

Correction factor in computing exp()?

Source: <http://sci.tech-archive.net/Archive/sci.math.num-analysis/2006-07/msg00219.html>

- *From:* "lcw1964" <leslie.wright@xxxxxxxxxxxxxx>
 - *Date:* 16 Jul 2006 00:22:12 -0700
-

I have recently developed an interest in Cody's approximations of erf() and erfc(). 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
```

The point of this is to simply compute $\exp(-y*y)$ and multiply to the result of a rational approximation calculation. Instead of doing it straight ahead, the code multiplies y by 16, takes the integer part, multiplies the 16 back, calls it YSQ, then computes some curious factor DEL. Then the routine takes $\exp()$ of these separated pieces and multiplies the whole thing together.

Does anyone know the computational point of doing this? Is or was there some bug in the Fortran implementation of $\exp()$ that required some sort of correction? Someone familiar with the GSL in C advises me that this turns up in that code too. I have Cody's original 1969 paper on which this code is based (indeed, Cody is the author of the code too), and there is no mention of this little adjustment, or correction, or whatever it is.

Many thanks in advance. This is driving me nuts with head scratching....

Les

.