Subject: Re: Interpolating a regular grid
Posted by David Fanning on Tue, 07 Mar 2006 15:10:39 GMT
View Forum Message <> Reply to Message

Clive Cook writes:

- > It seems that the hardest part in asking for help is knowing what to
- > ask. I am not sure if my subject header actually corresponds to my
- > question, however i though it was close enough.

>

- > I have a regularly gridded array of data (calibration data for an
- > instument) and i want to extract values from the array. The x-array is
- > pressure and the y-array is temperature. The data i require does not
- > match with any points in the array. For example the x-array runs from
- > 70 to 1050 mb in steps of 9 mb and the x-array goes from 180-330 K,
- > how do i interpolate the array so that i could look at values which
- > fall in between the regularly gridded data. So, clarifying this how
- > would i extract a value which corresponded to a pressure of 700.6 mb
- > and a temperature of 234.56 K.

You don't say what size your gridded array is, but I am going to assume it is a 110x150 array, so that we can get even values in pressure:

```
p = Scale_Vector(Indgen(110), 70, 1051); in units of 9 mb.
```

and temperature:

```
t = Scale \ Vector(Indgen(150), 180, 329); in units of 1 deg.
```

To interpolate your gridded array, you need to know the *indices* of pressure and temperature. One way to find the proper indices is like this:

```
xindex = Value_Locate(p, 700.6)
xindex = yindex + ((700.6 - p[yindex])/(p[yindex+1] - p[yindex]))
yindex = Value_Locate(t, 234.56)
yindex = yindex + ((234.56-t[yindex])/(t[yindex+1] - t[yindex]))
```

So, now, the value you are looking for is this:

interpValue = Interpolate(array, xindex, yindex)

Cheers.

David

--

Subject: Re: Interpolating a regular grid Posted by Mark Hadfield on Wed, 08 Mar 2006 01:34:42 GMT View Forum Message <> Reply to Message

David Fanning wrote:

- > To interpolate your gridded array, you need to know the *indices*
- > of pressure and temperature. One way to find the proper indices
- > is like this:

> xindex = Value_Locate(p, 700.6)

> xindex = value_Locate(p, 700.6) > xindex = yindex + ((700.6 - p[yindex])/(p[yindex+1] - p[yindex]))

> xindex = yindex + ((700.6 - p[yindex])/(p[yindex+1] - p[yindex])
>

> yindex = Value_Locate(t, 234.56)

yindex = yindex + ((234.56- t[yindex])/(t[yindex+1] - t[yindex]))

>

> So, now, the value you are looking for is this:

> interpValue = Interpolate(array, xindex, yindex)

For what it's worth, the Motley library at

http://www.dfanning.com/hadfield/idl/README.html

has a function called MGH_LOCATE which will calculated one or more "fractional index" values in the way you have shown. It also has cousins MGH_LOCATE2 & MGH_LOCATE2A that do the same thing for 2-D curvilinear arrays.

Note to self: I must update that library.

--

Mark Hadfield "Kei puwaha te tai nei, Hoea tahi tatou" m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)