

---

Subject: Re: Wind Vectors on Maps

Posted by [Sebastien DELECRAZ](#) on Sun, 14 Mar 1999 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Tony Bauna wrote:

>  
> David Fanning wrote in message ...  
>> Hi Folks,  
>>  
>> Does anyone have a good program to add wind vectors to maps?  
>> I get asked about this a couple of times a week, it seems.  
>> I hacked up VELOVECT (or was it VEL?) some time ago to sort of  
>> do the job, but I wondered if someone had something more elegant  
>> they would be willing to share.  
> I have the same request. I have a simple ascii-file with lat/lon and wind  
> direction (deg) and -speed (m/s). I've looked at mapvec which was posted  
> here sometime ago, but it doesn't quite work for me. So if anyone has  
> something which I can use to plot wind vectors on a map or on top of a  
> satellite SAR image, I would be thankful.

As a matter of fact I was looking for something similar and I've been working on it last week.

I use the procedure `adcp_vector.pro` to plot water flow vectors on maps (in my case data are collected with a doppler sensor under a ship). `ADCP_VECTOR` accepts 4 input vectors of same size (U, V, longitude, latitude). It sizes the plotting area around the specified coordinates and draw the vector field. A scaling factor is applied so that angles and length of vectors do not change with the size of both window & map. I added a `/NO_TRACK` keyword so that the procedure won't try to plot a ship track as you may not have one with your wind data. I didn't spend much time on the Postscript keyword and it works fine on 5.2 for Mac but not my 5.1 for Win32. I put a preview of what `adcp_vector` does at [http://gala.univ-perp.fr/~delecraz/Images/adcp\\_vector.gif](http://gala.univ-perp.fr/~delecraz/Images/adcp_vector.gif)

Hope you find it usefull.

> People will thank you by showering you with cold, hard cash.  
Credit cards also accepted. ;-)

Regards,  
SEb.

---

Sebastien DELECRAZ <[delecraz@univ-perp.fr](mailto:delecraz@univ-perp.fr)>  
59, cours Lassus, F-66000 Perpignan

```
;+
; NAME:
;   ADCP_VECTOR
;
; PURPOSE:
;   This procedure plots a two-dimensional velocity field where the
;   position of each vector is given by its geographic coordinates.
;   A line/arrow is drawn at each point showing the direction
;   and magnitude of the field.
;
;   In ADCP, it will show the water velocity field measured along
;   the ship track.
;
; CATEGORY:
;   Plotting, two-dimensional (ADCP utility).
;
; CALLING SEQUENCE:
;   ADCP_VECTOR, U, V, Lon, Lat
;
; INPUTS:
;   U: The X component of each velocity vector. Typically
;       E-O velocity components.
;       U must be a vector.
;
;   V: The Y component of each velocity vector. Typically
;       N-S velocity components.
;       V must be a vector and same length as U.
;
;   Lon: Longitude values. Lon must be a vector and same
;         length as U
;
;   Lat: Latitude values. Lat must be a vector and same
;         length as U
;
; OPTIONAL INPUT PARAMETERS: (none)
;
; KEYWORD INPUT PARAMETERS:
;   ARROW: Set this keyword to make IDL draw vector as arrows.
;           The default is to draw simple lines for vectors.
;
;   BATHY: A 2 line vector of coordinates along bathymetry lines.
;           [Lon, Lat]
```

```

; COAST: A 2 line vector of coordinates along coast line.
; [Lon, Lat]
;
; COLOR_IT: Set this keyword to render a full colored velocity
; field. The 'Rainbow + white' color table is loaded
; so that the color ranges from blue (small vect.) to
; red (long vect.)
; A color bar is displayed on the bottom right of the
; graphic.
;
; COLOR_PS: Set this keyword to select the color Postscript device.
;
; FILENAME: A filename (string) if postscript is selected.
;
; HELP: Set this keyword to make IDL print this header.
;
; LENGTH: Length factor. The default is 1.
;
; MISSING: Missing data value. Values greater than MISSING in the
; input arrays are ignored (U & V).
;
; NO_TRACK: Set this keyword to remove the track line (between
; origin of each vectors) (disable ship trace).
;
; PS: Set this keyword to select the B&W Postscript device.
;
; THICK: Thickness of vector lines/arrows. The default is 1.
;
; TITLE: Set this keyword to the Main title.
;
; _EXTRA: Set this keyword to apply extra keywords to the plot
; procedure (TITLE,...).
;
; OUTPUTS: (none)
;
; COMMON BLOCKS: (none)
;
; SIDE EFFECTS:
; Plotting on the selected device is performed.
;
; DEPENDANCES: (none)
;
; RESTRICTIONS:
; Only tested on IDL v5.1 for Win32x86.
;
; PROCEDURE:
;
; COMMENT:

```

```

; The trick is to keep real angles and lengths of vectors, for
; any window size and geographic area. A scale factor is calculated
; including the longitude/latitude ratio and the X/Y size of the
; window.
; The 'Rainbow + white' color table is loaded so that the color
; ranges from blue (small vect.) to red (long vect.)
;
;
; EXAMPLE:
; To render a velocity field (U & V) along a ship track
; (Lon & Lat) in full colored arrow style, enter:
;   IDL> ADCP_VECTOR, U, V, Lon, Lat, /ARROW, /COLOR_IT
;
; To render a velocity field (U & V) in the space Lon-Lat
; without ship track, omitting values greater than 999, using
; simple colored line style, enter:
;   IDL> ADCP_VECTOR, U, V, Lon, Lat, MISSING=999, /NO_TRACK
;
; MODIFICATION HISTORY:
; Written by: Sebastien DELECRAZ, March 99.
;   delecraz@univ-perp.fr
;   http://gala.univ-perp.fr/~delecraz
;-

```

```

PRO ADCP_VECTOR, U, V,      $
    Lon, Lat,  $
    ARROW=Arrow, $
    BATHY=Bathy, $
    COAST=Coast, $
    COLOR_IT=ColorIt, $
    COLOR_PS=ColorPs, $
    HELP=Help,  $
    FILENAME=Filename, $
    LENGTH=Length,$
    MISSING=Missing, $
    NO_TRACK=NoTrack,  $
    PS=Ps,      $
    THICK=VThick,    $
    TITLE=Title,    $
    _EXTRA=Extra

```

```

; SETUP ROUTINE

```

```

;Return to caller if an error occurs:
ON_ERROR,2

```

```

; Check keyword parameters:
IF KEYWORD_SET(HELP) THEN BEGIN
    DOC_LIBRARY,'adcp_vector'

```

```

RETURN      ; exit routine.
ENDIF      ; KEYWORD_SET(HELP)

; Check input parameters:
IF (N_PARAMS() LT 4) THEN RETURN
IF ( TOTAL(SIZE(U) EQ SIZE(V)) ) NE 4. THEN $
  MESSAGE, 'U & V parameters must be 1D and same size.'
IF ( TOTAL(SIZE(Lat) EQ SIZE(Lon)) ) NE 4. THEN $
  MESSAGE, 'Lat & Lon parameters must be 1D and same size.'
IF ( TOTAL(SIZE(U) EQ SIZE(Lon)) ) NE 4. THEN $
  MESSAGE, 'U, V, Lat & Lon parameters must be 1D and same size.'

; Set length factor for vectors:
IF KEYWORD_SET(Length) THEN Length=FLOAT(Length) $
  ELSE Length=1.

; Set value for missing data:
IF N_ELEMENTS(Missing) LE 0 THEN Missing = 1.0e30

; Select the PostScript device as specified:
IF KEYWORD_SET(Ps) THEN BEGIN
  OldDevice=!D.Name ; Save current graphic device:
  SET_PLOT, 'PS' ; set PS
  IF NOT KEYWORD_SET(Filename) THEN Filename = 'adcp_vector.ps'
  DEVICE, /ENCAPSULATED, FILENAME = Filename
ENDIF
IF KEYWORD_SET(ColorPs) THEN BEGIN
  OldDevice=!D.Name ; Save current graphic device:
  SET_PLOT, 'PS' ; set PS
  IF NOT KEYWORD_SET(Filename) THEN Filename = 'adcp_vector.ps'
  DEVICE, /ENCAPSULATED, /COLOR, FILENAME = Filename
ENDIF

; Set thickness of vector lines/arrows:
IF N_ELEMENTS(VThick) LE 0 THEN VThick = 1.

IF NOT KEYWORD_SET(Title) THEN Title=""

; SETUP GRAPHICS

; Save color table:
TVLCT, SaveR, SaveG, SaveB, /GET
OldColorTable = [[SaveR],[SaveG],[SaveB]]
; Load color table: Rainbow + white
LOADCT,39
; Name some color indexes for color table 39:
WHITE = 255B

```

```
BLACK   = 0B
DARK_BLUE = 30B
BLUE    = 60B
LIGHT_BLUE = 110B
GREEN   = 160B
YELLOW  = 190B
RED     = 254B
```

```
; Save index for default color and background color:
```

```
OldBg  = !P.Background
```

```
OldColor = !P.Color
```

```
; Setup a white bg for Postscripts and a black bg for windows:
```

```
IF !D.Name EQ 'ps' THEN BEGIN
```

```
  !P.Background=WHITE
```

```
  !P.Color = BLACK
```

```
ENDIF ELSE BEGIN
```

```
!P.Background=BLACK
```

```
!P.Color = WHITE
```

```
ENDELSE
```

```
; MAP
```

```
; Set map characteristics:
```

```
XMargin = 3B & YMargin = 4B
```

```
Rot = 0.0 ; Angle from North direction (MAP_SET)
```

```
LatMin = MIN(Lat) - 0.2*(MAX(Lat)-MIN(Lat))
```

```
LatMax = MAX(Lat) + 0.2*(MAX(Lat)-MIN(Lat))
```

```
LonMin = MIN(Lon) - 0.2*(MAX(Lon)-MIN(Lon))
```

```
LonMax = MAX(Lon) + 0.2*(MAX(Lon)-MIN(Lon))
```

```
LonMean = MEAN(Lon) & LatMean = MEAN(Lat)
```

```
LonOrigin = FIX(LonMin-0.5) & LatOrigin = FIX(LatMin-0.5)
```

```
LonDel = 1.0 & LatDel = 1.0 ; Spacing between meridians and  
; parallels (MAP_GRID)
```

```
; Plot map frame:
```

```
; Establish the specifications of the map
```

```
MAP_SET, LatMean, $
```

```
  LonMean, $
```

```
  Rot, $
```

```
  LIMIT=[LatMin, LonMin, LatMax, LonMax], $
```

```
  XMARGIN=XMargin, $
```

```
  YMARGIN=YMargin, $
```

```
  /MERCATOR, $
```

```
  /ISOTROPIC
```

```
; Draws the graticule of parallels and meridians:
```

```
MAP_GRID, LABEL=1, $
```

```
  LONS=LonOrigin, $
```

```
LATS=LatOrigin, $
LONDEL=LonDel, $
LATDEL=LatDel, $
CHARSIZE=1.5, $
/BOX_AXES
```

```
; DATA
```

```
; Find subscripts of bad elements:
NoGood = WHERE(U GE Missing, Count)
; NoGood = WHERE(U EQ MAX(U), Count)
; Correct velocity components (set to null):
IF (Count GT 0) THEN BEGIN
  U(NoGood)=0.01
  V(NoGood)=0.01
ENDIF
; Arbitrary set a max velocity for the upper bound of color scale:
; Arbitrary velocity range is 0 to 50.
MaxVelocity = 50B
```

```
; SCALING
```

```
; Calculate the scale factor for the current plot in
; order to always get true angles & length for vectors:
DLat=LatMax-LatMin
DLon=LonMax-LonMin
; Ratio for geographic correction:
GeographicScale = FLOAT(DLon/DLat)
; Ratio for window size correction:
WindowScale = FLOAT(!D.Y_VSize) / FLOAT(!D.X_VSize)
; Global ratio:
Scale = GeographicScale * WindowScale
; Calculate magnitude to scale colors & lines
Magnitude = SQRT(U^2.+V^2.)
; Ratio for vector length:
VFactor = (DLat < DLon)/(MAX(Magnitude)/Length) * 0.2
```

```
; PLOT
```

```
; Plot ship curse
IF NOT KEYWORD_SET(NoTrack) THEN $
  OPLOT, Lon, Lat, $
  COLOR=RED, $
  _EXTRA=Extra
```

```

; Plot velocity field:
FOR i=0,N_ELEMENTS(Lon)-1 DO BEGIN
  ; Set the color for the vector:
  ; if Magnitude(i) is greater then MaxVelocity then color index
  ; will be greater than 255 (WHITE) and we want it red.
  ; In this case, force the color index to 255-1 (RED)
  IF NOT KEYWORD_SET(ColorIt) THEN Color = 30 $
  ELSE Color = (255B * Magnitude(i) / MaxVelocity ) < 255-1
  ; Plot vectors (arrow or line style):
  IF KEYWORD_SET(Arrow) THEN $
  ARROW, Lon(i), $
    Lat(i), $
    Lon(i)+U(i)*VFactor*Scale, $
    Lat(i)+V(i)*VFactor, $
  /DATA, $
  THICK=VThick, $
  HTHICK=1.0, $
  HSIZE=-0.1, $
  COLOR=Color $
  ELSE $
  PLOTS, [Lon(i), Lon(i)+U(i)*VFactor*Scale], $
    [Lat(i), Lat(i)+V(i)*VFactor], $
  /DATA, $
  THICK=VThick, $
  COLOR=Color
ENDFOR

```

```

; COAST & BATHY

```

```

; Check parameters then plot:
IF N_ELEMENTS(Coast) GT 0 THEN $ ; Check if keyword present
  IF (SIZE(Coast))(0) EQ 2 $ ; check if 2D and 2 lines
  AND (SIZE(Coast))(2) EQ 2 THEN $
  OPLOT,Coast(*,0), $
    Coast(*,1), $
  COLOR=WHITE, $
  MAX_VALUE=Missing $
  ELSE PRINT, 'Wrong data for coast line'

IF N_ELEMENTS(Bathy) GT 0 THEN $ ; Check if keyword present
  IF (SIZE(Bathy))(0) EQ 2 $ ; check if 2D and 2 lines
  AND (SIZE(Bathy))(2) EQ 2 THEN $
  OPLOT,Bathy(*,0), $
    Bathy(*,1), $
  COLOR=DARK_BLUE, $
  MAX_VALUE=Missing $
  ELSE PRINT, 'Wrong data for bathymetry'

```

```

; LEGEND
; Write a title for this window
XYOUTS, 2, !D.Y_SIZE-15, Title, /DEVICE
; Display a color bar on the bottom right of the graphic.
IF KEYWORD_SET(ColorIt) THEN BEGIN
; Setup
Width = CEIL(!D.X_Size/50) & Height = CEIL(!D.Y_Size/4)
X0 = CEIL(!D.X_SIZE - Width) & Y0=0B
Position = FIX( [X0, Y0, X0+Width, Y0+Height] )
NDivisions = 5B
DataRange = [0B, MaxVelocity]
; Create data to fill the color bar:
Bar = REPLICATE(1B, 10) # BINDGEN(256) ; vertical bar
; Scale the color bar:
Bar = BYTSCL(TEMPORARY(Bar))
; Reform Data:
Bar = CONGRID(TEMPORARY(Bar), Width, Height, /MINUS_ONE)
; Fill color bar:
TV, TEMPORARY(Bar), X0, Y0
; Annotate the color bar (plot a box):
PLOT, DataRange, DataRange, $
  /NODATA, $ ; Only the axes, titles, and annotation.
  XTICKS=1, $ ; Number of major tick intervals.
  YTICKS=NDivisions, $ ; Number of major tick intervals.
  XSTYLE=1, $ ; Force exact axis range.
  YSTYLE=1, $ ; Force exact axis range & Suppress box style.
  /DEVICE, $ ; Device coordinates.
  POSITION=Position, $ ; Lower left & upper right corners of
data window.
  COLOR=RED, $ ; Color index of the data, text & line.
  CHARSIZE=0.8, $ ; Character size for the annotation.
  /NOERASE, $ ; Screen or page not to be erased.
  XTICKFORMAT='(A1)', $ ; Kind of deleting labels.
  YTICKLEN=1, $ ; Lengths of tick marks.
  YRANGE=DataRange ; Data range of the axis.
ENDIF ; KEYWORD_SET(ColorIt)

```

```

; CLEANUP

```

```

; Restore index for default color and background color:
!P.Background = OldBg
!P.Color = OldColor
; Return to previous output device if postscript was set:
IF !D.Name EQ 'ps' THEN BEGIN
  DEVICE, /CLOSE

```

```
SET_PLOT, OldDevice
ENDIF
IF KEYWORD_SET(Ps) THEN SET_PLOT, OldDevice
; Re-load previous color table:
TVLCT, OldColorTable
PRINT, 'color table restored'
```

```
END ;===== end of procedure ADCP_VECTOR =====
```

---