
Subject: Re: 2D Plot in IDL with shading?

Posted by [penteado](#) on Mon, 21 Jun 2010 20:16:07 GMT

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

On Jun 21, 4:59 pm, Ally <ally.desh...@gmail.com> wrote:

> I'm sorry, now I'm even more confused. I have my 3 data sets, which
> are all arrays with one column and 5273670 rows. I'm trying to plot
> them in this 3D graph as (x,y,z) points. I understand why you're
> saying that I need to reform the response (hadn't thought about the
> connectivity before) but don't understand the fastest-varying
> dimension concept.

You have 5273670 points for response, which are the product of some number of azimuths (I will call it `n_az`), by some number of elevations (I will call it `n_el`). It was written to the file as a 1D array, which can be done in one of two usual ways:

1) elevations varying fastest: responses for all elevations are given for the first azimuth, followed by the responses for all elevations of the next azimuth, and so on. In this case, you would reform it to

```
response_2D=reform(response,n_el,n_az)
```

2) azimuths vary the fastest: responses for all azimuths are given for the first elevation, followed by the responses for all azimuths of the next elevation, and so on. In this case, the reform would be

```
response_2D=reform(response,n_az,n_el)
```

> I tried `reform(response, 2, 5273670)` and
> `reform(response, 5273670, 2)` and got an error that said 'new
> subscripts must not change the number elements in response' each
> time. What dimension am I trying to add to reform exactly?

`Reform()` only changes the dimensions of something, it preserves the number of elements. So the products of all dimensions you give to `reform()` must be the same as the number of elements of the array you are reforming. You were trying to make an array with two columns and 5273670 rows (and 5273670 columns and 2 rows), which is not the shape your response should have, nor is the same number of elements. The shape should be either `(n_az,n_el)`, or `(n_el,n_az)`, depending on how it was written to that file.

If you inspect the values of the azimuths and elevations you can find out which is the order. If the first few `(n_el)` elements of azimuth are the same, then then fastest varying dimension is the elevation, and response should be reformed to `(n_el,n_az)`.
