Subject: Matrix expansion performance
Posted by Ricardo Bugalho on Mon, 28 Mar 2005 12:37:43 GMT
View Forum Message <> Reply to Message

Hi,
I have a matrix A (m,n) is and I want to create a matrix B(m,n,p) such that each B(*,*,i) slice equals A. p is very large and n is usually smaller than m so I have:

```
B=bytArr(m,n,p)
C=byteArr(p) + 1
FOR i = 0, n-1 DO B[*,i,*] = REFORM(A[*,i]) # p
```

Quite fast, but not enough for my needs. Any one has better sugestions?

Still stuck in IDL 5.4, by the way.

Thanks,
Ricardo

Subject: Re: Matrix expansion performance
Posted by JD Smith on Wed, 30 Mar 2005 16:51:02 GMT
View Forum Message <> Reply to Message

On Wed, 30 Mar 2005 16:21:02 +0100, Christopher Lee wrote:

> In article
> <Pine.LNX.4.44.0503301010060.7505-100000@localhost.localdomain>, "Timm
> Weitkamp" <dont.try@this.address> wrote:
>
>
>> On 29.03.05 at 10:55 +0100, Ricardo Bugalho wrote:
>>> I think I didn't make clear the ranges of m,n and p. In the problem I
>>> have at hand, m is always 8, n is usually 5 (min 1, max 16) and p is in
>>> the range of 10,000 to 100,000. Looping over p is a BadThing(tm) due to
>>> IDL's high interpretation overhead.
>> The method that Chris Lee suggested does not use loops. But I think
>> there is no need for any call to REFORM. And the dimension arguments to
>> REBIN must be scalars in IDL 5.4. A simple
>>
>> b = rebin(a, m, n, p, /sample)
>> should therefore work (and, hopefully, be fast enough for your
>> purposes). Timm
>>
>
> My first reaction was "when did that happen?", I tried it without the
> reform, and it works...except

>
> IDL> help, rebin(fltarr(4,5),[7,4,5,6])
> % REBIN: Result dimensions must be integer factor of original dimensions
>
> doesn't work (6.1.1 Linux), but the reform version does
>
> IDL> help, rebin(reform(fltarr(4,5),[1,4,5,1]),[7,4,5,6])
> <Expression> FLOAT = Array[7, 4, 5, 6]
>
> So my world-view isn't completely shattered :)

IDL can always add trailing shallow dimensions (final dimensions of depth 1) automatically for you. Thus dimensions [X,Y] can be turned into [X,Y,Z,P,D,Q,...] without resorting to REFORM. Adding dimensions before the last one does require REFORM. From the tutorial:

> We can also add new trailing dimensions with rebin, as long as all dimensions before it follow this rule. E.g. [2,3] could become [2,3,5] without trouble, but not [3,2,5]. (You can think of this by imagining a vector/array has implicity as many trailing shallow dimensions as you want (see below). IDL often truncates these, but also auto-creates them as necessary, as in this case!)

Adding it yourself doesn't hurt, but it isn't necessary either.

JD
