Subject: Addressing 3D arrays different from 2D arrays?
Posted by Jaron Kurk on Tue, 06 Nov 2007 17:13:06 GMT
View Forum Message <> Reply to Message

Dear readers,

Apologies if this question has long been answered, but I could not find anything on it.

Is there some fundamental difference in addressing 3D arrays and 2D arrays? In IDL 6.3 (and GDL), the following code fills a 2D array with a circle of 1's but a slice of a 3D array with a square of 1's, while I would expect just the same area filled with 1's as for the 2D case. Note that the use of reform() does not cause the difference, I have checked that.

```
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]
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)
```

If anybody could enlighten me, I would appreciate it!

Jaron Kurk.

Subject: Re: Addressing 3D arrays different from 2D arrays?
Posted by Foldy Lajos on Tue, 06 Nov 2007 17:27:59 GMT
View Forum Message <> Reply to Message

On Tue, 6 Nov 2007, Jaron Kurk wrote:

> Dear readers,
>
> Apologies if this question has long been answered, but I could not
> find anything on it.
>
> Is there some fundamental difference in addressing 3D arrays and 2D
> arrays? In IDL 6.3 (and GDL), the following code fills a 2D array with
> a circle of 1's but a slice of a 3D array with a square of 1's, while
> I would expect just the same area filled with 1's as for the 2D case.
> Note that the use of reform() does not cause the difference, I have
> checked that.

>
> 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]
> test2d = bytarr(10,10)
> test3d = bytarr(10,10,10)
> test2d[xidx,yidx] = 1

array subscripts

> test3d[0,xidx,yidx] = 1

mixed scalar and array subscripts. Different rules :-)

try: test3d[lonarr(13),xidx,yidx] = 1

regards,
lajos

Subject: Re: Addressing 3D arrays different from 2D arrays?
Posted by Spon on Tue, 06 Nov 2007 17:30:05 GMT
View Forum Message <> Reply to Message

On Nov 6, 5:13 pm, Jaron Kurk <jaron.k...@googlemail.com> wrote:
> Dear readers,
>
> Apologies if this question has long been answered, but I could not
> find anything on it.
>
> Is there some fundamental difference in addressing 3D arrays and 2D
> arrays? In IDL 6.3 (and GDL), the following code fills a 2D array with
> a circle of 1's but a slice of a 3D array with a square of 1's, while
> I would expect just the same area filled with 1's as for the 2D case.
> Note that the use of reform() does not cause the difference, I have
> checked that.
>
> 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]
> 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)
>
> If anybody could enlighten me, I would appreciate it!
>

> Jaron Kurk.

I can get rid of it, but I'm not sure why you're getting a square (as opposed to just junk):

```
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]
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)
```

Your test3d array wasn't shifting your first two arrays by a whole dimension, just one element.

Chris

Subject: Re: Addressing 3D arrays different from 2D arrays?
Posted by Jean H. on Tue, 06 Nov 2007 17:34:04 GMT
View Forum Message <> Reply to Message

> xidx=[5,4,5,6,3,4,5,6,7,4,5,6,5]

> test3d[0,xidx,yidx] = 1

Jaron, you must reference EVERY pixels in 3D, not just one..
You can try something like:

```
n_points = n_elements(xidx)
zIdx = bytarr(n_points)
test3d[zIdx,xidx,yidx] = 1      --> though having X,Y,Z instead of Z,X,Y would be easier to manipulate I guess
```

If I remember well there is an article on David Fanning's site, likely written by JD Smith.

Jean

Subject: Re: Addressing 3D arrays different from 2D arrays?
Posted by Jaron Kurk on Wed, 07 Nov 2007 09:14:37 GMT
View Forum Message <> Reply to Message

Thanks for the rapid response. Indeed, mixing scalar and array subscripts was the problem. Sometimes IDL is just to flexible for my mind...

Jaron

Subject: Re: Addressing 3D arrays different from 2D arrays?
Posted by Spon on Wed, 07 Nov 2007 11:15:05 GMT
View Forum Message <> Reply to Message

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

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

Subject: Re: Addressing 3D arrays different from 2D arrays?
Posted by Spon on Wed, 07 Nov 2007 13:09:22 GMT
View Forum Message <> Reply to Message

> 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

Ah, I get it. So in the original, the overlap of 4th to 8th elements
([3:7]) of both dimensions were set to 1, hence the square! Brilliant.

thanks a lot,
Chris
