Subject: Re: rounding errors
Posted by Paul van Delst on Fri, 27 Apr 2001 15:47:02 GMT
View Forum Message <> Reply to Message

Randall Skelton wrote:
>
> On Fri, 27 Apr 2001, Liam E. Gumley wrote:
>
>> This is a subtle but important point. DOUBLE() is a type conversion
>> function, and
>>
>> a = double(2.348339)
>>
>> shows a FLOAT argument being converted to a DOUBLE. The safest way to
>> 'cast' a double variable is
>>
>> a = 2.348339d
> [snip]
>
> Wow... I am glad that I have now learned that particular 'IDL feature'
> early on in my PhD. Just yesterday, I convinced the department that we
> really need a few good IDL programming books as the current
> 'learning-by-fire' approach could have some unfortunate consequences ;)

This "feature" has absolutely *nothing* to do with IDL. The same thing occurs in other
languages, e.g. Fortran, C, etc. Floating point numbers, in general, cannot be represented
exactly and you have to keep that in mind when writing code - particularly if the 6th or
7th decimal place is important to you. You could just as well ask why, when you try to
this:

IDL> print, (1.0e32)^2

you get the IDL "feature" result:

       Inf
% Program caused arithmetic error: Floating overflow

Computers have absolutely no intelligence at all - they just do what you tell 'em. If you
aren't explicit (e.g. declare a float 2.348339 when you *really* mean 2.3483390000000000)
there are defaults they fall back on (e.g. assume single precision float rather than
double).

> Oh well, back to searching my IDL code for incorrect double precision
> constants...

No, back to searching your IDL code for your implicitly declared single precision
constants and changing them to explicit double precision declarations.

paulv

--
Paul van Delst   A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP   Drink deep, or taste not the Pierian spring;
Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
Fax:(301)763-8545   And drinking largely sobers us again.
paul.vandelst@noaa.gov   Alexander Pope.