

---

Subject: visualizing data in 3-D

Posted by [patrick](#) on Tue, 07 Aug 2001 17:56:47 GMT

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

---

Folks-

I'm attempting to construct a 3-D visualization of oceanographic data. My data is in columnar ascii format. I need to interpolate corresponding data values at three different locations. For example, I have a density value of 21.9 kg/m<sup>3</sup> measured at three different locations and I want to visualize a density stratification surface interpolated between the three locations where the horizontal axes are space between the locations and the vertical axes are depth. What I have currently for the procedure is below. It is currently just a mangled set of lines which obviously won't work. the \*pstate variables correspond to 3 different data sets in a pointer array stacked on top of each other. So x1 is a depth column, and y1 is a density column, x2 & y2 are depth and density columns from a second location, ditto with x3/y3 at a third location. Could xplot3d or slicer3 work for this? Or should I use a combination of surfr and t3d as opposed to the current code?

```
x1 = reform((*pstate).profiledata(0,*,0))
y1 = reform((*pstate).profiledata(1,*,0))

x2 = reform((*pstate).profiledata(0,*,1))
y2 = reform((*pstate).profiledata(1,*,1))

x3 = reform((*pstate).profiledata(0,*,2))
y3 = reform((*pstate).profiledata(1,*,2))
print, 'working'
; interpolate 1st data onto 2nd data alt grid
y01 = interpol(y1, x1, x2)

; interpolate 3rd data onto 2nd data alt grid
y03 = interpol(y3, x3, x2)

; concatenate arrays

y = [transpose(y01), transpose(y2), transpose(y03)]
x = [0,1,2]

contour, y, x, x2
```

Thanks for any suggestions,

Patrick

---

---

Subject: Re: visualizing data in 3-D  
Posted by [Craig Markwardt](#) on Thu, 09 Aug 2001 21:10:19 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi Patrick--

I am not sure if you have gotten any responses to your question. I think your question is specialized enough that it may be difficult for us to understand it.

However, before you go delving into all the intricacies of "thinking in IDL", using HISTOGRAM and wrapping array indices around your brain, maybe you could take the low-tech approach.

I think what you want to do is simply locate the same density level in each measurement train. This might be best accomplished using VALUE\_LOCATE.

```
levels = [21.9, 25, 30, 35, 40]
i1 = value_locate(y1, levels)
i2 = value_locate(y2, levels)
i3 = value_locate(y3, levels)
```

This will find the index position of each of the levels, in each of the measurements. I1 are the indices of the 21.9, 25, ... values in the first set of measurements, etc. This is not exact actually, but it finds the nearest neighbor.

Then all you need to do is stitch them together.

```
for j = 0, n_elements(levels)-1 do $
  oplot, [x1(i1(j)), x2(i2(j)), x3(i3(j))], [y1(i1(j)), y2(i2(j)), y3(i3(j))]
```

Hopefully from there you can decide if you need more precision, faster, etc. You can also look at code like TABINV in the IDL Astronomy Library, which does tabular interpolation.

Craig

--

-----  
Craig B. Markwardt, Ph.D.      EMAIL: [craigmnet@cow.physics.wisc.edu](mailto:craigmnet@cow.physics.wisc.edu)  
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response  
-----

---

Subject: Re: visualizing data in 3-D  
Posted by [tbowers0](#) on Mon, 13 Aug 2001 17:41:50 GMT

Craig Markwardt <craigmnet@cow.physics.wisc.edu> wrote in message news:<onelql2b44.fsf@cow.physics.wisc.edu>...

> Hi Patrick--

>

> I am not sure if you have gotten any responses to your question. I  
> think your question is specialized enough that it may be difficult for  
> us to understand it.

[SNIP]

>

> Hopefully from there you can decide if you need more precision,  
> faster, etc. You can also look at code like TABINV in the IDL  
> Astronomy Library, which does tabular interpolation.

>

> Craig

I \*think\* he's just trying to get a true 3D field outta discrete x,y,z data, which requires either 1) 'stacked' 2D interpolations/gridding or 2) a 3D interpol./gridding.

Search IDL help on triangulate and trigridd. But, I'll warn you in advance, IDL's solution to 3D gridding is... how should i put this; poor? It'd be educational to you to do a search for posts on '3D interpolation', '3D gridding', 'trigridd' stuff like that and see the frustration evolve.

Luck,  
todd

---