Subject: Re: vector plots on maps
Posted by Martin Schultz on Tue, 03 Feb 1998 08:00:00 GMT
View Forum Message <> Reply to Message

mirko_vukovic@notes.mrc.sony.com wrote:
>
> hello,
>
> is there a routine by which I can overplot a vector field on a map?
>
> velovect cannot do the job.
>
> tia,
>
> mirko vukovic

... depends on *why* velovect can't do the job. When I had to do this a while ago, I had the problem that velovect would always autoscale the arrows,
but I wanted wind vectors to be the same scale on every plot I made. So a colleague of mine from Juelich had this routine velov_fr which has some adaptations. Try it (or describe in more detail what's wrong with velovec)!

Regards,
Martin.

---------- -------
Dr. Martin Schultz
Department for Earth&Planetary Sciences, Harvard University
186 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318
fax : (617)-495-4551

e-mail: mgs@io.harvard.edu
IDL-homepage: http://www-as.harvard.edu/people/staff/mgs/idl/
---------- -------

```
PRO VELOV_Fr,U,V,X,Y, Missing = Missing, Length = length, Dots = dots,  $
       Color=color, _EXTRA = extra
;
;+
; NAME:
;      VELOVECT
```

```
;
; 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:
;       VELOVECT, 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 abcissae 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.
;-
;
       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
; FR's
; alt      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

        ugood = u(good)
        vgood = v(good)
        x0 = min(x)                     ;get scaling
        x1 = max(x)
        y0 = min(y)
        y1 = max(y)
        x_step=(x1-x0)/s(1)
        y_step=(y1-y0)/s(2)
;FR's
        maxmag=max([max(ugood/x_step),max(vgood/y_step)])
        if n_elements(length) gt 0 then maxmag=length/x_step
        sina = (ugood/maxmag)
        cosa = (vgood/maxmag)
print,maxmag
; alt       sina = length * (ugood/maxmag)
; alt       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(position) eq 0 then begin
;         plots,[x_b0,x_b1],[y_b1,y_b0],/nodata,/xst,/yst, $
;           color=color, _EXTRA = extra
        endif else begin
;         plots,[x_b0,x_b1],[y_b1,y_b0],/nodata,/xst,/yst, $
;           color=color, _EXTRA = extra
        endelse
;
        r = .3                          ;len of arrow head
        angle = 22.5 * !dtor            ;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
                endfor
;       if nbad gt 0 then $             ;Dots for missing?
;               oplot, x(bad mod s(1)), y(bad / s(1)), psym=3, color=color
end
```

File Attachments

1) velov_fr.pro, downloaded 122 times