Sources of Error

Jonathan Goodman

January 20, 1996

In numerical computing we never expect to bet the *exact* answer.[1] There are three basic sources of error: roundoff error, truncation error, and statistical error. Roundoff error arises from the fact that computer arithmetic is not exact. If we know x and y and write

```
z = x + y
```

then the computed z will usually not be exactly the sum of x and y. Exactly what z will be is discussed when we cover IEEE arithmetic. Truncation error arises from the use of approximate formulas, such as

$$f'(x) \approx \left(f(x+h) - f(x) \right) / h \quad ,$$

that are not exact "even in exact arithmetic". Some computations, particularly in linear algebra, have no truncation error. In most computations that have truncation error, the truncation error is much larger than roundoff error. Also, as will become clear throughout the course, much more can be done to reduce truncation error than roundoff error. For these reasons, most scientific computing courses (including this one) spend more time analyzing truncation error than roundoff error. Statistical error arises only in Monte-Carlo computations.

Often, inaccuracy in a computed answer is very large compared to the size of roundoff or truncation (or statistical) error introduced during the computation. This is because error can be amplified during the stages of the computational algorithm. Such error amplification is inevitable if the problem being solved is *ill conditioned*. An ill conditioned problem is one in which the answer is so sensitive to the input parameters that no computational algorithm could would be expected to give an accurate approximation to it. However, even well conditioned problems may have solution algorithms that are *unstable*. To get an accurate answer, the problem being solved must be well conditioned (this may be out of the hands of the person trying to solve the problem) and the solution algorithm should be stable.

[1]This is almost the definition of numerical computation. Computations that aim for the exact answer, such as symbolic algebraic manipulation or prime factorization of large integers, are often called "non numerical".

It is important to be aware of the possibility of inaccuracy via error amplification because this source of error is hardest to discover by standard debugging techniques. In a large calculation, the error may grow a seemingly negligible amount at each step but grow to swamp the correct answer by the time the computation is finished. One of the main uses of mathematical analysis in scientific computing is in understanding the conditioning of problems and the stability of algorithms.