
Subject: Re: vectors on maps

Posted by [hcp](#) on Wed, 25 Nov 1998 08:00:00 GMT

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

In article <365B6CBB.F3134DCE@atmos.washington.edu>, Varavut Limpasuvan
<var@atmos.washington.edu> writes:

|> Is it possible to overlay vectors (say wind vectors) on a stereographic
|> (or any other type) projection map? I dont think "velovect" works for
|> this situation. Help.

Not as supplied it doesn't. I hacked it to work in this situation
(doable as it is written in IDL, not a core part of the language).

I append the hacked version below: I changed the name to velo.pro so
it doesn't conflict with the official version. You should be able to do
(e.g.)

```
IDL> map_set,-90,0,0,/stereo,/continents,/isotropic
IDL> velo,zonw,merw,windlongs,windlats,/overplot
```

(The zonal winds are in zonw, the meridional in merw)

I don't promise that the arrows point exactly where you think they will, but they
certainly provide a useful-looking visual aid.

All the best

Hugh

```
.....
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
PRO VELO,U,V,X,Y, Missing = Missing, Length = length, Dots = dots, $
    Color=color, noerase=noerase, xrange=xrange,yrange=yrange, $
    xstyle=xstyle,ystyle=ystyle, _EXTRA = extra,overplot=overplot
;
;+
; NAME:
; VELO
;
; PURPOSE:
; Produce a two-dimensional velocity field plot.
;
; A directed arrow is drawn at each point showing the direction and
; magnitude of the field.
;
; CATEGORY:
; Plotting, two-dimensional.
;
```

```

; CALLING SEQUENCE:
; VELO, U, V [, X, Y]
;
; INPUTS:
; U: The X component of the two-dimensional field.
; U must be a two-dimensional array.
;
; V: The Y component of the two dimensional field. Y must have
; the same dimensions as X. The vector at point (i,j) has a
; magnitude of:
;
;  $(U(i,j)^2 + V(i,j)^2)^{0.5}$ 
;
; and a direction of:
;
;  $ATAN2(V(i,j), U(i,j))$ .
;
; OPTIONAL INPUT PARAMETERS:
; X: Optional abscissae values. X must be a vector with a length
; equal to the first dimension of U and V.
;
; Y: Optional ordinate values. Y must be a vector with a length
; equal to the first dimension of U and V.
;
; KEYWORD INPUT PARAMETERS:
; MISSING: Missing data value. Vectors with a LENGTH greater
; than MISSING are ignored.
;
; LENGTH: Length factor. The default of 1.0 makes the longest (U,V)
; vector the length of a cell.
;
; DOTS: Set this keyword to 1 to place a dot at each missing point.
; Set this keyword to 0 or omit it to draw nothing for missing
; points. Has effect only if MISSING is specified.
;
; COLOR: The color index used for the plot.
;
; Note: All other keywords are passed directly to the PLOT procedure
; and may be used to set option such as TITLE, POSITION,
; NOERASE, etc.
; OUTPUTS:
; None.
;
; COMMON BLOCKS:
; None.
;
; SIDE EFFECTS:
; Plotting on the selected device is performed. System

```

```

; variables concerning plotting are changed.
;
;
; RESTRICTIONS:
; None.
;
;
; PROCEDURE:
; Straightforward. Unrecognized keywords are passed to the PLOT
; procedure.
;
;
; MODIFICATION HISTORY:
; DMS, RSI, Oct., 1983.
; For Sun, DMS, RSI, April, 1989.
; Added TITLE, Oct, 1990.
; Added POSITION, NOERASE, COLOR, Feb 91, RES.
; August, 1993. Vince Patrick, Adv. Visualization Lab, U. of Maryland,
; fixed errors in math.
; August, 1993. DMS, Added _EXTRA keyword inheritance.
; January, 1994, KDB. Fixed integer math which produced 0 and caused
; divide by zero errors.
; December, 1994, MWR. Added _EXTRA inheritance for PLOTS and OPLOT.

; .... Which prevents /noerase from working. /noerase put back as
; explicit keyword by HCP of the meteorology dept
; at the University of Edinburgh
; xrange and yrange also had to be done like this; this might
; be true of any keywords which are useful for plot but which
; cause plots and oplot to foul up. Function name changed to
; VELO to avoid conflicts with the official version.

;-
;
;
; on_error,2 ;Return to caller if an error occurs
; s = size(u)
; t = size(v)
; if s(0) ne 2 then begin
baduv: message, 'U and V parameters must be 2D and same size.'
; endif
; if total(abs(s(0:2)-t(0:2))) ne 0 then goto,baduv
;
; if n_params(0) lt 3 then x = findgen(s(1)) else $
; if n_elements(x) ne s(1) then begin
badxy: message, 'X and Y arrays have incorrect size.'
; endif
; if n_params(1) lt 4 then y = findgen(s(2)) else $
; if n_elements(y) ne s(2) then goto,badxy
;
; if n_elements(missing) le 0 then missing = 1.0e30
; if n_elements(length) le 0 then length = 1.0

```

```

mag = sqrt(u^2.+v^2.)          ;magnitude.
      ;Subscripts of good elements
nbad = 0                        ;# of missing points
if n_elements(missing) gt 0 then begin
    good = where(mag lt missing)
    if keyword_set(dots) then bad = where(mag ge missing, nbad)
endif else begin
    good = lindgen(n_elements(mag))
endelse

if n_elements(xrange) ne 2 then xrange=[x(0),x(n_elements(x)-1)]
if n_elements(yrange) ne 2 then yrange=[y(0),y(n_elements(y)-1)]
if n_elements(xstyle) ne 1 then xstyle=1
if n_elements(ystyle) ne 1 then ystyle=1
ugood = u(good)
vgood = v(good)
x0 = min(x)                    ;get scaling
x1 = max(x)
y0 = min(y)
y1 = max(y)
x_step=float(x1-x0)/float(s(1)) ; Convert to float. Integer math
y_step=float(y1-y0)/float(s(2)) ; could result in divide by 0

maxmag=max([max(abs(ugood/x_step)),max(abs(vgood/y_step))])
sina = length * (ugood/maxmag)
cosa = length * (vgood/maxmag)
;
    if n_elements(title) le 0 then title = "
;----- plot to get axes -----
    if n_elements(color) eq 0 then color = !p.color
    x_b0=x0-x_step
x_b1=x1+x_step
y_b0=y0-y_step
y_b1=y1+y_step
    if n_elements(overplot) ne 1 then overplot=0
    if overplot ne 1 then begin
        if n_elements(position) eq 0 then begin
            plot,[x_b0,x_b1],[y_b1,y_b0],/nodata, $
            color=color, _EXTRA = extra,xstyle=xstyle,$
            noerase=noerase,xrange=xrange,yrange=yrange,ystyle=ystyle
        endif else begin
            plot,[x_b0,x_b1],[y_b1,y_b0],/nodata, $
            color=color, _EXTRA = extra, noerase=noerase,$
            xrange=xrange,yrange=yrange,ystyle=ystyle,xstyle=xstyle
        endelse
    endif
;

```

```

r = .3 ;len of arrow head
angle = 22.5 * !dtr ;Angle of arrowhead
st = r * sin(angle) ;sin 22.5 degs * length of head
ct = r * cos(angle)
;
for i=0,n_elements(good)-1 do begin ;Each point
    x0 = x(good(i) mod s(1)) ;get coords of start & end
    dx = sina(i)
    x1 = x0 + dx
    y0 = y(good(i) / s(1))
    dy = cosa(i)
    y1 = y0 + dy
xd=x_step
yd=y_step
    plots,[x0,x1,x1-(ct*dx/xd-st*dy/yd)*xd, $
x1,x1-(ct*dx/xd+st*dy/yd)*xd], $
    [y0,y1,y1-(ct*dy/yd+st*dx/xd)*yd, $
y1,y1-(ct*dy/yd-st*dx/xd)*yd], $
    color=color, _EXTRA=extra,/clip
endfor
if nbad gt 0 then $ ;Dots for missing?
    oplot, x(bad mod s(1)), y(bad / s(1)), psym=3, color=color, $
    _EXTRA=extra
end

```

--

```

=====
=====
Hugh C. Pumphrey | Telephone 0131-650-6026
Department of Meteorology | FAX :hmm. Time to fix sig file
The University of Edinburgh | Replace 0131 with +44-131 if outside U.K.
EDINBURGH EH9 3JZ, Scotland | Email hcp@met.ed.ac.uk
OBDisclaimer: The views expressed herein are mine, not those of UofE.
=====
=====

```
