Subject: Re: Subscripting multidimensional arrays
Posted by Chris Lee on Fri, 12 Dec 2003 21:15:33 GMT
View Forum Message <> Reply to Message

JD Smith wrote:
> On Fri, 12 Dec 2003 07:55:16 -0700, Christopher Lee wrote:
<snip>
>
> PRODUCT works nicely for this:
>
> function linear_indices,array,vec_indices
> s=size(array,/DIMENSIONS)
> nd=n_elements(s)
> if nd eq 1 then return,s[0]

;you mean "return ,vec_indices" , yes?

> return,long(total([1.,product(s[0:nd-1],/CUMULATIVE)]*vec_in dices))
> end

PRODUCT doesn't seem to exist on my IDL installations (5.3->5.6 inclusive), is it an IDL 6 thing? I've written my own obviously, but not with CUMULATIVE. Hopefully the IDL 6 licence will work on Monday.

>
> to go the other direction, IDL6 offers ARRAY_INDICES. Or you can always
> just resort to:
>
> a[vec[0],vec[1],vec[2]]
>

Ahem, oops. Apparently I used a cluster bomb to open a can of beans...a switch case statement would take care of the dimensions upto the IDL limit of 8 dimensions, or using IDL's ability to ignore trailing dimensions if they're 0 hack to use 8 dimensions every time.

> A take home problem would be to modify this such that NxM input vectors,
> where N is the number of dimensions of "array", will return a vector of
> length M containing all the 1-D indices. Hints: REBIN/REFORM and the
> "dimension" argument to TOTAL.
>

I got that exercise too, but I thought I'd leave some fun for other people :) plus I always get into a "but what if I wanted the input

vector to be MxN" and "what if not enough arguments are supplied, or they're out of bounds" and I remember I have work to do, sometimes.

> JD

Chris.
