A FIRST COURSE

IN

NUMERICAL ANALYSIS

A FIRST COURSE

IN

NUMERICAL ANALYSIS

MAT4001 Notebook

Prof. Yutian Li

The Chinese University of Hongkong, Shenzhen

香港中文大學（深圳）

The Chinese University of Hong Kong, Shenzhen

Contents

Acknowledgments

This book is from the MAT4001 in fall semester, 2018.

CUHK(SZ)

Notations and Conventions

$\mathbb{R}^n$	n-dimensional real space
$\mathbb{C}^n$	n-dimensional complex space
$\mathbb{R}^{m\times n}$	set of all $m \times n$ real-valued matrices
$\mathbb{C}^{m\times n}$	set of all $m \times n$ complex-valued matrices
x_i	ith entry of column vector $\boldsymbol{x}$
a_{ij}	(i,j)th entry of matrix $\boldsymbol{A}$
$\boldsymbol{a}_i$	ith column of matrix $\boldsymbol{A}$
$\boldsymbol{a}_i^{\mathrm{T}}$	ith row of matrix $\boldsymbol{A}$
$\mathbb{S}^n$	set of all $n \times n$ real symmetric matrices, i.e., $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $a_{ij} = a_{ji}$ for all i, j
$\mathbb{H}^n$	set of all $n \times n$ complex Hermitian matrices, i.e., $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ and $\bar{a}_{ij} = a_{ji}$ for all i, j
$\boldsymbol{A}^{\mathrm{T}}$	transpose of $\boldsymbol{A}$, i.e, $\boldsymbol{B} = \boldsymbol{A}^{\mathrm{T}}$ means $b_{ji} = a_{ij}$ for all i, j
$\boldsymbol{A}^{\mathrm{H}}$	Hermitian transpose of $\boldsymbol{A}$, i.e, $\boldsymbol{B} = \boldsymbol{A}^{\mathrm{H}}$ means $b_{ji} = \bar{a}_{ij}$ for all i, j
$\mathrm{trace}(\boldsymbol{A})$	sum of diagonal entries of square matrix $\boldsymbol{A}$
$\mathbf{1}$	A vector with all 1 entries
$\mathbf{0}$	either a vector of all zeros, or a matrix of all zeros
$\boldsymbol{e}_i$	a unit vector with the nonzero element at the ith entry
$\mathcal{C}(\boldsymbol{A})$	the column space of $\boldsymbol{A}$
$\mathcal{R}(\boldsymbol{A})$	the row space of $\boldsymbol{A}$
$\mathcal{N}(\boldsymbol{A})$	the null space of $\boldsymbol{A}$
$\mathrm{Proj}_{\mathcal{M}}(\boldsymbol{A})$	the projection of $\boldsymbol{A}$ onto the set $\mathcal{M}$

Chapter 1

Week1

1.1. Monday

1.1.1. Introduction to Numerical Analysis

Solving Nonlinear Equations. For example, we want to solve a nonlinear equation $w(1)$ with w to be the LambertW function:

$$we^w = 1.$$

This topic will be taught in chapter 2.

Interpolation. Given a list of data points, our aim is to recover/approximate the origin function over a function class, i.e., piecewise linear functions or polynomials. This topic will be taught in chapter 3.

Numerical Integration. The cdf of the standard normal distribution is given by:

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{t^2}{2}\right) \mathrm{d}t$$

To approximate the values for $N(x)$, we need to apply a numerical method (quadrature rules) to evaluate. This topic will be taught in chapter 4.

Solving Linear Systems. To find the solutions of a linear system of equations

$$\boldsymbol{Ax} = \boldsymbol{b},$$

e.g., when we use *finite difference* method to solve a differential equation, it is necessary to apply some numerical method to solve it in computer. This topic will be taught in chapter 5.

Least Squares. If we have more time, we will teach how to fit a set of data points by a function from a function class.

1.1.2. Basic Concepts

Definition 1.1 [Truncation Error] The error made by numerical algorithms that arises from taking finite number of steps in computation ∎

For example, consider the taylor's theorem

$$f(x) = P_n(x) + R_n(x)$$

where

$$P_n(x) = f(x_0) + f'(x_0)(x - x_0) + \cdots + \frac{f^{(n))}(x_0)}{n!}(x - x_0)^n$$
$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)^{n+1}$$

If we use $P_n(x)$ to approximate $f(x)$, the error $R_n(x)$ is called the **truncation error**.

Definition 1.2 [Round-off Error] The error produced when a computer is used to perform real number calculations, i.e., the computer only gives approximate value for some real numebers. ∎

For example, for numbers $\frac{1}{3}, \pi, \sqrt{2}$, they cannot be represented exactly in the computation by a computer.

Such errors are invertible, but we can try to minimize the negative impact of these errors by rewriting the formula that are wish to compute.

Definition 1.3 [Binary Floating Point Number] A 64-bit (binary digit) representation is used for a real number:

$$(-1)^s 2^{c-1023}(1+f)$$

where

- The first bit s is said to be a **sign indicator**;
- The 11-bit exponent c is called the **characteristic**
- The 52-bit binray f is called the **mantissa**.

■

The **underflow** is given by (??):

$$2^{-1022} \cdot (1+0) \approx 0.22251 \times 10^{-307}$$

The **overflow** is given by (??):

$$2^{1023} \cdot (2-2^{-52}) \approx 0.17977 \times 10^{309}$$

Definition 1.4 [Decimal floating-point number] Any positive real number within the numerical range of the machine can be converted into the form

$$y = 0.d_1 d_2 \cdots d_k \cdots \times 10^n$$

The **floating-point** form of y, denoted by $fl(y)$, is obtained by terminating the mantissa of y at k decimal digits. ■

There are two common ways to perform such termination.

Chopping. Simply chops off the digits $d_{k+1}d_{k+2}\ldots$, which produces the floating-point of the form

$$fl(y) = 0.d_1d_2\cdots d_k \times 10^k$$

Rounding. Adds $5 \times 10^{n-(k+1)}$ to y and then chops the result to obtain a number of the form

$$fl(y) = 0.\delta_1\delta_2\cdots\delta_k \times 10^n.$$

Or equivalently, for rounding, we add 1 to d_k to obtain $fl(y)$ for $d_{k+1} \geq 5$ (round up); otherwise we simply chop off at k digits (round down).

Round-Off Error. The errors that results from replacomg a number with its floating-point form is called **round-off error**.

■ **Example 1.1** Determine the five-digit chopping and rounding values of π:

- Firstly we write π in normalized decimal form:

$$\pi = 0.3141592\cdots \times 10^1$$

- The floating-point form of π using five-digit chopping is

$$fl(\pi) = 0.31415 \times 10^1 = 3.1415$$

- The floating-point form of π using five-digit rounding is

$$fl(\pi) = (0.31415 + 0.00001) \times 10^1 = 3.1416$$

■

Definition 1.5 Suppose p^* is an approximation to p. The acutal error is $p - p^*$, the absolute error is $|p - p^*|$, and the relative error is $\frac{|p-p^*|}{|p|}$ provided that $p \neq 0$. ■

Definition 1.6 Suppose p^* is an approximation to p, then p^* is said to approximate p to t **significant digits** if t is the largest nonnegative integer for which

$$\frac{|p-p^*|}{|p|} \leq 5 \times 10^{-t}.$$

■

1.1.3. Examples in Numerical Calculations

1.1.3.1. Inaccuracy Issues

Inaccuracy of floating-point numbers. The relative error for floating-point numbers is

$$\left|\frac{y-fl(y)}{y}\right|.$$

For k-digit chopping arithmetic, its relative error is give by:

$$\left|\frac{y-fl(y)}{y}\right| = \left|\frac{0.d_{k+1}d_{k+2}\cdots \times 10^{n-k}}{0.d_1d_2\cdots \times 10^n}\right| \leq \frac{1}{0.1} \times 10^{-k} = 10^{-k+1}$$

Similarly, the bound for the relative error when using k-digit rounding arithmetic is $0.5 \times 10^{-k+1}$.

Inaccuray of finite-digit arithmetics. We assume that the finite-digit arithmetics are given by:

$$x$$

1.1.3.2. Reduce the loss of accuracy due to round-off error

Reformulate the problem. One common error during calculations is the cancelation of significant digits due to the **subtraction of nearly equal numbers**. Reformulation can avoid such a problem.

■ **Example 1.2** The two roots of $ax^2 + bx + c = 0$ are given by:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

For the equation $x^2 + 62.10x + 1 = 0$, the two roots are

$$x_1 = 0.01610723, \quad x_2 = 62.08390.$$

In this equation, the numerator of x_1 is a subtraction of two nearly equal numbers. If using 4-digit rounding we derive

$$fl(x_1) = \frac{-62.10 + 62.06}{2.000} = -0.02000 \implies \text{relative error} = 0.24 \times 10^0.$$

To improve the accuracy of calculation, we should change the formula by **rationalizing the numerator**:

$$x_1 = \frac{-2c}{b + \sqrt{b^2 - 4ac}}$$

It follows that

$$fl(x_1) = \frac{-2.000}{62.10 + 62.06} = -0.01610 \implies \text{relative error} = 6.2 \times 10^{-4}.$$

■

Rewrite polynomials into nested form. For example, evaluating $f(x) = x^3 - 6.1x^2 + 3.2x + 1.5$ with $x = 4.71$ directly will result in large relative errors. To improve the calculation, we change the polymomial to the following nested form:

$$f(x) = ((x - 6.1)x + 3.2)x + 1.5,$$

which results in small relative erros.

Polynomials should always be expressed in nested form before evaluation, since this form will minimize the number of arithmetic calculations

Avoid large numbers eat small numbers. We should add those numbers with small magnitude first when we do the summation

$$fl\left(\sum_{i=1}^{n} fl(x_i)\right).$$

■ **Example 1.3** For $x_1 = 100.5$ and $x_i = 0.01, i = 2,\ldots,101$, using 4-digit arithmetic, we have $fl(x_1) = 100.5$ and $fl(x_i) = 0.0100, i = 2,\ldots,101$.

- For doing the summation $s = fl\left(\sum_{i=1}^{101} fl(x_i)\right)$, since $100.5 + 0.0100 = 100.5100 = 100.5$, finally we have $s = 100.5$
- If we define $x_i = 0.01, i = 1,\ldots,100$ and $x_{101} = 100.5$, then we have $s = 101.5$, which is the true value of the summation.

■

1.1.4. Convergence and Stability

Definition 1.7 [Convergence] Suppose a sequence $\{\beta_n\}_{n=1}^{\infty}$ is known to converge to zero, and $\{\alpha_n\}_{n=1}^{\infty}$ converges to a number α. If there exists $K > 0$ such that

$$|\alpha_n - \alpha| \leq K|\beta_n| \quad \text{for large } n,$$

then $\{\alpha_n\}_{n=1}^{\infty}$ is said to converge to α with rate of convergence $\mathcal{O}(\beta_n)$, which is denoted as

$$\alpha_n = \alpha + \mathcal{O}(\beta_n).$$

■