# Re: Degree of accuracy needed to prove two algorithms are the same?

***Source:*** http://sci.tech−archive.net/Archive/sci.stat.math/2004−08/0089.html

---

***From:*** Mark Von Tress (*vontressms_at_cs.com*)
***Date:*** 08/04/04

```
Date: 4 Aug 2004 06:59:52 −0700
```

Eric,
It depends on your purpose. If you are just trying to make sure that you understand what your software's algorithms are doing, then you can accept more error in the output. Here is an example from my more industrious past. PROC MIXED from SAS was new. I wanted to make sure I understood it, or at least the parts of it that I used. So, I took to C++ and a matrix library and proceded to code the variance component models using Hartley's W transform. I got it to agree with proc mixed in the ML and REML estimates to about 4 or 5 decimal places. I was also able to get their degrees of freedom using there descriptions in the documentation. I did the same thing for proc logistic. I had confirmed my understanding.

However, I would not go so far as to report the results from my program to a regulatory agency like FDA because I only got 4 or 5 decimal accuracy, and SAS was already available to me. Plus, SAS has the benifit of having been developed in a strict environment by dedicated professional programmers and statisticians. There is no way I can reproduce that level of quality as an individual.

Now to answer the question you asked. I not aware that such a standard exists. If you are trying to develop a professional software for sale, you are headed for hard times. That market is dominated. Now, if you are just doing research and testing new ideas about the statistical properties about your research, then the size of the numerical error in your routines needs to be much less than the random error in the data. Mathematica is up to that challenge.

If you get accuracy to 4 or 5 decimal places for the same input to a complicated algorithm, then you will be fine for most things. My feeling is that iff the random variation in the data is at least 100 times larger than than the numerical error, then the variation in the output variables will be due to the randomness in the system being studied rather than numerical error.

Just make sure that any matrices you invert are well conditioned, or

that your inversion algorithm can tolerate poorly conditioned
matrices. This can intruduce large amounts of numerical error that
will be larger than the variation in the data you are trying to study.
The distribution functions in Mathematica are probably good enough in
the useful range of their data to be right. Most programs will handle
the extremes of a distribution differently, so there is a watchout.
They are all different in the centra