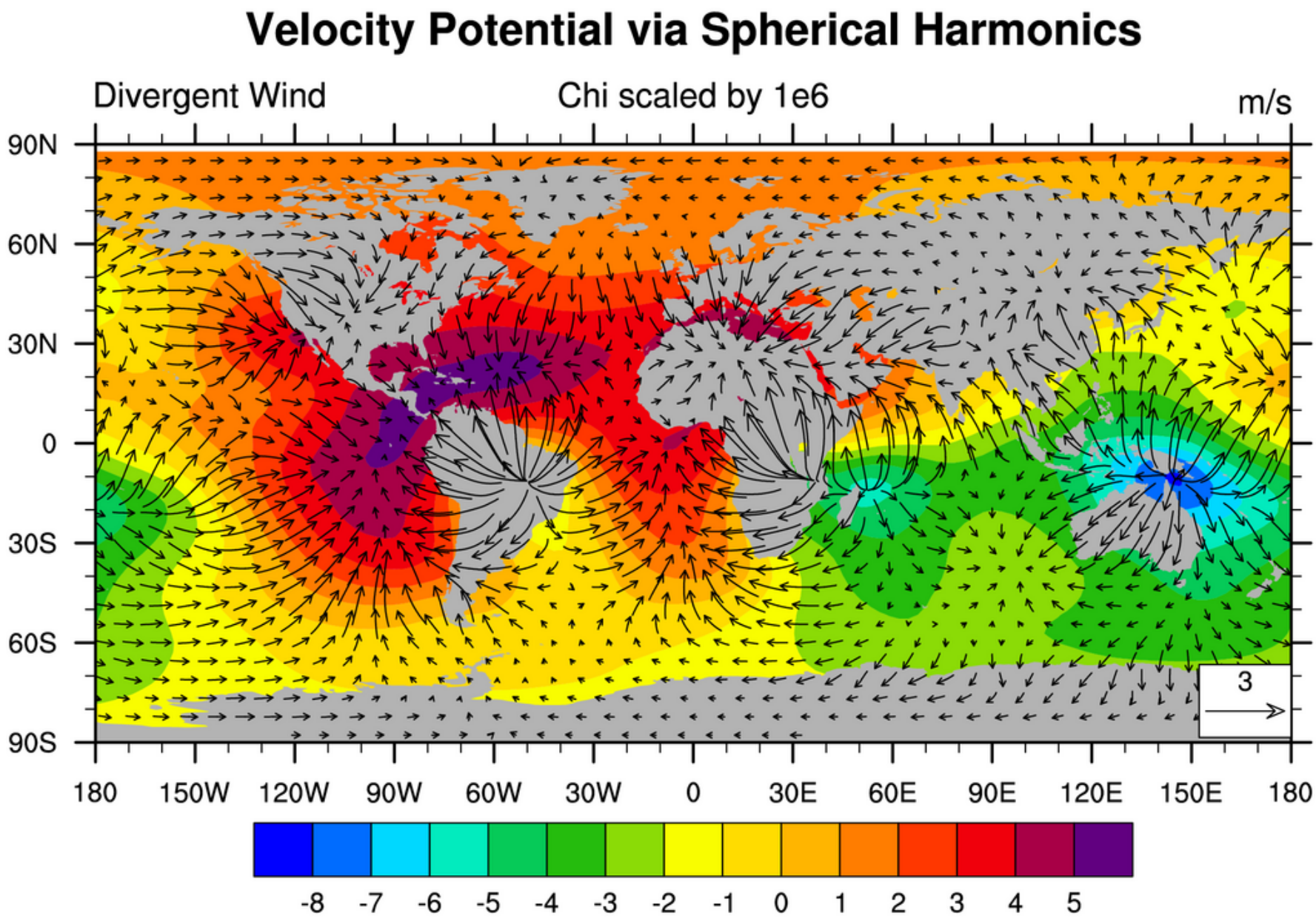


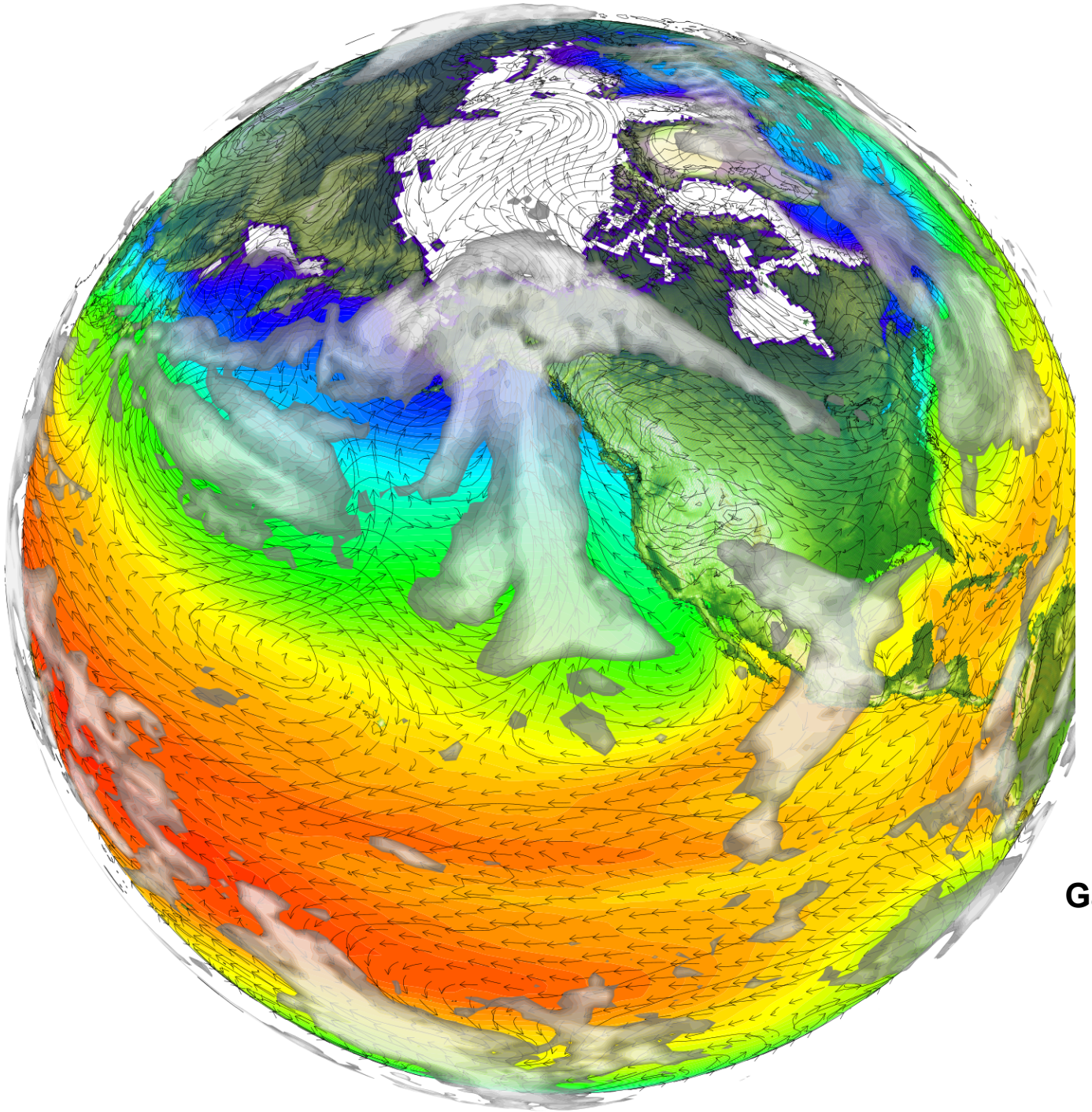
Ufuk Turuncoglu, ITU
Turkey Climate Change Scenarios

wrfout_d01_2005-12-14_13:00:00.nc



Multiple overlays (contours and vectors)





CCSM4 data
Six fields overlaid:

Ice thickness
(filled contours)

Sea surface
temperature
(filled contours)

Topo map
(filled contours)

Sea level pressure
(line contours)

UV winds

Vertically-integrated
clouds (partially
transparent
filled contours)

Gary Strand, NCAR/CGD

NCL Graphics topics for this lecture

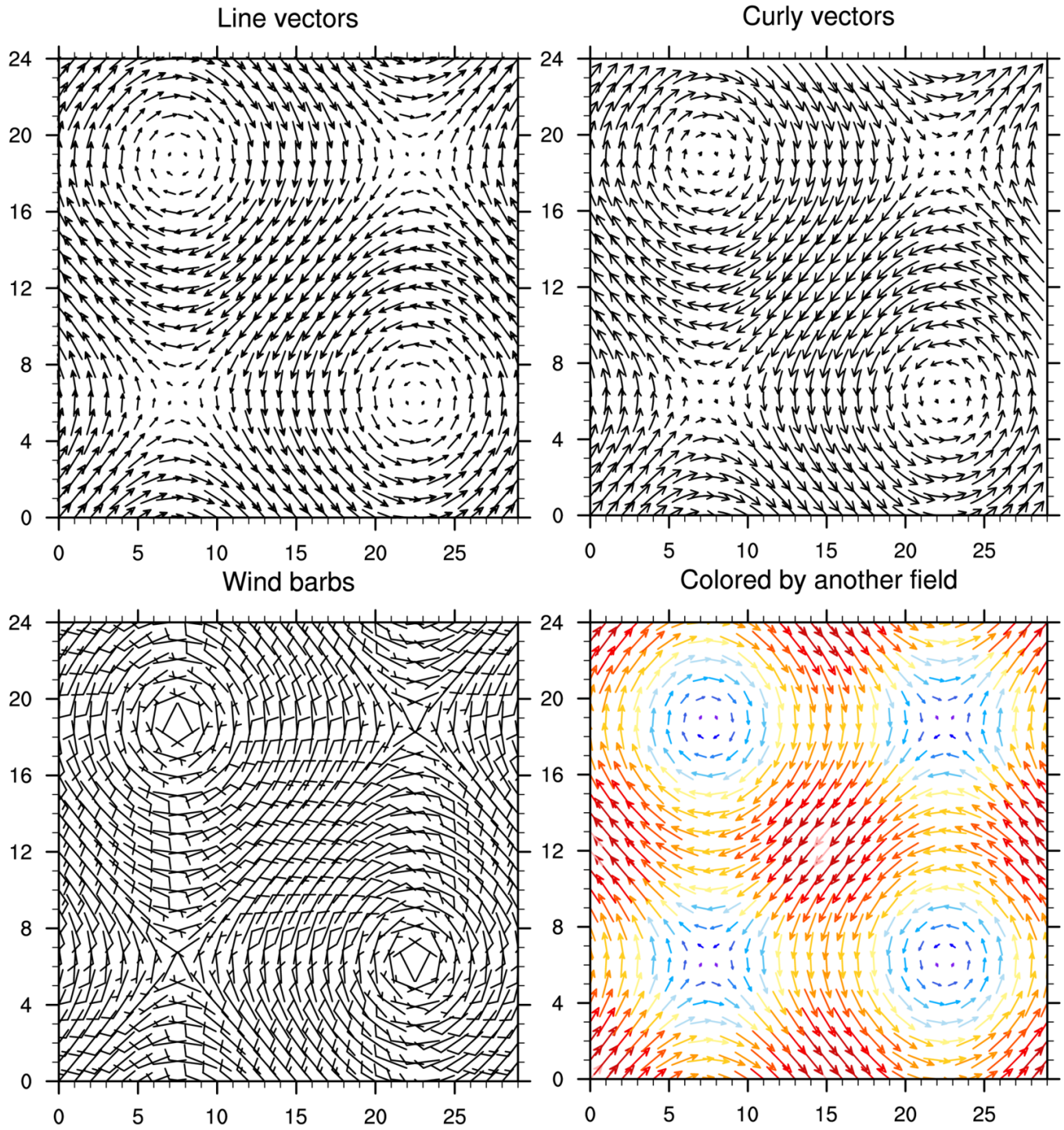
- Gallery
- Description of NCL vector plots
- Line-by-line vector plot example
- Demo
- Description of NCL overlay plots
- Line-by-line overlay plot example
- Demo

Vector plots

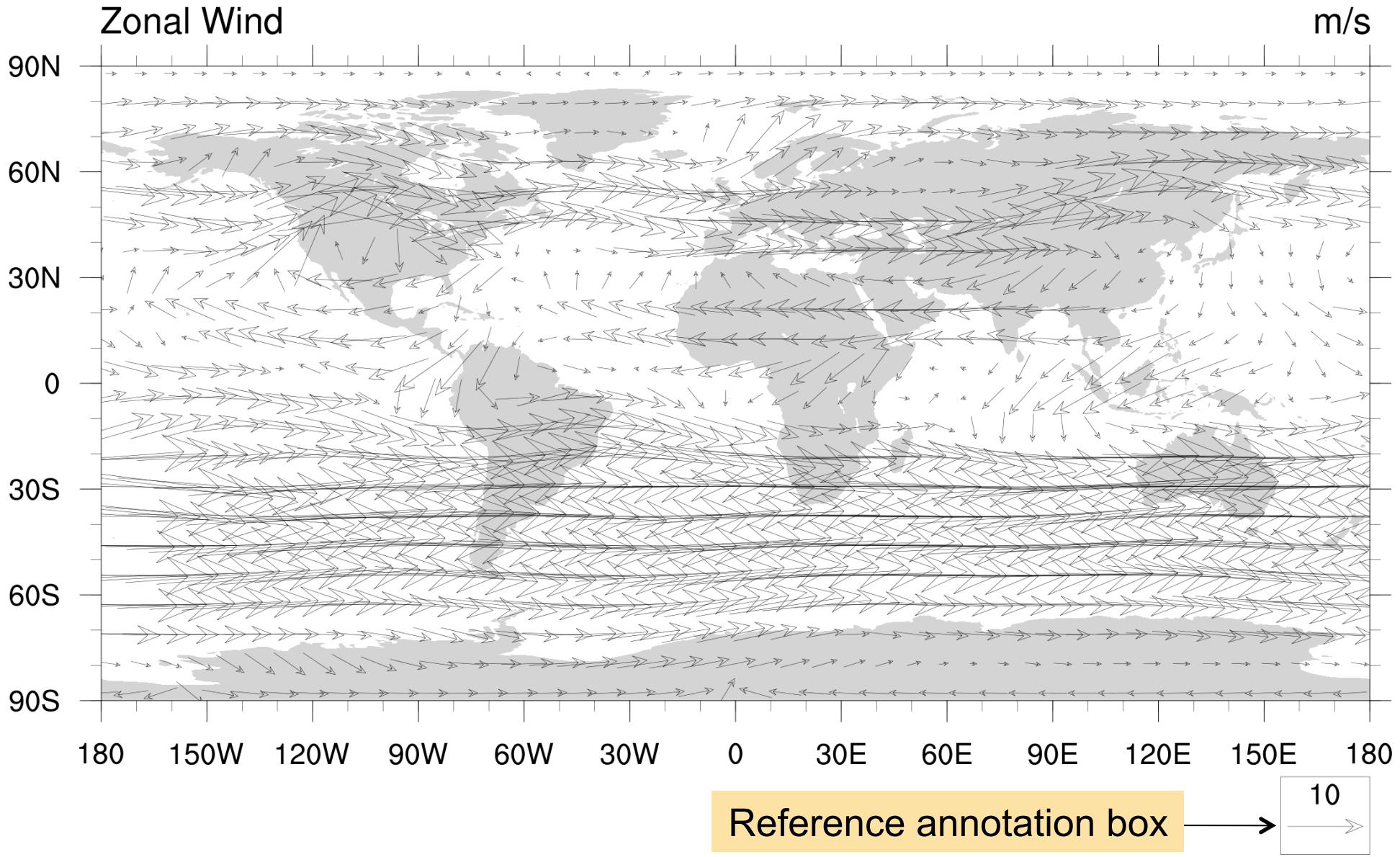
- Requires U, V two-dimensional arrays of the same size
- Includes a “reference annotation” box
- Can be colored by magnitude or another field
- Line, curly, wind barbs available
- When you call gsn_csm_*xxxx*_map, this is an overlay plot

Vector types

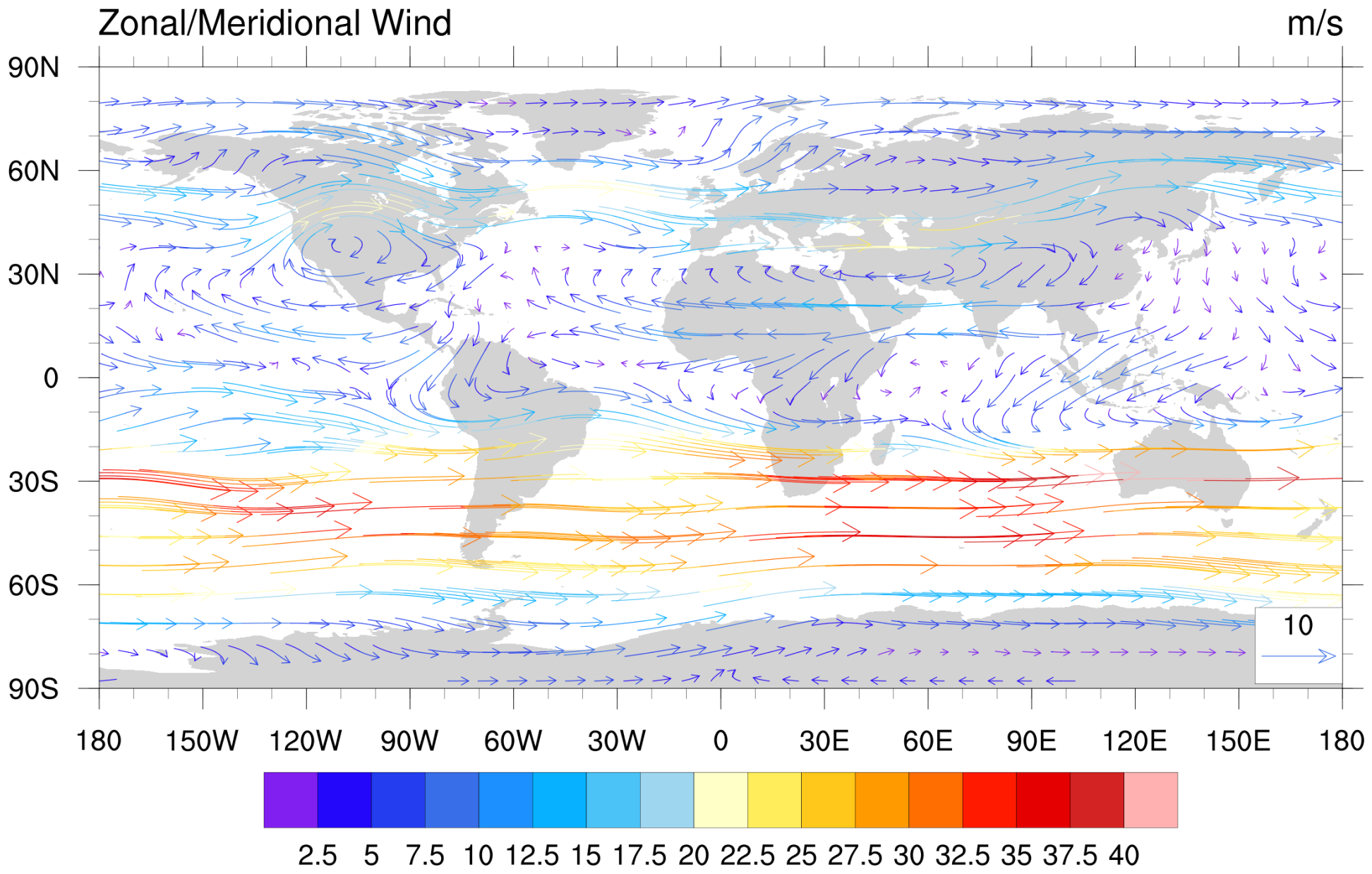
1. Line
2. Curly
3. Wind barb
4. Colored by another field



Sample line vector over map plot

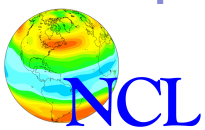


Sample of curly vectors colored by magnitude



NCL Graphics topics for this lecture

- Gallery
- Description of NCL vector plots
- Line-by-line vector plot example
- Demo
- Description of NCL overlay plots
- Line-by-line overlay plot example
- Demo



Example *vector1a.ncl*

- Generate dummy (10 x 10) U,V arrays
- No plot resources set
- gsn_csm_vector
- Scripts can be found at:

http://www.ncl.ucar.edu/Training/Webinars/NCL_Graphics/VectorOverlayDemo/

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

Optional in NCL V6.2.0 and later

```
begin
;---Generate dummy u, v arrays
  npts = 10
  ii   = ispan(10,10*npts,10)
  u    = conform_dims((/npts,npts/),ii,0)
  v    = conform_dims((/npts,npts/),ii,1)

;---Open an X11 window
  wks = gsn_open_wks("x11","vector1a")

;---Call the plotting function
  vector = gsn_csm_vector(wks,u,v,False)
end
```

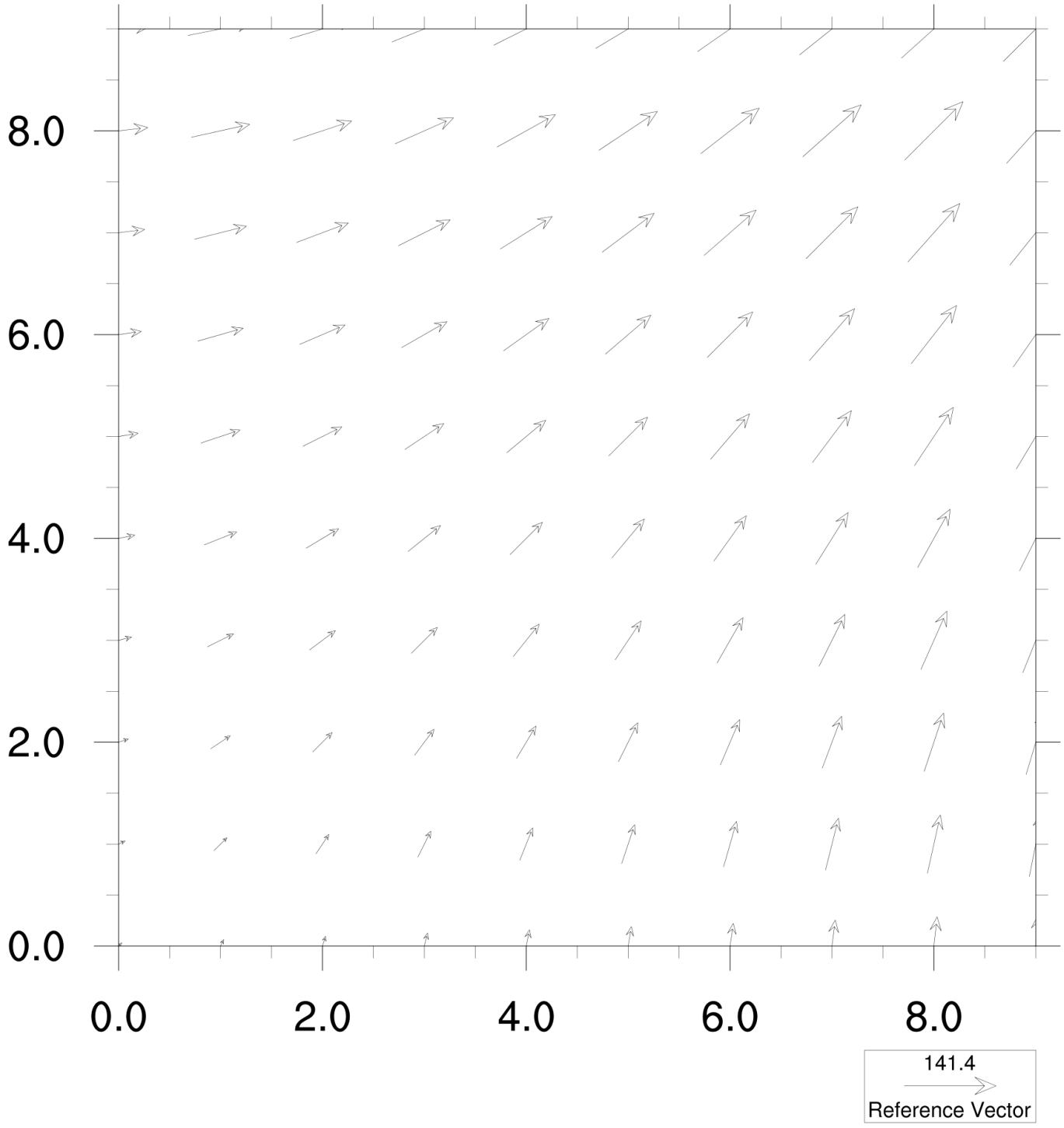
```
;---Generate dummy u, v arrays
n = 10
i = ispan(10,100,n)
u = conform_dims((/n,n/),i,0)
v = conform_dims((/n,n/),i,1)
```

u

100	100	100	100	100	100	100	100	100	100
90	90	90	90	90	90	90	90	90	90
80	80	80	80	80	80	80	80	80	80
70	70	70	70	70	70	70	70	70	70
60	60	60	60	60	60	60	60	60	60
50	50	50	50	50	50	50	50	50	50
40	40	40	40	40	40	40	40	40	40
30	30	30	30	30	30	30	30	30	30
20	20	20	20	20	20	20	20	20	20
10	10	10	10	10	10	10	10	10	10

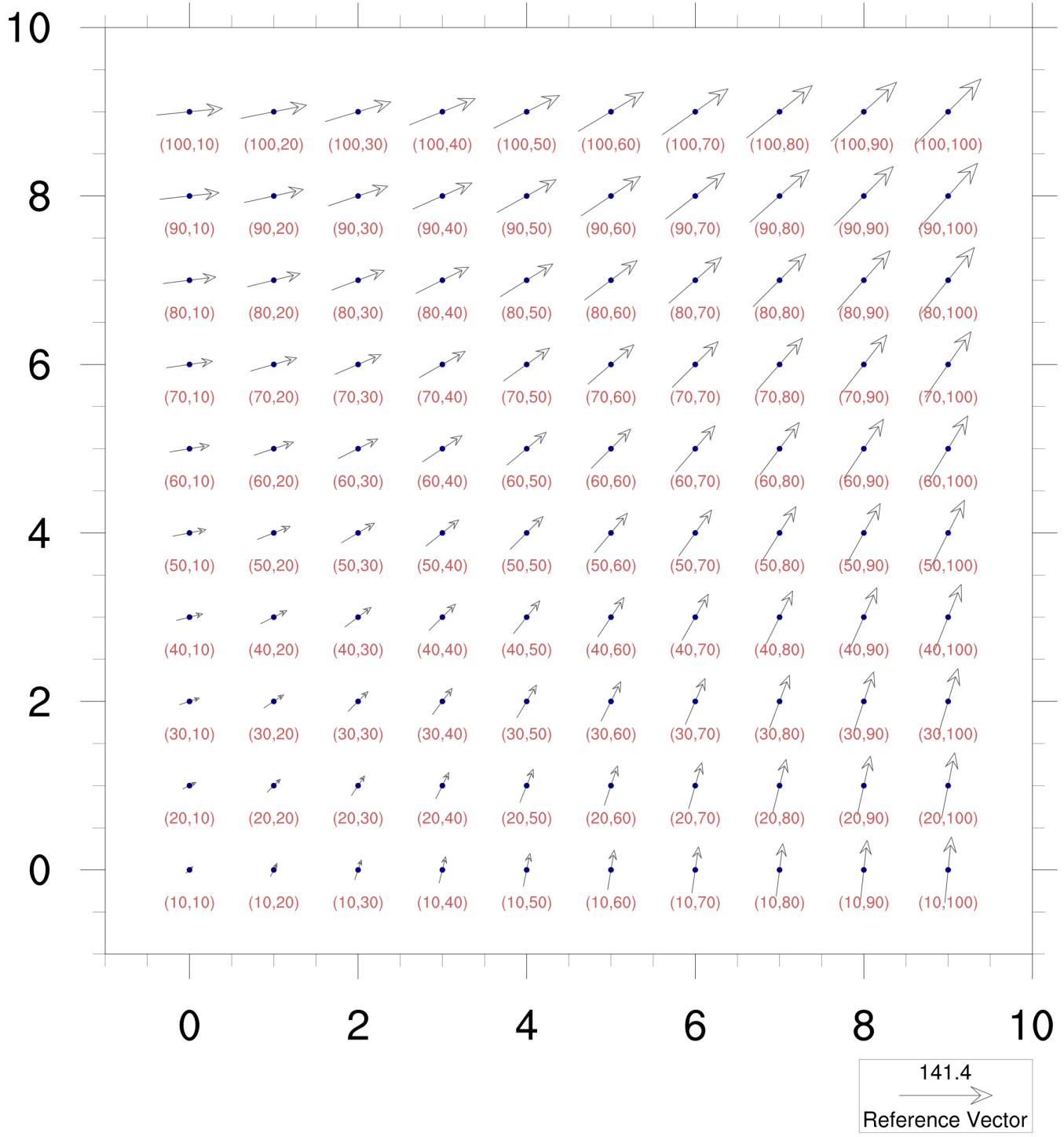
v

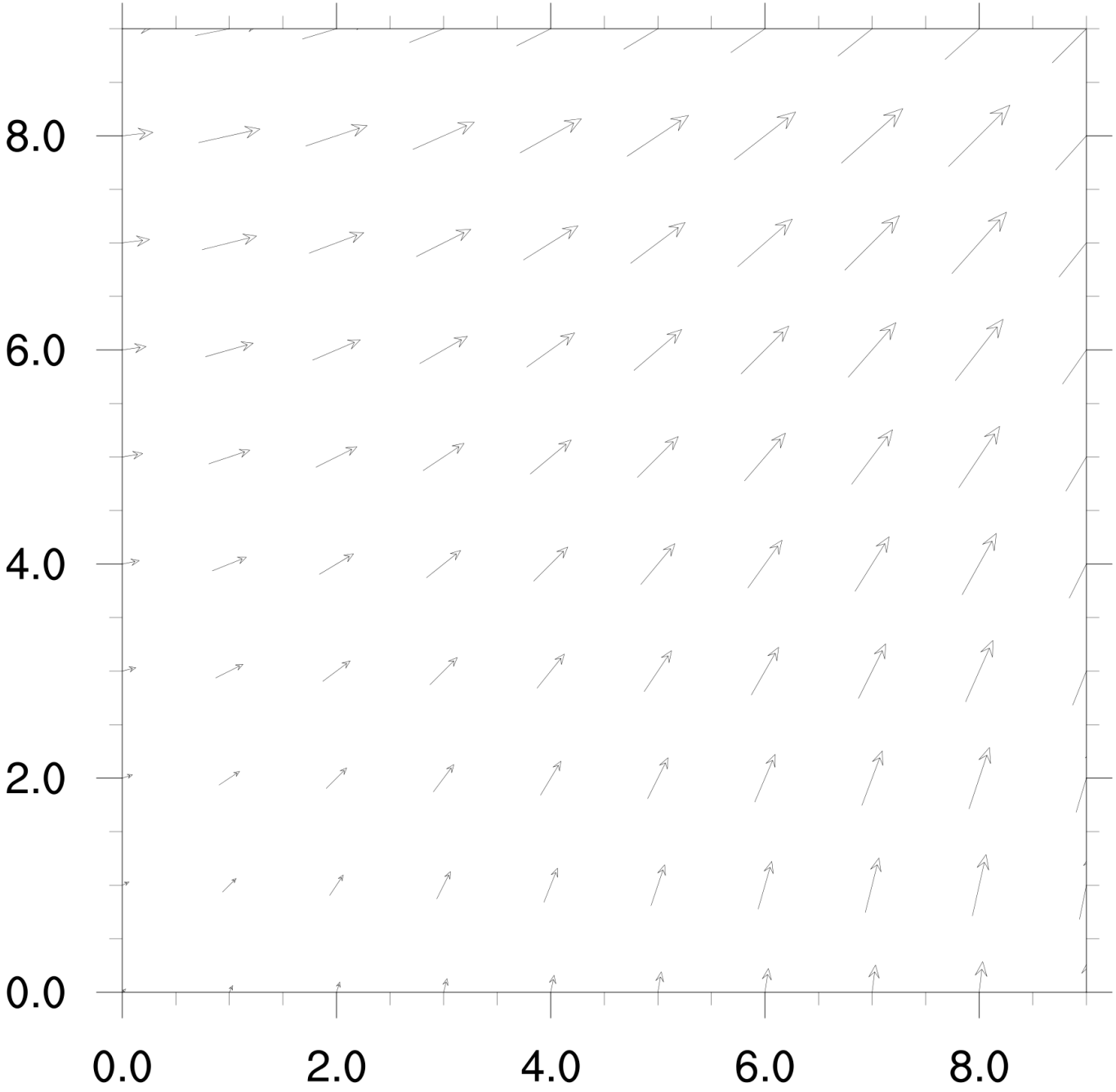
10	20	30	40	50	60	70	80	90	100
10	20	30	40	50	60	70	80	90	100
10	20	30	40	50	60	70	80	90	100
10	20	30	40	50	60	70	80	90	100
10	20	30	40	50	60	70	80	90	100
10	20	30	40	50	60	70	80	90	100
10	20	30	40	50	60	70	80	90	100
10	20	30	40	50	60	70	80	90	100
10	20	30	40	50	60	70	80	90	100
10	20	30	40	50	60	70	80	90	100


8.0
6.0
4.0
2.0
0.0
0.0
2.0
4.0
6.0
8.0
141.4
Reference Vector

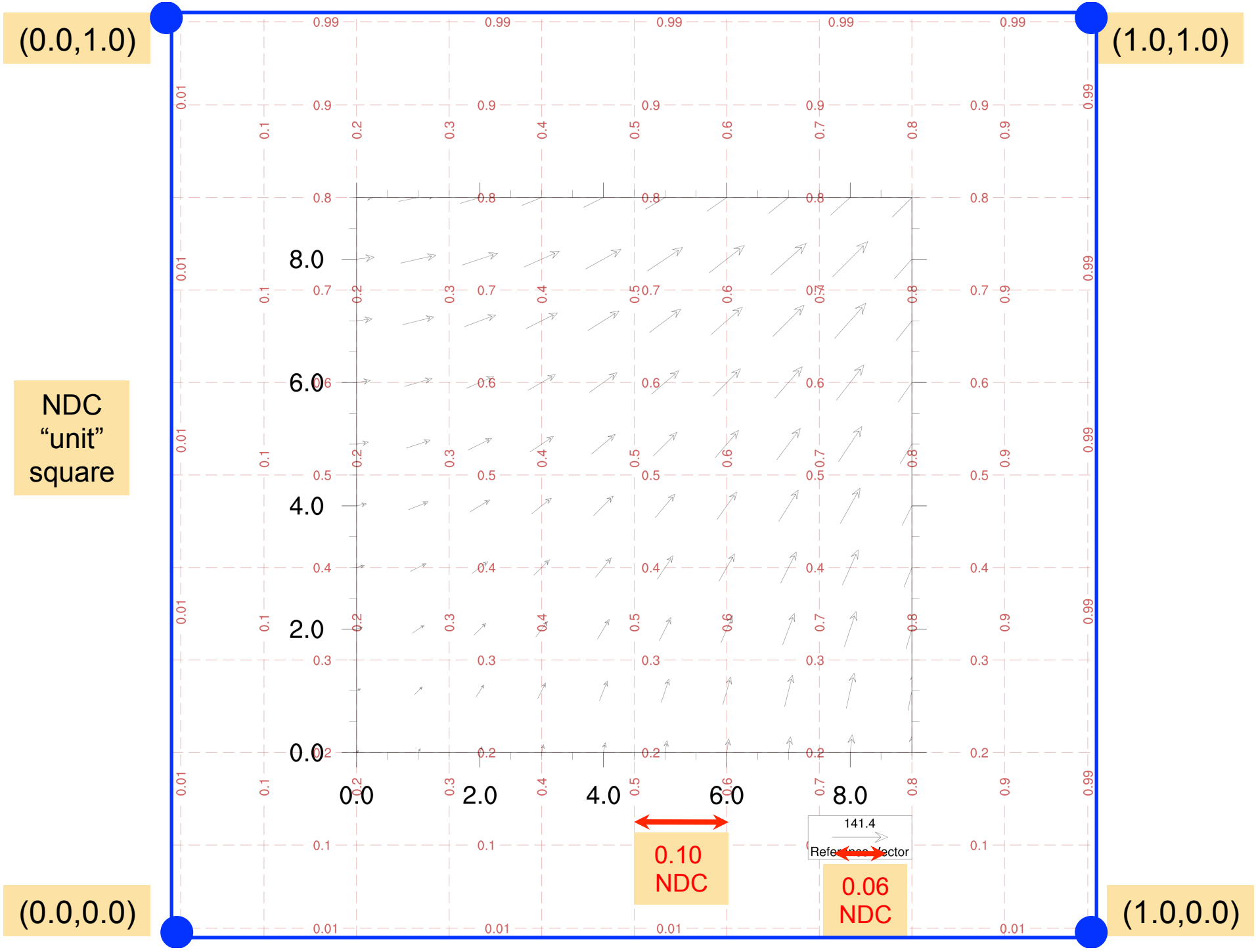
Arrows are centered on their locations (blue dot)

No coordinates provided, so 0 to *n*-1 are used (0 to 9)





Default vector reference length in this case is 0.06 in “NDC” units


(0.0,1.0)
(1.0,1.0)
NDC
“unit”
square
(0.0,0.0)
(1.0,0.0)
0.0
2.0
4.0
6.0
8.0
0.10
NDC
141.4
0.06
NDC

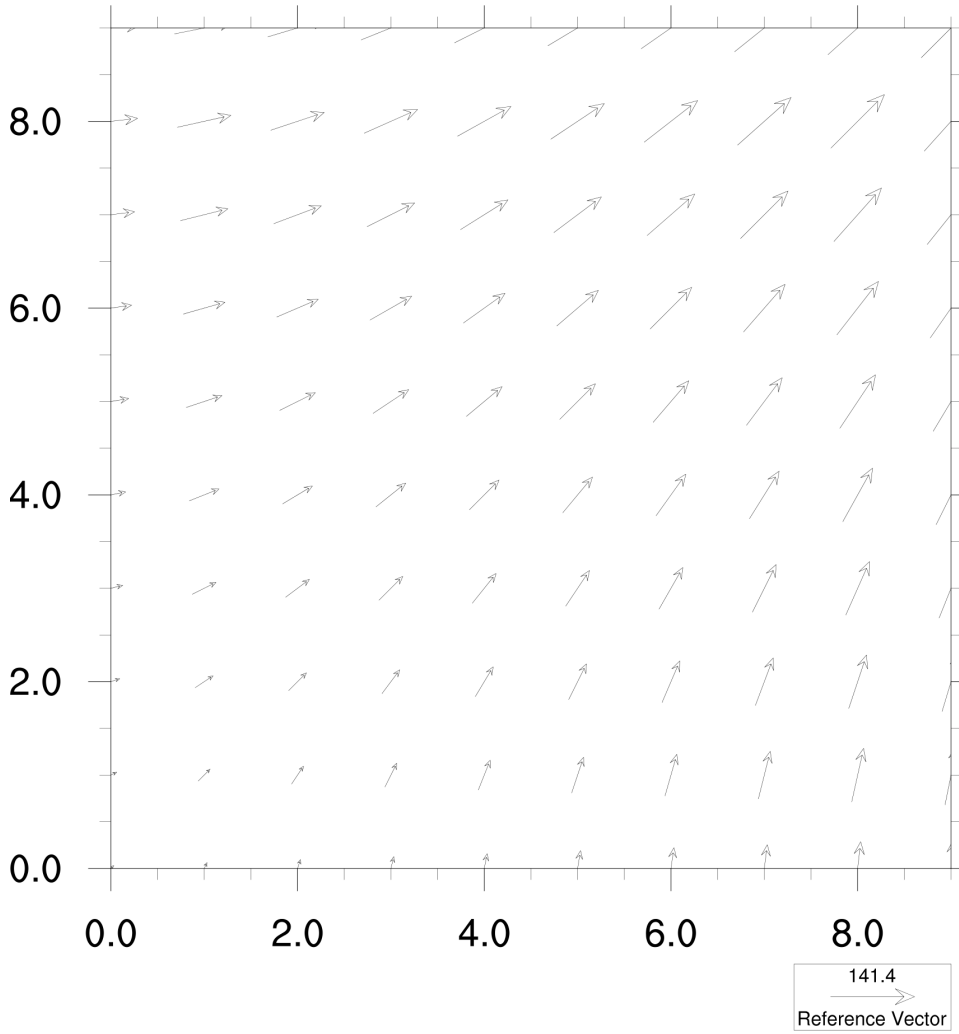
Example *vector1b.ncl*

- Change the vector reference length with vcRefLengthF

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

begin
;---Generate dummy u, v arrays
  npts = 10
  ii   = ispan(10,10*npts,10)
  u    = conform_dims((/npts,npts/),ii,0)
  v    = conform_dims((/npts,npts/),ii,1)

;---Open an X11 window
  wks = gsn_open_wks("x11","vector1b")

;---Set a vector resource
  res = True
  res@vcRefLengthF = 0.10    ; NCL chose 0.06

;---Call the plotting function
  vector = gsn_csm_vector(wks,u,v,res)
end
```

vector1a.ncl

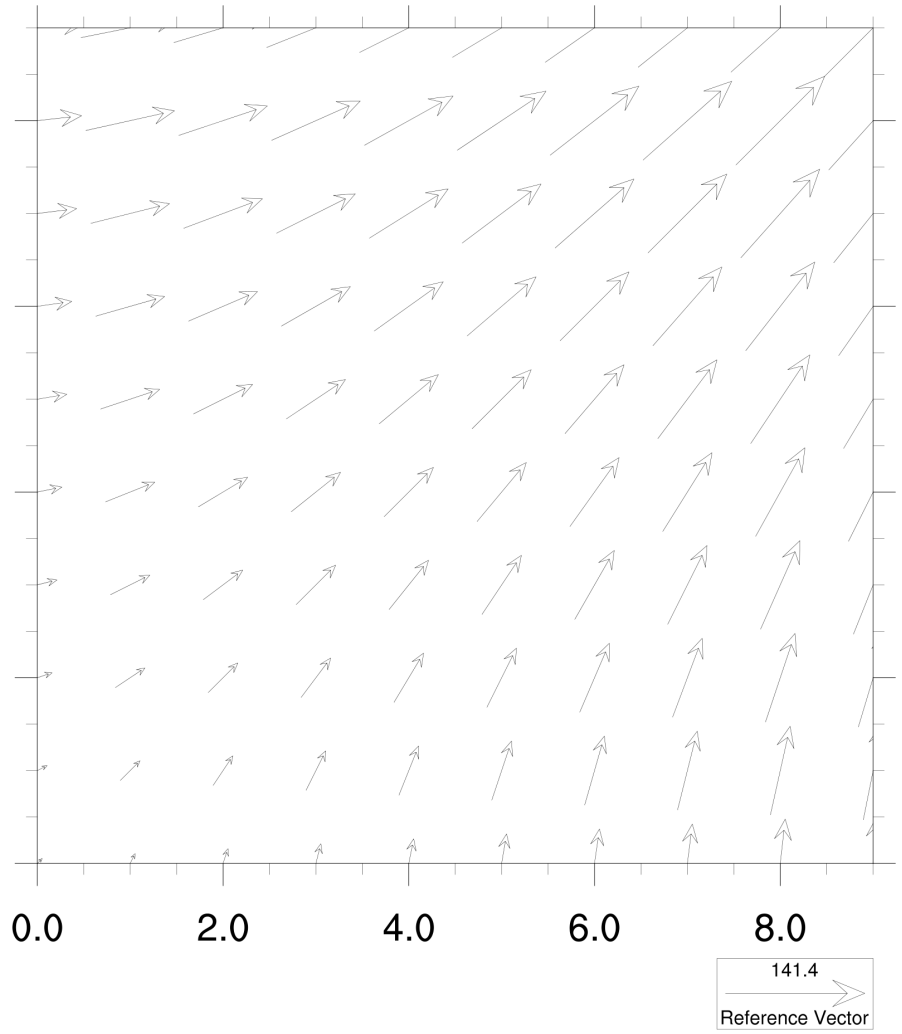
vcRefLengthF = 0.06



0.06 NDC

vector1b.ncl

vcRefLengthF = 0.10



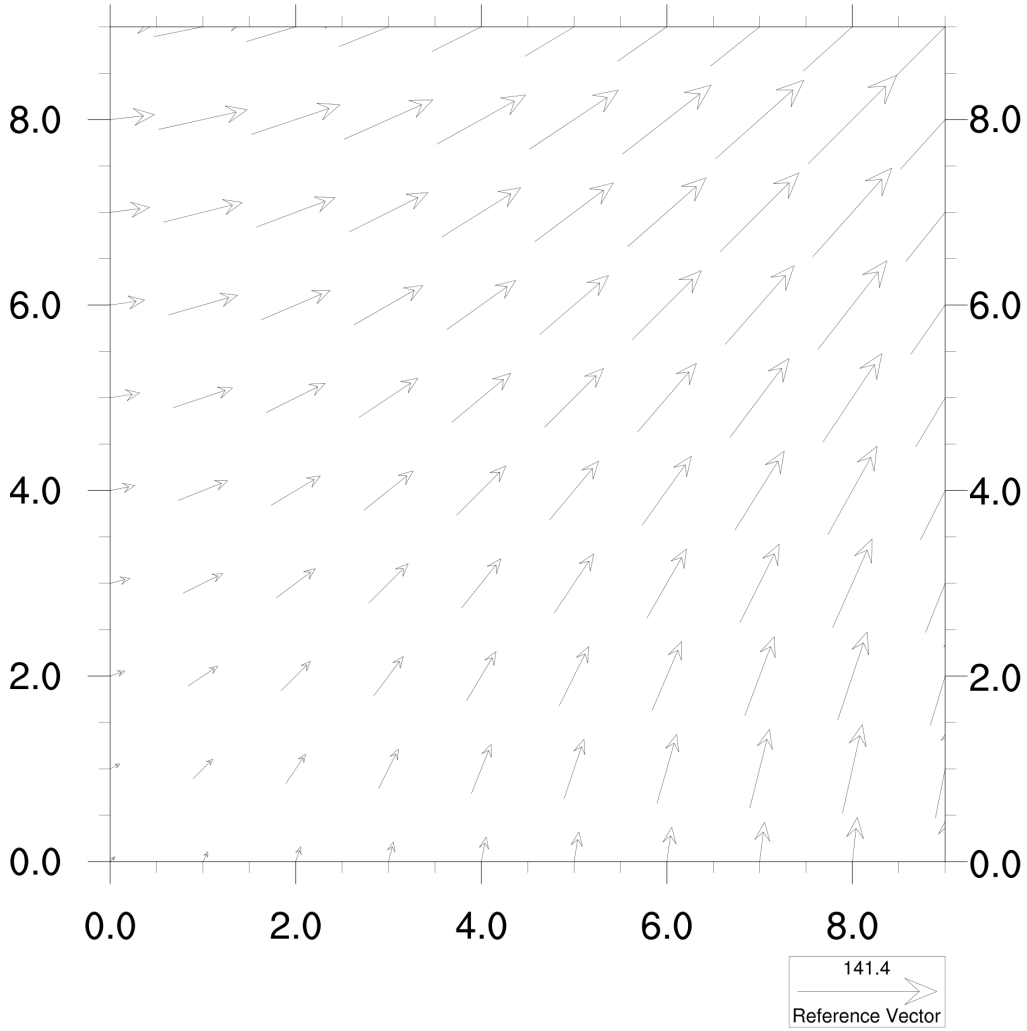
0.10 NDC

Example *vector1c.ncl*

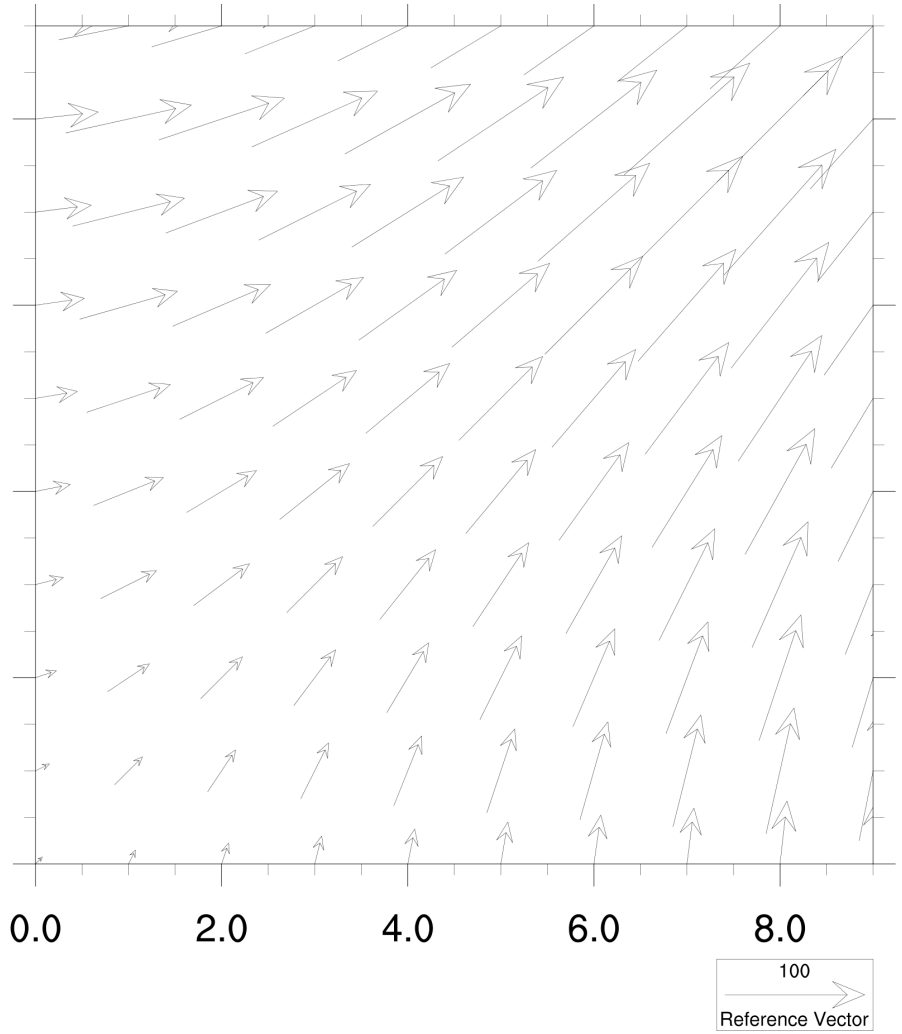
- Change which vector magnitude (vcRefMagnitudeF) is rendered in length specified by vcRefLengthF
- Default is to render the longest vector (the one with maximum magnitude) in this reference length

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

begin
  npts = 10
  ii   = ispan(10,10*npts,10)
  u    = conform_dims((/npts,npts/),ii,0)
  v    = conform_dims((/npts,npts/),ii,1)

  wks = gsn_open_wks("x11","vector1c")

  res                    = True
  res@vcRefLengthF       = 0.10    ; NCL chose 0.06
  res@vcRefMagnitudeF    = 100     ; max is 141.4

  vector = gsn_csm_vector(wks,u,v,res)
end
```

vector1b.ncl



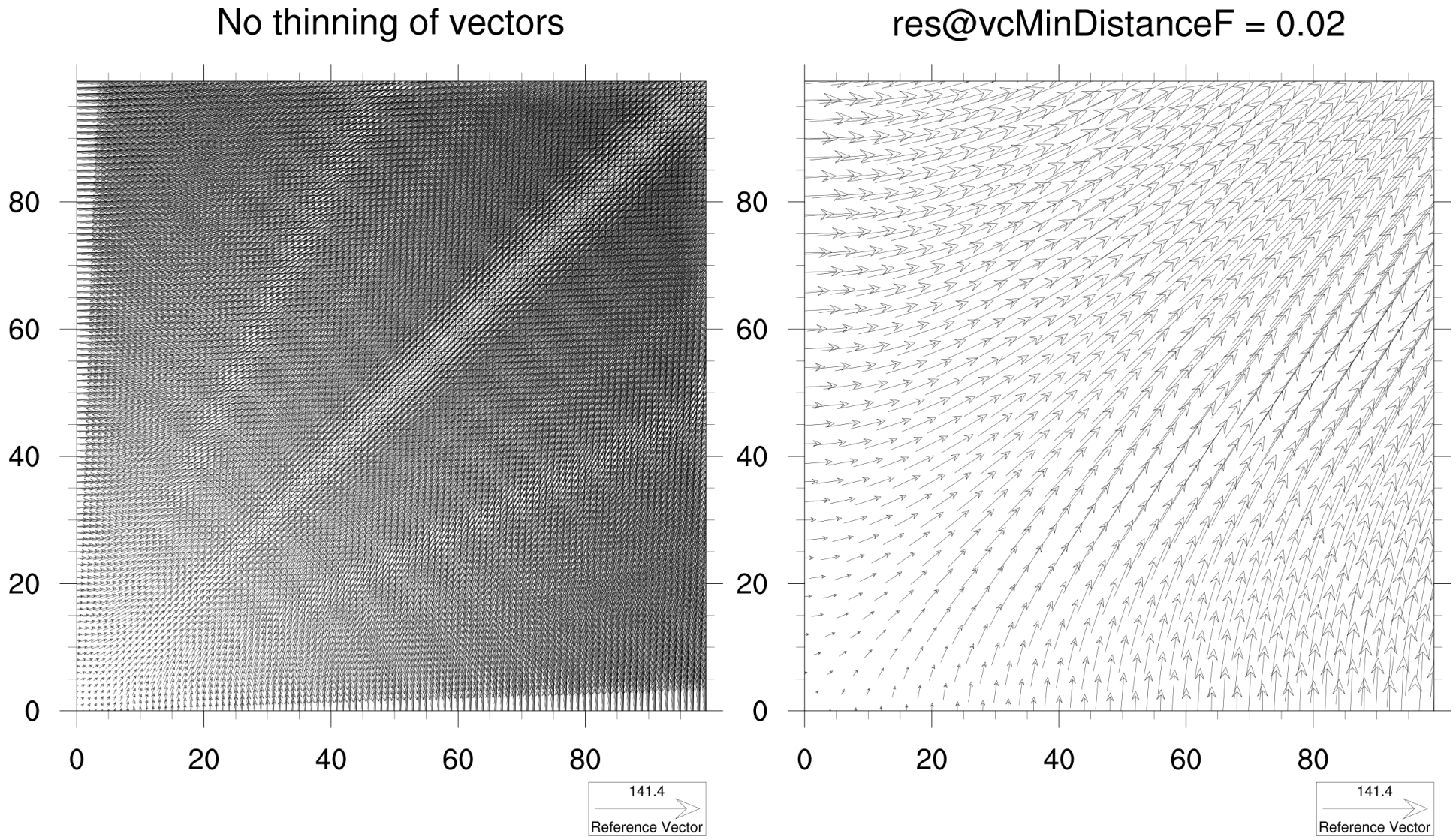
vector1c.ncl



Example *vector1d.ncl*

- Increase size of U, V to create dense plot
- Thin the vectors with vcMinDistanceF

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

begin
;---Generate larger (more dense) dummy u, v arrays
  npts = 100
  ii   = ispan(1,npts,1)
  u    = conform_dims((/npts,npts/),ii,0)
  v    = conform_dims((/npts,npts/),ii,1)

  wks = gsn_open_wks("x11","vector1d")

  res                    = True
  res@vcRefLengthF       = 0.10
  res@vcMinDistanceF     = 0.02      ; Thin the vectors

  vector = gsn_csm_vector(wks,u,v,res)
end
```


No thinning of vectors
res@vcMinDistanceF = 0.02
0
20
40
60
80
141.4
Reference Vector

Vector plot examples

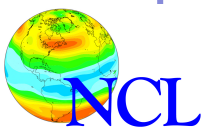
NCL Home Page -> Examples -> Vectors

NCL Home Page -> Examples -> Vector effects

http://www.ncl.ucar.edu/Applications/vector.shtml
http://www.ncl.ucar.edu/Applications/veceff.shtml

NCL Graphics topics for this lecture

- Gallery
- Description of NCL vector plots
- Line-by-line vector plot example
- Demo
- Description of NCL overlay plots
- Line-by-line overlay plot example
- Demo



Demo

- Vectors over a map – gsn_csm_vector_map
- Will use "uv300.nc" data file
- Scripts (vector2x.ncl) and data file available from:

  NCL Home Page -> Support -> Webinars -> NCL Graphics

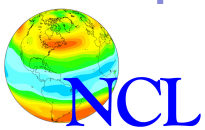
  http://www.ncl.ucar.edu/Training/Webinars/NCL_Graphics/VectorOverlayDemo/

NCL Graphics topics for this lecture

- Gallery
- Description of NCL vector plots
- Line-by-line vector plot example
- Demo
- Description of NCL overlay plots
- Line-by-line overlay plot example
- Demo

Overlays

- An “overlay plot” is a plot overlaid on top of a base plot
- Two types of overlays:
  1. “Transform” one plot into data space of another
  2. Simply “line up” two plots at their four corners
- Method #1 uses data space of plots’ axes
- Method #2 doesn’t do any transformation

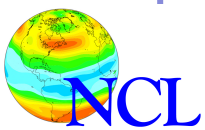


"overlay" procedure

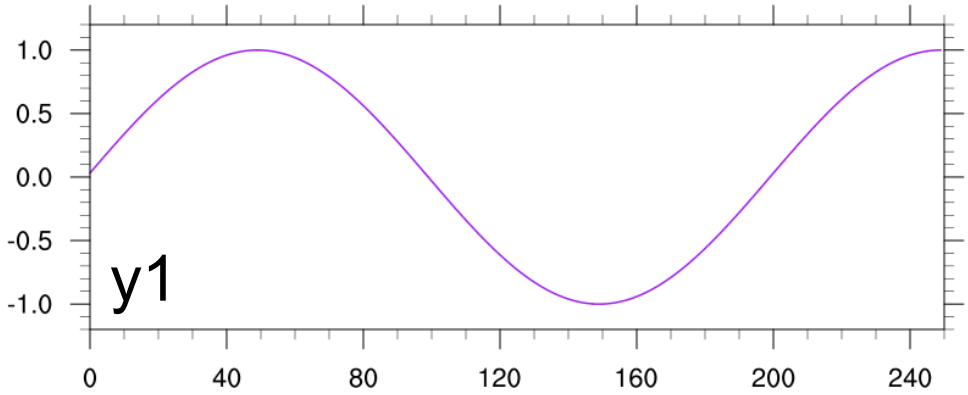
- overlay joins the "top" plot to "base" plot

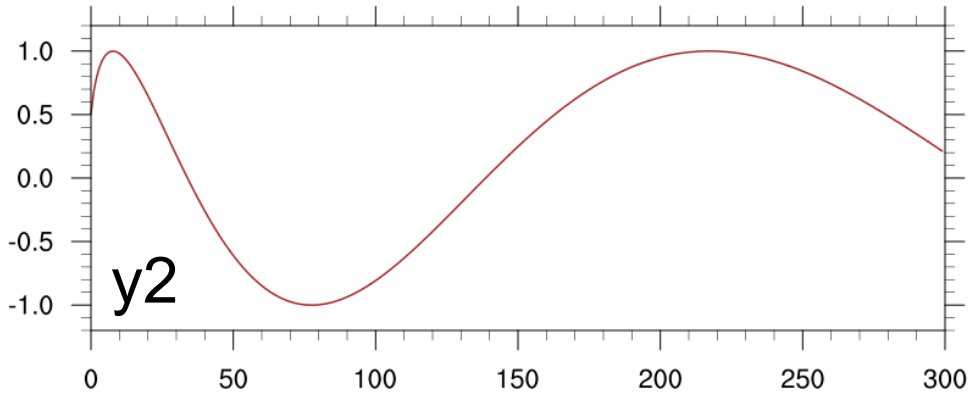
  `overlay(`*`base_plot,top_plot`*`)`

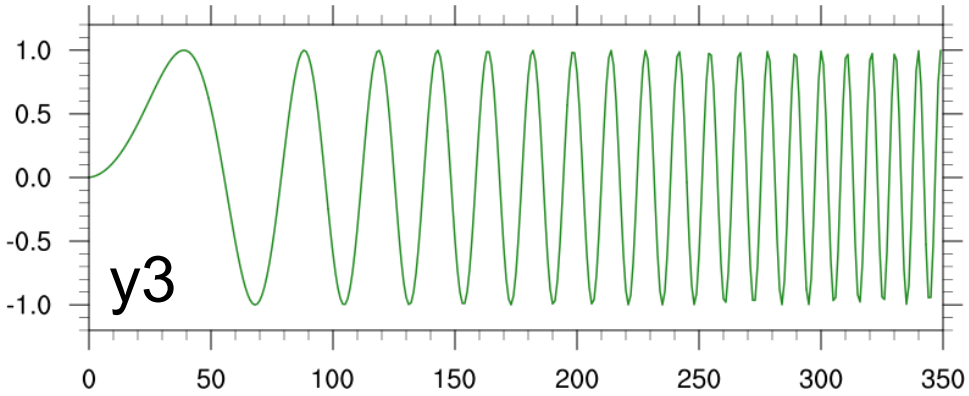
- The two plots must either have:
  1. similar data spaces
  2. the "tfDoNDCOverlay" resource set to True
- The *top_plot* main title and tickmarks will not be part of *base_plot*



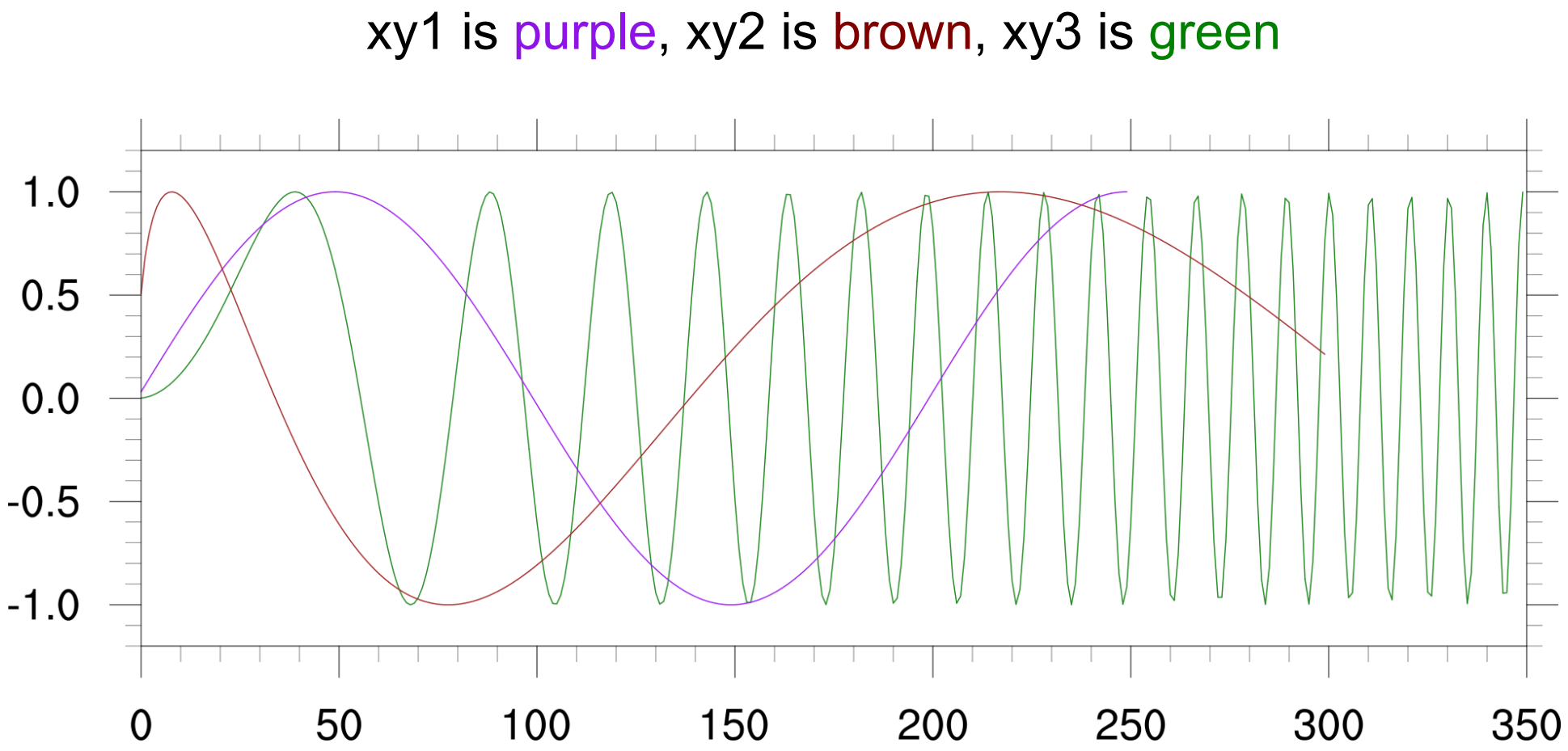
```
;---Create three curves
PI100  = 0.031415926535898
x1 = PI100*ispan(1,250,1)
x2 = PI100*ispan(1,300,1)
x3 = PI100*ispan(1,350,1)
y1 = sin(x1)
y2 = sin(3*sqrt(fabs(x2)))
y3 = sin(x3^2)
```



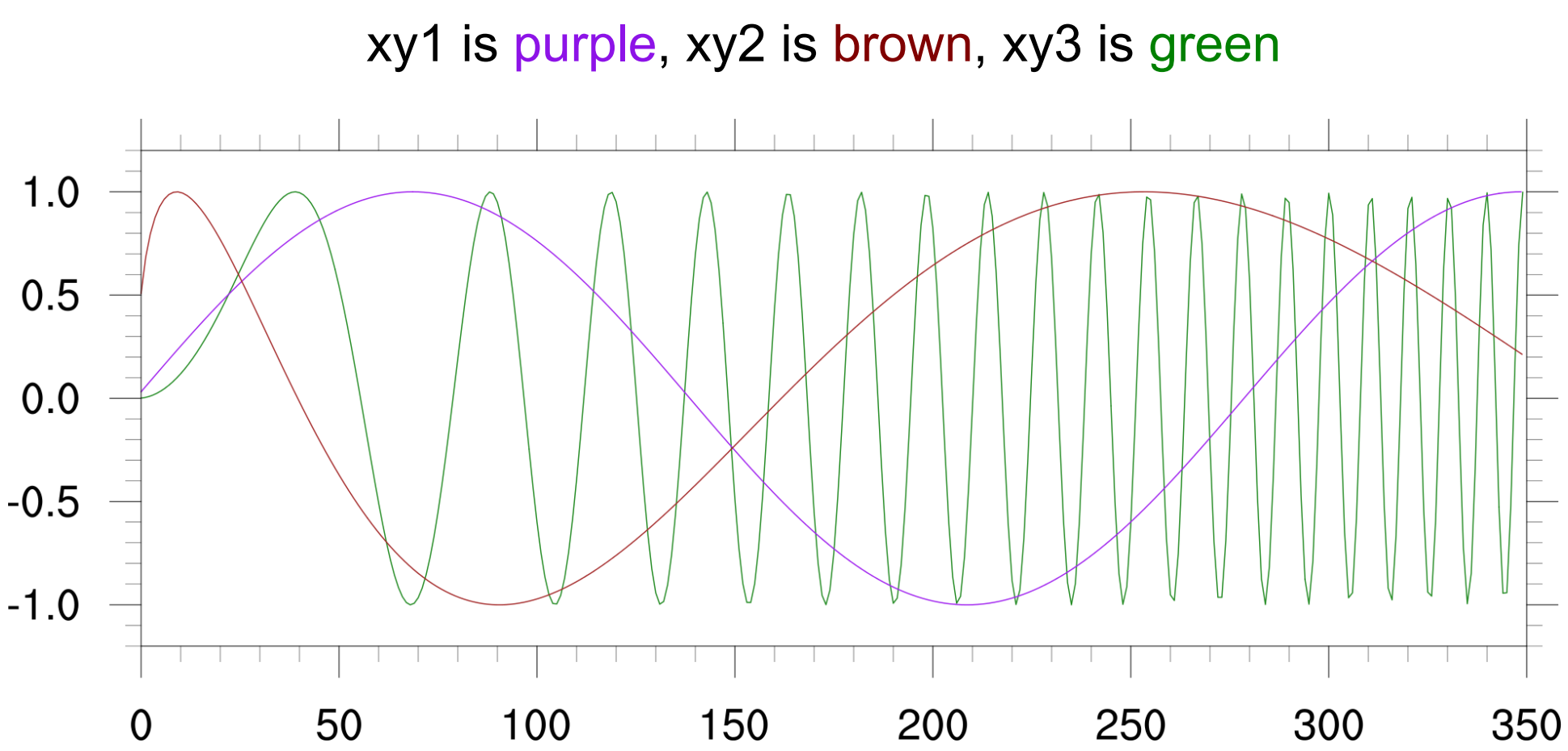




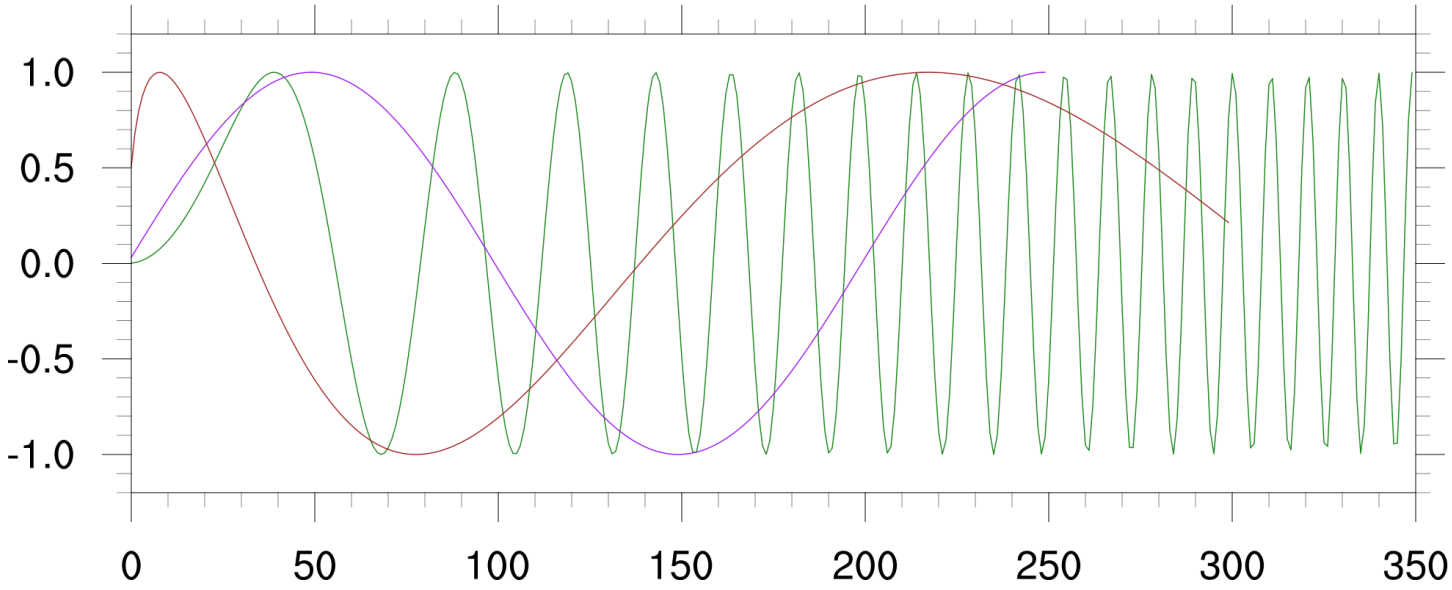
```
wks = gsn_open_wks("x11",”xy”)
res                  = True
res@vpWidthF         = 0.70
res@vpHeightF        = 0.25
res@xyLineThicknessF = 3

res@xyLineColor    = "Purple”
res@trXMaxF        = dimsizes(y1)
xy1 = gsn_csm_y(wks,y1,res)

res@xyLineColor    = "Brown”
res@trXMaxF        = dimsizes(y2)
xy2 = gsn_csm_y(wks,y2,res)

res@xyLineColor    = "ForestGreen”
res@trXMaxF        = dimsizes(y3)
xy3 = gsn_csm_y(wks,y3,res)

overlay(xy3,xy1) ; Overlay xy1 on xy3
overlay(xy3,xy2) ; Overlay xy2 on xy3
draw(xy3)        ; Draws all three plots in one data space
frame(wks)       ; Don’t forget to call this!
```


xy1 is purple, xy2 is brown, xy3 is green
1.0
0.5
0.0
-0.5
-1.0
0
50
100
150
200
250
300
350

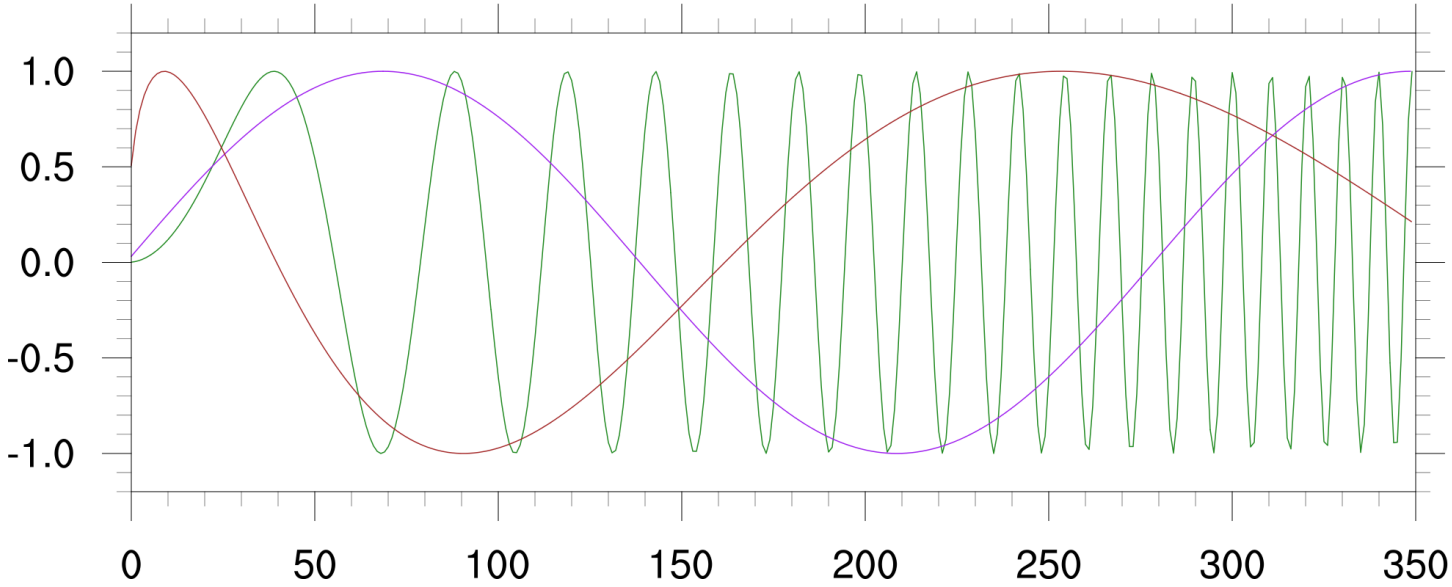
```
wks = gsn_open_wks("x11",”xy”)
res                  = True
res@vpWidthF         = 0.70
res@vpHeightF        = 0.25
res@xyLineThicknessF = 3
res@tfDoNDCOverlay   = True  ; line up four corners of each plot

res@xyLineColor      = "Purple”
res@trXMaxF          = dimsizes(y1)
xy1 = gsn_csm_y(wks,y1,res)

res@xyLineColor      = "Brown”
res@trXMaxF          = dimsizes(y2)
xy2 = gsn_csm_y(wks,y2,res)

res@xyLineColor      = "ForestGreen”
res@trXMaxF          = dimsizes(y3)
xy3 = gsn_csm_y(wks,y3,res)

overlay(xy3,xy1) ;---Overlay xy1 on xy3
overlay(xy3,xy2) ;---Overlay xy2 on xy3
draw(xy3)        ;---Draws all three plots
frame(wks)
```


xy1 is purple, xy2 is brown, xy3 is green
1.0
0.5
0.0
-0.5
-1.0
0
50
100
150
200
250
300
350

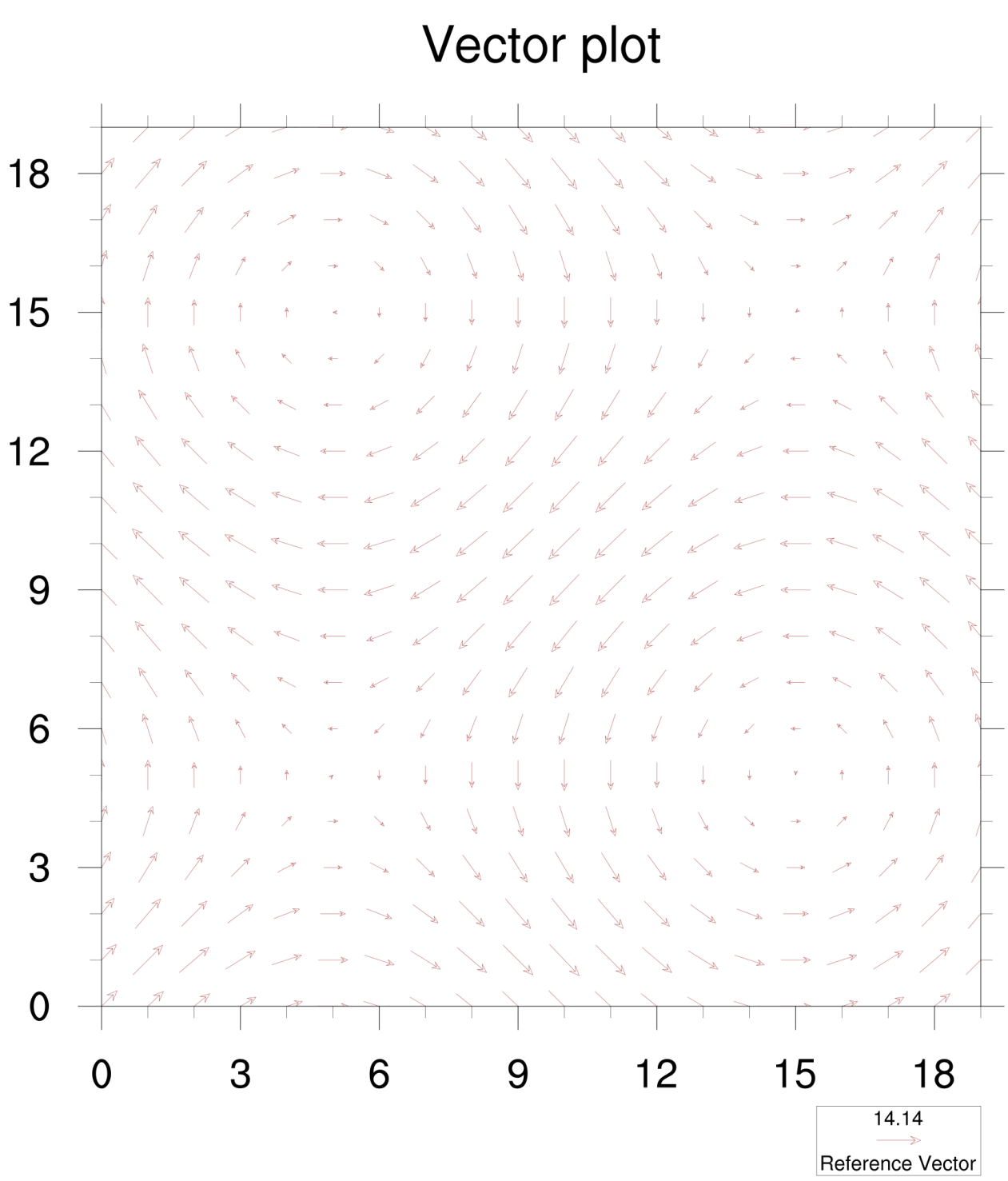
Method 1: overlay plots in data space of base plot



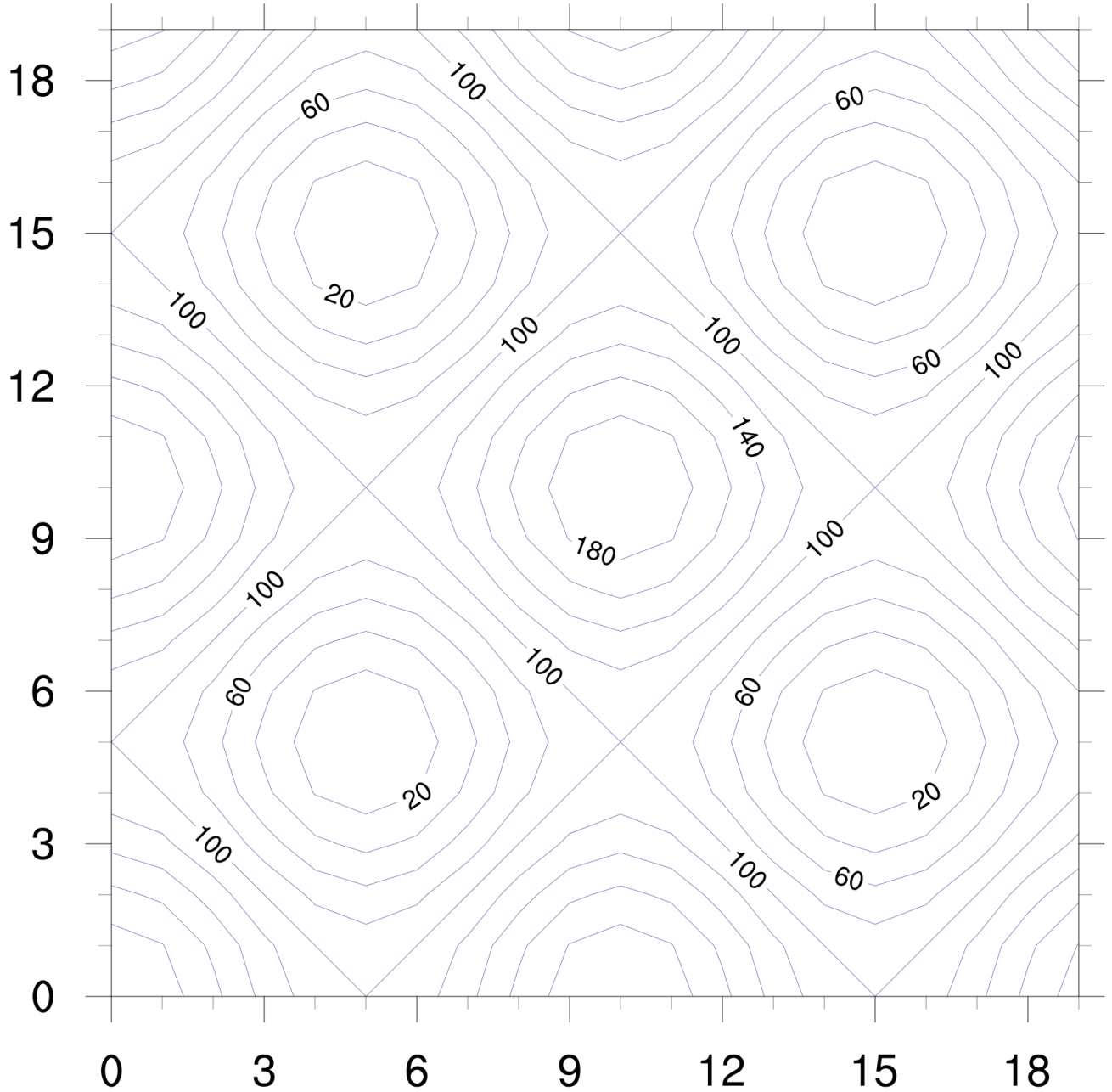
Method 2: line up plots by their four corners



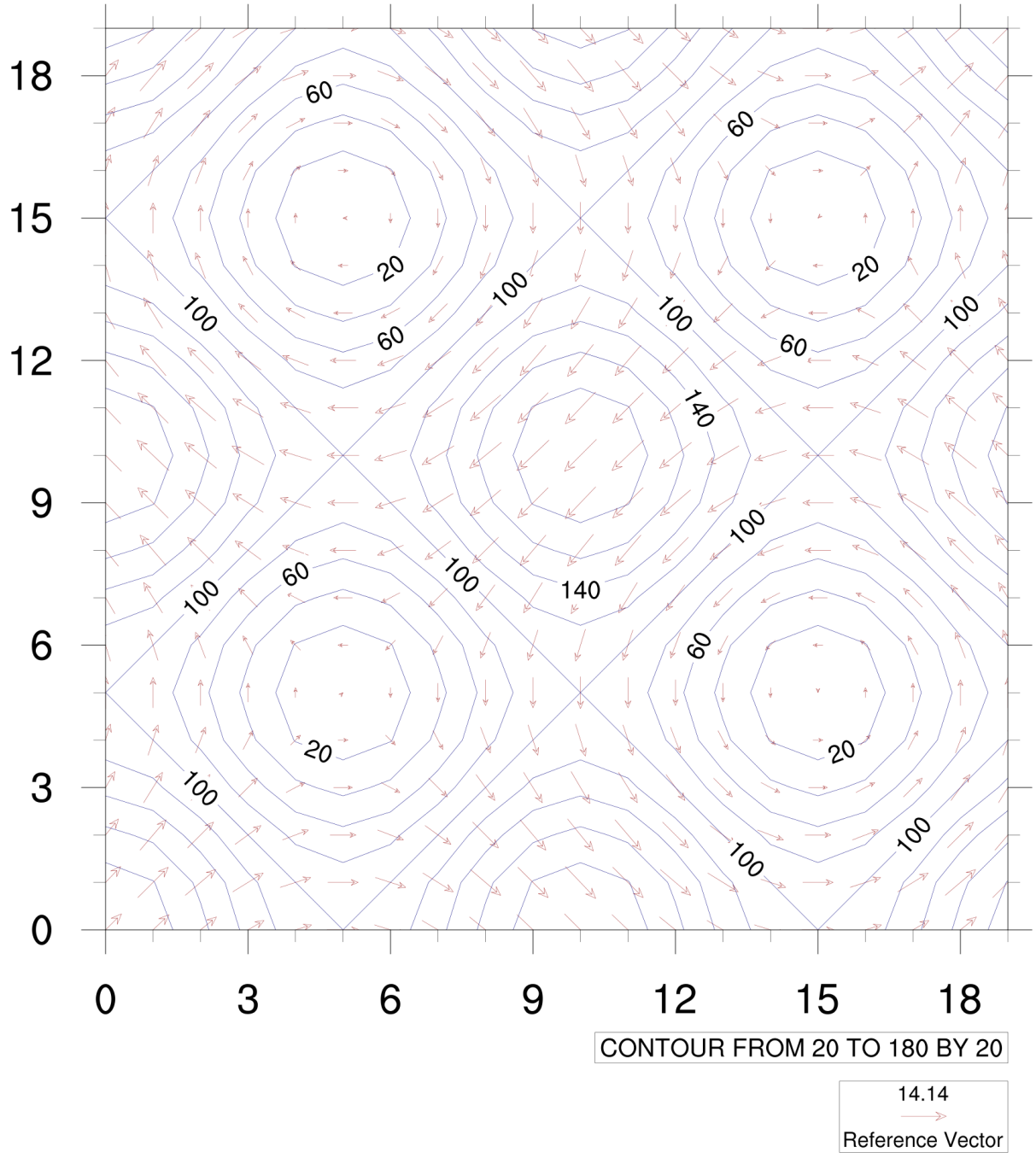
```
;---Generate dummy u, v, w arrays
  N  = 20
  PI = 3.14159
  I  = ispan(0,N-1,1)
  u  = 10. * cos(conform_dims((/N,N/),(2.0 * PI / N) * I, 0))
  v  = 10. * cos(conform_dims((/N,N/),(2.0 * PI / N) * I, 1))
  w  = u^2+v^2

  wks = gsn_open_wks("x11","overlay1a")

  vcres                        = True        ; vector resources
  vcres@vcLineArrowColor       = "Brown"
  vcres@tiMainString           = "Vector plot"
  vector  = gsn_csm_vector(wks,u,v,vcres)

  cnres              = True                  ; contour resources
  cnres@cnLineColor  = "NavyBlue"
  cnres@tiMainString = "Contour plot"
  contour = gsn_csm_contour(wks,w,cnres)

  overlay(vector,contour)      ; overlay contour plot on vector plot

  draw(vector)   ; This will draw both vector and contour plot
  frame(wks)
```


Vector plot
18
15
12
9
6
3
0
0
3
6
9
12
15
18
14.14
Reference Vector

Contour plot



Vector plot


18
15
12
9
6
3
0
0
3
6
9
12
15
18
60
20
100
140
CONTOUR FROM 20 TO 180 BY 20
14.14
Reference Vector

```
;---Generate dummy u, v, w arrays
  N  = 20
  PI = 3.14159
  I  = ispan(0,N-1,1)
  u  = 10. * cos(conform_dims((/N,N/),(2.0 * PI / N) * I, 0))
  v  = 10. * cos(conform_dims((/N,N/),(2.0 * PI / N) * I, 1))
  w  = u^2+v^2

  wks = gsn_open_wks("x11","overlay1b")

  vcres                        = True          ; vector resources
  vcres@vcLineArrowColor       = "Brown"
  vcres@tiMainString           = "Vector plot"
  vector  = gsn_csm_vector(wks,u,v,vcres)

  cnres              = True                    ; contour resources
  cnres@cnLineColor  = "NavyBlue"
  cnres@tiMainString = "Contour plot"
  contour = gsn_csm_contour(wks,w,cnres)

  overlay(contour,vector)       ; overlay vector plot on contour plot

  draw(contour)   ; This will draw both contour and vector plot
  frame(wks)
```

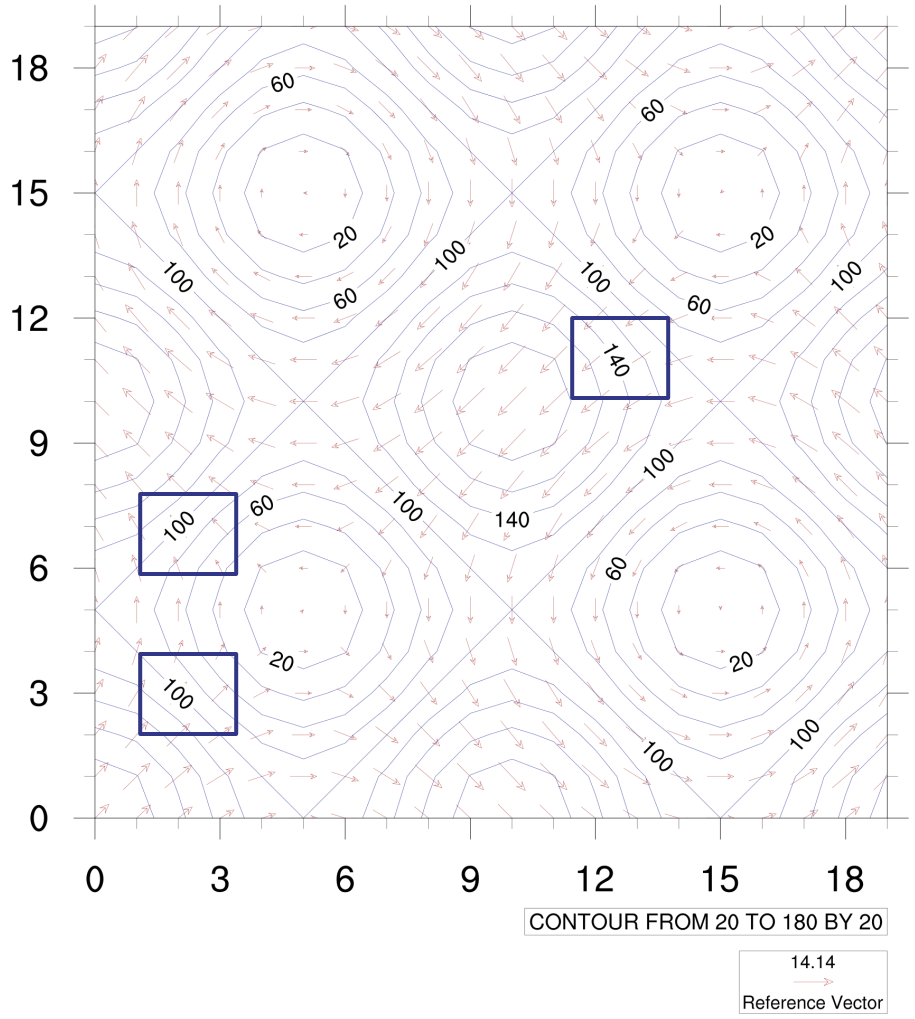
`overlay(vector,contour)`

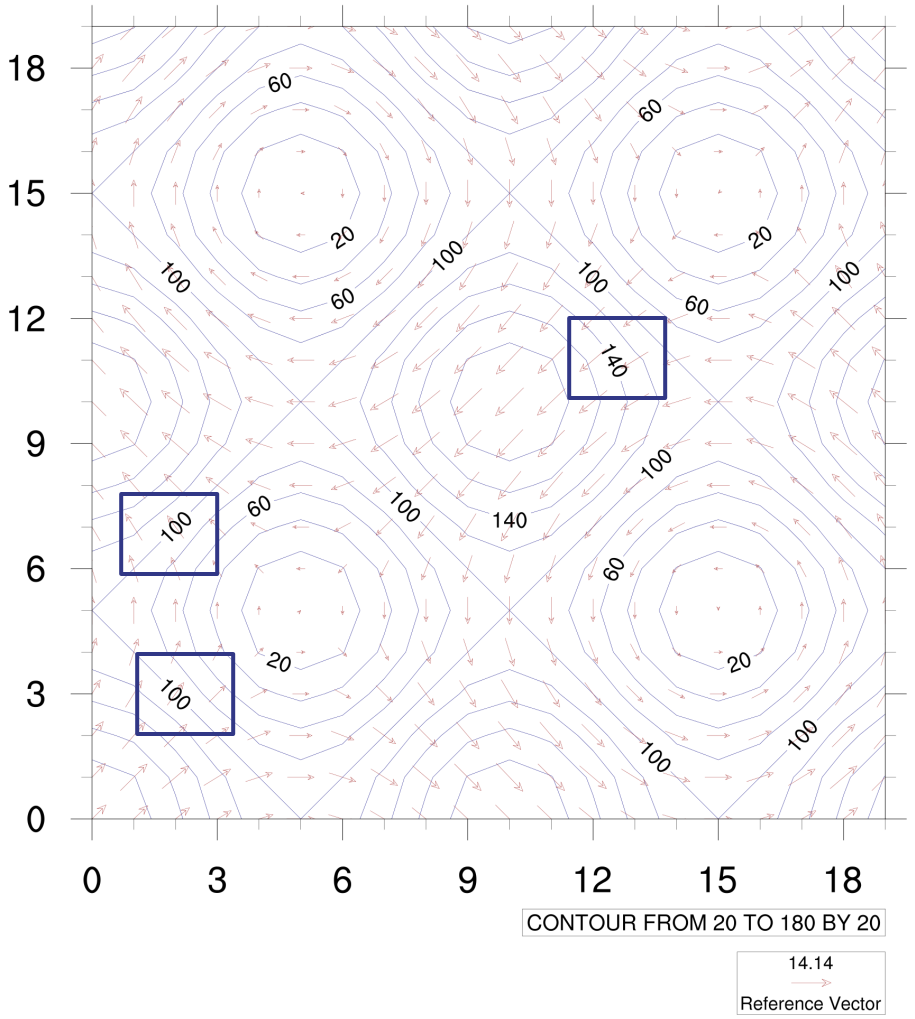
`overlay(contour,vector)`

Vector plot

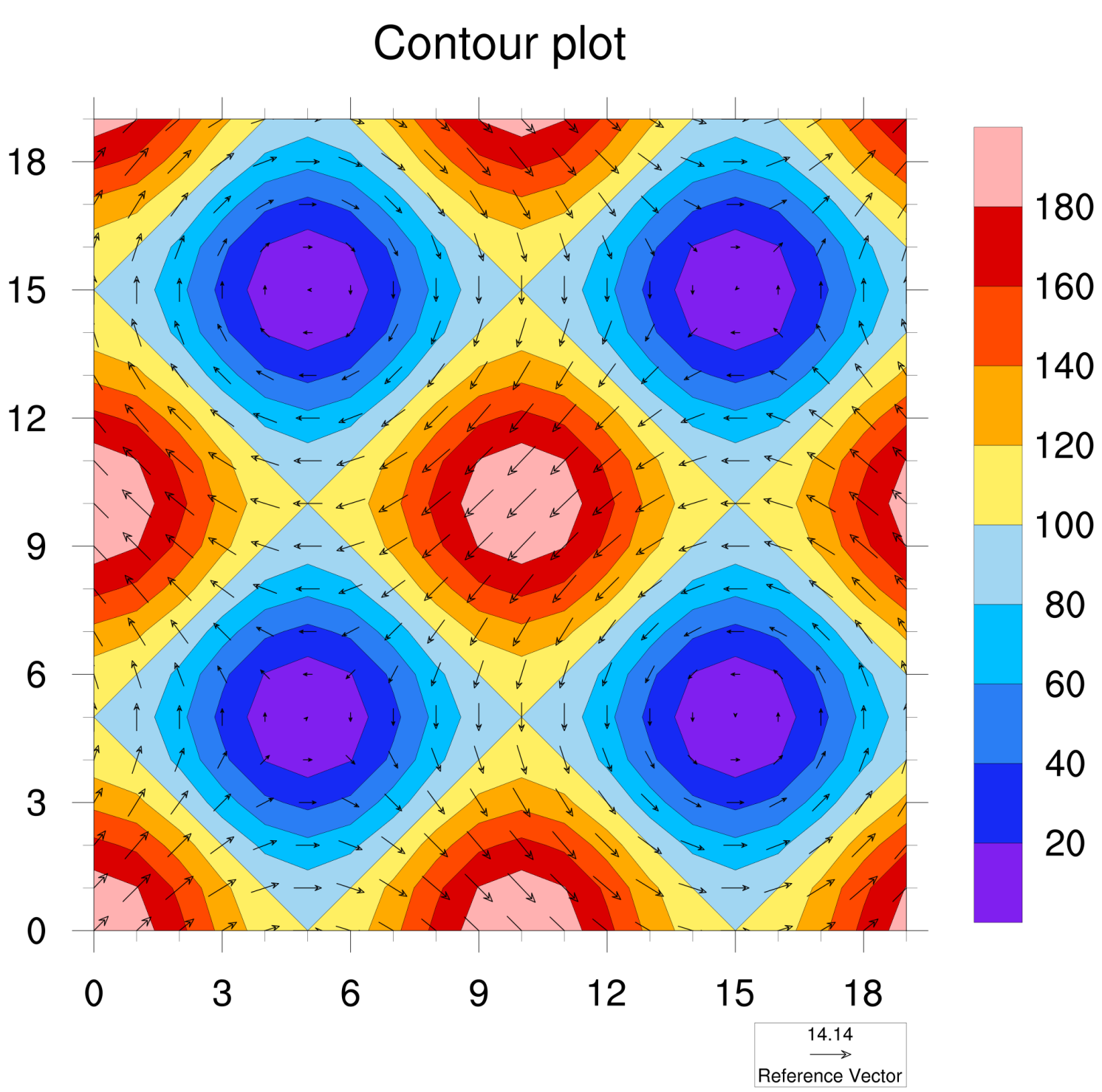
`What's the difference?`

Contour plot





```
;---Generate dummy u, v, w arrays
  N  = 20
  PI = 3.14159
  I  = ispan(0,N-1,1)
  u  = 10. * cos(conform_dims((/N,N/),(2.0 * PI / N) * I, 0))
  v  = 10. * cos(conform_dims((/N,N/),(2.0 * PI / N) * I, 1))
  w  = u^2+v^2

  wks = gsn_open_wks("x11","overlay1c")

  vcres                        = True         ; vector resources
  vcres@vcLineArrowThicknessF  = 3.0          ; 3x as thick
  vcres@tiMainString           = "Vector plot"
  vector  = gsn_csm_vector(wks,u,v,vcres)

  cnres                 = True                ; contour resources
  cnres@cnFillOn        = True                ; turn on color fill
  cnres@lbOrientation   = "Vertical"          ; make labelbar vertical
  cnres@tiMainString    = "Contour plot"
  contour = gsn_csm_contour(wks,w,cnres)

  overlay(contour,vector)       ; overlay vector plot on contour plot

  draw(contour)    ; This will draw both contour and vector plot
  frame(wks)
```


Contour plot
18
15
12
9
6
3
0
0
3
6
9
12
15
18
180
160
140
120
100
80
60
40
20
14.14
Reference Vector

Overlay examples

NCL Home Page -> Examples -> overlay

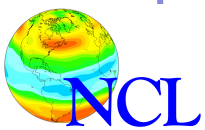
http://www.ncl.ucar.edu/Applications/overlay.shtml

Can also go to the “functions” link on Examples page and search for “overlay”.

Lots of examples use this procedure.

NCL Graphics topics for this lecture

- Gallery
- Description of NCL vector plots
- Line-by-line vector plot example
- Demo
- Description of NCL overlay plots
- Line-by-line overlay plot example
- Demo



Demo

- Overlays
- Will use “era40.pl.t85.tuvo.200001.nc”
- Scripts (overlay3x.ncl) and data file available from:

  NCL Home Page -> Support -> Webinars -> NCL Graphics
  http://www.ncl.ucar.edu/Training/Webinars/NCL_Graphics/VectorOverlayDemo

Questions?

ncl-talk@ucar.edu

http://mailman.ucar.edu/mailman/listinfo/ncl-talk