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
Solving Basic Linear Systems

- Solve $Dx = b$ if $D$ is diagonal
  \[ x_i = b_i / d_{ii} \text{ with total cost } O(n) \]
- Solve $Qx = b$ if $Q$ is orthogonal
  \[ x = Q^T b \text{ with total cost } O(n^2) \]
- Given SVD $A = U\Sigma V^T$, solve $Ax = b$
  - Compute $z = U^T b$
  - Solve $\Sigma y = z$ (diagonal)
  - Compute $x = V^T z$
Solving Triangular Systems

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

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

- Algorithm can also be formulated recursively by blocks:

\[
\begin{bmatrix}
l_{11} & 0 \\
l_{21} & L_{22}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix} =
\begin{bmatrix}
b_1 \\
b_2
\end{bmatrix}
\]

\( x_1 = b_1/l_{11} \), then solve recursively for \( x_2 \) in \( L_{22}x_2 = b_2 - l_{21}x_1 \).
Solving Triangular Systems

- **Existence of solution to** $Lx = b$:
  
  If some $l_{ii} = 0$, the solution may not exist, and $L^{-1}$ does not exist.

- **Uniqueness of solution**: Even if some $l_{ii} = 0$ and $L^{-1}$ does not exist, the system may have a solution. The solution will not be unique since columns of $L$ are necessarily linearly dependent if a diagonal element is zero. May want to select solution minimizing norm of $x$.

- **Computational complexity of forward/backward substitution**: The recursive algorithm has the cost recurrence,

  $$T(n) = T(n - 1) + n = \sum_{i=1}^{n} i = n(n + 1)/2.$$ 

  The total cost is $n^2/2$ multiplications and $n^2/2$ additions to leading order.
Properties of Triangular Matrices

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

\[
\begin{bmatrix}
  z_{11} & z_{12} \\
  z_{21} & Z_{22}
\end{bmatrix} =
\begin{bmatrix}
  x_{11} & x_{21} \\
  y_{11} & y_{21}
\end{bmatrix}
\begin{bmatrix}
  y_{11} & Y_{22} \\
  y_{21}
\end{bmatrix}.
\]

Clearly, \( z_{11} = x_{11}y_{11} \) and \( z_{12} = 0 \), then we proceed by the same argument for the triangular matrix product \( Z_{22} = X_{22}Y_{22} \).

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

We give a constructive proof by providing an algorithm for triangular matrix inversion. We need \( Y = X^{-1} \) so

\[
\begin{bmatrix}
  Y_{11} & Y_{12} \\
  Y_{21} & Y_{22}
\end{bmatrix}
\begin{bmatrix}
  X_{11} & X_{22} \\
  X_{21}
\end{bmatrix} =
\begin{bmatrix}
  I & I
\end{bmatrix},
\]

from which we can deduce

\( Y_{11} = X_{11}^{-1} \), \( Y_{22} = X_{22}^{-1} \), \( Y_{21} = -Y_{22}X_{21}Y_{11} \).
LU Factorization

- An **LU factorization** consists of a unit-diagonal lower-triangular factor $L$ and upper-triangular factor $U$ such that $A = LU$:
  - Unit-diagonal implies each $l_{ii} = 1$, leaving $n(n - 1)/2$ unknowns in $L$ and $n(n + 1)/2$ unknowns in $U$, for a total of $n^2$, the same as the size of $A$.
  - For rectangular matrices $A \in \mathbb{R}^{m \times n}$, one can consider a full LU factorization, with $L \in \mathbb{R}^{m \times \max(m,n)}$ and $U \in \mathbb{R}^{\max(m,n) \times n}$, but it is fully described by a reduced LU factorization, with lower-trapezoidal $L \in \mathbb{R}^{m \times \min(m,n)}$ and upper-trapezoidal $U \in \mathbb{R}^{\min(m,n) \times n}$.

- Given an LU factorization of $A$, we can solve the linear system $Ax = b$:
  - using forward substitution $Ly = b$
  - using backward substitution to solve $Ux = y$

Backward substitution is the same as forward substitution with a reversal of the ordering of the elements of the vectors and the ordering of the rows/columns of the matrix.
Gaussian Elimination Algorithm

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

\[
\begin{bmatrix}
    a_{11} & a_{12} \\
    a_{21} & A_{22}
\end{bmatrix} = \begin{bmatrix}
    1 & \\
    l_{21} & L_{22}
\end{bmatrix} \begin{bmatrix}
    u_{11} & u_{12} \\
    & U_{22}
\end{bmatrix} = \begin{bmatrix}
    L_{11} & \\
    L_{21} & L_{22}
\end{bmatrix} \begin{bmatrix}
    U_{11} & U_{12} \\
    & U_{22}
\end{bmatrix}
\]

- First, observe $\begin{bmatrix} u_{11} & u_{12} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \end{bmatrix}$
- To obtain $l_{21}$ compute $l_{21} = a_{21}/u_{11}$
- Obtain $L_{22}$ and $U_{22}$ by recursively computing LU of the Schur complement $S = A_{22} - l_{21}u_{12}$

- The computational complexity of LU is $O(n^3)$:

  Computing $l_{21} = a_{21}/u_{11}$ requires $O(n)$ operations, finding $S$ requires $2n^2$, so to leading order the complexity of LU is

  \[
  T(n) = T(n - 1) + 2n^2 = \sum_{i=1}^{n} 2i^2 \approx 2n^3/3
  \]
Existence of LU factorization

- The LU factorization may not exist: Consider matrix \[
\begin{bmatrix}
3 & 2 \\
6 & 4 \\
0 & 3
\end{bmatrix}
\].

Proceeding with Gaussian elimination we obtain

\[
\begin{bmatrix}
3 & 2 \\
6 & 4 \\
0 & 3
\end{bmatrix} = \begin{bmatrix}
1 & 0 \\
2 & 1 \\
0 & l_{32}
\end{bmatrix} \begin{bmatrix}
3 & 2 \\
0 & u_{21}
\end{bmatrix}.
\]

Then we need that \(4 = 4 + u_{21}\) so \(u_{21} = 0\), but at the same time \(l_{32}u_{21} = 3\).

More generally, if for any partitioning \[
\begin{bmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{bmatrix}
\] the leading minor is singular (\(\det(A_{11}) = 0\)), \(A\) has no LU factorization.

- Permutation of rows enables us to transform the matrix so the LU factorization does exist:

Gaussian elimination can only fail if dividing by zero. At every recursive step of Gaussian elimination, if the leading entry of the first row is zero, we permute it with a row with an leading nonzero (if \(a_{21} = 0\), we set \(u_{11} = 0\) and \(l_{21} = 0\)).
Gaussian Elimination with Partial Pivoting

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

  Based on our argument above, for any matrix $A$ there exists a permutation matrix $P$ that can permute the rows of $A$ to permit an LU factorization,

  \[ PA = LU. \]

  Partial pivoting finds such a permutation matrix $P$ one row at a time. The $i$th row is selected to maximize the magnitude of the leading element (over elements in the first column), which becomes the entry $u_{ii}$. This selection ensures that we are never forced to divide by zero during Gaussian elimination and that the magnitude of any element in $L$ is at most 1.

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

  If we permute row $i_j$ to be the leading ($i$th) row at the $i$th step, the overall permutation matrix is given by

  \[ P^T = \prod_{i=1}^{n-1} P_{ii_j}. \]
Partial Pivoting Example

- Let's consider again the matrix \( A = \begin{bmatrix} 3 & 2 \\ 6 & 4 \\ 0 & 3 \end{bmatrix} \).

- The largest magnitude element in the first column is 6, so we select this as our pivot and perform the first step of LU:

\[
\begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} 6 & 4 \\ 3 & 2 \\ 0 & 3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1/2 \\ 0 \end{pmatrix} \begin{pmatrix} 6 & 4 \\ 0 & 2 - (1/2) \cdot 4 \\ 0 & 3 - 0 \cdot 4 \end{pmatrix}
\]

- The Schur complement is \( \begin{pmatrix} 0 & 3 \end{pmatrix}^T \) and we proceed with pivoted LU,

\[
\begin{pmatrix} 1 & 1 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} 0 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 3 \end{pmatrix}
\]

- The overall LU factorization is then given by \( P_1 \begin{pmatrix} 1 & P_2 \end{pmatrix} A = \begin{pmatrix} 1 & 0 \\ 1/2 & 1 \end{pmatrix} \begin{pmatrix} 6 & 4 \\ 3 \end{pmatrix} \).
Complete Pivoting

- Complete pivoting permutes rows and columns to make divisor $u_{ii}$ is maximal at each step:
  - Partial pivoting ensures that the magnitude of the multipliers satisfies $|l_{21}| = |a_{21}| / |u_{11}| \leq 1$
  - Complete pivoting also gives $||u_{12}||_\infty \leq |u_{11}|$ and consequently $|l_{21}| \cdot ||u_{12}||_\infty = |a_{21}| \cdot ||u_{12}||_\infty / |u_{11}| \leq |a_{21}|$
  - Complete pivoting yields a factorization of the form $LU = PAQ$ where $P$ and $Q$ are permutation matrices

- Complete pivoting is noticeably more expensive than partial pivoting:
  - Partial pivoting requires just $O(n)$ comparison operations and a row permutation
  - Complete pivoting requires $O(n^2)$ comparison operations, which somewhat increases the leading order cost of LU overall
Round-off Error in LU

Let's consider factorization of \[ \begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix} \] where \( \epsilon < \epsilon_{\text{mach}} \):

- Without pivoting we would compute 
  \[ L = \begin{bmatrix} 1 & 0 \\ \frac{1}{\epsilon} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} \epsilon & 1 \\ 0 & 1 - \frac{1}{\epsilon} \end{bmatrix} \]

- Rounding yields 
  \[ \text{fl}(U) = \begin{bmatrix} \epsilon & 1 \\ 0 & -\frac{1}{\epsilon} \end{bmatrix} \]

- This leads to 
  \[ L_{\text{fl}}(U) = \begin{bmatrix} \epsilon & 1 \\ 1 & 0 \end{bmatrix}, \text{ a backward error of } \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \]

- Permuting the rows of \( A \) in partial pivoting gives \( PA = \begin{bmatrix} 1 & 1 \\ \epsilon & 1 \end{bmatrix} \):

- We now compute 
  \[ L = \begin{bmatrix} 1 & 0 \\ \epsilon & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 1 \\ 0 & 1 - \epsilon \end{bmatrix}, \text{ so } \text{fl}(U) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \]

- This leads to 
  \[ L_{\text{fl}}(U) = \begin{bmatrix} 1 & \frac{1}{1+\epsilon} \\ \epsilon & 1 + \epsilon \end{bmatrix}, \text{ a backward error of } \begin{bmatrix} 0 & 0 \\ 0 & \epsilon \end{bmatrix} \]
Error Analysis of LU

- The main source of round-off error in LU is in the computation of the Schur complement:
  - Recall that division is well-conditioned, while addition can be ill-conditioned
  - After \( k \) steps of LU, we are working on Schur complement \( A_{22} - L_{21}U_{12} \) where \( A_{22} \) is \((n - k) \times (n - k)\), \( L_{21} \) and \( U_{12}^T \) are \((n - k) \times k\)
  - Partial pivoting and complete pivoting improve stability by making sure \( L_{21}U_{12} \) is small in norm

- When computed in floating point, absolute backward error \( \delta A \) in LU (so \( \hat{L}\hat{U} = A + \delta A \)) is
  \[
  |\delta a_{ij}| \leq \epsilon_{\text{mach}} (|\hat{L}| \cdot |\hat{U}|)_{ij}
  \]

  For any \( a_{ij} \) with \( j \geq i \) (lower-triangle is similar), we compute

  \[
  a_{ij} - \sum_{k=1}^{i} \hat{l}_{ik} \hat{u}_{kj} = a_{ij} - \langle \hat{l}_i, \hat{u}_j \rangle,
  \]

  which in floating point incurs round-off error at most \( \epsilon_{\text{mach}} \| \hat{l}_i \| \| \hat{u}_j \| \). Using this, for complete pivoting, we can show

  \[
  |\delta a_{ij}| \leq \epsilon_{\text{mach}} n^2 \|A\|_\infty.
  \]
Helpful Matrix Properties

- **Matrix is diagonally dominant**, so \( \sum_{i \neq j} |a_{ij}| \leq |a_{ii}|. \\
  Pivoting is not required if matrix is strictly diagonally dominant \( \sum_{i \neq j} |a_{ij}| < |a_{ii}|. \)

- **Matrix is symmetric positive definite (SPD)**, so \( \forall x \neq 0, x^T A x > 0: \)
  \( L = U \) and pivoting is not required, *Cholesky* algorithm can be used

- **Matrix is symmetric but indefinite**:
  Compute pivoted *LDL* factorization \( P A P^T = LDL^T \)

- **Matrix is banded**, \( a_{ij} = 0 \) if \( |i - j| > b \):
  *LU* without pivoting and *Cholesky* preserve banded structure and require only \( O(nb^2) \) work.
Solving Many Linear Systems

- Suppose we have computed $A = LU$ and want to solve $AX = B$ where $B$ is $n \times k$ with $k < n$:
  
  Cost is $O(n^2 k)$ for solving the $k$ independent linear systems

- Supposed we have computed $A = LU$ and now want to solve a perturbed system $(A - uv^T)x = b$:
  
  Can use the Sherman-Morrison-Woodbury formula

  \[
  (A - uv^T)^{-1} = A^{-1} + \frac{A^{-1}uv^TA^{-1}}{1 - v^TA^{-1}u}
  \]

  - Consequently we have $Ax = b + \frac{uv^TA^{-1}b}{1 - v^TA^{-1}u} = b + \frac{v^TA^{-1}b}{1 - v^TA^{-1}u}u$

  - Need not form $A^{-1}$ or $L^{-1}$ or $U^{-1}$, suffices to use backward/forward substitution to solve $w^TA = v^T$, i.e. solve $U^TL^Tw = v$ and then solve

  \[
  LUx = b + \left(\frac{w^Tb}{1 - w^Tu}\right)u
  \]

  scalar