

MATH 3341: Introduction to Scientific Computing Lab

Melissa Butler

University of Wyoming

September 27, 2021



The background of the slide features a large, faint watermark of the University of Wyoming seal. The seal is circular with a rope-like border. Inside the border, the words "UNIVERSITY OF WY" are at the top, "EQUALITY" is in the center, and "1886" is at the bottom. There are also some stars and a small figure in the center.

Lab 06: LU Decomposition



The background of the slide features a large, faint watermark of the University of Wyoming seal. The seal is circular with a rope-like border. Inside the border, the words "UNIVERSITY OF WYOMING" are at the top, "EQUALITY" is in the center, and "1886" is at the bottom. A book is depicted in the center of the seal.

The LU Decomposition



Linear System and Its Matrix Form

Consider the system of equations

$$\begin{aligned}10x_1 - x_2 + 2x_3 &= 6 \\ -1x_1 + 11x_2 - x_3 + 3x_4 &= 25 \\ 2x_1 - x_2 + 10x_3 - x_4 &= -11 \\ 3x_2 - x_3 + 8x_4 &= 15\end{aligned}$$

In matrix form we have the equation $A\mathbf{x} = \mathbf{b}$

$$\underbrace{\begin{bmatrix} 10 & -1 & 2 & 0 \\ -1 & 11 & -1 & 3 \\ 2 & -1 & 10 & -1 \\ 0 & 3 & -1 & 8 \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 6 \\ 25 \\ -11 \\ 15 \end{bmatrix}}_{\mathbf{b}}$$



The LU decomposition allows us to factor the matrix A into two matrices, a lower triangular matrix L and an upper triangular matrix U . The LU decomposition can be viewed as the matrix form of Gaussian elimination. Computers usually solve square systems of linear equations using the LU decomposition, and it is also a key step when inverting a matrix or computing the determinant of a matrix.

$$\begin{aligned}
 A = LU &= \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}.
 \end{aligned}$$

Upper and lower triangular systems are easy to solve using forward or backward substitution algorithms.



Solve the Linear System using LU Decomposition

To solve the linear system $A\mathbf{x} = \mathbf{b}$, we perform the following:

- 1 Perform the LU decomposition of A using $[L \ U] = \text{lu}(A)$ in MATLAB.
- 2 Observe that

$$A\mathbf{x} = LU\mathbf{x} = L(U\mathbf{x}) = L\mathbf{z} = \mathbf{b} \implies L^{-1}L\mathbf{z} = \mathbf{z} = L^{-1}\mathbf{b},$$

where $\mathbf{z} = U\mathbf{x}$. In MATLAB, use $\mathbf{z} = L \setminus \mathbf{b}$ to solve for \mathbf{z} in $L\mathbf{z} = \mathbf{b}$.

- 3 Next, solve for \mathbf{x} in $U\mathbf{x} = \mathbf{z}$, we have

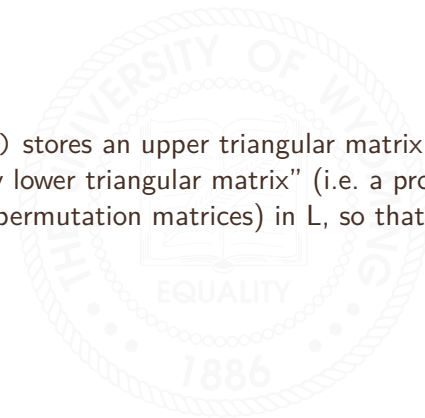
$$U\mathbf{x} = \mathbf{z} \implies U^{-1}U\mathbf{x} = I\mathbf{x} = \mathbf{x} = U^{-1}\mathbf{z}.$$

In MATLAB, use $\mathbf{x} = U \setminus \mathbf{z}$.



lu factorization.

$[L,U] = \text{lu}(A)$ stores an upper triangular matrix in U and a “psychologically lower triangular matrix” (i.e. a product of lower triangular and permutation matrices) in L , so that $A = L*U$. A can be rectangular.



\: Backslash or left matrix divide

$A \setminus B$ is the matrix division of A into B , which is roughly the same as $\text{inv}(A) * B$, except it is computed in a different way. If A is an N -by- N matrix and B is a column vector with N components, or a matrix with several such columns, then $X = A \setminus B$ is the solution to the equation $A * X = B$.



Norms

Let $\mathbf{x} = [x_1, x_2, \dots, x_n] \in \mathbb{R}^n$.

- $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|.$
- $\|\mathbf{x}\|_2 = \left(\sum_{i=1}^n x_i^2 \right)^{1/2} = (\mathbf{x} \cdot \mathbf{x})^{1/2}.$
- $\|\mathbf{x}\|_\infty = \max_{i=1, \dots, n} \{|x_i|\}.$



norm: Matrix or vector norm

- `norm(x, 1)` returns the 1-norm of `x`.
- `norm(x, 2)` returns the 2-norm of `x`.
- `norm(x, Inf)` returns the infinity norm of `x`.
- `norm(x)` is the same as `norm(x,2)`.
- Example:

```
x = -ones(4, 1)           % [-1;-1;-1;-1]
xNorm1 = norm(x, 1)       % 4
xNorm2 = norm(x, 2)       % 2
xNormInf = norm(x, Inf)   % 1
xNorm2 == dot(x, x)^(1/2) % logical 1 (true)
```

