Subject: Re: rounding errors
Posted by thompson on Fri, 27 Apr 2001 15:23:49 GMT
View Forum Message <> Reply to Message

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

> 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

The first question that comes into my mind is why don't you simply cast your literal as double precision in the first place:

```
 IDL> a = 2.56989d0
 IDL> print,a,format='(d)'
       2.5698900000000000
```

It's instructive to look at these numbers in the binary representation actually used by the computer. The floating point number 2.56989 has the following representations in hexadecimal and binary formats:

```
   Hex:     40247914
   Binary:  01000000010010001111100100010100
                 ^^^^^^^^^^^^^^^^^^^^^^^
```

when this is converted from floating point to double precision, as in your examples b and c above, you get

```
   Hex:     40048F2280000000
   Binary:   0100000000000100100011111001000101000000000000000000000000000 0000
                   ^^^^^^^^^^^^^^^^^^^^^^^
```

It's a little harder to tell in hex format, but the binary format makes it plain that the mantissa part is exactly the same as in the original floating point number, just shifted over a little (the exponent part is bigger in double precision), and filled with zeroes, just as you would expect. The confusion

results from the distinction between decimal representation used by people and the binary representation used by computers.

William Thompson
