

---

Subject: Re: rounding errors

Posted by [Craig Markwardt](#) on Fri, 27 Apr 2001 13:44:10 GMT

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

---

"Dominic R. Scales" <Dominic.Scales@aerosensing.de> writes:

> Alex Schuster wrote:

>> You just have to define the 2.56989 as a double: b = 2.56989d :)

>>

>> Remember that the floating point value 2.56989 is not the same as the

>> real number 2.5698900000000000..., but just some value around

>> 2.5698900. Converting it to a double introduces more digits to the

>> right, but they are pretty random.

>>

>> Alex

>> --

>> Alex Schuster Wonko@weird.cologne.de PGP Key available

>> alex@pet.mpin-koeln.mpg.de

>

> Danke Alex :)

>

> the prints were more of an example. In the 'real' world,

> the data is read in from a file with readu and is an array

> of float variables. I want to perform the following maths

> in double but with the cast to double I already introduce

> 'pretty random digits to the right', as you say.

> I'd really like to avoid calling something like double(string(a))

> for some large array...

In principle you should not have to worry about those extra random digits. Since your original data are floating point, then they contain no more than 7 digits of precision. Any digits beyond the 7th are simply \*undefined\*. If you want higher precision then you should recreate your file using double precision arithmetic from the start.

Even if you use double precision then you will still have random digits at the end, but just farther out:

```
IDL> print, 2.56989d, format='(D30.20)'  
2.56989000000000000000767
```

This is a consequence of how numbers are represented in the base-two number system. Generally speaking the relative uncertainty will be (MACHAR()).EPS for floating point in IDL.

Craig

--

-----  
Craig B. Markwardt, Ph.D.      EMAIL:   craigmnet@cow.physics.wisc.edu  
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response  
-----

---