Advanced Algorithmic Problem Solving

Le 3 – Arithmetic

Fredrik Heintz
Dept of Computer and Information Science
Linköping University

Overview

- Arithmetic
- Integer multiplication – Karatsuba's algorithm
- Multiplication of polynomials – Fast Fourier Transform
- Systems of linear equations – Naïve Gaussian Elimination

Arithmetic

- Range of default integer data types (C++)
  - unsigned int = unsigned long: 232 (9-10 digits)
  - unsigned long long: 264 (19-20 digits)
- How to represent 777!
- Operations on Big Integer
  - Basic: add, subtract, multiply, divide, etc
  - Use "high school method"

```
 1   ← carry
218
 45
--- +
263
```

```
 218
  45
 ---  |x
1090    (218*5)
872     (218*4)*10
----- +
9810
```

Arithmetic

- Greatest Common Divisor (Euclidean Algorithm)
  - GCD(a, 0) = a
  - GCD(a, b) = GCD(b, a mod b)
  - `int gcd(int a, int b) { return (b == 0 ? a : gcd(b, a % b)); }`
- Least Common Multiplier
  - LCM(a, b) = a*b / GCD(a, b)
  - `int lcm(int a, int b) { return (a / gcd(a, b) * b); }`
    - // Q: why we write the lcm code this way?
- GCD/LCM of more than 2 numbers:
  - GCD(a, b, c) = GCD(a, GCD(b, c))
- Find d, x, y such that d = ax + by and d = GCD(a,b) (Extended Euclidean Algorithm)
  - EGCD(a,0) = (a,1,0)
  - EGCD(a,b)
    - (d',x',y') = EGCD(b, a mod b)
    - (d,x,y) = (d',y',x' – a/b*y')

Arithmetic

- Representing rational numbers.
  - Pairs of integers a,b where GCD(a,b)=1.
- Representing rational numbers modulo m.
  - The only difficult operation is inverse, $ax = 1 \pmod{m}$, where an inverse exists if and only if a and m are co-prime (gcd(a,m)=1).
  - Can be found using the Extended Euclidean Algorithm
    $ax = 1 \pmod{m} \Rightarrow ax - 1 = qm \Rightarrow ax - qm = 1$
    $(d, x, y) = EGCD(a,m) \Rightarrow x$ is the solution iff $d = 1$.

Karatsuba's algorithm

- Using the classical pen and paper algorithm two n digit integers can be multiplied in $O(n^2)$ operations. Karatsuba came up with a faster algorithm.
- Let A and B be two integers with
  - $A = A_1 10^k + A_0,\ A_0 < 10^k$
  - $B = B_1 10^k + B_0,\ B_0 < 10^k$
  - $C = A^*B = (A_1 10^k + A_0)(B_1 10^k + B_0)$

    $= A_1 B_1 10^{2k} + (A_1 B_0 + A_0 B_1) 10^k + A_0 B_0$

  Instead this can be computed with 3 multiplications
  - $T_0 = A_0 B_0$
  - $T_1 = (A_1 + A_0)(B_1 + B_0)$
  - $T_2 = A_1 B_1$
  - $C = T_2 10^{2k} + (T_1 - T_0 - T_2) 10^k + T_0$

Complexity of Karatsuba's Algorithm

- Let T(n) be the time to compute the product of two n-digit numbers using Karatsuba's algorithm. Assume $n = 2^k$. $T(n) = \Theta(n^{\lg(3)})$, $\lg(3) \approx 1.58$

- $T(n) \le 3T(n/2) + cn$

  $\le 3(3T(n/4) + c(n/2)) + cn = 3^2T(n/2^2) + cn(3/2 + 1)$

  $\le 3^2(3T(n/2^3) + c(n/4)) + cn(3/2 + 1)$

  $= 3^3T(n/2^3) + cn(3^2/2^2 + 3/2 + 1)$

  …

  $\le 3^iT(n/2^i) + cn(3^{i-1}/2^{i-1} + \ldots + 3/2 + 1)$

  …

  $\le c3^k + cn[((3/2)^k - 1)/(3/2 - 1)]$ --- Assuming $T(1) \le c$

  $\le c3^k + 2c(3^k - 2^k) \le 3c3^{\lg(n)} = 3cn^{\lg(3)}$

Fast Fourier Transform

- See separate presentation

Systems of Linear Equations

A system of linear equations can be presented in different forms

$$\left.\begin{array}{l} 2x_1 + 4x_2 - 3x_3 = 3 \\ 2.5x_1 - x_2 + 3x_3 = 5 \\ x_1 \qquad\quad - 6x_3 = 7 \end{array}\right\} \Leftrightarrow \begin{bmatrix} 2 & 4 & -3 \\ 2.5 & -1 & 3 \\ 1 & 0 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}$$

Standard form — Matrix form

Solutions of Linear Equations

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \text{is a solution to the following equations:}$$

$$x_1 + x_2 = 3$$

$$x_1 + 2x_2 = 5$$

Solutions of Linear Equations

- A set of equations is **inconsistent** if there exists no solution to the system of equations:

$$x_1 + 2x_2 = 3$$

$$2x_1 + 4x_2 = 5$$

These equations are inconsistent

Solutions of Linear Equations

- Some systems of equations may have **infinite number of solutions**

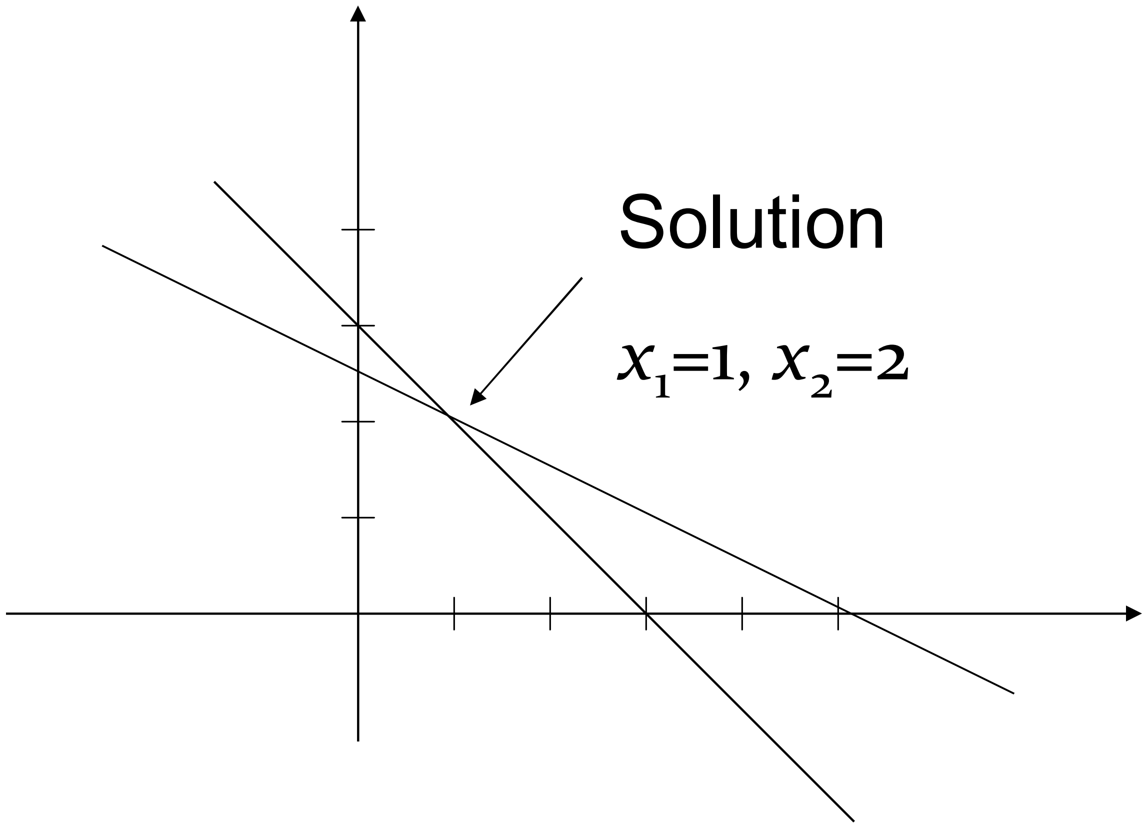
$$x_1 + 2x_2 = 3$$
$$2x_1 + 4x_2 = 6$$

have infinite number of solutions

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a \\ 0.5(3-a) \end{bmatrix} \text{ is a solution for all } a$$

Graphical Solution of Systems of Linear Equations

$$x_1 + x_2 = 3$$
$$x_1 + 2x_2 = 5$$



Cramer's Rule is Not Practical

Cramer's Rule can be used to solve the system

$$x_1 = \frac{\begin{vmatrix} 3 & 1 \\ 5 & 2 \end{vmatrix}}{\begin{vmatrix} 1 & 1 \\ 1 & 2 \end{vmatrix}} = 1, \qquad x_2 = \frac{\begin{vmatrix} 1 & 3 \\ 1 & 5 \end{vmatrix}}{\begin{vmatrix} 1 & 1 \\ 1 & 2 \end{vmatrix}} = 2$$

Cramer's Rule is not practical for large systems.

To solve N by N system requires $(N+1)(N-1)N!$ multiplications.

To solve a 30 by 30 system, 2.38×10^{35} multiplications are needed.

It can be used if the determinants are computed in efficient way

Naive Gaussian Elimination

- The method consists of two steps:
  - **Forward Elimination**: the system is reduced to upper triangular form. A sequence of elementary operations is used.
  - **Backward Substitution**: Solve the system starting from the last variable.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \Rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}' & a_{23}' \\ 0 & 0 & a_{33}' \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2' \\ b_3' \end{bmatrix}$$

Elementary Row Operations

- Adding a multiple of one row to another
- Multiply any row by a non-zero constant

Example: Forward Elimination

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ 26 \\ -19 \\ -34 \end{bmatrix}$$

Part 1 : Forward Elimination

Step 1 : Eliminate x_1 from equations 2, 3, 4

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & -12 & 8 & 1 \\ 0 & 2 & 3 & -14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ -6 \\ -27 \\ -18 \end{bmatrix}$$

Example: Forward Elimination

Step2 : Eliminate x_2 from equations 3, 4

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 4 & -13 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ -6 \\ -9 \\ -21 \end{bmatrix}$$

Step3 : Eliminate x_3 from equation 4

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ -6 \\ -9 \\ -3 \end{bmatrix}$$

Example: Forward Elimination

Summary of the Forward Elimination :

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ 26 \\ -19 \\ -34 \end{bmatrix} \Rightarrow \begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ -6 \\ -9 \\ -3 \end{bmatrix}$$

Example: Backward Substitution

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ -6 \\ -9 \\ -3 \end{bmatrix}$$

Solve for x_4, then solve for x_3,... solve for x_1

$$x_4 = \frac{-3}{-3} = 1, \qquad x_3 = \frac{-9+5}{2} = -2$$

$$x_2 = \frac{-6-2(-2)-2(1)}{-4} = 1, \qquad x_1 = \frac{16+2(1)-2(-2)-4(1)}{6} = 3$$

Forward Elimination

To eliminate x_1

$$\left.\begin{aligned} a_{ij} &\leftarrow a_{ij} - \left(\frac{a_{i1}}{a_{11}}\right) a_{1j} \quad (1 \le j \le n) \\ b_i &\leftarrow b_i - \left(\frac{a_{i1}}{a_{11}}\right) b_1 \end{aligned}\right\} 2 \le i \le n$$

To eliminate x_2

$$\left.\begin{aligned} a_{ij} &\leftarrow a_{ij} - \left(\frac{a_{i2}}{a_{22}}\right) a_{2j} \quad (2 \le j \le n) \\ b_i &\leftarrow b_i - \left(\frac{a_{i2}}{a_{22}}\right) b_2 \end{aligned}\right\} 3 \le i \le n$$

Forward Elimination

To eliminate x_k

$$\left.\begin{aligned} a_{ij} &\leftarrow a_{ij} - \left(\frac{a_{ik}}{a_{kk}}\right) a_{kj} \quad (k \le j \le n) \\ b_i &\leftarrow b_i - \left(\frac{a_{ik}}{a_{kk}}\right) b_k \end{aligned}\right\} k+1 \le i \le n$$

Continue until x_{n-1} is eliminated.

Backward Substitution

$$x_n = \frac{b_n}{a_{n,n}}$$

$$x_{n-1} = \frac{b_{n-1} - a_{n-1,n}x_n}{a_{n-1,n-1}}$$

$$x_{n-2} = \frac{b_{n-2} - a_{n-2,n}x_n - a_{n-2,n-1}x_{n-1}}{a_{n-2,n-2}}$$

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{i,j}x_j}{a_{i,i}}$$

Determinant

The elementary operations do not affect the determinant

Example:

$$A=\begin{bmatrix}1 & 2 & 3\\ 2 & 3 & 2\\ 3 & 1 & 2\end{bmatrix}\xrightarrow{\text{Elementary operations}} A'=\begin{bmatrix}1 & 2 & 3\\ 0 & -1 & -4\\ 0 & 0 & 13\end{bmatrix}$$

$$\det(A)=\det(A')=-13$$

How Many Solutions Does a System of Equations AX=B Have?

Unique	No solution	Infinite
$\det(A) \neq 0$	$\det(A) = 0$	$\det(A) = 0$
reduced matrix has no zero rows	reduced matrix has one or more zero rows corresponding B elements $\neq 0$	reduced matrix has one or more zero rows corresponding B elements $= 0$

Examples

Unique

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} X = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} 1 & 2 \\ 0 & -2 \end{bmatrix} X = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

solution:

$$X = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}$$

No solution

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} X = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} 1 & 2 \\ 0 & 0 \end{bmatrix} X = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

No solution

$$0 = -1 \; impossible!$$

infinte # of solutions

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} X = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} 1 & 2 \\ 0 & 0 \end{bmatrix} X = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

Infinite # solutions

$$X = \begin{bmatrix} \alpha \\ 1 - .5\alpha \end{bmatrix}$$

Pseudo-Code: Forward Elimination

do k = 1 **to** n-1

  do i = k+1 **to** n

    factor = $a_{i,k}$ / $a_{k,k}$

    do j = k+1 **to** n

      $a_{i,j} = a_{i,j} - \text{factor} * a_{k,j}$

    end do

    $b_i = b_i - \text{factor} * b_k$

  end do

end do

Pseudo-Code: Back Substitution

$x_n = b_n / a_{n,n}$

do i = n-1 **downto** 1

    sum = b_i

    do j = i+1 **to** n

        sum = sum – $a_{i,j} * x_j$

    end do

    x_i = sum / $a_{i,i}$

end do

Problems with Naive Gaussian Elimination

- The Naive Gaussian Elimination may fail for very simple cases. **(The pivoting element is zero).**

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

- **Very small pivoting element may result in serious computation errors**

$$\begin{bmatrix} 10^{-10} & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

How Do We Know If a Solution is Good or Not

Given AX=B

X is a solution if AX-B=0

Compute the residual vector R= AX-B

Due to rounding error, R may not be zero

$$\text{The solution is acceptable if } \max_i |r_i| \le \varepsilon$$

How Good is the Solution?

$$\begin{bmatrix} 1 & -1 & 2 & 1 \\ 3 & 2 & 1 & 4 \\ 5 & -8 & 6 & 3 \\ 4 & 2 & 5 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} \quad solution \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -1.8673 \\ -0.3469 \\ 0.3980 \\ 1.7245 \end{bmatrix}$$

$$\text{Residues}: \; R = \begin{bmatrix} 0.005 \\ 0.002 \\ 0.003 \\ 0.001 \end{bmatrix}$$

Summary

- Arithmetic
- Integer multiplication – Karatsuba's algorithm
- Multiplication of polynomials – Fast Fourier Transform
- Systems of linear equations – Naïve Gaussian Elimination