CS 450: Numerical Analysis
Lecture 5
Chapter 2 – Linear Systems
Solving Linear Systems

Edgar Solomonik

Department of Computer Science
University of Illinois at Urbana-Champaign

January 31, 2018
Solving Triangular Systems

- \( Lx = b \) if \( L \) is lower-triangular is solved by forward substitution:

\[
\begin{align*}
    l_{11}x_1 &= b_1 && x_1 = b_1/l_{11} \\
    l_{21}x_1 + l_{22}x_2 &= b_2 && \Rightarrow x_2 = (b_2 - l_{21}x_1)/l_{22} \\
    l_{31}x_1 + l_{32}x_2 + l_{33}x_3 &= b_2 && x_3 = (b_3 - l_{31}x_1 - l_{32}x_2)/l_{33} \\
    \vdots & & \vdots \\
\end{align*}
\]

- Computational complexity of forward/backward substitution:

\[
T(n) = O(n^2) \\
\frac{n^2}{2} \text{ multipli} \hspace{0.5cm} \frac{n^2}{2} \text{ addition}
\]
Solving Triangular Systems

- Existence of solution to $Lx = b$:

  May not exist if some $L_{i,i} = 0$, since then $L$ is singular.

- Invertibility of $L$ and existence of solution:

  Solution may exist even if $L_{i,i} = 0$, but is not unique.

  $L$ is full rank if each $L_{i,i} \neq 0$.

  (Determinant shows this formally, since $\det(L) = \prod_{i} L_{i,i}$, but can also see that left euler is not in the span of the last $n-k+1$, since $L_{k,i} \neq 0$.)
Properties of Triangular Matrices

- \( XY = Z \) is lower triangular is \( X \) and \( Y \) are both lower triangular:

\[
\begin{align*}
\begin{array}{c}
\Delta \\
\Delta \\
\end{array}
\end{align*}
\begin{align*}
\begin{array}{c}
\Delta \\
\Delta \\
\end{array}
\end{align*}
\begin{align*}
\begin{array}{c}
\Delta \\
\end{array}
\end{align*}
\begin{align*}
\begin{array}{c}
\Delta \\
\end{array}
\end{align*}
\]

- \( L^{-1} \) is lower triangular if it exists:
LU Factorization

- An **LU factorization** consists of a unit-lower-triangular factor $L$ and upper-triangular factor $U$ such that $A = LU$:

  $n^2$ variables in $L, U \leftarrow \frac{n(n+1)}{2}$

  $n^2$ variables in $A$
Gaussian Elimination

- The LU factorization may not exist:

Consider matrix

\[
\begin{bmatrix}
1 & 2 \\
2 & 4 \\
0 & 3 \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
1 & 0 & 0 \\
2 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix} \begin{bmatrix}
a_{11} & a_{12} \\
0 & a_{22} \\
0 & 0 \\
\end{bmatrix}
\]

- Permutation of variables enables us to transform the linear system so the LU factorization does exist:
Gaussian Elimination Algorithm

- Algorithm for factorization is derived from equations given by $A = LU$:

$$
\begin{bmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{bmatrix} =
\begin{bmatrix}
L_{11} & L_{12} \\
L_{21} & L_{22}
\end{bmatrix}
\begin{bmatrix}
u_{11} & u_{12} \\
u_{21} & u_{22}
\end{bmatrix}
$$

- The $k$th column of $L$ is given by the $k$th elementary matrix $M_k$:

$$L_{21} u_{12} + L_{22} u_{22} = A_{22}$$

$A_{21} \cdot L_{21} u_{12} = L_{22} u_{22}$

Solve completed.
An elimination matrix $M_k$ satisfies the following properties:
Gaussian Elimination with Partial Pivoting

- **Partial pivoting** permutes rows to make divisor $u_{ij}$ is maximal at each step:

$$A = PLU \quad P^TA = LU$$

- A row permutation corresponds to an application of a row permutation matrix $P_{jk} = I - (e_j - e_k)(e_j - e_k)^T$:

$$P^TP = I$$

$$P_{jk} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \\ -1 & \cdots & 0 \\ \vdots & \cdots & \vdots \\ 0 & \cdots & -1 \end{bmatrix}$$
Complete Pivoting and Error Bounds

- **Complete pivoting** permutes rows and columns to make divisor $u_{ii}$ maximal at each step:

\[ |L|_{i,j} \leq 1 \]

- For LU, the backward error $\delta A$, so that $\hat{L}\hat{U} = A + \delta A$, satisfies bound

\[ |\delta A_{ij}| \leq \epsilon (|\hat{L}| \cdot |\hat{U}|)_{ij} \]

[Diagram of absolute value signs and matrix operations]