Linear Systems

LU Decomposition

Sachin Shanbhag

Department of Scientific Computing
Florida State University,
Tallahassee, FL 32306.

References

- S. Chapra and R. Canale, Numerical Methods for Engineers
- A. Greenbaum and T. Chartier, Numerical Methods: Design, Analysis, and Computer Implementation of Algorithms
- M. Heath, Scientific Computing: An Introductory Survey
- wikipedia.org

Linear Systems of Equation

- We saw how to solve $\mathbf{Ax} = \mathbf{b}$ using Gauss elimination (GE). Recall, how most of the effort was spent in the "elimination" part.
- There are many scenarios where we have to repeatedly solve for $\mathbf{Ax} = \mathbf{b_1}$, $\mathbf{Ax} = \mathbf{b_2}$, $\mathbf{Ax} = \mathbf{b_3}$. (ex. iterative solutions, next)
- In such cases it may be advantageous to do the elimination part of GE once, and then solve a cheaper substitution problem over and over again.
- One way to accomplish this is LU decomposition - where we decompose the matrix $\mathbf{A} = \mathbf{LU}$
- Interestingly, GE itself can be expressed as an LU decomposition

Overview

- Given $\mathbf{Ax} = \mathbf{b}$, we decompose $\mathbf{A} = \mathbf{LU}$, where $\mathbf{L}$ and $\mathbf{U}$ are lower and upper triangular matrices.
- Decomposition:

$$\begin{aligned} \mathbf{Ax} &= \mathbf{b} \\ \mathbf{L}(\mathbf{Ux}) &= \mathbf{b} \end{aligned}$$

- If we set $\mathbf{Ux} = \mathbf{d}$, we can solve for $\mathbf{x}$ by back and forward substitution.
- Substitution:

$$\begin{aligned} \mathbf{Ld} &= \mathbf{b} \\ \mathbf{Ux} &= \mathbf{d} \end{aligned}$$

Naive LU Decomposition

- Let us first consider how to use GE to do LU decomposition without worrying about pivoting and scaling
- For simplicity, let us consider a simple 3×3 matrix $\mathbf{A}$
- That is:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}.$$

- Note the ones along the diagonal of $\mathbf{L}$. This form is called the Dolittle decomposition. If we require the diagonal elements of $\mathbf{U}$ to be 1, then it is called Crout's decomposition.

LU from GE

- Consider the system:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

- In GE, we zero the element a_{21} by performing the row operation $\mathbf{E}_2' = \mathbf{E}_2 - (a_{21}/a_{11})\mathbf{E}_1$
- Let $f_{21} = (a_{21}/a_{11})$, and similarly let $f_{31} = (a_{31}/a_{11})$ (used to zero out a_{31})
- At this point the matrix $\mathbf{A}$ looks like:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}' & a_{23}' \\ 0 & a_{32}' & a_{33}' \end{bmatrix}$$

LU from GE

- We then zero out the element a'_{32} by using the "factor" $f_{32} = a'_{32}/a'_{22}$, to end up with an upper triangular matrix

$$\mathbf{U} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{bmatrix}$$

- It turns out that

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

Example 1: LU Decomposition

- Consider[*]

$$\mathbf{A} = \begin{bmatrix} 25 & 5 & 1 \\ 64 & 8 & 1 \\ 144 & 12 & 1 \end{bmatrix}$$

- $f_{21} = 64/25 = 2.56$, and $f_{31} = 144/25 = 5.76$
- Performing elementary row operations using these factors in GE ($\mathbf{E}_2' = \mathbf{E}_2 - f_{21}\mathbf{E}_1$ etc.) we get the matrix:

$$\begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & -16.8 & -4.76 \end{bmatrix}$$

- We proceed to the third row, $f_{32} = -16.8/-4.8 = 3.5$

[*] `http://numericalmethods.eng.usf.edu/`

Example 1

- After the corresponding elementary row operation yields an upper triangular form:

$$\mathbf{U} = \begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & 0 & 0.7 \end{bmatrix}$$

- And

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 2.56 & 1 & 0 \\ 5.76 & 3.5 & 1 \end{bmatrix}$$

- Does $\mathbf{LU} = \mathbf{A}$? Check for yourself!

Example 2: Solving Equations with LU

- Consider the system

$$\begin{bmatrix} 25 & 5 & 1 \\ 64 & 8 & 1 \\ 144 & 12 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 106.8 \\ 177.2 \\ 279.2 \end{bmatrix}$$

- We have already done the LU-decomposition of the matrix, hence we first solve for $\mathbf{Ld} = \mathbf{b}$

$$\begin{bmatrix} 1 & 0 & 0 \\ 2.56 & 1 & 0 \\ 5.76 & 3.5 & 1 \end{bmatrix} \mathbf{d} = \begin{bmatrix} 106.8 \\ 177.2 \\ 279.2 \end{bmatrix}$$

Example 2

- Using forward substitution, we get

$$\mathbf{d} = \begin{bmatrix} 106.8 \\ -96.21 \\ 0.735 \end{bmatrix}$$

- Now we solve the last part $\mathbf{Ux} = \mathbf{d}$ by backward substitution

$$\begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & 0 & 0.7 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 106.8 \\ -96.21 \\ 0.735 \end{bmatrix}$$

- Yielding $\mathbf{x} = [0.29,\ 19.70,\ 1.05]^T$

Notes on LU decomposition

- To save memory, well-written LU decomposition routines return the **L** and **U** matrices *within* the original matrix **A**, since the ones and the zeros don't have to be stored.
- We can keep track of pivoting using an "order" vector **o**.
- Matlab, for example, stores the "order vector" as a permutation matrix $\mathbf{P}$ so that the command:

  ```
  > [L, U, P] = lu(A)
  ```

  produces a factorization:

  $$\mathbf{PA} = \mathbf{LU}$$

- The permutation matrix contains only the elements 0 and 1

Example with Random Matrix

```
octave:1> A=rand(3,3)
A =

   0.20470   0.58273   0.20475
   0.93541   0.75733   0.25327
   0.12666   0.45558   0.77896

octave:2> [L,U,P]=lu(A)
L =

   1.00000   0.00000   0.00000
   0.21884   1.00000   0.00000
   0.13541   0.84661   1.00000

U =

   0.93541   0.75733   0.25327
   0.00000   0.41700   0.14933
   0.00000   0.00000   0.61825

P =

Permutation Matrix

   0   1   0
   1   0   0
   0   0   1
```

Example with Random Matrix

```
octave:3> L*U
ans =

   0.93541   0.75733   0.25327
   0.20470   0.58273   0.20475
   0.12666   0.45558   0.77896

octave:4> P*A
ans =

   0.93541   0.75733   0.25327
   0.20470   0.58273   0.20475
   0.12666   0.45558   0.77896
```

Notes on LU decomposition

- We don't usually rescale the equations; instead we use scaled values only while pivoting
- To protect against singular matrices, we check the magnitude of the diagonal elements.
- If they are close to zero (determined by `tol`), then we pass an error code