Subject: Re: Problems with double precision in IDL
Posted by thompson on Fri, 18 Jun 1993 19:04:05 GMT
View Forum Message <> Reply to Message

In article <16JUN199312122750@stars.gsfc.nasa.gov> isaacman@stars.gsfc.nasa.gov (Subvert the Dominant Paradigm! (301) 513-7769) writes:
> We have a potentially VERY serious problem with the COBE data analysis
> because of the way IDL seems to be (mis?)handling double precision
> numbers. Here is an example of how IDL treats floating point numbers
> when converting them to double precision. The operations were performed
> on a DECstation.
>
> z=.32
> print,f2,double(z)   ; If a variable is declared DOUBLE this is what happens.
> 0.319999992847
> print,f2,.32d   ; If the "d" notation is used instead it's accurate.
> 0.320000000000
> print,double(z)-.32D
> -7.15255737e-09
> print,[double(z)-.32D]/.32D
> -2.23517418e-08

Stuff deleted

Here's a counter example from FORTRAN. I ran the following program on my Sun workstation

```
      program main
c
      real*4 a
      real*8 b
c
      a = 0.32
      b = a
      write (*,*) a,b
c
      end
```

and got the following results

```
   0.320000   0.31999999284744
```

This is the closest FORTRAN equivalent that I can think of to the IDL commands

```
z=.32
print,double(z)
```

that I can think of.

As has been mentioned before, this sort of behavior is endemic to all programming languages, and must be kept in mind when writing programs involving double precision variables. For example, I've seen FORTRAN programs that had statements in it like

```
DOUBLE PRECISION VAR1
VAR1 = 7./9.
```

which will introduce the same sorts of problems. In fact, this was in a commercial software package!

Bill Thompson
