Subject: Re: More on Exp bugs
Posted by Peter Mason on Fri, 13 Dec 1996 08:00:00 GMT
View Forum Message <> Reply to Message

On 11 Dec 1996, David Siskind wrote:

> Anyway, regarding the Crimson user who found wierd things with
> Exp(-710.something), it must be an SGI/IDL thing. I use
> an Indigo and IDL 4.0.1 and I once spent a day (plus a post to this
> group) trying to figure out what appears to be the exact same thing.
> Now I'm real careful about floating under or overflows when doing Exp.
> This did not seem to be a problem with 3.6.1.

Here I go holding forth and stirring once again.

I'm convinced that this problem is due to the introduction of "NaN" and "Infinity" support since IDL 4.0. I also think that the addition of these features is what caused the floating-point slowdown observed on at least some platforms, when comparing IDL 3.6.1 to IDL 4.0.x.

When IDL4.0 first came out, I must say that I didn't really notice the slowdown on the ALPHA/OSF system I use. (Perhaps this O/S supports denormals by default, and RSI's enhancements didn't add much extra overhead?) But I didn't like the way I could segfault IDL with just exp() calls. (By the way, I STILL can. On a DEC 3000/500 with exp(-90.0) followed by exp(-9000.0). (Single precision.) Can anyone else do this?)

Anyway, I think that RSI was faced with a tough task in implementing these FP features on all the IDL platforms; I imagine that each one has its qirks and peculiarities when it comes down to FP denormals and exceptions. And some take a bigger performance hit than others. (ALPHA/NT certainly takes a major performance hit. Doing FP the lean way it likes to, it makes ALPHA/OSF look tired. Doing it with denormals enabled and various exceptions changed takes away its edge.)

If anyone's still reading this...
I realise that this is a MAJOR stir, but I was wondering what people's views are on IDL's "NaN" and "Infinity" support?
Personally: I haven't yet implemented "Infinity" in my IDL programs, and I haven't used "NaN" much at all. I like the idea of "NaN", but I started many of my programs before it was around in IDL, and so I found other ways to cope with "bad values" and the like. I can't be bothered with FP underflows (just give me 0). Overall, I actually prefer FP support in IDL the way it was in 3.6.1.

Peter Mason