Subject: Re: For loops vs. matrix operations

Posted by Wonko[3] on Wed, 17 Dec 2003 23:24:00 GMT

View Forum Message <> Reply to Message

greenberg@ucdavis.edu (Jonathan Greenberg) wrote:

> I know some matrix programs perform better if you do straigh matrix math
> vs. a for-next loop -- is idl this way? E.g. is:

```
>  array=intarr(10000)
>  for i=0,(10000-1) do begin
>     array[i]=array[i]+1
>  endfor
```

> MUCH slower than:

```
>  array=intarr(10000)
>  array=array+1
```

Not only MUCH, but **MUCH** slower, at least.

Even faster is this: aray = temporary(array) + 1
This avoids duplicating the a variable first, saving time and memory.

But why don't you try it yourself?

```
IDL> a = intarr( 10000000L )
IDL> t=systime(1) & for i = 0L, 10000000-1 do a[i]=a[i]+1 & print, systime(1)-t
      4.2659999
IDL> t=systime(1) & a=a+1 & print, systime(1)-t
      0.18999994
IDL> t=systime(1) & a=temporary(a)+1 & print, systime(1)-t
      0.040000081
```

> I'm trying to figure out how much time I should be using rewriting some
> code to optimize the algorithm, which is why I'm asking (the code is more
> complex than above, obviously, but I did notice I could "matricize" some
> of the code in places)...

Matricyzation should always save time, especially if you have small
inner loops. I also think this makes the code more readable and
universal.

Alex

--
Alex Schuster   Wonko@wonkology.org   PGP Key available
alex@pet.mpin-koeln.mpg.de