
Subject: Variable stride in array indices

Posted by [bowman](#) on Fri, 07 May 1999 07:00:00 GMT

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

We haven't had a good argument about IDL syntax lately, so I thought I would ask why IDL does not allow variable stride in array indices. For example, if

```
zz = z[0:5:2]
```

then zz would contain [z[0], z[2], z[4]].

I realize that this kind of thing can be done with array subscript lists such as z[[0, 2, 4]], but this can be awkward for multidimensional arrays. It is also necessary to *store* the lists of subscripts, and I believe that implementation through indirect subscripts must be slower than implementing strides through direct iteration (i.e., internal compiled DO loops).

Any comments?

Ken Bowman

Subject: Re: Variable stride in array indices

Posted by [DBorland](#) on Tue, 18 May 1999 07:00:00 GMT

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

In article <bowman-1805990954070001@bowman.tamu.edu>, bowman@null.edu (Kenneth P. Bowman) wrote:

> I had a chance to look at my code to see if Liam's suggestion would work,
> and I realized that it won't, because I need to use the variable stride on
> the **left** side of the equal sign, i.e.

```
>  
> IDL> a = LINDGEN(6,6)  
> IDL> print, a  
>      0      1      2      3      4      5  
>      6      7      8      9     10     11  
>     12     13     14     15     16     17  
>     18     19     20     21     22     23  
>     24     25     26     27     28     29  
>     30     31     32     33     34     35  
> IDL> print, (a[*,2*LINDGEN(3)])[2*LINDGEN(3),*]  
>      0      2      4  
>     12     14     16  
>     24     26     28
```

What you need to do it then sent the results here as the index for the array. ie:

```
IDL> a[(a[*],2*LINDGEN(3))][2*LINDGEN(3),*] = -1
```

When you do this, the values from above are set to -1

```
IDL> print,a
```

-1	1	-1	3	-1	5
6	7	8	9	10	11
-1	13	-1	15	-1	17
18	19	20	21	22	23
-1	25	-1	27	-1	29
30	31	32	33	34	35

David Borland
Software Engineer
Electrical Geodesics, Inc.

Subject: Re: Variable stride in array indices
Posted by [bowman](#) on Tue, 18 May 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <3732EFA0.1F6C49C0@ssec.wisc.edu>, Liam Gumley
<Liam.Gumley@ssec.wisc.edu> wrote:

> Or I can select the columns and rows at the same time:

>

```
> IDL> print,(a[x,*])[*,x]
```

```
> 0 2 4 6 8
> 20 22 24 26 28
> 40 42 44 46 48
> 60 62 64 66 68
> 80 82 84 86 88
```

I had a chance to look at my code to see if Liam's suggestion would work, and I realized that it won't, because I need to use the variable stride on the *left* side of the equal sign, i.e.

```
IDL> a = LINDGEN(6,6)
```

```
IDL> print, a
```

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17

```

18      19      20      21      22      23
24      25      26      27      28      29
30      31      32      33      34      35
IDL> print, (a[* ,2*LINDGEN(3)])[2*LINDGEN(3),*]
0        2        4
12       14       16
24       26       28

```

works fine *extracting* data, but

```

IDL> (a[* ,2*LINDGEN(3)])[2*LINDGEN(3),*] = -1
% Temporary variables are still checked out - cleaning up...
IDL> PRINT, A

```

```

0        1        2        3        4        5
6        7        8        9       10       11
12       13       14       15       16       17
18       19       20       21       22       23
24       25       26       27       28       29
30       31       32       33       34       35

```

fails.

So it is possible to extract values (gather) but not to insert values (scatter) with a variable stride using this method.

I repeat my call for a simple stride syntax in IDL

```
a[0:5:2,0:5:2] = -1
```

Ken

--

Dr. Kenneth P. Bowman, Professor	409-862-4060
Department of Meteorology	409-862-4466 fax
Texas A&M University	bowmanATcsrp.tamu.edu
College Station, TX 77843-3150	Replace AT with @

Subject: Re: Variable stride in array indices
 Posted by [bowman](#) on Tue, 18 May 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <ySg03.87161\$A6.43176220@news1.teleport.com>, "DBorland" <dborland@egi.com> wrote:

```

> IDL> a[(a[* ,2*LINDGEN(3)])[2*LINDGEN(3),*]] = -1
>
> When you do this, the values from above are set to -1

```

```
> IDL> print,a
>      -1      1      -1      3      -1      5
>      6      7      8      9     10     11
>     -1     13     -1     15     -1     17
>     18     19     20     21     22     23
>     -1     25     -1     27     -1     29
>     30     31     32     33     34     35
```

This only works because the original array was created with LINDGEN. It won't work in the general case.

I still like

```
a[0:*:2,0:*:2] = -1
```

for aesthetic reasons alone.

Ken
