Subject: Re: IDLgrPolygon unexpected result Posted by Dick Jackson on Thu, 28 Feb 2013 09:03:30 GMT

View Forum Message <> Reply to Message

On Wednesday, February 27, 2013 10:41:58 AM UTC-8, nata wrote:

- > Hi guys,
- >
- > I am trying to create a polygon using the following set of points. I want my IDLgrPolygon to look like the plot so I add some vertices to close the shape.
- >
- > I don't understand what is going on here. I cannot get the polygon correctly. What am I missing?
- >
- > Thank you in advance for your help,
- >
- > nata

Hi nata,

IDL's rendering of polygons can really only display triangles and quadrilaterals. What happens when you give a list of points [p0, p1, ... pn] to IDLgrPolygon is that it makes triangles of them using points [p0, p1, p2], then [p0, p2, p3], then [p0, p3, p4] and so on, which is what you saw as a result, and *not* what you want.

Imagine 54 squares side-by-side, so that each pair of neighbouring squares shares a side and two vertices (this is a strip of quadrilaterals, or a quad-strip). Then imagine leaving all the bottom vertices where they are, and pulling the top ones up or down to different heights to make the shape of your plot. That's how we'll make the polygon mesh that will be your filled plot.

A handy routine for making a quad-strip is Mesh_Obj. Its documentation is tricky, but it works. We use type 1 (rectangular), give it a bunch of zeroes for desired Z values, give it the vector of X values, and leave it to create Y values of 0 and 1. It returns 2*n points in a [3, 2*n] array of XYZ values (verts). All the y=0 points are first (rows 0..n-1), then the y=1 points follow (rows n..2*n-1). We set the desired Y values as a second step. We also get the connectivity array (conn) that describes which vertices to use for each quadrilateral, just the way IDLgrPolygon wants it.

To convince yourself of how it's built, in XObjView, choose View:Drag Quality:Low, then drag the mouse to see the wireframe showing the quadrilaterals.

```
 \begin{array}{l} x = & [0.000, 0.019, 0.037, 0.056, 0.074, 0.093, 0.111, 0.130, 0.148, 0.1\\ 67, 0.185, 0.204, 0.222, 0.241, 0.259, 0.278, 0.296, 0.315, \$\\ & 0.333, 0.352, 0.370, 0.389, 0.407, 0.426, 0.444, 0.463, 0.481, 0.500,\\ 0.519, 0.537, 0.556, 0.574, 0.593, 0.611, 0.630, 0.648, \$\\ & 0.667, 0.685, 0.704, 0.722, 0.741, 0.759, 0.778, 0.796, 0.815, 0.833,\\ 0.852, 0.870, 0.889, 0.907, 0.926, 0.944, 0.963, 0.981, 1.000] \end{array}
```

y = [0.000, 0.

 $02,0.000,0.029,0.037,0.037,0.020,0.019,0.036,0.046,\$\\0.044,0.049,0.074,0.147,0.186,0.217,0.262,0.399,0.416,0.411,\\0.391,0.577,0.459,0.376,0.318,0.387,0.316,0.431,\$\\0.421,0.318,0.413,0.628,0.665,0.541,0.714,0.536,0.594,0.606,\\0.437,0.804,0.594,0.629,0.836,0.631,0.827,0.910,1.000]$

PLOT, x, y, /XS, /YS

N=N_ELEMENTS(X)

Mesh_Obj, 1, verts, conn, FltArr(n, 2), P1=x ; Provide X values, receive verts and conn verts[1, n:n+n-1] = y ; Set the desired Y values for the plot shape poly = OBJ_NEW('IDLgrPolygon', verts, Polygons=conn) XObjView, poly

Hope this helps!

-Dick

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