Subject: Creating mesh from XYZ points
Posted by mairan.teodoro on Wed, 13 Mar 2013 15:51:25 GMT
View Forum Message <> Reply to Message

Hi all!

I would like some advise on how to create a mesh grid from a set of XYZ coordinates.

Please, take a look at this figure to have an idea on what I'm talking about:

https://dl.dropbox.com/u/6573328/sample.png

The 3 plots show each of the points at their respective XYZ coordinate but I would like to show a "solid" structure with a surface and not a bunch of points.

I tried using the TRIANGULATE/TRIGRID approach without achieving what I want. In fact, the figure in the following link is the result from this approach:

https://dl.dropbox.com/u/6573328/sample2.png

Thank you in advance for any help.

Cheers,
m.

Subject: Re: Creating mesh from XYZ points
Posted by Dick Jackson on Thu, 14 Mar 2013 05:58:22 GMT
View Forum Message <> Reply to Message

Hi Mairan,

TRIANGULATE will work well on a set of 2-D points, so we'll need another trick for your XYZ points.

Option 1: Two convex hulls

These look like two convex shapes that may or may not intersect (it looks like they do). Would it suffice to make two solid intersecting meshes? I'll assume you have (or can make) these arrays:

xyz1, a FltArr(3, nPts1) with XYZ values for mesh 1
xyz2, a FltArr(3, nPts2) with XYZ values for mesh 2

```
;;    Random point sets that should make two intersecting lumps

xyz1 = RandomU(seed, 3, 100)
```

```
xyz2 = RandomU(seed, 3, 100)+0.7

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])

XObjView, [oMesh1, oMesh2]
```

Option 2: Two meshes made by Mesh_Obj

If you are generating these points on a spherical grid or some other regular way, then you may be able to use Mesh_Obj to make each of the grids. If Option 1 didn't meet your needs, post more info on how the points are generated.

Speaking of Mesh_Obj, if anyone's looking for my code to make the animated GIF for my spinning-balloon-letters logo, I moved the link off of my home page to my website's "Other" page. See the logo at www.d-jackson.com (click for a big rendering) and the code is here:
   http://www.d-jackson.com/images/djsclogo.pro

Cheers,
-Dick

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

On Wednesday, March 13, 2013 8:51:25 AM UTC-7, Mairan Teodoro wrote:
> Hi all!
>
> I would like some advise on how to create a mesh grid from a set of XYZ coordinates.
>
> Please, take a look at this figure to have an idea on what I'm talking about:
>
> https://dl.dropbox.com/u/6573328/sample.png
>
> The 3 plots show each of the points at their respective XYZ coordinate but I would like to show a "solid" structure with a surface and not a bunch of points.

>
> I tried using the TRIANGULATE/TRIGRID approach without achieving what I want. In fact, the figure in the following link is the result from this approach:
>
> https://dl.dropbox.com/u/6573328/sample2.png
>
> Thank you in advance for any help.
>
> Cheers,
>
> m.

Subject: Re: Creating mesh from XYZ points
Posted by mairan.teodoro on Thu, 14 Mar 2013 17:14:59 GMT

View Forum Message <> Reply to Message

Hi Dick,

Thank you very much for your valuable help!

Indeed, now I'm able to create what I wanted using Option 1 as you can see in this link:

https://dl.dropbox.com/u/6573328/hom_model2

However, I would need to be able to set the viewing parameters such as inclination angle and scale (like in the 2D figure I showed before). Do you think this is possible within Option 1?

Again, thank you for you help!
Cheers,
m.

Subject: Re: Creating mesh from XYZ points
Posted by Dick Jackson on Thu, 14 Mar 2013 18:37:40 GMT

View Forum Message <> Reply to Message

Hi Mairan,

Good to hear this is working so far. There are options to set scale
and window size in XObjView, and an XObjView_Rotate to spin it. Bonus:
XObjView_Write_Image to save a snapshot to a file.

There's not a simple way to get different axis scaling as you had in
your original triple-plot, but for this, I feel that keeping them the
same (isotropic) in all views is preferable for communicating the true
3-D shape.

I believe I have set up exactly the views you want (YvsX, YvsZ, XvsZ), but I'm not sure if your original views in the PDF were rendering exactly right. (if the second plot were correct, I'd expect in the first plot that the middle overlap section would show the red lump in front, not the blue)

I hope this helps!

```
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])

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

XObjView, Refresh=wXObjViewTLB  ; Ensure that scene is drawn
XObjView_Write_Image, 'snapYvsX.png', 'png'
Wait, 1
XObjView_Rotate, [0, 1, 0], 90 ; +90 around Y
XObjView_Write_Image, 'snapYvsZ.png', 'png'
Wait, 1
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

On Mar 14, 10:14 am, Mairan Teodoro <mairan.teod...@gmail.com> wrote:
> Hi Dick,
>
> Thank you very much for your valuable help!
>
> Indeed, now I'm able to create what I wanted using Option 1 as you can see in this link:
>
> https://dl.dropbox.com/u/6573328/hom_model2
>
> However, I would need to be able to set the viewing parameters such as inclination angle and scale (like in the 2D figure I showed before). Do you think this is possible within Option 1?
>
> Again, thank you for you help!
> Cheers,
> m.

Subject: Re: Creating mesh from XYZ points
Posted by mairan.teodoro on Fri, 15 Mar 2013 13:53:12 GMT
View Forum Message <> Reply to Message

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.

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.