Subject: Re: Addressing 3D arrays different from 2D arrays?
Posted by Foldy Lajos on Wed, 07 Nov 2007 13:01:46 GMT
View Forum Message <> Reply to Message

On Wed, 7 Nov 2007, Spon wrote:

>
> Spon wrote:
>> I can get rid of it, but I'm not sure why you're getting a square (as
>> opposed to just junk):
>>
> Ok, now *I'm* confused:
>
> *** Code
>
> pro threedtest
> xidx = [5,4,5,6,3,4,5,6,7,4,5,6,5]
> yidx = [3,4,4,4,5,5,5,5,5,6,6,6,7]
>
> print, 'Fixed version.'
> zidx = REPLICATE (0, N_ELEMENTS (xidx))
> test2d = bytarr(10,10)
> test3d = bytarr(10,10,10)
> test2d[xidx,yidx] = 1
> test3d[zidx,xidx,yidx] = 1
> print,test2d,total(test2d)
> print,reform(test3d[0,*,*]),total(test3d)
> print, ''
> print, 'Original version.'
> test2d = bytarr(10,10)
> test3d = bytarr(10,10,10)
> test2d[xidx,yidx] = 1
> test3d[0,xidx,yidx] = 1
> print,test2d,total(test2d)
> print,reform(test3d[0,*,*]),total(test3d)
> print, ''
> print, 'Concatenated version.'
> subscripts = [0, xidx, yidx]
> test2d = bytarr(10,10)
> test3d = bytarr(10,10,10)
> test2d[xidx,yidx] = 1
> test3d[subscripts] = 1
> print,test2d,total(test2d)
> print,reform(test3d[0,*,*]),total(test3d)
>
> return
> end
>

> *** End of Code
>
> ...
> IDL> Concatenated version.
> 0 0 0 0 0 0 0 0 0 0
> 0 0 0 0 0 0 0 0 0 0
> 0 0 0 0 0 0 0 0 0 0
> 0 0 0 0 0 1 0 0 0 0
> 0 0 0 0 1 1 1 0 0 0
> 0 0 0 1 1 1 1 1 0 0
> 0 0 0 0 1 1 1 0 0 0
> 0 0 0 0 0 1 0 0 0 0
> 0 0 0 0 0 0 0 0 0 0
> 0 0 0 0 0 0 0 0 0 0
>   13.0000
> 1 0 0 0 0 0 0 0 0 0
> 0 0 0 0 0 0 0 0 0 0
> 0 0 0 0 0 0 0 0 0 0
> 0 0 0 0 0 0 0 0 0 0
> 0 0 0 0 0 0 0 0 0 0
> 0 0 0 0 0 0 0 0 0 0
> 0 0 0 0 0 0 0 0 0 0
> 0 0 0 0 0 0 0 0 0 0
> 0 0 0 0 0 0 0 0 0 0
> 0 0 0 0 0 0 0 0 0 0
>   6.00000
>
> ***End of Output
>
> Why does OP get a nice square whereas I just get a solitary 1 in the
> corner? :-(
> What is concatenating before defining the subscripting causing IDL to
> do differently?
>
> Just curious,
> Chris
>
>

subscripts is an array, with elements 0,5,4,5,6,3,4,5,6,7,4,5,6,5,
3,4,4,4,5,5,5,5,5,6,6,6,7 (= six different values, 0 and 3-7).
test3d[subscripts]=1 will set elements test3d[0] and test3d[3:7]
(= test3d[0,0,0] and test3d[3:7, 0,0]). The test3d[0,*,*] slice
contains only one element set.

regards,
lajos
