
Subject: Re: Problem array subscripting

Posted by [Liam Gumley](#) on Thu, 19 Aug 1999 07:00:00 GMT

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

GRI wrote:

> I think I did not explain myself very well the first time... I'll try again
>
> What I want to do is to be able to get the evolution along time of some
> pixels of a given image series. For instance, I want to see what values
> pixel at the origin (x=0, y=0) takes as time goes by.
>
> Suppose I define:
>
> array= INDGEN(2,2,5) ; dimensions (x,y,t)
>
> I want to extract vectors along the time dimensions, i.e. of length 5,
> at any given combination of x and y.
>
> For instance, I may need the vector at x=0, y=0 (from t=0 to t=4), and also
> another vector at x=1, y=1 (I obtain the x and y coordinates from a masking
> image)
>
> This vectors would be, in this case [0,4,8,12,16] and [3,7,11,15,19].
>
> When using a combination of two vectors to index ARRAY like
>
> array[[0,1],[0,1],*]
>
> what IDL does is combine each value in the first vector with all
> elements in the second vector ([0,0], [0,1], [1,0], [1,1]), and all
> values in the third dimension, as indicated by the asterisk, and so
> I get (in this case) the original array.
>
> In a general case this would be ok if I wanted to profile the array along
> the time dimension at a regular grid ([0,0], [10,0], [20,0], [0,10],
> [10,10], [20,10] ...)
>
> The only method I have proved it works like I want is the 'ugly' one
> proposed by Craig Markwardt. I have also seen it is really faster than
> using a for loop when the number of vectors to extract is high. Looks
> like the use of the /OVERWRITE keyword makes it work very fast, since as
> IDL's help says it only changes the data descriptor, not the data itself.
>
> The other two methods give me the same results as indexing the time series
> directly with [[0,1],[0,1],*].
>
> I haven't been able to download Martin Schultz's arrex function (might be a
> problem with my browser, the tools library is there, but instead of

> downloading
> it my Netscape opens it like if it were text, which is not).

I've been scratching my head over this one for a while, and I don't see how this particular problem can be solved using traditional IDL array indexing syntax, even with added degenerate dimensions. I don't think arrex will solve it either (I'll be happy to be proven wrong).

Say you have the following arrays defined:

```
a = indgen(2, 2, 5)
x = [0, 1]
y = [0, 1]
z = [0, 1, 2, 3, 4]
```

When you tell IDL to extract a[x, y, z], it extracts the following elements:

```
0, 0, 0
1, 0, 0
0, 1, 0
1, 1, 0
0, 0, 1
1, 0, 1
0, 1, 1
1, 1, 1
0, 0, 2
1, 0, 2
0, 1, 2
1, 1, 2
0, 0, 3
1, 0, 3
0, 1, 3
1, 1, 3
0, 0, 4
1, 0, 4
0, 1, 4
1, 1, 4
```

You can see that it loops through the supplied index arrays an element at a time, with the leftmost index array in the innermost loop. The elements you really wanted were

```
0, 0, 0
1, 1, 0
0, 0, 1
1, 1, 1
0, 0, 2
```


1, 1, 2
0, 0, 3
1, 1, 3
0, 0, 4
1, 1, 4

I don't believe you can extract this sub-array using normal array indexing. It requires creative array re-arrangement, as Craig demonstrated rather elegantly IMHO.

Array indexing is always good for a long thread in this newsgroup.....

Cheers,
Liam.

--

Liam E. Gumley
Space Science and Engineering Center, UW-Madison
<http://cimss.ssec.wisc.edu/~gumley>
