MATH 3Q03 - Numerical Explorations

Sang Woo Park

January 9, 2017

Course Outline

- Website: https://ms.mcmaster.ca/ bprotas/MATH3Q03/
- Textbook: *Numerical Mathematics*
- Supplemental references: *Approximation Theory and Approximation Practice*
- Five assignments

Contents

1 Introduction and Review

Definition 1.1 (Numerical analysis)**.** *Numerical analysis is a branch of applied mathematics, studying computational algorithms to solve problems in calculus and analysis (e.g., differentiation, integration, etc.). It focuses on the transforming continuous (infinite dimensional) problems to finite dimensional representations.*

Definition 1.2 (Scientific computing)**.** *Scientific computing focuses on efficient implementation of numerical algorithms on computers; modern computers are able to store data, compare data, and add, subtract, multiply, and divide. It aims to transform problems in calculus and analysis into problems in algebra.*

1.1 Errors in numerical computation

All numerical methods are *approximate*. In most practical problems, solutions need only possess a certain accuracy. Even when a solution is available analytically, its numerical value may only determine with a finite precision.

Numerical analysis seeks to understand and characterize the structure of errors so that approximations can be made arbitrarily accurate by using more computational time and memory. So we are equally interested in efficiency and complexity (number of operations required) of numerical methods (given the size of the problem and the desired accuracy of the solution).

Definition 1.3 (Types of error)**.** *There are four types of errors:*

1. Model error *from simplifying model by making assumptions.*
2. Measurement error *from determining parameters or data with imperfect accuracy.*
3. Truncation error *from approximating infinite sums by finite sums.*
4. Roundoff error *from representing real numbers with a finite number of bits (floating point numbers).*

Example 1.1.1 (Truncation error)**.** In theory, we would have to use infinitely many operation to compute e^x:

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

However, we may approximate this sum by using a finite sum,

$$e^x \approx \sum_{n=0}^{k} \frac{x^n}{n!},$$

yielding a truncation error of $\varepsilon = \sum_{n=k+1}^{\infty} (x^n/n!)$. Note that $\varepsilon \to 0$ as $k \to \infty$.

Example 1.1.2 (Roundoff error)**.** Let a and b be adjacent floating point numbers. Real numbers $x \in (a, b)$ will be represented necessarily either by a or by b (rounding).

Definition 1.4 (Finite arithmetic precision errors)**.**

- *Rounding (real numbers are rounded towards the nearest floating-point number)*
- *Underflow (computer cannot distinguish a number from 0)*
- *Overflow (computer cannot distinguish a number from infinity)*

Note that there exists a realmin *and* realmax*, which are smallest and largest floating point numbers that a computer can represent respectively.*

Example 1.1.3. We know that $2 - (\sqrt{2})^2 = 0$. However, what happens if we type this on `MATLAB`?

```
2 - (sqrt(2))^2

% ans = -4.4409e-16

eps

% ans = 2.2204e-16

eps(1.0)

% ans = 2.2204e-16
```

We get a value that is different from 0. This phenomenon arises from a roundoff error. Since $\sqrt{2}$ is an irrational number, `MATLAB` rounds it to cloesest floating point value.

Note that increasing the number causes `eps` to increase as well. we can also use `realmin` and `realmax` to check the values of realmin and realmax.

Remark. Floating point arithmetic is not always associative. In other words,

$$(a + b) + c \neq a + (b + c)$$

Both truncation and round-off errors can propagate and accumulate at various stages of computations.

Definition 1.5. *The* total computational error *is the sum resulting from the interaction of the* component errrs*.*

Definition 1.6. $Absolute\ error = |true\ value - approx\ value|$

Definition 1.7. $Relativeerror = \frac{Absolute\ error}{|True\ value|}$

Example 1.1.4. Suppose we have a function, $y = f(x)$, we want to evaluate. Then, $\hat{x}$ is the approximate input (subject to round-off errors) and $\hat{f}$ is the approximate operation (subject to truncation errors).

Is there a way to find out how much error there will be? In other words, we want to understand $\hat{f}(\hat{x}) - f(x)$. In order to do so, we start by manipulating the equation by adding and subtracting $f(\hat{x})$ and applying triangle inequality ($|a + b| \leq |a| + |b|$)

$$\begin{aligned}
\left|\hat{f}(\hat{x}) - f(x)\right| &= \left|\left[\hat{f}(\hat{x}) - f(\hat{x})\right] + \left[f(\hat{x}) - f(x)\right|\right. \\
&\leq \underbrace{|\hat{f}(\hat{x}) - f(\hat{x})|}_{\text{truncation error}} + \underbrace{|f(\hat{x}) - f(x)|}_{\text{round-off errors}}
\end{aligned}$$

Definition 1.8. Conduction number *represents sensitivity of a mathematical procedure to input errors:*

$$\begin{aligned}
\kappa &= \frac{\textit{rel. output error}}{\textit{rel. input error}} \\
&= \frac{|\hat{y} - y|}{|y|} \Bigg/ \frac{|\hat{x} - x|}{|x|}
\end{aligned}$$

Remark. Interval arithmetics explicitly accounts for round-off errors

2 Interpolation

3 Approximation

4 Numerical Differentiation and Integration

5 Special Topics