
Subject: Re: 3D Vector Field Plot?

Posted by [marc schellens\[1\]](#) on Tue, 30 May 2000 07:00:00 GMT

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

John Boccio wrote:

>
> Hi,
>
> Is there any straightforward way to plot
> directed arrows representing a vector field
> in 3 dimensions?
>
> Anyone have a sample code for direct graphics?
> using IDLgrpolyline?
>
> Thanks,
>
> John Boccio
> boccio@swarthmore.edu

Here is a small sample function I extracted from one of my programs.

I simplified it and did not test it, but I think you can get the idea.

The 'trick' is that the top of the vectors consist of two more

IDLGrPolyline

objects which are thicker than the rest of the vector to fake an arrow.

Another thicker part is drawn at the actual position of the vector

(the middle of the vector line here).

With this you save the need to calculate the projection for the vector
top.

I think therefore this gives not the nicest vectors, but it should be
quite fast.

Just add the returned objarr to a IDLGrModel.

Hope this is helpful.

:-) marc

```
;; ---
```

```
;; vectorPosition, vectorDirection: [3,numVectors] float arrays
```

```
;; colorArray: [3,numVectors] byte array for color of each vector
```

```
function sample,vectorPosition,vectorDirection,colorArray
```

```
vectorObj=objarr(4)
```

```
line1=[[vectorPosition[*,*]-vectorDirection[*,*]],  
        [vectorPosition[*,*]+vectorDirection[*,*]]]
```

```
linkList=intarr(3,numPlot)
```

```
linkList[0,*]=2
```

```

linkList[1,*]=indgen(numPlot)
linkList[2,*]=indgen(numPlot)+numPlot

;; if we do not do this, polyline thinks we are using colortable
if numVectors eq 1 then vertColors=[[vertColors],[vertColors]]

line2=[[vectorPosition[*,*]+vectorDirection[*,*]*0.7],[vectorPosition[*,*]+vectorDirection[*,*]*.8]]
line3=[[vectorPosition[*,*]+vectorDirection[*,*]*0.8],[vectorPosition[*,*]+vectorDirection[*,*]*.9]]
line4=[[vectorPosition[*,*]-vectorDirection[*,*]*0.05],[vectorPosition[*,*]+vectorDirection[*,*]*.05]]

vectorObj[0]=obj_new('IDLgrPolyline',$
    line1,$
    POLYLINES=linkList,$
    VERT_COLORS=vertColors,$
    THICK=arrowThickness,$
    COLOR=colorArray)

vectorObj[1]=obj_new('IDLgrPolyline',$
    line2,$
    POLYLINES=linkList,$
    VERT_COLORS=vertColors,$
    THICK=arrowThickness*4,$
    COLOR=colorArray)

vectorObj[2]=obj_new('IDLgrPolyline',$
    line3,$
    POLYLINES=linkList,$
    VERT_COLORS=vertColors,$
    THICK=arrowThickness*2,$
    COLOR=colorArray)

vectorObj[3]=obj_new('IDLgrPolyline',$
    line4,$
    POLYLINES=linkList,$
    VERT_COLORS=vertColors,$
    THICK=arrowThickness*2,$
    COLOR=colorArray)

return,vectorObj
end

```

Subject: Re: 3D Vector Field Plot?

Posted by [Mark Hadfield](#) on Wed, 31 May 2000 07:00:00 GMT

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

"Mark Hadfield" <m.hadfield@niwa.cri.nz> wrote in message
news:959724982.192372@clam-ext...

>
> There is a class called MGHgrBarbPlot on my IDL page (see URLs below). It
> uses an IDLgrPolyline to represent a collection of vector barbs with
origins
> specified by properties DATAx, DATAy & DATAz and lengths specified by
> properties DATAu, DATAv, DATAw.

Correction: there is no DATAw property. It would be simple to add one
though.

Mark Hadfield
m.hadfield@niwa.cri.nz <http://katipo.niwa.cri.nz/~hadfield/>
National Institute for Water and Atmospheric Research
PO Box 14-901, Wellington, New Zealand

Subject: Re: 3D Vector Field Plot?
Posted by [Mark Hadfield](#) on Wed, 31 May 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

"John Boccio" <boccio@swarthmore.edu> wrote in message
news:250520001319518743%boccio@swarthmore.edu...

> Hi,
>
> Is there any straightforward way to plot
> directed arrows representing a vector field
> in 3 dimensions?
>
> Anyone have a sample code for direct graphics?
> using IDLgrpolyline?

There is a class called MGHgrBarbPlot on my IDL page (see URLs below). It
uses an IDLgrPolyline to represent a collection of vector barbs with origins
specified by properties DATAx, DATAy & DATAz and lengths specified by
properties DATAu, DATAv, DATAw. The geometry of the barb origins can be 1D,
2D, 3D, scattered, whatever. One slightly tricky thing that it handles
automatically is to get the aspect ratio "square", at least in normalised
coordinates. (Now that I think of it, it doesn't concern it self with
squareness in the vertical dimension--this would be a simple extension.)
However it doesn't meet your requirements in one significant way: there are
no heads on the arrows. My favoured way to make the array direction
unambiguous is to draw a symbol at each origin.

If you are rolling your own solution in object graphics I have one
suggestion: Use a single IDLgrPolyline for the entire collection of arrows
instead of a separate one for each arrow or part thereof. For large
collections the former is much faster.

URLs:

IDL page:

<http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/>

MGHgrBarbPlot:

http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/mghgrb_arbplot__define.pro

MGHgrBarbPlot example:

http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/mgh_example_barb.pro

Collection of all routines:

http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/MARKS_ROUTINES.tar.gz

http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/MARKS_ROUTINES.zip

Mark Hadfield

m.hadfield@niwa.cri.nz <http://katipo.niwa.cri.nz/~hadfield/>

National Institute for Water and Atmospheric Research

PO Box 14-901, Wellington, New Zealand
