Subject: Re: 3D Objects
Posted by rtowler on Mon, 30 Mar 2009 19:03:03 GMT
View Forum Message <> Reply to Message

On Mar 30, 7:52 am, JayDog wrote:
> Cheers Rick,
> What I actually have is a set of 2D points (x,y) which define an
> ellipse, then stacking many ellipses of varying size in the z-
> direction gives me what appears to be a 3D structure outlining a type
> of ellipsoid. What I would like to do is fill-in the structure so that
> I can make it a surface which I can then perform different operations
> on, such as taking measurements of widths etc. or doing some form of
> illumination with objects light source or other.

Flashback... I used to do this with fish. You have a couple of options. The easiest is to use MESH_OBJ to construct your stacked ellipsoid object by extruding your x,y ellipsoids in z and them assembling your individual IDLgrPolygons. If your ellipsoid vertices are regular(ish) and share the same number of verts between them you can mesh the entire set. (you can of course mesh the object if the number of vertices differ between ellipsoids but this is more difficult.)

Attached is an excerpt from my meshing code. It generates the vertex and polygon connectivity array that you can then pass to IDLgrPolygon. It is simple, but it works. If you're going to use the MESH_VOLUME or MESH_SURFACEAREA functions you will want to verify that the polygons are wound correctly (I believe they are, but have not verified this).

-Rick

xyz = struct containing arrays of X,Y, and Z vertex data in the form [nEllipsoids, mVertsPerEllipsoid]
numcyl = the number of ellipsoids in your vertex array (same as nEllipsoids)
nRollPts = the number of x,y vertices in your ellipsoid (same as mVertsPerEllipsoid).

```
    ;  Create the vertices and mesh connectivity arrays
    vertices = FLTARR(3,nRollPts * numcyl, /nozero)
    polygons =LONARR(5 * nRollPts * (numcyl-1), /nozero)

    for icyl=0, numcyl-1 do begin
        vertices[0,icyl * nRollPts:(icyl+1) * nRollPts-1]= xyz.z
[icyl,*] / 2.
```

```
        vertices[1,icyl * nRollPts:(icyl+1) * nRollPts-1]= xyz.y
[icyl,*]
        vertices[2,icyl * nRollPts:(icyl+1) * nRollPts-1]= xyz.x
[icyl,*]
        if (icyl lt (numcyl-1)) then begin
            roll_ids = LINDGEN(nRollPts)
            conn_off = nRollPts * icyl * 5
            count_ids = (roll_ids * 5) + conn_off
            cyl0_off = nRollPts * icyl
            cyl1_off = nRollPts * (icyl+1)
            polygons[count_ids] = 4
            polygons[count_ids+1] = roll_ids + cyl0_off
            polygons[count_ids+2] = (roll_ids+1) MOD nRollPts +
cyl0_off
            polygons[count_ids+3] = (roll_ids+1) MOD nRollPts +
cyl1_off
            polygons[count_ids+4] = roll_ids + cyl1_off
        endif
    endfor
```