Find the Point of Intersection of Two Lines

S A Aravind Eswar - EE24BTECH11053

January 21, 2025

Problem Statement

Solve the following set of equations,

$$5x - 3y = 11 \tag{2.1}$$

$$-10x + 6y = -22 \tag{2.2}$$

Theoretical Solution

By observing the equations, we can tell that they are a pair of co-incident lines.

$$\frac{5}{-10} = \frac{-3}{6} = \frac{11}{-22} = -2 \tag{3.1}$$

Thus there are infinite number of solutions to the given pair of equations.

Method of solving I

We can solve a given set of linear equation using LU decomposition.
Let,

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = a_0 \tag{3.2}$$

$$b_1x_1 + b_2x_2 + \cdots + b_nx_n = b_0 \tag{3.3}$$

$$\vdots \tag{3.4}$$

$$p_1x_1 + p_2x_2 + \cdots + p_nx_n = p_0 \tag{3.5}$$

Be a system of equations with n variables and n equations.
Then we can write the Given system of equations as,

$$A\mathbf{x} = \mathbf{b} \tag{3.6}$$

Method of solving II

where,

$$A = \begin{pmatrix} a_1 & a_2 & \dots & a_n \\ b_1 & b_2 & \dots & b_n \\ \vdots & \vdots & \vdots & \vdots \\ p_1 & p_2 & \dots & p_n \end{pmatrix} \tag{3.7}$$

$$\mathbf{b} = \begin{pmatrix} a_0 \\ b_0 \\ \vdots \\ p_0 \end{pmatrix} \tag{3.8}$$

$$\tag{3.9}$$

Method of solving III

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \tag{3.10}$$

We can decompose,

$$A = LU \tag{3.11}$$

Where L is a lower triangular matrix and U is a upper triangular matrix.
Now the equation can be written as,

$$LU\mathbf{x} = \mathbf{b} \tag{3.12}$$

Method of solving IV

Writing $U\mathbf{x} = \mathbf{y}$

$$L\mathbf{y} = \mathbf{b} \tag{3.13}$$

Using back Substitution Method we can solve for $\mathbf{y}$
Now,

$$U\mathbf{x} = \mathbf{y} \tag{3.14}$$

Using forward Substitution, we can compute $\mathbf{x}$

Gaussian Elimination I

One way to perform LU decomposition is using Gaussian Elimination. Gaussian Elimination can be written in the following way.
For a matrix,

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \tag{3.15}$$

Given that the diagonal elements aren't zero, we can reduce the first column as,

Gaussian Elimination II

$$R_i = R_i - \frac{a_{i1}}{a_{11}} R_1 \tag{3.16}$$

$$R_i = R_i - l_{i1} R_1 \tag{3.17}$$

where $i > 1$
Similarly reduing other $n-2$ columns, A is transformed into an upper triangular matrix U.
In general, we can write it as,

$$R_i = R_i - \frac{a_{ij}}{a_{jj}} R_1 \tag{3.18}$$

$$R_i = R_i - l_{ij} R_1 \tag{3.19}$$

Gaussian Elimination III

L is given by,

$$L = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{21} & 1 & 0 & \dots & 0 \\ l_{31} & l_{32} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \dots & 1 \end{pmatrix} \tag{3.20}$$

Where,

$$l_{ij} = \frac{a_{ij}}{a_{ii}} \tag{3.21}$$

If the diagonal elements are zero, then we multiply A with a permutation matrix P and pivot it such that the diagonal elements become non-zero.

Gaussian Elimination Algorithm

for i in range(n) **do**
  for j in range($n-1$, i, -1) **do**
    $l_{ji} = A_{j,i}/A_{i,i}$
    $U_{j,:} = A_{j,:} - l_{ji}A_{i,:}$
    $L_{j,i} = l_{ji}$
  end for
end for

Doolittle's Algorithm I

Doolittle's algorithm provides a more elegant and better method to perform LU decomposition.
Doolittle's algorithm is given by,
For U

$$\forall j \tag{3.22}$$

$$i = 0 \rightarrow U_{ij} = A_{ij} \tag{3.23}$$

$$i > 0 \rightarrow U_{ij} = A_{ij} - \sum_{k=0}^{i-1} L_{ik} U_{kj} \tag{3.24}$$

Doolittle's Algorithm II

For L

$$\forall i \tag{3.25}$$

$$j = 0 \to L_{ij} = \frac{A_{ij}}{U_{jj}} \tag{3.26}$$

$$j > 0 \to L_{ij} = \frac{A_{ij} - \sum_{k=0}^{j-1} L_{ik} U_{kj}}{U_{jj}} \tag{3.27}$$

After performing LU decomposition, we get,

$$A = LU \tag{3.28}$$

Doolittle's Algorithm III

where,

$$L = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix} \tag{3.29}$$

$$U = \begin{pmatrix} 5 & -3 \\ 0 & 0 \end{pmatrix} \tag{3.30}$$

Doolittle Algorithm I

$L,\ U \leftarrow n \times n$ zero arrays
for i in range(n) **do**
  for k in range(i, n) **do**
    $sum \leftarrow 0$
    for j in range(i) **do**
      $sum \leftarrow sum + L_{i,j}\, U_{j,k}$
    end for
    $U_{i,k} \leftarrow A_{i,k} - sum$
  end for
  for k in range($i,\ n$) **do**
    if $i = k$ **then**
      $L_{i,k} \leftarrow 1$
    else
      $sum \leftarrow 0$

Doolittle Algorithm II

```
            for j in range(i) do
                sum ← sum + L_{k,j} U_{j,i}
            end for
            L_{k,i} ← (A_{k,i} − sum) − U_{i,i}
        end if
    end for
end for
```

Plotting the curve