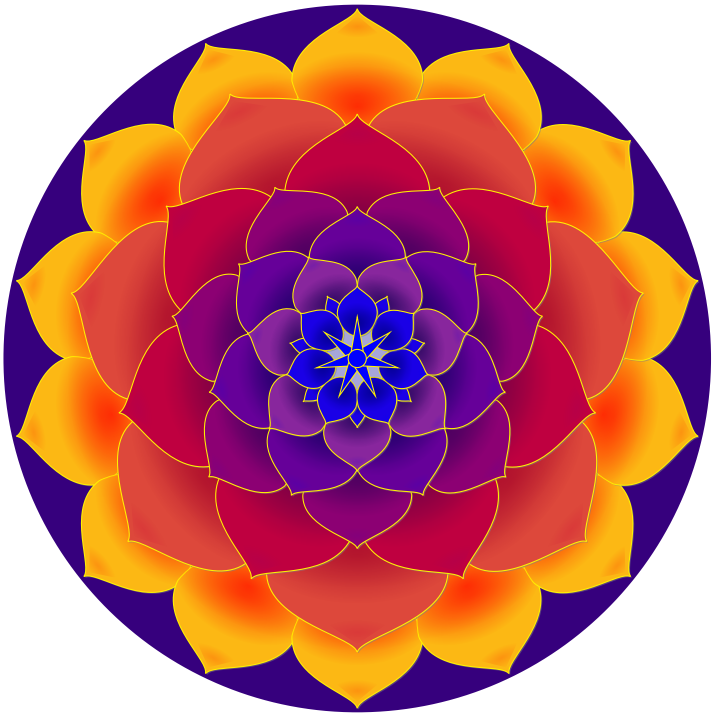

Introduction to Numerical Analysis

Mathematics With Computer Science

Author: Zewan LYU(999019888)

Date: 2025-10-15

Contents

Chapter 1

Introduction, Big-O-notation. Horner's method. [1.1, 2.6]

Motivation:

- Sometimes problems require too much time, effort, etc. to be praticallly solved wtih a computer.
- Some problems cannot be solved analytically.
- For applications, numerical solutions are often sufficient.

Example 1.0.1. *Can you solve :*

1. $\sin(0.67)$
2. $\int_0^1 e^{-x^2} \mathrm{d}x$
3. $x^2 - \sin(x) - 1 = 0, x = ?$

Numerical problems lead to new mathematical questions.

Example 1.0.2. *Suppose $Ax = b, A \in \mathbb{R}^{n\times n}, det(A) \neq 0$, and A is symmetric. Find $det(A)$.*

Approach I: Use Sarrus' rule $det(A) = \sum_{\sigma\in S_n} sgn(\sigma) \prod_{i=1}^n a_{i,\sigma(i)}$. For each permutation σ, we have $n!$ permutations. So we have $n \cdot n!$ product operations. Furthermore, we have $n! - 1$ additions, meaning that the method requires $n(n!) + n! - 1$ operations. This approach works fine for small n, but not for large n.

Approach II: Use the diagonalization method. Since A is symmetric, we can find an orthogonal matrix Q such that $Q^T AQ = D$, where D is a diagonal matrix. Then, we have $det(A) = det(Q)det(D) = det(D)$. We can compute $det(A)$ in $cn^3 + n - 1$ operations where c is a constant.

Definition 1.0.1. *Consider sequences (x_n) and (α_n), where $n = 0, 1, 2, \ldots$. We say that (x_n) is in $O(\alpha_n)$ if there exist constants C, N such that*

$$|x_n| \leq C|\alpha_n|, \quad \forall n \geq N$$

Example 1.0.3. $\frac{n+1}{n^2}$ *is in $O(\frac{1}{n})$. $Cn^3 + n - 1$ is in $O(n^3)$. $n(n!)$ is in $O(n(n!))$.*

Remark It is also true that $Cn^3 + n - 1$ is in $O(n(n!))$.

Definition 1.0.2. *Let $(x_n), (\alpha_n)$ be sequences, where $n = 0, 1, 2, \ldots$. We say that (x_n) is in $\Theta(\alpha_n)$ if there exist constants C_1, C_2, N such that*

$$C_1|\alpha_n| \leq |x_n| \leq C_2|\alpha_n|, \quad \forall n \geq N$$

Example 1.0.4. *$n(n!) \leq n(n!) + n! - 1 \leq 2n(n!)$ for $n \geq 1$. So $n(n!) + n! - 1$ is in $\Theta(n(n!))$.*

Example 1.0.5. *$Cn^3 \leq Cn^3 + n - 1 \leq (C + 1)n^3$ for $n \geq 1$. So $Cn^3 + n - 1$ is in $\Theta(n^3)$.*

Question How many operations are required to evaluate a polynomial $p(z) = a_n z^n + a_{n-1}z^{n-1} + \cdots + a_1 z + a_0$ at a point z_0 where $a_n, a_{n-1}, \ldots, a_1, a_0, z_0 \in \mathbb{R}$?

The simplest approach computes $a_k z^k$ by using k multiplications, and then sums them up. This requires $n + (n-1) + \cdots + 1 + n = \frac{n(n+1)}{2} + n$ which is in $\Theta(n^2)$.

The Horner's method is based on the remaider theorem, which reduce complexity of evaluation of a polynomial. It only requires $\Theta(n)$ operations.

Theorem 1.0.1 (Horner's Method)**.** *Let $p(z)$ be a polynomial of degree n, with real coefficients, and $z_0 \in \mathbb{R}$. Then there exists $r \in \mathbb{R}$ and a polynomial $q(z)$ of degree $n-1$ such that*

$$p(z) = r + (z - z_0)q(z)$$

Proof. Let $p(z) = a_n z^n + a_{n-1}z^{n-1} + \cdots + a_1 z + a_0$. Our goal is to find coefficients $b_{n-1}, b_{n-2}, \ldots, b_1, b_0$ and a constant r such that

$$p(z) = r + (z - z_0)(b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \cdots + b_1 z + b_0)$$

Equating coefficients of like powers of z on both sides, we get a system of equations:

$$\begin{aligned} a_n &= b_{n-1} \\ a_{n-1} &= b_{n-2} - z_0 b_{n-1} \\ a_{n-2} &= b_{n-3} - z_0 b_{n-2} \\ &\vdots \\ a_1 &= b_0 - z_0 b_1 \\ a_0 &= r - z_0 b_0 \end{aligned}$$

Solving the equations from the top the the bottom recursively leads to a unique solution:

$$\begin{aligned} b_{n-1} &= a_n \\ b_{n-2} &= a_{n-1} + z_0 b_{n-1} \\ b_{n-3} &= a_{n-2} + z_0 b_{n-2} \\ &\vdots \\ b_0 &= a_1 + z_0 b_1 \\ r &= a_0 + z_0 b_0 \end{aligned}$$

■

Remark. It is convenient to write $b_{-1} = r$ which leads to equations

$$b_{n-1} = a_n$$

$$a_k = b_{k-1} - z_0 b_k, \quad k = n-1, \ldots, 0$$

These equations can be graphically represented as:

$$\begin{array}{c|cccccc} & a_n & a_{n-1} & a_{n-2} & \cdots & a_1 & a_0 \\ z_0 & & -z_0 b_{n-1} & -z_0 b_{n-2} & \cdots & -z_0 b_1 & -z_0 b_0 \\ \hline & b_{n-1} & b_{n-2} & b_{n-3} & \cdots & b_0 & b_{-1} \end{array}$$

Remark We have n equations, each of which requires one addition and one multiplication. Overall complexity of computing $p(z_0)$ is in $\Theta(n)$.

Remark The algorithm can be used for finding all roots of $p(z)$ by starting with a root z_0, then writing $p(z) = (z - z_0)q(z)$, and finding z_1 and etc. We will later learn how to find these roots and how to apply the algorithm.

Remark Note that $p'(z) = q(z) + (z - z_0)q'(z)$, so $p'(z_0) = q(z_0)$ which can be evaluated by Horner again.

Example 1.0.6. *Let $p(z) = z^4 - 4z^3 + 7z^2 - 5z - 2$. We compute $p(3)$ and $p'(3)$.*

We set up the following table:

$$\begin{array}{c|ccccc} & 1 & -4 & 7 & -5 & -2 \\ 3 & & 3 & -3 & 12 & 21 \\ \hline & 1 & -1 & 4 & 7 & 19 \end{array}$$

So we have $p(3) = r = 19$. Then we set up the following table to compute $p'(3) = q(3)$:

$$\begin{array}{c|cccc} & 1 & -1 & 4 & 7 \\ & & & & \\ 3 & & 3 & 6 & 30 \\ & & & & \\ \hline & 1 & 2 & 10 & 37 \end{array}$$

So we have $p'(3) = q(3) = 37$.

Chapter 2

Floating point numbers. Convergence. Bisection method. [Section 1.2]

2.0.1 Round-off errors and Computer arithmetic

Remark: Computer don't operate with real numbers!
Representations of decimal numbers(rational).

1. **Fixed Point** Let $d_k = \{0, 1, \cdots, 9\}$ Then we can represent a decimal number as

$$x = (d_5 d_4 d_3 d_2 d_1 d_0.d_{-1} d_{-2} d_{-3} d_{-4} d_{-5}) = \sum_{k=-5}^{5} d_k 10^k \tag{2.1}$$

, where $m < 0 \leq M$. Similarly, for binary numbers:

$$x = b_M b_{M-1} \cdots b_1 b_0.b_{-1} b_{-2} \cdots b_m = \sum_{k=m}^{M} b_k 2^k, \quad b_k \in \{0, 1\} \tag{2.2}$$

This is called the fixed-point representation. However, it's inconvenient for large numbers, or ones with many decimals, so it is rarely used(nowadays).

2. **Floating Point** Idea: represent the number as an integer scaled by an exponent of a fixed base.

$$12.345 = (1)12345 \times 10^{-3} \tag{2.3}$$

where (1) is the sign bit, 12345 is the significand(mantissa), 10 is the base, and -3 is the exponent.

Remark Base 10 corresponds to the "scientific notation used in calculators. Computers(usually) use base 2.

Remark The common type of floating point representation follows the IEEE 754 standard.

   (a) **Single precision** (binary 32, called float in C) 32 bits, mantisa 24 bits(approximately 7 decimal digits).
   (b) **Double precision** (binary 64, called double in C) 64 bits, mantisa 53 bits(approximately 16 decimal digits).

Example 2.0.1. *The first* 33 *bits of* π *in binary is*

$$\pi = 110010010000111111011010101000100 \tag{2.4}$$

To write this as a single precision floating point number, take the 24 bit rounding approximation

$$110010010000111111011011 = (1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + \cdots + 1 \times 2^{-23}) \times 2^1 \approx 3.1415928 \tag{2.5}$$

where the last bit is rounded up and the exponent is 1*.*

Machine epsilon(or machine precision) is an best bound for relative approximation error due to the floating point arithmetic.

The standard values of machine epsilon are

- binary 32(single precision): $\epsilon \approx 1.19 \times 10^{-7}$
- binary 64(double precision): $\epsilon \approx 2.22 \times 10^{-16}$

This value is contained in the module numpy of python (import numpy as np)

```
import numpy as np
print(np.finfo(float).eps)
# 2.220446049250313e-16
print(np.finfo(np.float32).eps)
# 1.1920929e-07
```

Remark

1. "Machine epsilon accuracy" is the ultimate standard for numerical algorithms. (no better accuracy can be expected)
2. Accuracy within 3 or 4 decimals from the machine epsilon is already very sufficient and more than that cannot be expected in practice.

Definition 2.0.1. *Suppose $p*$ is an approximation of $p \in \mathbb{R}$. Then*

1. *The **actual error** is $p - p*$;*
2. *The **absolute error** is $|p - p*|$;*
3. *The **relative error** is $\dfrac{|p - p*|}{|p|}$, provided $p \neq 0$.*

The number $p*$ is said to approximate $p \neq 0$ to t significant digits if t is the largest non-negative integer such that

$$\frac{|p - p*|}{|p|} < 5 \times 10^{-t} \tag{2.6}$$

Definition 2.0.2 (Floating Point Arithmetic)**.** *Denote by $fl(x)$ the floating point approximation of x. Then*

$$x \oplus y = fl(fl(x) + fl(y)) \tag{2.7}$$

$$x \otimes y = fl(fl(x) \cdot fl(y)) \tag{2.8}$$

$$x \ominus y = fl(fl(x) - fl(y)) \tag{2.9}$$

2.0.2 Convergence

Definition 2.0.3 (Rates of Convergence)**.** *Suppose $(\alpha_n), n = 0, 1, 2, \cdots$ is a sequence of real numbers such that $\lim_{n\to\infty} \alpha_n = \alpha \in \mathbb{R}$ and (β_n) is a sequence such that $\lim_{n\to\infty} \beta_n = 0$. If there exist $K > 0, N \geq 0$ such that $|\alpha_n - \alpha| \leq K|\beta_n|$ for all $n \geq N$, then we say that (α_n) converges to α with rate $O(\beta_n)$.*

Remark Usually $\beta_n = \dfrac{1}{n^p}$ for some $p > 0$.

2.0.3 Bisection method

Recall: Intermediate Value Theorem(Bolzano's theorem): Suppose $f : [a, b] \to \mathbb{R}$ is continuous, and there exists K such that either $K \in (f(a), f(b))$ if $f(a) < f(b)$ or $K \in (f(b), f(a))$ if $f(a) > f(b)$. Then there exists $c \in (a, b)$ such that $f(c) = K$. As a corollary, if $f(a)f(b) < 0$, then there exists $c \in (a, b)$ such that $f(c) = 0$.
Application Finding an interval $[a, b]$ that contains a solution of $x - 2^{-x} = 0$. For $x = 0, f(0) = -1 < 0$. For $x = 1, f(1) = 0.5 > 0$. So by the corollary there exists $c \in (0, 1)$ such that $f(c) = 0$.

The following is an more advanced application.

Algorithm 2.0.1 (Bisection Method)**.** *Let $f : [a, b] \to \mathbb{R}$ be continuous and suppose that $f(a)f(b) < 0$. Then*

1. *Set $n = 1, a_1 = a, b_1 = b$.*
2. *Compute $c_n = \dfrac{a_n + b_n}{2}$. If $f(c_n) = 0$, stop. Otherwise, go to step 3.*
3. *If $f(a_n)f(c_n) < 0$, set $a_{n+1} = a_n, b_{n+1} = c_n$. If $f(b_n)f(c_n) < 0$, set $a_{n+1} = c_n, b_{n+1} = b_n$. Increment n by 1 and go to step 2.*

Example 2.0.2. *Let $f(x) = \cos(x) - 2x, [a, b] = [-8, 10]$. Then $f(-8) > 0, f(10) < 0$. So we can apply the bisection method.*

Theorem 2.0.1. *Suppose $f \in C([a,b])$ and $f(a)f(b) < 0$. Then the sequence (c_n) generated by the bisection method approximates the zero $p \in (a,b)$ of f with*

$$|c_n - p| \leq \frac{b-a}{2^n}, n \geq 1 \tag{2.10}$$

Proof Let $n \geq 1$. Then $b_n - a_n = \dfrac{b-a}{2^{n-1}}$ and $p \in (a_n, b_n)$ where a_n, b_n are the endpoints of the interval at the n-th step. Because $c_n = \dfrac{a_n + b_n}{2}$, it follows that

$$|c_n - p| \leq \frac{b_n - a_n}{2} = \frac{b-a}{2^n} \to 0 \tag{2.11}$$

as $n \to \infty$.

■

Remark As $|p_n - p| \leq (b-a)2^{-n}$. So the convergence converges with the rate $O(2^{-n})$. Note: Here we can take $K = b - a$.
Remark To avoid taking off points by graders, we're expected to show(at lest state) the conditions of the theorems we use. For example we need to say $K = b - a$ as the previous proof of the theorem.