

Re: Correction factor in computing exp()?

Source: <http://sci.tech-archive.net/Archive/sci.math.num-analysis/2006-07/msg00222.html>

- *From:* "Dave Dodson" <dave_and_darla@xxxxxxxx>
 - *Date:* 16 Jul 2006 08:42:40 -0700
-

lcw1964 wrote:

Here is a snippet of the Fortran code from

<http://www.netlib.org/specfun/erf>:

```
YSQ = AINT(Y*SIXTEN)/SIXTEN
DEL = (Y-YSQ)*(Y+YSQ)
RESULT = EXP(-YSQ*YSQ) * EXP(-DEL) * RESULT
```

Does anyone know the computational point of doing this?

It enhances accuracy on computers that don't have guard digits. On such a machine, typically, $Y*Y$ will be accurate to about 1/2 ULP (unit in the last place). But $YSQ*YSQ+DEL$ should have more accuracy because $YSQ*YSQ$ and $Y-YSQ$ can be computed exactly, and $Y-YSQ$ will be smaller than Y if $\text{abs}(Y) > 1/16$. Thus, the 1/2 ULP introduced in $Y+YSQ$ will be multiplied by a small number.

Example: Suppose the machine uses IEEE single precision and $Y = 4.00xxx$. Then YSQ will be 4.0 and $Y-YSQ = 0.00xxx$, which is less than $Y/400$. Then $DEL < (Y/400)*(2*Y) = Y*Y/200$, and $YSQ*YSQ+DEL$ will have 7 to 8 bits more accuracy than $Y*Y$. You can check this by comparing $Y*Y - YSQ*YSQ$ to DEL .

Dave