Introduction to Numerical Analysis I

Yimin Zhong

Feb 16, 2024

CONTENTS

This repository hosts the course material for Math 5630/6630: **Introduction to Numerical Analysis I** at Auburn University. The course will use the textbook written by A. Quarteroni, R. Saaco, and F. Saleri, *Numerical Mathematics*, Second Edition, Springer, New York, 2007.

The course will cover the following topics:

- Floating point arithmetic
- Root finding
- Interpolation and approximation
- Numerical differentiation and integration
- Numerical solutions of ordinary differential equations

This course is an introduction to numerical analysis. It is designed to provide a solid foundation in numerical methods and their applications. Both fundamental theory and programming are required throughout the course. The prerequisites for the theory part are

- Linear Differential Equations (2650)
- Topics in Linear Algebra (2660)

Note: The default programming language for this class is `Python` and `MATLAB`, the other script languages such as `R`, `Julia` are also supported.

CHAPTER

ONE

FLOATING POINT ARITHMETIC

In this chapter, we will introduce some basics on the real number system for modern computers and discuss the arithmetic operations of the number system.

1.1 Representation of Real Numbers

Any nonzero real number $x \in \mathbb{R}$ can be accurately represented with an infinite sequence of digits. This can be understood as the consequence that rational numbers are **dense** on any interval.

What does "dense" mean?

Dense means that between any two distinct real numbers, there is always a rational number. It is a fundamental property of the real number system.

Therefore, with the binary representation, we can write

$$x = \pm(0.d_1 d_2 d_3 \dots d_{t-1} d_t d_{t+1} \dots) \times 2^e,$$

where e is an integer exponent and $d_1 = 1$, the other binary digits $d_i \in \{0, 1\}$. The mantissa part

$$0.d_1 d_2 d_3 \cdots = \frac{d_1}{2} + \frac{d_2}{2^2} + \frac{d_3}{2^3} + \cdots.$$

Note: In order to guarantee the uniqueness of the above representation, we need further assumption that there exists an infinite subset $S \subset \mathbb{N}$ that $d_j \neq 1$ for $j \in S$. For example, under binary representation

$$0.111 \cdots = (0.1) \times 2^1,$$

then we will take the latter representation.

1.2 Floating Point Numbers

The floating point numbers generally refer to a set of real numbers with **finite** mantissa length. More precisely, we consider the set of real numbers $\mathbb{F} = \mathbb{F}(t, e_{\text{min}}, e_{\text{max}}) \subset \mathbb{R}$ that

$$\mathbb{F} := \{x \in \mathbb{R} \mid x = \pm(0.d_1 d_2 d_3 \dots d_{t-1} d_t) \times 2^e, d_1 = 1, e_{\text{min}} \le e \le e_{\text{max}}\} \cup \{0\}.$$

It can be seen that there are only finite numbers in $\mathbb{F}$ with the smallest positive element $x_{\text{min}} = 2^{e_{\text{min}}-1}$ and the largest element $x_{\text{max}} = (1 - 2^{-t}) \times 2^{e_{\text{max}}}$. Therefore

$$\mathbb{F} \subset \overline{\mathbb{F}} := \{x \in \mathbb{R} \mid x_{\text{min}} \le |x| \le x_{\text{max}}\} \cup \{0\}.$$

Note: The elements in $\mathbb{F}$ are called normalized. If we allow $d_1 = 0$ in the definition of $\mathbb{F}$, then the numbers in the set are called denormalized.

Theorem 1 (Distribution of Floating Numbers)

For any $e_{\text{min}} \le e \le e_{\text{max}}$, the distribution of the floating point number system $\mathbb{F}$ on interval $[2^{e-1}, 2^e]$ is equidistant with distances of length 2^{e-t}.

Proof. For any $x \in \mathbb{F} \cap [2^{e-1}, 2^e]$, it can be represented by

$$x = (0.d_1 d_2 \dots d_t) \times 2^e$$

where $d_1 = 1$. The mantissa is equidistantly distributed with distance 2^{-t}, therefore the floating point numbers are equidistantly distributed with distances of length 2^{e-t}.

To understand the approximation to real numbers by the floating point number system $\mathbb{F}$, it is important to consider the maximal relative distance between the numbers in $\overline{\mathbb{F}}$ and their respective closest element in $\mathbb{F}$, which is the following quantity:

$$\max_{x \in \overline{\mathbb{F}}, x \neq 0} \min_{z \in \mathbb{F}} \frac{|z - x|}{|x|}.$$

The following holds:

Theorem 2 (Machine Precision)

$$\max_{x \in \overline{\mathbb{F}}, x \neq 0} \min_{z \in \mathbb{F}} \frac{|z - x|}{|x|} \le 2^{-t}.$$

The number $\mathrm{u} := 2^{-t}$ is also called rounding unit or machine precision.

The definition of machine precision has two versions. The formal definition 2^{-t} appears mostly in research literature and numerical packages (LAPACK). In modern programming languages like `Python`, `MATLAB`, `C++`, the machine precision is defined by 2^{t-1} instead. The meaning is the **difference** between one and the next floating point number.

In other words, two versions of machine precision are corresponding to different rounding strategies. For the former, the rounding strategy is to round to the nearest floating point number, while for the latter, the rounding strategy is to round-by-chop.

In practice, it is not necessary to distinguish the two versions of machine precision, since the difference is only a factor of 2.

Proof. Without loss of generality, we only need to consider the positive numbers in $\overline{\mathbb{F}}$, then one can represent any nonzero $x \in [x_{\text{min}}, x_{\text{max}}]$ by

$$x = (0.d_1 d_2 \dots d_t \dots) \times 2^e \in [2^{e-1}, 2^e].$$

Since the floating point numbers are equidistantly distributed on $[2^{e-1}, 2^e]$ from *Theorem 1*, one can find $z^* \in \mathbb{F}$ such that

$$|z^* - x| \le \frac{1}{2} 2^{e-t},$$

therefore

$$\frac{|z^* - x|}{|x|} \le \frac{1}{2} 2^{e-t} \frac{1}{2^{e-1}} = 2^{-t}.$$

Note: On modern computers, the following two floating point number systems

$$\mathbb{F}_{32} := \mathbb{F}(24, -125, 128), \quad \mathbb{F}_{64} := \mathbb{F}(53, -1021, 1024)$$

are supported, they are often called single precision and double precision, respectively.

IEEE754 Standard

The IEEE754 standard for floating point arithmetic is slightly different from the note. For instance, without **underflow** (all exponent bits are zeros), the standard `float32` is represented as

$$\pm 1.d_1 d_2 \cdots d_{23} \times 2^e$$

where the sign occupies 1 bit, the mantissa occupies 23 bits, and the exponent occupies 8 bits, ranging from -126 to 127 instead. The floating number is stored as

$$\text{sign} \mid e_7 e_6 \cdots e_0 \mid d_1 d_2 \cdots d_{23}$$

and $e = \sum_{j=0}^{7} 2^j e_j - 127$.

1.3 Rounding

The rounding operation fl is to map any real numbers of $\overline{\mathbb{F}}$ into the floating point number system $\mathbb{F}$ with smallest error. Such rounding operation can be written out explicitly, let $x = \pm(0.d_1 d_2 \dots d_t d_{t+1} \dots) \times 2^e$, then

$$\text{fl}(x) = \begin{cases} \pm(0.d_1 d_2 \dots d_t) \times 2^e & \text{if } d_{t+1} = 0, \\ \pm(0.d_1 d_2 \dots d_t + 2^{-t}) \times 2^e & \text{if } d_{t+1} = 1. \end{cases}$$

It is clear that rounding fl is monotone and idempotent, which means

- $x \le y \Rightarrow \text{fl}(x) \le \text{fl}(y)$.
- $\text{fl}(z) = z$ if $z \in \mathbb{F}$.

Theorem 3

For any $x \in \overline{\mathbb{F}}$, $|\text{fl}(x) - x| = \min_{z \in \mathbb{F}} |z - x|$. If $x \neq 0$, then

$$\frac{|\text{fl}(x) - x|}{|x|} \le \mathrm{u} = 2^{-t}.$$

Proof. The special case that $x = 0$ is trivial, we only consider $x \in [x_{\text{min}}, x_{\text{max}}]$, it can be seen that

$$|\text{fl}(x) - x| = |(0.d_1 d_2 \dots \tilde{d}_t) - (0.d_1 d_2 \dots d_t d_{t+1} \dots)| \times 2^e \le 2^{-(t+1)} \times 2^e,$$

where $\tilde{d}_t$ is the rounding bit, therefore

$$\frac{|\text{fl}(x) - x|}{|x|} \le \frac{2^{e-(t+1)}}{2^{e-1}} = 2^{-t}.$$

Corollary 1

For any $x \in \overline{\mathbb{F}}$, $\text{fl}(x) = x(1 + \delta)$ with $|\delta| \le \mathrm{u}$.

PROOF INDEX