Subject: Re: temporary() pitfall
Posted by landsman on Mon, 18 Dec 2000 23:00:23 GMT
View Forum Message <> Reply to Message

In article <3A3E0D3C.23741E8E@fz-juelich.de>,
j.c.van.gorkom@fz-juelich.de wrote:
> I use the temporary(MyArray) function to conserve memory when doing
> operations on large arrays. This works fine, except if there is still
> not enough memory to do the operation. Then IDL stops execution, tells
> me there is not enough memory, and MyArray is undefined. IDL follows
the
> documentation here, but I lose my array contents!
>
> Does anyone know of a way
> - to test in advance whether the operation is going to fail, or
> - to recover the contents of my original array?
>
> The operation I do varies, today I was trying to do
> MyArray = transpose(temporary(MyArray))

Jaco,

Well, I think this is a case where you can't get something for nothing.
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.)

If you have IDL V5.4 and are transposing prior to a matrix
multiplication, you might look at the /ATRANSPOSE and /BTRANSPOSE
keywords to MATRIX_MULTIPLY which do an implicit transpose to save
memory. (The transpose is done simultaneously with the matrix
mutliplication.)

Finally, I'd assert (without complete confidence) that there is no sense
in using the TEMPORARY function unless you have at least two variables
or constants on the right hand side of the "=" sign. One way to see
this is to use the /HIGHWATER keyword to the MEMORY function introduced
in V5.4. This returns the maximum amount of dynamic memory used since
the last time the MEMORY function or HELP,/MEMORY was called. (Users
without 5.4 can look at the MAX value in a HELP,/MEMORY call.)

First, look at a case where a call to TEMPORARY() really is useful.

IDL>print,!VERSION
sparc sunos unix 5.4 Sep 25 2000 64 64}

```
IDL> a = lonarr(2048,2048) & tmp =memory() &  a = a+1  &
print,memory(/high)
   50834756
IDL> a = lonarr(2048,2048) & tmp =memory() & a = temporary(a)+1
IDL> print,memory(/high)
   34057465
```

So, here a = a+1 requires 50 Mb of memory, but a = temporary(a)+1 only requires 34 Mb of memory.

But as shown below, there is no advantage of writing
a=transpose(temporary(a))
rather than simply a = transpose(a).

```
IDL> a = lonarr(2048,2048) & tmp =memory() &  a = transpose(a)
IDL> print,memory(/high)
   50834836
IDL> a = lonarr(2048,2048)& tmp =memory() &  a = transpose(temporary(a))
IDL> print,memory(/high)
   50834836
```

One reason I don't have complete confidence in this result is that at least one RSI-supplied procedure (laguerre.pro) includes the statement

```
   xPrec = FLOAT(TEMPORARY(xPrec))
```

which I'd say is a useless use of TEMPORARY().

Wayne Landsman landsman@mpb.gsfc.nasa.gov