Subject: Re: Use of Temporary( ) vs an Optimised Compiler
Posted by Pavel A. Romashkin on Tue, 27 Nov 2001 17:24:01 GMT

View Forum Message <> Reply to Message

Craig Markwardt wrote:

>
>> a = 2*a + b/TEMPORARY(a)
>
> Since A appears twice on the right hand side, the compiler would need
> to be smart enough to not overwrite A after its first appearance. In
> fact, I am not sure that IDL makes any guarantees about order of
> evaluation and side effects. Isn't it possible that the TEMPORARY()
> gets called before the first A is evaluated?

I recently had to think about memory allocation in IDL for the first
time as I had to use large arays (well, some 4E+7 points, astronomy guys
don't laugh).
In my case it appeared faster (and sometimes the only way possible to
avoid infufficient memory errors) to split the above expression

```
a = 2*a +b/temporary(a)
```

into

```
a = 2*temporary(a)
a = 2*b/temporary(a)
```

It takes a while even to do math on xE+7 points, and allocating arrays
further slows things down.
In this simple case, this is easy to do, but some expressions were quite
hard to split like this. There are probably better ways.
Cheers,
Pavel