
Subject: Re: Creating mesh from XYZ points
Posted by [Dick Jackson](#) on Fri, 15 Mar 2013 17:30:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

[sorry if this is a duplicate]

Hi Mairan,

Great to hear this is working out. In case you want to try putting the axes in the XObjView window, I added code in here to get you started.

This result isn't quite the final result you're looking for. One issue I see is that if we put the axes in, you don't want to rotate the space by 41 degrees, you want to put the meshes into an IDLgrModel and rotate *that* by 41 degrees (using a transformation matrix found through T3D, for example). Then you would only use 90-degree XObjView_Rotate calls for the three views.

I hope this makes enough sense, but feel free to follow this up further with me.

(I hope my "delta" characters come out OK here! :-)

PRO QHullTest

```
xyz1 = RandomU(seed, 3, 100) + Rebin([0, 0.35, 0.35], 3, 100)
xyz2 = RandomU(seed, 3, 100) + Rebin([0, -0.35, -0.35], 3, 100)
```

```
QHull, xyz1, tri1
QHull, xyz2, tri2
```

```
:: Make connectivity arrays from list of triangles
```

```
conn1 = [Replicate(3, 1, (N_Elements(tri1)/3)), tri1]
conn2 = [Replicate(3, 1, (N_Elements(tri2)/3)), tri2]
```

```
oMesh1 = Obj_New('IDLgrPolygon', xyz1, Polygons=conn1, Style=2, Color=[255,0,0])
; You probably want to use Vert_Colors to add the colour information!
oMesh2 = Obj_New('IDLgrPolygon', xyz2, Polygons=conn2, Style=2, Color=[0,0,255])
```

```
:: Create X, Y, Z axes
```

```
axisColor = [0, 0, 0]
minXYZ = Min([[xyz1],[xyz2]], Dimension=2, Max=maxXYZ)
oAxes = ObjArr(3)
xyzStrs = ['X','Y','Z']
```

```
FOR axisI = 0, 2 DO BEGIN
```

```

        Color=axisColor)
    oAxes[axisI] = Obj_New('IDLgrAxis', axisI, /Exact, $
        Range=[minXYZ[axisI], maxXYZ[axisI]], $
        title=oAxisTitle, ticklen=.05, $
        Color=axisColor, TickDir=1)
ENDFOR

XObjView, [oMesh1, oMesh2, oAxes], $
    XSize=200, YSize=300, Scale=(1/Sqrt(3)), $ ; Default scale 1/Sqrt(3)
    TLB=wXObjViewTLB

inc = 41

XObjView_Rotate, [1,0,0], inc-90. ; +inc around X
XObjView_Write_Image, 'snapYvsX.png', 'png'

XObjView_Rotate, [0, 1, 0], -90 ; -90 around Y
XObjView_Write_Image, 'snapYvsZ.png', 'png'

XObjView_Rotate, [1, 0, 0], +90 ; +90 around X
XObjView_Write_Image, 'snapXvsZ.png', 'png'

END

```

Cheers,
-Dick

Dick Jackson Software Consulting
Victoria, BC, Canada
www.d-jackson.com

Mairan Teodoro wrote:> Hi Dick,

```

>
> That's exactly what I wanted! Thanks!
>
> Regarding the viewing, I had to use this version of your code to get the same
orientation as in the PDF I sent before (note that in my code, the variable inc
is equal to 41 degrees):
>
> XObjView_Rotate, [1,0,0], inc-90. ; +inc around X
> XObjView_Write_Image, 'snapYvsX.png', 'png'
>
> XObjView_Rotate, [0, 1, 0], -90 ; -90 around Y
> XObjView_Write_Image, 'snapYvsZ.png', 'png'
>
> XObjView_Rotate, [1, 0, 0], +90 ; +90 around X

```

> XObjView_Write_Image, 'snapXvsZ.png', 'png'
>
> With your help, I can now make the plots the way I wanted. But I still miss
the axes, though. I was thinking about reading the output images from your code
into a separate program to overplot the axes. This might be a workaround.
>
> Thanks again for your help!
>
> Cheers,
> m.
