Subject: Re: Wow. exp() difficulties...
Posted by William Clodius on Fri, 25 Jul 1997 07:00:00 GMT
View Forum Message <> Reply to Message

I have a few comments on Amara Graps post, mostly related to efficiency:

Amara Graps wrote:
>
> <snip>
>
> The primary advantages of using integer and fixed point numbers are that
> they take up less storage disk space and calculations with these numbers
> are fast.

Because of their simplicity it is in principle possible to make integer and fixed point number arithmetic faster than floating point arithmetic. However, in practice nowadays floating point arithmetic is as fast as integer arithmetic for addition and subtraction and often faster than integer arithmetic for multiplication and division. Both integer and floating point arithmetic use a lot of additions and multiplications, so processor designer emphasize making them go as fast as possible, typically a fixed multiple of processor clock speed. Integer arithmetic typically rarely uses multiplication or division so they are not highly optimized, while floating point arithmetic often uses multiplication and division and is often well optimized for those operations.

(A side note: while floating point arithmetic having performance comparable to integer arithmetic is generally a new phenomena, poor performance of integer multiplications and divisions is not new. There was a processor design in the early 60's, I believe by IBM, that had an integer division operation that was something like 1/10,000 the speed of other integer operations. Although integer division operations are rare, they are typically almost exclusively used to convert to and from decimal representations, they are used in more than 1/10,000 operations and the computer spent most of its time inefficiently computing integer divisions.)

> <snip>
>
> Double Precision Floating Numbers
> The advantages of using double precision number types are many. A double
> precision number usually yields fifteen or sixteen decimal places, which
> is more than adequate for most calculations. The largest normalized
> double precision number is about 1.798E+308. The smallest normalized
> number is about 2.225E-308. (Again put a negative sign in front to
> determine the largest and smallest *negative* number.) Even though the
> speed of performing double precision computations is twice as slow as
> performing single precision computations, the computation speed on

> today's computers is still fast.

In recent years this speed ratio is no longer valid. Some floating point processing units perform all their calculations in double or extended precision. As a result single precision can have the additional overhead of conversion to and from the double precision format. Whether single or double precision is more efficient in such cases depends on the magnitude of this overhead vs. increased cache misses due to the larger memory requirements for double.

> <snip>

--

William B. Clodius Phone: (505)-665-9370
Los Alamos Nat. Lab., NIS-2 FAX: (505)-667-3815
PO Box 1663, MS-C323 Group office: (505)-667-5776
Los Alamos, NM 87545 Email: wclodius@lanl.gov