

---

Subject: Re: vector plots on maps  
Posted by [Theo Brauers](#) on Thu, 05 Feb 1998 08:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi Mirko.

I had the same problem a few months ago. I solved it by modifying the partvelvec routine written by Joop Schaye (jschaye@astro.rug.nl). see attached file.

Theo

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
>
> -----==== Posted via Deja News =====
> http://www.dejanews.com/ Search, Read, Post to Usenet
;+
; NAME:
; partvelvec
;
; PURPOSE:
; This procedure plots the velocity vectors of particles (at the
; positions of the particles).
;
; CATEGORY:
; Plotting, Two-dimensional.
;
; CALLING SEQUENCE:
; PARTVELVEC, VELX, VELY, POSX, POSY [, X, Y]
;
; INPUTS:
; VELX: An array of any dimension, containing the x-components
; of the particle velocities.
; VELY: An array of the same dimension as velx, containing the
; y-components of the particle velocities.
; POSX: An array of the same dimension as velx, containing the
; x-components of the particle positions.
; POSY: An array of the same dimension as velx, containing the
; y-components of the particle positions.
```

```

;
;
; OPTIONAL INPUTS:
;   X: Optional abscissae values. X must be a vector.
;   Y: Optional ordinate values. Y must be a vector. If only X
;       is specified, then Y is taken equal to be equal to X.
;
;
; OPTIONAL INPUT KEYWORD PARAMETERS:
;   FRACTION: The fraction of the vectors to plot. They are
;               taken at random from the complete sample. Default is
;               FRACTION = 1.0, use all vectors
;
;   LENGTH: The maximum vectorlength relative to the plot data
;             window. Default = 0.08
;
;   COLOR: The color for the vectors, axes and titles.
;           Default=!P.COLOR
;
;   OVER: Plot over the previous plot
;
;   Plot All other keywords available to PLOT are also used
;   Keywords: by this procedure.
;
; OUTPUTS:
;   This procedure plots the velocity vectors (VELX,VELY) at the
;   positions of the particles, (POSX,POSY). If X and Y are not
;   specified, then the size of the plot is such that all vectors
;   just fit within in the plot data window.
;
; SIDE EFFECTS:
;   Plotting on the current device is performed.
;
; EXAMPLE:
;   Generate some particle positions and velocities.
;
;   POSX=RANDOMU(seed,200)
;   POSY=RANDOMU(seed,200)
;   VELX=RANDOMU(seed,200)-0.5
;   VELY=RANDOMU(seed,200)-0.5
;
;   Plot the particle velocities.
;
;   PARTVELVEC, VELX, VELY, POSX, POSY
;
; MODIFICATION HISTORY:
;   Written by: Joop Schaye (jschaye@astro.rug.nl), Sep 1996.
;   Modified: Theo Brauers (th.brauers@fz-juelich.de) Oct. 1997
;               use with maps, incl. debug
;
;-

```

```
PRO partvelvec,velx,vely,posx,posy,x,y, OVER=over, $
    FRACTION=fraction,LENGTH=length,COLOR=color,_EXTRA=extra
```

```
debug, '1.10 T.B. 1997-OCT-20'
```

```
;-----
; Various settings, modify these to customize
;-----
```

```
c={customize, $
    length: 0.08, $ ; Maximum vector length relative to plot region. (*)
    lengtharrow: 0.3, $ ; Length of arrowhead legs relative to vectorlength.
    angle: 22.5 } ; 1/2 times the angle between the arrowhead legs.
```

```
; (*) Not used if keyword LENGTH is present
```

```
;-----
; Some error handling
;-----
```

```
on_error,2 ; Return to caller if an error occurs.
```

```
nparams=n_params()
IF nparams NE 4 THEN BEGIN
    IF (nparams NE 5 AND nparams NE 6) THEN BEGIN
        message,'Wrong number of parameters!'/,continue
        message,'Syntax: PARTVELVEC, VELX, VELY, POSX, POSY [, X, Y]', $
            /noname,/noprefix
    ENDIF
    IF nparams EQ 5 THEN y=x
    sizex=size(x)
    sizey=size(y)
    IF (sizex(0) NE 1 OR sizey(0) NE 1) THEN $
        message,'X and Y must be vectors!'
ENDIF
```

```
sizevelx=size(velx)
sizevely=size(vely)
sizeposx=size(posx)
sizeposy=size(posy)
```

```
IF (total(sizevelx(0:sizevelx(0))-sizevely(0:sizevelx(0))) NE 0 $
    OR total(sizevelx(0:sizevelx(0))-sizeposx(0:sizevelx(0))) NE 0 $
    OR total(sizevelx(0:sizevelx(0))-sizeposy(0:sizevelx(0))) NE 0) THEN $
    message,'All arguments must have the same dimension and size!'
```

```
IF n_elements(fraction) GT 0 THEN $
  IF (fraction LT 0.0 OR fraction GT 1.0) THEN $
    message,'Fraction has to be between 0.0 and 1.0.'
```

```
;-----
; Prepare plot
;-----
```

```
nvecs=n_elements(velx) ; Number of particles.
vel=sqrt(velx^2+vely^2) ; Total velocity.
maxvel=max(vel) ; Maximum velocity.
```

```
; Compute maximum length of vectors.
IF n_elements(length) LE 0 THEN length=c.length
minposx=min(posx)
maxposx=max(posx)
minposy=min(posy)
maxposy=max(posy)
length=length*((maxposx-minposx) > (maxposy-minposy))
```

```
; Convert velocities.
vx=length*velx/maxvel
vy=length*vely/maxvel
vel=length*temporary(vel)/maxvel
```

```
; Make sure no vectors extend beyond the plot data window.
x1=posx+vx ; End of vector.
y1=posy+vy
IF nparams EQ 4 THEN BEGIN
  minposx=min(x1)<minposx
  maxposx=max(x1)>maxposx
  minposy=min(y1)<minposy
  maxposy=max(y1)>maxposy
ENDIF
```

```
angle=c.angle*!dtr ; Convert from degrees to radians.
sinangle=sin(angle) ; Need these.
cosangle=cos(angle)
```

```
;-----
; Plot axes
;-----
```

```
IF n_elements(color) EQ 0 THEN color=!p.color
```

```
IF n_elements(over) EQ 0 THEN BEGIN
```

```

IF nparams EQ 4 THEN $
  plot,[minposx,maxposx],[minposy,maxposy], $
  /nodata,/xstyle,/ystyle,COLOR=color,_EXTRA=extra $
ELSE plot,x,y,/nodata,/xstyle,/ystyle,COLOR=color,_EXTRA=extra
ENDIF

```

```

;-----
; Plot vectors
;-----

```

```

IF n_elements(fraction) GT 0 THEN BEGIN
  IF fraction EQ 1.0 THEN GOTO,plotall
  nrgood=long(fraction*nvecs) ; # of vectors to plot.
  IF nrgood EQ 0 THEN return
  ; Compute indices of vectors to plot. I use two lines to get more
  ; random "random numbers".
  good=long(randomu(seed,nrgood+1)*(nvecs-1.0))
  good=good(1:*)
  vx=temporary(vx(good))
  vy=temporary(vy(good))
  px=posx(good) ; Can't use temporary if we wan't to keep the data.
  py=posy(good)
  x1=temporary(x1(good))
  y1=temporary(y1(good))
  nvecs=nrgood
ENDIF ELSE BEGIN
plotall:
  px=posx
  py=posy
ENDELSE

```

```

FOR i=0I,nvecs-1I DO BEGIN ; Loop over particles.
  ; Note that we cannot put the next three lines outside the loop,
  ; because we want the arrow size to be relative to the vector length.
  r=c.lengtharrow*vel(i) ; Length of arrow head.
  rsin=r*sinangle
  rcos=r*cosangle
  ; Draw basis, arrow leg, same arrow leg, other arrow leg.
  ; One arrow leg is drawn twice, because we need to return to the end
  ; of the vector to draw the other leg.
  plots,[px(i),x1(i),x1(i)-(vx(i)*rcos+vy(i)*rsin)/vel(i), $
        x1(i),x1(i)-(vx(i)*rcos-vy(i)*rsin)/vel(i)], $
        [py(i),y1(i),y1(i)-(vy(i)*rcos-vx(i)*rsin)/vel(i), $
        y1(i),y1(i)-(vy(i)*rcos+vx(i)*rsin)/vel(i)],COLOR=color
ENDFOR

```

```

END ; End of procedure PARTVELVEC.

```

## File Attachments

---

1) [partvelvec.pro](#), downloaded 128 times

---