
Subject: Re: rounding errors

Posted by [James Kuyper](#) on Wed, 02 May 2001 15:07:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

Paul van Delst wrote:

>
> 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

I think you're misunderstanding the "feature" of IDL that surprised me as much as it surprised Randall and Liam. This has everything to do with IDL, and nothing to do with expecting exact representation of a finite-length decimal fraction. By default, in C 2.348339 represents a double precision number, not a single precision one, and I'd never realized that the IDL convention was different. That probably explains a number of the wierd results I've seen.
