Subject: Re: rounding errors
Posted by Karl Schultz on Fri, 27 Apr 2001 16:07:21 GMT
View Forum Message <> Reply to Message

You could try

d = 2.56989d

to use a double-precision literal.

Also,

b=double(2.56989)

causes a single-to-double conversion because the literal is single-precision.

The IEEE mantissa (fractional part) of this single-precision literal is (in hex): 48F228
The conversion of a single to double will make the mantissa something like 48F2280000000 because it just adds zeros to pad out the mantissa.
But the mantissa of 2.56989d is 48F227D028A1E.

Comparing:
48F2280000000 (single-precision literal converted to double)
48F227D028A1E (double precision literal stored as double)

The first number is slightly larger, which accounts for the extra non-zero decimal digits you point out below. When you convert the 2.56989 literal to a single-precision float, the last mantissa bit is rounded up to get as close to the literal as possible. When you convert that single to a double, the "too high" bit is simply carried over to the double precision mantissa and the rest of the lesser-significant bits are zeroed out. So, the number still seems "too high". But if you store a double-precision literal as a double, all the bits in a double-precision mantissa are used and the result is what you expect.

And '2.56989' is different from 2.56989 because the latter is an implied single-precision floating point number. The former is a string which gets converted directly to double-precision if you say double('2.56989').

"Dominic R. Scales" <Dominic.Scales@aerosensing.de> wrote in message news:3AE9330C.29D059E9@aerosensing.de...
> HELP!
>
> What gives? Is there any numerical math guy/gal out there
> who can tell me how this happens? It seems to me, that

```
>    the accuracy of the second/third cast ist WAY off.
>
>     a=double('2.56989')
>     b=double( 2.56989 )
>     c=double(float('2.56989'))
>  .
>     print,a,b,c,format='(d)'
>
>         2.5698900000000000 <---- this is what i want to have
>         2.5698900222778320
>         2.5698900222778320
>
```