Subject: Re: Inaccuracies
Posted by thompson on Tue, 14 Nov 1995 08:00:00 GMT
View Forum Message <> Reply to Message

Andy Loughe <afl@cdc.noaa.gov> writes:

> Ok, I am sure this has been discussed before, but let
> me start this thread again. I wish to create a 15-element
> vector which contains the numbers -1.4 to 1.4 by an increment
> of 0.2 I also wish the sum of these elements to be zero
> (No, this isn't the new math). Here is what I tried...

> TRIAL #1
> ========
> IDL> a = findgen(15)*.2 - 1.4
> IDL> print, total(a)
> 7.15256e-07

> Hmmm! Not so good.

> Maybe I am missing something here, but this kind of behavior
> makes IDL a bit problematical for scientific use. With only 15
> numbers and double precision arithmetic, I can't believe this
> would fail in FORTRAN or C!

Here's a FORTRAN program I tried this on.

```
      program test
c
      total = 0
      do i = 1,15
        a = (i-1)*0.2 - 1.4
        total = total + a
      enddo
      write (*,*) total
c
      end
```

and the result of running this program?

```
 > a.out
   7.1525574E-07
```

Exactly the same answer as IDL gives! If instead I do it in double precision,
I get

```
 > a.out
   4.440892098500626E-015
```

Again, exactly the same answer as IDL! It appears that IDL is working correctly within the confines of the floating point arithmetic of the computer.

Interestingly enough, the same problem done in IDL on an OpenVMS computer gives exactly zero when done in single precision, but 8.8817842e-16 in double precision. VMS uses a different floating point format than the standard IEEE representation used in most modern computers.

The bottom line is that all computers are subject to round-off errors when doing floating point arithmetic. This will occur no matter what software package is used.

Bill Thompson
