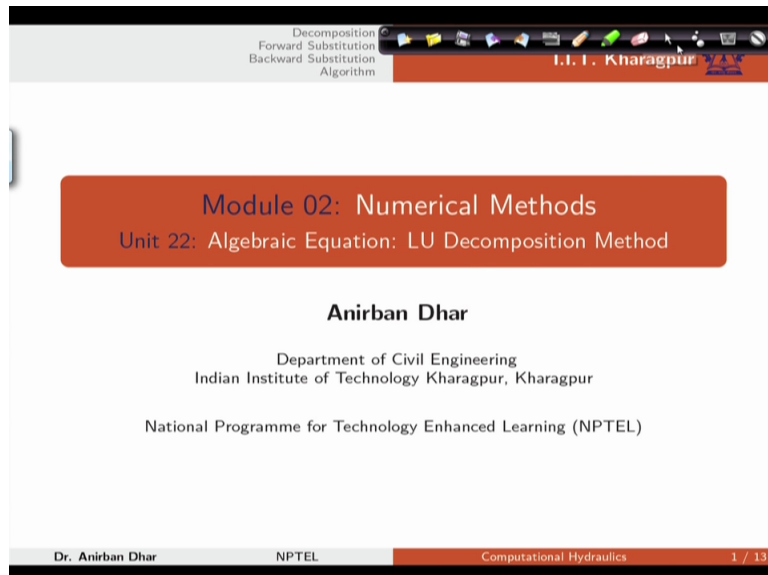


**Computational Hydraulics**  
**Professor Anirban Dhar**  
**Department of Civil Engineering**  
**Indian Institute of Technology Kharagpur**  
**Lecture 26**  
**Algebraic Equation: LU Decomposition Method**

Welcome to this lecture number 26 of the course computational hydraulics and we are in module 2, numerical methods. And in this particular lecture we will discuss unit 22, algebraic equation and we will try to discuss LU decomposition method for decomposition of matrixes to get the solution using direct numerical approach.

(Refer Slide Time 00:55)



The image shows a presentation slide with a white background and a dark blue header. The header contains the text "Decomposition", "Forward Substitution", "Backward Substitution", and "Algorithm" on the left, and "I.I.T. Kharagpur" on the right. The main content of the slide is centered and includes the following text: "Module 02: Numerical Methods", "Unit 22: Algebraic Equation: LU Decomposition Method", "Anirban Dhar", "Department of Civil Engineering", "Indian Institute of Technology Kharagpur, Kharagpur", and "National Programme for Technology Enhanced Learning (NPTEL)". At the bottom of the slide, there is a footer with "Dr. Anirban Dhar", "NPTEL", "Computational Hydraulics", and "1 / 13".

So what is the learning objective? At the end of this particular unit students will be able to apply LU decomposition method for direct solution.

(Refer Slide Time 01:11)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

### Learning Objective

- To apply LU Decomposition Method for direct solution.

Dr. Anirban Dhar NPTEL Computational Hydraulics 2 / 13

So we already know what is our matrix form? We have A as constant coefficient matrix which is a square matrix. This is our variable vector and r is right hand side vector. In this case we have again a square matrix format, N cross N and N cross 1, then again N cross 1 here. So these are compatible matrices.

(Refer Slide Time 01:53)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

### Matrix Form

Full Matrix

$$A\phi = r$$

$$\begin{pmatrix} \times & \times & \times & \dots & \times & \times & \times \\ \times & \times & \times & \dots & \times & \times & \times \\ \times & \times & \times & \dots & \times & \times & \times \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \times & \times & \times & \dots & \times & \times & \times \\ \times & \times & \times & \dots & \times & \times & \times \\ \times & \times & \times & \dots & \times & \times & \times \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_{N-2} \\ \phi_{N-1} \\ \phi_N \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_{N-2} \\ r_{N-1} \\ r_N \end{pmatrix}$$

$N \times N$        $N \times 1$        $N \times 1$

Dr. Anirban Dhar NPTEL Computational Hydraulics 3 / 13

In this case our LU decomposition process it has got few steps. First step is decomposition step. We will decompose this A matrix into one lower and one upper triangular matrix. Next step is forward substitution. So in this process we will utilize our lower triangular matrix for forward substitution step. And this psi, this is some kind of secondary variable that we will utilize for forward substitution.

So in this process essentially we are solving  $L\psi = r$  equation. And we can get this  $\psi$  value here. But  $\psi$  is not our ultimate variable. We need to get the  $\phi$ .

(Refer Slide Time 03:18)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

### Basic Steps

LU Decomposition

- Decomposition:  $A = LU$
- Forward Substitution:  $L\psi = r$

$\psi = ?$

Dr. Anirban Dhar NPTEL Computational Hydraulics 4 / 13

So in this last step, backward substitution step, we will utilize  $U\phi = \psi$ . So  $U$  is the upper triangular matrix that is generated from our decomposition step and  $\phi$  is a variable vector. At  $\psi$  whatever we have calculated from our substitution or forward substitution step or second step, we will directly utilize it in the right hand side vector and we will calculate the  $\phi$  value which is our desired value.

(Refer Slide Time 04:05)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

### Basic Steps

LU Decomposition

- Decomposition:  $A = LU$
- Forward Substitution:  $L\psi = r$
- Backward Substitution:  $U\phi = \psi$

Dr. Anirban Dhar NPTEL Computational Hydraulics 4 / 13

So what is the basis for this one? Basis is that if we multiply this we can write this for this step as  $U\phi - \psi = 0$ . Now this step if you multiply  $L$  in this case, so this is essentially first multiplication it is coming as  $LU$ .  $LU$  is  $A$ . Again  $L\psi$  from our forward substitution step, this is  $r$ . So essentially we are solving the same problem but with multiple steps. So  $LU$  equals to  $A$  and  $L\psi$  is equal to  $r$ .

(Refer Slide Time 05:11)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

### Basic Steps

LU Decomposition

- **Decomposition:**  $A = LU$
- **Forward Substitution:**  $L\psi = r$
- **Backward Substitution:**  $U\phi = \psi \Rightarrow L(U\phi - \psi) = 0$

Overall calculation can be presented as

$$L(U\phi - \psi) = \underbrace{LU}_{A}\phi - \underbrace{L\psi}_{r} = A\phi - r$$

with

$$\begin{aligned} \underline{LU} &= \underline{A} \\ \underline{L\psi} &= \underline{r} \end{aligned}$$

in Dhar NPTEL Computational Hydraulics 4 / 13

Now we will try to utilize this concept and we will try to implement this in algorithm format. So if we see our Gauss elimination thing, in LU decomposition we can utilize our Gauss elimination concept. So what is the basis? Basis is that matrix generated from forward elimination process is this one, where first one is unchanged, second one is changed, third one changed, fourth one and fifth one, these are changed values of the coefficient matrix. And we have generated upper triangular matrix out of this Gauss elimination or forward elimination process.



(Refer Slide Time 06:22)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

### Gauss Elimination

LU Decomposition

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{pmatrix}$$

Matrix form generated from forward elimination process

$$\begin{pmatrix} u_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a'_{22} & a'_{23} & a'_{24} & a'_{25} \\ 0 & 0 & a''_{33} & a''_{34} & a''_{35} \\ 0 & 0 & 0 & a'''_{44} & a'''_{45} \\ 0 & 0 & 0 & 0 & a^{IV}_{55} \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix} = \begin{pmatrix} r_1 \\ r'_2 \\ r''_3 \\ r'''_4 \\ r^{IV}_5 \end{pmatrix}$$

an Dhar NPTEL Computational Hydraulics 5 / 13

Now in this casewe can see that we have zero values stored in this lower triangular portion.

(Refer Slide Time 06:34)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

### Gauss Elimination

LU Decomposition

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{pmatrix}$$

Matrix form generated from forward elimination process

$$\begin{pmatrix} u_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a'_{22} & a'_{23} & a'_{24} & a'_{25} \\ 0 & 0 & a''_{33} & a''_{34} & a''_{35} \\ 0 & 0 & 0 & a'''_{44} & a'''_{45} \\ 0 & 0 & 0 & 0 & a^{IV}_{55} \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix} = \begin{pmatrix} r_1 \\ r'_2 \\ r''_3 \\ r'''_4 \\ r^{IV}_5 \end{pmatrix}$$

an Dhar NPTEL Computational Hydraulics 5 / 13

Now in this process, in the first step gamma 12. That means gamma multiplied with first row for second row, gamma multiplied with first row for third row, gamma multiplied with first row for fourth and fifth row were multiplied with row 1. So in this case multiplication factors can be stored. So multiplication factors in this case stored like this. This is gamma 1 2, gamma 13, gamma 14, gamma 15.

(Refer Slide Time 08:15)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

### Gauss Elimination

LU Decomposition

In the first step  $\gamma_1^2, \gamma_1^3, \gamma_1^4, \gamma_1^5$  were multiplied for Rows 2, 3, 4, and 5 respectively.

The multiplication factors can be stored as

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ \gamma_1^2 & a'_{22} & a'_{23} & a'_{24} & a'_{25} \\ \gamma_1^3 & 0 & a''_{33} & a''_{34} & a''_{35} \\ \gamma_1^4 & 0 & 0 & a'''_{44} & a'''_{45} \\ \gamma_1^5 & 0 & 0 & 0 & a''''_{55} \end{pmatrix}$$

an Dhar NPTEL Computational Hydraulics 6 / 13

Similarly for this casewe can again store other values. Other values are generated from different steps. These are actually factors that we have utilized to multiply our reference row.

(Refer Slide Time 08:44)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

### Gauss Elimination

LU Decomposition

In the first step  $\gamma_1^2, \gamma_1^3, \gamma_1^4, \gamma_1^5$  were multiplied for Rows 2, 3, 4, and 5 respectively.

The multiplication factors can be stored as

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ \gamma_1^2 & a'_{22} & a'_{23} & a'_{24} & a'_{25} \\ \gamma_1^3 & 0 & a''_{33} & a''_{34} & a''_{35} \\ \gamma_1^4 & 0 & 0 & a'''_{44} & a'''_{45} \\ \gamma_1^5 & 0 & 0 & 0 & a''''_{55} \end{pmatrix}$$

Similarly,

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ \gamma_1^2 & a'_{22} & a'_{23} & a'_{24} & a'_{25} \\ \gamma_1^3 & \gamma_2^2 & a''_{33} & a''_{34} & a''_{35} \\ \gamma_1^4 & \gamma_2^4 & \gamma_3^4 & a'''_{44} & a'''_{45} \\ \gamma_1^5 & \gamma_2^5 & \gamma_3^5 & \gamma_4^5 & a''''_{55} \end{pmatrix}$$

an Dhar NPTEL Computational Hydraulics 6 / 13

Now for this one, LU decomposition step we can write it as, this is our upper triangular matrix or U. And this is our lower triangular matrix with diagonal term 1, 1, 1, 1 and the factors are stored in the lower portion. These are all factors that we have utilized. And this is the thing that we have got from our Gauss elimination process.

(Refer Slide Time 09:30)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

### Gauss Elimination

LU Decomposition

where

$$A \leftarrow LU$$

$$U = \begin{pmatrix} a'_{11} & a'_{12} & a'_{13} & a'_{14} & a'_{15} \\ 0 & a''_{22} & a''_{23} & a''_{24} & a''_{25} \\ 0 & 0 & a'''_{33} & a'''_{34} & a'''_{35} \\ 0 & 0 & 0 & a''''_{44} & a''''_{45} \\ 0 & 0 & 0 & 0 & a''''_{55} \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 & 0 \\ l_{41} & l_{42} & l_{43} & 1 & 0 \\ l_{51} & l_{52} & l_{53} & l_{54} & 1 \end{pmatrix}$$

an Dhar NPTEL Computational Hydraulics 7 / 13

Now in this case forward substitution, we can say that in place of gamma we are writing it in a more structured way. That  $l_{21}$  this is essentially  $\gamma_{12}$ . So why this is forward substitution? So we are directly substituting the value in this case  $\psi_1$  equals to your  $r_1$ . And in this case next is  $\psi_2$  which is  $r_2$  minus  $\psi_1$  and this  $l_{21}$ .

(Refer Slide Time 10:24)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

### Gauss Elimination

Substitution Step

Forward Substitution

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 & 0 \\ l_{41} & l_{42} & l_{43} & 1 & 0 \\ l_{51} & l_{52} & l_{53} & l_{54} & 1 \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \\ \psi_5 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{pmatrix}$$

$\psi_1 = r_1$   
 $\psi_2 = r_2 - l_{21}\psi_1$

an Dhar NPTEL Computational Hydraulics 8 / 13

So obviously we can utilize our values which we have calculated in our first decomposition step using Gauss elimination and we can get this forward substitution. Now in this case general algorithm is like this.  $\psi_i$  can be generated  $r_i$  minus  $\sum_{j=1}^{i-1} l_{ij}\psi_j$  because for  $i$  minus 1 number of variables only we will have updated values available and our matrix that is a lower triangular matrix. So we will consider  $i$  values starting from 2 to  $N$ .

Now this means that for all i within this set 2 to N, we can utilize this. So this is i minus 1. For 2 this will be up to 1. That means only one term. If i is equal to 3, so obviously that means these two values should be available. That's why this j equals to 1 to 2.

(Refer Slide Time 11:57)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

## Gauss Elimination

Substitution Step

**Forward Substitution**

$$2 \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 & 0 \\ l_{41} & l_{42} & l_{43} & 1 & 0 \\ l_{51} & l_{52} & l_{53} & l_{54} & 1 \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \\ \psi_5 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{pmatrix}$$

**General Algorithm**

$$\psi_1 = r_1$$

$$\psi_i = r_i - \sum_{j=1}^{i-1} a_{ij} \psi_j, \quad \forall i \in \{2, 3, \dots, N\}$$

in Dhar    NPTEL    Computational Hydraulics    8 / 13

Now in this case the next step is backward substitution. Backward substitution again we can directly write in terms of phi N. Phi N is psi N divided by a NN. A NN because in this case we are storing all values in our A matrix and in general because psi values are calculated values. That's why with the red color. And phi j values, these are also calculated values. Because for i if you are considering i equals to 4, so obviously this should be phi to 5. That means only one term.

(Refer Slide Time 12:54)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

## Gauss Elimination

Substitution Step

**Backward Substitution**

$$\begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} & u_{15} \\ 0 & u_{22} & u_{23} & u_{24} & u_{25} \\ 0 & 0 & u_{33} & u_{34} & u_{35} \\ 0 & 0 & 0 & u_{44} & u_{45} \\ 0 & 0 & 0 & 0 & u_{55} \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix} = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \\ \psi_5 \end{pmatrix}$$

**General Algorithm**

$$\phi_N = \frac{\psi_N}{a_{NN}}$$

$$\phi_i = \frac{1}{a_{ii}} \left[ \psi_i - \sum_{j=i+1}^N a_{ij} \phi_j \right], \quad \forall i \in \{N-1, N-2, \dots, 1\}$$

in Dhar    NPTEL    Computational Hydraulics    9 / 13

So our calculation is valid from N minus 1 to 1. And with this general steps we can directly calculate our values. Obviously I have not utilized this U 55 here because we are storing all the values in a 11 to a nn format where a n1 to a 1n. That means N into N one matrix is our A matrix. So we are utilizing all the values here for our calculations.

(Refer Slide Time 13:51)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

### Gauss Elimination Substitution Step

$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix}$   $N \times N$

**Backward Substitution**

$$\begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} & u_{15} \\ 0 & u_{22} & u_{23} & u_{24} & u_{25} \\ 0 & 0 & u_{33} & u_{34} & u_{35} \\ 0 & 0 & 0 & u_{44} & u_{45} \\ 0 & 0 & 0 & 0 & u_{55} \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix} = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \\ \psi_5 \end{pmatrix}$$

**General Algorithm**

$$\phi_N = \frac{\psi_N}{u_{NN}}$$

$$\phi_i = \frac{1}{a_{ii}} \left[ \psi_i - \sum_{j=i+1}^N a_{ij} \phi_j \right], \quad \forall i \in \{N-1, N-2, \dots, 1\}$$

Anirban Dhar NPTEL Computational Hydraulics 9 / 13

Now let us see what is there in the algorithm of our LU decomposition. In LU decomposition this step is similar to our Gauss elimination decomposition step. So same we are starting from reference row and we are going up to n minus 1 and i, k plus 1, that means leaving that reference row we are considering other rows for calculation. And we are generating this gamma.

(Refer Slide Time 14:37)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

Data: Matrix A, Vector r  
Result:  $\phi$

**Decomposition**

```

for k=1,n-1 do
  for i=k+1,n do
    for j=k+1,n do
       $\gamma = a_{i,k} / a_{k,k}$ 
       $a_{i,k} = \gamma$ 
      for j=k+1,n do
         $a_{i,j} = a_{i,j} - \gamma \cdot a_{k,j}$ 
      end
    end
  end
end

```

**Forward Substitution**

```

 $\psi_1 = r_1$ 
for i=2,n do
  sum=0
  for j=1,i-1 do
    sum=sum+a_{i,j} *  $\psi_j$ 
  end
   $\psi_i = \text{sum}$ 
end

```

**Back Substitution**

```

 $\phi_n = \psi_n / a_{n,n}$ 
for i=n-1,1,1 do
  sum=0
  for j=i+1,n do
    sum=sum+a_{i,j} *  $\phi_j$ 
  end
   $\phi_i = \text{sum} / a_{i,i}$ 
end
return  $\phi$ 

```

Dr. Anirban Dhar NPTEL Computational Hydraulics 10 / 13

So gamma is your actor ik and divided by kk. So ik again we are storing this gamma because this is for storage of lower triangular matrix here. And this process is as usual  $a_{ij}$ ,  $a_{ij}$  minus gamma  $a_{kj}$ . Now in this process please remember one thing that we are not considering changes in r. So leaving r, other calculations are same. Only thing is that this step where we are storing aik. In place of aik we are storing this gamma factor values.

(Refer Slide Time 15:33)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

Data: Matrix A, Vector r  
Result:  $\phi$

```

Decomposition
for k=1,n-1 do
  for i=k+1,n do
     $\gamma = a_{i,k} / a_{k,k}$ 
     $a_{i,k} = \gamma$ 
    for j=k+1,n do
       $a_{i,j} = a_{i,j} - \gamma \cdot a_{k,j}$ 
    end
  end
end
end

Forward Substitution
 $\psi_1 = r_1$ 
for i=2,n do
  sum = ri
  for j=1,i-1 do
    sum = sum - ai,j ·  $\psi_j$ 
  end
   $\psi_i = sum$ 
end

Back Substitution
 $\phi_n = \psi_n / a_{n,n}$ 
for i=n-1,1 do
  sum =  $\psi_i$ 
  for j=i+1,n do
    sum = sum - ai,j ·  $\phi_j$ 
  end
   $\phi_i = sum / a_{i,i}$ 
end
return  $\phi$ 

```

Dr. Anirban Dhar NPTEL Computational Hydraulics 10 / 13

So next step is forward substitution,  $\psi_1$  equals to  $r_1$ . And we are running this loop from 2 to N. Sum equals to  $r_i$ , j equals to 1 to i minus 1. That means to consider the diagonal term only. Corresponding to diagonal term we are calculating the psi values. So psi should be sum directly, because we have considered initial value as i and from that we are subtracting other j related values. So directly we are getting  $\psi_i$  equals to sum. Because in diagonal term for lower triangular matrix only unity is there. So no division is required.

(Refer Slide Time 16:34)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

Data: Matrix A, Vector r  
Result:  $\phi$

```

Decomposition
for k=1,n-1 do
  for i=k+1,n do
     $\gamma = a_{i,k} / a_{k,k}$ 
     $a_{i,k} = \gamma$ 
    for j=k+1,n do
       $a_{i,j} = a_{i,j} - \gamma \cdot a_{k,j}$ 
    end
  end
end
Forward Substitution
 $\psi_1 = r_1$ 
for i=2,n do
  sum = r_i
  for j=1,i-1 do
    sum = sum - a_{i,j} \cdot \psi_j
  end
   $\psi_i = \text{sum}$ 
end
Back Substitution
 $\phi_n = \psi_n / a_{n,n}$ 
for i=n-1,1 do
  sum =  $\psi_i$ 
  for j=i+1,n do
    sum = sum - a_{i,j} \cdot \phi_j
  end
   $\phi_i = \text{sum} / a_{i,i}$ 
end
return  $\phi$ 
  
```

Dr. Anirban Dhar NPTEL Computational Hydraulics 10 / 13

Next is backward or back substitution. Back substitution  $\phi_n$  equals to  $\psi_n$  divided by  $a_{n,n}$ . And this is more or less similar to our Gauss elimination process. So except the  $\psi_i$ , in place of  $\psi_i$  this  $r$  was there. So in this case  $\psi_i$  we are placing here, so sum equals to sum minus  $a_{ij} \phi_j$ .  $\phi_j$  is already calculated value from our calculation process. So in this case coefficient terms are there. Because diagonal term we have upper triangular matrix including the diagonal term.

So we need to divide it with  $a_{ii}$  which is part of upper triangular matrix. So  $\phi_i$  equals to sum divided by  $a_{ii}$ . Now in this process we have generated this vector  $\phi$ . And this is our solution. So result is  $\phi$ .

(Refer Slide Time 17:57)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

Data: Matrix A, Vector r  
Result:  $\phi$

```

Decomposition
for k=1,n-1 do
  for i=k+1,n do
     $\gamma = a_{i,k} / a_{k,k}$ 
     $a_{i,k} = \gamma$ 
    for j=k+1,n do
       $a_{i,j} = a_{i,j} - \gamma \cdot a_{k,j}$ 
    end
  end
end
Forward Substitution
 $\psi_1 = r_1$ 
for i=2,n do
  sum = r_i
  for j=1,i-1 do
    sum = sum - a_{i,j} \cdot \psi_j
  end
   $\psi_i = \text{sum}$ 
end
Back Substitution
 $\phi_n = \psi_n / a_{n,n}$ 
for i=n-1,1 do
  sum =  $\psi_i$ 
  for j=i+1,n do
    sum = sum - a_{i,j} \cdot \phi_j
  end
   $\phi_i = \text{sum} / a_{i,i}$ 
end
return  $\phi$ 
  
```

Dr. Anirban Dhar NPTEL Computational Hydraulics

Now we need to implement this again with the scilab code. Now I will discuss the implementation of the code so that you will understand what is therewithin the process. Now again we will consider two example problems. One with a simple our banded matrix, one diagonal and 2 of diagonal terms. And second one is bit complicated with multiple number of variables.

(Refer Slide Time 18:50)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I.I.T. Kharagpur

### Example

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 0 & 1 & 3 & -1 & 0 \\ 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 12 \\ 11 \\ 28 \\ 9 \end{pmatrix}$$

**Solution:**

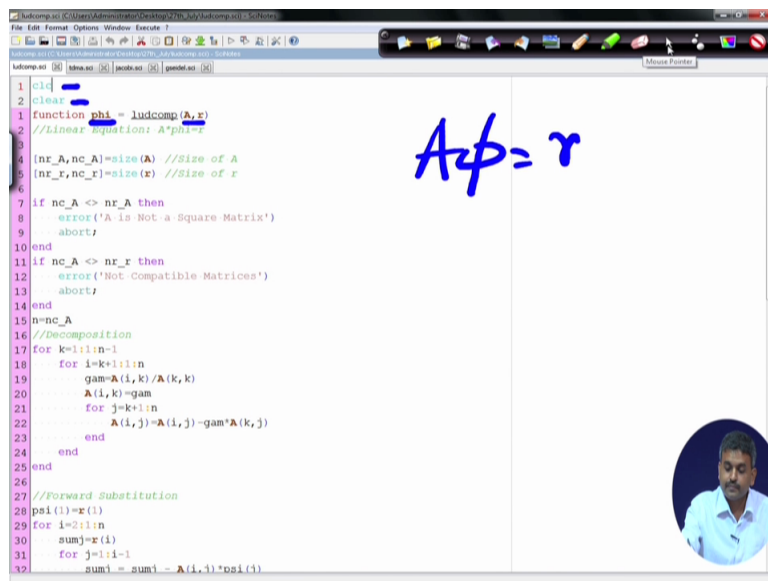
$$\begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix}$$

Dr. Anirban Dhar      NPTEL      Computational Hydraulics

Now if we consider our problem with scilab then we can say that this is our LU decomposition. So it starts with again the clcr clear command or clear screen command. Then clear, clear means clear of variables. Again ludcomp is our function name and function output is phi and input required is a, r. And this is linear equation because we are considering A phi equals to r problem in our case.




(Refer Slide Time 19:35)



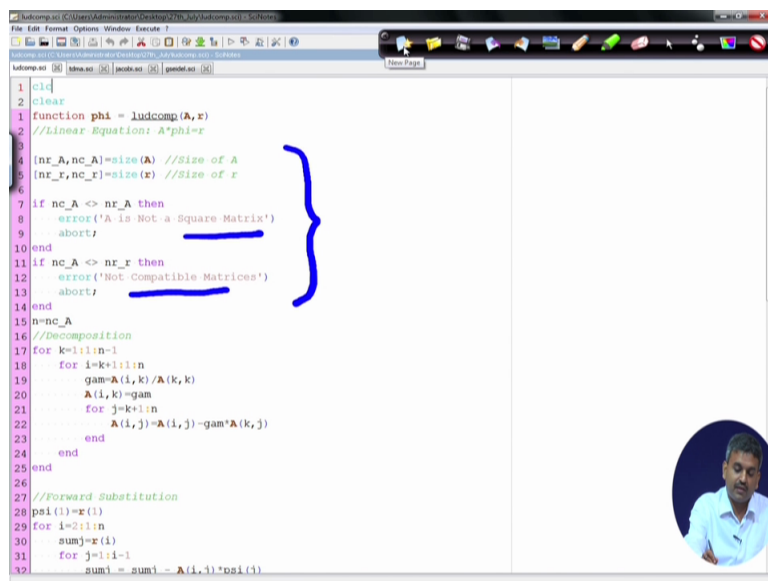
```
1 clear
2 clear
3 function phi = ludcomp(A,r)
4 //Linear Equation: A*phi=r
5
6 [nr_A,nc_A]=size(A) //Size of A
7 [nr_r,nc_r]=size(r) //Size of r
8
9 if nc_A <> nr_A then
10     error('A is Not a Square Matrix')
11     abort;
12 end
13 if nc_A <> nr_r then
14     error('Not Compatible Matrices')
15     abort;
16 end
17 n=nc_A
18 //Decomposition
19 for k=1:n-1
20     for i=k+1:n
21         gam=A(i,k)/A(k,k)
22         A(i,k)=gam
23         for j=k+1:n
24             A(i,j)=A(i,j)-gam*A(k,j)
25         end
26     end
27 end
28 //Forward Substitution
29 psi(1)=r(1)
30 for i=2:n
31     sumj=r(i)
32     for j=1:i-1
33         sumj = sumj - A(i,j)*psi(j)
34     end
35 end
36 psi(i)=sumj/A(i,i)
37 end
```

$A\phi = r$




So this part is common. We can check whether it is a square matrix or not. Or compatibility issues, because we need to operate on a and r. If there is problem in the input it can create further calculation related problem and your LU decomposition thing is not valid in that case.

(Refer Slide Time 20:11)



```
1 clear
2 clear
3 function phi = ludcomp(A,r)
4 //Linear Equation: A*phi=r
5
6 [nr_A,nc_A]=size(A) //Size of A
7 [nr_r,nc_r]=size(r) //Size of r
8
9 if nc_A <> nr_A then
10     error('A is Not a Square Matrix')
11     abort;
12 end
13 if nc_A <> nr_r then
14     error('Not Compatible Matrices')
15     abort;
16 end
17 n=nc_A
18 //Decomposition
19 for k=1:n-1
20     for i=k+1:n
21         gam=A(i,k)/A(k,k)
22         A(i,k)=gam
23         for j=k+1:n
24             A(i,j)=A(i,j)-gam*A(k,j)
25         end
26     end
27 end
28 //Forward Substitution
29 psi(1)=r(1)
30 for i=2:n
31     sumj=r(i)
32     for j=1:i-1
33         sumj = sumj - A(i,j)*psi(j)
34     end
35 end
36 psi(i)=sumj/A(i,i)
37 end
```



Now this is our decomposition step. Now in this decomposition step we are following the same algorithm that we have used in the Gauss elimination. We are starting from 1 to N minus 1. And i is running from k plus 1 to N. And this is also k plus 1 to N. This gamma is  $a_{ik}$  divided by  $a_{kk}$ .  $k$  is your reference row and  $a_{ik}$  we are storing this gamma. Now this part is different with respect to our Gauss elimination.

And for  $j = k + 1$  again we are changing our  $a_{ij}$  value. So obviously in this case these are updated  $a_{ij}$  values. So this is all about the decomposition step. Now in this step we have divided the matrix into lower and upper.

(Refer Slide Time 21:37)

```

15 function A
16 //Decomposition
17 for k=1:n-1
18     for i=k+1:n
19         gam=A(i,k)/A(k,k)
20         A(i,k)=gam
21         for j=k+1:n
22             A(i,j)=A(i,j)-gam*A(k,j)
23         end
24     end
25 end
26
27 //Forward Substitution
28 psi(1)=x(1)
29 for i=2:n
30     sumj=x(i)
31     for j=1:i-1
32         sumj = sumj - A(i,j)*psi(j)
33     end
34     psi(i)=sumj
35 end
36
37 //Backward Substitution
38 phi(n)=psi(n)/A(n,n)
39 for i=n-1:-1:1
40     sumj=psi(i)
41     for j=i+1:n
42         sumj=sumj-A(i,j)*phi(j)
43     end
44     phi(i)=sumj/A(i,i)
45 end
46 endfunction
49
50 //A=[1 2 -3 4 5

```

Handwritten annotations in blue ink:  $A = L U$  and a small matrix structure  $\begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}$ .

Now one thing maybe question is there that in lower triangular matrix we have considered that all values are in diagonal as one. But we have not written that thing here, because there is no scope for incorporation of this unit value in diagonal. So this diagonal thing we are directly utilizing in this forward substitution process.

(Refer Slide Time 22:12)

```

15 function A
16 //Decomposition
17 for k=1:n-1
18     for i=k+1:n
19         gam=A(i,k)/A(k,k)
20         A(i,k)=gam
21         for j=k+1:n
22             A(i,j)=A(i,j)-gam*A(k,j)
23         end
24     end
25 end
26
27 //Forward Substitution
28 psi(1)=x(1)
29 for i=2:n
30     sumj=x(i)
31     for j=1:i-1
32         sumj = sumj - A(i,j)*psi(j)
33     end
34     psi(i)=sumj
35 end
36
37 //Backward Substitution
38 phi(n)=psi(n)/A(n,n)
39 for i=n-1:-1:1
40     sumj=psi(i)
41     for j=i+1:n
42         sumj=sumj-A(i,j)*phi(j)
43     end
44     phi(i)=sumj/A(i,i)
45 end
46 endfunction
49
50 //A=[1 2 -3 4 5

```

Handwritten annotations in blue ink:  $A = L U$  and the matrix  $L = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}$ .

We are considering that the multiplication term with psi, so L multiplied psi equals to r is your forward substitution process. So directly we are writing it psi equals to r1.

(Refer Slide Time 22:29)

```

15 function A
16 //Decomposition
17 for k=1:1:n-1
18     for i=k+1:n
19         gam=A(i,k)/A(k,k)
20         A(i,k)=gam
21         for j=k+1:n
22             A(i,j)=A(i,j)-gam*A(k,j)
23         end
24     end
25 end
26
27 //Forward Substitution
28 psi(1)=r(1)
29 for i=2:1:n
30     sumj=r(i)
31     for j=1:i-1
32         sumj = sumj - A(i,j)*psi(j)
33     end
34     psi(i)=sumj
35 end
36
37 //Backward Substitution
38 phi(n)=psi(n)/A(n,n)
39 for i=n-1:-1:1
40     sumj=psi(i)
41     for j=i+1:n
42         sumj=sumj-A(i,j)*phi(j)
43     end
44     phi(i)=sumj/A(i,i)
45 end
46 endfunction
49
50 //A=[1 2 -3 4 5

```

Handwritten annotations in blue ink:

- Three horizontal lines with a minus sign:  $\equiv -$
- Two vertical parallel lines:  $\parallel$
- Equation:  $A = LU$
- Equation:  $L = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}$
- Equation:  $LY = r$

So psi 1 equals to r1. That means we are implicitly considering our coefficient as one. So we are not multiplying any quantity here or dividing any quantity here and we are not storing this value in our A matrix. We are implicitly implementing this thing.

(Refer Slide Time 22:58)

```

15 function A
16 //Decomposition
17 for k=1:1:n-1
18     for i=k+1:n
19         gam=A(i,k)/A(k,k)
20         A(i,k)=gam
21         for j=k+1:n
22             A(i,j)=A(i,j)-gam*A(k,j)
23         end
24     end
25 end
26
27 //Forward Substitution
28 psi(1)=r(1)
29 for i=2:1:n
30     sumj=r(i)
31     for j=1:i-1
32         sumj = sumj - A(i,j)*psi(j)
33     end
34     psi(i)=sumj
35 end
36
37 //Backward Substitution
38 phi(n)=psi(n)/A(n,n)
39 for i=n-1:-1:1
40     sumj=psi(i)
41     for j=i+1:n
42         sumj=sumj-A(i,j)*phi(j)
43     end
44     phi(i)=sumj/A(i,i)
45 end
46 endfunction
49
50 //A=[1 2 -3 4 5

```

Handwritten annotations in blue ink:

- Equation:  $\psi_1 = r_1$
- Equation:  $A = LU$
- Equation:  $L = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}$
- Equation:  $LY = r$

Now forward substitution, it is varying from 2 to N. So sum equals to ri. And sum j, aij, psi j, whatever value we have and psi i equals to sum j. As I have told that implicitly we are

considering that our diagonal term already coefficient one is there. So we are not multiplying or dividing any quantity here.

(Refer Slide Time 23:35)

```

15 function A
16 //Decomposition
17 for k=1:1:n-1
18     for i=k+1:n
19         gam=A(i,k)/A(k,k)
20         A(i,k)=gam
21         for j=k+1:n
22             A(i,j)=A(i,j)-gam*A(k,j)
23         end
24     end
25 end
26
27 //Forward Substitution
28 psi(1)=x(1)
29 for i=2:1:n
30     sumj=x(i)
31     for j=1:i-1
32         sumj = sumj - A(i,j)*psi(j)
33     end
34     psi(i)=sumj
35 end
36
37 //Backward Substitution
38 phi(n)=psi(n)/A(n,n)
39 for i=n-1:-1:1
40     sumj=psi(i)
41     for j=i+1:n
42         sumj=sumj-A(i,j)*phi(j)
43     end
44     phi(i)=sumj/A(i,i)
45 end
46 endfunction
49
50 //A=[1 2 -3 4 5

```

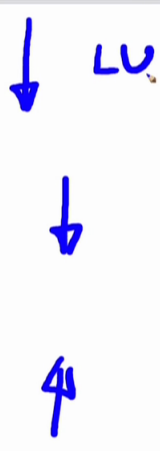
Then last step is our backward substitution. Again we are following the Gauss elimination steps. So  $\phi_n$  equals to  $\psi_n$  divided by  $a_{nn}$ . So this is the first calculation step. But we are starting from last row and we are moving upward direction. In the forward direction moment is there in the forward elimination. Decomposition step is also forward. In forward step we are constructing our L and U matrixes.

(Refer Slide Time 24:22)

```

15 function A
16 //Decomposition
17 for k=1:1:n-1
18     for i=k+1:n
19         gam=A(i,k)/A(k,k)
20         A(i,k)=gam
21         for j=k+1:n
22             A(i,j)=A(i,j)-gam*A(k,j)
23         end
24     end
25 end
26
27 //Forward Substitution
28 psi(1)=x(1)
29 for i=2:1:n
30     sumj=x(i)
31     for j=1:i-1
32         sumj = sumj - A(i,j)*psi(j)
33     end
34     psi(i)=sumj
35 end
36
37 //Backward Substitution
38 phi(n)=psi(n)/A(n,n)
39 for i=n-1:-1:1
40     sumj=psi(i)
41     for j=i+1:n
42         sumj=sumj-A(i,j)*phi(j)
43     end
44     phi(i)=sumj/A(i,i)
45 end
46 endfunction
49
50 //A=[1 2 -3 4 5

```



So in backward step we have already got the value of phi n. And from this phi n we are calculating the values for n minus 1 to 1 with the increment of minus 1. So sum j is equal to psi i, because you have this thing. In the next step or backward substitution step this is upper triangular matrix.

(Refer Slide Time 24:59)

```

15 function A
16 //Decomposition
17 for k=1:n-1
18     for i=k+1:n
19         gam=A(i,k)/A(k,k)
20         A(i,k)=gam
21         for j=k+1:n
22             A(i,j)=A(i,j)-gam*A(k,j)
23         end
24     end
25 end
26
27 //Forward Substitution
28 psi(1)=x(1)
29 for i=2:n
30     sumj=x(i)
31     for j=i-1:1
32         sumj = sumj - A(i,j)*psi(j)
33     end
34     psi(i)=sumj
35 end
36
37 //Backward Substitution
38 phi(n)=psi(n)/A(n,n)
39 for i=n-1:-1:1
40     sumj=psi(i)
41     for j=i+1:n
42         sumj=sumj-A(i,j)*phi(j)
43     end
44     phi(i)=sumj/A(i,i)
45 end
46 endfunction
47
48 //A=[1 2 -3 4 5

```

Uφ = ψ

Now in this case psi values are already known values and these are acting as right hand side vector. So  $\sum_j A_{ij} \phi_j$ , we are subtracting this thing and calculating sum j. But in this case, in upper triangular matrix we have diagonals available. And those diagonal terms are either 1 or any value. So we cannot generalize that. So we are utilizing those diagonal terms for calculation of phi i. So phi i equals to i divided by sum j equals to sum j divided by a ii. So we're getting phi i values.

(Refer Slide Time 25:57)

```

15 function A
16 //Decomposition
17 for k=1:n-1
18     for i=k+1:n
19         gam=A(i,k)/A(k,k)
20         A(i,k)=gam
21         for j=k+1:n
22             A(i,j)=A(i,j)-gam*A(k,j)
23         end
24     end
25 end
26
27 //Forward Substitution
28 psi(1)=r(1)
29 for i=2:n
30     sumj=r(i)
31     for j=1:i-1
32         sumj = sumj - A(i,j)*psi(j)
33     end
34     psi(i)=sumj
35 end
36
37 //Backward Substitution
38 phi(n)=psi(n)/A(n,n)
39 for i=n-1:-1:1
40     sumj=psi(i)
41     for j=i+1:n
42         sumj=sumj-A(i,j)*phi(j)
43     end
44     phi(i)=sumj/A(i,i)
45 end
46 endfunction
49
50 //A=[1 2 -3 4 5

```

$U =$

$U\phi = \psi$

Now if we utilize our concept wise thing, so let us consider our simplified case where we have only phi1, phi2, phi3, phi4 and phi5. So these are odd numbers 1, 3, 5, 7 and last one is 9.

(Refer Slide Time 26:30)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm

I. I. Kharagpur

Example

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 0 & 1 & 3 & -1 & 0 \\ 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix} = \begin{pmatrix} 6 \\ 12 \\ 11 \\ 28 \\ 9 \end{pmatrix}$$

Solution:

$$\begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \\ 9 \end{pmatrix}$$

Dr. Anirban Dhar      NPTEL      Computational Hydraulics      13

And storing this A value here and this is r. So with this we should get 1, 3, 5, 7, 9 as our solution. Now we can select this with the right click, evaluate command, we can check whether we are getting.

(Refer Slide Time 26:50)

```
40 sum=phi(i)
41 for j=1+i:n
42 sum=sum-A(i,j)*phi(j)
43 end
44 phi(i)=sum/A(i,i)
45 end
46 endfunction
49
50 //A=[1 2 -3 4 5
51 //0 3 -5 -2 3
52 //5 -4 3
53 //1 4 -7
54 // -15 13
55
56 //r=[7]
57 //b=[4]
58 //c=[1]
59 //l=[1]
60 //l0=[1]
61
62 //=[1 0 0
63 // 2 1 0
64 // -1 3 -1
65 // 0 0 1
66 // 0 0 0 1]
67
68 //i=
69 //j=
70 //k=
71 //l=
72 //l0=
73
74 phi=luDecomp(A,b)
75
```

So we can see that exactly we are getting the solution from our LU decomposition step. This is 1, 3, 5, 7, 9.

(Refer Slide Time 27:06)

```
--> //0 3 -5 -2 3
--> //5 -4 3 -2 1
--> //1 4 -7 -10 13
--> // -15 13 11 -9 21;
-->
--> //r=[7]
--> //b=[4]
--> //c=[1]
--> //l=[1]
-->
--> //=[1 0 0 0
--> // 2 1 0 0
--> // 0 1 3 -1 0
--> // 0 0 1 2 1
--> // 0 0 0 0 1];
-->
--> //i=1
--> //j=12
--> //k=11
--> //l=28
--> //l0=91;
-->
--> phi=luDecomp(A,b)
phi =
     1
     3
     5
     7
     9
```

Now in this case we are calling the function phi LU decomposition a, r with this. But if we have let us say we're changing it to a1 and we are utilizing our that general matrix structure problem, 5 by 5 although it is 5 by 5 for this LU decomposition, we can check that we are getting the solution 1, 2, 3, 4, 5. Again this is our solution as discussed in our Gauss elimination slides.





(Refer Slide Time 30:00)

Decomposition  
Forward Substitution  
Backward Substitution  
Algorithm


I.I.T. Kharagpur

### Example

$A$   $N \times N$

$$\begin{pmatrix} 1 & 2 & -3 & 4 & 5 \\ 0 & 3 & -5 & -7 & 9 \\ 5 & -4 & 3 & -2 & 1 \\ 1 & 4 & -7 & -10 & 13 \\ -15 & 13 & 11 & -9 & 2 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix} = \begin{pmatrix} 37 \\ 8 \\ 3 \\ 13 \\ 18 \end{pmatrix}$$

**Solution:**

$$\begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}$$


Dr. Anirban Dhar NPTEL Computational Hydraulics 13

But if you have some diagonal structure or banded structure of the matrix, you don't need to store the full matrix. Without storing the full matrix, we can solve the problem. In the next lecture we will be discussing that special kind of algorithms. Thank you.