Lecture Notes

Kyle Chui

2022-01-04

Contents

1 Lecture 1

The goal of this class is to solve *mathematical* problems with the help of *computers*.

1.1 Chapter Summaries

1. Computations in computers
   - How to store (real) numbers in the computer

     > **Note.** If the number has finitely many digits, then it is simple. What about numbers with infinitely many digits, i.e. $\frac{1}{3}$? We have to truncate or round, and store an approximation with finitely many digits.

   - How to perform computations

     > **Note.** From regular math, we know that $\frac{1}{3} + \frac{1}{3} = \frac{2}{3}$. However, in the computer, due to errors, we have $\frac{1}{3} \oplus \frac{1}{3} = ? \oplus ? = ??$.

   - Errors
2. - Find roots of $f(x) = 0$ using bisection, Newton's method, ...
   - Convergence
   - Convergence order (how fast it converges)
3. Polynomial interpolation
   - Approximate a function $f(x)$ by a polynomial $P(x)$, where $f(x_i) = P(x_i)$ for finitely many x_i
   - *Accuracy* of the polynomial approximations
4. Numerical differentiations and numerical integrations
   - Using the approximations from chapter 3, we can approximate using

     $$f'(x^*) \approx P'(x^*) = \sum_{i=0}^{k} f(x_k)c_k,$$

     and

     $$\int_a^b f(x)\,\mathrm{d}x \approx \int_a^b P(x)\,\mathrm{d}x = \sum_{i=1}^{k} f(\overline{x_k})\overline{c_k}.$$

   - Error analysis

6.7. Solving linear systems of equations
   - Direct methods: Gaussian elimination (computationally expensive)
   - Iterative methods: (faster and cheaper)
   - Solution stability

1.2 Round-off errors and computer arithmetics

There are three kinds of errors:

- Modeling Error: Occurs when we convert a problem from the real world into the mathematical world.
- Method Error: Occurs when we try to solve the mathematical problem numerically.
- Round-off error: Occurs when the computer gives an incorrect result with the correct algorithm (comes from storage and computation).

1.2.1 Storage

Infinite digit real numbers are *stored* as finite digit numbers, using the normalized decimal form of real numbers.

Definition. *Normalized decimal form of a real number*
For any $y \in \mathbb{R}$, we may write

$$y = \pm 0.d_1 d_2 d_3 \dots d_k d_{k+1} \dots \cdot 10^n,$$

where $0 < d_1 \leq 9$, $0 \leq d_i \leq 9$, n are integers. For the particular case where $y = 0$, we write $y = 0.0 \cdot 10^0$.

Definition. *Normalized machine numbers (Floating-point form)*
Any machine number y can be written as

$$y = \pm 0.d_1 d_2 \cdots d_k \cdot 10^n,$$

where $0 < d_1 \leq 9$, $0 \leq d_i \leq 9$, n are integers.

We can think of the storage process as mapping normalized real numbers to normalized machine numbers. We do this via rounding or truncating.

Consider some $y \in \mathbb{R} \setminus \{0\}$.

- Truncating (k-digit truncation of $y = \pm 0.d_1 d_2 d_3 \dots d_k d_{k+1} d_{k+2} \dots \cdot 10^n$) Simply omit the digits from d_{k+1} and onwards, in other words

  $$\text{fl}(\pm 0.d_1 d_2 d_3 \dots d_k d_{k+1} d_{k+2} \dots \cdot 10^n) = \pm 0.d_1 d_2 \dots d_k \cdot 10^n.$$

  Thus we have $y \approx \text{fl}(y)$.

- Rounding (k-digit rounding of $y = \pm 0.d_1 d_2 d_3 \dots d_k d_{k+1} d_{k+2} \dots \cdot 10^n$)

  If $d_{k+1} < 5$, then we drop $d_{k+1} d_{k+2} \dots$ (same with truncating)

  If $d_{k+1} \geq 5$, then add 1 to d_k and drop $d_{k+1} d_{k+2} \dots$

  $$\text{fl}(\pm 0.d_1 d_2 d_3 \dots d_k d_{k+1} d_{k+2} \dots \cdot 10^n) = \pm \delta_1 \delta_2 \dots \delta_k \cdot 10^m.$$

2 Lecture 2

Note. Since we use the notation fl to denote both k-digit truncation as well as k-digit rounding, be sure not to mix the two up.

Definition. *Errors*
Suppose p^* is an approximation of p. Then the *actual error* is $p - p^*$, the *absolute error* is $|p - p^*|$, and the relative error is $\frac{|p-p^*|}{|p|}$, where $p \neq 0$.

Definition. *Significant Digits*
The number p^* is said to approximate p to "t" significant digits if "t" is the largest non-negative integer for which

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

2.1 Computer Arithmetic

Assume that x, y are real numbers, then

$$\begin{aligned} x \oplus y &= \text{fl}(\text{fl}(x) + \text{fl}(y)) \\ x \ominus y &= \text{fl}(\text{fl}(x) - \text{fl}(y)) \\ x \otimes y &= \text{fl}(\text{fl}(x) \cdot \text{fl}(y)) \\ x \oslash y &= \text{fl}(\text{fl}(x) + \text{fl}(y)) \end{aligned}$$

3 Lecture 3

Note. When computing relative error, keep 2 non-zero digits.

If after performing our operation with k-digit chopping, we still have k significant digits, then our operation is pretty good (because it did not lose precision). In general, the $\oplus$ operator will lose at most one significant digit. Unlike addition, the $\ominus$ operator can lose multiple significant digits (when you subtract almost identical numbers).

We see errors introduced in computations:

- Subtraction of almost identical numbers results in loss of significant digits
- More operations $\to$ more errors $\to$ loss of significant digits

To avoid loss of *accuracy*, we can perform some manipulations to avoid scenarios that involve inaccuracies:

Example. *Quadratic formula*
For a given quadratic $ax^2 + bx + c = 0$, we know that a solution is

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

Suppose $b > 0$ and $b^2 \gg 4ac$. Then $b^2 - 4ac \approx b^2$, so $b \approx \sqrt{b^2 - 4ac}$. This can lead to inaccuracies because we are subtracting nearly identical numbers. To avoid this, we can perform some algebra as follows:

$$\begin{aligned} x_1 &= \frac{-b + \sqrt{b^2 - 4ac}}{2a} \\ &= \frac{-b + \sqrt{b^2 - 4ac}}{2a} \cdot \frac{-b - \sqrt{b^2 - 4ac}}{-b - \sqrt{b^2 - 4ac}} \\ &= \frac{b^2 - (b^2 - 4ac)}{2a(-b - \sqrt{b^2 - 4ac})} \\ &= \frac{4ac}{2a(-b - \sqrt{b^2 - 4ac})} \\ &= \frac{2c}{-b - \sqrt{b^2 - 4ac}}. \end{aligned}$$

Since this solution avoids the subtraction of two nearly identical numbers, it is more accurate. For the other root, we are subtracting two negative numbers that are nearly identical, so there is no issue.

4 Lecture 4

4.1 Nested Algorithm

Goal. We want to reduce our error by reducing the number of operations done.

Example. Consider the polynomial

$$p(x) = 4x^4 + 5x^2 + 2x^2 + 5x - 10.$$

We need to perform 4 multiplications for the first term, 3 for the second, etc., for a total of 10 multiplication operations and 4 addition operations. Notice that we can factor out a common x from the first few terms, to get

$$p(x) = x \cdot (4x^3 + 5x^1 + 2x + 5) - 10.$$

Continuing this pattern we get

$$p(x) = x(x(x(4x + 5) + 2) + 5) - 10.$$

We now only have 4 multiplications and 4 additions, which can improve the accuracy of our algorithm.

4.2 Convergence Order

Definition. *Convergence Order*
Assume $\{P_n\}_{n=0}^{\infty}$ converges to P with $P_n \neq P$. If there exists $0 < \lambda < \infty$ and $\alpha > 0$ such that

$$\lim_{n\to\infty} \frac{|P_{n+1} - P|}{|P_n - P|^{\alpha}} = \lambda.$$

Then we say $\{P_n\}_{n=0}^{\infty}$ converges to P with order α.

Note. Larger α implies a faster convergence.

Definition. *Linear and Quadratic Convergence*
If $\alpha = 1$, then $\{P_n\}_{n=0}^{\infty}$ converges to P *linearly*. If $\alpha = 2$, then $\{P_n\}_{n=0}^{\infty}$ converges to P *quadratically*.

5 Lecture 5

5.1 Root Finding (Single Variable)

Given a function $f(x) = 0$ with a root $p \in [a, b]$, we want to find a sequence $p_1, p_2, \ldots$ such that $p_n \to p$. We want to find the root using the Intermediate Value Theorem. Suppose f is a continuous function on $[a, b]$ with $f(a) \cdot f(b) < 0$. Then f has at least one root in (a, b).

Note. This statement holds true because it means that $f(a)$ and $f(b)$ have opposite sign, or that 0 is between $f(a)$ and $f(b)$. Hence there must exist some $p \in (a, b)$ such that $f(p) = 0$.

5.1.1 Bisection Method

Idea. Use binary search to find the root.

You first find the midpoint of your interval (let's call this m), and check whether the sign of $f(m)$ is the same as $f(a)$ or $f(b)$. If the former, then we may "shrink" the interval to $[m, b]$, otherwise we may "shrink" the interval to $[a, m]$. If $f(m) = 0$, then we have found our root. If we perform this algorithm recursively, then we may find a sequence of points that converges to our root.

Note.

- The bisection method *always* converges to a root.
- We stop the iteration if:
  - We reach the maximum iteration number.
  - For some small $\varepsilon > 0$,

$$|p_n - p_{n-1}| < \varepsilon \text{ or } \frac{|p_n - p_{n-1}|}{|p_n|} < \varepsilon \text{ or } |f(p_n)| < \varepsilon.$$

Theorem. Suppose $f \in C[a, b]$ and $f(a) - f(b) < 0$. The Bisection method generates a sequence $\{p_n\}_{n=1}^{\infty}$ approximating a zero p of $f(x)$ with

$$|p_n - p| \leq \frac{b - a}{2^n}, \text{ where } n \geq 1.$$

Proof. For $n \geq 1$, we have $p_n = \frac{a_n + b_n}{2^n}$, $p \in (a_n, b_n)$. Then we know that

$$\begin{aligned} |p_n - p| &\leq \frac{1}{2}(b_n - a_n) \\ &= \frac{1}{2}\left(\frac{1}{2}(b_{n-1} - a_{n-1})\right) \\ &= \cdots \\ &= \underbrace{\frac{1}{2} \cdot \cdots \cdot \frac{1}{2}}_{n \text{ times}}(b_1 - a_1) \\ &= \frac{1}{2^n}(b - a). \end{aligned}$$

□