EAD 115

Numerical Solution of Engineering and Scientific Problems

David M. Rocke
Department of Applied Science
Matrix Decomposition Methods

• Matrix decomposition or matrix factoring is a powerful approach to the solution of matrix problems.

• The LU decomposition takes a square matrix $A$ and writes it as the product of a lower triangular matrix $L$ and an upper triangular matrix $U$

\[ A = LU \]
The LU Decomposition

• The LU decomposition is used to solve linear equations.
• It is essentially equivalent to Gaussian elimination when only one linear system is being solved.
• If we want to solve $Ax = b$ for one matrix $A$ and many RHS’s $b$, then the LU decomposition is the method of choice.
The Cholesky Decomposition

• The Cholesky decomposition is a kind of matrix square root.
• If $A$ is a symmetric matrix, then there is a lower triangular matrix $L$ such that

$$A = LL^T$$

• $L$ is called the Cholesky factor
Singular Value Decomposition

• For any $n \times p$ matrix $A$, we can write

$A = UDV^T$

$U$ is $n \times p$ with orthonormal columns
$D$ is $p \times p$ diagonal (singular values)
$V$ is $p \times p$ with orthonormal rows and columns
**Eigenvalue/Eigenvector Decomposition**

- Under some conditions, a square $p \times p$ matrix $A$ can be written as

  \[ A = VDV^T \]

  - $D$ diagonal matrix of eigenvalues
  - $V$ orthogonal matrix of eigenvectors
LU and Gaussian Elimination

Solve

\[ Ax = b \]

Suppose that \( A \) can be written as \( A = LU \),
the product of a lower and an upper triangular matrix.

To solve

\[ LUx = b \]

First solve

\[ Ly = b \]

where \( y \) will later be \( Ux \).
\[ \begin{align*}
  l_{11}y_1 &= b_1 \\
  l_{21}y_1 + l_{22}y_2 &= b_2 \\
  l_{31}y_1 + l_{32}y_2 + l_{33}y_3 &= b_3 \\
  l_{41}y_1 + l_{42}y_2 + l_{43}y_3 + l_{44}y_4 &= b_4
\end{align*} \]

\[
\begin{bmatrix}
  l_{11} & 0 & 0 & 0 & 0 \\
  l_{21} & l_{22} & 0 & 0 & 0 \\
  l_{31} & l_{32} & l_{33} & 0 & 0 \\
  l_{41} & l_{42} & l_{43} & l_{44} & 0 \\
\end{bmatrix}
\begin{bmatrix}
  y_1 \\
  y_2 \\
  y_3 \\
  y_4 \\
\end{bmatrix}
= 
\begin{bmatrix}
  b_1 \\
  b_2 \\
  b_3 \\
  b_4 \\
\end{bmatrix}
\]
\[ u_{11} x_1 + u_{12} x_2 + u_{13} x_3 + u_{14} x_4 = y_1 \]
\[ u_{22} x_2 + u_{23} x_3 + u_{24} x_4 = y_2 \]
\[ u_{33} x_3 + u_{34} x_4 = y_3 \]
\[ u_{44} x_4 = y_4 \]

\[
\begin{bmatrix}
 u_{11} & u_{12} & u_{13} & u_{14} \\
 0 & u_{22} & u_{23} & u_{24} \\
 0 & 0 & u_{33} & u_{34} \\
 0 & 0 & 0 & u_{44}
\end{bmatrix}
\begin{bmatrix}
 x_1 \\
 x_2 \\
 x_3 \\
 x_4
\end{bmatrix}
=
\begin{bmatrix}
 y_1 \\
 y_2 \\
 y_3 \\
 y_4
\end{bmatrix}
\]
\[ Ax = b \]
\[ LUx = b \]
\[ L(Ux) = b \]
\[ Ly = b \]
\[ Ux = y \]
\[
\begin{pmatrix}
 a_{11} & a_{12} & a_{13} \\
 a_{21} & a_{22} & a_{23} \\
 a_{31} & a_{32} & a_{33}
\end{pmatrix}
\]

\[
f_{21} = \frac{a_{21}}{a_{11}}
\]

\[
f_{31} = \frac{a_{31}}{a_{11}}
\]

\[
\begin{pmatrix}
 a_{11} & a_{12} & a_{13} \\
 a_{21} - f_{21}a_{11} & a_{22} - f_{21}a_{12} & a_{23} - f_{21}a_{13} \\
 a_{31} - f_{31}a_{11} & a_{32} - f_{31}a_{12} & a_{33} - f_{31}a_{13}
\end{pmatrix}
\]

\[
\begin{pmatrix}
 a_{11} & a_{12} & a_{13} \\
 0 & a_{22} - f_{21}a_{12} & a_{23} - f_{21}a_{13} \\
 0 & a_{32} - f_{31}a_{12} & a_{33} - f_{31}a_{13}
\end{pmatrix}
\]
\[
\begin{bmatrix}
    a_{11} & a_{12} & a_{13} \\
    0 & a_{22} - f_{21} a_{12} & a_{23} - f_{21} a_{13} \\
    0 & a_{32} - f_{31} a_{12} & a_{33} - f_{31} a_{13}
\end{bmatrix}
\]

\[
f_{32} = \frac{(a_{32} - f_{31} a_{12})}{(a_{22} - f_{21} a_{12})}
\]

\[
\begin{bmatrix}
    a_{11} & a_{12} & a_{13} \\
    0 & a_{22} - f_{21} a_{12} & a_{23} - f_{21} a_{13} \\
    0 & a_{32} - f_{31} a_{12} - f_{32} (a_{23} - f_{21} a_{13}) & a_{33} - f_{31} a_{13} - f_{32} (a_{23} - f_{21} a_{13})
\end{bmatrix}
\]

\[
\begin{bmatrix}
    a_{11} & a_{12} & a_{13} \\
    0 & a_{22} - f_{21} a_{12} & a_{23} - f_{21} a_{13} \\
    0 & 0 & a_{33} - f_{31} a_{13} - f_{32} (a_{23} - f_{21} a_{13})
\end{bmatrix}
\]
\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & a_{12} & a_{13} \\
0 & a_{22} - f_{21} a_{12} & a_{23} - f_{21} a_{13} \\
0 & 0 & a_{33} - f_{31} a_{13} - f_{32} (a_{23} - f_{21} a_{13}) \\
\end{bmatrix}
\]

= 
\[
\begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33} \\
\end{bmatrix}
\]
\[ f_{21} = a_{21} / a_{11} \]
\[ f_{31} = a_{31} / a_{11} \]
\[ f_{32} = (a_{32} - f_{31} a_{12}) / (a_{22} - f_{21} a_{12}) \]
\[ = (a_{32} - (a_{31} / a_{11}) a_{12}) / (a_{22} - (a_{21} / a_{11}) a_{12}) \]
\[ = a_{11} a_{32} - a_{31} a_{12}) / (a_{11} a_{22} - a_{21} a_{12}) \]

\[ f_{21} a_{11} = (a_{21} / a_{11}) a_{11} = a_{21} \]
\[ f_{21} a_{12} + a_{22} - f_{21} a_{12} = a_{22} \]
\[ f_{21} a_{13} + a_{23} - f_{21} a_{13} = a_{23} \]
\[ f_{31} a_{11} = (a_{31} / a_{11}) a_{11} = a_{31} \]
\[ f_{31} a_{12} + f_{32} (a_{22} - f_{21} a_{12}) \]
\[ = (a_{31} / a_{11}) a_{12} + ((a_{11} a_{32} - a_{31} a_{12}) / (a_{11} a_{22} - a_{21} a_{12})) (a_{11} a_{22} - a_{21} a_{12}) / a_{11} \]
\[ = a_{31} a_{12} / a_{11} + (a_{11} a_{32} - a_{31} a_{12}) / a_{11} = a_{32} \]
\[ f_{31} a_{13} + f_{32} (a_{23} - f_{21} a_{13}) + a_{33} - f_{31} a_{13} - f_{32} (a_{23} - f_{21} a_{13}) = a_{33} \]
• The LU decomposition thus is derivable from Gaussian elimination
• The upper triangular matrix $U$ is the set of coefficients of the matrix after elimination
• The lower triangular matrix is the set of factors used to perform the elimination, with 1’s on the main diagonal
• In general, pivoting is necessary to make this reliable
$A = \begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{bmatrix}$

$L = \begin{bmatrix}
1 & 0 & 0 \\
f_{21} & 1 & 0 \\
f_{31} & f_{32} & 1
\end{bmatrix}$

$U = \begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
0 & a_{22} - f_{21}a_{12} & a_{23} - f_{21}a_{13} \\
0 & 0 & a_{33} - f_{31}a_{13} - f_{32}(a_{23} - f_{21}a_{13})
\end{bmatrix}$

$A = LU$
\[
L = \begin{bmatrix}
1 & 0 & 0 \\
f_{21} & 1 & 0 \\
f_{31} & f_{32} & 1 \\
\end{bmatrix}
\]

L and U can share storage

\[
U = \begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
0 & a_{22} - f_{21} a_{12} & a_{23} - f_{21} a_{13} \\
0 & 0 & a_{33} - f_{31} a_{13} - f_{32} (a_{23} - f_{21} a_{13}) \\
\end{bmatrix}
\]
Pivoting

• The naïve version of the LU algorithm uses equation 1 for variable 1, equation 2 for variable 2, etc., just as the naïve version of Gaussian elimination.

• Partial Pivoting uses the equation with the largest coefficient for variable 1 to eliminate variable 1, the equation of the remaining with the largest coefficient of variable 2 to eliminate variable 2, etc.
• One way to do partial pivoting is to maintain an array order() of dimension n, which contains the list of the order in which equations are processed.

• Anytime the index $k$ refers to an equation, we substitute order($k$).

• On each major iteration, the new $k$th element of order() is set to be the equation number not already used that has the largest coefficient in absolute value.
Computational Effort

• Total effort to solve one set of linear equations $Ax=b$ by LU is the same as Gaussian elimination

• This is $O(n^3)$ for the LU decomposition and $O(n^2)$ for the substitution

• To solve many sets of equation with the same LHS is much less effort

$Ax = b_1 \quad Ax = b_2 \quad Ax = b_3 \ldots$
The Matrix Inverse

- The matrix I which has 1’s on the diagonal and zeros elsewhere has the property
  \[ AI = IA = A \]
  for any n by n matrix A

- If A is an n by n matrix, then the \textit{matrix inverse} \( A^{-1} \) is another n by n matrix such that (if it exists)
  \[ A A^{-1} = A^{-1}A = I \]
• This turns out to be easy for upper and lower triangular matrices

• If $A = LU$, and if $L^{-1}$ and $U^{-1}$ are the respective inverses, then $U^{-1}L^{-1} = A^{-1}$ because $AU^{-1}L^{-1} = LUU^{-1}L^{-1} = LL^{-1} = I$

• In general, matrix inverses are not needed; rather solution of linear equations. There are exceptions, though.
\[
\begin{bmatrix}
  a_{11} & 0 & 0 \\
  a_{21} & a_{22} & 0 \\
  a_{31} & a_{32} & a_{33}
\end{bmatrix}\begin{bmatrix}
  b_{11} \\
  b_{21} \\
  b_{31}
\end{bmatrix} = \begin{bmatrix}
  1 \\
  0 \\
  0
\end{bmatrix}
\]

\[
\begin{bmatrix}
  a_{11}b_{11} & a_{11}b_{12} & a_{11}b_{13} \\
  a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} & a_{21}b_{13} + a_{22}b_{23} \\
  a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31} & a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32} & a_{31}b_{13} + a_{32}b_{23} + a_{33}b_{33}
\end{bmatrix} = \begin{bmatrix}
  1 \\
  0 \\
  0
\end{bmatrix}
\]

\[
b_{11} = 1 / a_{11}
\]

\[
b_{12} = b_{13} = 0
\]

\[
a_{21}b_{11} + a_{22}b_{21} = 0
\]

\[
b_{21} = - \frac{a_{21}b_{11}}{a_{22}} = - \frac{a_{21}}{a_{11}a_{22}}
\]

\[
a_{21}b_{12} + a_{22}b_{22} = 1
\]

\[
b_{22} = \frac{1 - a_{21}b_{12}}{a_{22}} = 1 / a_{22}
\]

\[\text{.........} \]
Matrix Norms and Condition

- A *Norm* is a measure of the size of some object
- Euclidean norm in the plane is

\[
\| (x_1, x_2) \| = \sqrt{x_1^2 + x_2^2}
\]

\[
\| \{ a_{ij} \} \| = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^2}
\]
\[ D^2 \]

\[ X_1^2 \]

\[ X_2^2 \]
Condition Number

• One definition of the condition number of a matrix $A$ is

$$||A|| \cdot ||A^{-1}||$$

• Another definition is the ratio of the largest to the smallest eigenvalue

• Large condition number means instability of the solution of linear equations
\[
\frac{\|\Delta X\|}{\|X\|} \leq \text{Cond}(A) \frac{\|\Delta A\|}{\|A\|}
\]

\[
\begin{bmatrix}
1 & 1/2 & \cdots & 1/n \\
1/2 & 1/3 & \cdots & 1/(n+1) \\
\vdots & \vdots & \ddots & \vdots \\
1/n & 1/(n+1) & \cdots & 1/(2n-1)
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 1/2 & 1/3 \\
1/2 & 1/3 & 1/4 \\
1/3 & 1/4 & 1/5
\end{bmatrix}
= 1.41
\]

\[
\begin{bmatrix}
1 & 1/2 & 1/3 \\
1/2 & 1/3 & 1/4 \\
1/3 & 1/4 & 1/5
\end{bmatrix}^{-1} = \begin{bmatrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{bmatrix} = 372.21
\]

\[
\text{Cond} = (1.41)(372.21) = 526.16
\]
Iterative Refinement

• If one has an approximate solution to a set of linear equations, then a more nearly exact solution can be derived by iterative refinement

• This can be used when there are conditioning problems, though (as always) double precision is advisable
\[ a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \]
\[ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \]
\[ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \]

\[ a_{11}\ddot{x}_1 + a_{12}\ddot{x}_2 + a_{13}\ddot{x}_3 = \ddot{b}_1 \]
\[ a_{21}\ddot{x}_1 + a_{22}\ddot{x}_2 + a_{23}\ddot{x}_3 = \ddot{b}_2 \]
\[ a_{31}\ddot{x}_1 + a_{32}\ddot{x}_2 + a_{33}\ddot{x}_3 = \ddot{b}_3 \]

\[ x_1 = \ddot{x}_1 + \Delta x_1 \]
\[ x_2 = \ddot{x}_2 + \Delta x_2 \]
\[ x_3 = \ddot{x}_3 + \Delta x_3 \]
\[ a_{11} \Delta x_1 + a_{12} \Delta x_2 + a_{13} \Delta x_3 = b_1 - \tilde{b}_1 = E_1 \]
\[ a_{21} \Delta x_1 + a_{22} \Delta x_2 + a_{23} \Delta x_3 = b_2 - \tilde{b}_2 = E_2 \]
\[ a_{31} \Delta x_1 + a_{32} \Delta x_2 + a_{33} \Delta x_3 = b_3 - \tilde{b}_3 = E_3 \]

- Solve these equations for correction factor
- Add correction factors to approximate solution
- Especially handy with LU method because the LHS is the same at each step
Gauss-Seidel

- Instead of direct solution methods like Gaussian elimination or the LU method, one can use iterative methods.
- Often good for very large, sparse systems.
- May or may not converge; requires matrix to be *diagonally dominant*.
\[a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1\]
\[a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2\]
\[a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3\]

\[x_1 = \frac{(b_1 - a_{12}x_2 + a_{13}x_3)}{a_{11}}\]
\[x_2 = \frac{(b_2 - a_{21}x_1 - a_{23}x_3)}{a_{22}}\]
\[x_3 = \frac{(b_3 - a_{31}x_1 + a_{32}x_2)}{a_{33}}\]
\[3x_1 - 0.1x_2 - 0.2x_3 = 7.85\]
\[0.1x_1 + 7x_2 - 0.3x_3 = -19.3\]
\[0.3x_1 - 0.2x_2 + 10x_3 = 71.4\]

True solution \((3, -2.5, 7)\)

\[x_1 = (7.85 + 0.1x_2 + 0.2x_3) / 3\]
\[x_2 = (-19.3 - 0.1x_1 + 0.3x_3) / 7\]
\[x_3 = (71.4 - 0.3x_1 + 0.2x_2) / 10\]

\[x_1 = x_2 = x_3 = 0\]
\[x_1 = 7.85 / 3 = 2.616667\]
\[x_2 = [-19.3 - (0.1)(2.616667)] / 7\]
\[= -2.794524\]
\[x_3 = [71.4 - (0.3)(2.616667) + (0.2)(-2.794524)] / 10\]
\[= 7.005610\]
True solution (3, -2.5, 7)

\[ x_1 = x_2 = x_3 = 0 \]

\[ x_1 = 7.85/3 = 2.616667 \]

\[ x_2 = \left[ -19.3 - (0.1)(2.616667) \right]/7 \\
= -2.794524 \]

\[ x_3 = \left[ 71.4 - (0.3)(2.616667) + (0.2)(-2.794524) \right]/10 \\
= 7.005610 \]

\[ x_1 = 2.990557 \]

\[ x_2 = -2.499625 \]

\[ x_3 = 7.000291 \]

\[ \varepsilon_1 = .0031 \]

\[ \varepsilon_2 = .00015 \]

\[ \varepsilon_3 = .000042 \]
Convergence of Gauss-Seidel

- Gauss-Seidel converges only when the diagonal elements are much larger than the others \((\text{diagonally dominant})\) for each equation

\[
|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|
\]
Solution of Linear Systems with Packages

• The built in Excel functions MINVERSE and MMULT can be used
• The Solver can be used to solve multiple linear equations
• Matlab is the best such software with a good user interface to excellent matrix software such as Linpack and LAPack
This is the Hilbert matrix
of order 3
\[ B = \begin{bmatrix} 7 \\ 3 \\ 4 \end{bmatrix} \]

\[ B = \begin{bmatrix} 7 \\ 3 \\ 4 \end{bmatrix} \]

\[ X = A\backslash B \]

\[ X = \begin{bmatrix} 75.0000 \\ -396.0000 \\ 390.0000 \end{bmatrix} \]

The backslash operator \ solves a linear system using Gaussian elimination.
$$\begin{align*}
\text{>> } [L, U] &= \text{lu}(A) \\
L &= \\
1.0000 & 0 & 0 \\
0.5000 & 1.0000 & 1.0000 \\
0.3333 & 1.0000 & 0 \\
U &= \\
1.0000 & 0.5000 & 0.3333 \\
0 & 0.0833 & 0.0889 \\
0 & 0 & -0.0056
\end{align*}$$
\[ \begin{align*}
\text{>> } [L, U, P] &= \text{lu}(A) \\
L &= \\
&= \begin{bmatrix}
1.0000 & 0 & 0 \\
0.3333 & 1.0000 & 0 \\
0.5000 & 1.0000 & 1.0000 \\
\end{bmatrix} \\
U &= \\
&= \begin{bmatrix}
1.0000 & 0.5000 & 0.3333 \\
0 & 0.0833 & 0.0889 \\
0 & 0 & -0.0056 \\
\end{bmatrix} \\
P &= \\
&= \begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 1 \\
0 & 1 & 0 \\
\end{bmatrix}
\end{align*} \]