Subject: Re: Variable stride in array indices
Posted by bowman on Fri, 07 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:

> 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

Thanks for the suggestion. I had to stare at this for a while to
understand it, but I do see how it works.

I maintain, however, that

  a[0:8:2, 0:9:3]

is simpler, clearer, similar to Fortran 90, and much more amenable to
optimization than

  (a[2*FINDGEN(5),*])[*,3*FINDGEN(4)]

In fact, Fortran 90 even allows negative strides. Also, one is never sure
what is going on under the hood in IDL ... i.e., how much array copying
and indirect indexing is happending ... so performance on large arrays may
not be great.

So it seems that IDL (*the array language*) has finally been surpassed by
Fortran! (That's a clumsy attempt to goad RSI into adding this syntax to
the language.)

>> Just for sake of argument, how can this be extended to
>> a five-dimensional parabolic rhomboid?
>
> a = lindgen(10,10,10,10,10)
> x = lindgen(5)*2
> help, ((((a[x,*,*,*,*])[*,x,*,*,*])[*,*,x,*,*])[*,*,*,x,*])[*,*,*, *,x]
> <Expression> LONG = Array[5, 5, 5, 5, 5]

I showed this to a colleague, who's response was, "He's a madman!".

:-)

Ken

Subject: Re: Variable stride in array indices
Posted by davidf on Fri, 07 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Liam Gumley (Liam.Gumley@ssec.wisc.edu) writes:

> a = lindgen(10,10,10,10,10)
> x = lindgen(5)*2
> help, ((((a[x,*,*,*,*])[*,x,*,*,*])[*,*,x,*,*])[*,*,*,x,*])[*,*,*, *,x]
> <Expression> LONG = Array[5, 5, 5, 5, 5]

Thank you. This solves a long-standing problem that I have been
having with IDL!!!!!!! :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Variable stride in array indices
Posted by Liam Gumley on Fri, 07 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Liam Gumley (Liam.Gumley@ssec.wisc.edu) writes:
>
>> Here's what we resolved:
>>
>> 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
>
> Just for sake of argument, how can this be extended to
> a five-dimensional parabolic rhomboid?

```
a = lindgen(10,10,10,10,10)
x = lindgen(5)*2
help,  ((((a[x,*,*,*,*])[*,x,*,*,*])[*,*,x,*,*])[*,*,*,x,*])[*,*,*, *,x]
<Expression>    LONG      = Array[5, 5, 5, 5, 5]
```

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

Subject: Re: Variable stride in array indices
Posted by davidf on Fri, 07 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Liam Gumley (Liam.Gumley@ssec.wisc.edu) writes:

```
>  Here's what we resolved:
>
>  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
```

Just for sake of argument, how can this be extended to
a five-dimensional parabolic rhomboid?

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Variable stride in array indices
Posted by Liam Gumley on Fri, 07 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Kenneth P. Bowman wrote:

> 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).

In fact we had a good argument recently here at SSEC about indexing 2D arrays with variable stride. The argument was this: If I have an array a, and an index array x, why doesn't a[x,x] extract a 2D subsampled array? (it returns a 1D array).

Here's what we resolved:

Say I want to extract every 2nd row and column of an array:

```
IDL> a=indgen(10,10)
```

First, I create the index array:

```
IDL> x=indgen(5)*2
IDL> print,x
      0       2       4       6       8
```

I can select the columns:

```
IDL> print,a[x,*]
      0       2       4       6       8
     10      12      14      16      18
     20      22      24      26      28
     30      32      34      36      38
     40      42      44      46      48
     50      52      54      56      58
     60      62      64      66      68
     70      72      74      76      78
     80      82      84      86      88
     90      92      94      96      98
```

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
```

This is equivalent to the following

```
IDL> a=a[x,*]
IDL> print,a
       0       2       4       6       8
      10      12      14      16      18
      20      22      24      26      28
      30      32      34      36      38
      40      42      44      46      48
      50      52      54      56      58
      60      62      64      66      68
      70      72      74      76      78
      80      82      84      86      88
      90      92      94      96      98
IDL> a=a[*,x]
IDL> print,a
       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
```

Cheers,
Liam.

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