Subject: Re: matching 2 grids
Posted by Mark Hadfield on Thu, 09 Mar 2006 20:54:22 GMT

View Forum Message <> Reply to Message

David Fanning wrote:
> I don't think this is the right approach, but I don't
> have the time to look into it at the moment. Mark is
> probably scratching his ass and getting his coffee together
> right now. He'll be with you shortly. :-)

Ta da!

Thanks for working out that MGH_MOTLEY is a script, David. To use the
Motley library you *must* add its directory to your path and you *must*
run "@MGH_MOTLEY" first.

It's MGH_EXAMPLE_LOCATE. It has 1D and 2D examples, selected by a
positional parameter called "option".

The term "index space" is one I made up (or plagiarised). Think of a 1-D
IDL array with n elements. The elements are indexed 0 to n-1. Now give
this array some values, monotonically increasing or decreasing (usually
the former) so that we can interpret the array as a 1-D grid in space.
We have a value that represents a position somewhere in the range
covered by this grid and we want to know where it is relative to the
grid. Let the grid array be x and the position we're trying to locate be
x. Let's say we find an index i such that $x[i] <= xp <= x[i+1]$. In fact,
let's go further and say that xp is exactly half-way between $x[i]$ and
$x[i+1]$. Then I would say that xp is at position i+0.5 in the index space
of grid x.

For a 2D curvilinear grid (defined by a pair of 2D arrays) the index
space is 2D. The concept is very similar but it's a bit harder to explain.

One of the reasons "index space" is a handy concept is that it is used
by IDL's INTERPOLATE function.

To give an example of the above in terms of Motley functions:

```
IDL> x = mgh_range(100, 200, STRIDE=20)
IDL> print, x
      100      120      140      160      180      200
IDL> xp = 133.
IDL> print, mgh_locate(x, XOUT=xp)
    1.65000
```

--
Mark Hadfield   "Kei puwaha te tai nei, Hoea tahi tatou"

m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)