1 Lecture 1: Introduction

This is a course in numerical analysis. The goal of this course is to understand methods for computing numerical answers to various mathematical problems and to appreciate the dangers that can arise when we attempt to actually compute solutions.

The problems we will consider are: 1. Solving equations. 2. Computing derivatives and integrals. 3. Solving systems of linear equations. 4. Splines, one way of generating a smooth curve to fit given data.

It is not right to say that we compute solutions. In almost all instances we will actually *approximate* the solution. Thus, an important part of this course will be ways to estimate and manage the error that arises in computations.

1.1 Types of error

In this course, we will consider two types of error: truncation error and rounding error. I will try to define these below.

Truncation error is error that arises in replacing an infinite by a finite process that can be computed. Thus, if we approximate an integral by a Riemann sum, a derivative by a difference quotient or a series by a partial sum, we have commmitted truncation error.

Rounding error is error that arises in doing arithmetic on a computer. Memory is cheap, but we still cannot store an infinite binary expansion on a computer. Thus, additional errors are committed as the computer computes the solution to the truncated problem. When doing simple arithmetic, such errors are extremely small. However, if we are not clever, it is possible to carry out computations which magnify this error and lead to answers that are not useful.

Example. To give a simple example of these errors, consider the series which defines e,

$$e = \sum_{n=0}^{\infty} \frac{1}{n!}.$$

It would take a long time to compute this series. Thus, we might decide to approximate by a partial sum

$$S_N = \sum_{n=0}^{N} \frac{1}{n!}.$$

The truncation error is the difference $|e - S_N|$.

If we attempt to compute S_N, most of the terms in the series will not be represented exactly in the computer and thus there will be small errors in computing each term. Additional errors will be made when add the terms and round to the precision of the computer. Rounding is an attempt to minimize error, but it cannot eliminate error.

1.2 Ways to measure error

We suppose that we have a number α and a is a computed approximation to α. The *absolute error* of the approximation is the quantity

$$|\alpha - a|$$

while the *relative error* is the quantity

$$\frac{|\alpha - a|}{|\alpha|}.$$

Relative error is typically more useful–if we know the distance from the earth to the sun with an absolute error of one mile, this is very different from knowing the distance from Funkhouser to the Office Tower with an error of one mile.

1.3 Taylor's theorem-review

Suppose we want to approximate a function by a simpler function. One way to do this is to use a polynomial. Usually, this represents progress since we may compute values of a polynomial using arithmetic while evaluating a function such as sin requires special instructions on the computer.

Approximating a function can mean different things: we can ask that the polynomial agree with the function at specified points in the domain, or we can ask that the polynomial agree with the function very well at one point. Taylor polynomials are the answer to the second question.

We say that f and a polynomial p agree to order n at c if for some constant M all x with $|x - c| < 1$ we have

$$|f(x) - p(x)| \leq M|x - c|^{n+1}. \tag{1}$$

We will prove the following simple fact.

Proposition 1 *If f satisfies (1), f has n continuous derivatives and $p(x)$ is an nth degree polynomila, then*

$$p(x) = \sum_{k=0}^{n} \frac{f^{(k)}(c)}{k!}(x - c)^k.$$

Proof. According to 1, we have

$$\lim_{n\to\infty} \frac{f(x) - p(x)}{(x - c)^k} = 0, \qquad k = 0, \ldots, n.$$

If we apply L'Hopital's rule k times, we obtain,

$$f^{(k)}(c) - p^{(k)}(c) = 0.$$

But if p is of the form $p(x) = \sum_{k=0}^{n} a_k(x - c)^k$, then $p^{(k)}(c) = k!a_k$. Solving the equation $f^{(k)}(c) = k!a_k$ for a_k gives the coefficients of p. ∎