Subject: Re: Strange problem
Posted by Martin Downing on Mon, 26 Nov 2001 15:00:17 GMT
View Forum Message <> Reply to Message

Andre,

Joshi is right, this behaviour is due to the lack of precision in
floating point number representation. With your for loop

```
for i=0., 0.801, 0.1 do print,i
```

The code execution can more easily be visualised as

```
i = 0.
while i LE 0.8 do begin
   print, i
   i  = i + 0.1
endwhile
```

Thus:

```
IDL> for i=0., 0.8, 0.1 do print,i
0.000000
0.100000
0.200000
0.300000
0.400000
0.500000
0.600000
0.700000
```

So what was the final value of i?

```
IDL> print, i
0.800000
```

Oh, isnt that the value of the upper bound?

```
IDL> print, i EQ 0.8
0
IDL> print, i - 0.8
5.96046e-008
```

Clearly not! Slightly more than 0.1 was added each time, so there was a
small excess to i when representing 0.8

So the moral is that you have to be very careful when applying comparison
operators to floating point numbers, one of which is implicitly applied in

the FOR statement. Now you realise the problem, the answer is to be a little less strict with your comparisons. With FOR loops you can add a small excess, relative to the increment, to the upper bound:

```
IDL> for i=0., 0.8001, 0.1 do print,i
0.000000
0.100000
0.200000
0.300000
0.400000
0.500000
0.600000
0.700000
0.800000
```

Out of interest notice that the final value of "i" is now 0.9:

```
IDL> print, i
0.900000
```

> should i be worried?

Well if you write code which depends on floating point numbers having perfect precision then yes!
If you wanted to compare two floats for equality, you have to rethink what you mean by "equal", i.e. how exact does this application need the variables to be?
Relying on doubles is not a robust solution, so instead of writing:

```
IF a EQ b THEN ...
```

write

```
myPrecision = 0.001
IF abs(a-b) LT myPrecision THEN ......
```

I hope this helps

Martin

"Bhautik Joshi" <nbj@imag.wsahs.nsw.gov.au> wrote in message news:3C019CFD.B20DB925@imag.wsahs.nsw.gov.au...
>> Anybody know what's going on?
>
> My spin on it:
>
> Well, I think it may be a problem that goes right down to the core, and
> is not just restricted to FOR loops.

>
> Example:
>
> MOO>a=replicate(0.1,100)
> MOO>print, total(a)
>   10.0000
> MOO>print, total(a) - 0.1*100
>   1.90735e-06
>
> argh! where I think the problem may lie is with the binary
> representation of floating point numbers. Now, its been a
> looooooooooooooooooooooong while since I did computer architecture
> (yes, it is a real subject at uni) but as far as I remember, in floating
> point binary, there is no 'exact' representation for 0.1.
>
> However, lets take 0.5 - which is a quarter of 2; something nice and
> (pardon the expression) base-twoish.
>
> MOO>a=replicate(0.5,100)
> MOO>print, total(a) - 0.5*100
>   0.00000
>
> Lets change the game slightly again:
>
> MOO>a=replicate(0.51,100)
> MOO>print, total(a) - 0.51*100
> -4.95911e-05
>
> should i be worried?
>
>
> --
> /---(__)----- ----\
> | nbj@imag.wsahs.nsw.gov.au | phone: 0404032617 |..|--\ -moo |
> | ICQ #: 2464537 | http://cow.mooh.org | |--| |
> \---\OO/|| ------/