Subject: Re: about replicate_inplace
Posted by Mark Hadfield on Tue, 22 Jul 2003 21:46:05 GMT
View Forum Message <> Reply to Message

Xiaoying Jin wrote:
> I am really puzzled by the IDL usage of memory.

Me too.

> Here is an example:
>
> T = systime(1)
> b = bytarr(nreg, nreg)
> print, systime(1) - T, ' seconds.'
>
> T = systime(1)
> replicate_inplace, b, 0
> print, systime(1) - T, ' seconds.'
>
> In the example, the first part is to create an array with values 0.
> The second part is to assign each element in the array the value 0.
> The running time for the first part is 0.12sec, while the running time
> for the second part is 0.37sec.

Interesting. On my system (Pentium 4 2.67 GHz with 1 GB RAM) I get similar results, though the difference is not so marked. eg with nreg = 10000, my times are 0.22 s & 0.34 s respectively. (I have to run these tests a couple of times to get stable results.)

However creating the array with REPLICATE takes 0.37 s, ie. significantly longer than BYTARR and slightly less than REPLICATE_INPLACE.

> My question is: since the first part needs to allocate the memory and
> assign values to elements, it should take longer time than the second
> part. In IDL help, it says: "REPLICATE_INPLACE can be faster and use
> less memory than the IDL function REPLICATE or the IDL array notation
> for large arrays that already exist. " However, why the first part
> takes less time?

Well, based on my experience, REPLICATE_INPLACE *does* take (slightly) less time than REPLICATE.

> Any suggestion?

For most purposes, just use the approach that gives the most straightforward, readable code. Where performance or memory usage becomes an issue, test different approaches.

--
Mark Hadfield "Ka puwaha te tai nei, Hoea tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)