Subject: Re: print and precision
Posted by cmancone on Thu, 14 Feb 2008 19:47:27 GMT
View Forum Message <> Reply to Message

On Feb 14, 11:50 am, elwood <epolo...@uwsp.edu> wrote:
> I have a tiny understanding of how numbers are stored in computers
> and how a float only has 32 bits to store a number, so for some
> numbers it may run
> out of bits before it can store the precise value that the user
> intended.
>
> My question is, what is the default form of the print statement doing?
>
> for example:
> x=3.3
>
> If i Understand correctly, the floating point binary representation of
> this number is
> 11.01001100110011001100110011001100
> which exceeds 32 bits
>
> so I'd expect to get something like
> 3.299999999813735
> due to truncation
>
> But if I print,x
> I get 3.3
> I'm sure theres some misunderstanding on my part here, but is there a
> document I could read
> concerning how the print command works with regard to floating point
> precision?
>
> Thanks

As near as I can tell the print command pretty much just picks a random number of digits to print out. I generally just don't use the print statement by itself in a program, I always pass it a format command. In general it just prints out 5 or 6 decimal places. Because the print statement just uses a fixed number of decimal places, you will often get the value you want to get rather than the floating point storage errors. So for example:

```
x = 3.3
IDL> print,x
      3.30000
IDL> print,x,format='(f9.7)'
3.3000000
IDL> print,x,format='(f11.9)'
```

3.299999952

Of course floating point doesn't store 9 digits of precision, so I
have no idea where the extra digits are coming from anyway...

My rule is to just always give a format statement in programs. That
way you always know exactly what you're getting.

Subject: Re: print and precision
Posted by lasse on Fri, 15 Feb 2008 07:44:46 GMT
View Forum Message <> Reply to Message

Hi there

> As near as I can tell the print command pretty much just picks a
> random number of digits to print out.

Far from it! With the 32bits available for type float numbers, you can
only archive a relative precision of 1e-6 to 1e-7 so it makes
absolutely no sense to print more then 6-7 digits of a float type
variable.

> Of course floating point doesn't store 9 digits of precision, so I
> have no idea where the extra digits are coming from anyway...

Precisely. So why use a format code to print them out? If you do a

x = 3.3

then IDL saves precisely that number to a relative precision of the
float type.

Cheers
Lasse Clausen
