Revision - System of Linear Equations

$$Ax = b$$

1. Direct Methods

a. Gauss Elimination (GE)

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \xrightarrow[\text{elimination}]{\text{Forward}} \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ 0 & & \ddots & \\ & & & u_{nn} \end{bmatrix} \xrightarrow[\text{Substitution}]{\text{Backward}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

3 issues

(i) Zero - pivot

(ii) Round off

(iii) Ill conditioning

Scaling $- a_{ij} = \dfrac{a_{ij}}{|A_i|}$ $\qquad A_i = \max\limits_{\forall j} a_{ij}$

decide pivot equation

Partial pivoting - Exchange rows

b. Gauss Jordan

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{n1} & \cdots & \cdots & a_{nn} \end{bmatrix} \xrightarrow{\text{Elimination}} \underbrace{\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & \ddots \ 1 \end{bmatrix}}_{I} \rightarrow \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Which of the two algorithms is better?

- Minimum Roundoff errors — C_A
- Minimum Storage requirement
- Minimum computational time
- Programming ease — (Subjective)

Computing Time

- Speed of computer
- Programming languages
- Input data

Computational or Algorithmic complexity

Instead of measuring time in μ seconds, we measure time in terms of number of basic steps executed by the algorithm

(+, -, ×, ÷, assignment, comparison)

Instead of representing algorithmic complexity as a single number we represent it in terms of size of data

Example.

Sum of n numbers

$X = [x_1\ x_2 \ldots x_n]$

```
Sum = 0
For i = 1 to n
    sum = sum + x(i)
end
```

print x(1)

$n + 1$

$n = 1000$

1001

Sum and product of n numbers

```
Sum = 0
product = 1
for i = 1 to n
    sum = sum + x(i)
    product = product * x(i)
end
```

$2(n+1)$

$2n$

Sum of all possible pairs

```
for i = 1 to n
    for j = 1 to n
        sum(i,j) = x(i) + x(j)
    end
end
```

n^2

No of basic steps

$n^2 \sim n^2 + n + c$

TWO THINGS

1. Worst case scenario

Find a number (x_0) in the vector x

$x(1) = x_0$

```
f = 0 ; i = 0
while f == 0
    i = i + 1
    if x(i) == x0
        f = 1
    end
end
```

2. Asymptotic Analysis

- Any algorithm is sufficiently efficient for small input
- When comparing algorithms for "computational time" one is interested in very large inputs
- As a proxy for "very large" asymptotic analysis that consider size of input data $\longrightarrow$ infinity

" Big O "

Gives an upper bound on the asymptotic growth of the algorithm

- The complexity of the function/algorithm is $O(n^2)$ is means that for the worst case $O(n^2)$ steps are needed to estimate function when n is very large.

- If the computation time is the sum of multiple terms, keep the one which has the largest growth rate and drop the others

- $2n^2$

$O(2n^2)$ $O(n^2)$

→ An algorithm takes 10 days if the complexity is $O(2n^2)$

Reduce $O(n^2)$ 5 days

Common complexity classes

Constant	$O(1)$
Logarithmic	$O(\log(n))$
Linear	$O(n)$
log. linear	$O(n \log n)$
Quadratic	$O(n^2)$
Cubic	$O(n^3)$
Exponential	$O(c^n)$ $c > 1$

Polynomial: Quadratic, Cubic

Computational Complexity of GE and GJ

Recap.

$$\sum_{i=1}^{n} 1 = n$$

$$\sum_{i=k}^{n} 1 = n-k+1$$

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} = \frac{n^2}{2} + \frac{n}{2}$$

$O(n^2/2)$

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

$O\left(\frac{n^3}{3}\right)$

Gauss Elimination

Forward elimination

for $K = 1$ to $n-1$

for $i = K+1$ to n

$l_{ik} = \frac{a_{ik}}{a_{kk}}$

for $j = K+1$ to n

$a_{ij} = a_{ij} - l_{ik} a_{ij}$

end

end

end

$n-k$

$(n-k+1)$

No of divisions

$$\sum_{k=1}^{n} (n-k) = \frac{n(n-1)}{2}$$

No. of multiplications

$$\sum_{k=1}^{n-1} (n-k)(n-k+1)$$

$$= \frac{n^3}{3} - n$$

No of additions/subtraction

$$\frac{n^3}{3} - n$$

Total steps in forward elimination

no of divisions + no of multiplication + no of subtraction

$$= \frac{n(n-1)}{2} + 2\left(\frac{n^3}{2} - \frac{n}{3}\right)$$

$$= \frac{2n^3}{3} + \frac{n^2}{2} - \frac{7n}{6}$$

$$= \frac{2n^3}{3} + O(n^2)$$

Backward substitution

No of steps $n^2 + O(n)$

Total Gauss elimination

$$\frac{2n^3}{3} + O(n^2) + n^2 + O(n)$$

$$= \boxed{\frac{2}{3}n^3 + O(n^2)}$$

$O(n^3)$

$O\left(\frac{2}{3}n^3\right)$

Gauss Jordan

No of steps $n^3 + n^2 - n$

$$= \boxed{n^3 + O(n^2)}$$

3. LU Decomposition

Motivation

$$A x = b$$

System characteristic

external forcing