
Subject: Re: plotting vectors in 3D

Posted by [John Boccio](#) on Mon, 06 Nov 2000 08:00:00 GMT

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

In article <8tu9l1\$uhb\$1@peque.uv.es>, M Carmen Gonzalez
<mcgonzal@uv.es> wrote:

> mcgonzal@uv.es

Hi,

Here is a very, very crude program that we use.
No fancy vector objects (just a polyline with
small sphere on end for each vector).

No complaints please from all the experts on the list!

It is an object-based program so that you can
use the mouse to find best view. Requires routine
xobjview in V5.3.

The example is a simple radial field emanating from
the corner of the box.

John Boccio
boccio@swarthmore.edu

----- cut here -----

```
function fx,x,y,z
xc=-0.2
yc=-0.2
zc=-0.2
r=sqrt((x-xc)^2+(y-yc)^2+(z-zc)^2)
return,(x-xc)/r^2
end
```

```
function fy,x,y,z
xc=-0.2
yc=-0.2
zc=-0.2
r=sqrt((x-xc)^2+(y-yc)^2+(z-zc)^2)
return,(y-yc)/r^2
end
```

```
function fz,x,y,z
xc=-0.2
```

```

yc=-0.2
zc=-0.2
r=sqrt((x-xc)^2+(y-yc)^2+(z-zc)^2)
return,(z-zc)/r^2
end

```

```

pro vectfield3d
;Create model to contain surface
omodel = OBJ_NEW('IDLgrModel')
;create vector field
n=6
range=1.0
fac=0.05
step=range/(n-1)
x=step*findgen(n)
y=x
z=x
vx=fltarr(n,n,n)
vy=vx
vz=vx
v=vx
x1=vx
y1=vx
z1=vx
m=n-1
for i=0,m do begin
  for j=0,m do begin
    for k=0,m do begin
      vx(i,j,k)=fx(x(i),y(j),z(k))
      vy(i,j,k)=fy(x(i),y(j),z(k))
      vz(i,j,k)=fz(x(i),y(j),z(k))
    endfor
  endfor
endfor
for i=0,m do begin
  for j=0,m do begin
    for k=0,m do begin
      x1(i,j,k)=x(i)+fac*vx(i,j,k)
      y1(i,j,k)=y(j)+fac*vy(i,j,k)
      z1(i,j,k)=z(k)+fac*vz(i,j,k)
    endfor
  endfor
endfor
; normalize field
mnx=min([min(x),min(x1)])
mxx=max([max(x),max(x1)])
mny=min([min(y),min(y1)])
mxy=max([max(y),max(y1)])

```

```

mnz=min([min(z),min(z1)])
mxz=max([max(z),max(z1)])
max1=max([mxx,mxy,mxz])
min1=min([mnx,mny,mnz])
x1=-1.0+2.0*(x1-min1)/(max1-min1)
y1=-1.0+2.0*(y1-min1)/(max1-min1)
z1=-1.0+2.0*(z1-min1)/(max1-min1)
x=-1.0+2.0*(x-min1)/(max1-min1)
y=-1.0+2.0*(y-min1)/(max1-min1)
z=-1.0+2.0*(z-min1)/(max1-min1)
; create field vectors
for i=0,m do begin
  for j=0,m do begin
    for k=0,m do begin
      xs=[x1(i,j,k),x(i)]
      ys=[y1(i,j,k),y(j)]
      zs=[z1(i,j,k),z(k)]
      opolyline = OBJ_NEW('IDLgrPolyline', xs,ys,zs, $
        color=[255,0,0],thick=1,shading=0)
      omodel -> ADD, opolyline
      mesh_obj,4,verts,conn,Replicate(0.01 ,6,6)
      T3d,/Reset
      T3d, Translate=[x1(i,j,k),y1(i,j,k),z1(i,j,k)]
      verts=Vert_T3d(verts)
      oSphere=OBJ_NEW('IDLgrpolygon', verts,poly=conn,
COLOR=[255,0,0],style=0)
      omodel -> ADD, oSphere
    endfor
  endfor
endfor
;create box
opolyline = OBJ_NEW('IDLgrPolyline',
[-1.0,1.0],[-1.0,-1.0],[-1.0,-1.0], $
  color=[0,255,0],thick=1,shading=0)
omodel -> ADD, opolyline
opolyline = OBJ_NEW('IDLgrPolyline',
[-1.0,-1.0],[-1.0,-1.0],[-1.0,1.0], $
  color=[0,255,0],thick=1,shading=0)
omodel -> ADD, opolyline
opolyline = OBJ_NEW('IDLgrPolyline', [-1.0,1.0],[-1.0,-1.0],[1.0,1.0], $
  color=[0,255,0],thick=1,shading=0)
omodel -> ADD, opolyline
opolyline = OBJ_NEW('IDLgrPolyline', [1.0,1.0],[-1.0,-1.0],[-1.0,1.0], $
  color=[0,255,0],thick=1,shading=0)
omodel -> ADD, opolyline
opolyline = OBJ_NEW('IDLgrPolyline', [1.0,1.0],[1.0,1.0],[-1.0,-1.0], $
  color=[0,255,0],thick=1,shading=0)
omodel -> ADD, opolyline

```

```

opolyline = OBJ_NEW('IDLgrPolyline', [1.0,1.0],[1.0,1.0],[-1.0,1.0], $
    color=[0,255,0],thick=1,shading=0)
omodel -> ADD, opolyline
opolyline = OBJ_NEW('IDLgrPolyline', [-1.0,-1.0],[-1.0,1.0],[1.0,1.0], $
    color=[0,255,0],thick=1,shading=0)
omodel -> ADD, opolyline
opolyline = OBJ_NEW('IDLgrPolyline',
[-1.0,-1.0],[-1.0,1.0],[-1.0,-1.0], $
    color=[0,255,0],thick=1,shading=0)
omodel -> ADD, opolyline
opolyline = OBJ_NEW('IDLgrPolyline', [-1.0,-1.0],[1.0,1.0],[-1.0,1.0], $
    color=[0,255,0],thick=1,shading=0)
omodel -> ADD, opolyline
opolyline = OBJ_NEW('IDLgrPolyline', [-1.0,1.0],[1.0,1.0],[1.0,1.0], $
    color=[0,255,0],thick=1,shading=0)
omodel -> ADD, opolyline
opolyline = OBJ_NEW('IDLgrPolyline', [1.0,1.0],[-1.0,1.0],[1.0,1.0], $
    color=[0,255,0],thick=1,shading=0)
omodel -> ADD, opolyline
opolyline = OBJ_NEW('IDLgrPolyline', [-1.0,1.0],[1.0,1.0],[-1.0,-1.0], $
    color=[0,255,0],thick=1,shading=0)
omodel -> ADD, opolyline

;Create a fixed light
ofixedlight = OBJ_NEW('IDLgrLight', TYPE = 1, $
    LOCATION = [-1,-1,1], COLOR = [128,128,128])
omodel -> ADD, ofixedlight
;Create another fixed light
ofixedlight1 = OBJ_NEW('IDLgrLight', TYPE = 1, $
    LOCATION = [-1,1,1], COLOR = [128,128,128])
omodel -> ADD, ofixedlight1
;Create movable light that goes with surface
olight = OBJ_NEW('IDLgrLight', TYPE = 2, LOCATION = [-1,-1,-1])

omodel -> ADD, olight
xobjview,omodel
end

```
