
Subject: Re: Displaying Cartesian coordinate data on a sphere

Posted by [dplatten](#) on Tue, 11 Dec 2012 15:28:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi David,

Thanks for the reply. I don't think that it was clumped data - I think it was just my lack of understanding of how the GRIDDATA command works. I've put together a simple test scenario so that I can check my sanity. It behaves as I expect, so I think things are OK. My test creates a data point at the north and south poles, and four around the equator and then interpolates between.

```
x = [0, 0, 1, -1, 0, 0]
```

```
y = [0, 0, 0, 0, 1, -1]
```

```
z = [1, -1, 0, 0, 0, 0]
```

```
values = [255, 255, 50, 50, 150, 150]
```

```
grid = GRIDDATA(x, y, z, values, DIMENSION=[30, 30], /SPHERE)
```

```
image = BYTSCL(grid)
```

```
MESH_OBJ, 4, vertices, polygons, REPLICATE(0.25, 101, 101)
```

```
oModel = OBJ_NEW('IDLgrModel')
```

```
oPalette = OBJ_NEW('IDLgrPalette')
```

```
oPalette -> LOADCT, 33
```

```
oPalette -> SetRGB, 255, 255, 255, 255
```

```
oImage = OBJ_NEW('IDLgrImage', image, PALETTE = oPalette)
```

```
vector = FINDGEN(101)/100.
```

```
texture_coordinates = FLTARR(2, 101, 101)
```

```
texture_coordinates[0, *, *] = vector # REPLICATE(1., 101)
```

```
texture_coordinates[1, *, *] = REPLICATE(1., 101) # vector
```

```
oPolygons = OBJ_NEW('IDLgrPolygon', $
```

```
    DATA = vertices, POLYGONS = polygons, $
```

```
    COLOR = [255, 255, 255], $
```

```
    TEXTURE_COORD = texture_coordinates, $
```

```
    TEXTURE_MAP = oImage, /TEXTURE_INTERP)
```

```
oModel -> ADD, oPolygons
```

```
oModel -> ROTATE, [1, 0, 0], -90
```

```
oModel -> ROTATE, [0, 1, 0], -90
```

```
XOBJVIEW, oModel
```

Regards,

David
