Subject: Re: fix(4.70*100) is... 469
Posted by mmeron on Thu, 19 Apr 2007 16:50:36 GMT
View Forum Message <> Reply to Message

In article <MPG.20912dc2d706b423989f47@news.frii.com>, David Fanning
<news@dfanning.com> writes:
> mmeron@cars3.uchicago.edu writes:
>
>> Consider what "same number of significant digits mean. For example,
>> consider that 1.23456*10^20 and 1.23456*10^(-20) have same number of
>> significant digits.
>
> Alright, you have completely lost me here. Can you
> expand this argument just a wee bit more? :-)
>

Certainly. The floating number is stored as two parts, mantissa and power (for the garden variety float you've 24 bits for the mantissa and 8 for the power). The mantissa specifies the significant digits, which are then multiplied by the appropriate power. The storage is binary, of course, but for the purpose of this argument we may look at decimal. So, if you store, say, 7 significant digits, your number is of the form 0.abcdefg * 10^p, where a...f are digits between 0 and 9. If you take two numbers such that their true (as opposed to stored) expansion has same first 7 significant digits while differing at the 8th, they'll be stored as same number. So, roughly, one can say that the accuracy of the stored number is 0.00000005 *10^p (note, 7 zeroes for the significant digits, then half the maximum for the next). So, the storage error, for fixed number of decimal places, is relative, not absolute, it is around 0.00000005/0.abcdefg. As the magnitude of the number grows, so does the error. As you can see in the following sequence

```
IDL> print, 1 + 1e-8 - 1
    0.000000
IDL> print, 1e4 + 1e-4 - 1e4
    0.000000
IDL> print, 1e8 + 1 - 1e8
    0.000000
IDL> print, 1e28 + 1e20 - 1e20
 1.00000e+028
IDL> print, 1e28 + 1e20 - 1e28
    0.000000
```

Mati Meron | "When you argue with a fool,
meron@cars.uchicago.edu | chances are he is doing just the same"