# CS475/CS675
# Lecture 1: May 3, 2016

Basic Theory of Linear Algebra
Reading: [TB] chapt 1 (p. 1-10), chapt 20 (p. 147-152)

# Range

- Definition: the range of A is defined as
  - $range(A) = \{y: y = Ax \text{ for some } x\}$

- Theorem:
  - $range(A) =$ space spanned by the columns of $A = [a_1\ a_2 \ldots a_n]$
    $= \{y = x_1a_1 + x_2a_2 + \cdots + x_na_n = \sum_{j=1}^{n} a_jx_j\}$

- Thus $range(A)$ is also called the column space of $A$

# Rank

- <u>Definition:</u>
  - Column rank = dimension of column space
  - Row rank = dimension of row space

- <u>Theorem:</u> column rank = row rank

- Thus we simply call it the rank of $A$, $rank(A)$

# Full Rank

- <u>Definition:</u> An $m \times n$ matrix is of full rank if
$$rank(A) = \min\{m, n\}$$

- Thus, if $m \geq n$, $(A = \square)$, then a full rank matrix has n independent column vectors

- <u>Definition:</u> A nonsingular (invertible) matrix is a square matrix of full rank
- <u>Definition:</u> The null space of $A$, $null(A)$ is defined as
$$null(A) = \{x: Ax = 0\}$$

# Matrix inverse

- Matrix inverse

  $(AB)^{-1} = B^{-1}A^{-1}$

  $(A^{-1})^T = (A^T)^{-1} = A^{-T}$

- Interesting identity:

  $B^{-1} = A^{-1} - B^{-1}(B - A)A^{-1}$

  Proof: $B(A^{-1} - B^{-1}(B - A)A^{-1})$

  $= BA^{-1} - (B - A)A^{-1}$

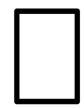
  $= BA^{-1} - BA^{-1} + I$

  $= I$

# Updating Matrix Inverses

- Sherman-Morrison-Woodbury formula

$$\left(\mathrm{A} + \mathrm{UV}^{\mathrm{T}}\right)^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1} U)^{-1} V^T A^{-1}$$

where $U, V \in \Re^{n \times k}$ ($U =$  n ).
k

- Thus a rank $k$ correction to $A$ results in a rank $k$ correction of the inverse

# Example

# How to compute $x = A^{-1}b$?

- In numerical linear algebra, NEVER compute $A^{-1}$ and then $A^{-1}b$.

- We always consider $x$ as the solution of the eqn:
$$Ax = b$$

- We compute $x$ by solving the equation by Gaussian Elimination

# Gaussian Elimination (GE)

- Big picture of GE:

# GE algorithm

For $i = 1,2, \ldots, n-1$

For $k = i+1, \ldots, n$

$mult = a_{ki}/a_{ii}$ $\quad (a_{ki} = mult)$

for $j = i+1, \ldots, n$

$a_{kj} = a_{kj} - mult \times a_{ij}$

end

} Update $row(k)$

$b_k = b_k - mult \times b_i$ } Update RHS

end

End

At the end, $A^{(n-1)}x = b^{(n-1)}$ is solved by back substitution

# LU Factorization

- Theorem: $A = LU$ where

  $L =$ lower $\triangle$, unit diagonal $= \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ \text{mult} & & 1 \end{bmatrix}$

  $U =$ upper $\triangle$ $= A^{(n-1)}$

# Solve $Ax = b$

- $Ax = b \rightarrow LUx = b$
- Let $y = Ux$, then we have $Ly = b$
  - Solve $Ly = b$ by forward solve
  - Solve $Ux = y$ by back solve

# Forward solve algorithm

For $i = 1,2,\ldots,n$

$\quad y_i = b_i$

$\quad$ For $j = 1,2,\ldots,i-1$

$\quad\quad y_i = y_i - l_{ij} \times y_j$

$\quad$ end

end

$$y_i = b_i - \sum_{j=1}^{i-1} l_{ij} y_j$$

# Complexity

- 1 flop = + / - / x / ÷
- Consider forward solve
  - For each $i$, the $j$-loop performs $2(i-1)$ flops
  - Total flops $= \sum_{i=1}^{n}(2i-2)$

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

# Complexity (continued)

- $\text{flops}(\text{back solve}) = n^2$ (exercise)
- $\text{flops}(LU) = \frac{2}{3}n^3 + O(n^2)$

- For large $n$, factorization is more expensive than forward or backward solves