
Subject: Re: GRIDDATA woes

Posted by [ben.bighair](#) on Tue, 04 Mar 2008 22:46:46 GMT

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

On Mar 4, 4:14 pm, David Fanning <n...@dfanning.com> wrote:

> Kenneth P. Bowman writes:

>>> Compute the "interpolation coordinates" from the original grid

>>> j = VALUE_LOCATE(y_original, y)

>>> yj = j + (y - y_original[j])/(y_original[j+1] - y_original[j])

>

> To tell you the truth, I can't get this to work at all. :-(

>

> IDL> lat = [-87.5, 50, 25, 0, 30, 45, 64, 87.5]

> IDL> y = Scale_Vector(findgen(7), -87.5, 87.5)

> IDL> j = Value_Locate(lat, y)

> IDL> yj = j + (y - lat[j])/(lat[j+1] - lat[j])

> % Program caused arithmetic error: Floating illegal operand

> IDL> print, yj

> 0.000000 0.212121 0.424242 3.00000 3.97222 5.70175 -NaN

>

Hi,

I think it comes a little clearer for me to leave VALUE_LOCATE out of it. Instead simply normalize the output coords into the input coordinate space. Something like this...

```
ilon = Scale_Vector(findgen(10), 10.0, 70.0)
```

```
ilat = [-87.5, 50, 25, 0, 30, 45, 64, 87.5]
```

```
olon = Scale_Vector(findgen(12), 0.0, 90.0)
```

```
olat = Scale_Vector(findgen(7), -87.5, 87.5)
```

```
ix = (olon - min(ilon)) / (max(ilon) - min(ilon))
```

```
iy = (olat - min(ilat)) / (max(ilat) - min(ilat))
```

```
idim = [n_elements(ilon), n_elements(ilat)]
```

```
iZ = HANNING(idim[0], idim[1])
```

```
oZ = INTERPOLATE(iZ, ix, iy, /GRID, MISSING = !VALUES.F_NAN)
```

It doesn't address the question about assuming a flat dataspace when the data are really in spherical coordinates. You might have a good solution to transform the coordinates to a projected space then interpolate, but I really don't know for sure. Taking the data as delivered, it is not projected at all - it just comes with two of the three parts needed to specify spherical coords plus a data value. The radius is whatever you want (in this case I was hoping to be on

earth).

Ben
