Subject: Re: Strange problem
Posted by Bhautik Joshi on Mon, 26 Nov 2001 03:55:17 GMT
View Forum Message <> Reply to Message

> type: for i=0., 0.8., 0.1 do print,i
> That's bad isn't it?
OK lets drag this out a little further :)

Lets translate this into a WHILE loop and try it again, using INTS.

```
MOO>i=0
MOO>while i le 8 do begin print, i & i=i+1
      0
      1
      2
      3
      4
      5
      6
      7
      8
```

Now again, using FLOAT:

```
MOO>i=0.0
MOO>while i le 0.8 do begin print, i & i=i+0.1
     0.00000
    0.100000
    0.200000
    0.300000
    0.400000
    0.500000
    0.600000
    0.700000
```

What's going on? Shouldn't it count up to 0.8?

```
MOO>print, i-0.8
  5.96046e-08
```

Basically what has happened is that when 0.1 gets added to 0.7, the result is > 0.8, so the loop gets terminated early!

This is because what you see in the screen is rounded to a couple of decimal places; where it says 0.7 it should really be something like 0.7000000001. As the addition occurs, accuracy is lost.

To demonstrate, try this routine:

```
pro foo, l
 loopvar=FLOAT(0.0)
 lim=FLOAT(l)
 inc=FLOAT(0.1)
 i=FIX(0)

 while (loopvar le l) do begin
  actvar=FLOAT(i)*inc
  print, 'loopvar: ',loopvar,' actvar: ',actvar,'
diff:',(loopvar-actvar)
  loopvar=loopvar+inc
  i=i+1
 end
end
```

So, trying it out:

```
MOO>foo, 0.8
loopvar:       0.00000 actvar:       0.00000 diff:      0.00000
loopvar:      0.100000 actvar:      0.100000 diff:      0.00000
loopvar:      0.200000 actvar:      0.200000 diff:      0.00000
loopvar:      0.300000 actvar:      0.300000 diff:      0.00000
loopvar:      0.400000 actvar:      0.400000 diff:      0.00000
loopvar:      0.500000 actvar:      0.500000 diff:      0.00000
loopvar:      0.600000 actvar:      0.600000 diff:      0.00000
loopvar:      0.700000 actvar:      0.700000 diff:  5.96046e-08
```

So whats printed on the screen is 0.700000, but the actual number is 0.70000006 (roughly - remember that on a computer, only exact multiples or um.. whats the word - numbers that have been divided exactly by - 2 can be exactly represented. ie. ..2.0,1.0,0.5,0.25.. are prescise; however, ..2.1,1.1,0.51,0.251.. have a degree of inaccuracy).

The exact same thing occurs when you write the same thing in C, so its not just something limited to IDL.

The morals of this story:
* don't let your loop variable=data variable
* floats have only 7 digits of prescision (on most platforms); where you are trying to be exact (and loop variables have to be exact) either use a DOUBLE (which has 15 digits of prescision on most systems) or use an INT to count the number of times the loop goes around
* The 29,249th digit of pi is a seven.

Cheers,
Bhautik

--
 /----------(__)----- ----\
| nbj@imag.wsahs.nsw.gov.au | phone: 0404032617 |..|--\ -moo |
| ICQ #: 2464537 | http://cow.mooh.org | |--| |
\----------\OO/|| ------/