3 Errors

There will always be a small but steady demand for error analysts to … expose bad algorithms' big errors and , more importantly, supplant bad algorithms with provably good ones

W. Kahan (1980)

There's no sense in being precise when you don't even know what you're talking about

John von Neumann

Topics Covered:

- What types of errors might one encounter in numerical computing and what are their sources?
- When is a computed solution accurate enough?
- What is the difference between precision and accuracy?

3.1 Sources of Errors

One of the first challenges when working in numerical analysis is the question of how to determine when a computed solution is sufficiently accurate. In order to talk about this we first need to develop some nomenclature and definitions that will allow us to quantify more precisely how good a solution we have.

Definition 3.1 Suppose we have x and an approximation $\hat{x}$. Then the **absolute error** is given by

$$|x - \hat{x}|.$$

Definition 3.2 Similarly the **relative error** is defined by

$$\frac{|x - \hat{x}|}{|x|}.$$

Both absolute and relative errors are used in our analysis, but the relative error is generally preferred in practice as it is scale independent, and as long as you stay away from $x = 0$.

Precision and Accuracy

Precision and accuracy are often confused. In numerical analysis, they will have a specific meaning.

By **accuracy**, we mean the absolute or relative error of an approximation as defined in (Definition 3.1,Definition 3.2). **Precision** will refer to the accuracy with which the basic arithmetic operations (+, -, *, /) are performed. More of this to come.

One should also note that accuracy is not limited by the precision of a computer - there are many software packages that can provide extended precision in numerical calculations.(Bailey 1993)

3.2 Sources of Errors

The next question that naturally arises is what types of errors might we encounter in solving a problem numerically and where might they arise? As it turns out, there are many different ways to classify errors. We will use the following general categories:

- **model errors/uncertainty**:
  - errors in mathematical models used to approximate a real world problem
  - errors in the input data, e.g. physical measurements, observations, etc.
- **approximation errors**:
  - errors in approximating our problem due to truncation/discretization , e.g. Taylor series, interpolation, integration.
- **roundoff errors**:
  - errors due to the finite precision inherent of computer arithmetic. This is a basic fact of computational mathematics and there is not much we can do to limit these. However, wise choices of algorithms can circumvent some of the problems and bad choices of algorithms can exarcebate them.

Remarks

1. Model errors and their quantification is a subject area all of its own - we will not cover it. The interested student can check out the vast literature on topics such as *uncertainty quantification, verification, and validation.*

2. Approximation errors (truncation/discretization) usually dominate other types of errors and their analysis is a major task of numerical analysis

Example 3.1 (Approximation/Discretization) Error)

Suppose we want to compute an approximation to $f'(x)$ for some function $f(x)$ at the point $x = x_0$.

This situation might arise if you have the function but do not know $f'(x)$, or perhaps it is too expensive to compute.

Let's use Taylor's Theorem. Recall

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x - x_0)^2}{2!} f''(x_0) + \cdots + \frac{(x - x_0)^{(n+1)}}{(n+1)!} f^{(n+1)}(c).$$

for some c between x and x_0. Note, that we need to assume f has $n+1$ derivatives in some interval.

Now let's introduce some new notation. In particular, let

$$x = x_0 + h, \quad h > 0,$$

so that

$$h = x - x_0.$$

Then we can rewrite the Taylor expansion as

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2!} f''(x_0) + \cdots + \frac{h^{(n+1)}}{(n+1)!} f^{(n+1)}(c).$$

Rearranging the terms to put the derivative on the left hand side of the equation gives us:

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} - \left[\frac{h}{2} f''(x_0) + \cdots + \frac{h^{(n)}}{(n+1)!} f^{(n+1)}(c)\right].$$

ⓘ **Idea**

Use the first two terms on the RHS to approximate the derivative at x_0 and "throw away" the rest of the terms.

Taking absolute values, we can see that:

$$\left| f'(x_0) - \frac{f(x_0 + h) - f(x_0)}{h} \right| = \left| \frac{h}{2} f''(x_0) + \cdots + \frac{h^{(n)}}{(n+1)!} f^{(n+1)}(c) \right|, \quad (3.1)$$

where we denote the RHS of Equation 3.1 by the **discretization error**. If $f''(x_0) \neq 0$, then for small enough h, we can estimate the discretization error by:

$$\left| f'(x_0) - \frac{f(x_0 + h) - f(x_0)}{h} \right| \approx \left| \frac{h}{2} f''(x_0) \right|.$$

Again, we are discarding all higher order terms, which if h is "small enough" seems like a reasonable idea (of course we will need to prove this rigorously, but that comes later). In the meantime, it will prove helpful to denote quantities like the right hand side by the following notation:

$$\left|\frac{h}{2} f''(x_0)\right| = O(h).$$

Here by the notation, $O(h)$ (read Big O of h), we mean that a quantity is at most proportional to h, for example Ch, for some constant C. One way to think of it is as

$$\lim_{h \to 0} \frac{O(h)}{h} = C < \infty$$

This can be generalized to higher orders of h as well. (We'll talk more about his later as well.)

The important concept is that we talk about the above *approximation for the derivative as being an $O(h)$ approximation to the true derivative.* For more details (or if you need a refresher) on Big O notation see Section A.1.

Example 3.2 Approximate $f'(x)$ for $f(x) = \cos(x)$ and $h = 10^{-3} - 10^{-15}, x = \pi/6$.

Table 3.1: Absolute and Relative Error in finite difference (F.D) approximation as a function of stepsize h.

h	F.D	abserr	relerr
1e-01	-0.5424323	4.243228e-02	8.486456e-02
1e-02	-0.5043218	4.321758e-03	8.643515e-03
1e-03	-0.5004329	4.329293e-04	8.658587e-04
1e-04	-0.5000433	4.330044e-05	8.660088e-05
1e-05	-0.5000043	4.330117e-06	8.660235e-06
1e-06	-0.5000004	4.330569e-07	8.661137e-07
1e-07	-0.5000000	4.359063e-08	8.718126e-08
1e-08	-0.5000000	8.063495e-09	1.612699e-08
1e-09	-0.5000000	4.137019e-08	8.274037e-08
1e-10	-0.5000000	4.137019e-08	8.274037e-08
1e-11	-0.5000000	4.137019e-08	8.274037e-08
1e-12	-0.5000445	4.445029e-05	8.890058e-05
1e-14	-0.4996004	3.996389e-04	7.992778e-04
1e-15	-0.5551115	5.511151e-02	1.102230e-01

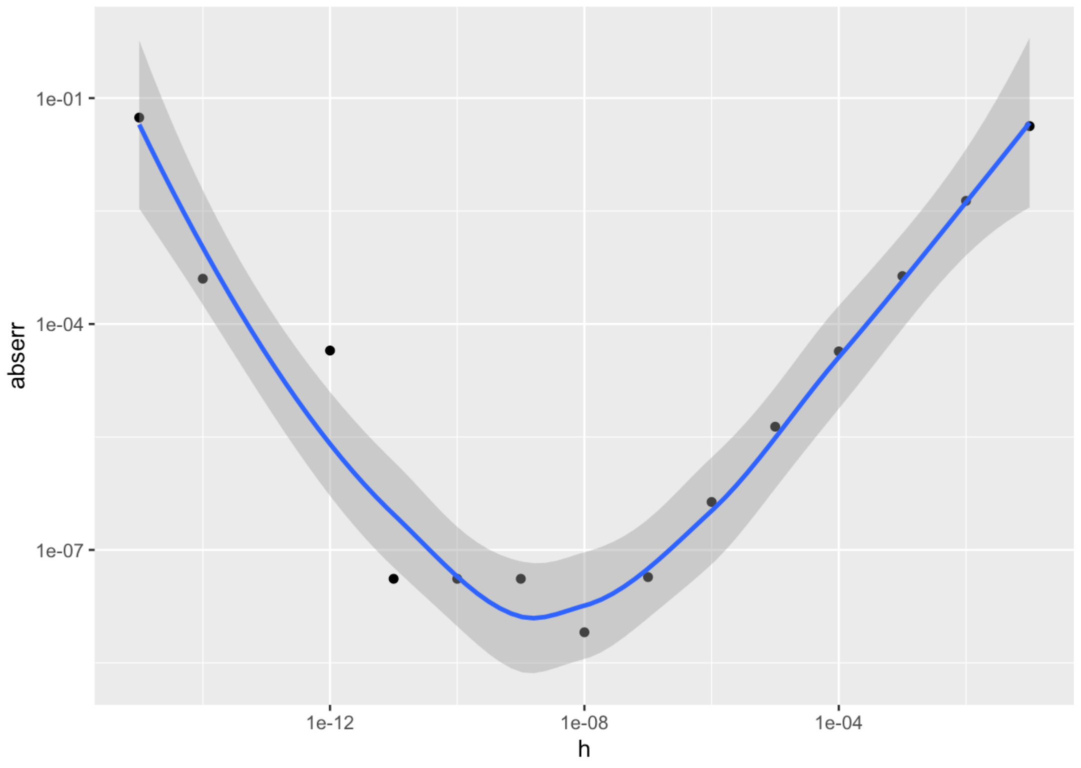

Figure 3.1: Absolute Error in finite difference approximation as a function of stepsize h. Notice the decrease in error as h decreases. Eventually the error increases as roundoff error dominates the truncation error.

Remark

Notice that for each decrease in the value of h by an order of magnitude, both the absolute and relative error have a corresponding decrease in their values. *This is exactly what the theory predicts.* However, it is important to note that this is true only up to a certain point. We'll discuss this more fully when we get to the sections on computer arithmetic and roundoff error.

Exercise 3.1 Approximate $f'(x)$ for $f(x) = \sin(x)$ at the point $x_0 = 1.2$ and using $h = 0.1, 0.01, 0.001$. Can you estimate the discretization error? Explain.

Solution.

3.3 Summary

This section covered

- the concepts of absolute error and relative error,
- the difference between accuracy and precision,
- presented some of the most common sources of errors when modeling problems through the use of mathematical or computational models,
- the effects of discretization error and,
- provided an example for approximating the first derivative of some known function.

▶ Code

```
[1] "Revised: March 05 2024"
```