Subject: Re: temporary() pitfall
Posted by Jaco van Gorkom on Tue, 19 Dec 2000 17:42:11 GMT
View Forum Message <> Reply to Message

Thanks, Wayne. I gained some new insights here.

> The memory that you save with TEMPORARY() comes at the cost of losing
> the original array contents. If you are worried about losing the
> result of a long computation because of hitting a memory limit, then I
> would SAVE the array to disk first. (I find that programs that use a
> lot of TEMPORARY calls are also difficult to debug.)

I agree that losing the original contents is the price that I was willing to pay. I guess I was just hoping that someone here would come up with another secret and magical keyword to ROUTINE_NAMES(), to recover that which seems lost forever. Always keep hoping for a miracle...

SAVEing to disk is of course the best option, but I did not expect TRANSPOSE to need a lot of memory. Especially not since it is mentioned as an example of smart memory use in the Online Help (under "Virtual Memory"). Now that I tried your examples, I realize that I should not have used A = TRANSPOSE(TEMPORARY(A)), but something like A = TRANSPOSE(A, /INPLACE), using an (IDL5.5???) /INPLACE or /OVERWRITE or /NOCOPY keyword to prohibit TRANSPOSE from copying the whole array. Bad luck for now.

Thanks for pointing out the behaviour of the MAX value in HELP,/MEMORY ; that seems to be a good method for testing the memory expense of certain operations. (I have not installed IDL5.4 yet, because I really do not want to miss the colored listings in idlde. And yes, I have *tried* to install xemacs for IDLWAVE. But let's not get started on installing under Unix again...)

Jaco

Jaco van Gorkom gorkom@rijnh.nl
FOM-Instituut voor Plasmafysica Rijnhuizen