Subject: Re: exp function bug
Posted by Peter Mason on Fri, 06 Dec 1996 08:00:00 GMT
View Forum Message <> Reply to Message

On Thu, 5 Dec 1996, Robert Cannon wrote:

> Can anyone tell me what is going on here, or if it happens on other
> platforms?
> The first time I call exp (-710.72026d0) after starting idl I get:
>   hera:~> idl
>   IDL. Version 4.0.1 (IRIX mipseb).
>   IDL> print, exp (-710.72026d0)
>   2.8462073e+134
>   IDL> print, exp (-710.72026d0)
>   2.1782760e-309
> This is on an sgi crimson. I get the impression the same may happen on
> an alpha, but for a different value - it has to be -710.72026d0 on the
> sgi: -710.72025d0 works fine.....

I think that exp(-710.72026d0) is on the borderline of giving a floating-point underflow on your platform.   Here are some approximate values of "a" in EXP(a) beyond which a floating-point underflow occurs on IDL platforms I have available here.   (I haven't observed inconsistencies like you did.)

PLATFORM	A
Intel Pentium, NT3.51	-708.D
DEC ALPHASTATION 200 4/233, NT3.51	-708.D
DEC ALPHA/AXP 3000/500, OSF	-744.D

(|A| is much smaller for single precision, of course.)

I don't really understand what could cause those inconsistent answers.
Here's, well, a stream of consciousness...
EXP(large neg value) used to silently return 0.0 in older versions of IDL (the way I prefer it to do), and when RSI introduced support for some floating point "denormals" (Infinity and NaN), things were quite strange initially:
On my DEC OSF platform, EXP(-x) would generate an underflow error for moderately large x, but would cause IDL to crash with a segfault given a suitably large (much larger) x.
I think that when you're working right at the edge of FP precision like this, you can easily run into platform-specific differences.   (For example, ALPHA/NT doesn't handle denormals in hardware.)
But I also think that EXP(-710.72026d0) == 2.8462073e+134 is unacceptable at any time, and that you should send in a bug report.   (Perhaps there is some bug in IDL's FP exception-handling code on the SGI.)

Peter Mason