

j.vanknippenberg writes:

- > Thanks for the input so far. :)
- >
- > I'll have a look at it, but if the person who prefers not to be famous
- > :p can provide us with an example, it would be greatly appreciated ;)

OK, after much to-ing and fro-ing, I think we have a solution in search of an explanation.

Because I grew up in a direct graphics world (in my day the most exciting computer graphics was done with monospaced fonts on huge line printers) I like to use a viewplane coordinate system that goes from 0 to 1, or something close to it. And I position my axes and data and whatnot into that coordinate system with my NORMALIZE function, which produces a two-element array for scaling and translating data into this coordinate system.

In the SCATTER\_SURFACE code we are talking about, I do this to position the three axes:

```
xs = Normalize(xrange, Position=[-0.5,0.5])
ys = Normalize(yrange, Position=[-0.5,0.5])
zs = Normalize(zrange, Position=[-0.5,0.5])
```

```
; Scale the axes and place them in the coordinate space.
; Note that not all values in the Location keyword are
; used. (I've put really large values into the positions
; that are not being used to demonstrate this.) For
; example, with the X axis only the Y and Z locations are used.
```

```
xAxis->SetProperty, Location=[9999.0, -0.5, -0.5], XCoord_Conv=xs
yAxis->SetProperty, Location=[-0.5, 9999.0, -0.5], YCoord_Conv=ys
zAxis->SetProperty, Location=[-0.5, 0.5, 9999.0], ZCoord_Conv=zs
```

Note the use of the [XYZ]COORD\_CONV keywords. These are meant to scale and translate (if I read the documentation correctly) the object from it's native data coordinate system into MY coordinate system.

So far, so good. The axes end up where they are suppose to be, etc. But in the very next line, I also use the [XYZ]COORD\_CONV keywords to position the orbs (heads of the pins) and lines (shank of the pins)

in my data coordinate system:

```
FOR j=0,npts-1 DO BEGIN
  (line[j]) -> SetProperty, XCoord_Conv=xs, $
    YCoord_Conv=ys, ZCoord_Conv=zs
  (orbs[j]) -> SetProperty, XCoord_Conv=xs, $
    YCoord_Conv=ys, ZCoord_Conv=zs
ENDFOR
```

This appears to be the problem. When the range of the X, Y, and Z axes are similar, the orb appears as I expect it to appear, as a nice ball-like structure. But if, say, the X range is 100 times larger than the Y and Z range, the orb turns into a flat disk.

I have learned in discussions this morning that I can avoid this problem if I *\*don't\** use the [XYZ]COORD\_CONV keywords with the orbs, but simply position them with the DATA keyword *\*while\** scaling and translating them. Huh!?

Here is the corrected code:

```
FOR j=0,npts-1 DO BEGIN
  (line[j]) -> SetProperty, XCoord_Conv=xs, $
    YCoord_Conv=ys, ZCoord_Conv=zs
  (line[j]) -> GetProperty, Data = Data
  orbs[j] -> SetProperty, Pos = [data[0, 1]*xs[1] + xs[0], $
    data[1, 1]*ys[1] + ys[0], data[2, 1]*zs[1] + zs[0]]
ENDFOR
```

This appears to work for any axis data range. But *\*WHY\** it works is a complete mystery to me. I'd be interested in hearing any good theories. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

---