
Subject: Re: Another simple one

Posted by [MarioIncandenza](#) on Sun, 29 Jul 2007 19:39:23 GMT

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

On Jul 28, 12:58 am, snudge42 <snudg...@gmail.com> wrote:

> How do I truncate a multidimensional array at a different place for
> each dimension i.e. I start with 3x6 array of values A=

>
> 0 0 0
> 0 0 0
> 0 0 0
> 0 0 0
> 0 0 0
> 0 0 0
> 0 0 0
>

> and want to truncate each dimension at a fixed value stored in an
> array B, where the values are (2,3,4) for example so that I get:

>
> 0 0 0
> 0 0 0
> 0 0 0
> 0 0
> 0
>

By 'truncate', you mean either "perform calculations on only part of array A" or "write output of only part of array A". For the second case, the I/O penalty is far greater than the for-loop penalty, just use a loop. For the first case, consider this:

;NOTE: this example truncates along rows, you'll need to TRANSPOSE to do columns

btrunc=a*0; initialize truncation helper array

for i=0,l_n_elements(b)-1 do btrunc[0]=(lindgen(b[i]+1))+1 gt 0; anyone care to try getting rid of this loop?

print,trunc

> 1 1 1 0 0
> 1 1 1 1 0
> 1 1 1 1 1

; from here, you can manipulate A in several ways:

; such as setting truncated components to NaN:

ftrunc=where(trunc eq 0)

anew=a & anew[ftrunc]=!values.NaN

; setting truncated components to 0:

anew= a * btrunc;

Hope this helps,

--Edward H.
