Subject: Re: array
Posted by Jeremy Bailin on Mon, 09 Mar 2015 19:47:04 GMT
View Forum Message <> Reply to Message

> Since you're the Value_locate guy, can it be used with 2d arrays? If so, how is monotonic defined in these cases? By using 1d indexes for a 2d array?

Yes, exactly -- it needs to be monotonic with respect to 1D indices.

> And in case of a 2d array, when does one end up with a monotonic 2d array (however this is defined)?

For example:

q = indgen(N1, N2)

gives you an array that is monotonic as far as Value_Locate is concerned. Also, I will sometimes create a 2D array that is monotonic by stacking together individually-monotonic 1D arrays with offsets in the second dimension for the express purpose of using Value_Locate on it. For example, if W is an N1 x N2 array where each W[*,i] is sorted, then you can create a monotonic 2D array is follows:

```
; in order to make the array monotonic, we need to add an offset to
; each row that will ensure that the minimum value for every subsequent
; row is pushed to being greater than the maximum value of each previous
; row. The following example will technically only work for non-negative W
; and for either integers or not-too-large floats, but can be generalized.
offset = max(W)+1
Wdimen = size(W, /dimen)
; create an N1 x N2 array that adds an appropriate increment to each row
W2 = W + rebin(offset * lindgen(1, Wdimen[1]), Wdimen, /sample)
```

-Jeremy.