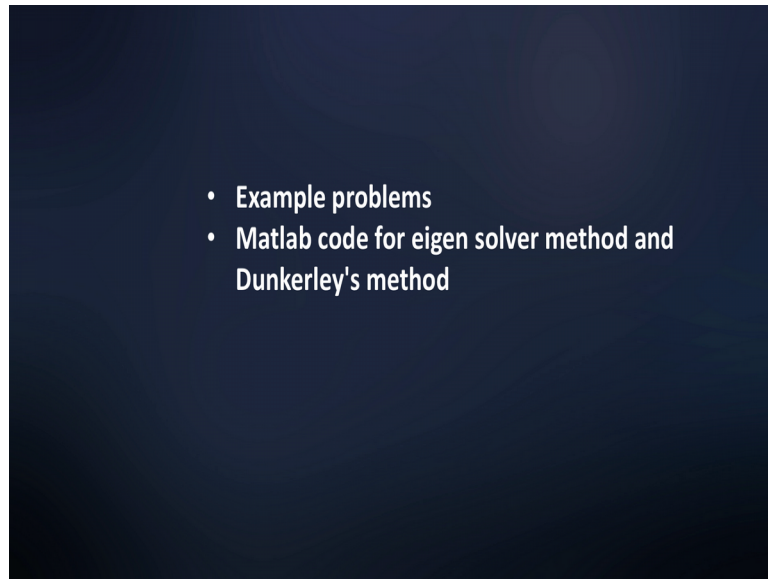


**Computer Methods of Analysis of Offshore Structures**  
**Prof. Srinivasan Chandrasekaran**  
**Department of Ocean Engineering**  
**Indian Institute of Technology, Madras**

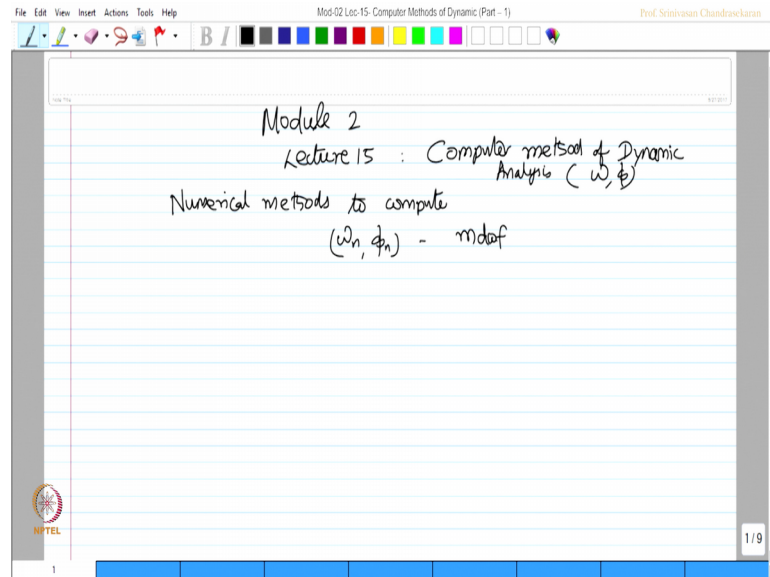
**Module - 02**  
**Lecture - 15**  
**Computer methods of dynamic analysis**

(Refer Slide Time: 00:16)



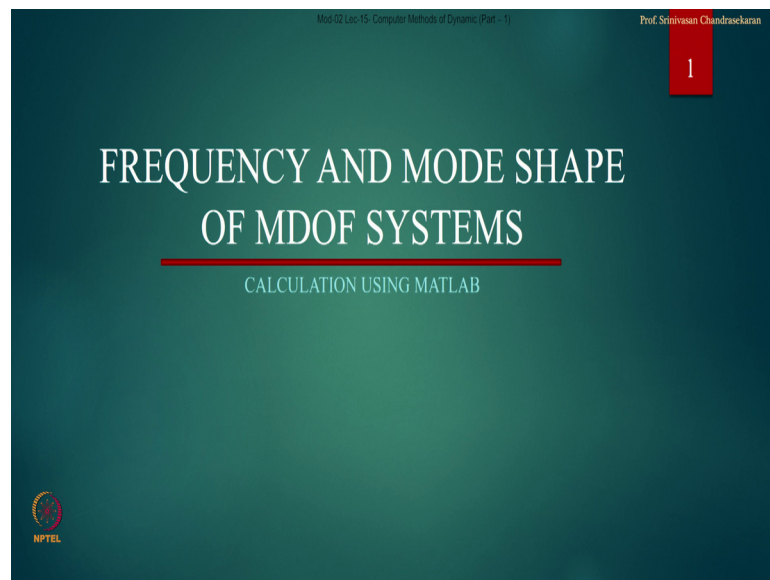
Friends welcome to the 15th lecture in module 2. In the previous lectures we discussed about different numerical methods to compute the natural frequency and mode shape of a multi degree freedom system. We have also shown you the comparison of the results and we realize that the values are tallying and they are in good agreement.

(Refer Slide Time: 00:38)



In this lecture we will explain you how a computer code is being written to estimate the natural frequency and mode shape we will again compare the rules of a specific problem what we did in the last lecture behind, compare the results with computer codes then we will see that they are again in better agreements.

(Refer Slide Time: 01:21)



So, we will **now** discuss about the frequency and mode shape of multi degree freedom system model. We will take an example problem.

(Refer Slide Time: 01:33)

Mod-02 Lec-13 Computer Methods of Dynamic (Part - 1) Prof. Srinivasan Chandrasekaran

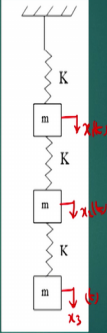
### EXAMPLE PROBLEM

▶ Obtain all mode shapes and corresponding frequency of the system shown below:

▶ Mass Matrix,  $M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

▶ Stiffness Matrix,  $K = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$

$m_1 = m_2 = m_3 = m$   
 $k_1 = k_2 = k_3 = k$



NPTEL

The problem demands to obtain all mode shapes and a corresponding frequency of a system shown the system has got 3 stiffness  $K$   $K$   $K$  and  $m$  which are equal. And now marking the degrees of freedom this is my  $x_1$  this is my  $x_2$ , this is my  $x_3$ , I can say even  $x_1$  of  $t$   $x_2$  of  $t$  and  $x_3$  of  $t$ . So, one can very clearly see here  $m_1$  equals  $m_2$  equals  $m_3$  is simply  $m$  maybe some numeric value similarly  $k_1$ ,  $k_2$  and  $k_3$  are taken as  $k$  some numeric value. This problem we solved in the last lecture behind compared the results by different methods we found them that they are in good agreement.

So, easily when the degrees of freedom are marked at the point where mass is lumped the mass matrix will be the diagonally dominant matrix whereas, off diagonal elements will all be 0 which we already know and there is no confusion in writing the mass matrix readily. To write the stiffness matrix we need to formulate the equation of motion which we already said by Newton's method let us try to do that quickly for this specific problem, let us compare this and see how we are able to get the stiffness matrix.

(Refer Slide Time: 03:15)

The slide shows a three-degree-of-freedom mass-spring system and its equations of motion. The system consists of three masses, each of mass  $m$ , connected in series by springs with stiffness  $k$ . The top mass is fixed to a ceiling. The displacement of the masses from their equilibrium positions are denoted by  $x_1$ ,  $x_2$ , and  $x_3$  respectively.

The equations of motion are derived as follows:

For the first mass (top):

$$F = ma$$

$$M\ddot{x}_1 = -kx_1 - k(x_1 - x_2)$$

$$= -kx_1 - kx_1 + kx_2$$

$$= -2kx_1 + kx_2$$

$$M\ddot{x}_1 + 2kx_1 - kx_2 = 0 \quad \text{--- (1)}$$

For the second mass (middle):

$$M\ddot{x}_2 = -k(x_2 - x_1) - k(x_2 - x_3)$$

$$= -kx_2 + kx_1 - kx_2 + kx_3$$

$$= kx_1 - 2kx_2 + kx_3$$

$$M\ddot{x}_2 - kx_1 + 2kx_2 - kx_3 = 0 \quad \text{--- (2)}$$

For the third mass (bottom):

$$M\ddot{x}_3 = -k(x_3 - x_2)$$

$$= -kx_3 + kx_2$$

$$M\ddot{x}_3 - kx_2 + kx_3 = 0 \quad \text{--- (3)}$$

The equations are written in matrix form as:

$$[M] \ddot{\mathbf{x}} + [K] \mathbf{x} = \mathbf{0}$$

where  $[M] = m \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ ,  $[K] = k \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$ , and  $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ .

So, the system what we have is  $m$   $m$  and  $m$ ,  $k$   $k$  and  $k$ , this is  $x$   $1$   $x$   $2$  and  $x$   $3$ . So, we got a free body diagram let us say unit displacement, this is going to be  $k$  times of  $x$   $1$  and when this spring will now oppose  $k$  times of  $x$   $1$  minus  $x$   $2$  and we are giving unit displacement in the degree of freedom  $x$   $1$ . Similarly for the second degree we give unit displacement along  $x$   $2$  and this spring will try to oppose the movement which will be  $k$  of  $x$   $2$  minus  $x$   $1$  and this spring will now again oppose the moment of  $k$  of  $x$   $2$  minus  $x$   $3$ . And we have the third degree this is also  $m$   $m$  and  $m$ . So, this is again  $x$   $3$ , so this spring will try to oppose the moment which is  $x$   $3$  minus  $x$   $2$ .

Let us write the force balance equation for the first case. So, force is equal to  $ma$ . So,  $m$   $x$   $1$  double dot,  $x$   $1$  double dot should be equal to minus of  $kx$   $1$  minus  $k$   $x$   $1$  minus  $x$   $2$  which is simplify as minus  $kx$   $1$  minus  $kx$   $1$  plus  $kx$   $2$  which is minus  $2kx$   $1$  plus  $kx$   $2$ . So,  $m$   $x$   $1$  double dot plus  $2kx$   $1$  minus  $kx$   $2$  is  $0$  that is my first equation of motion.

Now, for the second degree similarly  $m$   $x$   $2$  double dot will be equal to minus  $k$  of  $x$   $2$  minus  $x$   $1$  minus  $k$  of  $x$   $2$  minus  $x$   $3$  which will be minus  $kx$   $2$  plus  $kx$   $1$  minus  $kx$   $2$  plus  $kx$   $3$  which is say  $kx$   $1$  minus  $2kx$   $2$  plus  $kx$   $3$  that is  $m$   $x$   $2$  double dot. So, I can say  $m$   $x$   $2$  double dot minus  $kx$   $1$  plus  $2kx$   $2$  minus  $kx$   $3$  is  $0$ , that is my second equation of motion.

Let us say the third equation which I write here  $m$   $x$   $3$  double dot is minus  $k$   $x$   $3$  minus  $x$   $2$ . So,  $m$   $x$   $3$  double dot is minus  $kx$   $3$  plus  $kx$   $2$ . So,  $m$   $x$   $3$  double dot minus  $kx$   $2$  plus



$kx_3 = 0$  that is my third equation. Now, writing all these equations in a simultaneous form let us say  $m \ddot{x}_1 + 2kx_1 - kx_2 = 0$ ,  $m \ddot{x}_2 - kx_1 + 2kx_2 - kx_3 = 0$ ,  $m \ddot{x}_3 - kx_2 + kx_3 = 0$ . I can now convert these equations into a matrix form, I can now say this can be expressed as  $m$  times of  $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$  of  $\ddot{x}_1 \ddot{x}_2 \ddot{x}_3$  plus  $k$  times of  $\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$  of  $x_1 x_2 x_3 = 0$ .

So, this is my mass matrix and this is my stiffness matrix. So, if you look at the file what we just now showed to you the mass matrix and the stiffness matrix are exactly in the same style, so  $k$  times of or  $m$  times of  $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$  and  $k$  times of  $\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$ . So, getting a  $k$  matrix as an input to the program is not difficult we can write equations of motion, getting mass matrix and stiffness matrix as an input for a given program is not a problem.

(Refer Slide Time: 09:40)



Having said this let us try to write the program by 4 methods Eigen Solver, Dunkerley, Influence coefficients, Stodola and Rayleigh Ritz.

(Refer Slide Time: 09:52)

Mod-02 Lec-13 Computer Methods of Dynamic (Part - 1) Prof. Srinivasan Chandrasekaran

## EIGEN SOLVER METHOD

5

MATLAB code:

```
% Program for finding frequency and mode shape using EIGEN SOLVER METHOD
% Includes check for orthogonality and normalization
clear;
clear;
% Enter dof
dof=3;
%Enter Mass matrix
M=[1 0 0;0 1 0;0 0 1]; % mass in kg
%Enter Stiffness Matrix
K=[2 -1 0;-1 2 -1;0 -1 1]; % Stiffness in N/m
fprintf('Mass Matrix\n')
disp(M);
fprintf('Stiffness Matrix\n')
disp(K);
```

**INPUT**

- ❖ The Inputs given for calculating the frequency and mode shape of MDOF system are
  - ❖ Degrees of Freedom
  - ❖ Mass Matrix
  - ❖ Stiffness Matrix
- ❖ Check the size of the matrices while giving the input (Here, it is a 3x3 matrix).
- ❖ The Mass values are given in 'kg' and stiffness values in 'N/m'.

Let us take the Eigen solver method the coding is available on the screen. Now the hatch box shows you the input file actually inputs are given in multiple formats the first input refers to the degree of freedom, second line of the input refers to the mass matrix value in kg you are going to give the value with a space bar and separate each row by a semicolon. The third line of the program input is for stiffness the units are in Newton per meter similarly you have to enter row wise the stiffness matrix what we just now saw is 2 minus 1 0 is a first row. So, you can see here 2 minus 1 0 with a space bar in between and a semicolon separating the lines open with the bracket and closed with a bracket.

So, you can enter the input file as degree of freedom of your choice mass matrix of your choice and k matrix of your choice. Once you do this the program is now going to evaluate the eigen value and eigen vector for this problem. So, it calculates the stiffness matrix it prints the stiffness matrix. Then it computes the eigen values and eigen vectors, it also estimates the modal matrix.

(Refer Slide Time: 11:12)

Mod-02 Lec-15 Computer Methods of Dynamic (Part - 1) Prof. Srinivasan Chandrasekaran

### EIGEN SOLVER METHOD - contd

6

**MATLAB code**

```
%% eigen values and eigen vectors ✓  
[mode,w_square]=eig(K,M);  
freq=sqrt(w_square);  
for i=1:dof  
    wn3(i)=freq(i,i);  
    moden(:,i)=mode(:,i)/mode(1,i);  
    fprintf('Frequency: wn = %6.3f rad/s \n',wn3(i));  
end  
fprintf('Modal Matrix \n x = \n');  
disp(moden);
```

❖ This section of code will calculate the frequency and mode shape of the MDOF system.  
❖ This methods gives all frequencies and mode shapes.  
❖ The mode shapes are combined and represented as the modal matrix here.

$\Phi = \begin{bmatrix} | & | & | \\ \vdots & \vdots & \vdots \\ | & | & | \end{bmatrix}$

This method gives all frequencies and mode shapes the mode shapes are combined in a modal matrix is written here. So, in a modal matrix of 3 by 3 because the 3 degree freedom system model, each column will refer to the each vector so this is phi 1, this column will be phi 2 and this column will be phi 3 together this matrix is called phi matrix or modal matrix.

(Refer Slide Time: 11:56)

Mod-02 Lec-15 Computer Methods of Dynamic (Part - 1) Prof. Srinivasan Chandrasekaran

### EIGEN SOLVER METHOD - contd

7

**MATLAB code**

```
%% Check for orthogonality  
for i=1:dof-1  
    c1(i)=fix(moden(:,i)*moden(:,i+1))*1000/1000;  
end  
if c1=0  
    fprintf('The modes are orthogonal to each other. \n');  
else  
    fprintf('The modes are not orthogonal to each other. \n');  
    break  
end
```

❖ This section checks the orthogonality of the mode shapes.  
❖ The higher modes can be easily interpreted if they are orthogonal to each other.

Then orthogonality condition is also checked for this given problem and if the modes are not orthogonal the modes are made orthogonal and one can find out that.

(Refer Slide Time: 12:08)

Mod-02 Lec-15 Computer Methods of Dynamic (Part - 1) Prof. Srinivasan Chandrasekaran


### EIGEN SOLVER METHOD - contd

8

**MATLAB code**

```
%% Normalization and check
for i=1:dof
    mnew=moden(:,i)*M*moden(:,i);
    moden(:,i)=moden(:,i)/(sqrt(1/mnew));
end
for i=1:dof-1
    d1(i)=round((moden(:,i)*M*moden(:,i))*1000/1000);
    d2(i)=fix((moden(:,i)*M*moden(:,i+1))*1000/1000);
end
s1=sum(d1)/1;
s2=sum(d2);
if s1==dof && s2==0
    fprintf('The Normalized Modal Matrix 'nx - n');
    disp(modenor);
else
    break;
end
```

- ❖ The process of orthogonalising the modes is called Normalization.
- ❖ This section of code will calculate the normalized mode shapes.
- ❖ The output is given as Modal Matrix.



So, now normalization is being done and we get ultimately the normalized modal matrix with respect to mass. You can normalize the matrix modal matrix either with respect to mass or with respect to stiffness. So, in this case it is normalized with respect to mass.

(Refer Slide Time: 12:30)

Mod-02 Lec-15 Computer Methods of Dynamic (Part - 1) Prof. Srinivasan Chandrasekaran


### EIGEN SOLVER METHOD - contd

9

**MATLAB code:**

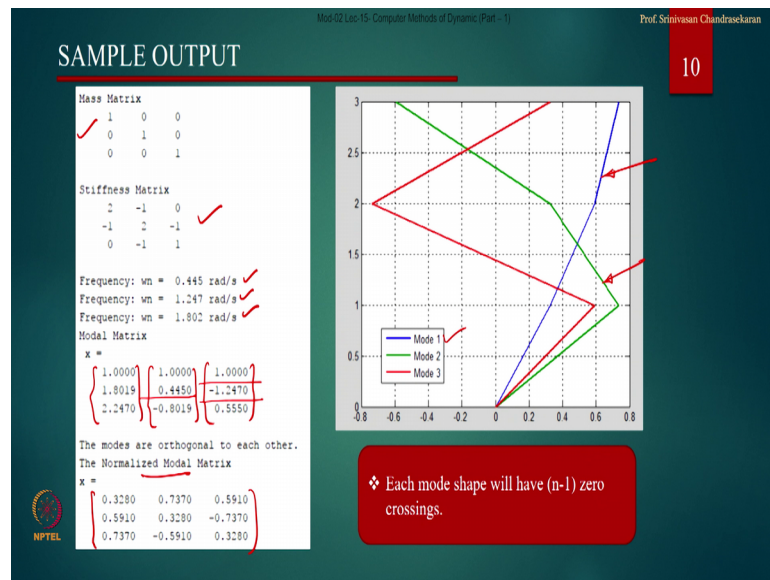
```
%% Mode shape plot
modenew = [modenor(dof,:);modenor(dof-1,:);modenor(dof-2,:)];
ys= dof-1:0;
mp = [modenew;zeros(1,dof)];
plot(mp,ys,'linewidth',2);
%% check
a=transpose(modenor)*M*modenor;
b=transpose(modenor)*K*modenor;
```

- ❖ The last section of code will give the mode shape plot.
- ❖ Check for the mass normalised modal matrix.



So, the last section of the code actually gives you the mode shape plot also you can see here, the mode shape plot is being given in this section, it even plots the mode shape.

(Refer Slide Time: 12:42)

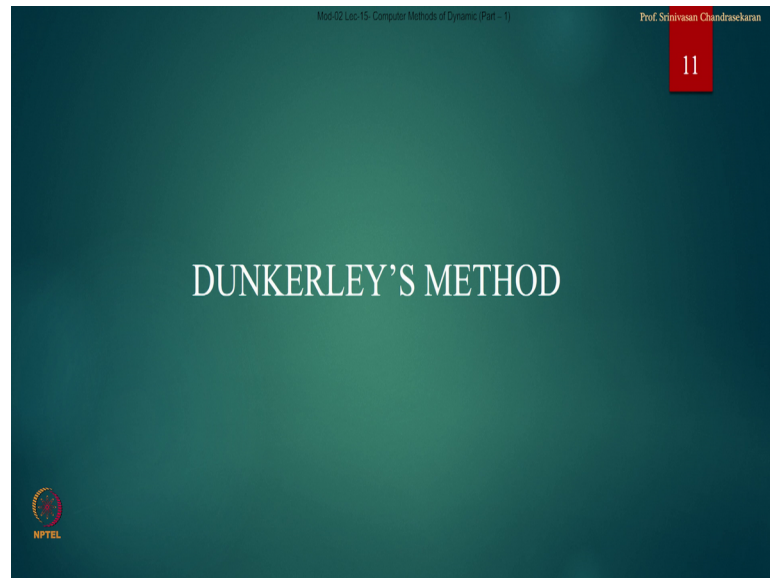


And there is a sample input mass matrix is been given stiffness matrix has been printed and it calculates all the 3 frequencies it calculates the corresponding omega 1 this is phi 1, omega 2 this is phi 2 and omega 3 this is phi 3.

We can see here phi 1 does not have any zero crossing phi 2 has 1 zero crossing and phi 3 has 2 zero crossings. So, these are first mode shape, second mode shape and third mode shape. These modes are naturally not orthogonal they are now normalized and you get the normalized mode shape like this. They are also plotted the blue one, the blue one shows mode 1 the green one shows mode 2 and the red one of course, shows the mode 3. So, this has been obtained with the sample program which has been given to you on the screen I will also give you the complete coding at the end of this presentation.

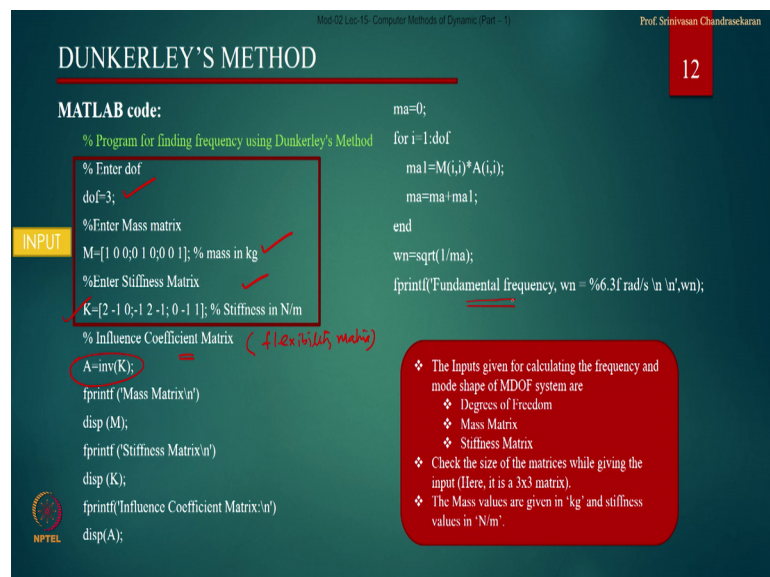


(Refer Slide Time: 13:47)



Let us move further let us take the Dunkerleys method. Dunkerley method is a shortcut method which again requests you to give the input as degree of freedom, mass and stiffness.

(Refer Slide Time: 13:58)



Then it calculates the influence coefficient matrix by inverting by inverting the stiffness matrix, inverse the standard subroutine in MATLAB. So, once k matrix is designated can invert the matrix to get influence coefficient. So, friends please remember influence coefficient matrix is nothing, but the flexibility matrix.

So, inverse, influence coefficient matrix is obtained once you have this coding with you then it estimates only the fundamental frequency with of course, a high degree of approximation let us see the sample input.

(Refer Slide Time: 14:48)

Mod 02 Lec 15: Computer Methods of Dynamic (Part - 1) Prof. Srinivasan Chandrasekaran

### SAMPLE OUTPUT

Mass Matrix

1	0	0
0	1	0
0	0	1

Stiffness Matrix

2	-1	0
-1	2	-1
0	-1	1

Influence Coefficient Matrix:

1.0000	1.0000	1.0000
1.0000	2.0000	2.0000
1.0000	2.0000	3.0000

Fundamental frequency (Dunkerley),  $\omega_n = 0.408$  rad/s

$\frac{1}{\omega_n^2} = \sum m_i \alpha_{ii}$

❖ Dunkerley's Method gives only the fundamental frequency, with serious approximation.

NPTEL

So, the sample input for the mass and stiffness is same this becomes my flexibility matrix now which otherwise a call as influence coefficient matrix and based on the Dunkerleys algorithm which says  $1/\omega^2$  is actually sum of  $m_i \alpha_{ii}$  it calculates  $\omega_1$  it gives you only the fundamental frequency with a serious level of approximation. The value what you obtained is 0.408, in the previous case the value what we obtained is 0.445 for the first mode shape and frequency. So, there is an approximation.