UNIVERSITY AT ALBANY
STATE UNIVERSITY OF NEW YORK

CSI 401 (Fall 2025)

Numerical Methods

Lecture 2: Asymptotic Notations & Machine Arithmetic

Chong Liu

Department of Computer Science

Aug 25, 2025

Agenda

- Asymptotic notations
- Machine arithmetic
  - Decimal and binary expansions
  - Scientific notation

Asymptotic notations

- Used to compare the growth of two functions $f(x)$ and $g(x)$ as x tends to some limit point x_0.

- Discussion:
  - $f(x) = x^2, g(x) = x$. Which notation shall we use?

To do this, we look at the absolute value of the ratio of the two:

$$\left|\frac{f(x)}{g(x)}\right|. \tag{2.6}$$

The behavior of this can be one of three different things as $x \to x_0$:

- $$\left|\frac{f(x)}{g(x)}\right| \to 0. \tag{2.7}$$

  In this case, we say that $f(x)$ is asymptotically negligible compared to $g(x)$. We also say that $f(x) = o(g(x))$ (i.e., "$f(x)$ is small 'oh' of $g(x)$") as $x \to x_0$.

- $$\left|\frac{f(x)}{g(x)}\right| \tag{2.8}$$

  converges to a positive constant or oscillates but stays bounded.

- $$\left|\frac{f(x)}{g(x)}\right| \to \infty. \tag{2.9}$$

  In this case, we say that $f(x)$ is asymptotically dominant compared to $g(x)$. This implies that $g(x) = o(f(x))$.

Asymptotic notations

- Additionally, we say that $f(x) = O(g(x))$ as $x \to x_0$ if there is some positive constant C such that
- $\left|\frac{f(x)}{g(x)}\right| \leqslant C$.

- Thus, the $O(\cdot)$ notation means that $f(x)$ is **asymptotically upper bounded** by $g(x)$.
- We also say that $f(x) = \Omega(g(x))$ if $g(x) = O(f(x))$.
- We say that $f(x) = \Theta(g(x))$ if $f(x) = O(g(x))$ and $f(x) = \Omega(g(x))$.

Properties of asymptotic notations

Theorem 3.2 (Properties of asymptotic notations). *Let $C > 0$ be some positive constant, and let $x_0 \in \mathbb{R} \cup \{\pm\infty\}$. Then, for any function $g(x)$, as $x \to x_0$,*

$$C \cdot O(g(x)) = O(g(x)) \tag{3.1}$$

$$C \cdot \Theta(g(x)) = \Theta(g(x)) \tag{3.2}$$

$$C \cdot \Omega(g(x)) = \Omega(g(x)) \tag{3.3}$$

$$C \cdot o(g(x)) = o(g(x)). \tag{3.4}$$

Additionally, for any $f(x)$

$$f(x)O(g(x)) = O(f(x)g(x)). \tag{3.5}$$

- The above theorem allows us to simplify expressions asymptotically. E.g., as $x \to \infty$
- $4e^x(\sin(x) + 5) + 3x = \Theta(e^x(\sin(x) + 5)) = \Theta(e^x)$,
- where the first equality is because $x = o(e^x \sin(x))$, and the second equality is because $0 \leqslant |\sin(x)| \leqslant 1$.
- Note that all of this can be verified by looking at ratios of functions, as in the definition of the notation.

Properties of polynomials

Corollary 3.4. *As* $x \to \infty$*, for any fixed* k*, nonzero constant* c_k*, and constants (possibly* 0*)* c_j *for* $j \in \{0, 1, ..., k-1\}$,

$$P(x) = c_k x^k + c_{k-1} x^{k-1} + ... + c_1 x + c_0 = \Theta(x^k). \tag{3.8}$$

As $x \to 0$*, if* j *is the smallest number for which* $c_j \neq 0$,

$$P(x) = \Theta(x^j). \tag{3.9}$$

Let us consider the following question: suppose $f(x) = \Theta(g(x))$. Is it true in general that $f(x) - g(x) = \Theta(g(x))$? **NO**. For instance,

$$f(x) = 3x, g(x) = 3x + 5 \implies f(x) - g(x) = -5 = o(g(x)). \tag{3.10}$$

In-class exercise of asymptotic notations

- Identify the asymptotic relationship: For each pair of functions $f(n)$ and $g(n)$,determine whether
  - $f(n) = O\big(g(n)\big)$,
  - $f(n) = \Omega\big(g(n)\big)$,
  - $f(n) = \Theta\big(g(n)\big)$,
  - or **none of the above**.

1. $f(n) = 3n^2 + 5n$,$g(n) = n^2$
2. $f(n) = \log(n^2)$, $g(n) = \log(n)$
3. $f(n) = n^{1.01}$,$g(n) = n \log n$
4. $f(n) = 2^n$,$g(n) = n^{100}$

Solutions

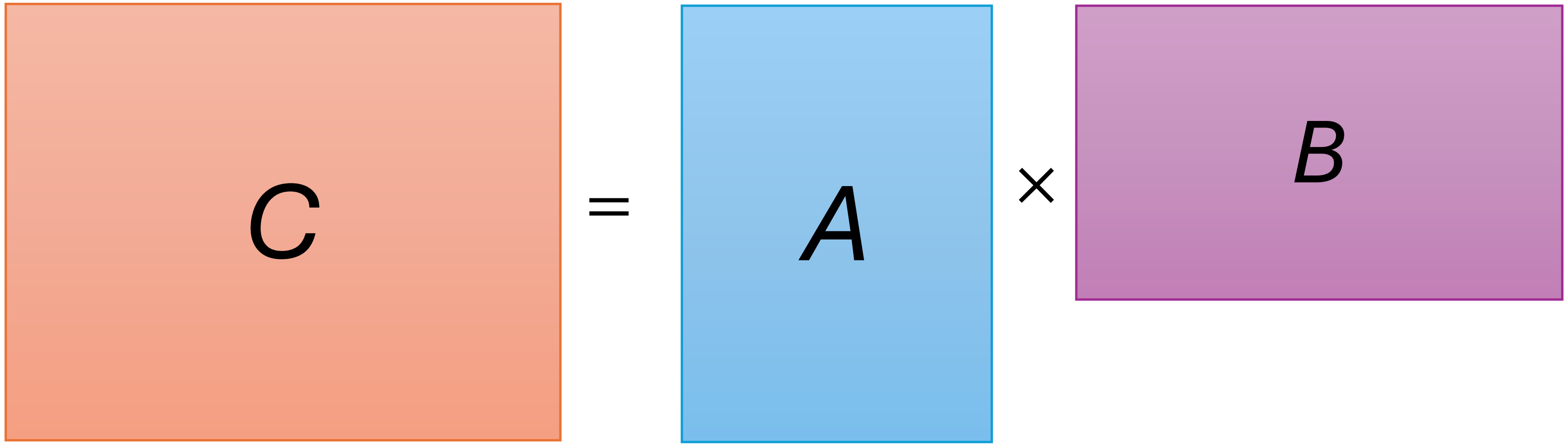
1. $3n^2 + 5n$ vs n^2 :$\Theta(n^2)$
2. $\log(n^2)$ vs $\log n$:$\Theta(\log n)$
3. $n^{1.01}$ vs $n \log n$:
4. vs n^{100} :$\Omega(n^{100})$

In-class exercise of asymptotic notations

- Arrange the following functions in increasing order of asymptotic growth rate (ignore constants):
- $n, \log n\ , n \log n\ , 2^n, n^3, \sqrt{n}, n!$

- **Solution:**
- $\log n < \ \sqrt{n} \ < \ n \ < \ n \log n \ < \ n^3 < \ 2^n \ < n!$

Computational Complexity of matrix Multiplication?



- How many dot product needed? (A is m by n and B is n by p)

Fun fact: complexity of matrix multiplication is still an open problem

- 2 by 2 matrix multiplication
  - Naïve algorithm takes 8 multiplication
  - **Strassen** showed that one can get away with 7
- Divide and conquer gives $O\left(n^{\log_2 7}\right) \approx O\left(n^{2.807}\right)$
  - Improves over $O(n^3)$ for reasonable sized matrices
- Actually used in practice!

Timeline of matrix multiplication exponent

Year	Bound on omega	Authors
1969	2.8074	Strassen[1]
1978	2.796	Pan[11]
1979	2.780	Bini, Capovani [it], Romani[12]
1981	2.522	Schönhage[13]
1981	2.517	Romani[14]
1981	2.496	Coppersmith, Winograd[15]
1986	2.479	Strassen[16]
1990	2.3755	Coppersmith, Winograd[17]
2010	2.3737	Stothers[18]
2013	2.3729	Williams[19][20]
2014	2.3728639	Le Gall[21]
2020	2.3728596	Alman, Williams[6][22]
2022	2.371866	Duan, Wu, Zhou[3]
2023	2.371552	Williams, Xu, Xu, and Zhou[2]

Best lower bound is still $\Omega(n^2 \log n)$

Machine arithmetic - Decimal expansion

- Take 316.1415 for example:

$$316.1415 = 3 \cdot 10^2 + 1 \cdot 10^1 + 6 \cdot 10^0 + 1 \cdot 10^{-1} + 4 \cdot 10^{-2} + 1 \cdot 10^{-3} + 5 \cdot 10^{-4}.$$

- Any real number x can be written as

$$x = \pm \sum_{j=-\infty}^{\infty} d_j \cdot 10^j$$

- In-class exercise: Decimal expansions for (1) -2, (2) π.

Machine arithmetic - Binary expansion

- Similar to decimal expansion, every real number x has a binary (i.e., base B = 2) expansion:

$$x = \pm \sum_{j=-\infty}^{\infty} b_j \cdot 2^j$$

- In class exercise: consider a number $x = -(1011.01)_2$, what's the binary expansion of it?

$$x = -(1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2})$$

Decimal to binary conversion

- Every number has a decimal and a binary expansion. Given a decimal expansion for a number x, how do we determine its binary expansion?

- We set $y = x$ and repeatedly do the following:
- 1. Compute the maximum integer j such that $y \geqslant 2^j$.
- 2. Output j.
- 3. Compute $y = y - 2^j$ and go to step 1.
- The algorithm terminates when $y = 0$.

Conversion example of x=3.25

- We set $y = x$ and repeatedly do the following:
- 1. Compute the maximum integer j such that $y \geqslant 2^j$.
- 2. Output j.
- 3. Compute $y = y - 2^j$ and go to step 1.
- The algorithm terminates when $y = 0$.

1. $y = 3.25$. $j = 1$, *since* $3.25 \geqslant 2^1$, *but* $3.25 < 2^2$. *So output* 1.
2. $y = 3.25 - 2^1 = 1.25$. $j = 0$, *since* $1.25 \geqslant 2^0$ *but less than* 2^1. *So output* 0.
3. $y = 1.25 - 2^0 = 0.25$. $j = -2$, *since* $0.25 = 2^{-2}$. *So output* -2.
4. $y = 0$, *so stop.*

This shows that

$$x = (3.25)_{10} = (11.01)_2.$$

Conversion example of x=0.10

- We set $y = x$ and repeatedly do the following:
- 1. Compute the maximum integer j such that $y \geqslant 2^j$.
- 2. Output j.
- 3. Compute $y = y - 2^j$ and go to step 1.
- The algorithm terminates when $y = 0$.

1. $y = 0.10$. $j = -4$, *since* $0.10 \geqslant 2^{-4}$ *but less than* $2^{-3} = 0.125$. *So output* -4.

2. $y = 0.10 - 2^{-4} = 0.0375$. $j = -5$, *since* $0.0375 \geqslant 2^{-5} = 0.03125$. *So output* -5.

3. $y = 0.0375 - 0.03125 = 0.00625$. $j = -8$, *since* $0.00625 \geqslant 2^{-8} = 0.00390625$. *So output* -8.

We can keep doing this, and the process never terminates. We get

$$x = (0.10)_{10} = (0.0001100110011...)_2. \tag{5.8}$$

In-class exercise: Conversion of x=4.125

- We set $y = x$ and repeatedly do the following:
- 1. Compute the maximum integer j such that $y \geqslant 2^j$.
- 2. Output j.
- 3. Compute $y = y - 2^j$ and go to step 1.
- The algorithm terminates when $y = 0$.

- Solution: $4.125_{10} = 100.001_2$

Scientific notation

- Our ultimate goal: come up with a reasonable binary representation of numbers, suitable for storage and manipulation on a computer.
  - Why not just store the binary expansion? The trouble with this is that large numbers can take up a lot more space than smaller numbers, even if they don’t have many nonzero digits.
- For instance, consider the following very large number (Avogadro’s constant) that arises in chemistry:

$$602000000000000000000000$$

- How can we store this number in a compact way?

Scientific notation

Recall how scientific notation works. In decimal, we can write any real number other than 0 as

$$x = \pm m \times 10^E, \tag{5.12}$$

for a unique **mantissa** m and exponent E, with $1 \leqslant m < 10$ and E some integer. For example, consider the number 314.159. In scientific notation, this is written as

$$3.14159 \times 10^2. \tag{5.13}$$

In the same fashion, a number can be written in base 2 scientific notation: it takes the form

$$x = \pm m \times 2^E, \tag{5.14}$$

where this time $1 \leqslant m < 2$. For instance, consider the number 3.25. We converted this to binary to get $(11.01)_2$. In scientific notation, this becomes

$$(1.101)_2 \times 2^1. \tag{5.15}$$

- In-class exercise: scientific notations of 4125, 40.125, 4.125

What’s next?

- Mon Sep 1 – NO CLASS
  - Due to Labor Day
  - No Instructor’s office hour either
  - But HW 1 will be released, based on Lecture 1, 2, 3

- Mon Sep 8 – TA review sessions
  - First session: Lecture 3: Review of Linear Algebra
  - Second session: Tutorials of Matlab/Python/LaTeX
  - Study group registration due and Course project registration due