Subject: Best way to generate arrays of coordinates for hypersurface calculations?
Posted by James[2] on Tue, 06 Apr 2010 20:47:26 GMT
View Forum Message <> Reply to Message

I'm interested in making a 3D array, representing a volume, where the voxels inside a cylinder have high values and the rest are zero. I suppose this would be called a 3D hypersurface. When dealing with 2D plots, I have generally made two coordinate arrays, one with values increasing across columns only, and the other across rows. I've done this with code that's like:

```
x = (findgen(11,21) mod 11) - 5
y = transpose(findgen(21,11) mod 21) - 10
```

If I wanted to plot on x = [-5..5], y = [-10..10]. Then I would use a function like:

```
gaussian = exp((-1/10)*sqrt(x^2 + y^2))
```

to calculate my surface. However, this looks like it's going to become a pain in the ass really quickly for making coordinate arrays of higher dimensions. I want to do something like this:

```
cylinder = bytarr(10,10,10)
cylinder[where(logical_and((x^2 + y^2 lt 4), abs(z) lt 2)] = 255
```

to make a cylinder of 255s with radius 2 and height 4 centered on the z axis. Is there a better way to make the coordinate arrays x, y, and z that I would need for this function, assuming that the dimensions aren't always going to be a perfect cube? Should I be doing this way differently?

Thanks,
James Preiss