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