Subject: Re: Problems with double precision in IDL
Posted by zawodny on Fri, 18 Jun 1993 16:41:06 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

>
>
> Thanks,
>
> Rich Isaacman
>
> COBE Project
> NASA/Goddard Space Flight Center

Isn't this apparent error at about the magnitude of the "fluctuations" in the residual temperature maps from COBE that are being used to "prove" some aspects of Big Bang theories? Gee, I hope that all the kudos and PR that has been given to COBE are not for nought because of a programming error. I was excited by those results myself.

Obviously a prior post is correct, numbers are not stored in computers as ASCII representations. You cannot gain precision in a previously declared varaible by simply declaring it to be double precision later on. This is true of a program written in any language and is not a "problem" with IDL.

--
Joseph M. Zawodny (KO4LW) NASA Langley Research Center
Internet: zawodny@arbd0.larc.nasa.gov MS-475, Hampton VA, 23681-0001
Packet: ko4lw@wb0tax.va.usa