#Lecture 5

~~I~~

Inverses:

what is the inverse of AB?

$$(AB)^{-1} = B^{-1}A^{-1}$$

$$(ABC)^{-1} = C^{-1}B^{-1}A^{-1}$$

Factorization: $A = LU$

Our elimination matrix E takes A to U. We will show how reversing those steps is achieved by a lower triangular matrix L. Let's work with a 2x2 matrix –

$$\cancel{\begin{bmatrix} 1 & 0 \\ -4 & 1 \end{bmatrix} \overset{A}{\begin{bmatrix} 2 & 1 \\ 8 & 7 \end{bmatrix}} =}$$

$$\overset{E_{21}}{\begin{bmatrix} 1 & 0 \\ -4 & 1 \end{bmatrix}} \overset{A}{\begin{bmatrix} 2 & 1 \\ 8 & 7 \end{bmatrix}} = \overset{U}{\begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}}$$

$$\Rightarrow \quad A = E_{21}^{-1} U = LU$$

$$\underset{A}{\begin{bmatrix} 2 & 1 \\ 8 & 7 \end{bmatrix}} = \underset{L}{\begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix}} \underset{U}{\begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}}$$

$L \downarrow$ Low Triangula

L will always have 1 as pivots. U may or may not have one as pivots, but we can ~~extract it~~ separate up the pivots.

$$= \underset{L}{\begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix}} \underset{D}{\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}} \underset{U}{\begin{bmatrix} 1 & 1/2 \\ 0 & 1 \end{bmatrix}}$$

$D \uparrow$ Pivots

Let's work with a 3×3 example -

For eliminations -

$$E_{32}E_{31}E_{21}A = U \text{ (no row exchange)}$$

$$A = E_{21}^{-1}E_{31}^{-1}E_{32}^{-1}U = LU$$

Now, L has a better shape than E. for Example -

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 10 & -5 & 1 \end{bmatrix} \qquad EA = U$$

E_{32} $\quad$ E_{21} $\quad$ E

How did it get here? The effect of Row 1 on Row 3.

Inverses -

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 5 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 5 & 1 \end{bmatrix} \qquad A = LU$$

E_{21}^{-1} $\quad$ E_{32}^{-1} $\quad$ L

So, my point here is, in lower triangular matrix L, the multipliers come in the right form. So to form L, we have no work to do ~~ex~~ except finding out the inverses.

So, for A=LU, if ~~~~ no row exchanges, the multipliers go directly into L. So, matrix E is not particularly attractive, but matrix L is.

Now, the question here is, why do we need LU factorization?

In many engineering applications, when you solve $Ax = b$, the matrix $A \in \mathbb{R}^{N\times N}$ remains unchanged, while the right hand side vector b keeps changing. Examples -

1. when you're solving a partial differential equation for different forcing function. For these different forcing functions, the meshing is usually kept the same.

2. when you're solving a time dependent problem, where the unknown evolve with time.

The key idea behind LU factorization is to decouple the factorization phase from the actual solving phase. The factorization phase only needs the matrix A, while the actual solving phase makes use of the factored form of A and the right hand side to solve the linear system. Hence once we have the factorization, we can make use of the factored form of A; to solve for different right hand sides at a relatively moderate computational cost.

The cost of factorizing the matrix A into LU is $O(N^3)$.

Not only this. A lot of matrix ~~ma~~ operations are easier for triangular matrices. 'Easier' here meant that the time-complexity for a computer to calculate the result will be lower.

One clear example is calculating the determinant of a matrix. If we have LU factorization, $\det(A) = \det(L)\det(U)$

However, for a triangular matrix, the determinant is just the product of its diagonal entries. So we just need $(2n-1)$ multiplications to get the result.

Once you've ~~solved two~~ the factorization, the cost of solving $LUx=b$ is just $O(N^2)$. So if you have a total of 'r' right hand sides, the total complexity would be -

$$O(N^3+rN^2)$$

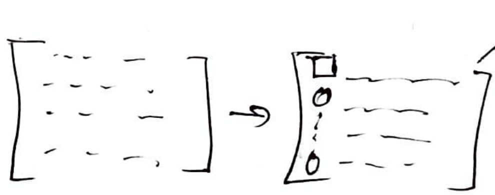
On the other hand, Gaussian elimination independently costs $O(N^3)$, so for 'r' systems, the complexity is $O(rN^3)$.

However, typically when people say Gauss elimination, they usually refer to LU decomposition.

Let's calculate the complexity.

How many operations do we need for a $n \times n$ matrix while elimination?
↳ (multiply+subtract)

Lets say $n=100$, so for the first step



A typical operation is to multiply one row and then subtract it from another, which requires on the order of n operations. There are n rows, so the total number of operations used in eliminating entries in the ~~fris~~ first column is n^2.

The count is → $n^2+(n-1)^2+\dots+3^2+2^2+1^2 = \frac{n(n+1)(2n+1)}{6} \approx n^3$

And the cost of b is n^2