RMIT University

OENG-1116: MODELLING & SIMUALTION OF ENGINEERING SYSTEMS

Week 3b

Introduction into FEM. Plotting Surfaces with MATLAB.

Prof Pavel M. Trivailo

pavel.trivailo@rmit.edu.au

FEEDBACK:

QUESTIONS
HOME WORK
OBSERVATIONS
GENERAL DISCUSSIONS

PROJECT:

Modelling & Simulation of Bending Vibrations of Beams, using FEM (Structural Dynamics Application)

WHERE IS THE PROJECT (Part-1) ASSIGNMENT ?

ONE OF THE PREVOUS YEAR EXAMPLE (MOCK UP for 2020)

OENG1116_2018_Project_Pt1

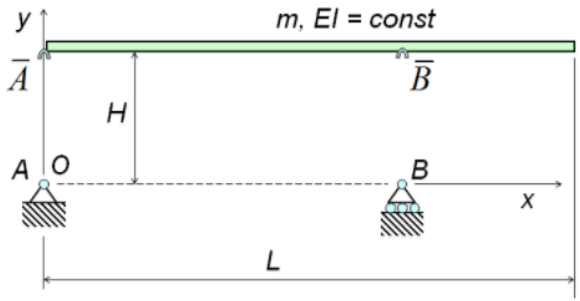
Your email address (**pavel.trivailo@rmit.edu.au**) will be recorded when you submit this form. Not you? Switch account

* Required

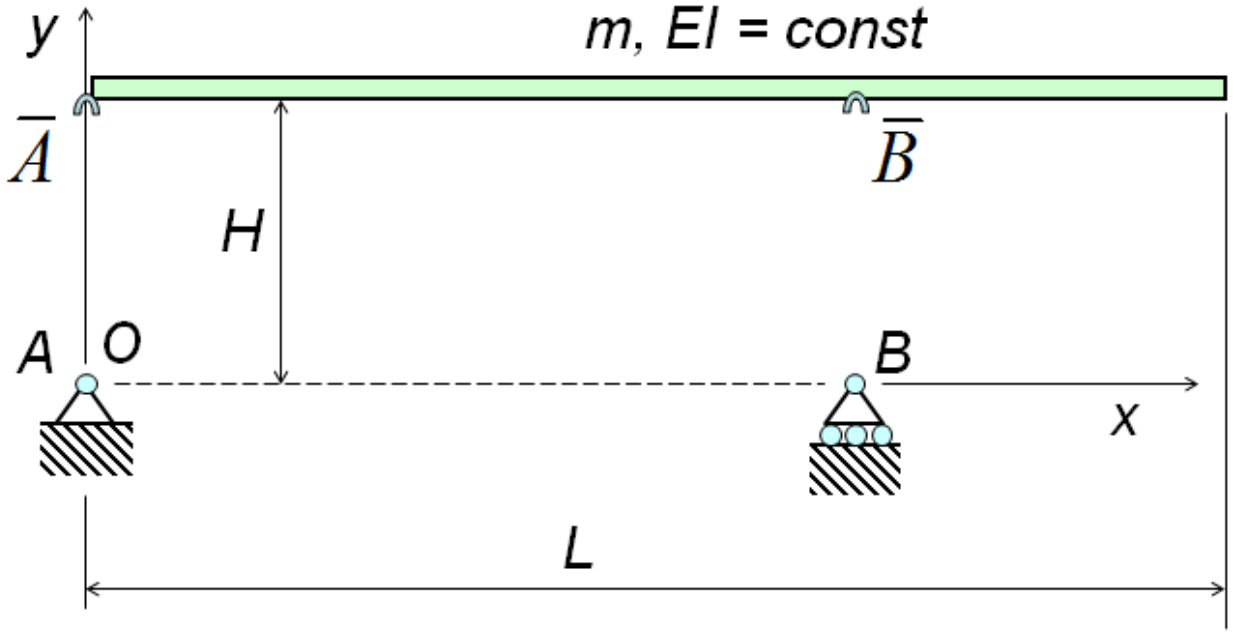
TASK: Modelling & Simulation of Bending Vibrations of Beams, using FEM (Structural Dynamics Application)

Figure shows a simple uniform beam of the length L, constant mass per unit length m, and constant bending stiffness EI. The beam is initially held at rest and then is dropped from the horizontal position from the height y=H. When it reaches the level y=0, it collides with the obstacles A and B at x=0 and x=a, which both automatically lock as pins on the beam's matching points A and B , arresting the motion of the beam. Assume the instant of the collision as initial time t=0 and consider the idealized case where there are no energy losses due to friction, impact, etc., and where no rebound occurs at the supports.

A uniform beam (shown at the approaching stage (t<0).



THE PROJECT TASK-1



A uniform beam (shown at the approaching stage (t<0).

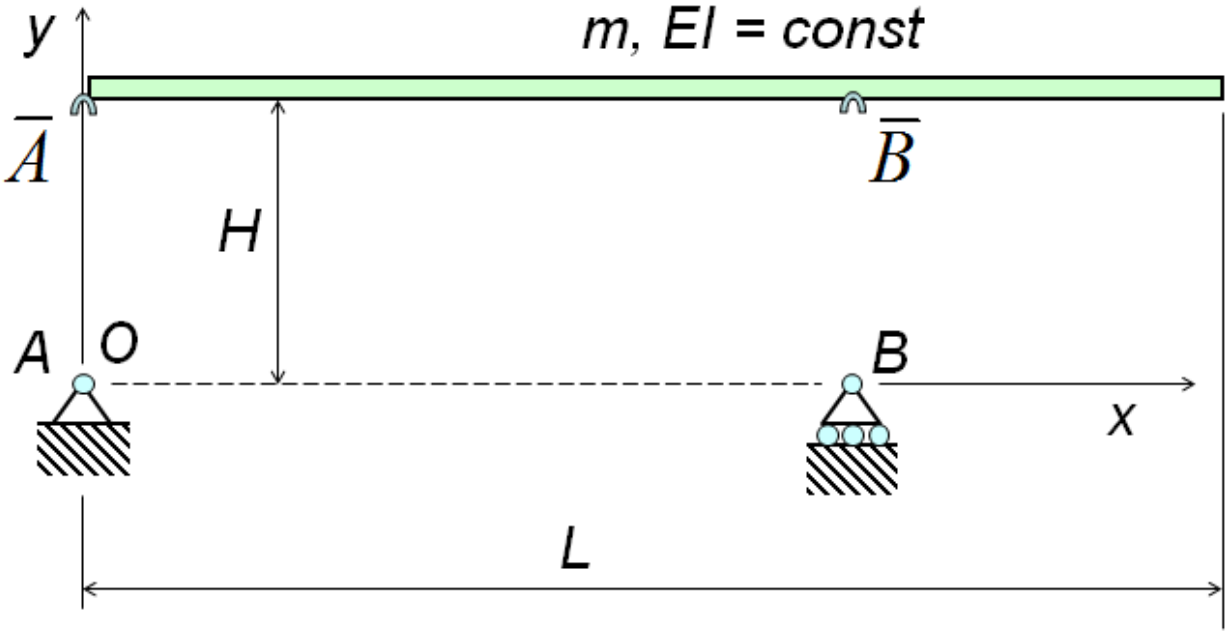
Figure shows a simple uniform beam of the length *L*, constant mass per unit length *m*, and constant bending stiffness *EI*. The beam is initially held at rest and then is dropped from the horizontal position from the height *y=H.* When it reaches the level *y=0*, it collides with the obstacles *A* and *B* at *x=0* and *x=a*, which both automatically lock as pins on the beam's matching points *A* and *B*, arresting the motion of the beam. Assume the instant of the collision as initial time *t=0* and consider the idealized case where there are no energy losses due to friction, impact, etc., and where no rebound occurs at the supports.

Assume the following parameters for the simulation case of the beam:

- Width of the beam: *b*=60mm; Height of the beam: *h*=10mm; Length of the beam : *L*=2m;
- Young Modulus of the Beam: *E*=200 GPa;
- Density of the material: ρ=7500 kg/m^3 (stainless steel);
- Height *H*=2m;
- Location of the right automatic hinge: *a*=0.6 *L* = 1.2 m

TASK-1:
Program in MATLAB the Finite Element Model of the beam (modelled with 10 equal length FEs) and determine the first natural frequency of the beam in rad/s.

THE PROJECT TASK-2



A uniform beam (shown at the approaching stage (t<0).

TASK-2:
Simulate response of the beam, modelled with 10 Finite Elements and present your result as a 3D SURFACE PLOT in the following axes:

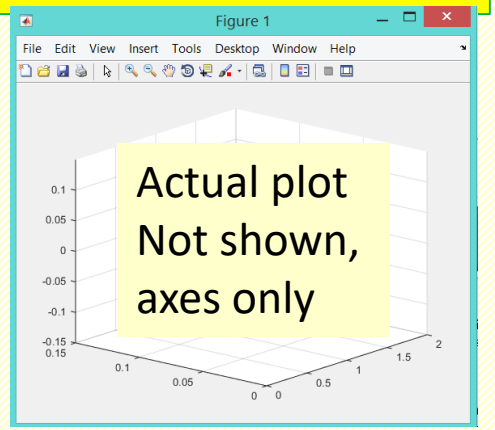
x(length in m, from 0 to 2 m) –
t(time in s, from 0 to 0.15 s) –
y (lateral displacement in m, from -0.15 to 0.15 m).

Please, make sure that your plot satisfies the following two conditions:

(1) axis([0 2 0 0.15 -0.15 0.15]) and
(2) view([-50 20]).

Make sure that your last plotting commands for Task-2 are:

```
axis([0 2  0 0.15  -0.15 0.15]);
view([-50 20]);
grid on;
```



SUBMIT PROJECT (Part-1):

SUBMIT VIA GOOGLE FORM (not Canvas)
BEFORE THE DUE DATE

SUBMIT YOUR MATLAB SCRIPT FOR Task-1 AS ONE (SINGLE) FILE INTO THE WINDOW BELOW, using "cut-and-paste" method, cutting the script from your *.M file. The first line should have double percentage, space, and student numbers of all Members of the Team, separated by commas, each number starting with "s". *

Your answer

Task-2: PLOT RESPONSE OF THE BEAM

Simulate response of the beam, modelled with 10 Finite Elements and present your result as a 3D SURFACE PLOT in the following axes: x(length in m, from 0 to 2 m) - t(time in s, from 0 to 0.15 s) - y (lateral displacement in m, from -0.15 to 0.15 m). Please, make sure that your plot satisfies the following two conditions: (1) axis([0 2 0 0.15 -0.15 0.15]) and (2) view([-50 20])

*

Please, upload your plot as a single image file.

ADD FILE

SUBMIT YOUR MATLAB SCRIPT FOR Task-2 AS ONE (SINGLE) FILE INTO THE WINDOW BELOW, using "cut-and-paste" method, cutting the script from your *.M file. The first line should have double percentage, space, and student numbers of all Members of the Team, separated by commas, each number starting with "s". *

Your answer

A copy of your responses will be emailed to pavel.trivailo@rmit.edu.au.

SUBMIT

Never submit passwords through Google Forms.

PROJECT-2020

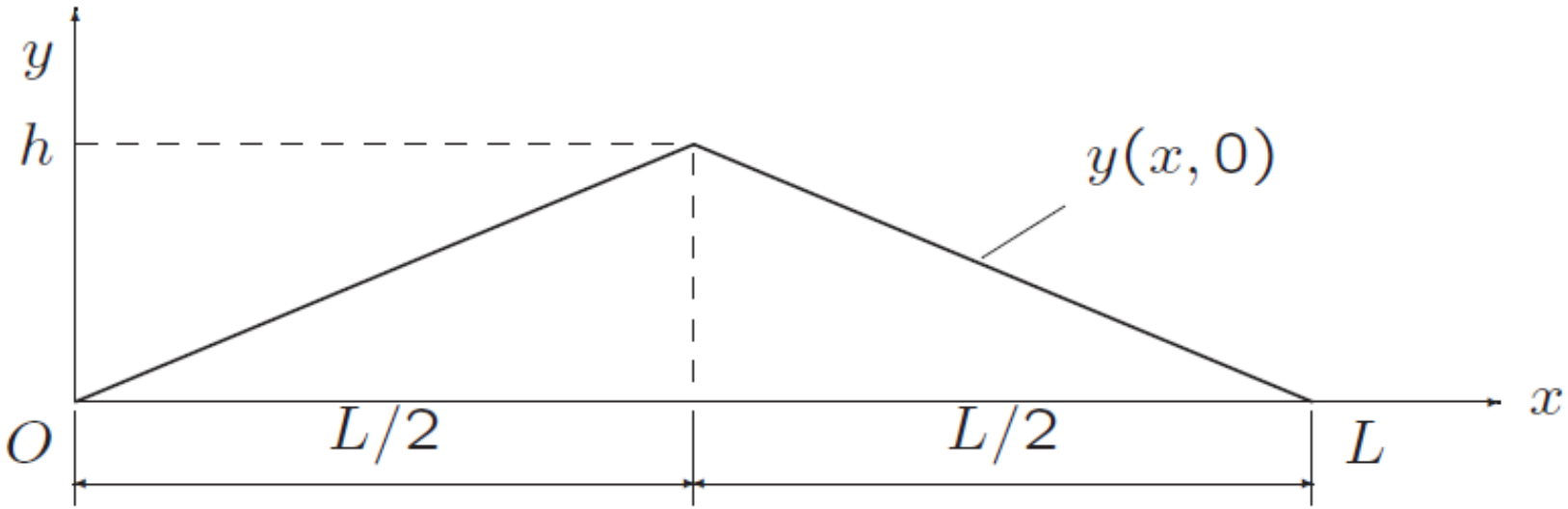
with due on 19-May-2020, Wk.11:

Individual Variants will be sent to you via your RMIT student email during next week.

Debugger in MATLAB (Introduction),

illustrated at the example of the String Response task

Example: Dynamics Response of a Plucked String



Signs in the series are intermittently changing

$$y(x,t) = \frac{8h}{\pi^2}\left[\frac{1}{1^2}\sin\frac{\pi x}{L}\cos\frac{\pi ct}{L} - \frac{1}{3^2}\sin\frac{3\pi x}{L}\cos\frac{3\pi ct}{L} + \ldots\right] =$$

$$= \frac{8h}{\pi^2}\left[\frac{1}{1^2}\sin\frac{\pi x}{L}\cos\omega_1 t - \frac{1}{3^2}\sin\frac{3\pi x}{L}\cos 3\omega_1 t + \ldots\right]$$

We need to program a “Sign” Factor

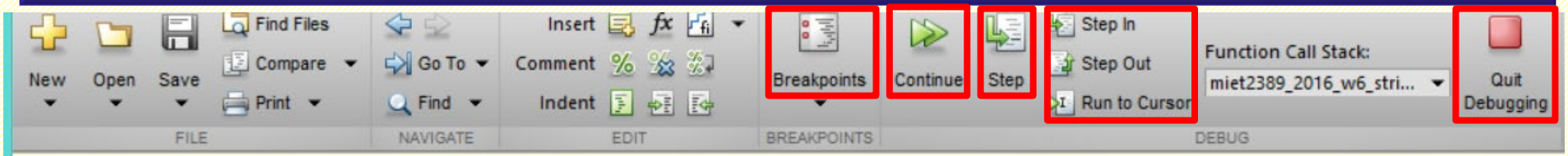
```
1   %% INTRODUCTION INTO DEBUGGER
2   %                        programmed by P.M.Trivailo
3   %-----------------------------------------------------------------
4   clear; clc; close('all');
5   N=5;
6   for ii=1:N
7       if mod(ii,2)==0,
8       else
9           mult=(-1)^((ii+1)/2+1);
10          disp(sprintf('ii=%2i; mult = %2i',ii,mult))
11      end
12  end
```

We need to program a “Sign” Factor

```
1     %% INTRODUCTION INTO DEBUGGER
2     %                    programmed by P.M.Trivailo
3     %-------------------------------------------------
4 -   clear; clc; close('all');
5 -   N=5;
6 -   for ii=1:N
7 -       if mod(ii,2)==0,
8 -       else
9 -           mult=(-1)^((ii+1)/2+1);
10            disp(sprintf('ii=%2i; mult = %2i',ii,mult))
11 -      end
12 -  end
```

SUSPENSION POINT: Execution of the program will be suspended here!

Some Options of the Actions

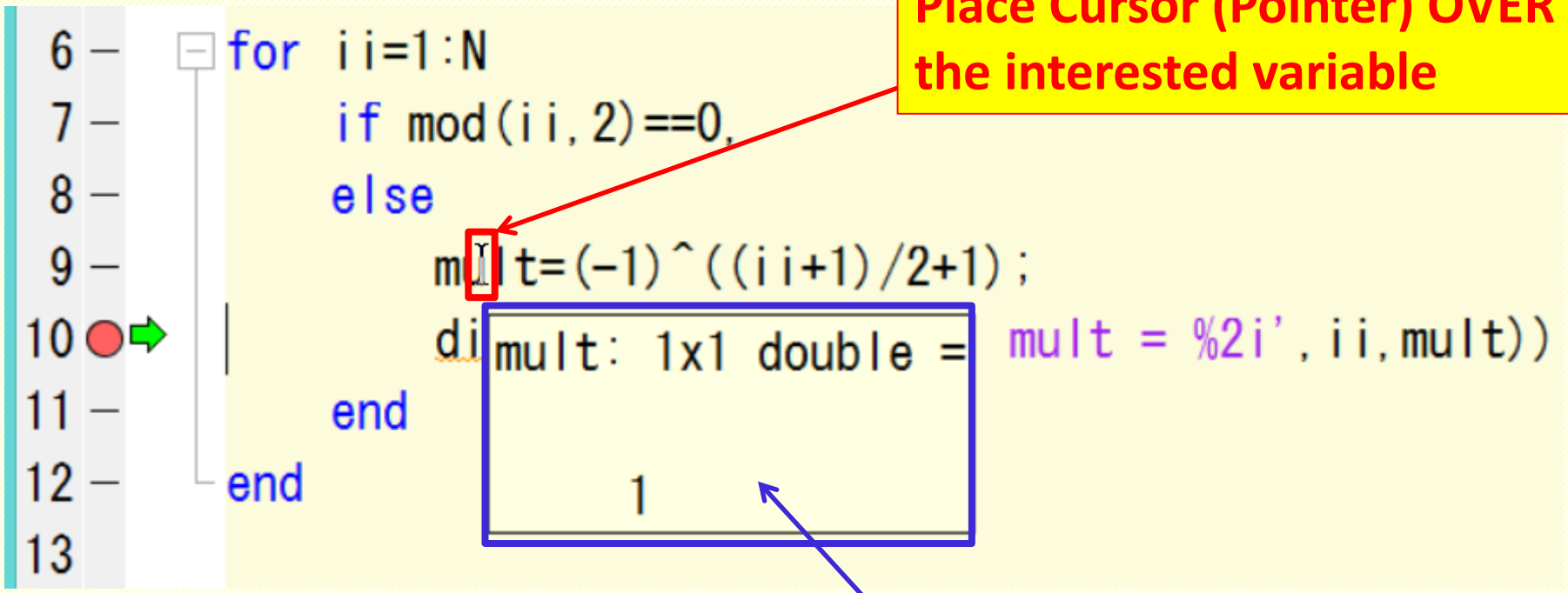


This file can be published to a formatted document. For more information, see the publishing video or help.

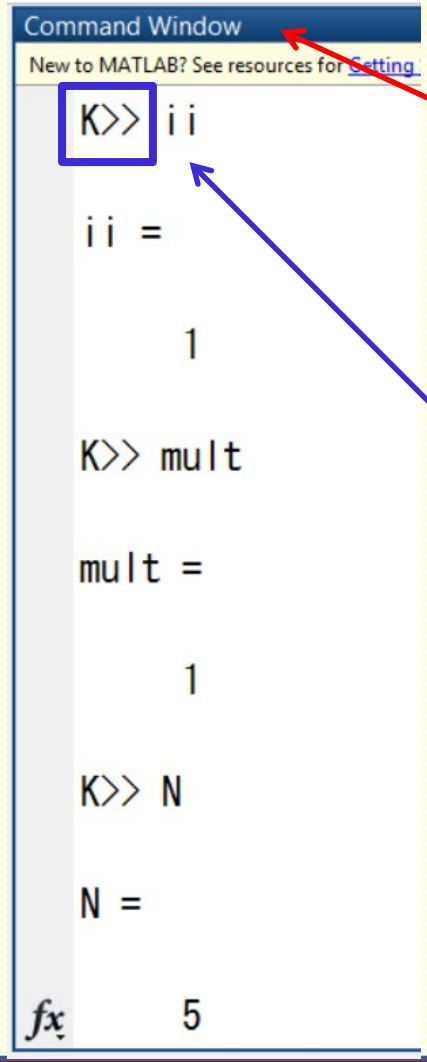
```
1     %% INTRODUCTION INTO DEBUGGER
2     %                                   by P.M.Trivailo
3     %-------------------------------------------------------------
4 -   clear; clc; close('all');
5 -   N=5;
6 -   for ii=1:N
7 -       if mod(ii,2)==0,
8 -       else
9 -           mult=(-1)^((ii+1)/2+1);
10 -          disp(sprintf('ii=%2i; mult = %2i',ii,mult))
11 -      end
12 -  end
```

Note: When the program is suspended, old points can be cleared and/or more points can be added !

Checking the Status of the Variables



Checking the Status of the Variables



"Information" on the variables can be obtained from the Command Window.

The variables, IF needed, can be also changed from the Command Window.

Note: Change in the sign here in the "debugging" mode!

Approximation of the Initial Shape of the String in MATLAB

(Experimenting with the number N of terms in the Analytical Solution)

Exercise-1: MATLAB Script

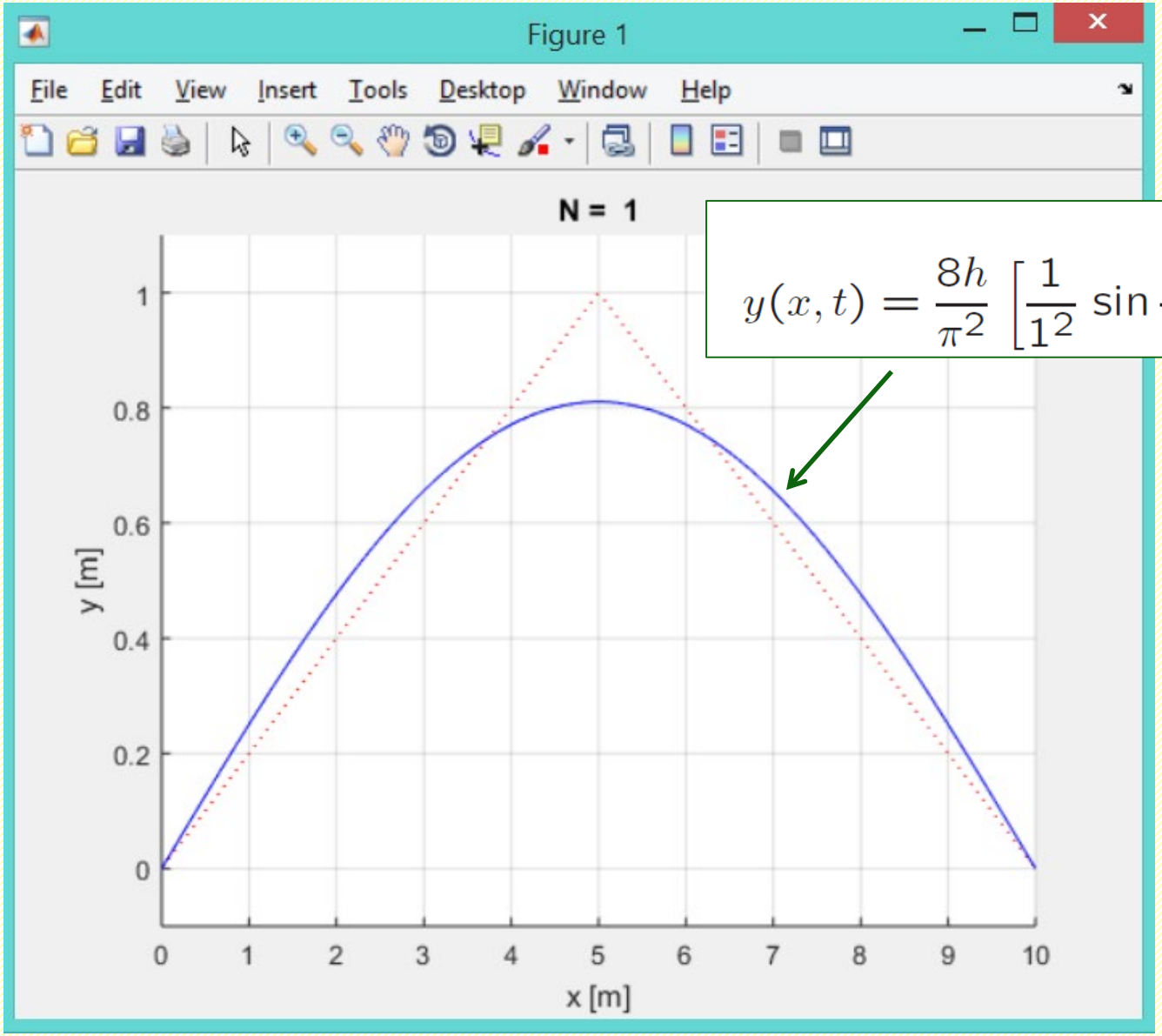
```
%% ACCURACY of APPROXIMATION OF THE STRING RESPONSE
%                                   Designed by Prof P.M.Trivailo
%----------------------------------------
clear; clc; close('all');
L=10; h=1; c=70;
w=pi*c/L; disp(sprintf('omega_1 = %g rad/s; T = %g s',w,2*pi/w))
x=[0:0.01:1]*L; t=0;
figure;
h1=line('XData',[0 .5 1]*L,'YData',[0 h 0],'LineStyle',':','Color',[1 0 0]);
axis([0 L [-0.1 1.1]*h ])
xlabel('x [m]'); ylabel('y [m]'); grid on;
h2=line('XData',x,'YData',NaN*x,'Color',[0 0 1]);
for N=1:5,
   y=0;
   for ii=1:N
      if mod(ii,2)==0,
      else
         mult=(-1)^((ii+1)/2+1);
         y=y + mult*(1/ii^2)*sin(ii*pi*x/L)*cos(ii*pi*c*t/L);
      end
   end
   y=y*(8*h/pi^2);
   set(h2,'YData',y);
   title(sprintf('N = %2i',N));
   drawnow;    pause(0.4);
end
```

Shape for initial time (only) is plotted: i.e. t=0.

Change the upper limit in the cycle to assign the number of terms in the approximation of the initial shape of the string

$$y(x,t) = \frac{8h}{\pi^2}\left[\frac{1}{1^2}\sin\frac{\pi x}{L}\cos\frac{\pi ct}{L} - \frac{1}{3^2}\sin\frac{3\pi x}{L}\cos\frac{3\pi ct}{L} + \ldots\right]$$

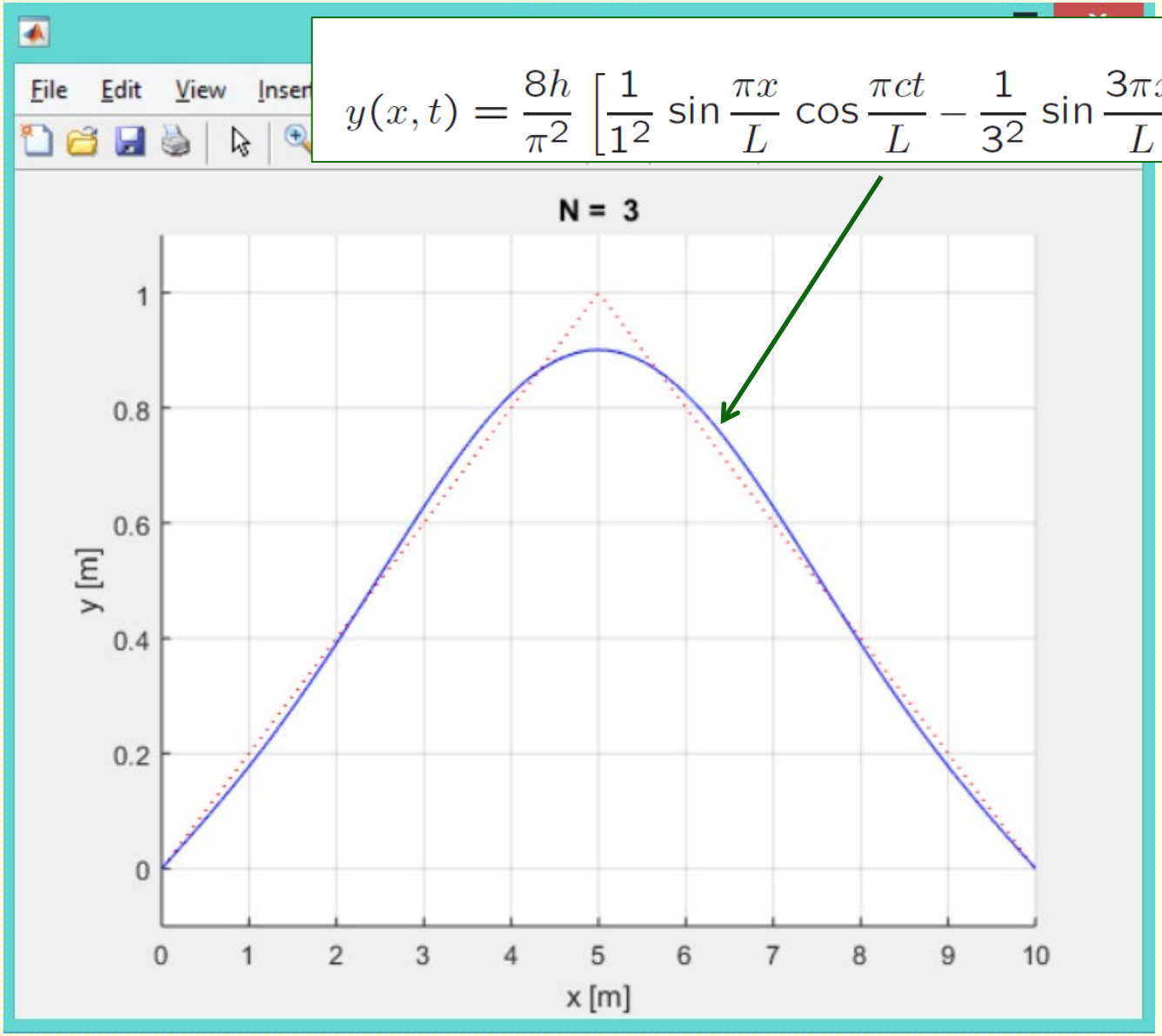
Results for N=1:



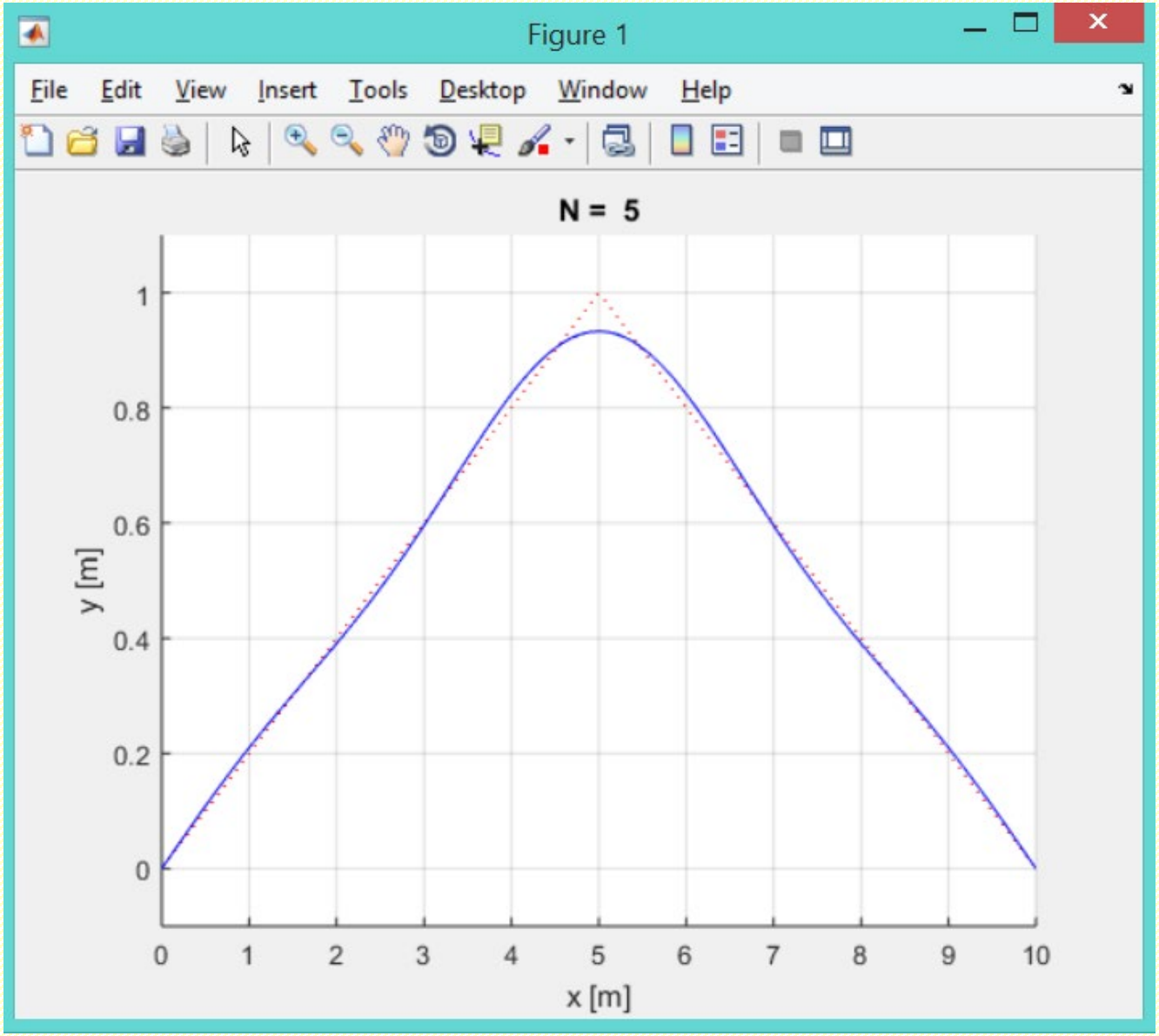
$$y(x,t) = \frac{8h}{\pi^2}\left[\frac{1}{1^2}\sin\frac{\pi x}{L}\cos\frac{\pi ct}{L}\right]$$

Results for N=3:

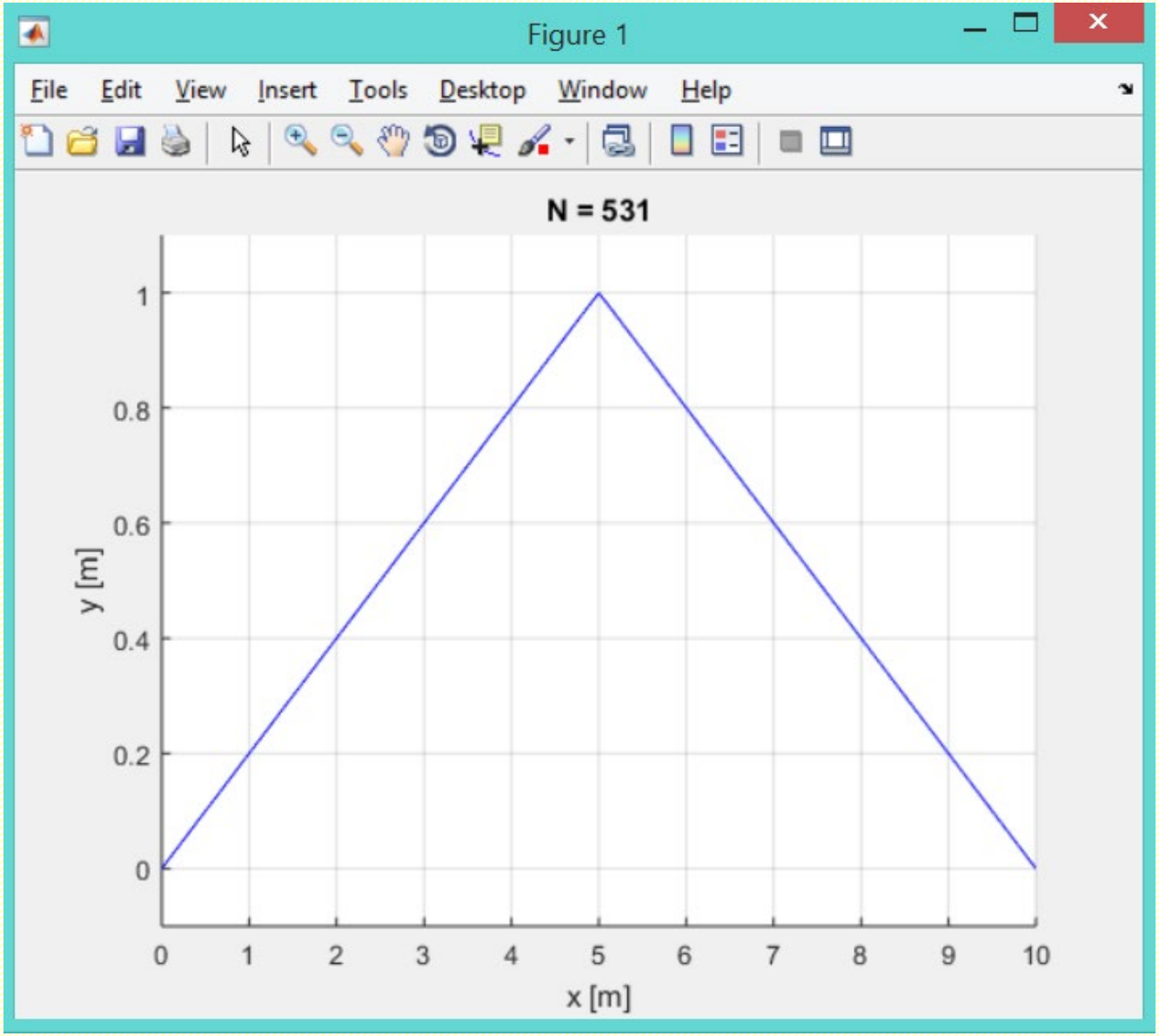
$$y(x,t) = \frac{8h}{\pi^2}\left[\frac{1}{1^2}\sin\frac{\pi x}{L}\cos\frac{\pi ct}{L} - \frac{1}{3^2}\sin\frac{3\pi x}{L}\cos\frac{3\pi ct}{L}\right]$$



Results for N=5:



Results for N=531:



MATLAB:

PLOTTING 3D SURFACES

("meshgrid" and "surf")

MATLAB:

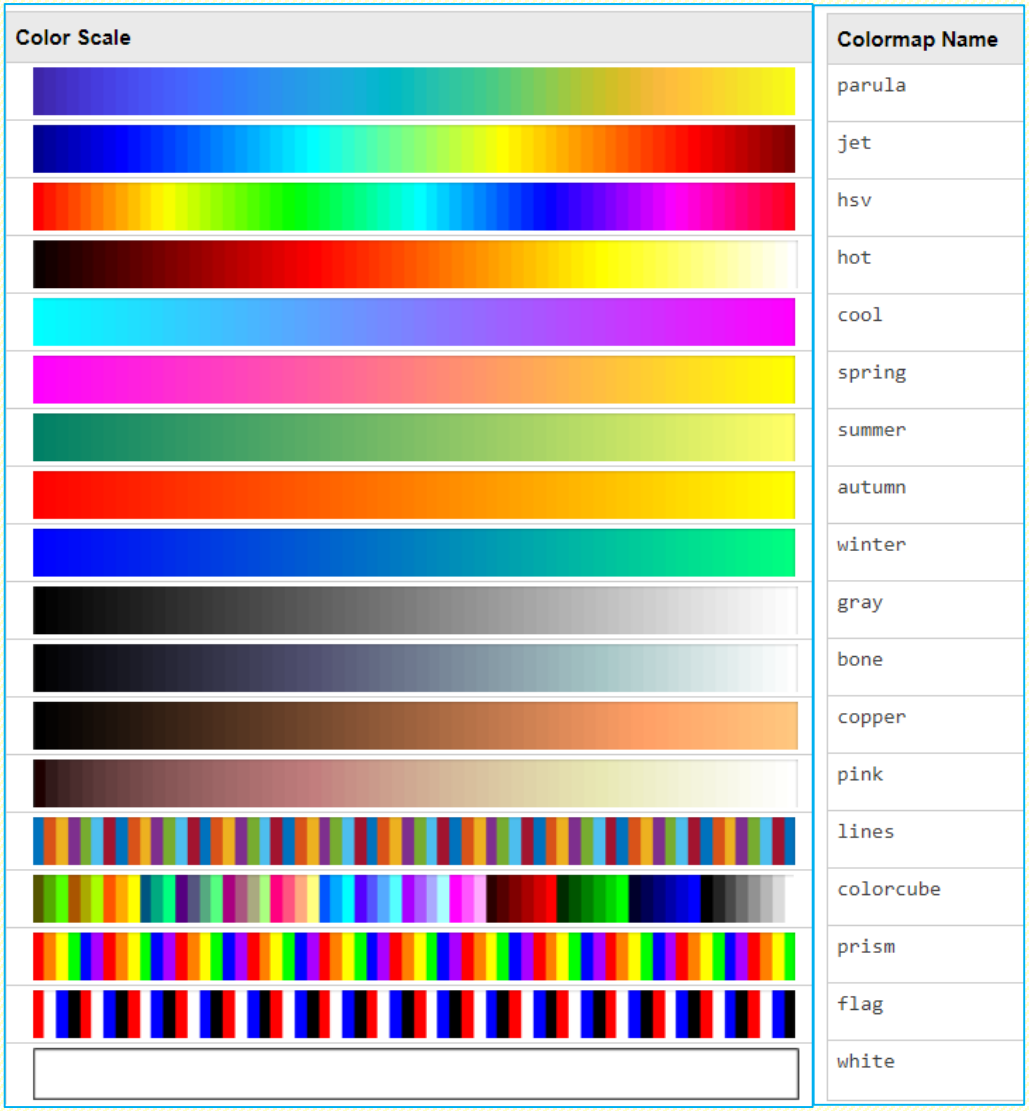
Understanding Colormaps (examples of the same plot, presented with different colormaps)

MATLAB: BUILT-IN COLORMAPS

What Is a Colormap?

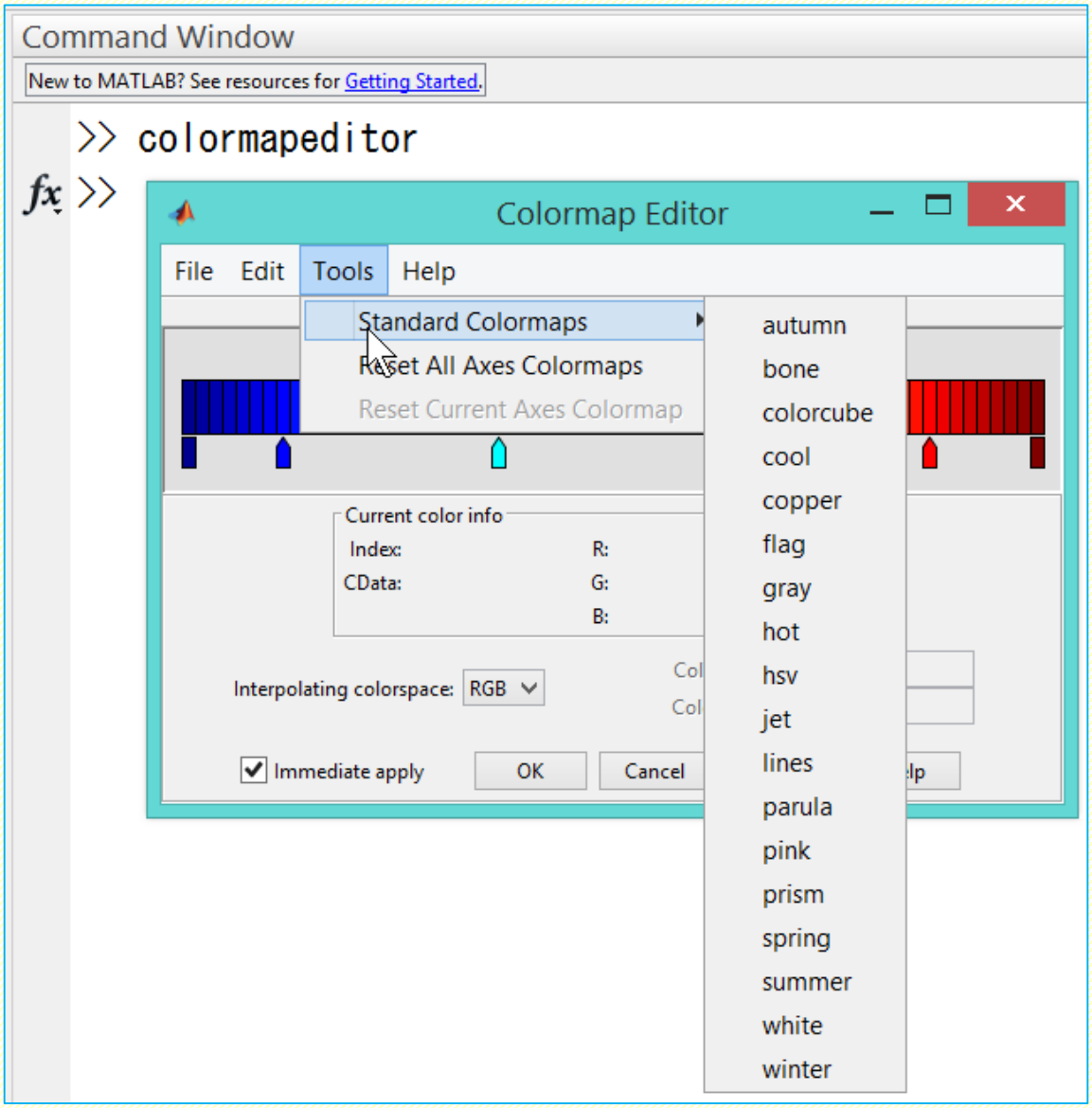
A colormap is matrix of values between 0 and 1 that define the colors for graphics objects such as surface, image, and patch objects. MATLAB® draws the objects by mapping data values to colors in the colormap.

Colormaps can be any length, but must be three columns wide. Each row in the matrix defines one color using an RGB triplet. An RGB triplet is a three-element row vector whose elements specify the intensities of the red, green, and blue components of the color. The intensities must be in the range [0, 1]. A value of 0 indicates no color and a value of 1 indicates full intensity.



Curtesy: https://au.mathworks.com/help/matlab/ref/colormap.html

MATLAB: STANDARD COLORMAPS



MATLAB: CUSTOM COLORMAPS

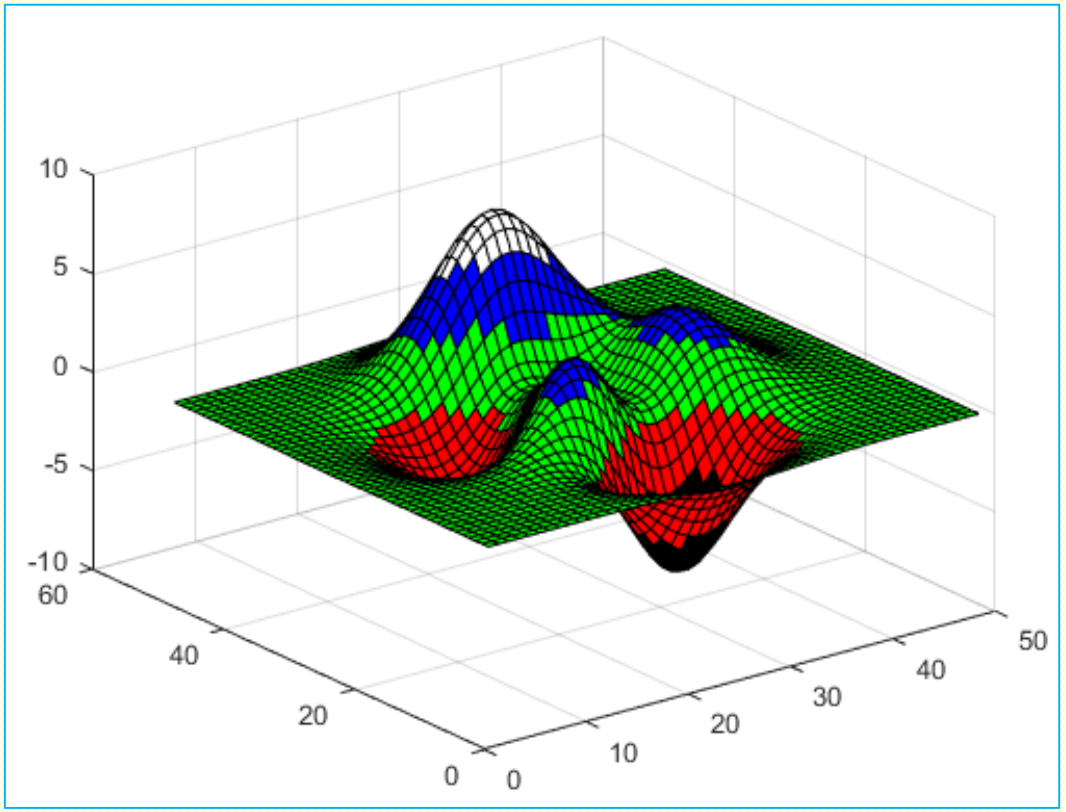
Custom-Built Colormap: Example

For example, this command creates a colormap that has five colors: black, red, green, blue, and white.

```
mymap = [0 0 0
   1 0 0
   0 1 0
   0 0 1
   1 1 1];
```

To change the color scheme of a visualization, call the colormap function to change the colormap of the containing axes or figure. For example, the following commands create a surface plot and set the colormap of the figure to **mymap**.

```
surf(peaks)
colormap(mymap)
```



Curtesy: https://au.mathworks.com/help/matlab/ref/colormap.html

MATLAB: Plot BUILT-IN COLORMAPS

Curtesy: https://au.mathworks.com/matlabcentral/fileexchange/35242-matlab-plot-gallery-colormap-chart?focused=6792976&tab=example

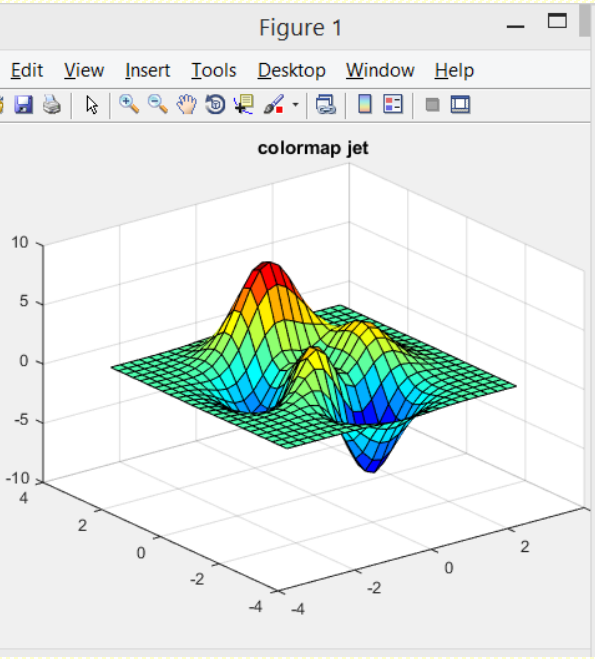


Figure 1
Edit View Insert Tools Desktop Window Help
colormap jet

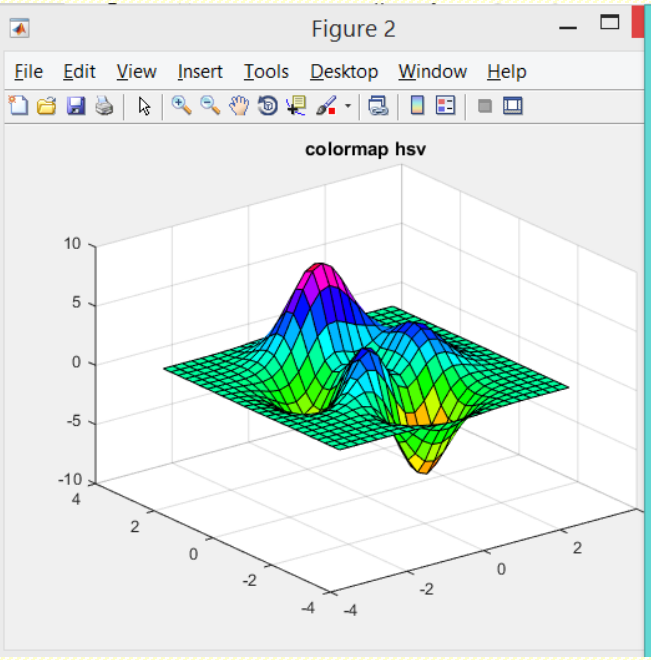


Figure 2
File Edit View Insert Tools Desktop Window Help
colormap hsv

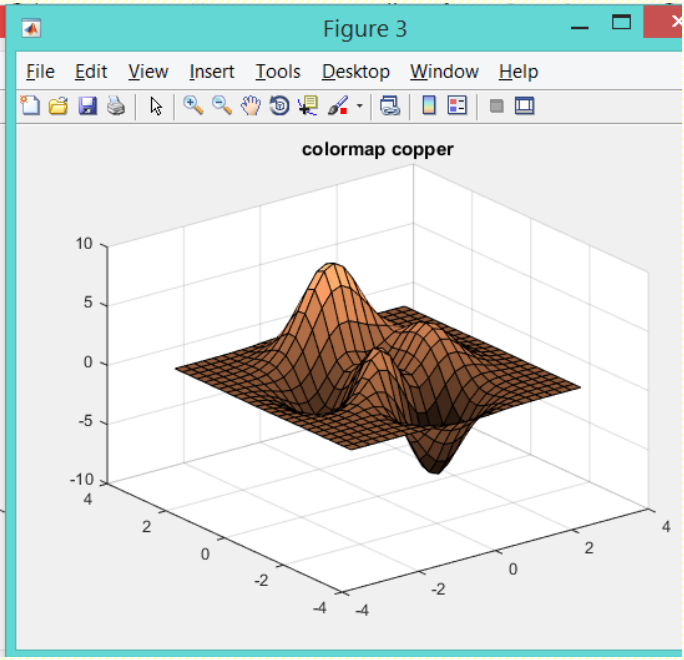


Figure 3
File Edit View Insert Tools Desktop Window Help
colormap copper

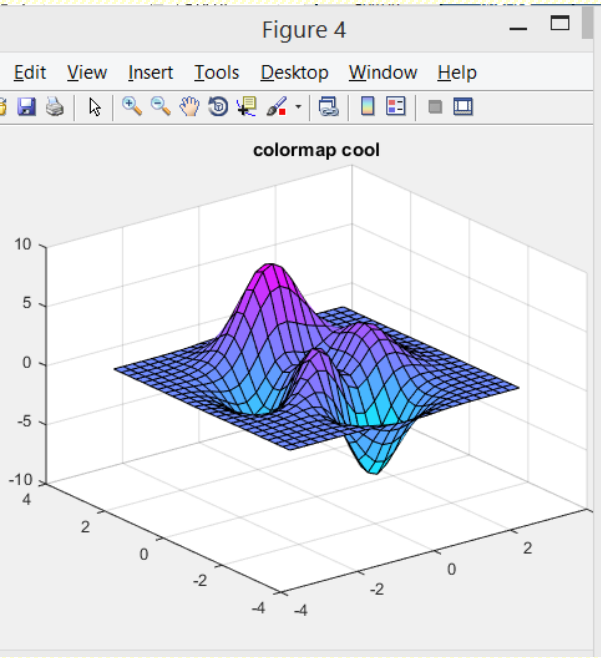


Figure 4
Edit View Insert Tools Desktop Window Help
colormap cool

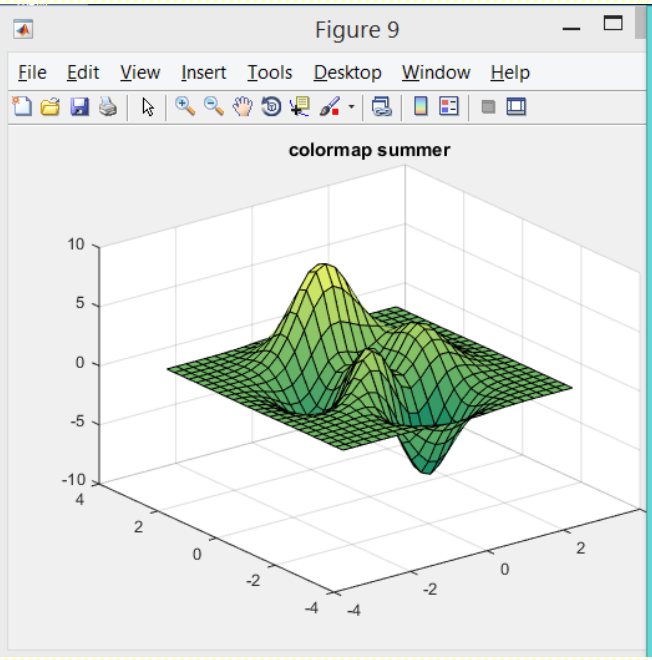


Figure 9
File Edit View Insert Tools Desktop Window Help
colormap summer

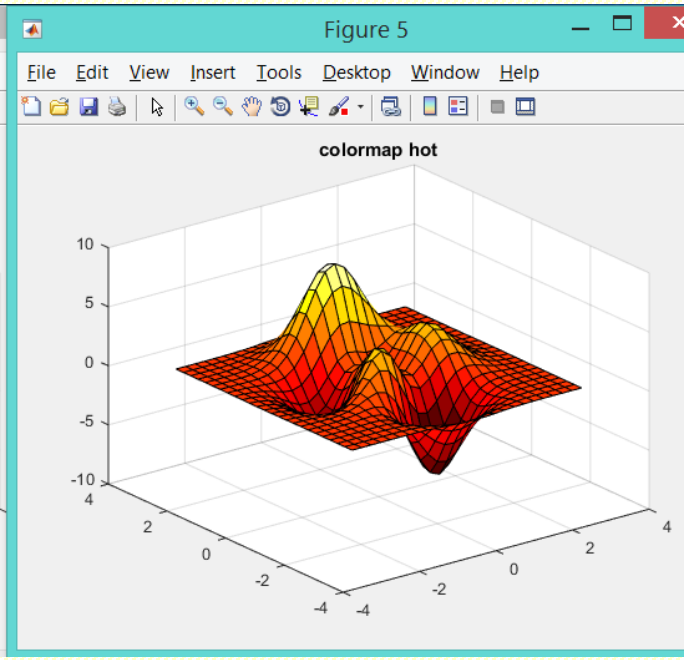


Figure 5
File Edit View Insert Tools Desktop Window Help
colormap hot

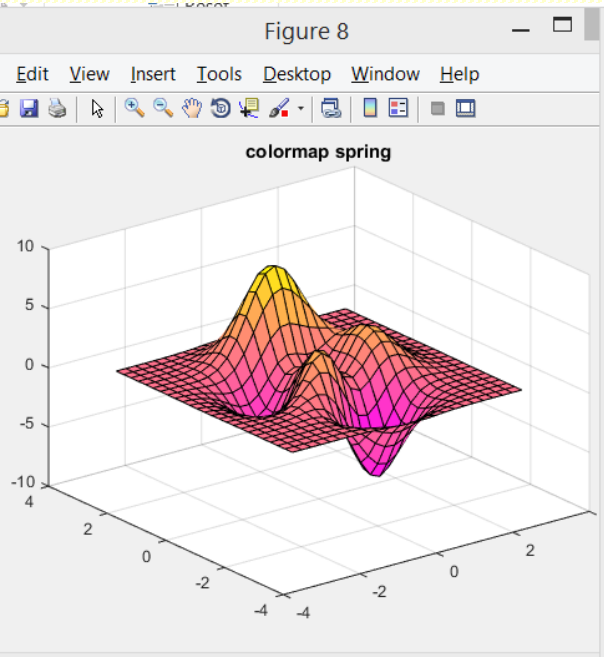


Figure 8
Edit View Insert Tools Desktop Window Help
colormap spring

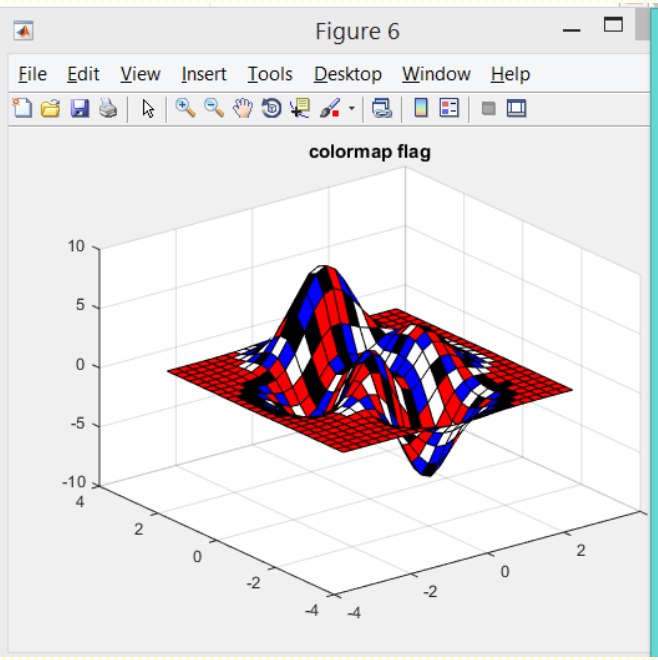


Figure 6
File Edit View Insert Tools Desktop Window Help
colormap flag

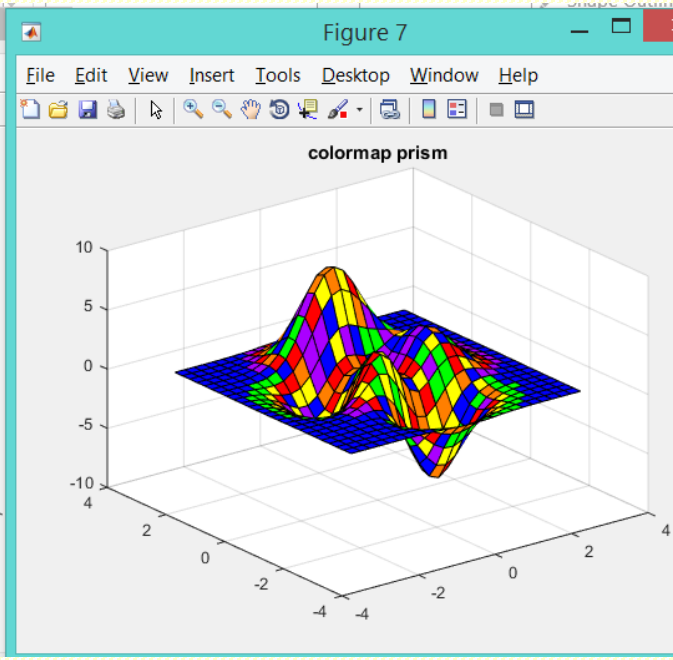


Figure 7
File Edit View Insert Tools Desktop Window Help
colormap prism

MATLAB Script for the "colormap" examples

```
%% OENG1116-S1-2020
% Designed by Prof P.M.Trivailo (C) 2020
%--------------------------
figure;
[X,Y,Z] = peaks(25);
surf(X,Y,Z);
colormap jet;
title('colormap jet')
pause(1)
%------------------
figure;
surf(X,Y,Z);
colormap hsv;
title('colormap hsv')
pause(1)
%------------------
figure;
surf(X,Y,Z);
colormap copper;
title('colormap copper')
pause(1)
%------------------
figure;
surf(X,Y,Z);
colormap cool;
title('colormap cool')
pause(1)
%------------------
```

```
%------------------
figure;
surf(X,Y,Z);
colormap hot;
title('colormap hot')
pause(1)
%------------------
figure;
surf(X,Y,Z);
colormap summer;
title('colormap summer')
pause(1)
%------------------
figure;
surf(X,Y,Z);
colormap spring;
title('colormap spring')
pause(1)
%------------------
figure;
surf(X,Y,Z);
colormap flag;
title('colormap flag')
pause(1)
%------------------
figure;
surf(X,Y,Z);
colormap prism;
title('colormap prism')
```

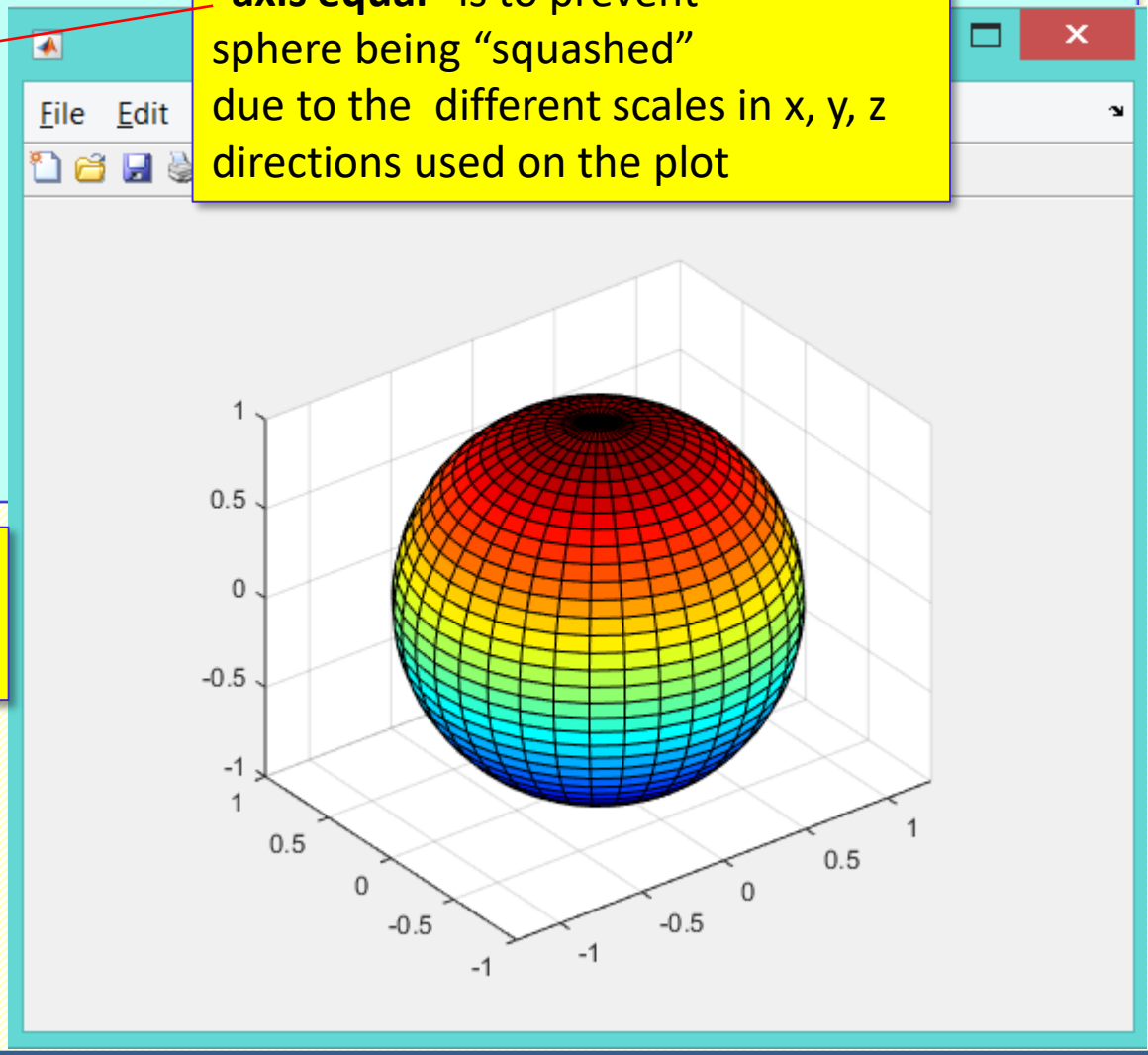
MATLAB:

Plotting Spheres with various Options

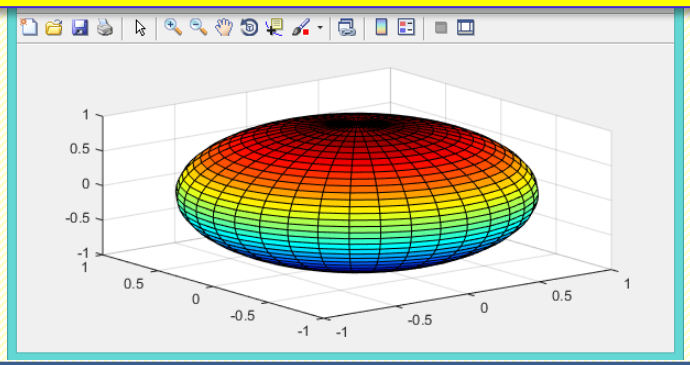
MATLAB: "sphere" & "surf"

```
%%
clear; close all; clc;
figure; hold on;
axis equal; grid on

colormap jet;
[x,y,z]=sphere(36);
surf(x,y,z);
view(3);
```

"**axis equal**" is to prevent sphere being "squashed" due to the different scales in x, y, z directions used on the plot

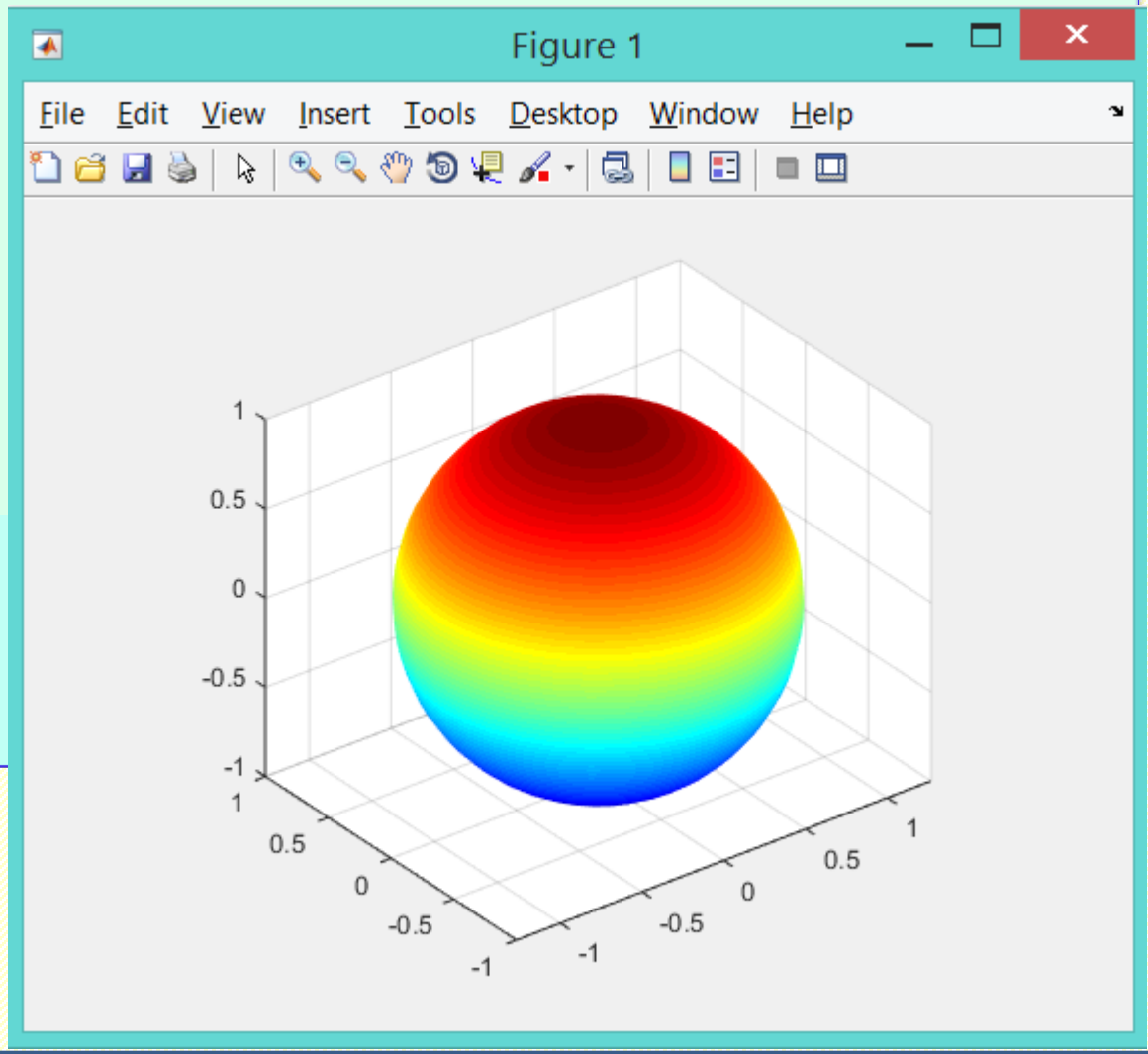


Without "**axis equal**" sphere can be seen as "squashed" due to the different scales in x, y, z directions used on the plot



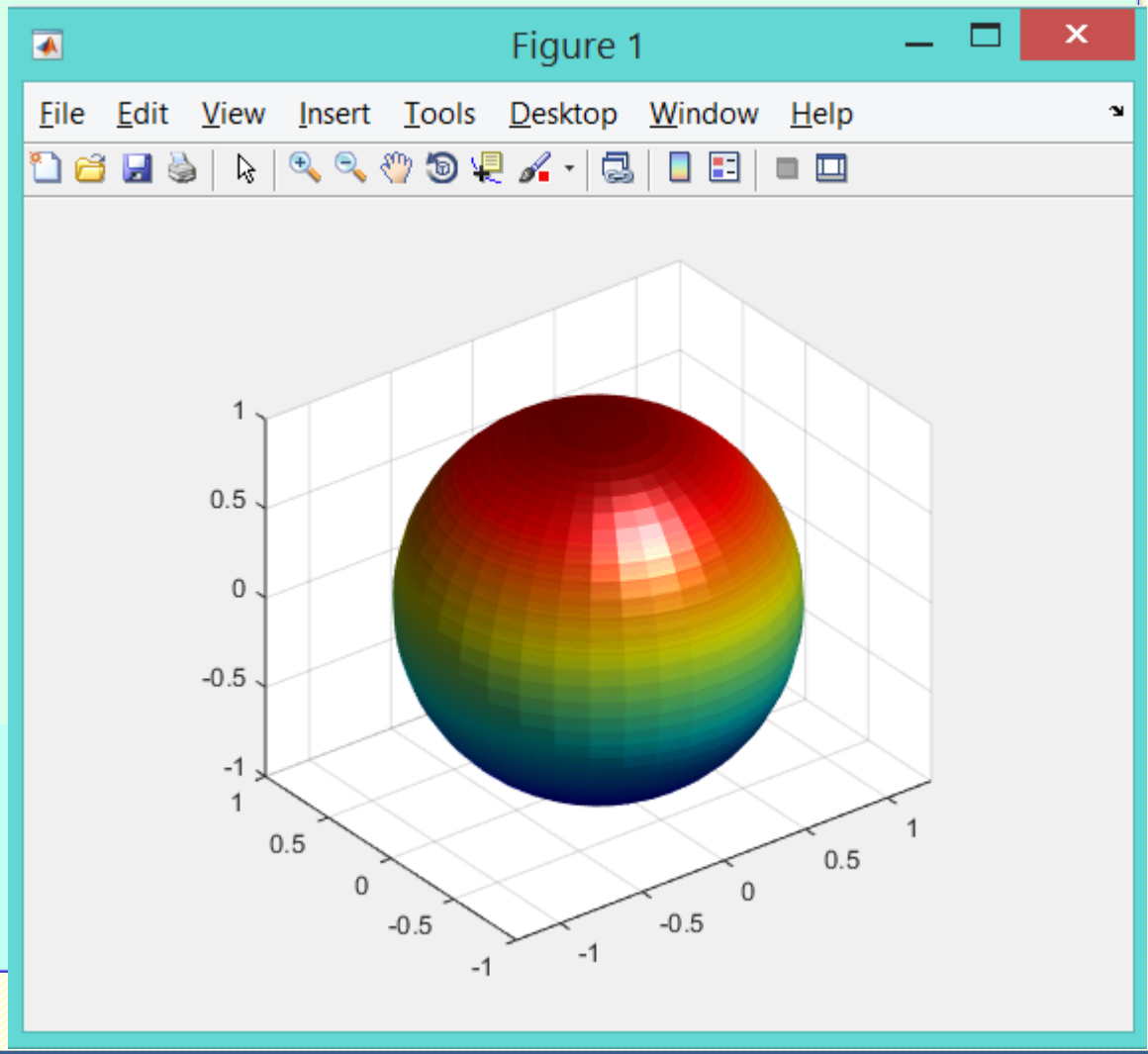
MATLAB: "shading"

```
%%
clear; close all; clc;
figure; hold on;
axis equal; grid on

colormap jet;
[x,y,z]=sphere(36);
surf(x,y,z);
view(3);

%%
shading interp;
```



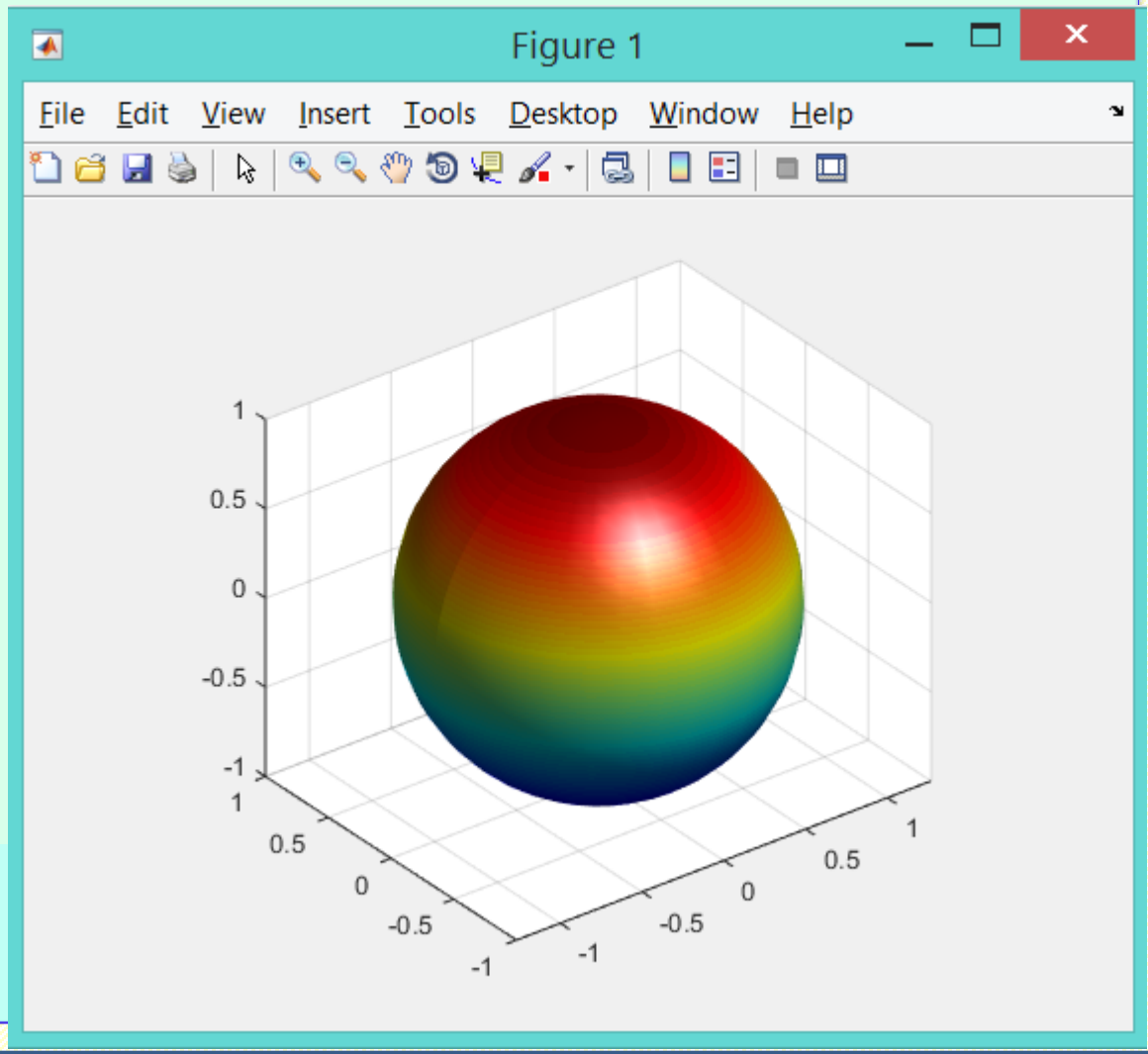
MATLAB: "camlight" & "light"

```
%%
clear; close all; clc;
figure; hold on;
axis equal; grid on

colormap jet;
[x,y,z]=sphere(36);
surf(x,y,z);
view(3);

%%
shading interp;

%%
camlight right;
```



MATLAB: "lighting"

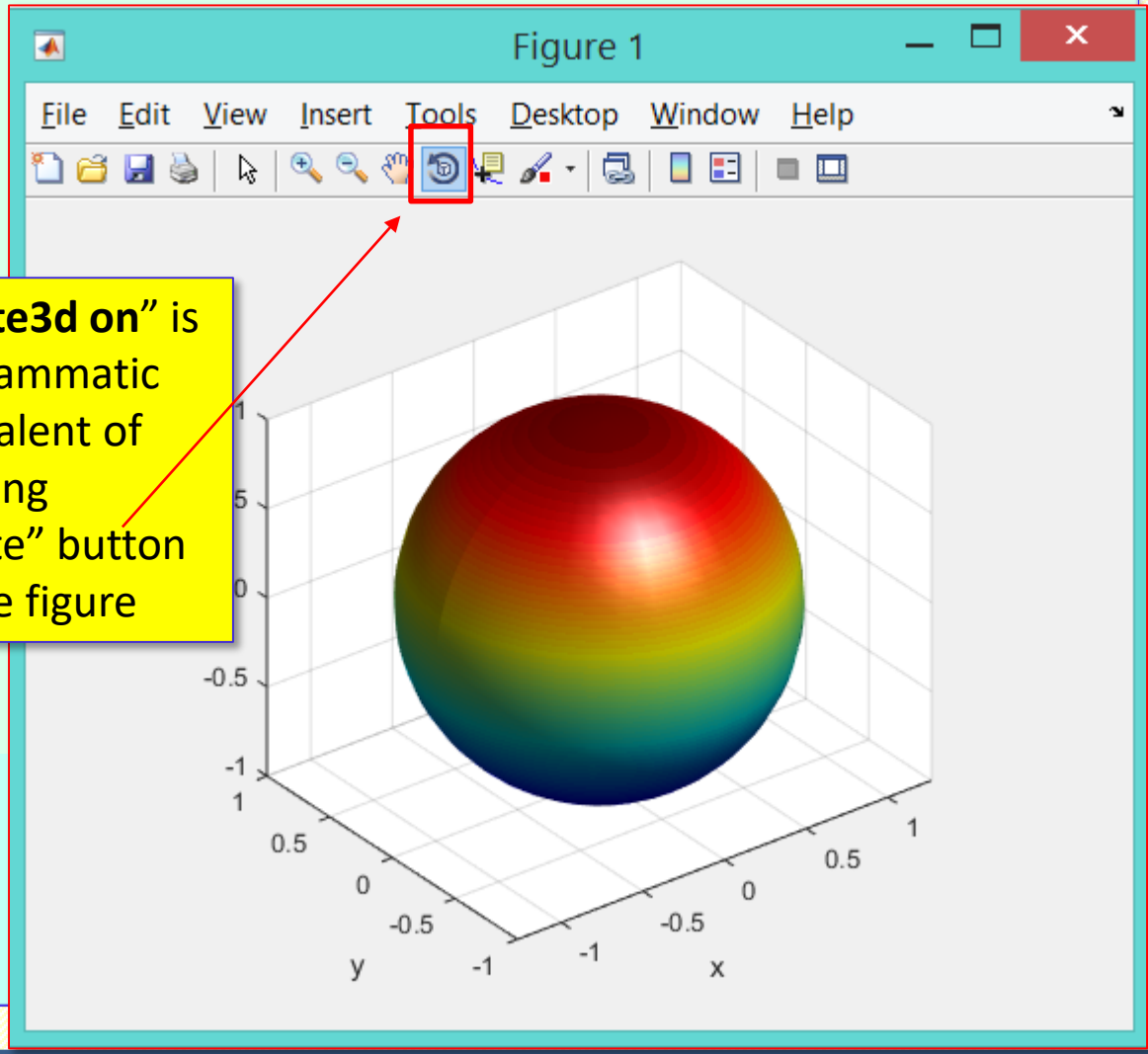
```
%%
clear; close all; clc;
figure; hold on;
axis equal; grid on

colormap jet;
[x,y,z]=sphere(36);
surf(x,y,z);
view(3);

%%
shading interp;

%%
camlight right;

%%
lighting phong;
```



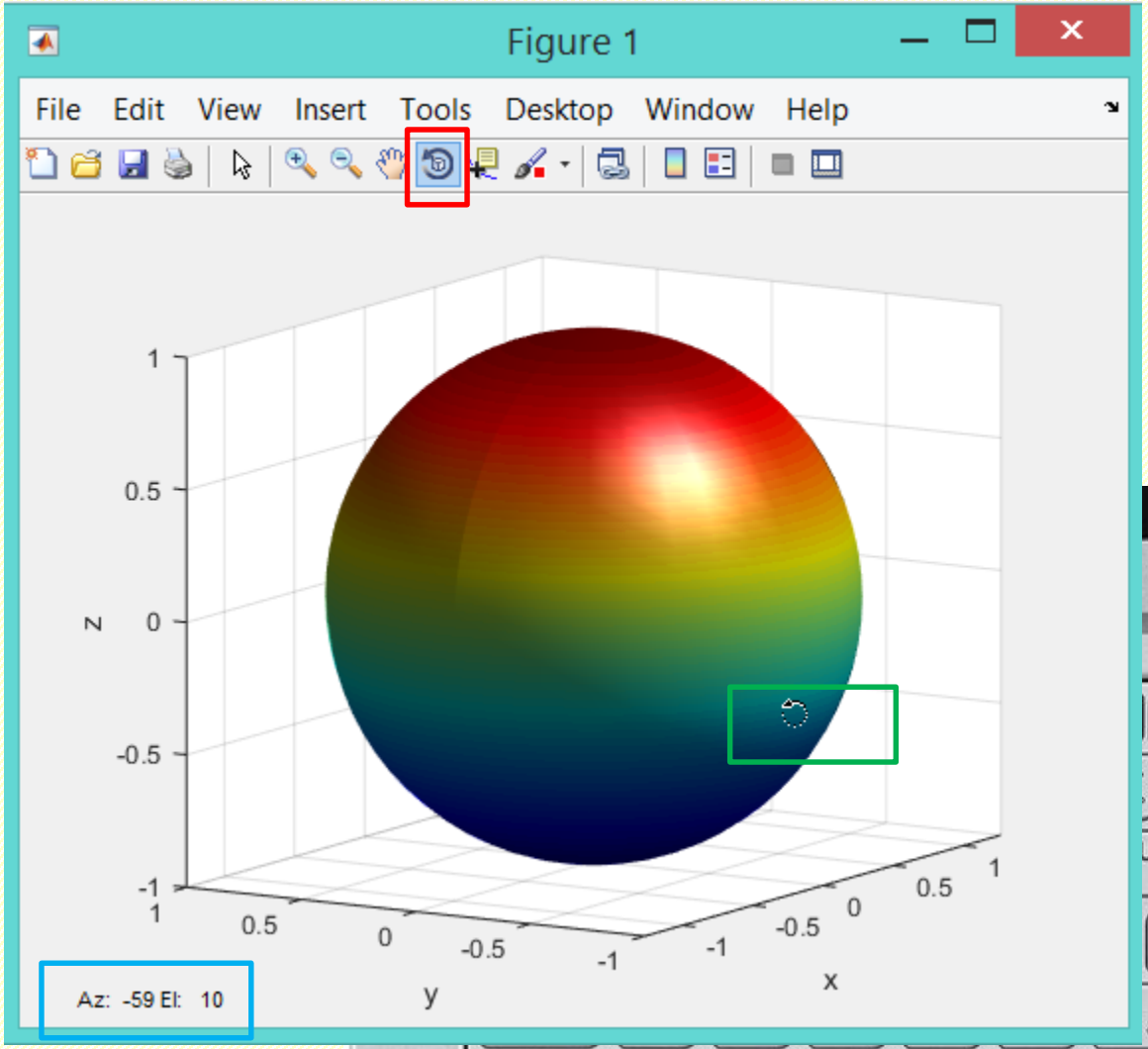
MATLAB: "rotate3d"

```
%%
clear; close all; clc;
figure; hold on;
axis equal; grid on

colormap jet;
[x,y,z]=sphere(36);
surf(x,y,z);
view(3);
%%
shading interp;
%%
camlight right;
%%
lighting phong;

%%
xlabel('x');
ylabel('y'); zlabel('z');
rotate3d on
```

"**rotate3d on**" is programmatic equivalent of pressing "rotate" button on the figure

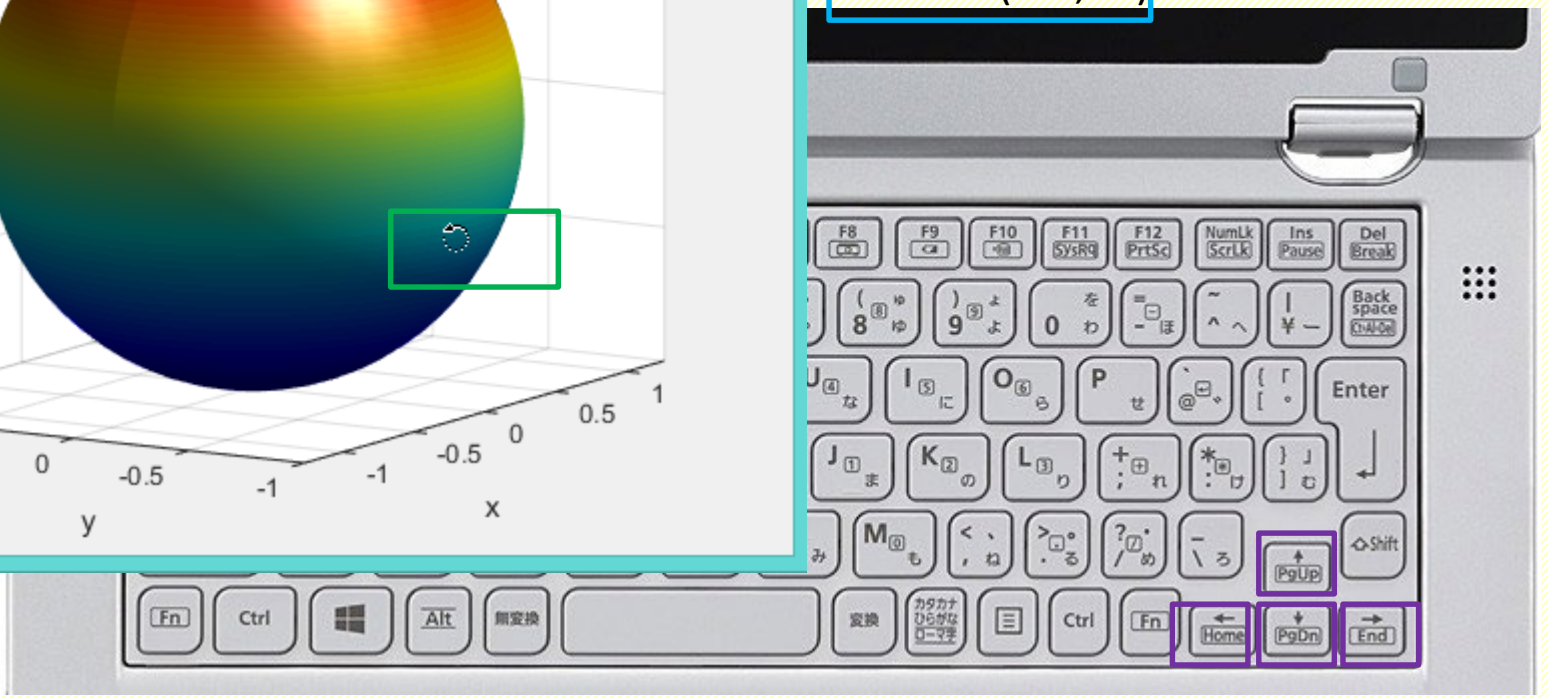


MATLAB: "rotate3d" & "view"



After "rotate3d" is set to ON, you can use MOUSE or KEYBOARD (Up-Right-Down-Left Buttons) or "view" command to change the view angles [for example,

>> view(-59,10)



MATLAB:

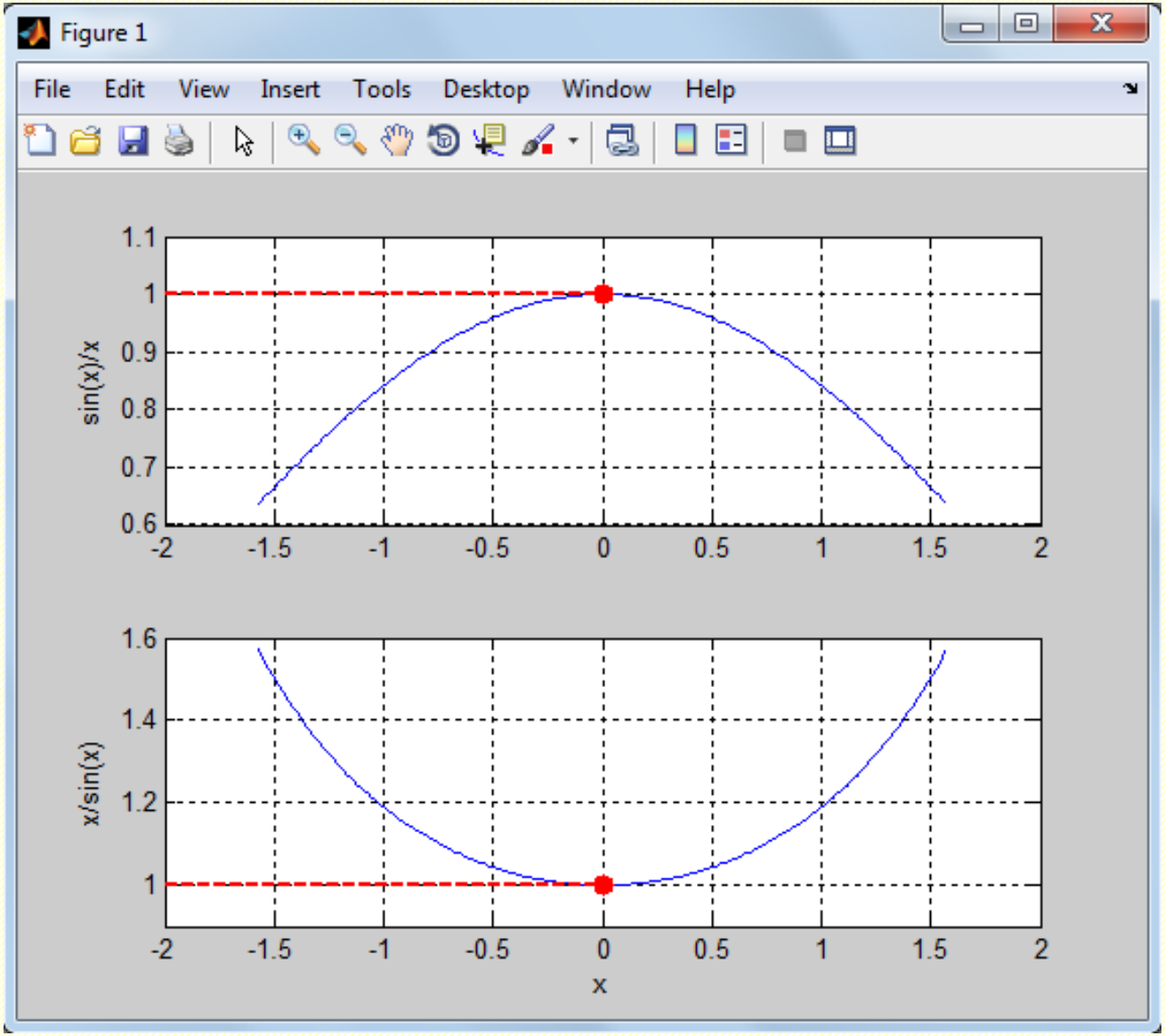
TRICK !!!

WOW !!!

Introduction: MATLAB: NaN

```
%%          Work in Class
% Determine numerically the limit of sin(x)/x when x->0
% Example-1
%-------------------
x=[-1:0.01:1]*(pi/2);
y1=sin(x)./x;
y2=x./sin(x);
subplot(2,1,1); plot(x,y1);
grid on; ylabel('sin(x)/x');
line(0,1,'Marker','.','MarkerSize',24,'Color',[1 0 0]);
axis([-2 2 0.6 1.1]);
line([-2 0],[1 1],'LineStyle','--','Color',[1 0 0],'LineWidth',2);

%% Example-2
%-------------------
subplot(2,1,2); plot(x,y2);
grid on; ylabel('x/sin(x)'); xlabel('x');
line(0,1,'Marker','.','MarkerSize',24,'Color',[1 0 0]);
line([-2 0],[1 1],'LineStyle','--','Color',[1 0 0],'LineWidth',2);
axis([-2 2 0.9 1.6]);
```

Introduction: MATLAB: NaN



Introduction: MATLAB: NaN

$$\lim_{x \to 0} \frac{\sin(x)}{x} = ?$$

```
>> syms x;
>> limit(sin(x)/x)
```

```
ans =

1
```

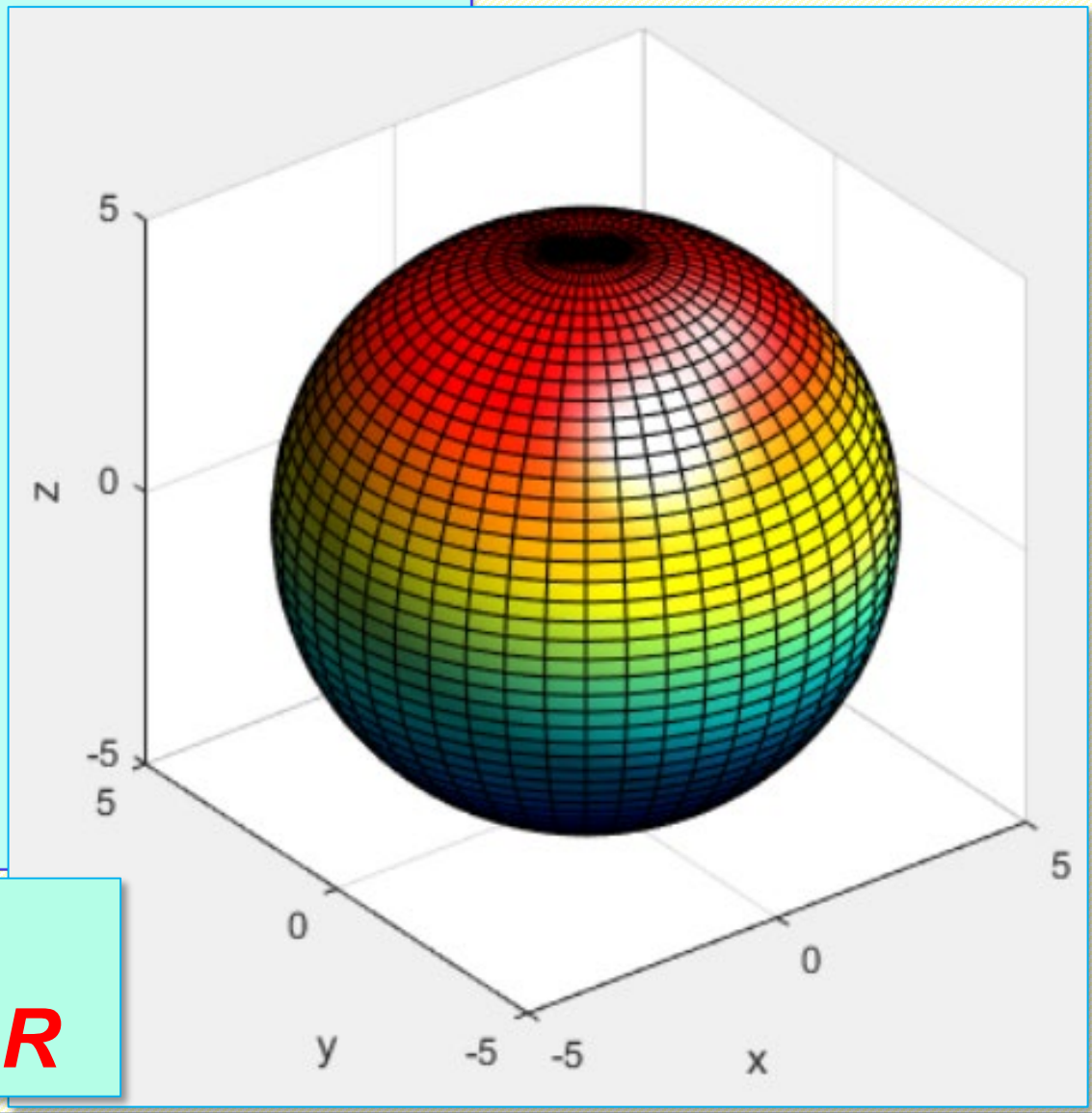
$$\frac{0}{0} = ?$$

```
>> 0/0
>> inf/inf
```

```
ans =

NaN
```

Introduction: MATLAB: just a SPHERE

```
R=5; quality=48;
[x,y,z]=sphere(quality);
x=R*x; y=R*y; z=R*z;
surf(x,y,z);
colormap jet;
camlight right;
lighting phong;
view(3); rotate3d on;
camlight right; light;
xlabel('x');
ylabel('y');
zlabel('z');
axis equal tight;
grid on
```

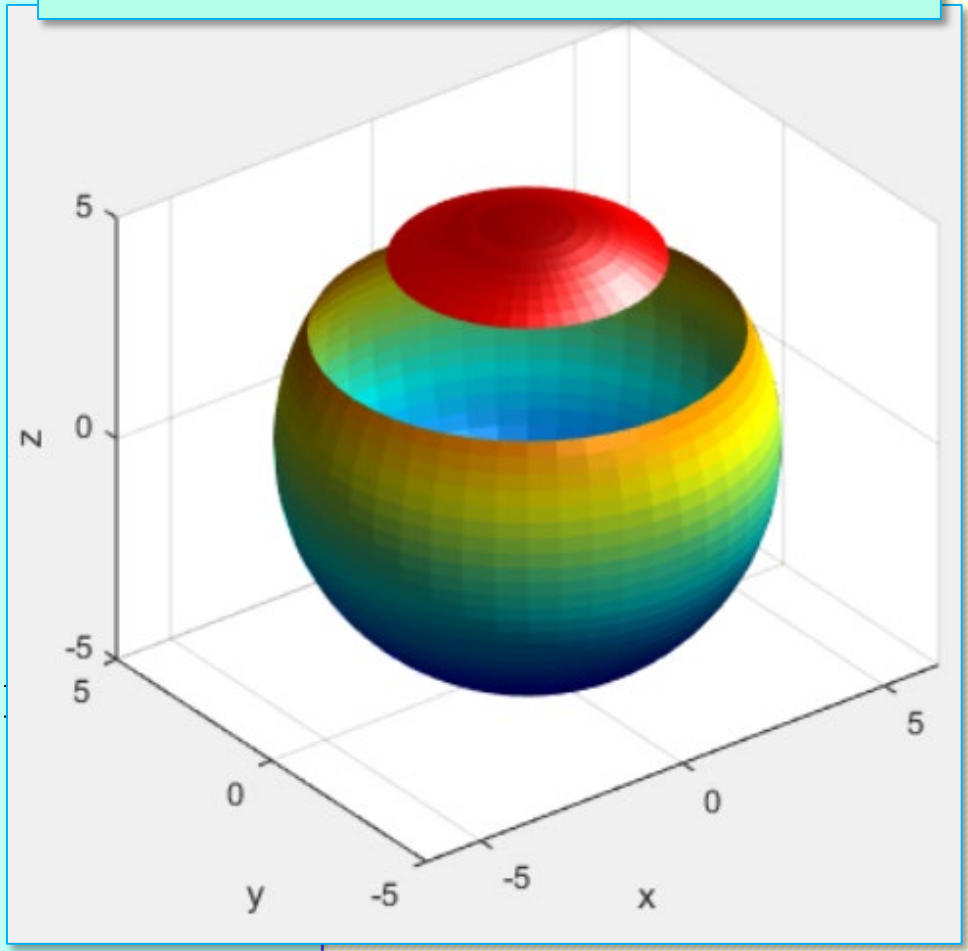


Producing a SPHERE of radius *R*

MATLAB: SPHERE and NaN

```
%% Designed by Prof P.M.Trivailo
clear; close all; clc; figure;
R=5; quality=48;
[x,y,z]=sphere(quality);
x=R*x; y=R*y; z=R*z;
view(3)
idx=find(z>2.5 & z<4);
z(idx)=NaN;
S1=surf(x,y,z);
camlight right; light;
shading flat;

axis equal tight; grid on
xlabel('x'); ylabel('y'); zla
rotate3d on
colormap jet;
```

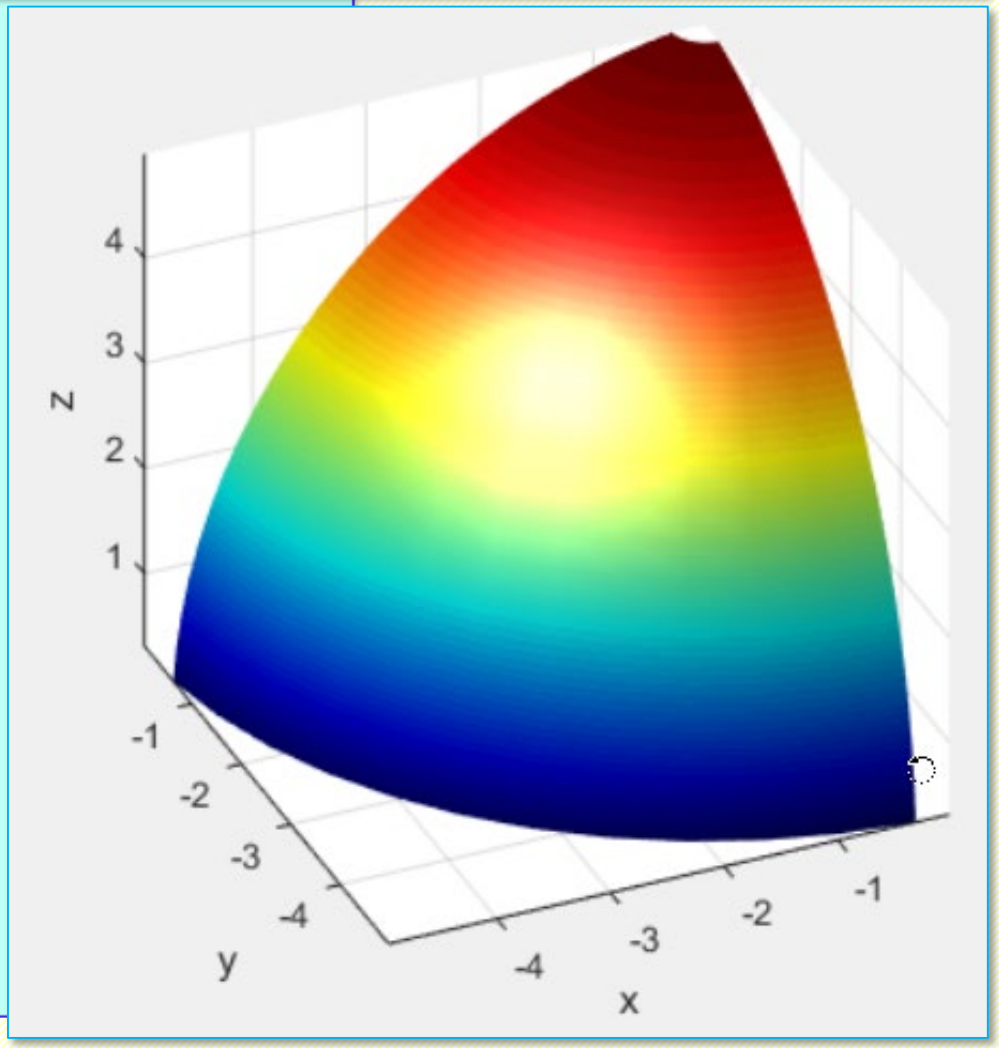
REMOVING a "BELT" from a SPHERE



MATLAB: SPHERE and NaN

```
close all; clc; clear;
R=5; quality=50;
[X,Y,Z]=sphere(quality);
X=R*X; Y=R*Y; Z=R*Z;

Z(find(Z<=0))= NaN;
X(find(X>=0))= NaN;
Y(find(Y>=0))= NaN;

h=surf(X,Y,Z);
axis equal tight;
xlabel('x'); ylabel('y');
zlabel('z');
set(gca,'View', [-23.5 32]);
colormap jet
shading INTERP
lighting PHONG
camlight left
rotate3d ON
```

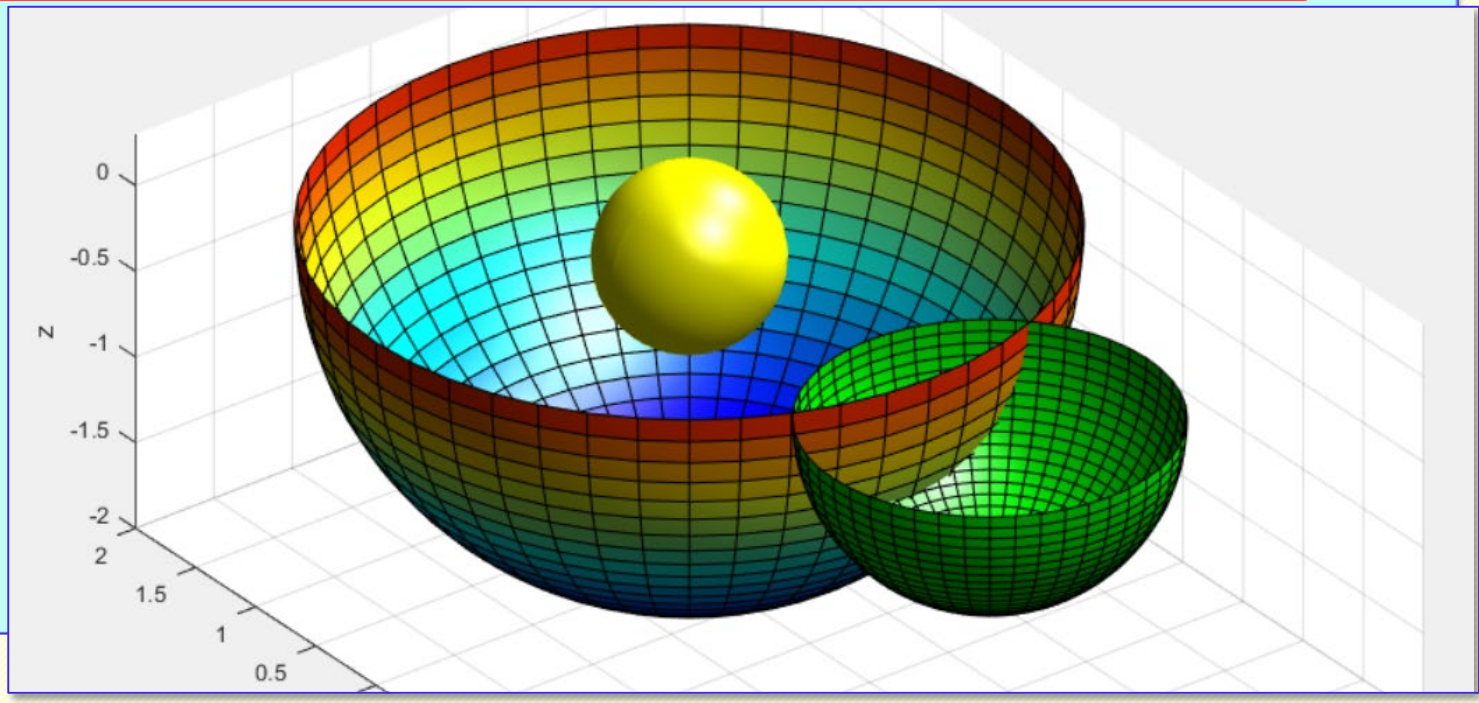
PRODUCING 1/8-th of a SPHERE

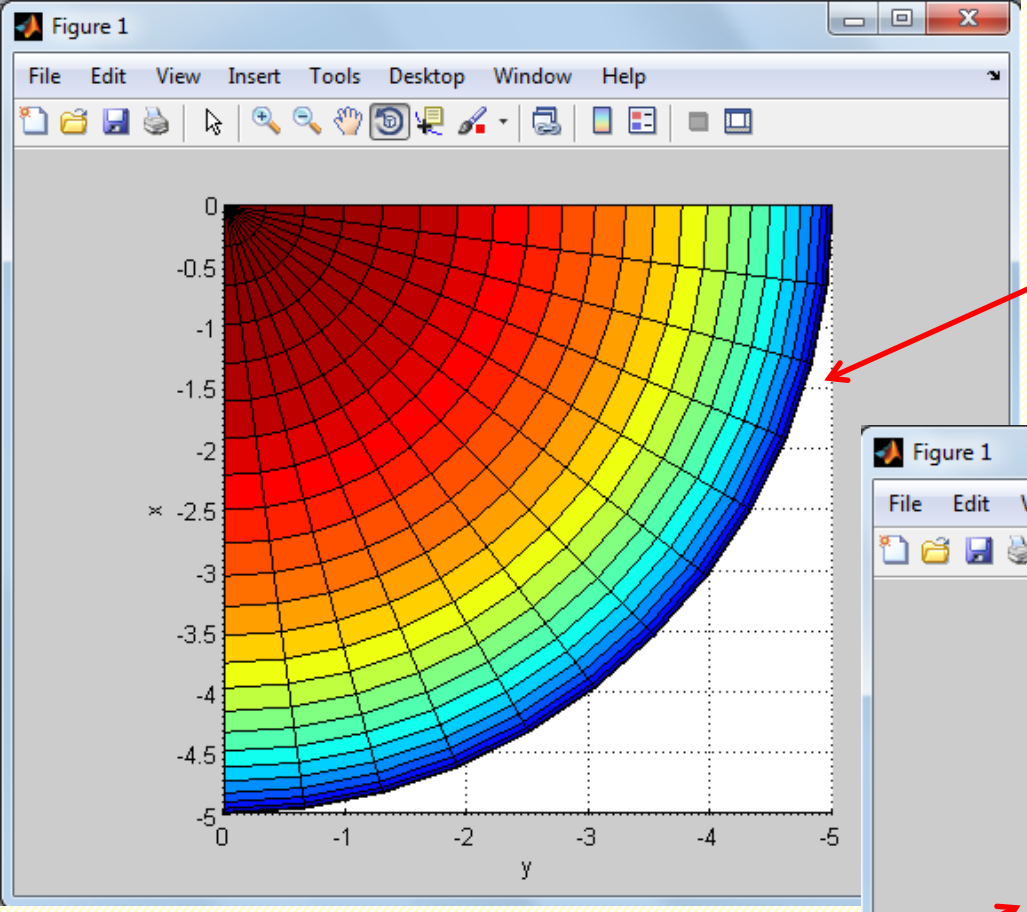


MATLAB: SPHERE and NaN

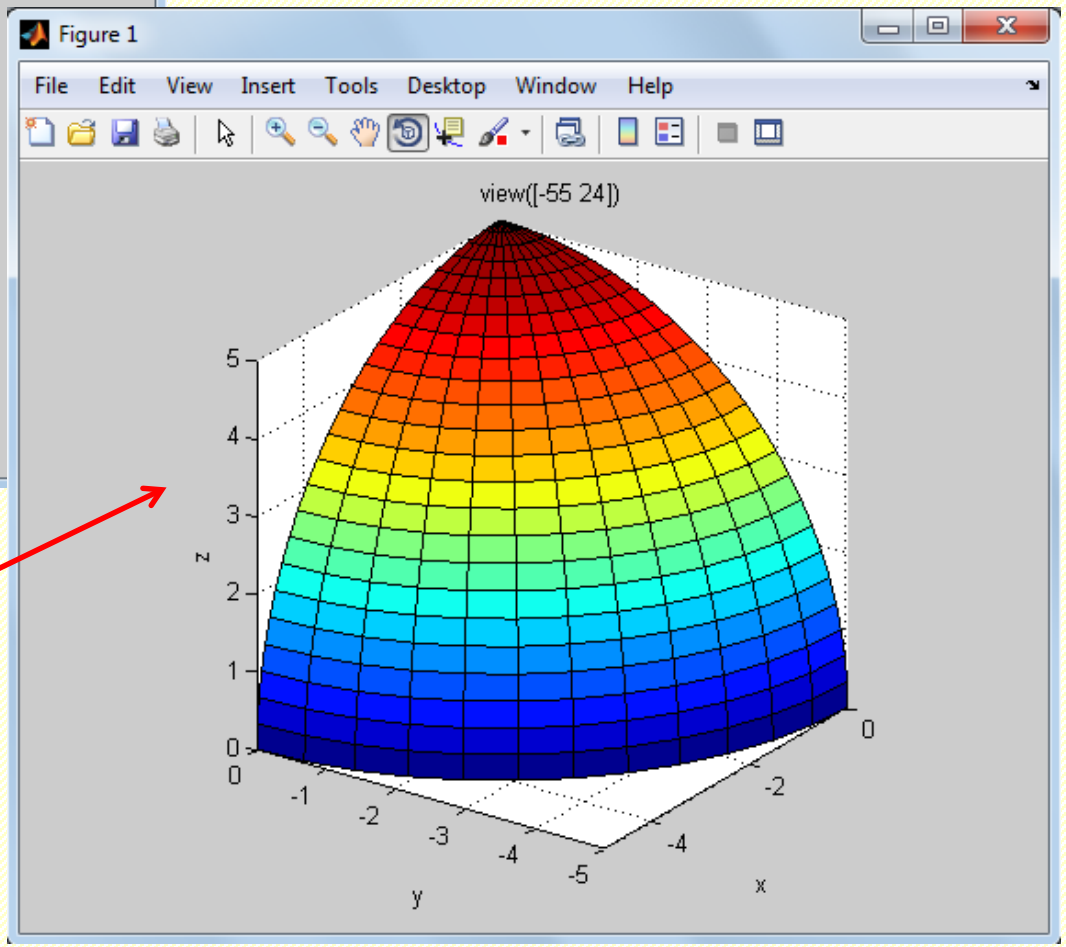
```
%%
clear; close all; clc;
figure; hold on; axis equal; grid on;
quality=48;
[x,y,z]=sphere(quality);
S3=surf(0.5*x,0.5*y,0.5*z-0.2,'FaceColor','y','EdgeColor','none');
z(find(z>0))=NaN;
S1=surf(2*x,2*y,2*z);
S2=surf(x,y-2.5,z,'FaceColor','g');
view(3)
camlight right;
light
lighting phong;

colormap jet;
xlabel('x');
ylabel('y');
zlabel('z');
rotate3d on;
axis tight;
```

Combination of "processed" Spheres





>> view([-90 90])



>> view([-55 24])

MATLAB:

Understanding "meshgrid"

MATLAB: MESHGRID

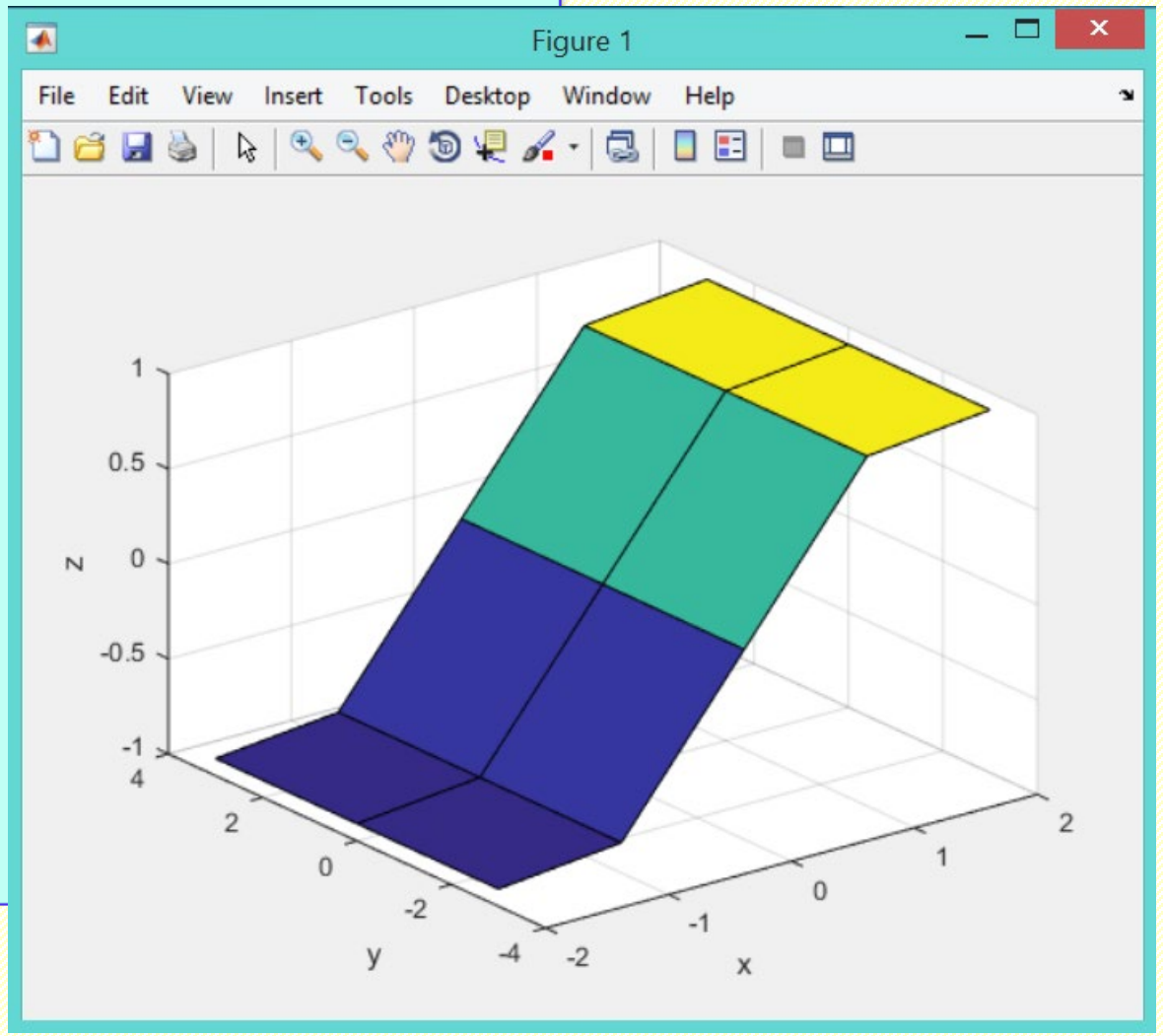
Understanding inputs and outputs of meshgrid

```
>> [X,Y] = meshgrid(-2:1:2, -3:3:3)

X =

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


Y =

    -3    -3    -3    -3    -3
     0     0     0     0     0
     3     3     3     3     3
```

```
>> size(X)

ans =

     3     5

>> size(Y)

ans =

     3     5
```

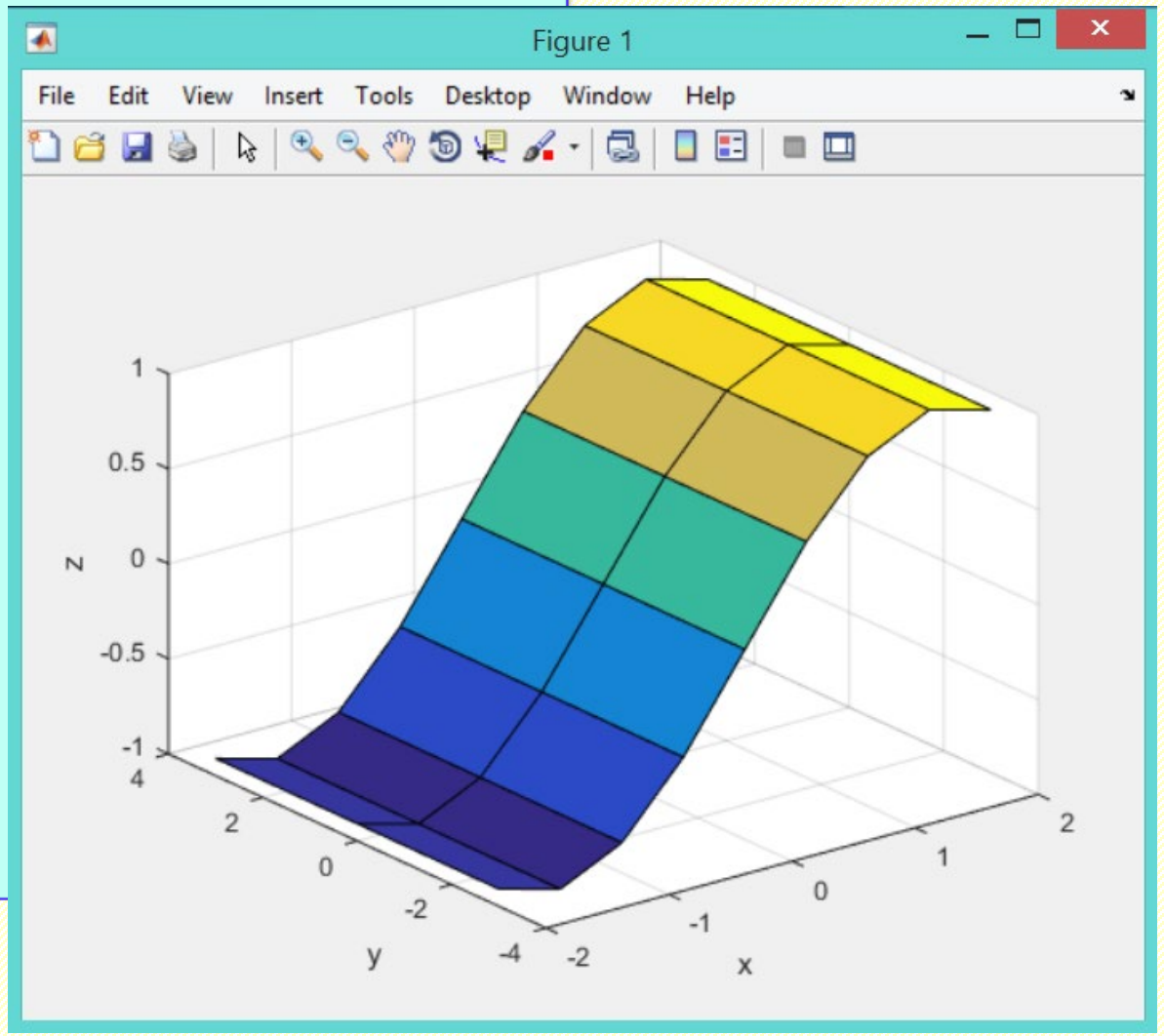
MATLAB: MESHGRID: DISCUSSION

```
[X,Y] = meshgrid(-2: 1 : 2,     -3:3:3);

Z = sin(X);


surf(X,Y,Z);


xlabel('x');
ylabel('y');
zlabel('z');
```



MATLAB: MESHGRID: DISCUSSION

```
[X,Y] = meshgrid(-2: 0.5 : 2,   -3:3:3);

Z = sin(X);


surf(X,Y,Z);


xlabel('x');
ylabel('y');
zlabel('z');
```

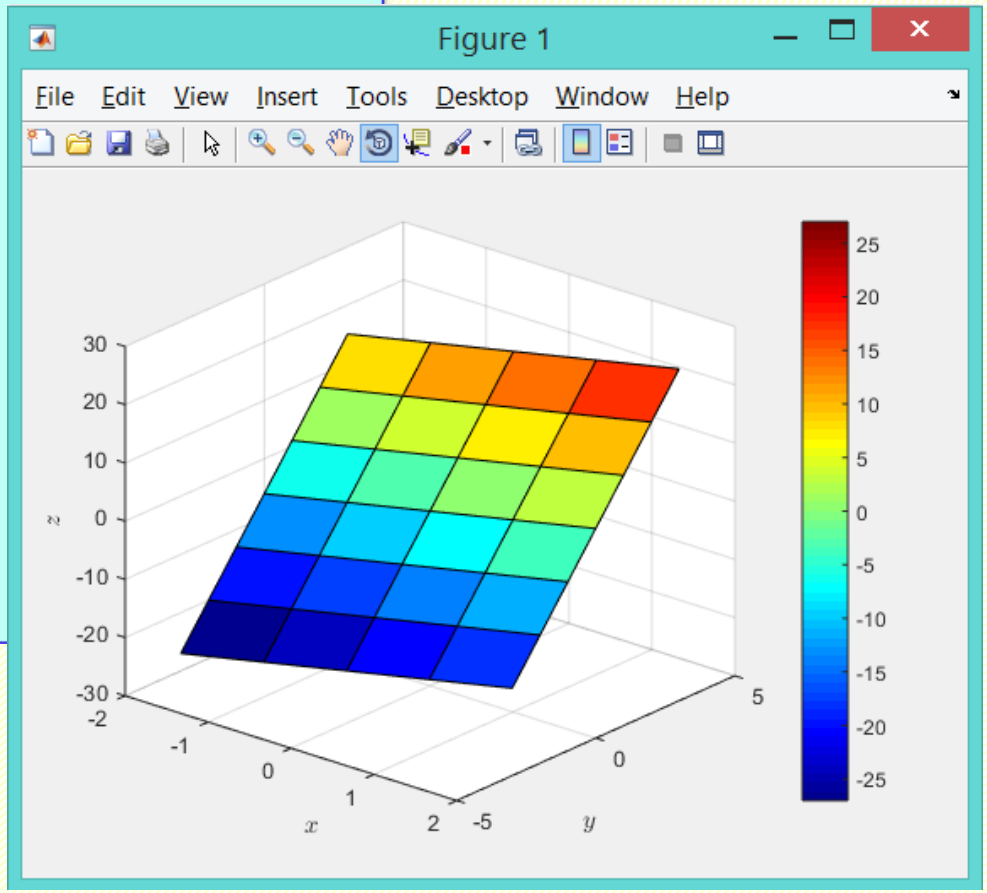


MATLAB:

Detailed Simple Example of Plotting 3D Surface Plot with Countour Level Lines

MATLAB: Surface Plot of a Plane

```
%% --- Basic "surf" plot ---
figure; hold on; grid on; colorbar;

[X,Y] = meshgrid(-2: 1 : 2,   -3:1:3);
Z=3*X+7*Y;
surf(X,Y,Z); view(40,25);

xl=xlabel('$x$');
yl=ylabel('$y$');
zl=zlabel('$z$');
set([xl,yl,zl],…
   'Interpreter','LaTeX');
colormap jet;
rotate3d on
```



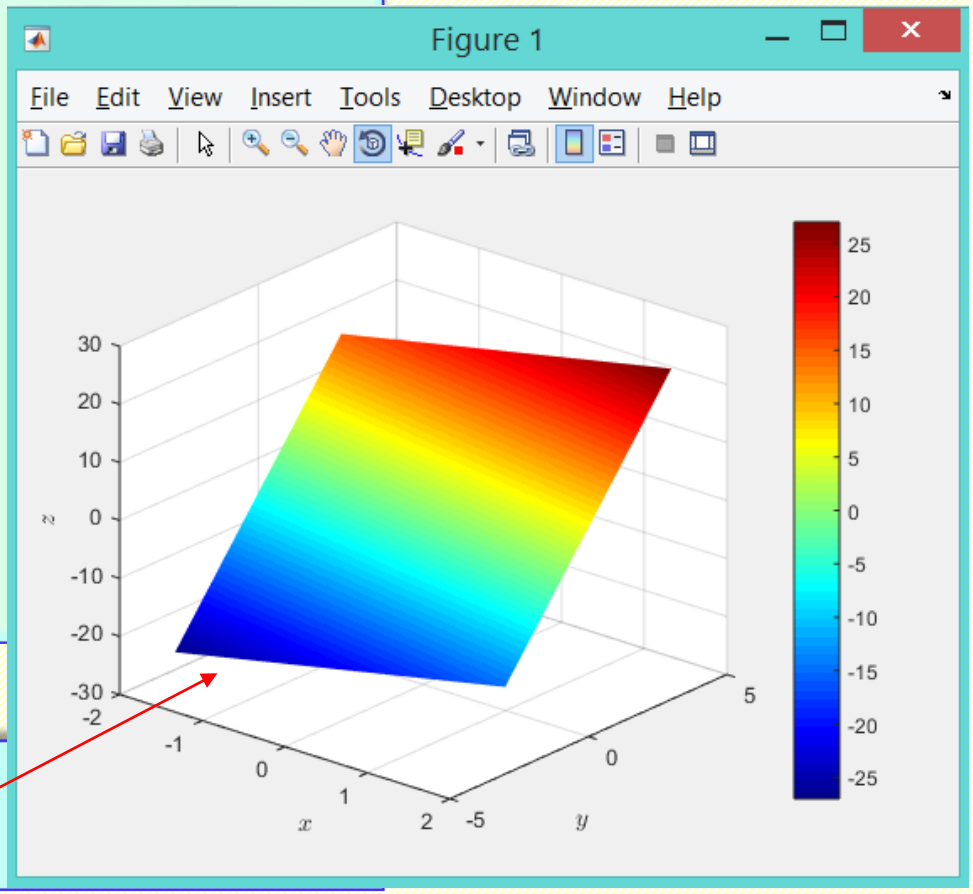
MATLAB: Surface Plot of a Plane

```
%% --- Basic "surf" plot ---
figure; hold on; grid on; colorbar;

[X,Y] = meshgrid(-2: 1 : 2,   -3:1:3);
Z=3*X+7*Y;
surf(X,Y,Z); view(40,25);

xl=xlabel('$x$');
yl=ylabel('$y$');
zl=zlabel('$z$');
set([xl,yl,zl],…
   'Interpreter','LaTeX');
colormap jet;
rotate3d on
```

Let us also add these commands:

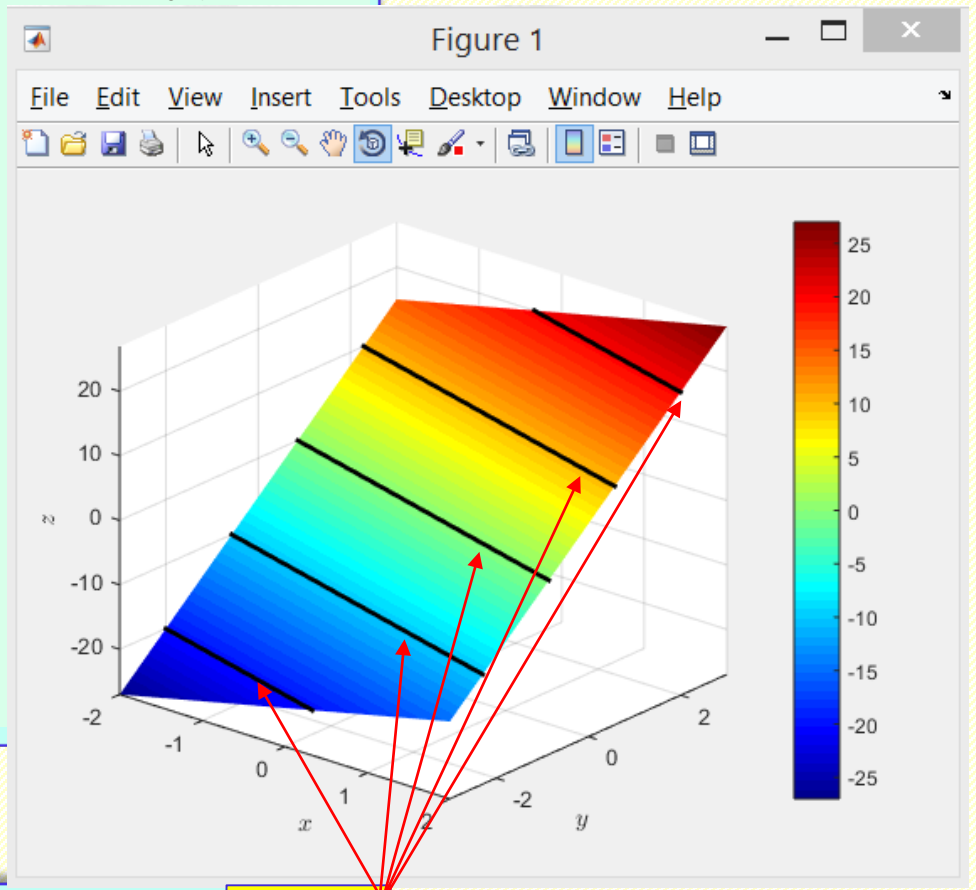
```
lighting phong;
shading interp;
```

result:



MATLAB: Surface Plot of a Plane

```
%% --- Basic "surf" plot ---
figure; hold on; grid on; colorbar;

[X,Y] = meshgrid(-2: 1 : 2,   -3:1:3);
Z=3*X+7*Y;
surf(X,Y,Z); view(40,25);

xl=xlabel('$x$');
yl=ylabel('$y$');
zl=zlabel('$z$');
set([xl,yl,zl],…
   'Interpreter','LaTeX');
colormap jet;
rotate3d on
lighting phong;
shading interp;
```

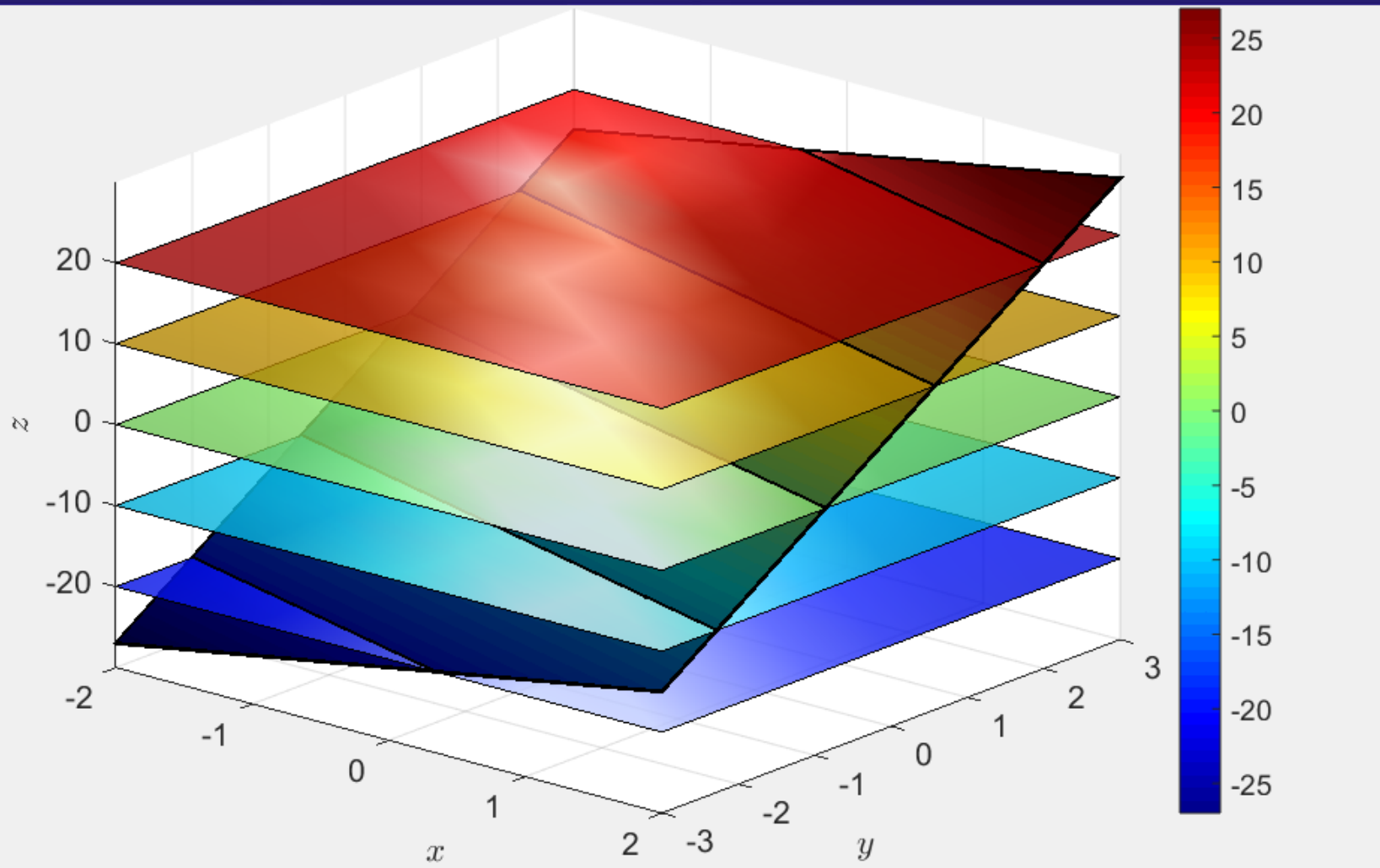


Let us also add this command:

```
contour3(X,Y,Z, [-30:10:30],'k');
```

result:

Explaining Contour Lines by adding Horizontal Level Planes

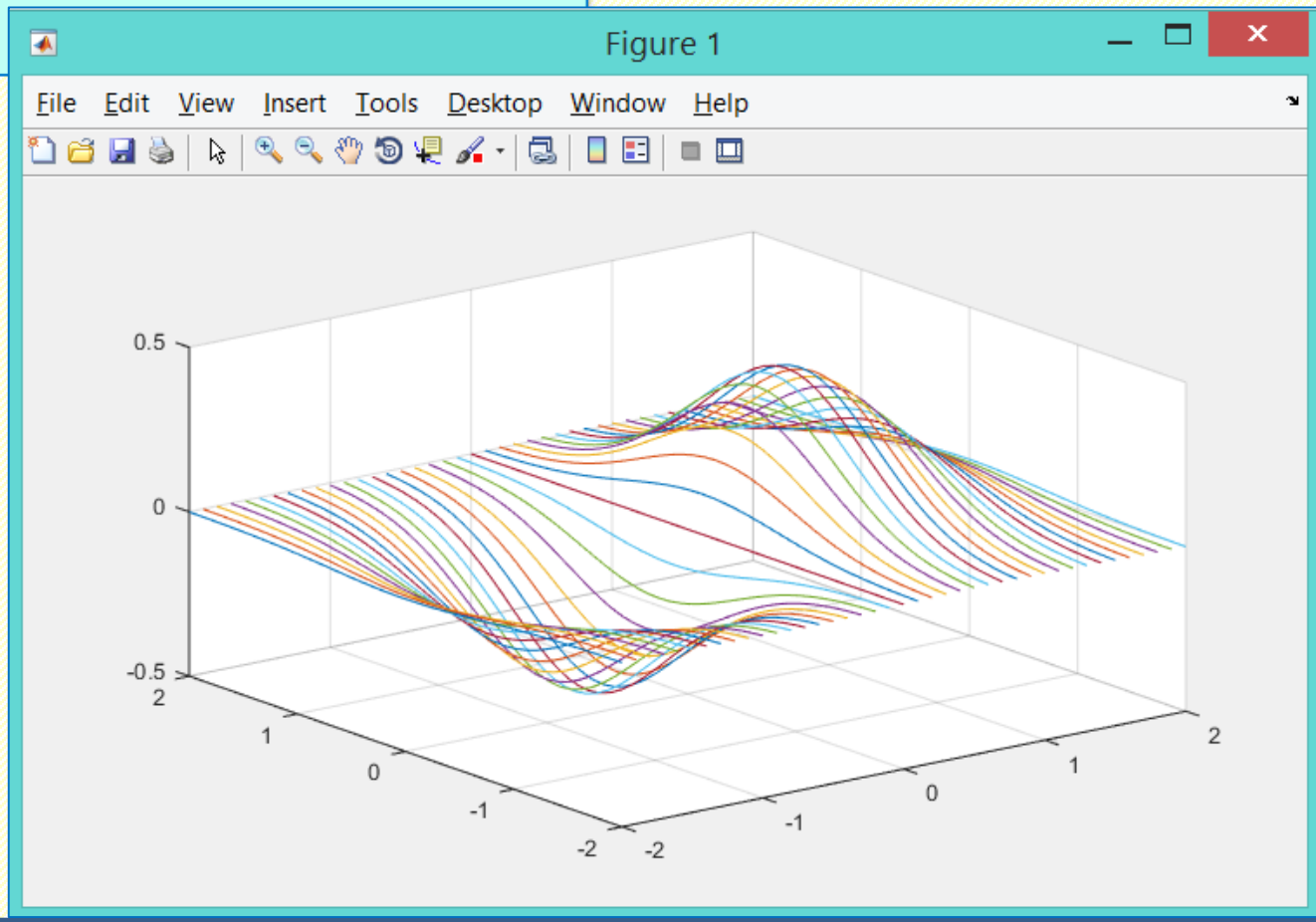


MATLAB:

Other Examples

MATLAB: MORE SURFACES

```
[X,Y] = meshgrid([-2:0.1:2]);
Z = X.*exp(-X.^2-Y.^2);
plot3(X,Y,Z)
grid on
```

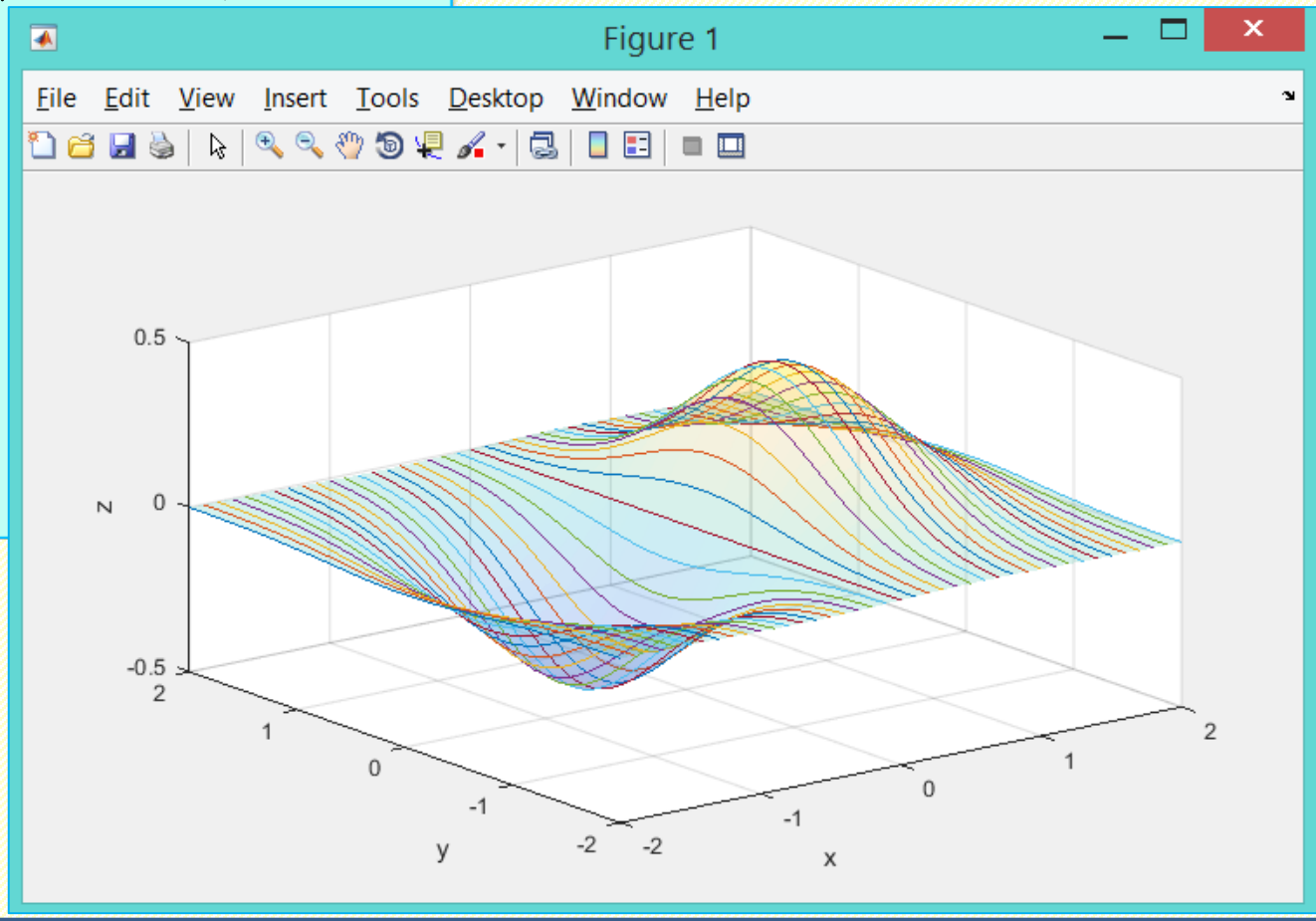


Surface shown as a set of just 3D lines

MATLAB: MORE SURFACES

```
[X,Y] = meshgrid([-2:0.1:2]);
Z = X.*exp(-X.^2-Y.^2);
h=surf(X,Y,Z);
set(h,'EdgeColor','none');
alpha(0.2)

hold on;
plot3(X,Y,Z);
grid on
xlabel('x');
ylabel('y');
zlabel('z');
```

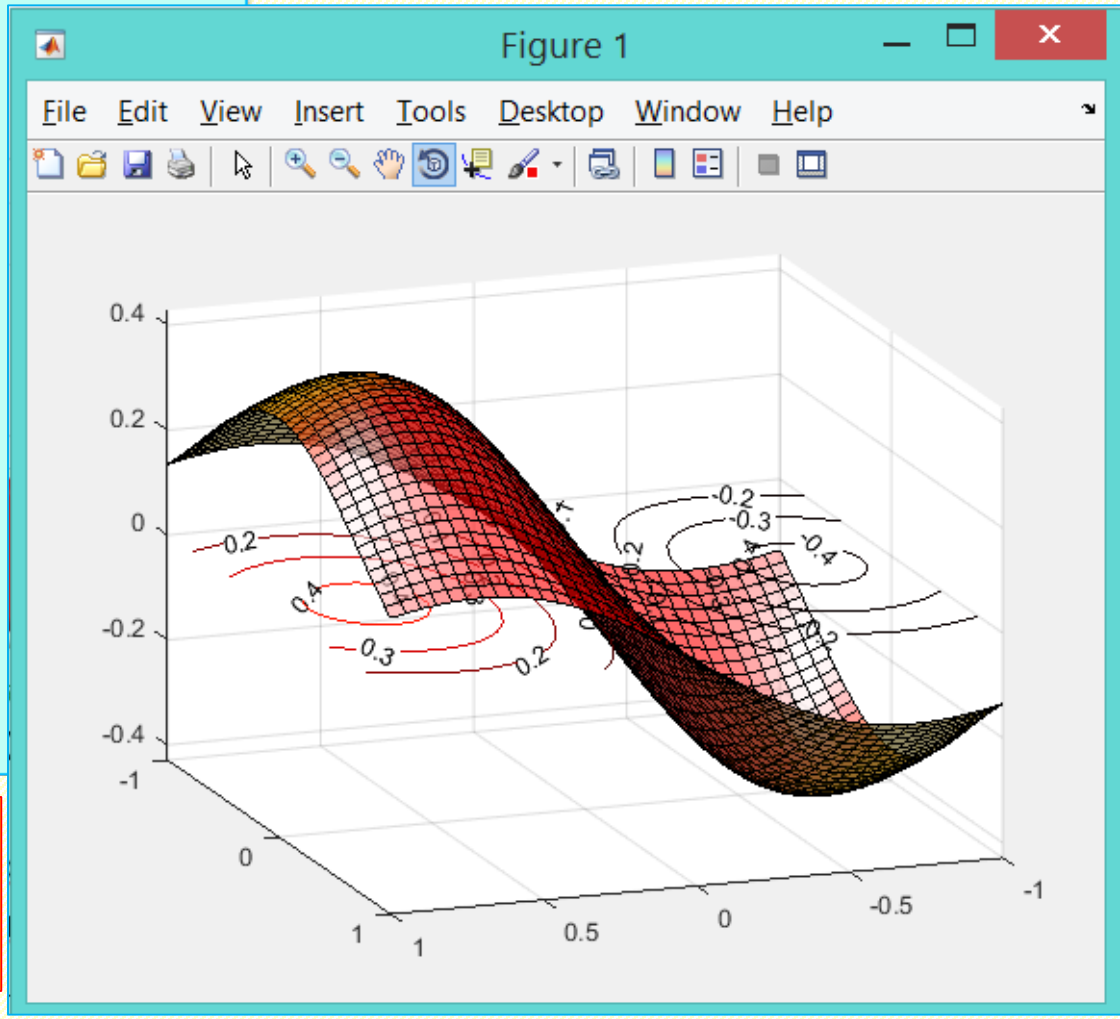
Surface shown as a Combination of TWO components: (1) almost-transparent surface AND (2) set of 3D lines [from previous example]



MATLAB: MORE SURFACES

```
%%
[X,Y] = meshgrid(-1:.05:1);
Z = X.*exp(-X.^2-Y.^2);
colormap hot
surfl(X,Y,Z);
view(-200, 20)
alpha(0.6)
camlight(10,30)
light;

hold on
[c,h]=contour(X,Y,Z);
clabel(c,h);
rotate3d on
```

Recommendation: plot this figure and change view angles, using mouse



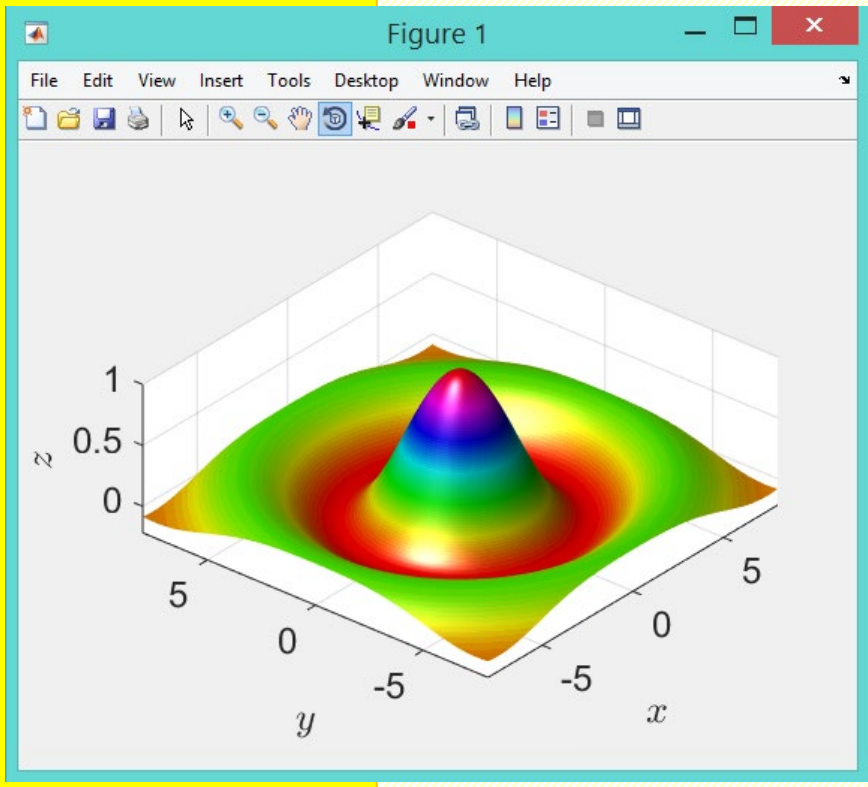
Features: (1) semi-transparent surface AND (2) set of contour lines at z=0 with numeric labels.

MATLAB:

Other Examples to be Discussed during Tutorial

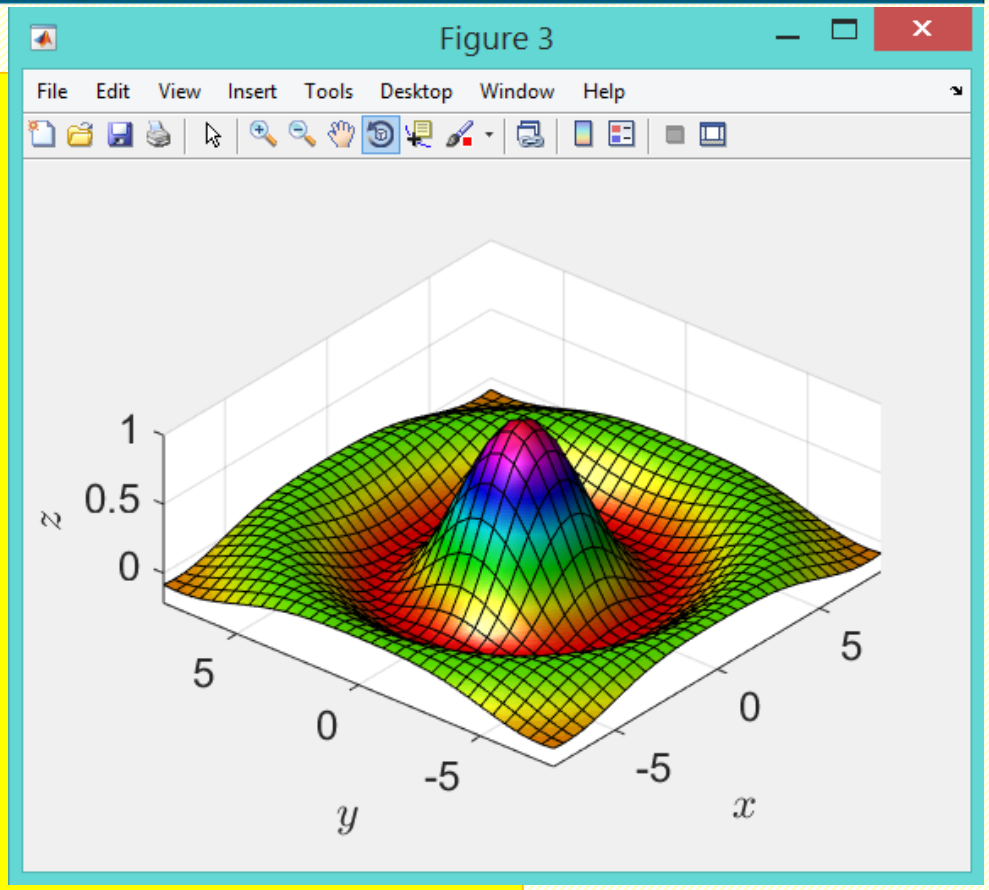
MANIPULATIONS WITH SURFACES: Class Exercises (lighting, rotate3d)

```
clear; close a[X,Y] = meshgrid(-8:.1:8);
R = sqrt(X.^2 + Y.^2) + eps;
Z = sin(R)./R;

figure
colormap hsv
surf(X,Y,Z,'FaceColor','interp',...
    'EdgeColor','none',...
    'FaceLighting','gouraud')
daspect([5 5 1])
axis tight
view(-50,30)
camlight left
xl=xlabel('$x$');
yl=ylabel('$y$');
zl=zlabel('$z$');
set([xl,yl,zl],'Interpreter','LaTeX');
set(gca,'FontSize',18);
rotate3d on
```



MANIPULATIONS WITH SURFACES: Class Exercises (lighting, rotate3d)

```
figure
colormap hsv
surf(X,Y,Z,…
    'FaceColor','interp',...
    'FaceLighting','gouraud')
%    'EdgeColor','none',...

daspect([5 5 1])
axis tight
view(-50,30)
camlight left
xl=xlabel('$x$');
yl=ylabel('$y$');
zl=zlabel('$z$');
set([xl,yl,zl],'Interpreter','LaTeX');
set(gca,'FontSize',18);
rotate3d on
```

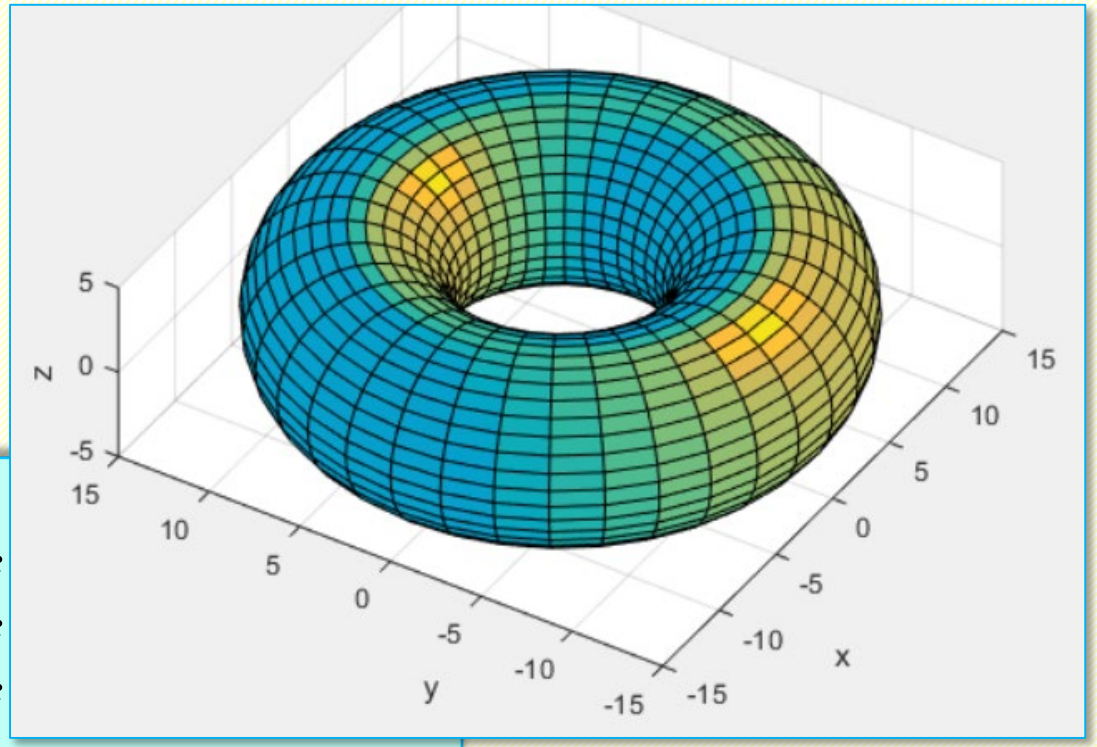


MATLAB:

Plotting Non-Standard Surfaces (Advanced Examples)

MATLAB: TORUS

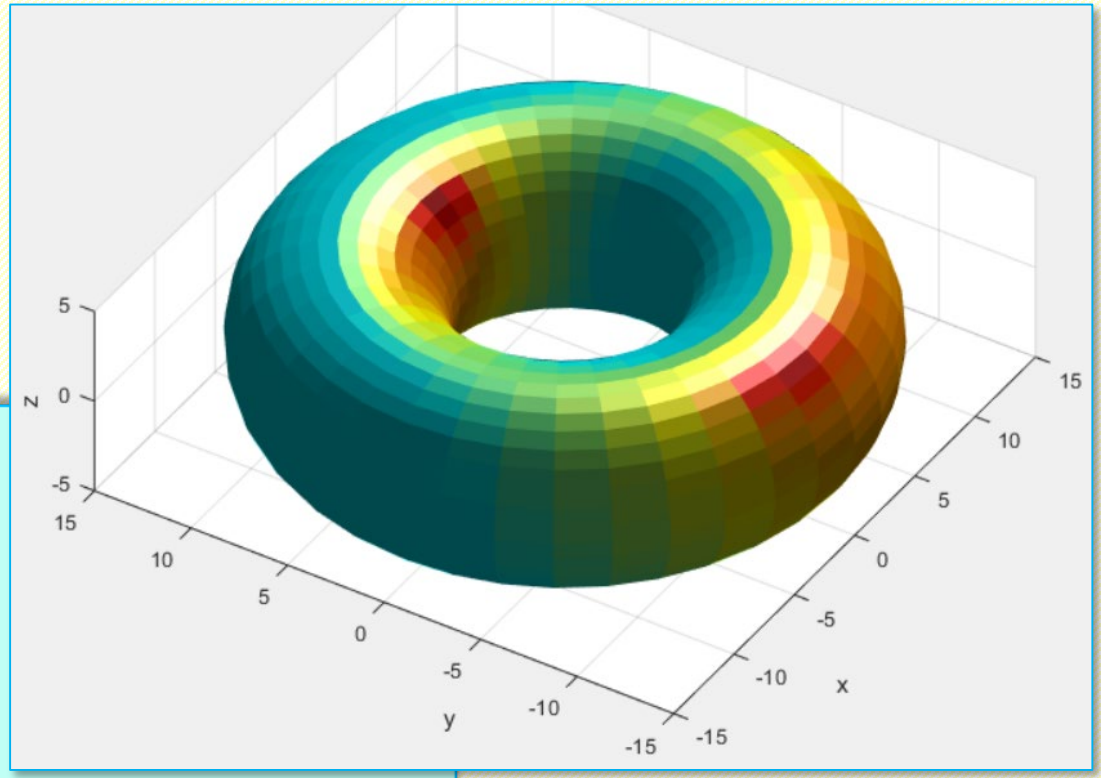
PRODUCING A TORUS !!!




```
a=5; c=10;
[u,v]=meshgrid(0:10:360);
x=(c+a*cosd(v)).*cosd(u);
y=(c+a*cosd(v)).*sind(u);
z=a*sind(v);
surfl(x,y,z)
axis equal;
xlabel('x'); ylabel('y'); zlabel('z');
rotate3d on
view([-58 38])
```

MATLAB: TORUS & SHADING

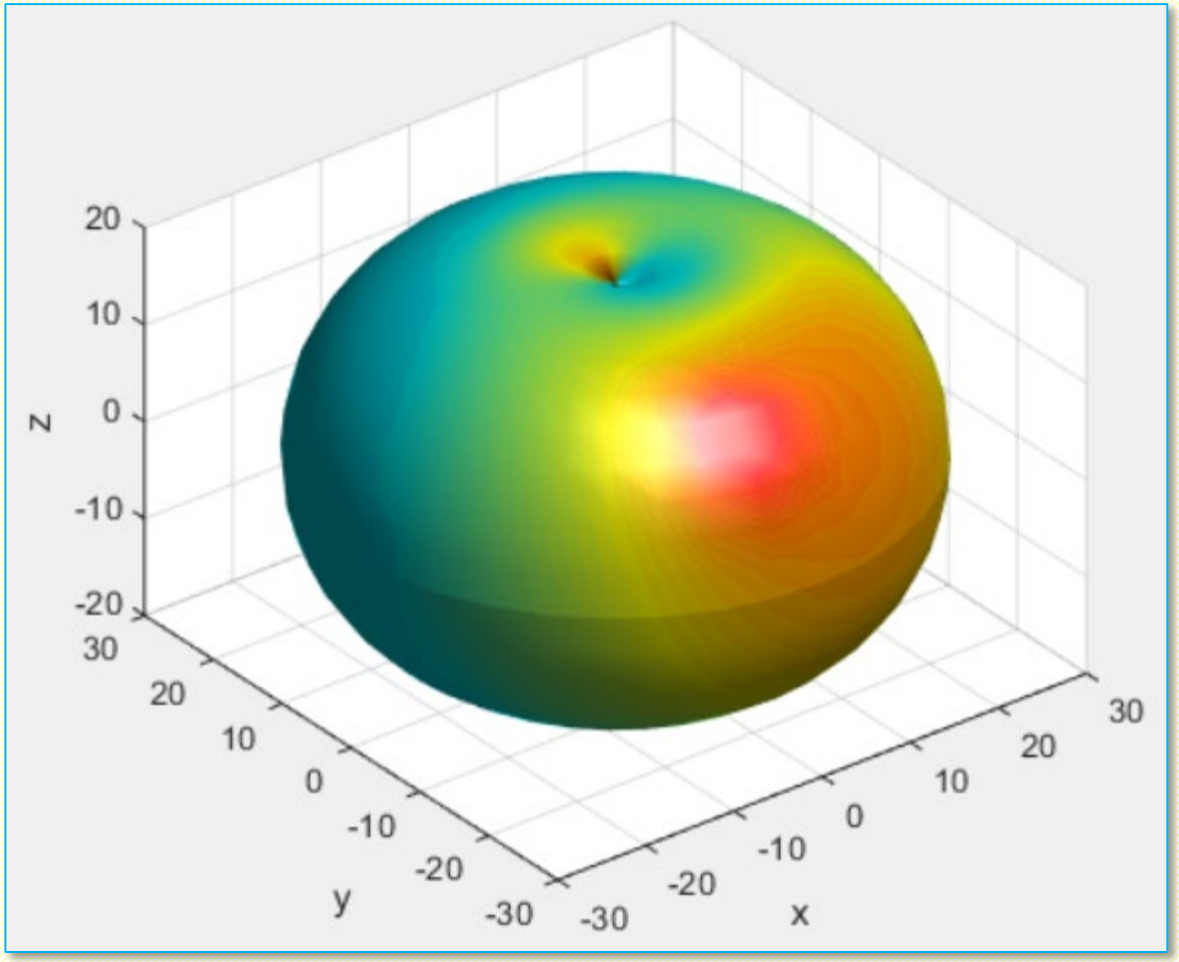
PRODUCING A TORUS !!!




```
a=5; c=10;
[u,v]=meshgrid(0:10:360);
x=(c+a*cosd(v)).*cosd(u);
y=(c+a*cosd(v)).*sind(u);
z=a*sind(v);
surfl(x,y,z)
axis equal;
xlabel('x'); ylabel('y'); zlabel('z');
rotate3d on
view([-58 38])
shading flat; light
```

MATLAB: Producing an Apple!

```
%% PMT's APPLE
close all; clc; clear
a=20; c=10;
[u,v]=meshgrid(0:10:360);
x=(c+a*cosd(v)).*cosd(u);
y=(c+a*cosd(v)).*sind(u);
z=a*sind(v);
surfl(x,y,z)
axis equal;
xlabel('x');
ylabel('y');
zlabel('z');
rotate3d on
view([-38 30])
camlight right
shading interp;
lighting phong
colormap jet
```

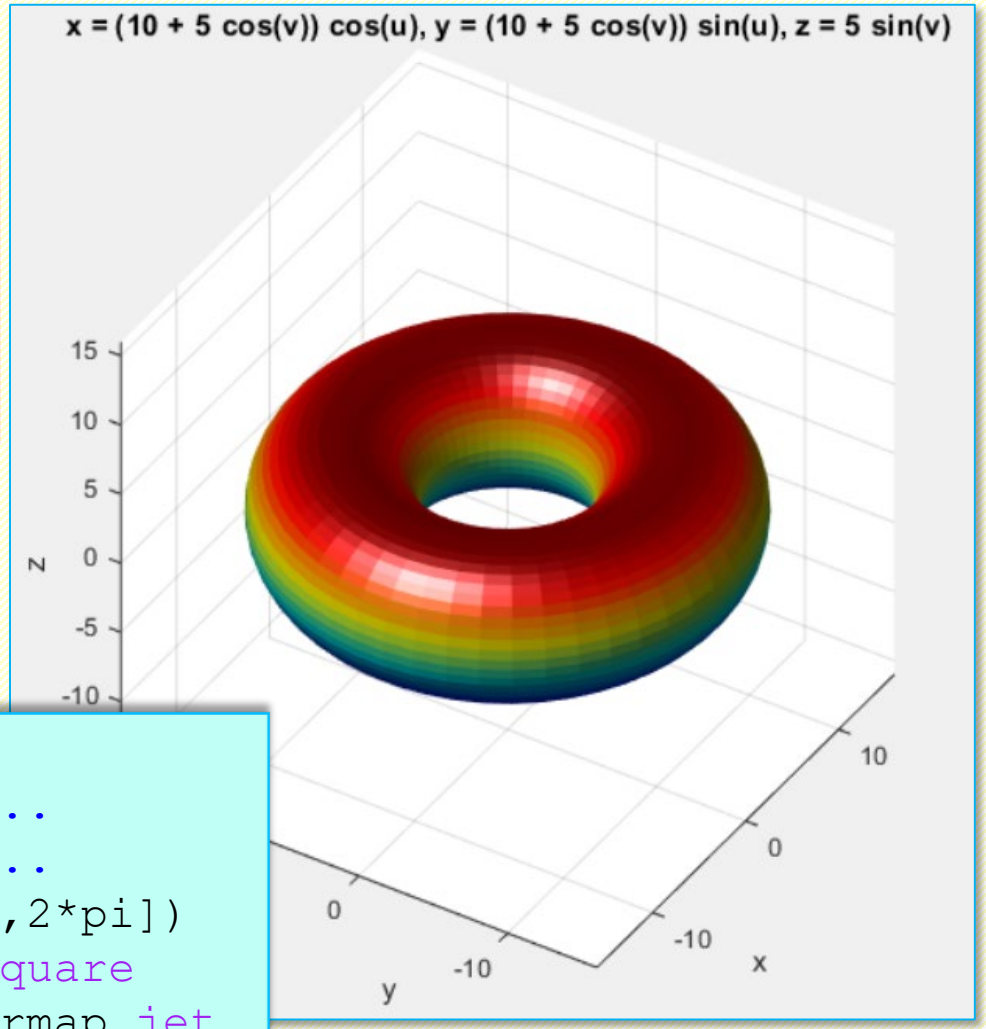


MATLAB: Producing an Apple!

```
%% PMT's APPLES
close all; clc; clear
???????
??????????
????????????????????
???????????
????????????
????????????????
???????
??????????
% do not report an error
% this is just a joke!
????????????
????????????????
???????
??????????
????????????????????
???????????
????????????
????????????????
!!!!!!!!!!!!!!!!!!!!
!!!!!!!
```

MATLAB: TORUS & SYMBOLIC

PRODUCING A TORUS
Using
SYMBOLIC TOOLBOX !!!

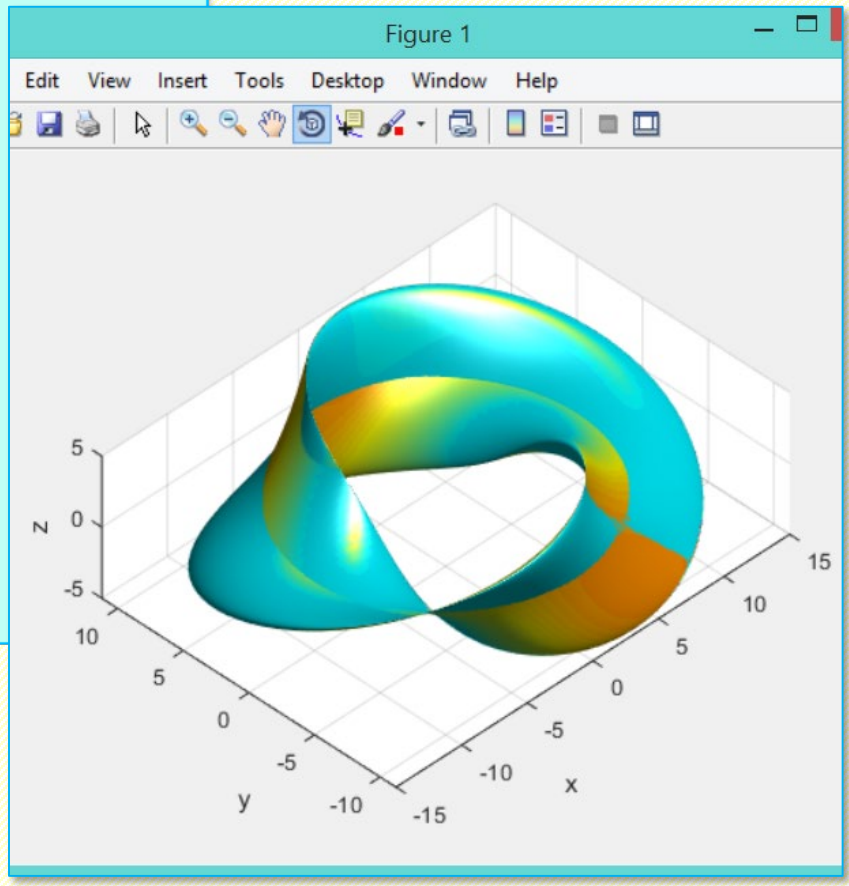



```
%%
ezsurf('(10 + 5*cos(v))*cos(u)',...
       '(10 + 5*cos(v))*sin(u)',...
            '5*sin(v)',[0,2*pi,0,2*pi])
axis([-1 1 -1 1 -1 1]*16); axis square
view([-58 38]); rotate3d on; colormap jet
h=findobj('type','surf');
set(h,'EdgeColor','none'); camlight left
```

MATLAB: MÖBIUS STRIP

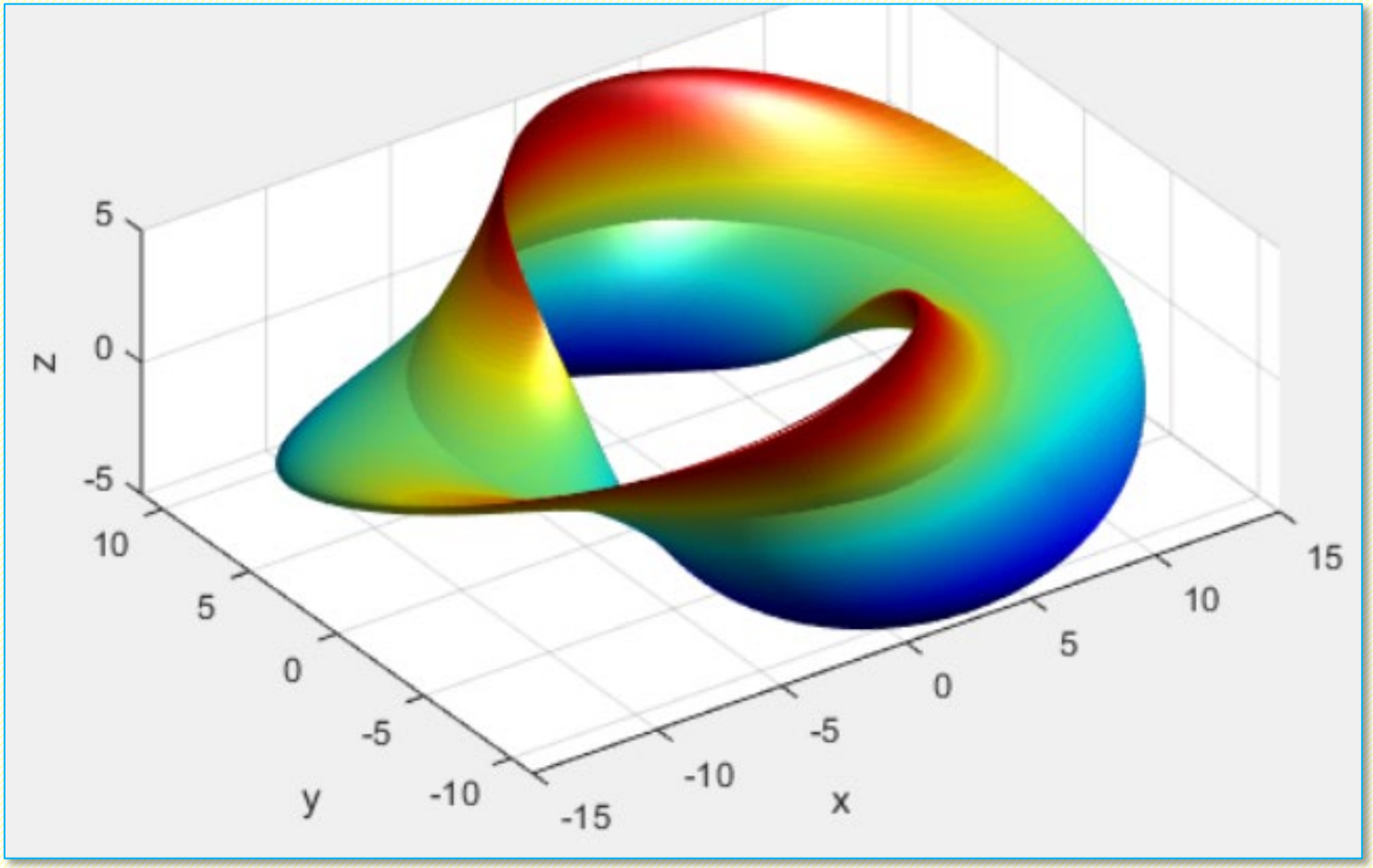
PMT IS PRODUCING A MÖBIUS STRIP !!!

```
%% Programmed by Prof P.M.Trivailo
a=5; c=10;
[u,v]=meshgrid([0:0.01:1]*2*pi);
x=cos(u).*(c + a*sin(v).*cos(u)) - ...
                    sin(2*v).*sin(u)/2;
y=sin(u).*(c + a*sin(v).*cos(u)) - ...
                    sin(2*v).*sin(u)/2;
z=a*sin(u).*sin(v) + cos(u).*sin(2*v)/2;
surfl(x,y,z)
axis equal; xlabel('x'); ylabel('y');
zlabel('z');
colormap jet; rotate3d on
view([-45 40])
shading interp; camlight right;
lighting phong;
```



MATLAB: MÖBIUS STRIP

THERE ARE INTERESTING EXPERIMENTS WITH MÖBIUS STRIPS, using scissors !!!



FEM:

For more materials (with emphasis on analytical aspects), please, refer to the Separate File

https://drive.google.com/open?id=1-6Sh6mctXu4kYy5Sp8oOBTo-hd2u1mdU

This Chapter on FEM is extracted from the following Book:

REFERENCE: **Trivailo P.M.** (2008), Vibrations: Theory & Aerospace Applications, Vol.1&2, - (The textbook for senior undergraduate and graduate aerospace students). - Melbourne: RMIT Publisher - 2008. - 247pp +348pp=595 pp., 355 ill, 4 software programs.

Study Case:

Excitation of Axial Vibrations of Rods

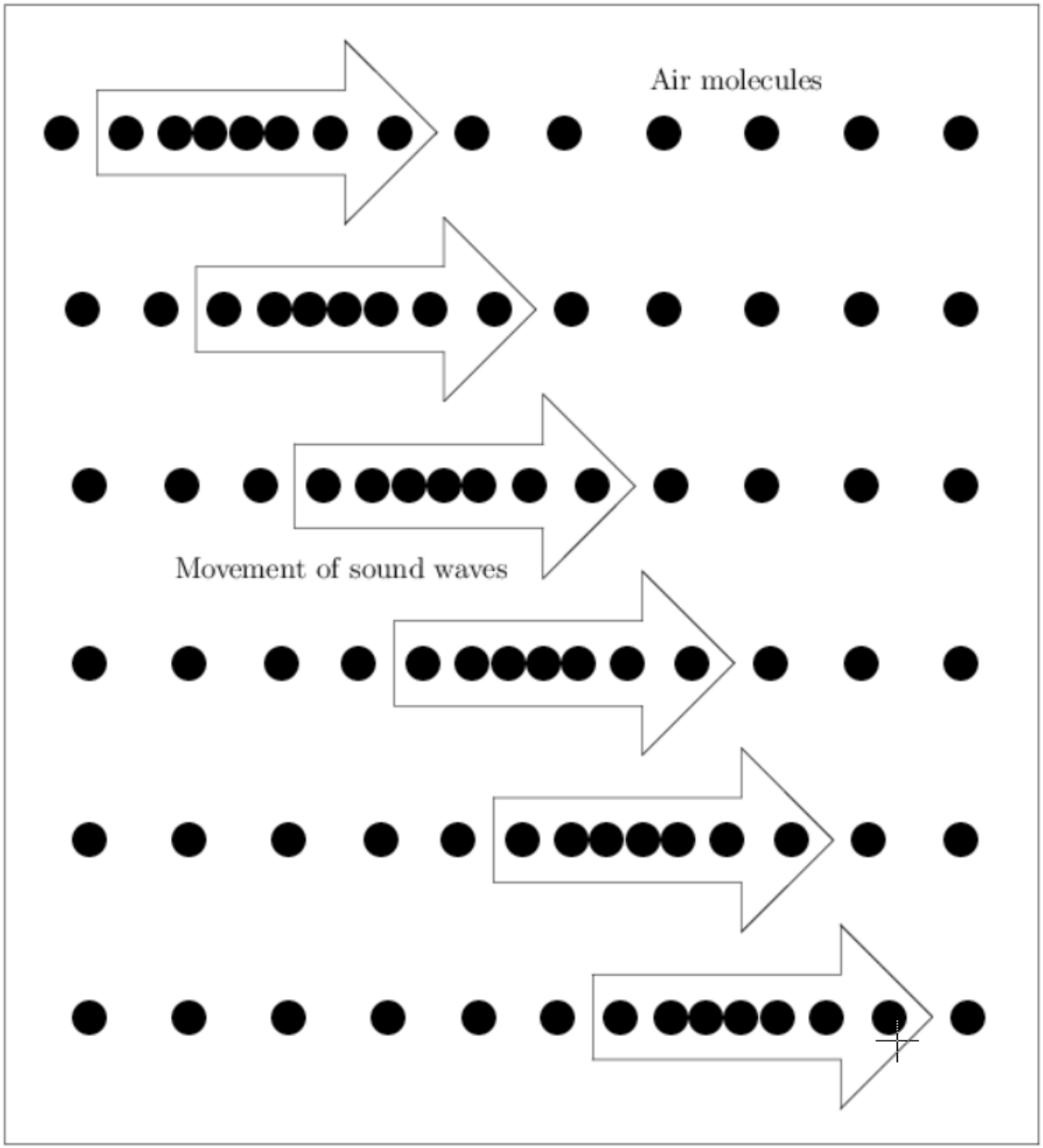


Figure 8.3: Analogy with a sound wave.

- It is easy to send longitudinal waves along a coiled spring: pull end of spring and out.

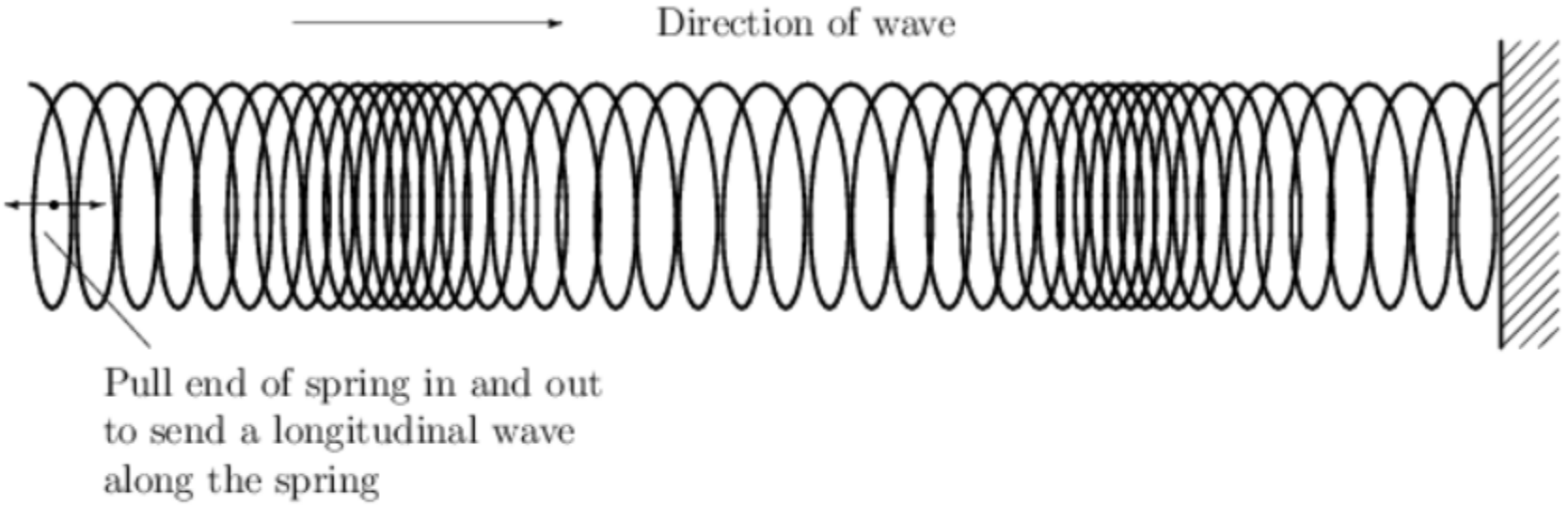


Figure 8.4: Analogy with transverse wave in a spring.

Analytical Solution for Free-Free Rod:

Used for future comparisons with FEM solutions

8.7 TAKING INTO ACCOUNT BOUNDARY CONDITIONS

8.7.1 Example: Longitudinal Vibration of a Free-Free Uniform Rod

Solve the eigenvalue problem associated with a uniform rod, vibrating longitudinally, with both ends free (*see* Fig. 8.8).

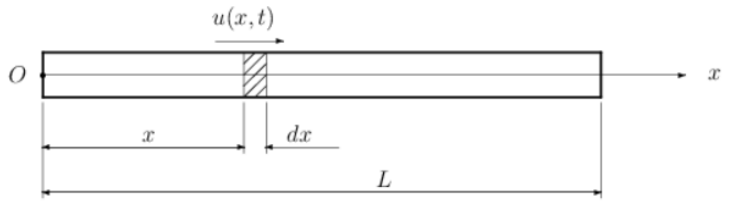


Figure 8.8: A free-free uniform rod in longitudinal vibration.

Solution

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2},$$

$$u(x,t) = \left(A \sin\frac{\omega}{c}x + B\cos\frac{\omega}{c}x\right) \times (C\sin\omega t + D\cos\omega t).$$

The arbitrary constants A, B, C, D depend on the *boundary conditions* and the *initial conditions*.

Since the bar has free ends, *the axial force*, which is proportional to dU/dx, *must be zero at each extremity*. Thus the boundary conditions for this problem may be written as

$$EA\left.\frac{\partial u(x,t)}{\partial x}\right|_{x=0} = 0, \qquad EA\left.\frac{\partial u(x,t)}{\partial x}\right|_{x=L} = 0.$$

The first boundary condition will require that $A = 0$, so

$$u(x,t) = B\cos\frac{\omega}{c}x\,(C\sin\omega t + D\cos\omega t).$$

The second boundary condition then leads to the *characteristic equation*:

$$\sin\frac{\omega L}{c} = 0, \quad \text{or} \quad \frac{\omega_r L}{c} = r\pi, \quad r = 1,2,3,\ldots,$$

to which corresponds the infinite set of *eigenfunctions*:

$$U_r(x) = B_r\cos\frac{r\pi x}{L}.$$

The first natural modes are plotted in Figure 8.9, where the modes have been normalized by letting $B_r = 1$. We note that the first mode has one node, the second has two nodes and the third has three nodes. In general the r-th mode has r nodes $(r = 1, 2, \ldots)$.

The system natural frequencies are:

$$\omega_r = \frac{r\pi c}{L} = r\pi\sqrt{\frac{E}{\rho L^2}}, \qquad r = 1,2,3,\ldots.$$

FREE-FREE ROD:

In the more general case of free vibration initiated in any manner, the solution will contain many of the normal modes:

$$u(x,t) = \sum_{r=1}^{\infty}\cos\frac{r\pi x}{L}\,(C_r\sin\omega_r t + D_r\cos\omega_r t)$$

$$\omega_r = \frac{r\pi c}{L}, \quad r = 1,2,3,\ldots.$$

The arbitrary constants C_r, D_r depend on the *initial conditions*.

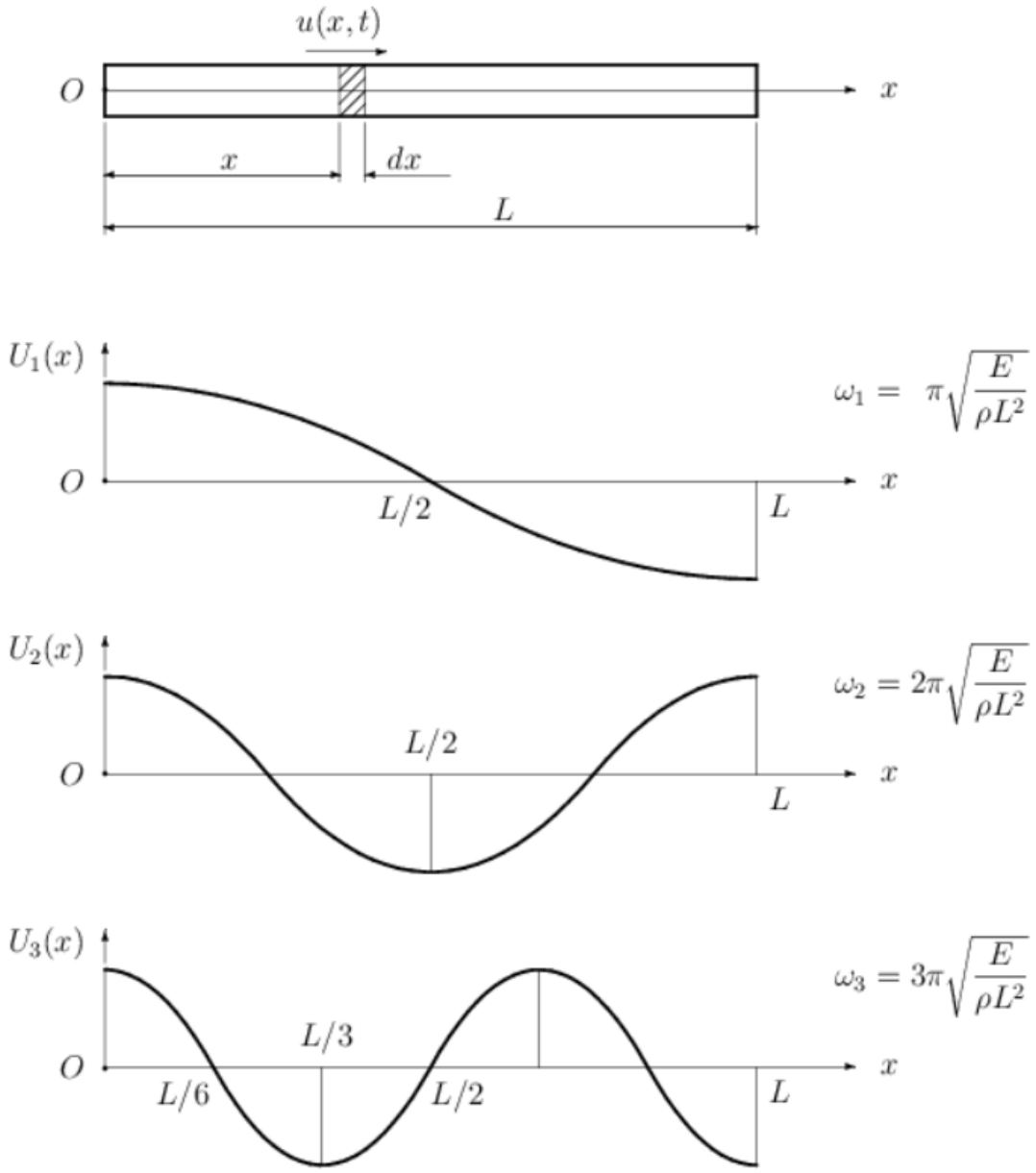
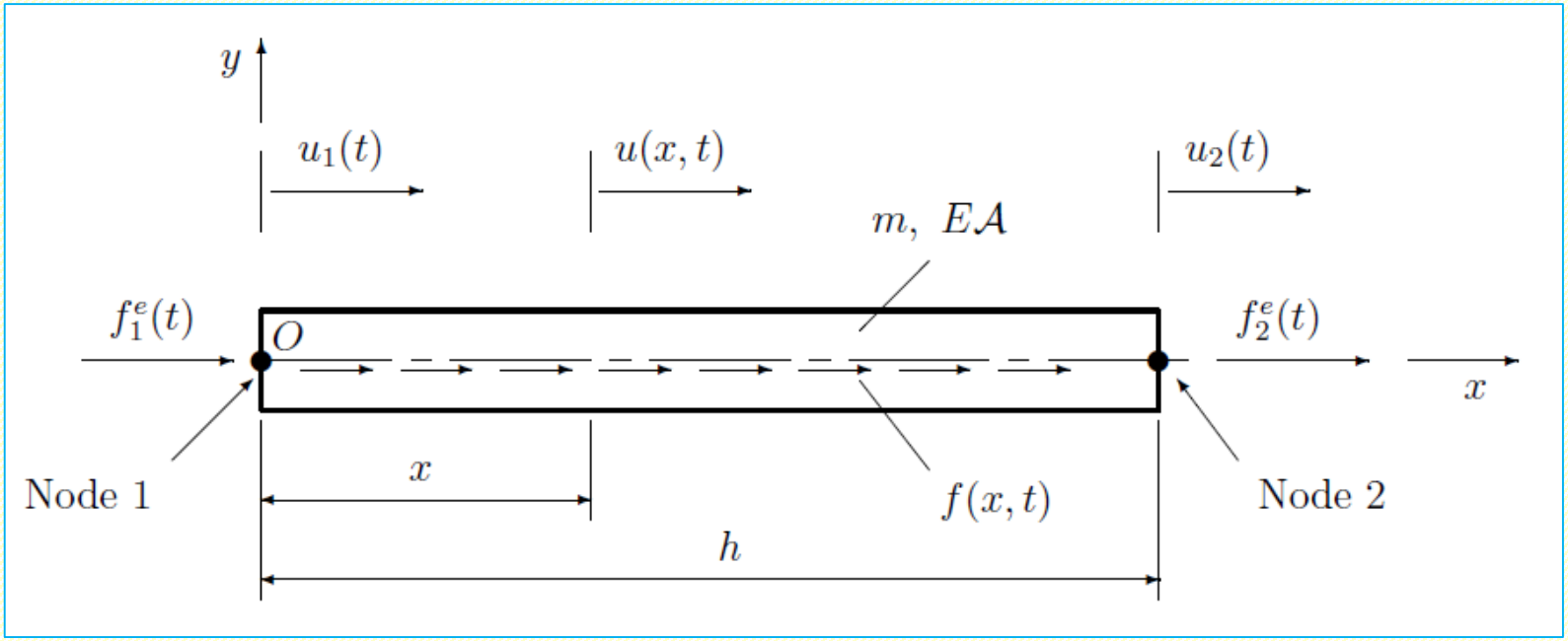


Figure 8.9: The first three natural modes of longitudinal vibration for a free-free bar.

FEM: Truss Element:

Shape Functions;
$[m^e]$ & $[k^e]$ matrices;
interpolation of axial displacements (example)

FEM: TRUSS ELEMENT



$$u(x,t) = \left(1 - \frac{x}{h}\right) u_1(t) + \frac{x}{h}\, u_2(t).$$

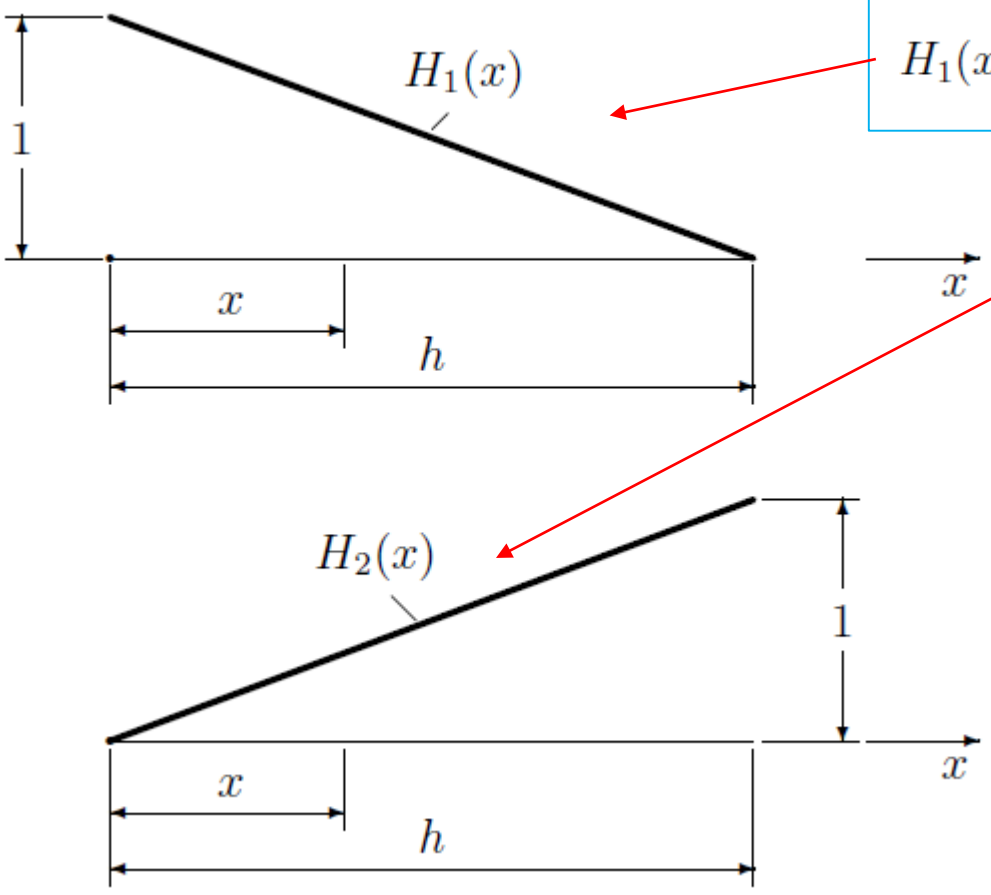
$$u(x,t) = H_1(x)\, u_1(t) + H_2(x)\, u_2(t), \qquad \text{where}$$

$$H_1(x) = 1 - \frac{x}{h}; \qquad H_2(x) = \frac{x}{h}$$

FEM: SHAPE FUNCTIONS for TRUSS ELEMENT

$$u(x,t) = H_1(x)\,u_1(t) + H_2(x)\,u_2(t), \qquad \text{where}$$

$$H_1(x) = 1 - \frac{x}{h}; \qquad H_2(x) = \frac{x}{h}$$



$$u(x,t) = \left(1 - \frac{x}{h}\right) u_1(t) + \frac{x}{h}\,u_2(t).$$

FEM: STIFFNESS & MASS MATRICES

Stiffness Matrix for a 1D Truss FE:
(Global and FE Local Coordinate Systems)

$$[k^e] = \frac{E\mathcal{A}}{h}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Consistent Mass Matrix for a 1D Truss FE:
(Global and FE Local Coordinate Systems)

$$[m^e] = \frac{mh}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

FEM: NODAL FORCES

Nodal Forces for a 1D Truss FE:

$$f_1^e(t) = \int_0^h f(x,t)\left(1 - \frac{x}{h}\right)\,dx = \int_0^h f(x,t)\,H_1(x)\,dx$$

$$f_2^e(t) = \int_0^h f(x,t)\left(\frac{x}{h}\right)\,dx = \int_0^h f(x,t)\,H_2(x)\,dx$$

FEM: Truss Element:

Static case of the Rod modelled with only 1 Finite Element:

ANALYSIS of DISPLACEMENTS

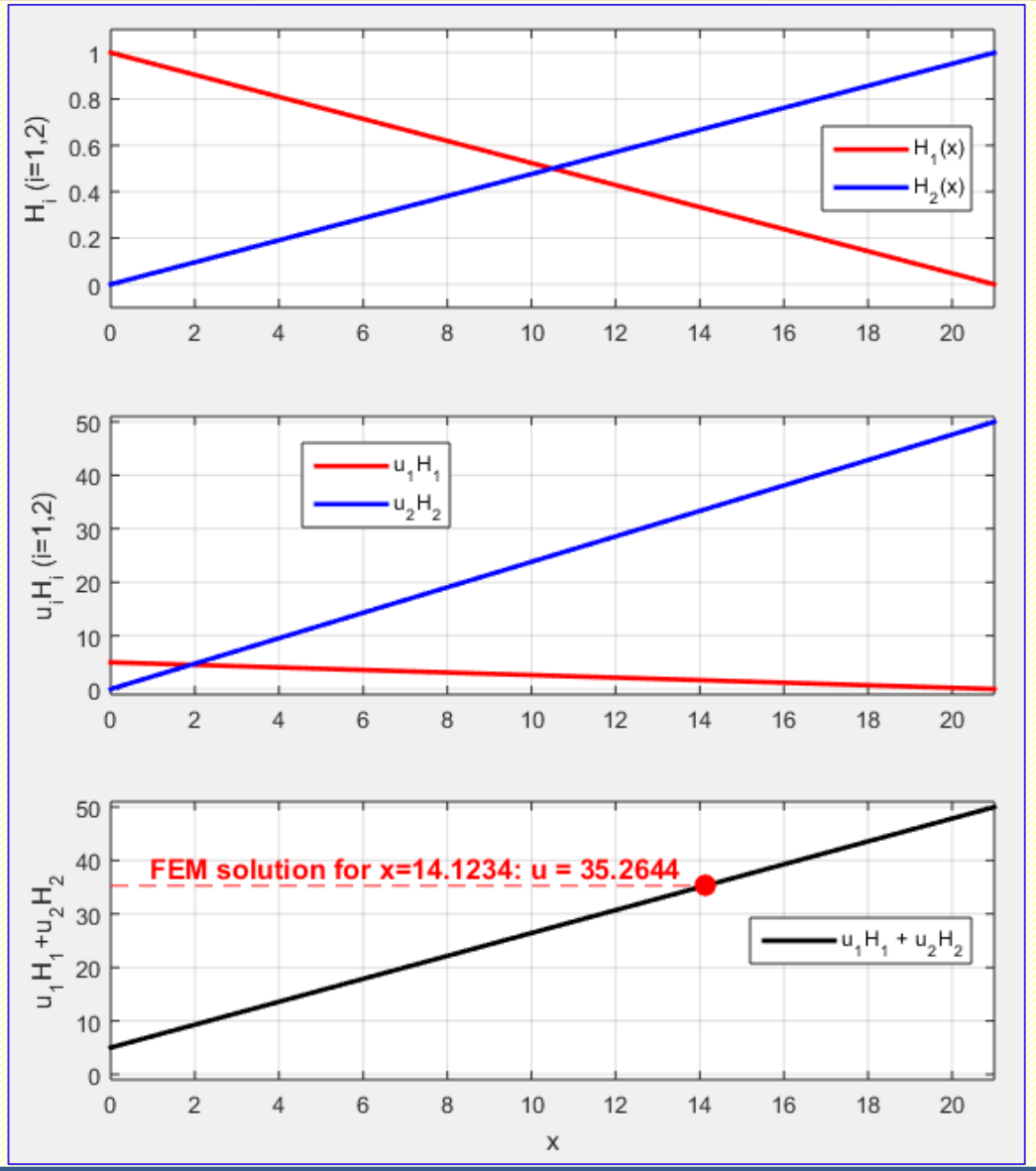
FEM: EXAMPLE-1 with TRUSS ELEMENT

$$h = 21 \quad [\text{m}]$$

$$u_1 = 5 \quad [\text{m}]$$

$$u_2 = 50 \quad [\text{m}]$$

$$u(14.1234) = ?$$



FEM: MATLAB Script for the EXAMPLE-1

```
%% OENG1116-S1-2020
% Designed by Prof P.M.Trivailo (C) 2020
% FEM example for the 2DOF Axial Rod
%---------------------------

clear; clc; close('all')
h=21; u1=5; u2=50; % m

x=[0:0.01:1]*h;

H1x = 1-x/h; H2x = x/h;

figure;
%---------------------------
subplot(3,1,1)
plot(x,H1x,'r', x,H2x,'b','LineWidth',2);
h_legend=legend('H_1(x)','H_2(x)','Location','East');
ylabel('H_i (i=1,2)')
grid on
axis([0 h -.1 1.1])
```

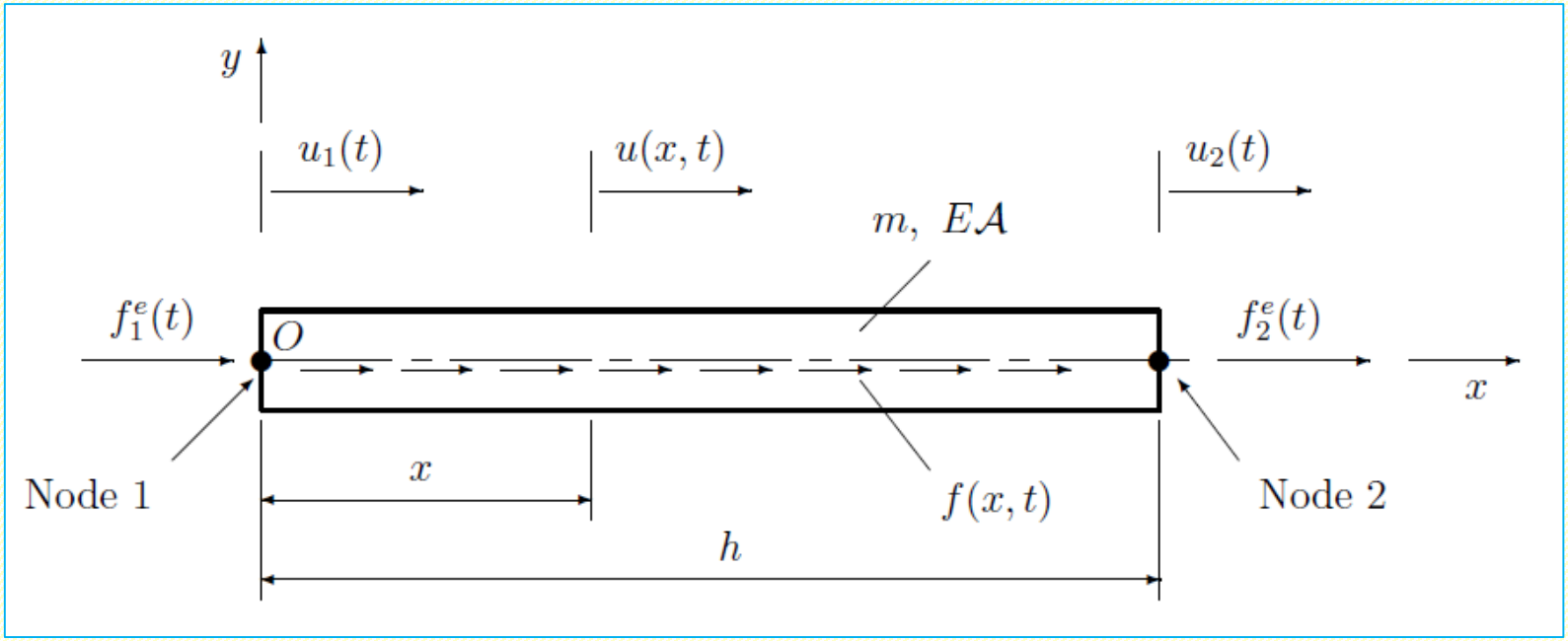
```
% Continuation of the Script
%-------------------------
subplot(3,1,2)
plot(x,u1*H1x,'r',  x,u2*H2x,'b','LineWidth',2);
legend('u_1H_1','u_2H_2','Location','best');
ylabel('u_iH_i (i=1,2)')
grid on
axis([0 h -1 51])
%-------------------------
subplot(3,1,3)
u = u1*H1x + u2*H2x;
plot(x,u,'k','LineWidth',2);
legend('u_1H_1 + u_2H_2','Location','East');
grid on
axis([0 h -1 51])
ylabel('u_1H_1+u_2H_2')
xlabel('x')
xC=14.1234; %Just an example: we would like to know u at this point
yC=interp1(x,u,xC);
line('XData',xC, 'YData',yC,'Marker','o','MarkerSize',8,'Color',[1 0 0])
set(gcf,'Position', [488 47 649 733])
```

FEM: Truss Element:

Static case of the Rod modelled with only 1 Finite Element:

ANALYSIS of FORCES

FEM: TRUSS ELEMENT: STATIC EQUATION



Model of the Rod, having only One Finite Element:

$$\left[k^e\right]\{u\} = \{F\} \quad \text{(matrix equation), or,}$$

$$\frac{EA}{h}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} \quad \text{(the same, but in the expanded format)}$$

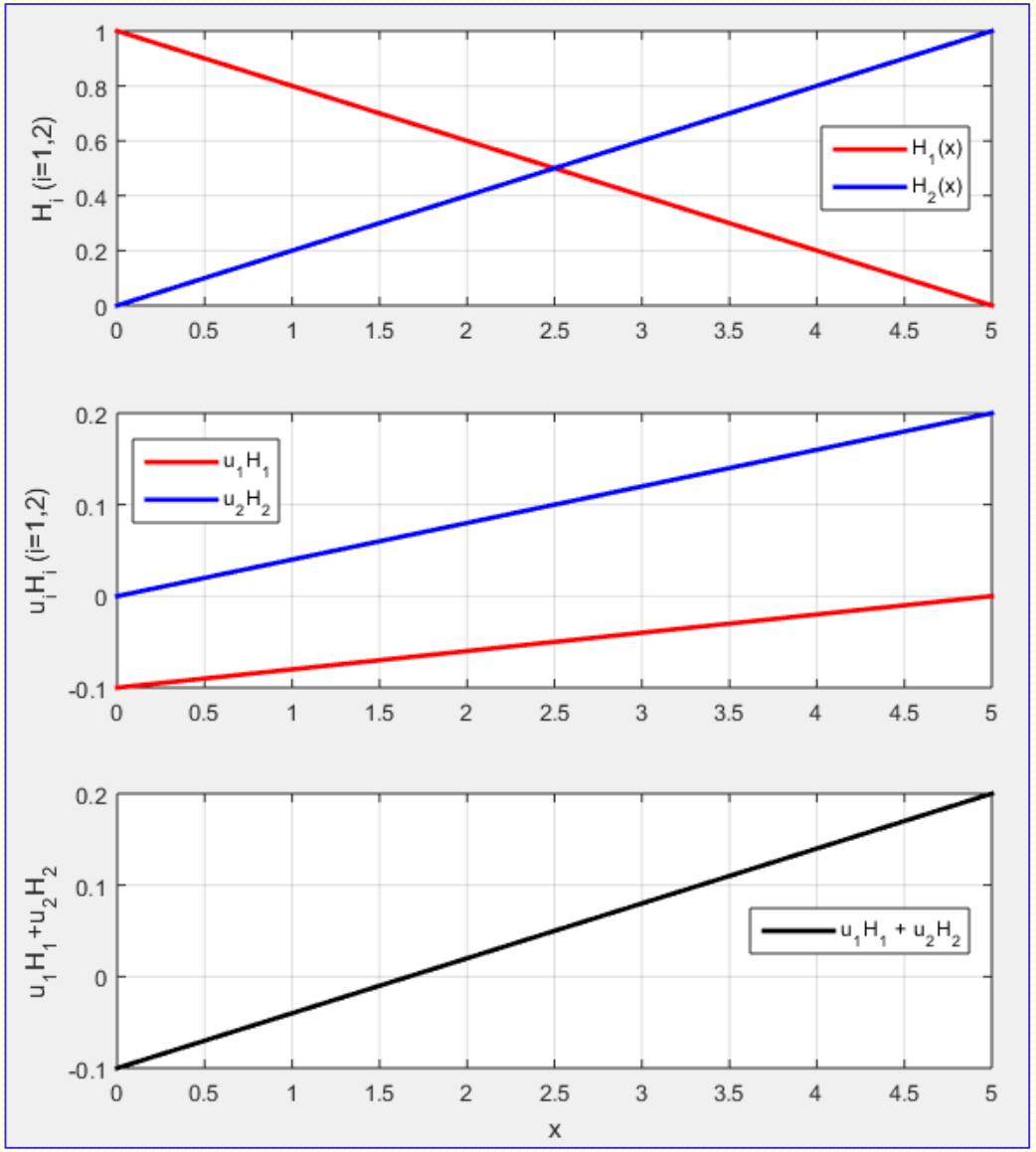
FEM: EXAMPLE-2 with TRUSS ELEMENT

$$h = 5 \quad [\text{m}]$$

$$u_1 = -0.1 \quad [\text{m}]$$

$$u_2 = 0.2 \quad [\text{m}]$$

$$F = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} = ?$$



FEM: MATLAB Script for the EXAMPLE-2

```
%% OENG1116-S1-2020
% Designed by Prof P.M.Trivailo (C) 2020
% FEM Example-2 for the 2DOF Axial Rod
%--------------------------
clear; clc; close('all')
h=21;
x=[0:0.01:1]*h; H1x = 1-x/h; H2x = x/h; % Shape Functions
figure; a=0.08; b=0.08; h=5; u1=-0.1; u2=0.2; % Data in [m]
E=0.01*10^9;                              % Young Modulus [Pa]
%--------------------------
subplot(3,1,1)
plot(x,H1x,'r', x,H2x,'b','LineWidth',2);
h_legend=legend('H_1(x)','H_2(x)','Location','East');
ylabel('H_i (i=1,2)'); grid on; axis tight;
%--------------------------
subplot(3,1,2)
plot(x,u1*H1x,'r',  x,u2*H2x,'b','LineWidth',2);
legend('u_1H_1','u_2H_2','Location','best');
ylabel('u_iH_i (i=1,2)'); grid on; axis tight;
%--------------------------
subplot(3,1,3)
u = u1*H1x + u2*H2x;
plot(x,u,'k','LineWidth',2);
legend('u_1H_1 + u_2H_2','Location','East');
grid on; axis tight; ylabel('u_1H_1+u_2H_2'); xlabel('x');
set(gcf,'Position', [488    47   649   733])
```

FEM: MATLAB Script for the EXAMPLE-2

```
% Continuation of the Script for Example-2
%--------------------------
%
% --- ANALYSING AXIAL FORCES ---
A=a*b; EA=E*A;
% --- Calculate Forces using FEM ---
ke=(EA/h)*[1 -1; -1 1];
F=ke*[u1;u2];
disp(sprintf('=== Case: u1=%g; u2=%g; h=%g; E=%g; a=%g;
b=%g',u1,u2,h,E,a,b))
disp(sprintf('F1=%12.2f [N]; F2=%12.2f [N]; ',F(1),F(2)))

% --- Checking Axial Forces using Hooke's Law ---
Fchk=(u2-u1)*EA/h;
disp(sprintf('Check by Hooke''s Law: |F|=%12.2f [N]',Fchk))

commandwindow
```

Command Window

New to MATLAB? See resources for Getting Started.

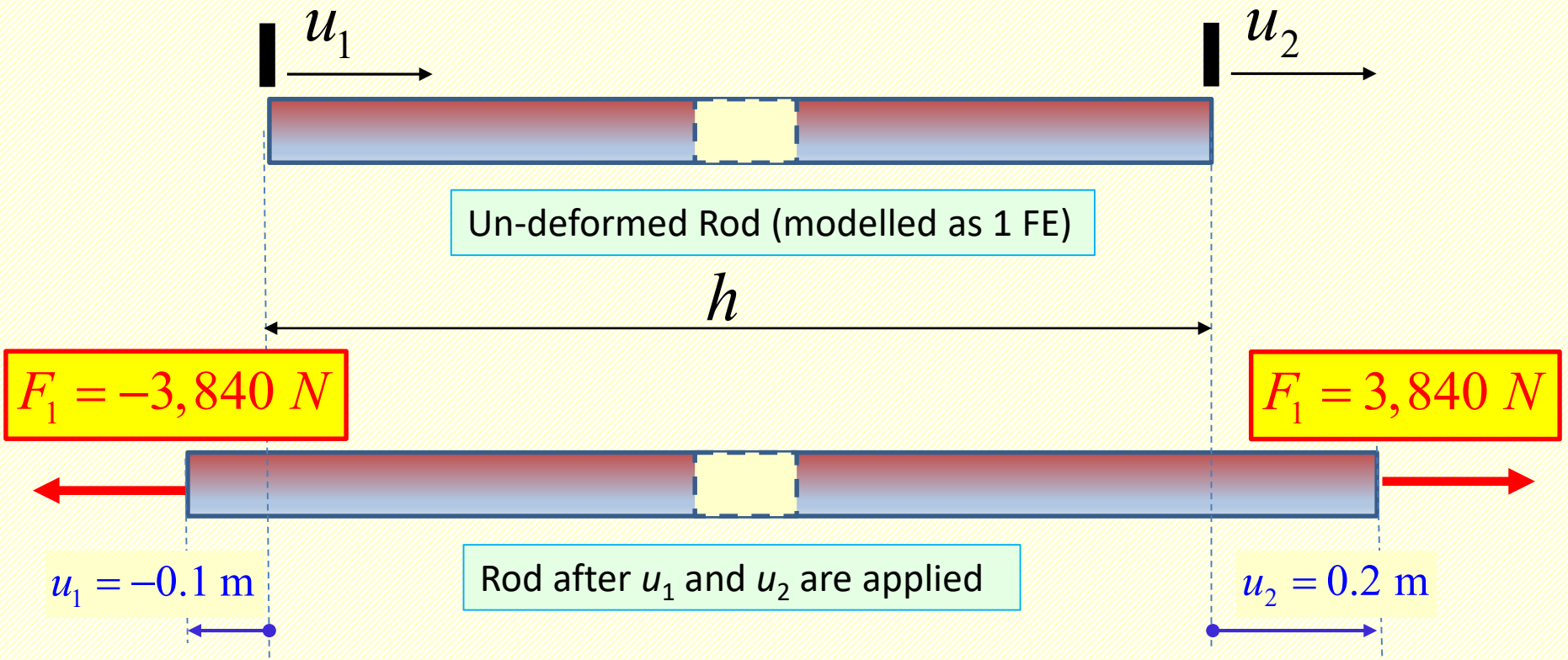
```
=== Case: u1=-0.1; u2=0.2; h=5; E=1e+07; a=0.08; b=0.08
F1=    -3840.00 [N]; F2=     3840.00 [N];
Check by Hooke's Law: |F|=     3840.00 [N]
```

FEM: Interpretation of Results for EXAMPLE-2

Command Window

New to MATLAB? See resources for Getting Started.

```
=== Case: u1=-0.1; u2=0.2; h=5; E=1e+07; a=0.08; b=0.08
F1=    -3840.00 [N]; F2=     3840.00 [N];
Check by Hooke's Law: |F|=     3840.00 [N]
```



FEM: Truss Element:

Comparison of frequencies, calculated with FEM and exact analytical expressions (for 50 FEs model)

FEM: MATLAB Script for the 50 FE Rod

```
%% ----- EXAMPLE: FEM modelling of free-free axial rod
% Designed by Prof P.M.Trivailo (C) 2020

%=== ENTERING DATA =========
clear; clc; close('all');
L=1; E=0.01*10^9; % Pa
rho=1.2*10^3;     % kg/m^3
a=0.08;  b=0.08;  A=a*b;  EA=E*A;
c=sqrt(E/rho);
NumFE=50;
h=L/NumFE; m=A*1*rho;

%=== BUILDING [M] & [K] global matrices =========
ke=(EA/h)*[1 -1; -1 1];
me=(m*h/6)*[2 1; 1 2];
M=zeros([1,1]*(NumFE+1)); K=zeros([1,1]*(NumFE+1));
for ii=1:NumFE
    idx=[1 2]+(ii-1);
    M(idx, idx)=M(idx, idx)+me;   K(idx, idx)=K(idx, idx)+ke;
end
```

$$[k^e] = \frac{EA}{h}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$[m^e] = \frac{mh}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

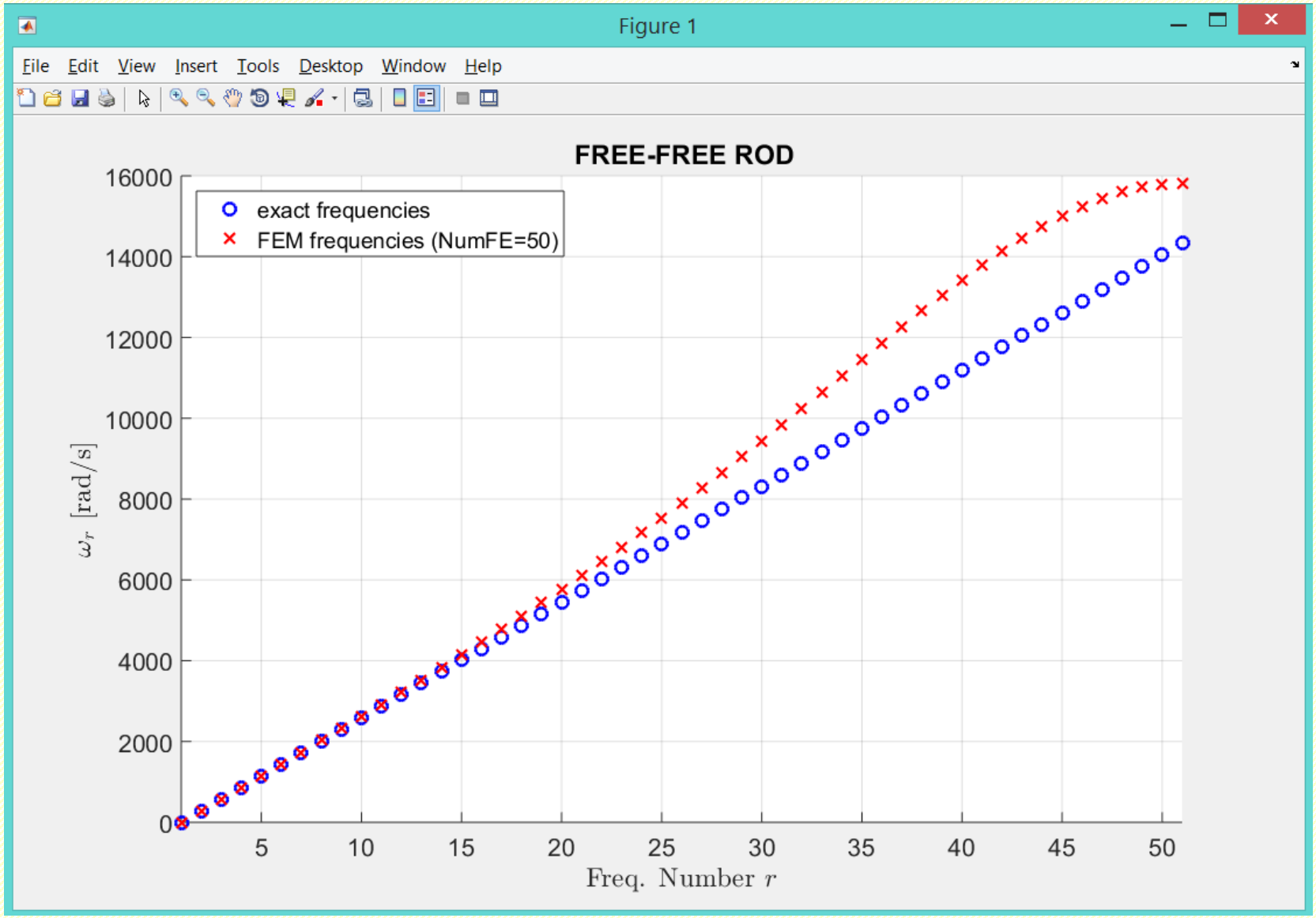
FEM: MATLAB Script(Continued-1)

```
%=== Solving Eigenvalue Problem =========
[U,D]=eig(K,M);

%=== Plotting FEM and exact frequencies =========
w1_exact=1*pi*sqrt(E/(rho*L^2));

figure; grid on; hold on;
for i=1:NumFE+1
    w_exact=(i-1)*w1_exact;
    w_FEM=sqrt(abs(diag(D(i,i))));
    disp(sprintf('w_%2i = %7.2f rad/s   w_exact_%2i = %7.2f rad/s',i,w_FEM,i,w_exact))
    plot(i,w_exact,'ob','MarkerSize',8); % plot i-th exact frequency
    plot(i,w_FEM,'xr','MarkerSize',8);    % plot i-th FEM frequency
end
```

FEM: MATLAB Script(Continued-2)

```
%=== "Decorations": ading legend, labels and title =========
str=sprintf('FEM frequencies (NumFE=%i)',NumFE);
legend('exact frequencies',str,'Location','NorthWest');
xlabel('Freq. Number $r$','Interpreter','LaTeX');
ylabel('$\omega_r$ [rad/s]','Interpreter','LaTeX');
title('\bf FREE-FREE ROD');
%
xlim([1 NumFE+1]);
set(gca,'FontSize',16);
set(gcf,'Position', [65 14 1107  680]);
if NumFE<10, set(gca,'XTick',[1:NumFE+1]); end
commandwindow
```

50 FE Model: Analytical & FEM Frequencies



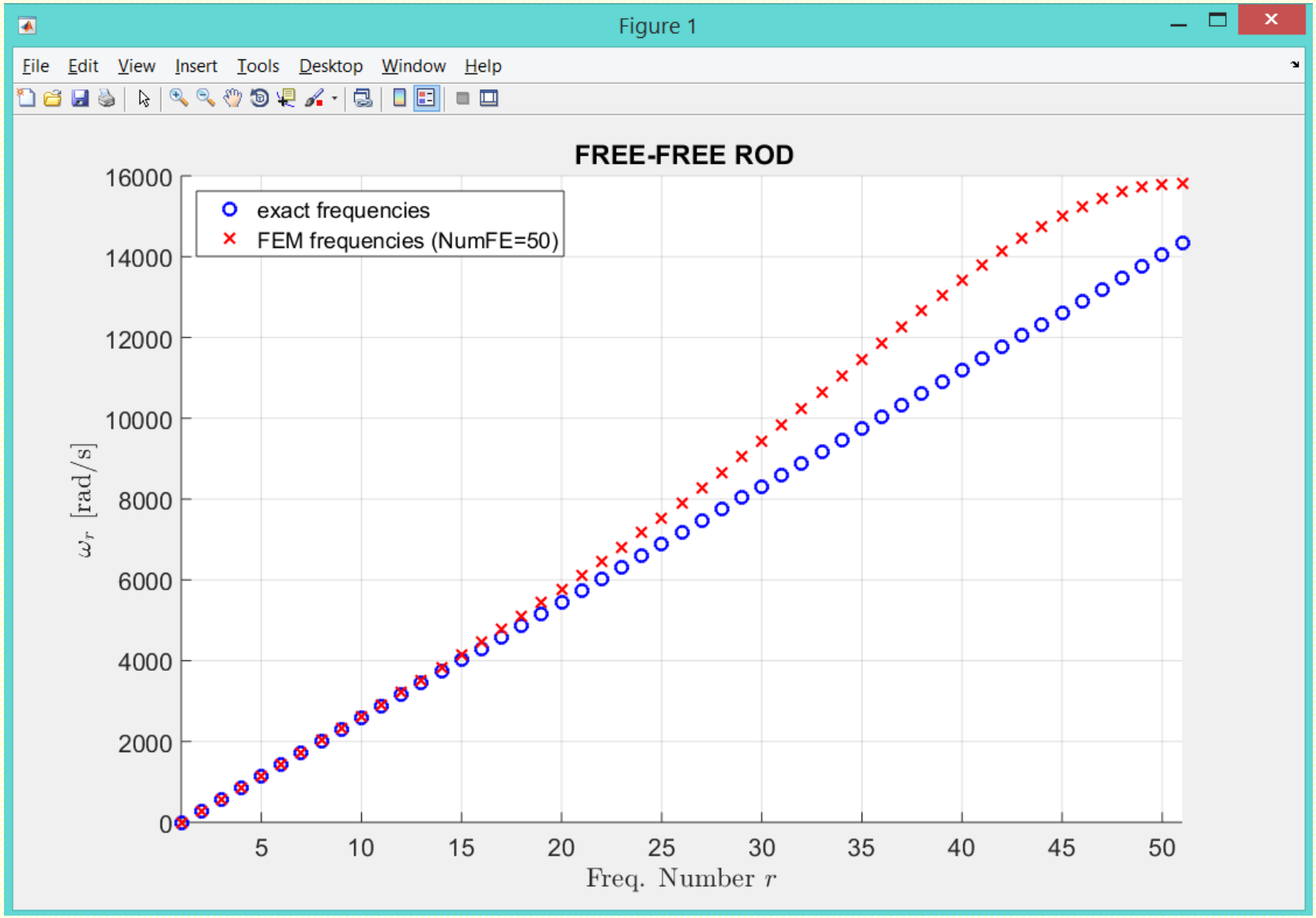
FEM: MATLAB Command Window Output

w_ 1 = 0.00 rad/s w_exact_ 1 = 0.00 rad/s
w_ 2 = 286.83 rad/s w_exact_ 2 = 286.79 rad/s
w_ 3 = 573.95 rad/s w_exact_ 3 = 573.57 rad/s
w_ 4 = 861.63 rad/s w_exact_ 4 = 860.36 rad/s
w_ 5 = 1150.17 rad/s w_exact_ 5 = 1147.15 rad/s
w_ 6 = 1439.84 rad/s w_exact_ 6 = 1433.93 rad/s
w_ 7 = 1730.93 rad/s w_exact_ 7 = 1720.72 rad/s
w_ 8 = 2023.73 rad/s w_exact_ 8 = 2007.51 rad/s
w_ 9 = 2318.52 rad/s w_exact_ 9 = 2294.29 rad/s
w_10 = 2615.60 rad/s w_exact_10 = 2581.08 rad/s
w_11 = 2915.25 rad/s w_exact_11 = 2867.87 rad/s
w_12 = 3217.76 rad/s w_exact_12 = 3154.66 rad/s
w_13 = 3523.43 rad/s w_exact_13 = 3441.44 rad/s
w_14 = 3832.54 rad/s w_exact_14 = 3728.23 rad/s
w_15 = 4145.38 rad/s w_exact_15 = 4015.02 rad/s
w_16 = 4462.24 rad/s w_exact_16 = 4301.80 rad/s
w_17 = 4783.38 rad/s w_exact_17 = 4588.59 rad/s
w_18 = 5109.09 rad/s w_exact_18 = 4875.38 rad/s
w_19 = 5439.62 rad/s w_exact_19 = 5162.16 rad/s
w_20 = 5775.22 rad/s w_exact_20 = 5448.95 rad/s
w_21 = 6116.11 rad/s w_exact_21 = 5735.74 rad/s
w_22 = 6462.49 rad/s w_exact_22 = 6022.52 rad/s
w_23 = 6814.53 rad/s w_exact_23 = 6309.31 rad/s
w_24 = 7172.37 rad/s w_exact_24 = 6596.10 rad/s
w_25 = 7536.08 rad/s w_exact_25 = 6882.88 rad/s
w_26 = 7905.69 rad/s w_exact_26 = 7169.67 rad/s
w_27 = 8281.15 rad/s w_exact_27 = 7456.46 rad/s
w_28 = 8662.31 rad/s w_exact_28 = 7743.25 rad/s
w_29 = 9048.92 rad/s w_exact_29 = 8030.03 rad/s
w_30 = 9440.63 rad/s w_exact_30 = 8316.82 rad/s
w_31 = 9836.89 rad/s w_exact_31 = 8603.61 rad/s
w_32 = 10237.04 rad/s w_exact_32 = 8890.39 rad/s
w_33 = 10640.16 rad/s w_exact_33 = 9177.18 rad/s
w_34 = 11045.16 rad/s w_exact_34 = 9463.97 rad/s
w_35 = 11450.64 rad/s w_exact_35 = 9750.75 rad/s
w_36 = 11854.97 rad/s w_exact_36 = 10037.54 rad/s
w_37 = 12256.18 rad/s w_exact_37 = 10324.33 rad/s
w_38 = 12651.99 rad/s w_exact_38 = 10611.11 rad/s
w_39 = 13039.78 rad/s w_exact_39 = 10897.90 rad/s
w_40 = 13416.61 rad/s w_exact_40 = 11184.69 rad/s
w_41 = 13779.19 rad/s w_exact_41 = 11471.47 rad/s
w_42 = 14123.97 rad/s w_exact_42 = 11758.26 rad/s
w_43 = 14447.18 rad/s w_exact_43 = 12045.05 rad/s
w_44 = 14744.89 rad/s w_exact_44 = 12331.84 rad/s
w_45 = 15013.13 rad/s w_exact_45 = 12618.62 rad/s
w_46 = 15248.04 rad/s w_exact_46 = 12905.41 rad/s
w_47 = 15445.96 rad/s w_exact_47 = 13192.20 rad/s
w_48 = 15603.63 rad/s w_exact_48 = 13478.98 rad/s
w_49 = 15718.34 rad/s w_exact_49 = 13765.77 rad/s
w_50 = 15788.02 rad/s w_exact_50 = 14052.56 rad/s
w_51 = 15811.39 rad/s w_exact_51 = 14339.34 rad/s

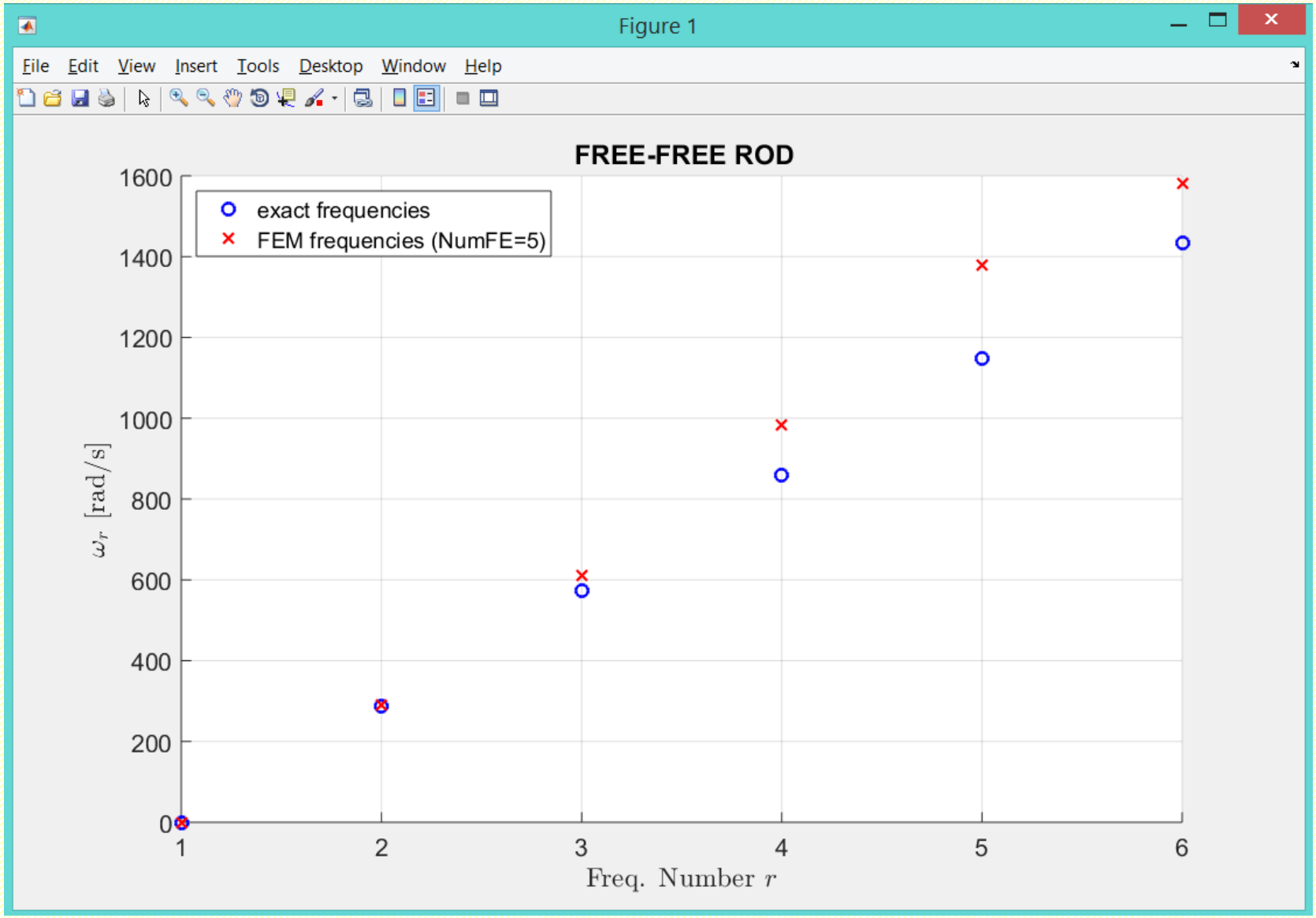
FEM: Truss Element:

Comparison of frequencies, calculated with FEM and exact analytical expressions (for 50, 5, 2 & 1 FEs model)

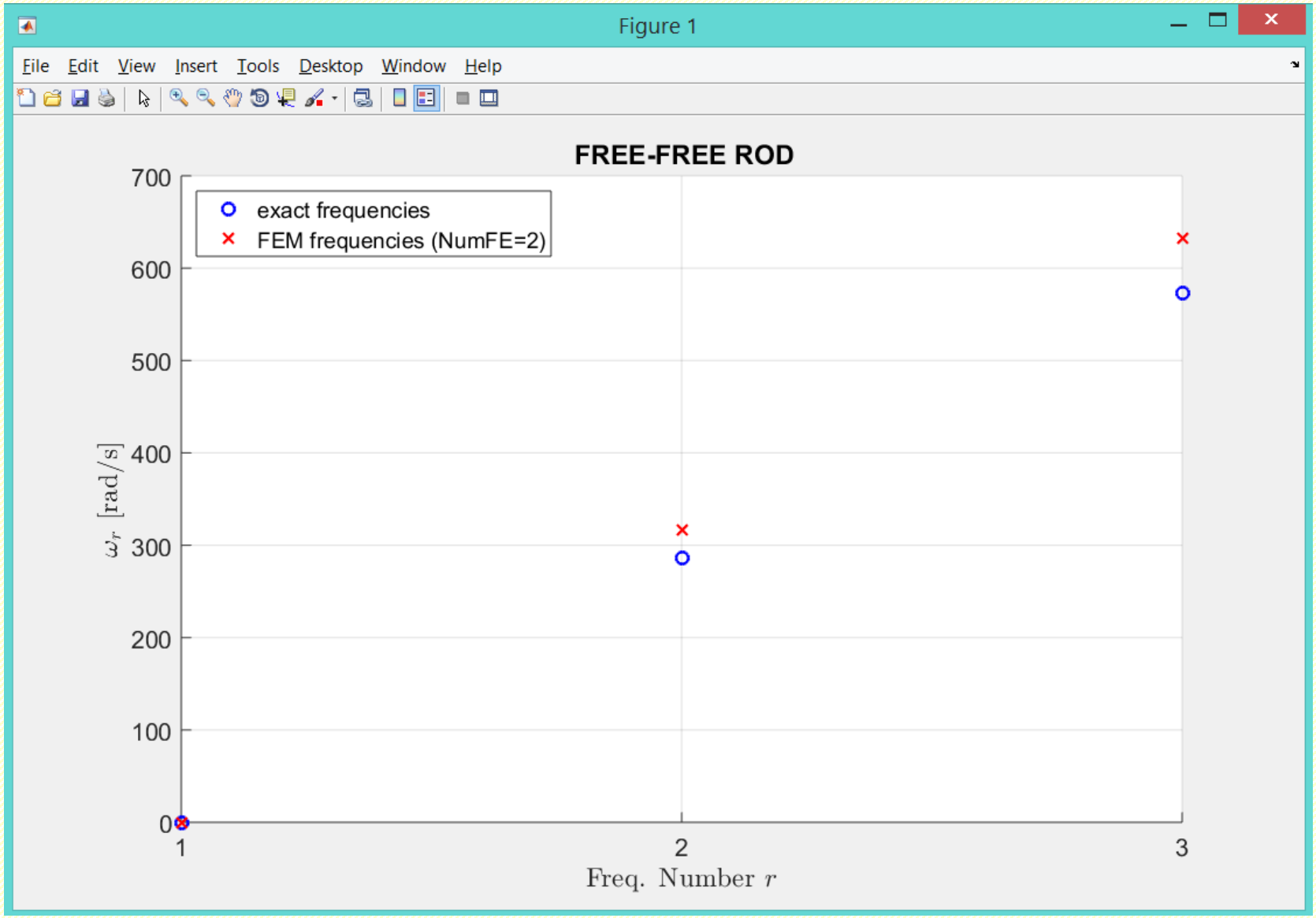
50 FE Model: Analytical & FEM Frequencies



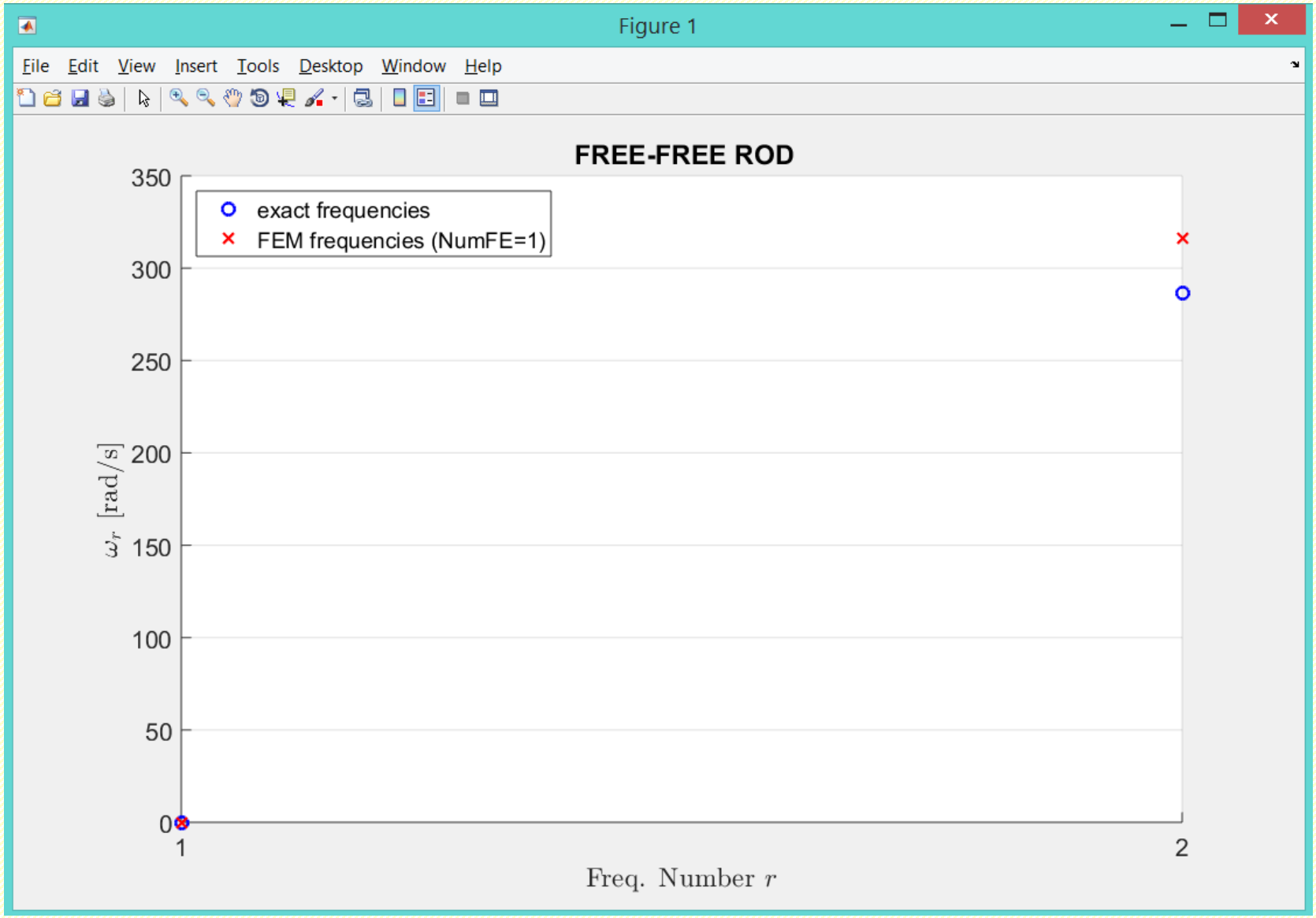
5 FE Model: Analytical & FEM Frequencies



2 FE Model: Analytical & FEM Frequencies



1 FE Model: Analytical & FEM Frequencies



FEM: Truss Element:

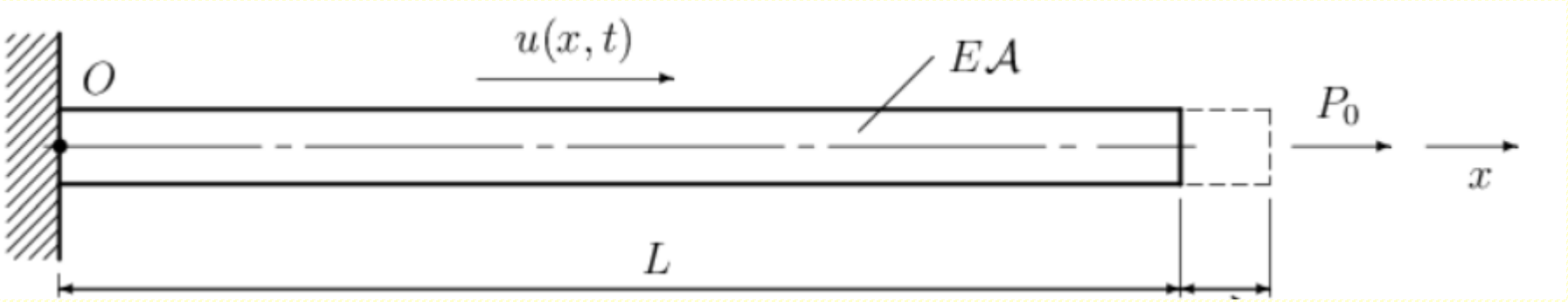
Dynamic case of the Rod modelled with only 1 Finite Element:

RESPONSE TO INITIAL CONDITIONS

EXAMPLE-2: Axial Vibrations & FEM

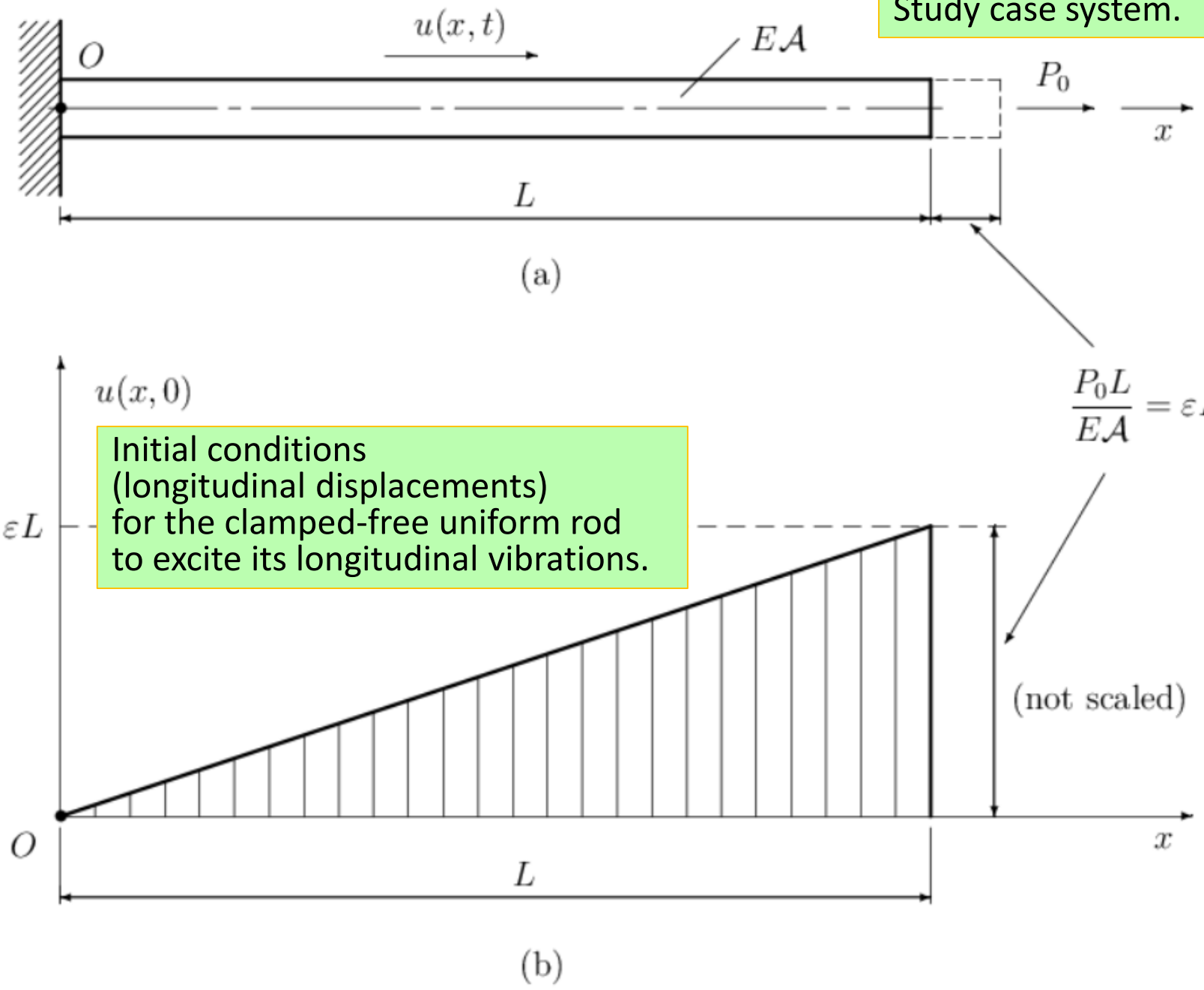
PROBLEM: As an exercise, we will use only ONE Finite Element to model rod, excited to vibrate in axial direction by application of the initial conditions (and no external force).

TASK: A uniform rod, with one end fixed and other free, is stretched under a static load, as shown in Figure, and suddenly released from rest at time t=0. From these initial conditions, determine the longitudinal displacements $u(x,t)$.



REFERENCE: **Trivailo P.M.** (2008), Vibrations: Theory & Aerospace Applications, Vol.1&2, - (The textbook for senior undergraduate and graduate aerospace students). - Melbourne: RMIT Publisher - 2008. - 247pp +348pp=595 pp., 355 ill, 4 software programs.

Study case system.



Initial conditions
(longitudinal displacements)
for the clamped-free uniform rod
to excite its longitudinal vibrations.

EQUATIONS OF MOTION

$$\frac{mh}{6}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\begin{Bmatrix}\ddot{u}_1\\ \ddot{u}_2\end{Bmatrix}+\frac{EA}{h}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}\begin{Bmatrix}u_1\\ u_2\end{Bmatrix}=\begin{Bmatrix}0\\ 0\end{Bmatrix}$$ (for supplementary FREE-FREE ROD)

Applying Boundary Conditions and FIXING left end (u_1=0)
by crossing out first equation and removing first rows on the remaining [m^e] and [k^e] matrices, we reduce previous equation to the following:

$$\frac{mh}{6}[2]\{\ddot{u}_2\}+\frac{EA}{h}[1]\{u_2\}=\{0\}$$ (for FIXED-FREE ROD) or

$$[M]\ddot{u}_2+[K]u_2=[0] \quad \left(\text{here } [M]=\frac{mh}{6}[2] \text{ and } [K]=\frac{EA}{h}[1]\right)$$

This matrix EQ can be re-written using state-spaces concept. Assume 2 states: $\mathbf{x}=\begin{Bmatrix}u_2\\ \dot{u}_2\end{Bmatrix}$ then

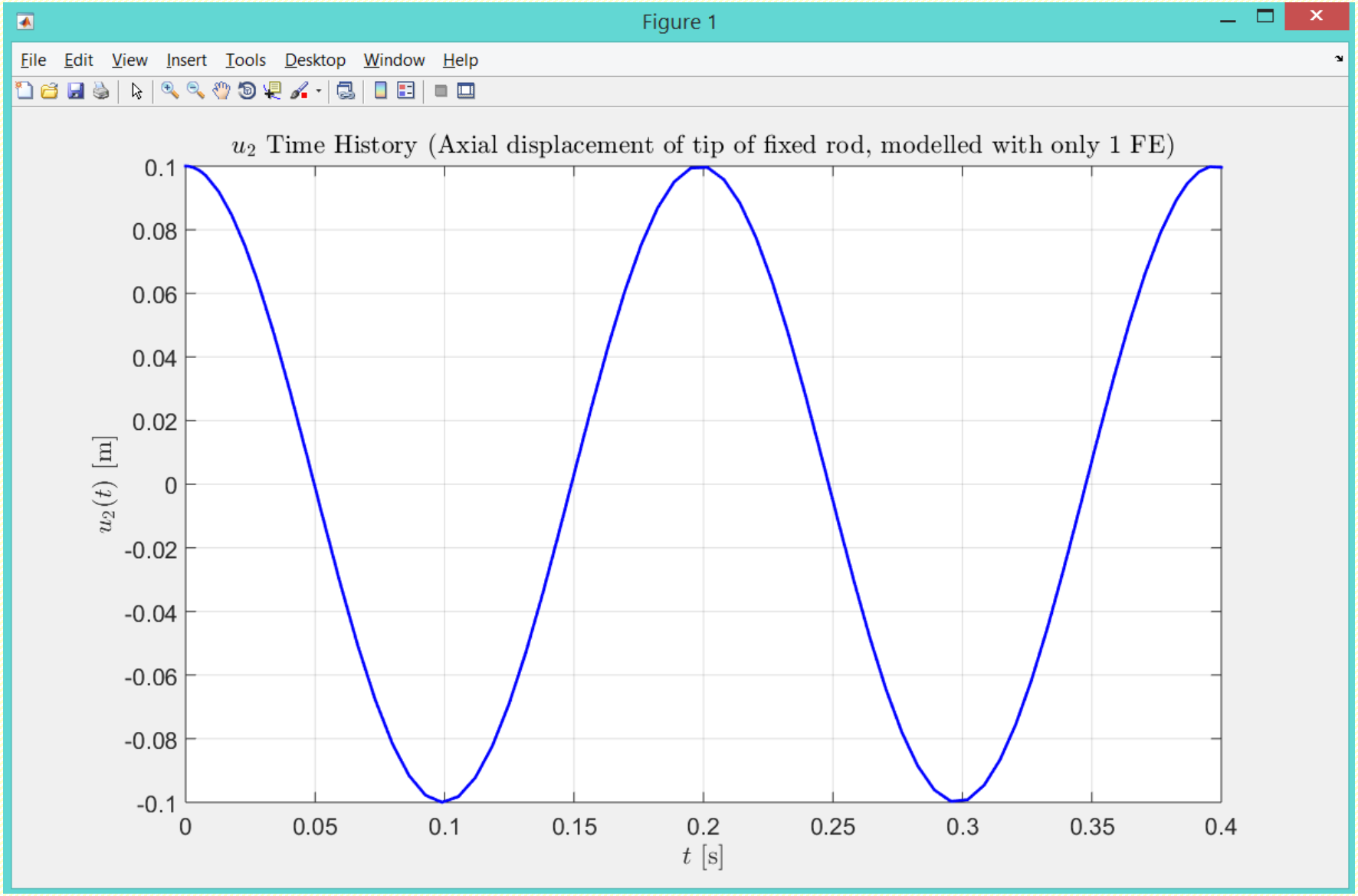
$$\begin{Bmatrix}\dot{x}_1\\ \dot{x}_2\end{Bmatrix}=\begin{Bmatrix}x_2\\ -\text{inv}([M])*[K]*x_1\end{Bmatrix}$$

MATLAB SCRIPT (Solving Response to Initial Conditions):

```
%% OENG1116-S1-2020
% Designed by Prof P.M.Trivailo (C) 2020
%--------------------------
clear; clc; close('all')
a=0.08; b=0.08; h=5; e=0.1; % Data in [m]
E=0.01*10^9;                % Young Modulus [Pa]
rho=1.2*10^3;               % density [kg/m^3]
tmax=0.4;                   % Simulation time [s]
A=a*b; EA=E*A; m=rho*a*b*1;

x=[0:0.01:1]*h; H1x = 1-x/h; H2x = x/h; % Shape Functions
% --- SUPPLEMENTARY SYSTEM: Modelling rod with only 1 FE ---
ke=(EA/h)*[1 -1; -1 1]; me=(m*h/6)*[2 1; 1 2];
% --- MAIN SYSTEM: Constraining Left Boundary ---
me(:,1)=[]; me(1,:)=[]; ke(:,1)=[]; ke(1,:)=[];

invMxK=inv(me)*ke; x0=[e; 0];
FEM_xdot_anon = @(t,x) [x(2); -invMxK*x(1) ];
[tt,xx] = ode45(FEM_xdot_anon,[0 tmax],x0);
figure; plot(tt,xx(:,1),'b','LineWidth',2); grid on;

xl=xlabel('$t$  [s]'); yl=ylabel('$u_2(t)$   [m]')
str=sprintf('$u_2$ Time History (Axial displacement of tip of fixed rod, modelled with only 1 FE)');
ti=title(str,'FontWeight','bold');
set([xl,yl,ti],'Interpreter','LaTeX');
set(gca,'FontSize',16);
set(gcf,'Position',[120 30  1200  700])
```

Main Commands

Plot *x-t-u* with "*u*" contour lines



Plotting Response Results for Clamped-Free Rod, modelled with only 1 FE:

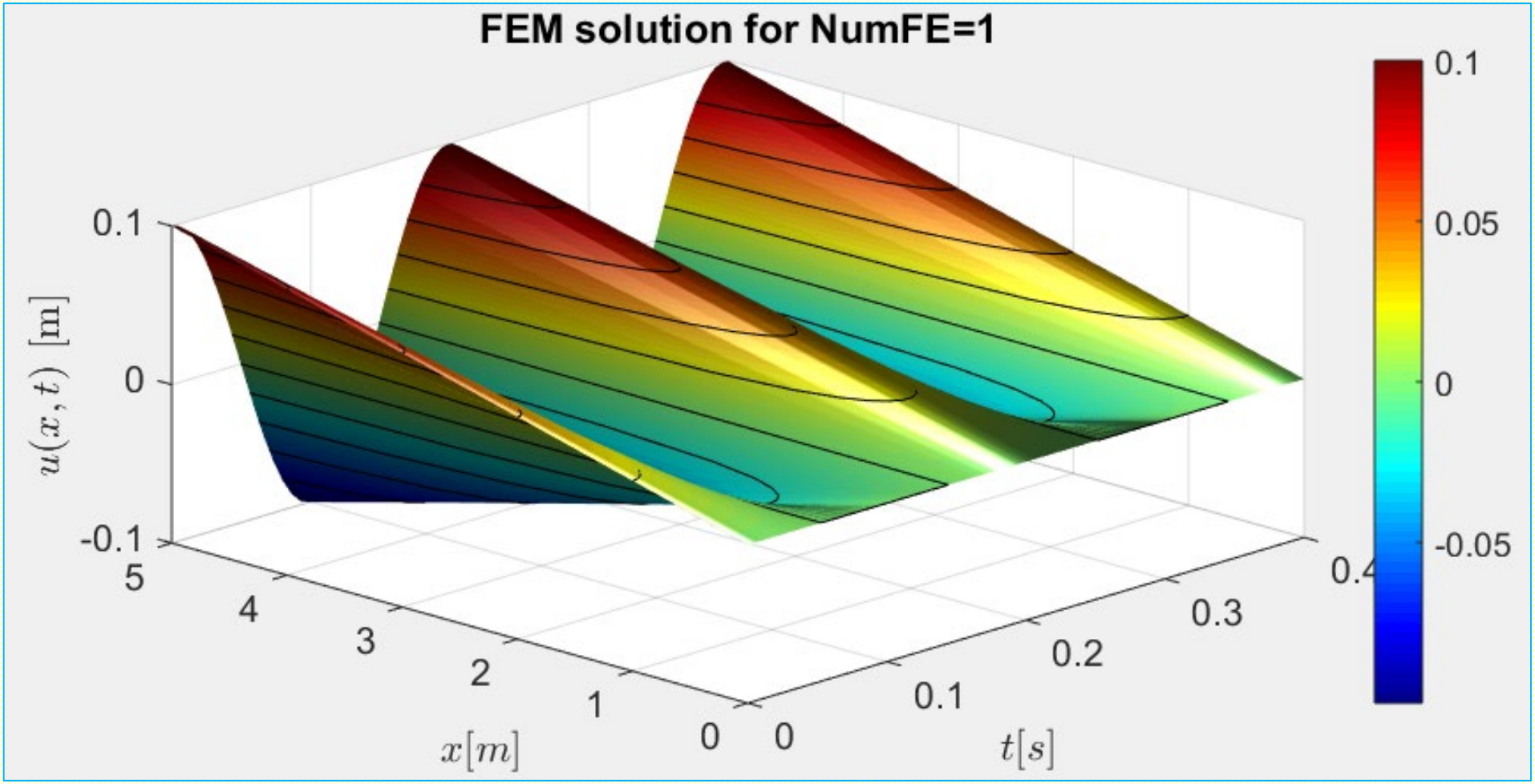
The same data is plotted, using different order of axes for 3D surface plot & "slicing" surface along each of the axes

MATLAB SCRIPT
(Plot *x-t-u* with "*u*" contour lines):

```
%% OENG1116-S1-2020
% Designed by Prof P.M.Trivailo (C) 2020
%--------------------------
%%
figure; hold on; grid on; rotate3d on;
[TT,XX]=meshgrid(tt,[0 h]);
ZZ=[0*xx(:,1)'; xx(:,1)'];
surf(TT,XX,ZZ);
colormap jet; lighting phong; shading interp;
contour3(TT,XX,ZZ,[-1:0.2:1]*e,'k');
view([-46, 36]);
%--- "Decorations" ---
colorbar; hh=camlight;
set(hh,'Position',[0.09,0.04,0.03]);
xl=xlabel('$t [s]$'); yl=ylabel('$x [m]$');
zl=zlabel('$u(x,t)$ [m]');
set([xl,yl,zl],'Interpreter','LaTeX');
str=sprintf('FEM solution for NumFE=1');
title(str,'FontWeight','bold');
set(gca,'FontSize',16);
set(gcf,'Position',[120   300   920   450]);
```

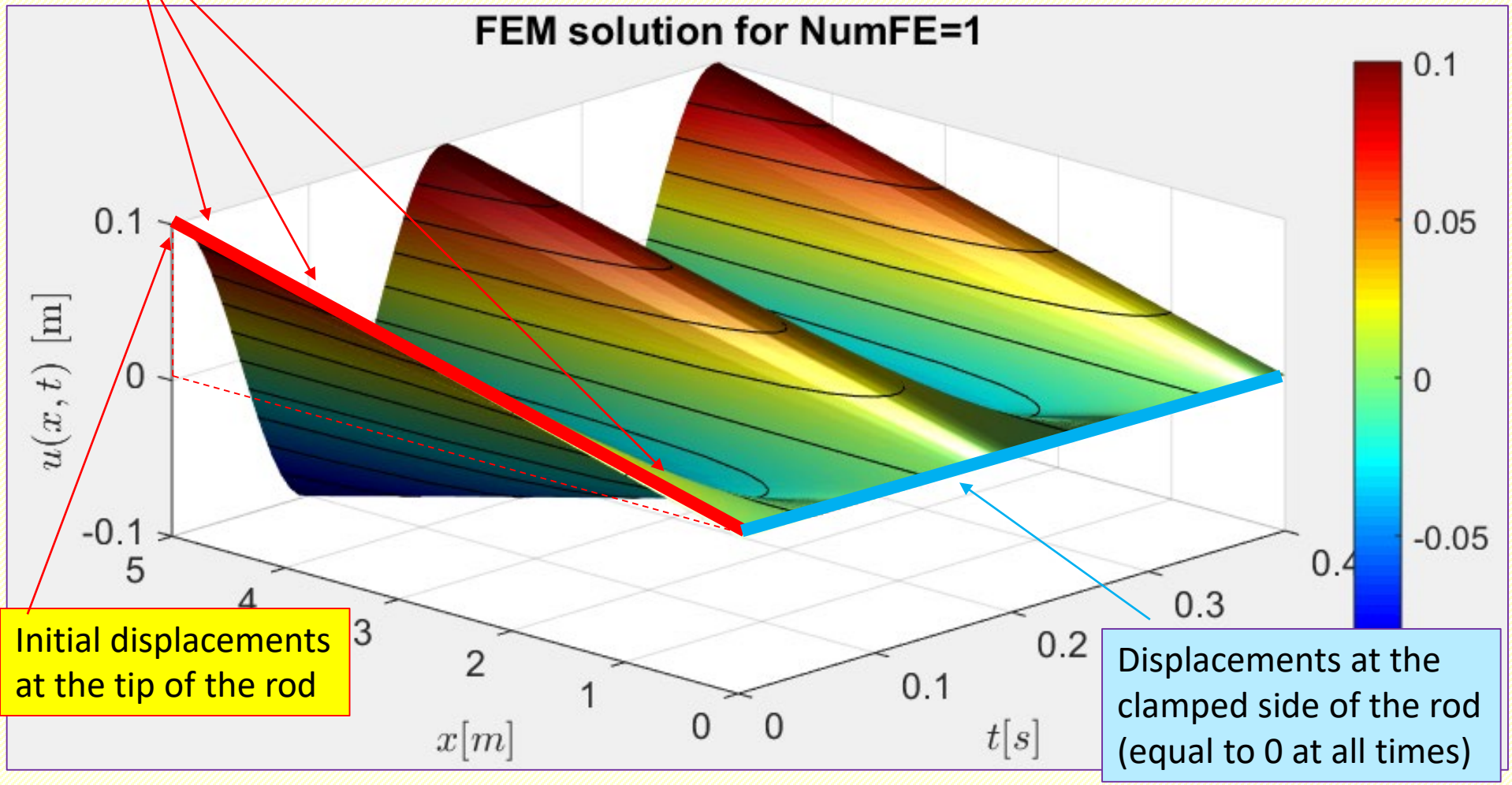
Main Commands

Plot *x-t-u* with "*u*" contour lines



Plot *x-t-u* with "*u*" contour lines

Initial axial displacements along the rod



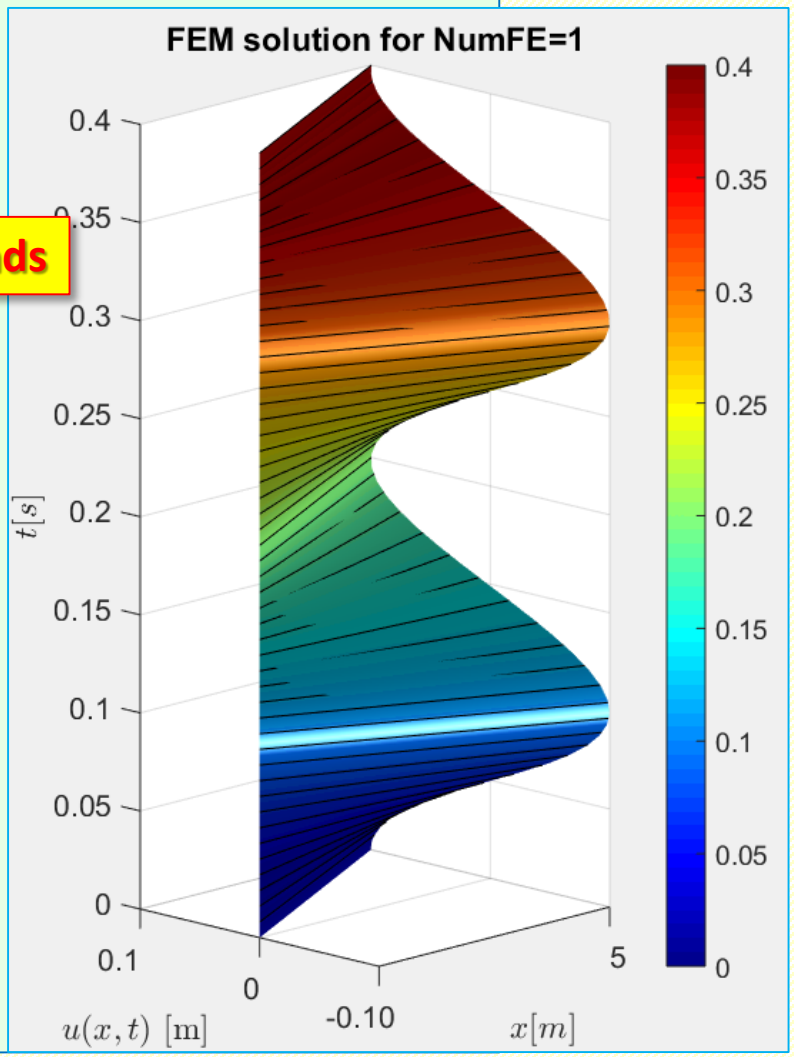
Initial displacements at the tip of the rod

Displacements at the clamped side of the rod (equal to 0 at all times)

MATLAB SCRIPT
(Plot *x-u-t* with "*t*" contour lines):

```
% Continuation of the Script
% Designed by Prof P.M.Trivailo (C) 2020
%-------------------------
%%
figure; hold on; grid on; rotate3d on;
surf(XX,ZZ,TT);
colormap jet;
lighting phong; shading interp;
contour3(XX,ZZ,TT,[0:0.02:1]*tmax,'k');
view([-46, 6]);

%--- "Decorations"
colorbar; hh=camlight;
set(hh,'Position',[2,-4, 0.11]);
zl=zlabel('$t [s]$'); xl=xlabel('$x [m]$');
yl=ylabel('$u(x,t)$ [m]');
set([xl,yl,zl],'Interpreter','LaTeX');
str=sprintf('FEM solution for NumFE=1');
title(str,'FontWeight','bold');
set(gca,'FontSize',16);
set(gcf,'Position',[488   10   560   814]);
```

Main Commands

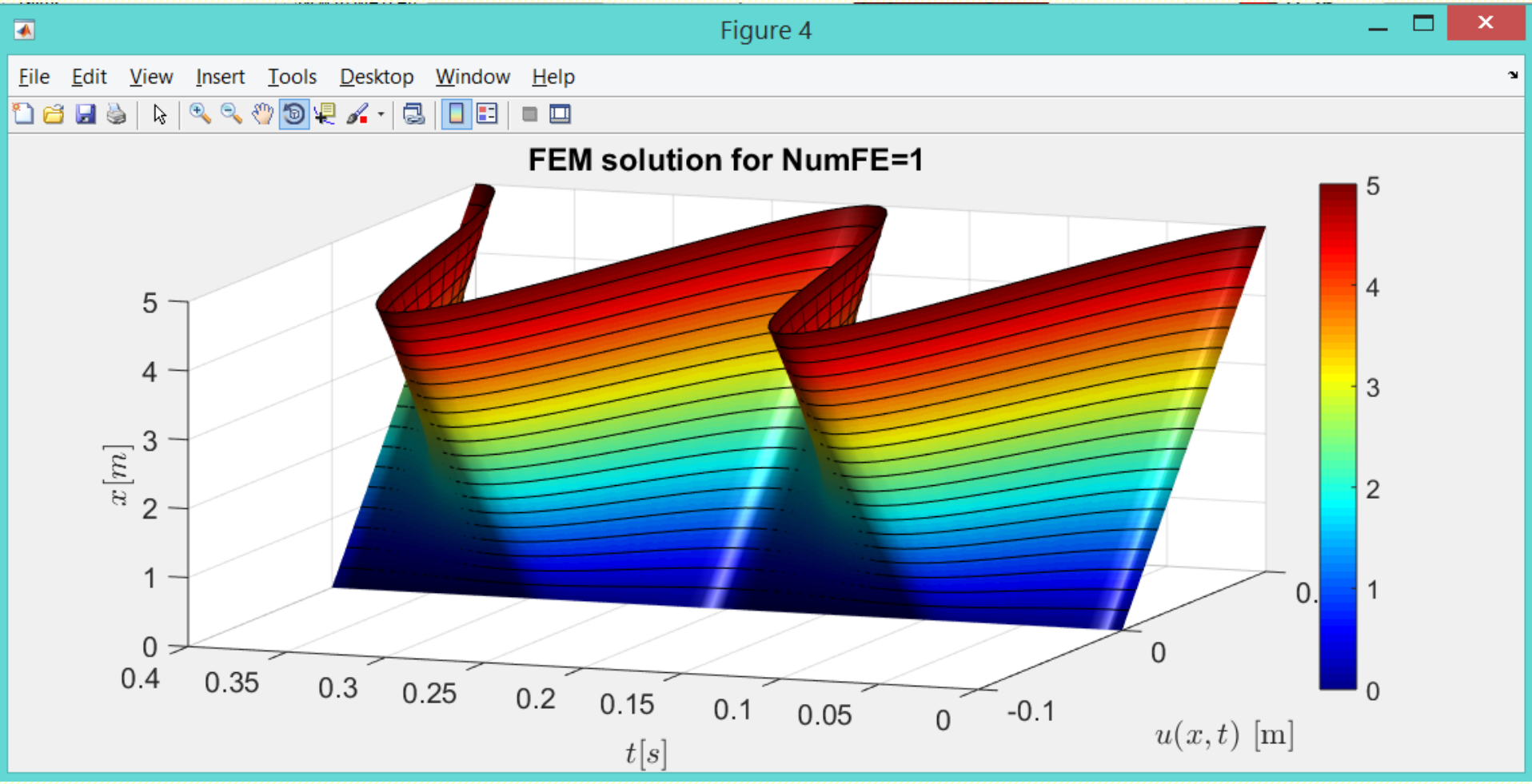


MATLAB SCRIPT
(Plot *u-t-x* with "*x*" contour lines):

```
% Continuation of the Script
% Designed by Prof P.M.Trivailo (C) 2020
%-------------------------
%%
figure; hold on; grid on; rotate3d on;
surf(ZZ,TT,XX);
colormap jet;
lighting phong; shading interp;
contour3(ZZ,TT,XX,[0:0.05:1]*h,'k');
view([-70, 20]);

%--- "Decorations"
colorbar; hh=camlight;
set(hh,'Position',[-0.4, -1.2, 2]);
yl=ylabel('$t [s]$'); zl=zlabel('$x [m]$');
xl=xlabel('$u(x,t)$ [m]');
set([xl,yl,zl],'Interpreter','LaTeX');
str=sprintf('FEM solution for NumFE=1');
title(str,'FontWeight','bold');
set(gca,'FontSize',16);
set(gcf,'Position',[40 80  1120   470]);
```

Main Commands

MATLAB SCRIPT
(Plot *u-t-x* with "*x*" contour lines):



Plotting Response Results for Clamped-Free Rod, modelled with 50 FE:

The data is plotted, using "*x-t-u*" axes

MATLAB SCRIPT
(Plot *x-t-u* with "*u*" contour lines):

```
%% OENG1116-S1-2020
% Designed by Prof P.M.Trivailo (C) 2020
%--------------------------
clear; clc; close all
a=0.08; b=0.08; L=5; e=0.1; % Data in [m]
E=0.01*10^9;                % Young Modulus [Pa]
rho=1.2*10^3;               % density [kg/m^3]
tmax=0.4;                   % Simulation time [s]
A=a*b; EA=E*A; m=rho*a*b*1;
NumFE=50;
h=L/NumFE;
x=[0:0.01:1]*h; H1x = 1-x/h; H2x = x/h; % Shape Functions
% Elementary [ke] and [me] matices ---
ke=(EA/h)*[1 -1; -1 1]; me=(m*h/6)*[2 1; 1 2];

% --- SUPPLEMENTARY (FREE-FREE) SYSTEM ---- :
M=zeros(NumFE+1); K=zeros(NumFE+1);
for i=1:NumFE
    idx=[1 2]+(i-1);
    M(idx,idx)=M(idx,idx)+me; K(idx,idx)=K(idx,idx)+ke;
end
```

Assembling global [M] and [K] matrices for the FREE-FREE rod (supplementary system)

MATLAB SCRIPT (Continued-1)

CORE of the Program

```
%% Continuation of the script
% --- MAIN SYSTEM: Constraining Left Boundary ---
M(:,1)=[]; M(1,:)=[]; K(:,1)=[]; K(1,:)=[];

invMxK=inv(M)*K;
% --- Formulating Boundary Conditions & EOM ---
x0=[e*[1:NumFE]/NumFE, zeros(1,NumFE)]';
FEM_xdot_anon = @(t,x) [eye(NumFE)*x(NumFE+1:2*NumFE); -invMxK*x(1:NumFE)];
[tt,xx] = ode45(FEM_xdot_anon,[0 tmax],x0);

% --- Plot axial displacements of the TIP of the ROD --------
figure; plot(tt,xx(:,NumFE),'b','LineWidth',2); grid on;
xl=xlabel('$t$  [s]'); yl=ylabel('$u_2(t)$   [m]')
str=sprintf('$u_2$ Time History (Axial displacement of tip of fixed rod, modelled with %i FE)',NumFE);
ti=title(str,'FontWeight','bold');
set([xl,yl,ti],'Interpreter','LaTeX');
set(gca,'FontSize',16);
set(gcf,'Position',[30 60  1200  700])
```

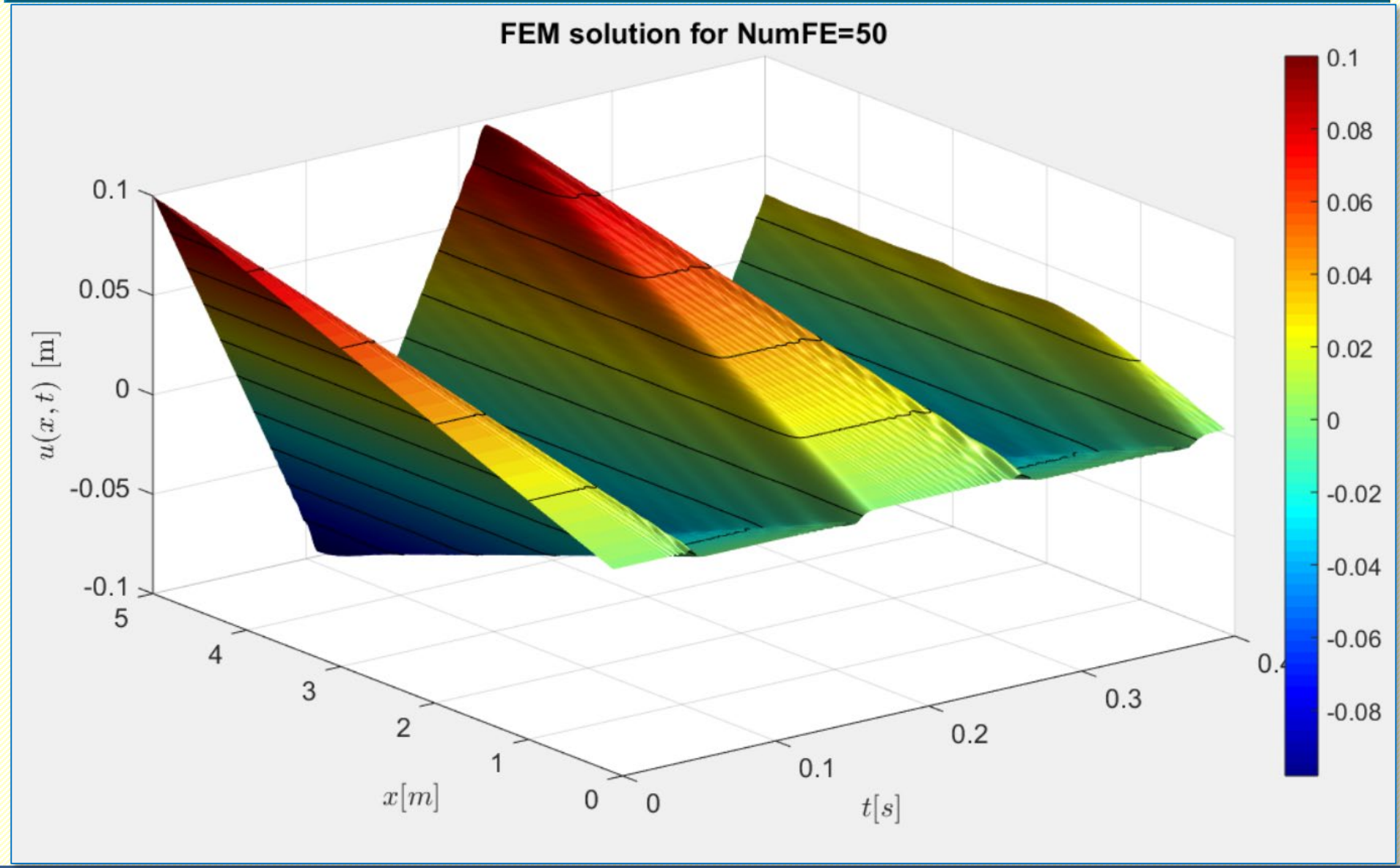
MATLAB SCRIPT (Continued-2)

Main Surface Plotting Commands

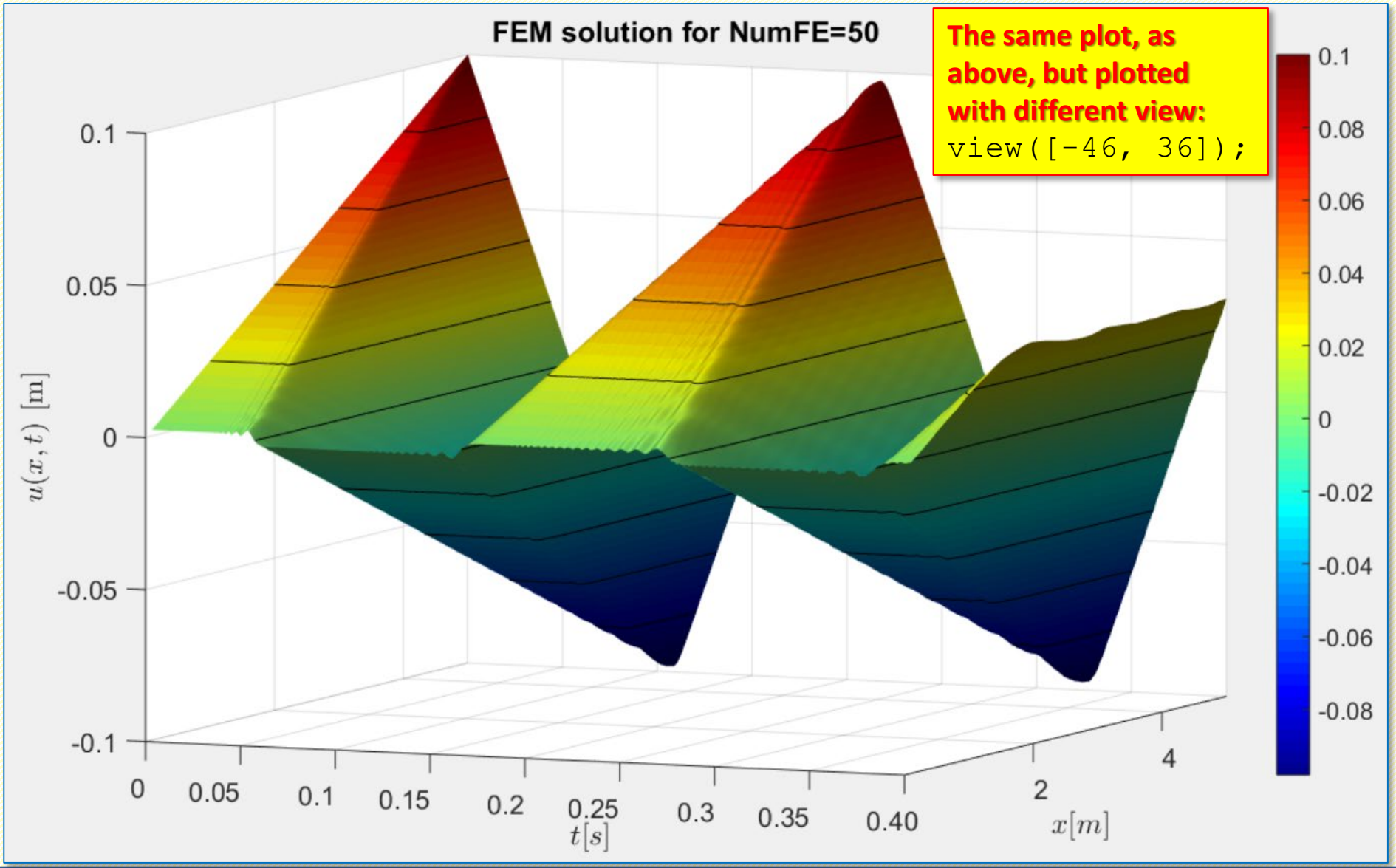
"Decorations"

```
%% Continuation of the script
% --- SURFACE Plot of axial displacements --------
if NumFE>1,
    [TT,XX]=meshgrid(tt,L*[1:NumFE]/NumFE);
    surf(TT,XX,xx(:,1:NumFE)');
    axis([0 tmax 0 L -e e]);
    grid on; hold on;
    xlabel('$t [s]$','Interpreter','LaTeX');
    ylabel('$x [m]$','Interpreter','LaTeX');
    zlabel('$u(x,t)$ [m]','Interpreter','LaTeX');
    str=sprintf('FEM solution for NumFE=%i',NumFE);
    title(str,'FontWeight','bold');
    rotate3d on;
    colormap jet; colorbar;
    lighting phong; shading interp; camlight;
    contour3(TT,XX,xx(:,1:NumFE)',[-0.1:0.02:0.1],'k');
    set(gca,'FontSize',16);
end
```

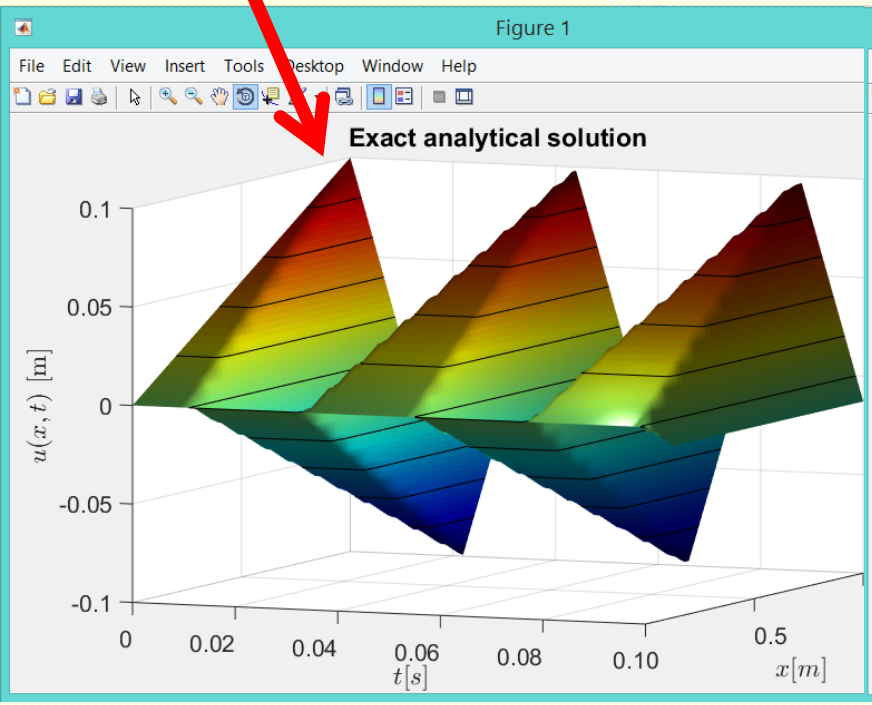
Resultant plot *x-t-u* with "*u*" contour lines

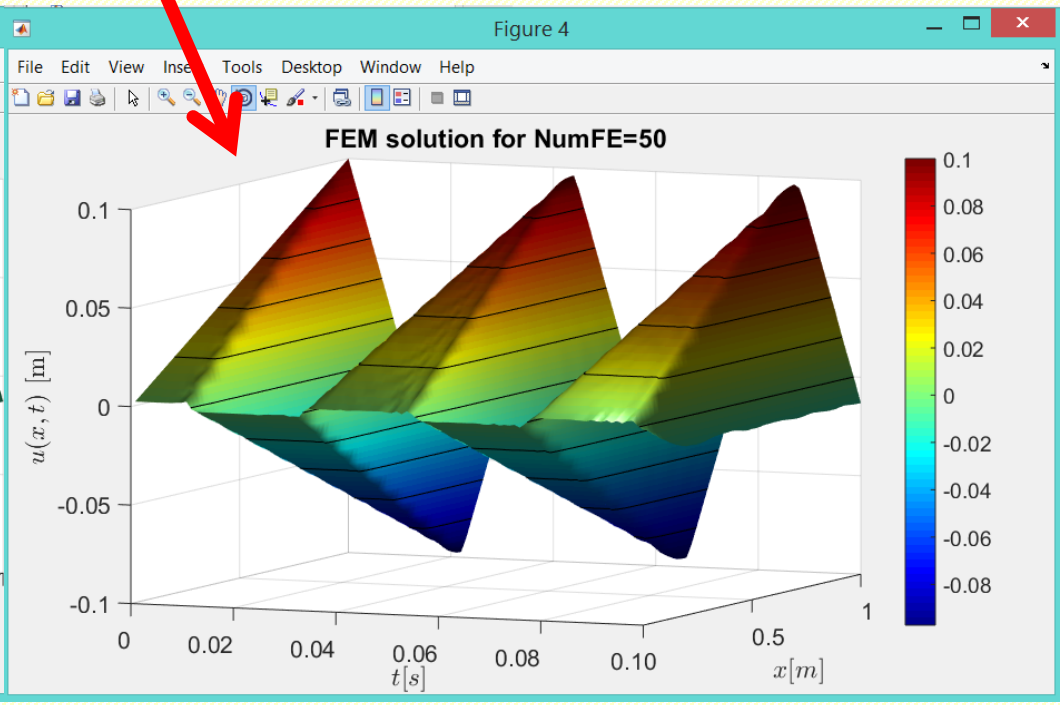


Resultant plot *x-t-u* with "*u*" contour lines



SIDE BY SIDE COMPARISON: EXACT (left) & FEM (right) SOLUTIONS and their representations



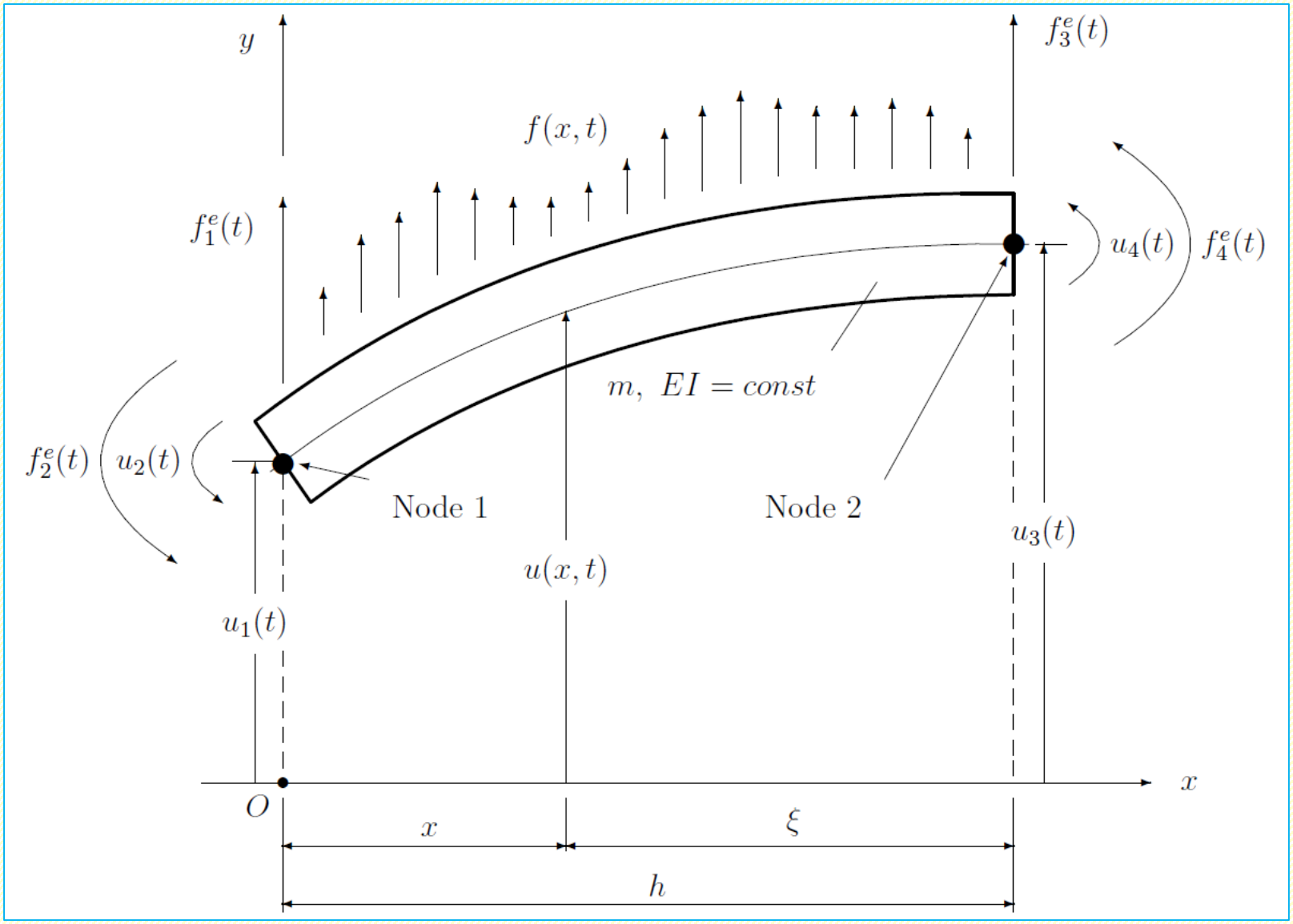


Comment: for comparison we are using the same axis limits and the same lighting

FEM: Beam Element:

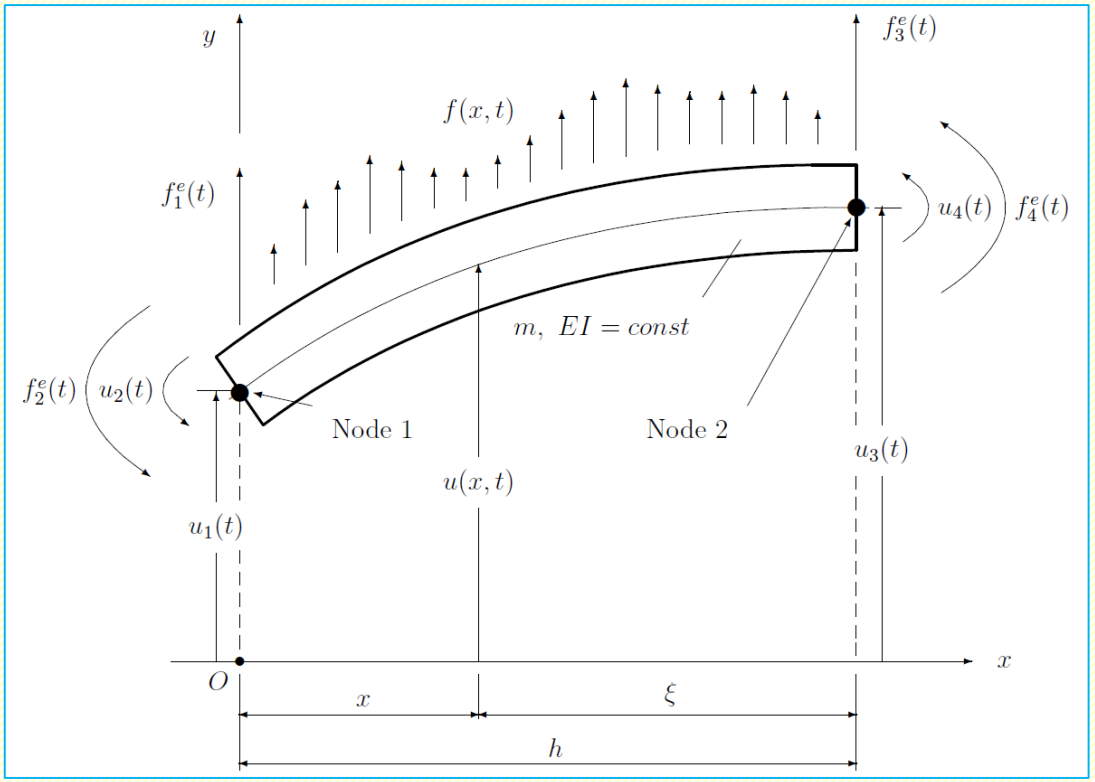
Shape Functions;
$[m^e]$ & $[k^e]$ matrices

FEM: “Euler–Bernoulli Beam” ELEMENT



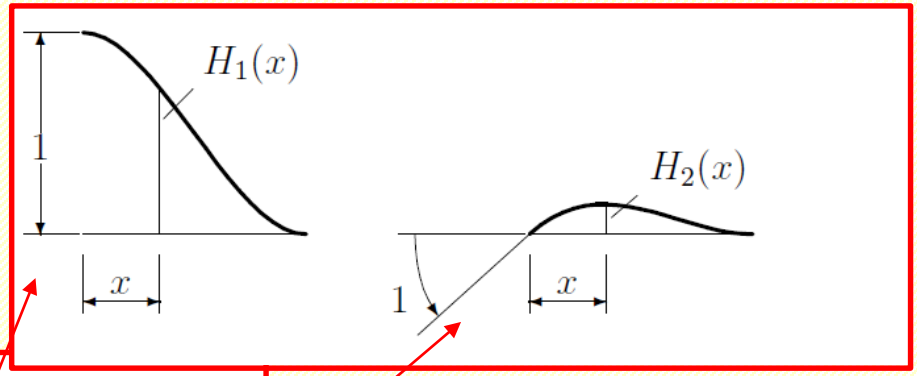
FEM: "Euler–Bernoulli Beam" ELEMENT

$$u(x,t) = \left[1 - 3\left(\frac{x}{h}\right)^2 + 2\left(\frac{x}{h}\right)^3\right] u_1(t) + h\left[\left(\frac{x}{h}\right) - 2\left(\frac{x}{h}\right)^2 + \left(\frac{x}{h}\right)^3\right] u_2(t) +$$
$$+ \left[3\left(\frac{x}{h}\right)^2 - 2\left(\frac{x}{h}\right)^3\right] u_3(t) + h\left[-\left(\frac{x}{h}\right)^2 + \left(\frac{x}{h}\right)^3\right] u_4(t). \qquad (1.40)$$



$$u(x,t) = H_1(x)\,u_1(t) + H_2(x)\,u_2(t) + H_3(x)\,u_3(t) + H_4(x)\,u_4(t)$$

FEM: SHAPE FUNCTIONS for BEAM ELEMENT



Hermite Interpolation Functions for a 2D Beam FE:

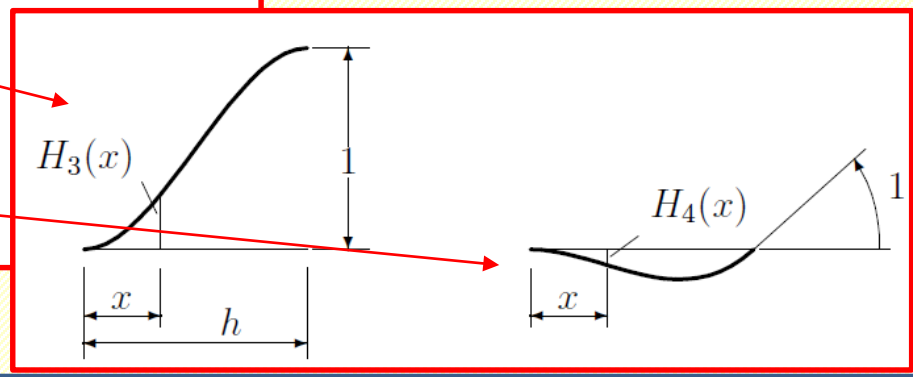
(Local FE Coordinate System)

$$H_1(x) = 1 - 3\left(\frac{x}{h}\right)^2 + 2\left(\frac{x}{h}\right)^3;$$

$$H_2(x) = h\left[\left(\frac{x}{h}\right) - 2\left(\frac{x}{h}\right)^2 + \left(\frac{x}{h}\right)^3\right];$$

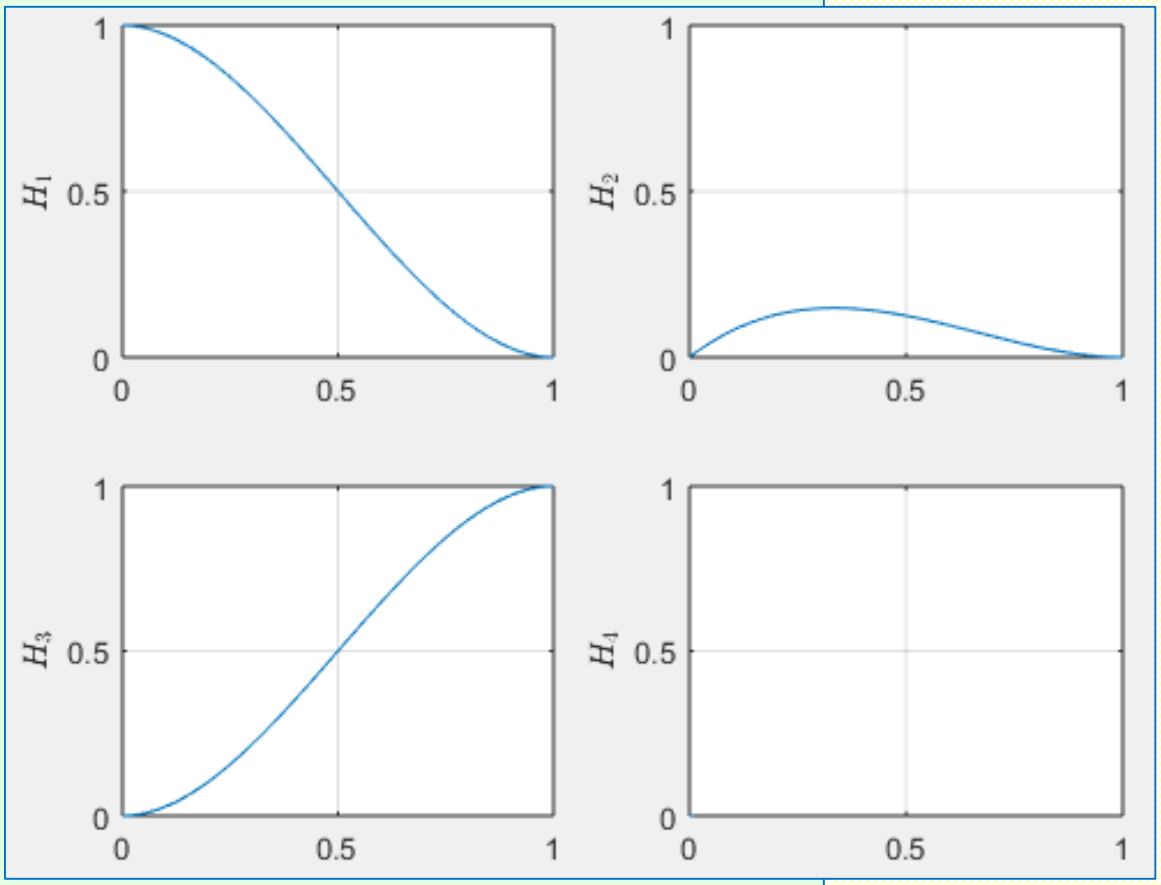
$$H_3(x) = 3\left(\frac{x}{h}\right)^2 - 2\left(\frac{x}{h}\right)^3;$$

$$H_4(x) = h\left[-\left(\frac{x}{h}\right)^2 + \left(\frac{x}{h}\right)^3\right]$$



PLOTTING SHAPE FUNCTIONS for BEAM

```
h=1; x=[0:0.01:1]*h;
H1=1-3*(x/h).^2+2*(x/h).^3;
H2=h*( (x/h) -2*(x/h).^2 + (x/h).^3);
H3= 3*(x/h).^2 - 2*(x/h).^3;
H4=h*(-(x/h).^2 + (x/h).^3);
%~~~~~H1
subplot(2,2,1);
plot(x,H1); grid on
yl1=ylabel('$H_1$')
%~~~~~H2
subplot(2,2,2);
plot(x,H2); grid on;
axis([0 1 0 1]);
yl2=ylabel('$H_2$')
%~~~~~H3
subplot(2,2,3);
plot(x,H3); grid on
yl3=ylabel('$H_3$')
%~~~~~H4
subplot(2,2,4);
plot(x,H4); grid on
yl4=ylabel('$H_4$');
axis([0 1 0 1]);

set([yl1 yl2 yl3 yl4],'Interpreter','LaTeX')
```



Historical reference: Charles HERMITE

Charles **HERMITE** (1822-1901), a French mathematician, Foreign Corresponding Member (since 1857) and Foreign Honourable Member (since 1895) of the St.Petersburg Academy of Sciences. His work in the theory of functions includes the application of elliptic functions to provide the first solution to the general equation of the fifth degree, the quintic equation.

He was appointed as professor at the École Normale, Paris in 1869. In 1970 he became professor of higher algebra at the Sorbonne.

In 1873 Hermite published the first proof that e is a transcendental number; i.e. it is not the root of any algebraic equation with rational coefficients. Hermite was a major figure in the development of the theory of algebraic forms, the arithmetical theory of quadratic forms, and the theories of elliptic and Abelian functions. He first studied the representation of integers in what are now called Hermitian forms. His famous solution of the general quintic equation appeared in 1858. Many late-19th-century mathematicians first gained recognition for their work largely through the encouragement and publicity supplied by Hermite.

In our days the Hermite polynomials are widely used in the finite element methods.

FEM: MASS & STIFFNESS MATRICES for BEAM

Consistent Mass Matrix for Translational Inertia for a 2D Beam FE:
(Local FE Coordinate System)

$$[m^e] = \frac{mh}{420}\begin{bmatrix} 156 & 22h & 54 & -13h \\ 22h & 4h^2 & 13h & -3h^2 \\ 54 & 13h & 156 & -22h \\ -13h & -3h^2 & -22h & 4h^2 \end{bmatrix}$$

Stiffness Matrix for a 2D Beam FE:
(Local FE Coordinate System)

$$[k] = \frac{EI}{h^3}\begin{bmatrix} 12 & 6h & -12 & 6h \\ 6h & 4h^2 & -6h & 2h^2 \\ -12 & -6h & 12 & -6h \\ 6h & 2h^2 & -6h & 4h^2 \end{bmatrix}$$

FEM: NODAL FORCES for BEAM

Nodal Forces for a 2D Beam FE:
(Local FE Coordinate System)

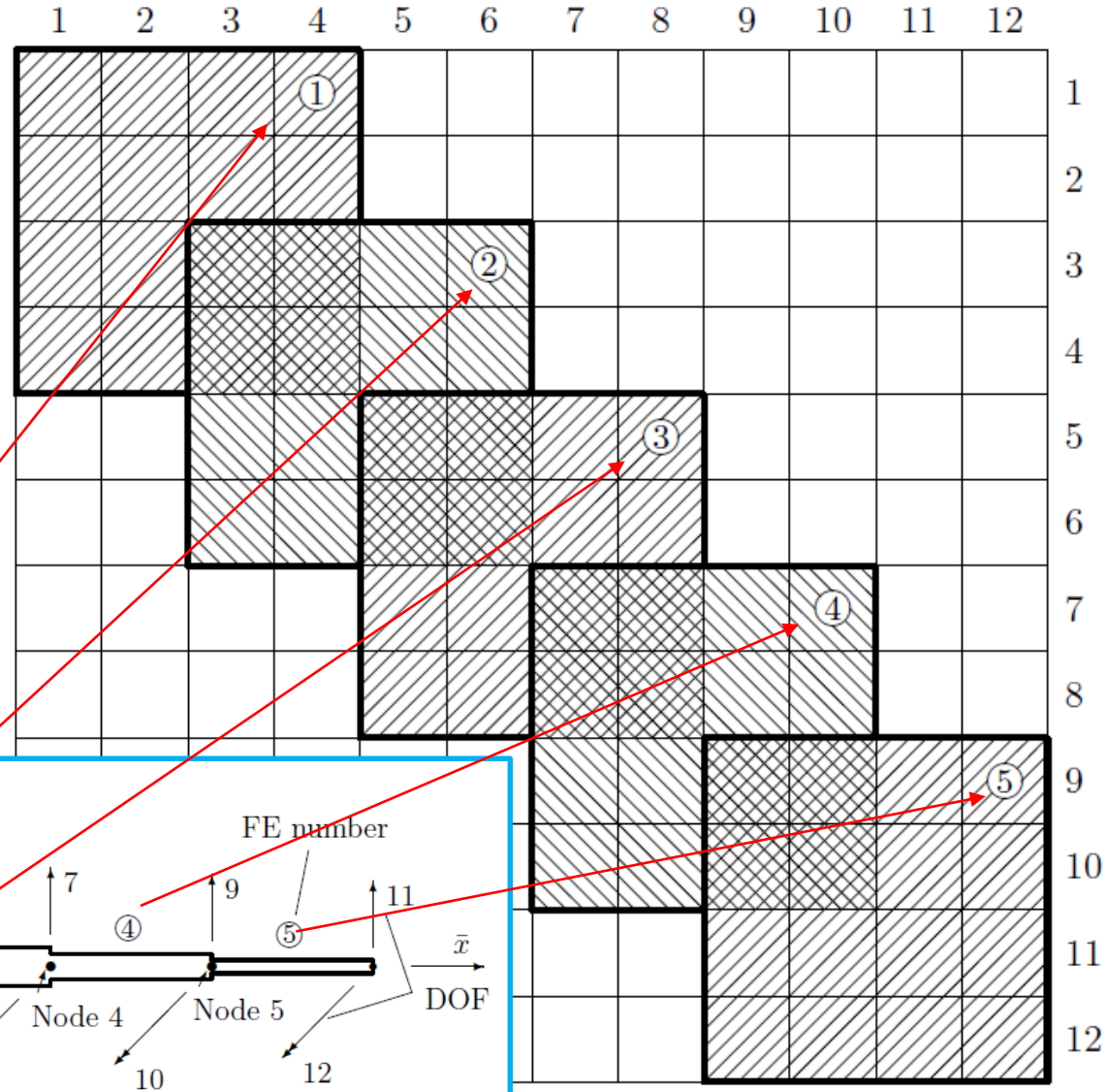
$$f_1^e(t) = \int_0^h f(x,t)\, H_1(x)\, dx = \int_0^h f(x,t) \left[1 - 3\left(\frac{x}{h}\right)^2 + 2\left(\frac{x}{h}\right)^3\right] dx$$

$$f_2^e(t) = \int_0^h f(x,t)\, H_2(x)\, dx = \int_0^h f(x,t)\, h \left[\left(\frac{x}{h}\right) - 2\left(\frac{x}{h}\right)^2 + \left(\frac{x}{h}\right)^3\right] dx$$

$$f_3^e(t) = \int_0^h f(x,t)\, H_3(x)\, dx = \int_0^h f(x,t) \left[3\left(\frac{x}{h}\right)^2 - 2\left(\frac{x}{h}\right)^3\right] dx$$

$$f_4^e(t) = \int_0^h f(x,t)\, H_4(x)\, dx = \int_0^h f(x,t)\, h \left[-\left(\frac{x}{h}\right)^2 + \left(\frac{x}{h}\right)^3\right] dx$$

ASSEMBLING GLOBAL MATRICES PROCEDURE



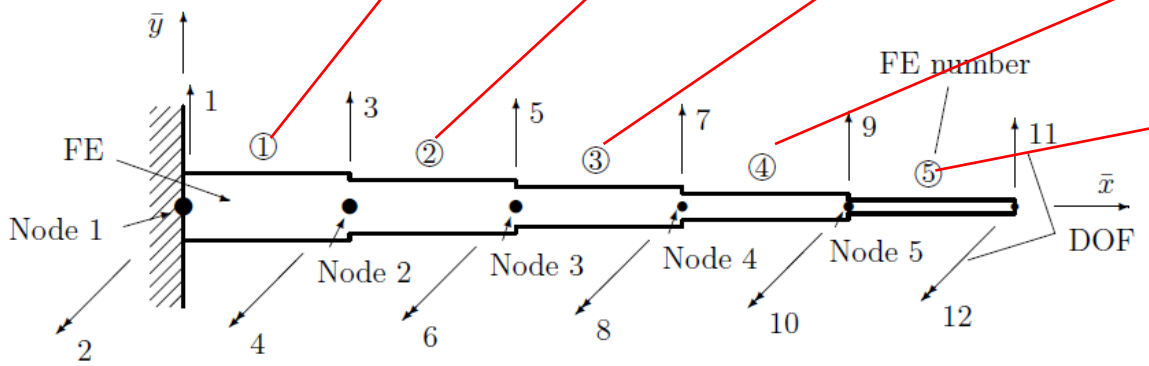


Figure 1.12: Simplified finite element model of an aircraft wing.

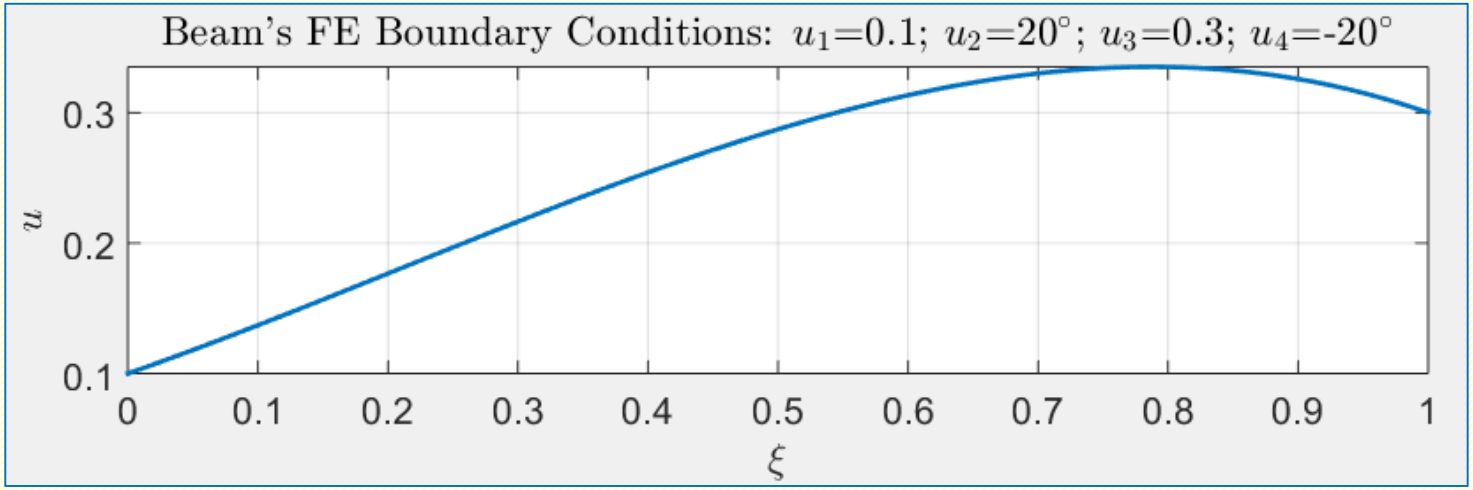
Figure 1.13: Demonstration of the assembling procedure of the global stiffness matrix.

FEM: Beam Element:

Plotting Shape of the FE with Specified Boundary Conditions (Example-1)

Shape of the FE with given Boundary Conditions

```
xi=0:0.01:1; h=1;
H1=1 - 3*xi.^2 + 2*xi.^3;     H2=  h*(xi - 2*xi.^2 + xi.^3);
H3=3*xi.^2 - 2*xi.^3;         H4= h*(-xi.^2 + xi.^3);

u1=0.1; u3=0.3; u2=20; u4=-20;
str=sprintf(['Beam''s FE Boundary Conditions: ', ...
'$u_1$=%g; $u_2$=%g$^\\circ$; $u_3$=%g; $u_4$=%g$^\\circ$'],u1,u2,u3,u4);
u2=u2*(pi/180); u4=u4*(pi/180); % Convert deg -> rad

u=u1*H1 + u2*H2 + u3*H3 + u4*H4;
figure;  plot(x,u,'LineWidth',2);   grid on;   axis tight equal;
xl=xlabel('$x$');   yl=ylabel('$u$');   ti=title(str);
set([xl yl ti],'Interpreter','LaTeX');
set(gca,'FontSize',16);   set(gcf,'Position',[120 191 949 309]);
```

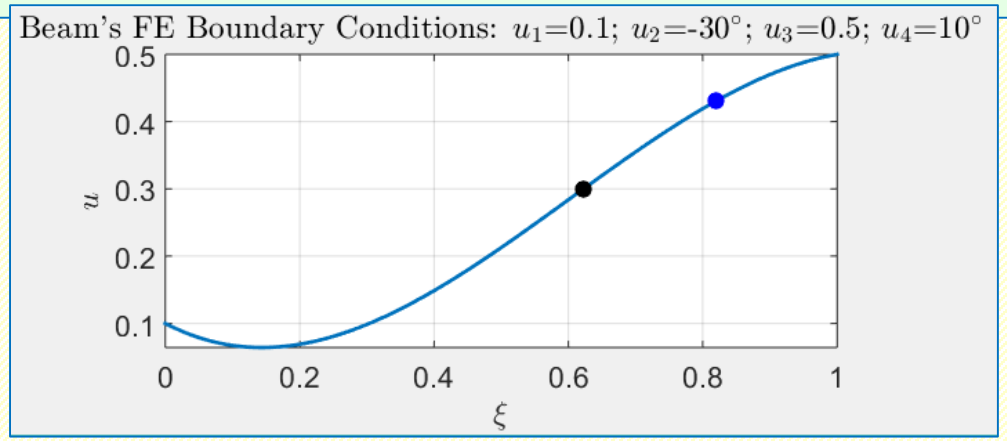


FEM: Beam Element:

Plotting Shape of the FE with Specified BCs and Interpolating *u(x)* (Example-2)

Shape of the FE with given Boundary Conditions

```
xi=0:0.01:1; h=1;
H1=1 - 3*xi.^2 + 2*xi.^3;     H2=  h*(xi - 2*xi.^2 + xi.^3);
H3=3*xi.^2 - 2*xi.^3;         H4= h*(-xi.^2 + xi.^3);

u1=0.1; u3=0.5; u2=-30; u4=10;
str=sprintf(['Beam''s FE Boundary Conditions: ', ...
'$u_1$=%g; $u_2$=%g$^\\circ$; $u_3$=%g; $u_4$=%g$^\\circ$'],u1,u2,u3,u4);
u2=u2*(pi/180); u4=u4*(pi/180); % Convert deg -> rad

u=u1*H1 + u2*H2 + u3*H3 + u4*H4;
figure;  plot(x,u,'LineWidth',2);   grid on;   axis tight equal;
xl=xlabel('$x$');   yl=ylabel('$u$');   ti=title(str);
set([xl yl ti],'Interpreter','LaTeX');
set(gca,'FontSize',16); set(gcf,'Position',[120   191   949   309]);

xiQ=0.82;
line('XData',xiQ,'YData',interp1(xi,y,xiQ),'Marker','.','MarkerSize',32,'Color',[0 0 1]);
uQ=0.3;
line('XData',interp1(y,xi,yQ),'YData',uQ,'Marker','.','MarkerSize',32,'Color',[0 0 0]);
```



FEM: Beam Element:

Assembling Global Mass & Stiffness Matrices

Example of a Beam Modelled with TWO FINITE ELEMENTS

Example: Cantilevered Beam, Modelled with 2 FEs

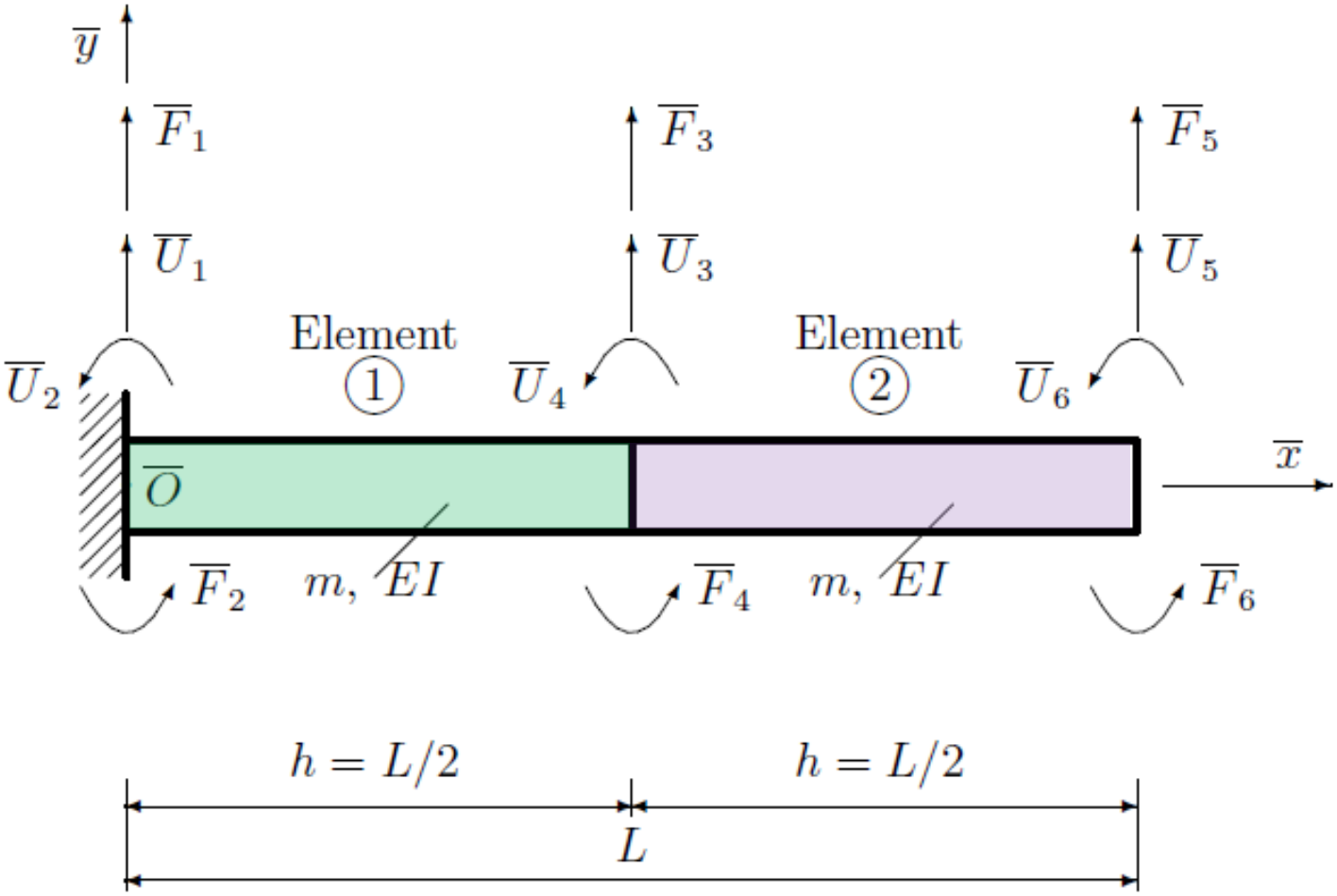


Figure 1.14: Two-finite-element model of the beam.

Example: Element Matrices, related to each FE

$$[m^e] = [\overline{m}^e] = [\overline{m}] = \frac{mh}{420}\begin{bmatrix} 156 & 22 & 54 & -13 \\ 22 & 4 & 13 & -3 \\ 54 & 13 & 156 & -22 \\ -13 & -3 & -22 & 4 \end{bmatrix}.$$

FOR ILLUSTRATION PURPOSE, *ASSUME THAT h=1*, then variable "h" will not appear in the M and K matrices and we can work with numbers only.

$$[k^e] = [\overline{k}^e] = [\overline{k}] = \frac{EI}{h^3}\begin{bmatrix} 12 & 6 & -12 & 6 \\ 6 & 4 & -6 & 2 \\ -12 & -6 & 12 & -6 \\ 6 & 2 & -6 & 4 \end{bmatrix}.$$

NOTE:

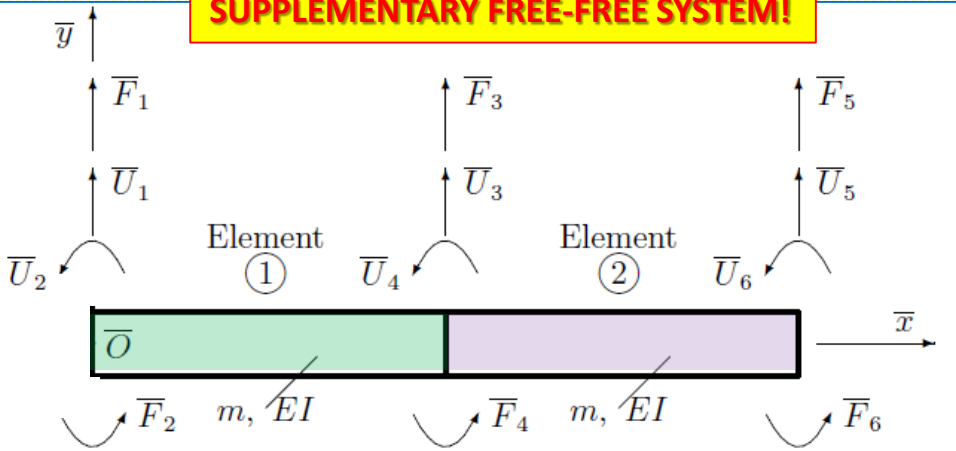
When the DOFs in the System are constrained,

we start simulation process, considering completely FREE SUPPLEMENTARY SYSTEM first;

then, when adjust its matrices to the MAIN case

Example: Assembling Global Mass Matrix Procedure

SUPPLEMENTARY FREE-FREE SYSTEM!



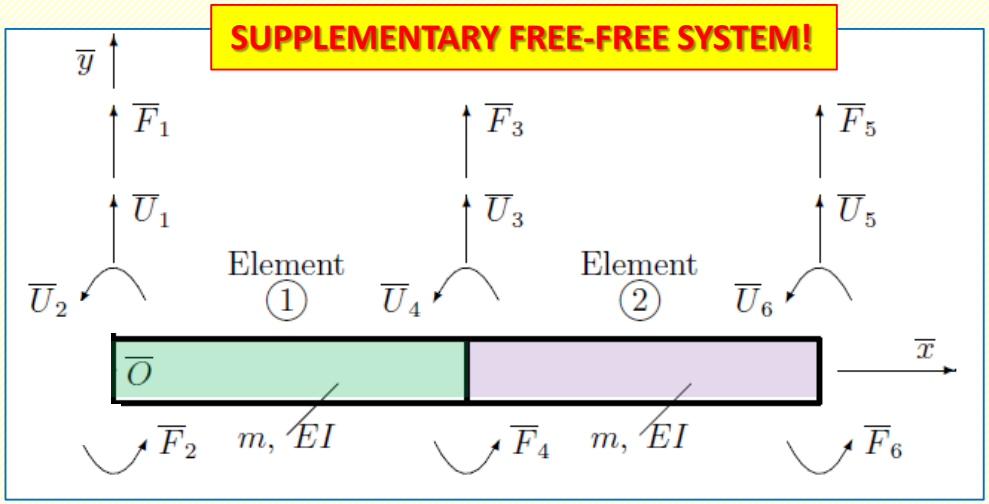
FOR ILLUSTRATION PURPOSE, *ASSUME THAT h=1*, then variable "h" will not appear in the M and K matrices and we can work with numbers only.

$$\left[\overline{M}^{①}\right] = \frac{mh}{420}\begin{bmatrix} 156 & 22 & 54 & -13 & 0 & 0 \\ 22 & 4 & 13 & -3 & 0 & 0 \\ 54 & 13 & 156 & -22 & 0 & 0 \\ -13 & -3 & -22 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\left[\overline{M}^{②}\right] = \frac{mh}{420}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 156 & 22 & 54 & -13 \\ 0 & 0 & 22 & 4 & 13 & -3 \\ 0 & 0 & 54 & 13 & 156 & -22 \\ 0 & 0 & -13 & -3 & -22 & 4 \end{bmatrix}$$

$$\left[\overline{M}\right] = \sum_{e=1}^{2}\left[\overline{M}^{e}\right] = \frac{mh}{420}\begin{bmatrix} 156 & 22 & 54 & -13 & 0 & 0 \\ 22 & 4 & 13 & -3 & 0 & 0 \\ 54 & 13 & 312 & 0 & 54 & -13 \\ -13 & -3 & 0 & 8 & 13 & -3 \\ 0 & 0 & 54 & 13 & 156 & -22 \\ 0 & 0 & -13 & -3 & -22 & 4 \end{bmatrix}$$

Example: Assembling Global Mass Matrix Procedure



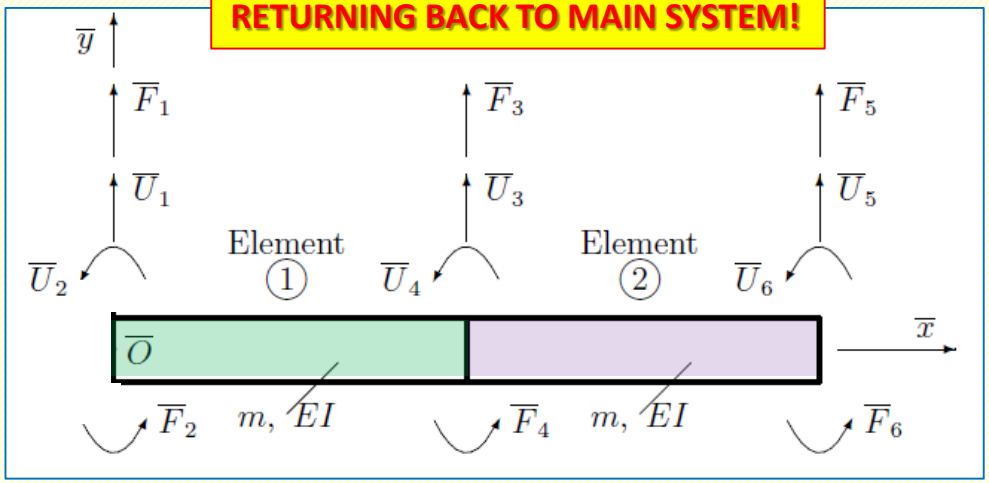
$$\left[\overline{K}^{①}\right] = \frac{EI}{h^3}\begin{bmatrix} 12 & 6 & -12 & 6 & 0 & 0 \\ 6 & 4 & -6 & 2 & 0 & 0 \\ -12 & -6 & 12 & -6 & 0 & 0 \\ 6 & 2 & -6 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\left[\overline{K}^{②}\right] = \frac{EI}{h^3}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 12 & 6 & -12 & 6 \\ 0 & 0 & 6 & 4 & -6 & 2 \\ 0 & 0 & -12 & -6 & 12 & -6 \\ 0 & 0 & 6 & 2 & -6 & 4 \end{bmatrix}$$

$$\left[\overline{K}\right] = \sum_{e=1}^{2}\left[\overline{K}^{e}\right] = \frac{EI}{h^3}\begin{bmatrix} 12 & 6 & -12 & 6 & 0 & 0 \\ 6 & 4 & -6 & 2 & 0 & 0 \\ -12 & -6 & 24 & 0 & -12 & 6 \\ 6 & 2 & 0 & 8 & -6 & 2 \\ 0 & 0 & -12 & -6 & 12 & -6 \\ 0 & 0 & 6 & 2 & -6 & 4 \end{bmatrix}$$

Example: Assembling Global Mass Matrix Procedure

RETURNING BACK TO MAIN SYSTEM!

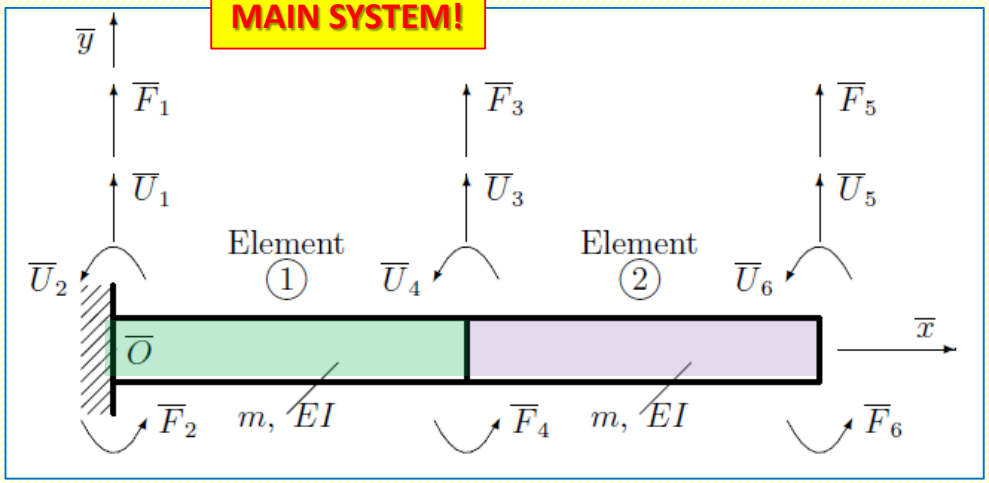


$$\left[\overline{M}^{①}\right] = \frac{mh}{420}\begin{bmatrix} 156 & 22 & 54 & -13 & 0 & 0 \\ 22 & 4 & 13 & -3 & 0 & 0 \\ 54 & 13 & 156 & -22 & 0 & 0 \\ -13 & -3 & -22 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\left[\overline{M}^{②}\right] = \frac{mh}{420}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 156 & 22 & 54 & -13 \\ 0 & 0 & 22 & 4 & 13 & -3 \\ 0 & 0 & 54 & 13 & 156 & -22 \\ 0 & 0 & -13 & -3 & -22 & 4 \end{bmatrix}$$

$$\left[\overline{M}\right] = \sum_{e=1}^{2}\left[\overline{M}^{e}\right] = \frac{mh}{420}\begin{bmatrix} 156 & 22 & 54 & -13 & 0 & 0 \\ 22 & 4 & 13 & -3 & 0 & 0 \\ 54 & 13 & 312 & 0 & 54 & -13 \\ -13 & -3 & 0 & 8 & 13 & -3 \\ 0 & 0 & 54 & 13 & 156 & -22 \\ 0 & 0 & -13 & -3 & -22 & 4 \end{bmatrix}$$

Example: Assembling Global Mass Matrix Procedure



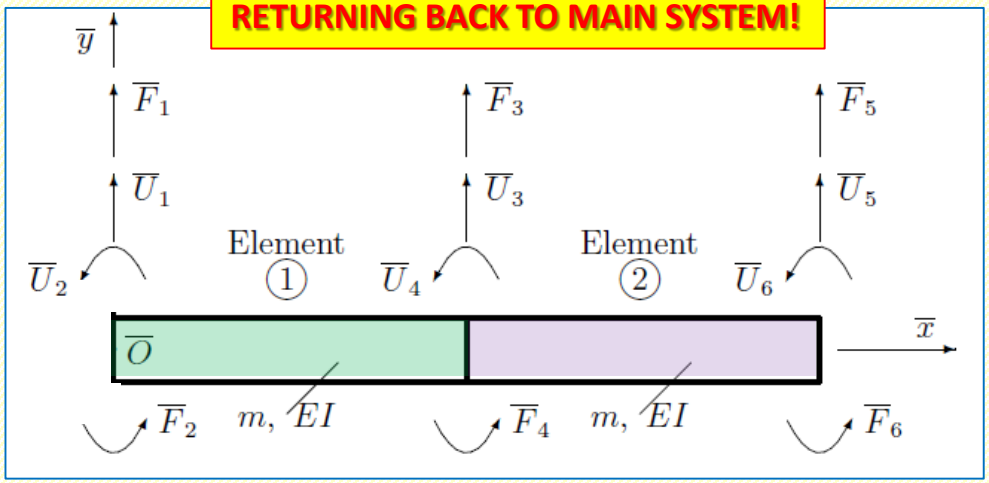
$$[\overline{M}^{①}] = \frac{mh}{420}\begin{bmatrix} 156 & 22 & 54 & -13 & 0 & 0 \\ 22 & 4 & 13 & -3 & 0 & 0 \\ 54 & 13 & 156 & -22 & 0 & 0 \\ -13 & -3 & -22 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$[\overline{M}^{②}] = \frac{mh}{420}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 156 & 22 & 54 & -13 \\ 0 & 0 & 22 & 4 & 13 & -3 \\ 0 & 0 & 54 & 13 & 156 & -22 \\ 0 & 0 & -13 & -3 & -22 & 4 \end{bmatrix}$$

$$[M] = \frac{mh}{420}\begin{bmatrix} 312 & 0 & 54 & -13 \\ 0 & 8 & 13 & -3 \\ 54 & 13 & 156 & -22 \\ -13 & -3 & -22 & 4 \end{bmatrix}$$

Example: Assembling Global Mass Matrix Procedure

RETURNING BACK TO MAIN SYSTEM!

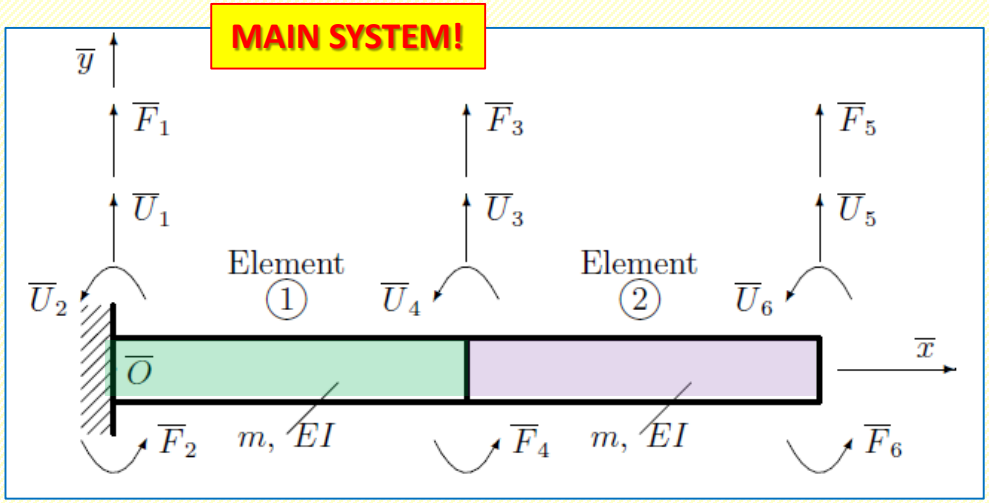


$$\left[\overline{K}^{①}\right] = \frac{EI}{h^3}\begin{bmatrix} 12 & 6 & -12 & 6 & 0 & 0 \\ 6 & 4 & -6 & 2 & 0 & 0 \\ -12 & -6 & 12 & -6 & 0 & 0 \\ 6 & 2 & -6 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\left[\overline{K}^{②}\right] = \frac{EI}{h^3}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 12 & 6 & -12 & 6 \\ 0 & 0 & 6 & 4 & -6 & 2 \\ 0 & 0 & -12 & -6 & 12 & -6 \\ 0 & 0 & 6 & 2 & -6 & 4 \end{bmatrix}$$

$$\left[\overline{K}\right] = \sum_{e=1}^{2}\left[\overline{K}^{e}\right] = \frac{EI}{h^3}\begin{bmatrix} 12 & 6 & -12 & 6 & 0 & 0 \\ 6 & 4 & -6 & 2 & 0 & 0 \\ -12 & -6 & 24 & 0 & -12 & 6 \\ 6 & 2 & 0 & 8 & -6 & 2 \\ 0 & 0 & -12 & -6 & 12 & -6 \\ 0 & 0 & 6 & 2 & -6 & 4 \end{bmatrix}$$

Example: Assembling Global Mass Matrix Procedure



$$[\overline{K}^{①}] = \frac{EI}{h^3}\begin{bmatrix} 12 & 6 & -12 & 6 & 0 & 0 \\ 6 & 4 & -6 & 2 & 0 & 0 \\ -12 & -6 & 12 & -6 & 0 & 0 \\ 6 & 2 & -6 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$[\overline{K}^{②}] = \frac{EI}{h^3}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 12 & 6 & -12 & 6 \\ 0 & 0 & 6 & 4 & -6 & 2 \\ 0 & 0 & -12 & -6 & 12 & -6 \\ 0 & 0 & 6 & 2 & -6 & 4 \end{bmatrix}$$

$$[K] = \frac{EI}{h^3}\begin{bmatrix} 24 & 0 & -12 & 6 \\ 0 & 8 & -6 & 2 \\ -12 & -6 & 12 & -6 \\ 6 & 2 & -6 & 4 \end{bmatrix}$$

APPENDIX-2: Further Reading for Interested Students

REFERENCE: **Trivailo P.M.** (2008), Vibrations: Theory & Aerospace Applications, Vol.1&2, - (The textbook for senior undergraduate and graduate aerospace students). - Melbourne: RMIT Publisher - 2008. - 247pp +348pp=595 pp., 355 ill, 4 software programs.

8.6 FREE VIBRATION. EIGENVALUE PROBLEM

Let us consider the vibrating rod. In the case of free vibration, namely, when the distributed force is zero, $f(x,t) = 0$, the boundary-value problem reduces to the differential equation:

$$\frac{\partial}{\partial x}\left[E\mathcal{A}(x)\frac{\partial u(x,t)}{\partial x}\right] = m(x)\frac{\partial^2 u(x,t)}{\partial t^2}, \qquad 0 < x < L. \tag{8.6}$$

The solution of (8.6) is assumed in the form:

$$u(x,t) = U(x) \cdot F(t),$$

where $U(x)$ represents the general rod deformation and depends on the spatial variable x alone, and where $F(t)$ indicates the type of motion of rod configuration executes with time and depends on t alone.

$$\frac{1}{m(x)\,U(x)}\frac{d}{dx}\left[E\mathcal{A}(x)\frac{dU(x)}{dx}\right] = \frac{1}{F(t)}\frac{d^2F(t)}{dt^2}.$$

Since the left side of this equation is independent of t, whereas the right side is independent of x, it follows that each side must be a constant.

$$\frac{d^2F(t)}{dt^2} + \omega^2 F(t) = 0, \tag{8.7}$$

$$-\frac{d}{dx}\left[E\mathcal{A}(x)\frac{dU(x)}{dx}\right] = \omega^2 m(x)\,U(x), \qquad 0 < x < L. \tag{8.8}$$

The problem of determining the values of the parameter ω^2 for which nontrivial solutions $U(x)$ exist, where the solutions are subject to *boundary conditions*, is called the characteristic-value, or *eigenvalue problem* (from German *eigen*, characteristic).

Examples of boundary conditions are:

- $u(0,t) = u(L,t) = 0$ clamped-clamped rod
- $u'(0,t) = u'(L,t) = 0$ free-free rod

The differential equation (8.8) possesses space-dependent coefficients, so that in general no closed-form solution can be expected. A closed-form solution can be obtained in the special case of a *uniform rod* with $m(x) = m = const$, $E\mathcal{A}(x) = E\mathcal{A} = const$. Considering that case, Eq.(8.8) reduces to

$$\frac{d^2U(x)}{dx^2} + \beta^2\,U(x) = 0, \qquad \beta^2 = \omega^2\frac{m}{E\mathcal{A}} = \left(\frac{\omega}{c}\right)^2, \tag{8.9}$$

which must be satisfied over the domain $0 < x < L$.

The solution of the equation (8.9) has the form

$$U(x) = \left(A\sin\frac{\omega}{c}x + B\cos\frac{\omega}{c}x\right) \tag{8.10}$$

where the arbitrary constants A, B depend on the *boundary conditions*.

The general solution of the Eq.(8.7) is

$$F(t) = (C\sin\omega t + D\cos\omega t), \tag{8.11}$$

where the arbitrary constants C, D depend on the *initial conditions*.

Combining Eq.(8.10) and (8.11) we can write the general solution for $u(x,t)$ in the following form

$$u(x,t) = \left(A\sin\frac{\omega}{c}x + B\cos\frac{\omega}{c}x\right) \times (C\sin\omega t + D\cos\omega t). \tag{8.12}$$

8.7 TAKING INTO ACCOUNT BOUNDARY CONDITIONS

8.7.1 Example: Longitudinal Vibration of a Free-Free Uniform Rod

Solve the eigenvalue problem associated with a uniform rod, vibrating longitudinally, with both ends free (*see* Fig. 8.8).

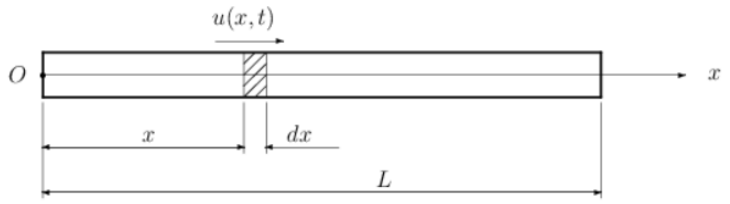


Figure 8.8: A free-free uniform rod in longitudinal vibration.

Solution

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2},$$

$$u(x,t) = \left(A \sin\frac{\omega}{c}x + B\cos\frac{\omega}{c}x\right) \times (C\sin\omega t + D\cos\omega t).$$

The arbitrary constants A, B, C, D depend on the *boundary conditions* and the *initial conditions*.

Since the bar has free ends, *the axial force*, which is proportional to dU/dx, *must be zero at each extremity*. Thus the boundary conditions for this problem may be written as

$$EA\left.\frac{\partial u(x,t)}{\partial x}\right|_{x=0} = 0, \qquad EA\left.\frac{\partial u(x,t)}{\partial x}\right|_{x=L} = 0.$$

The first boundary condition will require that $A = 0$, so

$$u(x,t) = B\cos\frac{\omega}{c}x\,(C\sin\omega t + D\cos\omega t).$$

The second boundary condition then leads to the *characteristic equation*:

$$\sin\frac{\omega L}{c} = 0, \quad \text{or} \quad \frac{\omega_r L}{c} = r\pi, \quad r = 1,2,3,\ldots,$$

to which corresponds the infinite set of *eigenfunctions*:

$$U_r(x) = B_r\cos\frac{r\pi x}{L}.$$

The first natural modes are plotted in Figure 8.9, where the modes have been normalized by letting $B_r = 1$. We note that the first mode has one node, the second has two nodes and the third has three nodes. In general the r-th mode has r nodes $(r = 1, 2, \ldots)$.

The system natural frequencies are:

$$\omega_r = \frac{r\pi c}{L} = r\pi\sqrt{\frac{E}{\rho L^2}}, \qquad r = 1,2,3,\ldots.$$

FREE-FREE ROD:

In the more general case of free vibration initiated in any manner, the solution will contain many of the normal modes:

$$u(x,t) = \sum_{r=1}^{\infty}\cos\frac{r\pi x}{L}\,(C_r\sin\omega_r t + D_r\cos\omega_r t)$$

$$\omega_r = \frac{r\pi c}{L}, \quad r = 1,2,3,\ldots.$$

The arbitrary constants C_r, D_r depend on the *initial conditions*.

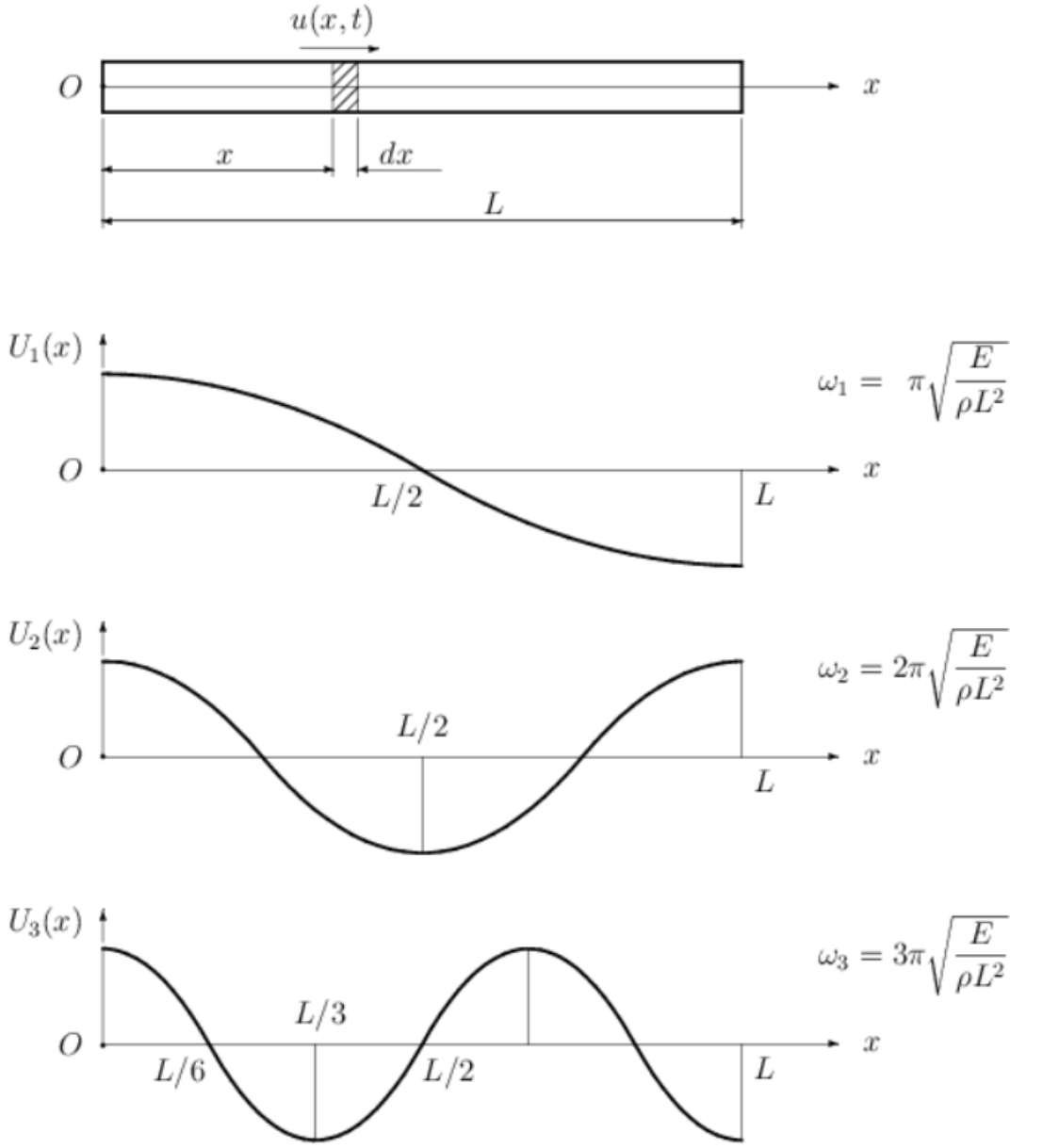


Figure 8.9: The first three natural modes of longitudinal vibration for a free-free bar.

8.7.2 Example: Axial Vibration of a Clamped-Free Uniform Rod

Derive an expression for the free longitudinal vibration of a uniform bar of length L, one end of which is fixed and the other end free (it see Fig. 8.10).

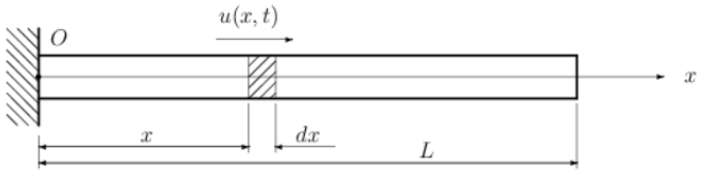


Figure 8.10: A clamped-free rod in longitudinal vibration.

Solution

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2},$$

The general solution for the free longitudinal vibration of uniform bars is given as

$$u(x,t) = \left(A \sin\frac{\omega}{c}x + B\cos\frac{\omega}{c}x\right) \times (C\sin\omega t + D\cos\omega t).$$

The arbitrary constants A, B, C, D depend on the *boundary conditions* and the *initial conditions*.

The *tensile force at the free end of this bar is equal to zero* while the *displacement of the fixed end of the bar is also equal to zero*; i.e. the boundary conditions of the problem are:

$$\begin{aligned} EA\left.\frac{\partial u(x,t)}{\partial x}\right|_{x=L} &= 0, \\ \left.u(x,t)\right|_{x=0} &= 0. \end{aligned}$$

The condition that $u(0,t) = 0$ will require that $B = 0$, so

$$u(x,t) = A\sin\frac{\omega}{c}x\,(C\sin\omega t + D\cos\omega t).$$

The condition $\frac{\partial u(L,t)}{\partial x} = 0$ then leads to the *characteristic equation*:

$$\cos\frac{\omega L}{c} = 0, \quad \text{or} \quad \frac{\omega_r L}{c} = \frac{(2r-1)\pi}{2}, \quad r = 1, 2, 3, \ldots,$$

to which corresponds the infinite set of *eigenfunctions*:

$$U_r(x) = A_r \sin\frac{(2r-1)\pi x}{2L}.$$

We note that the first mode, presented in Fig. 8.11, has no nodes, the second has one node and the third has two nodes. In general the r-th mode has $r-1$ nodes $(r = 1, 2, \ldots)$.

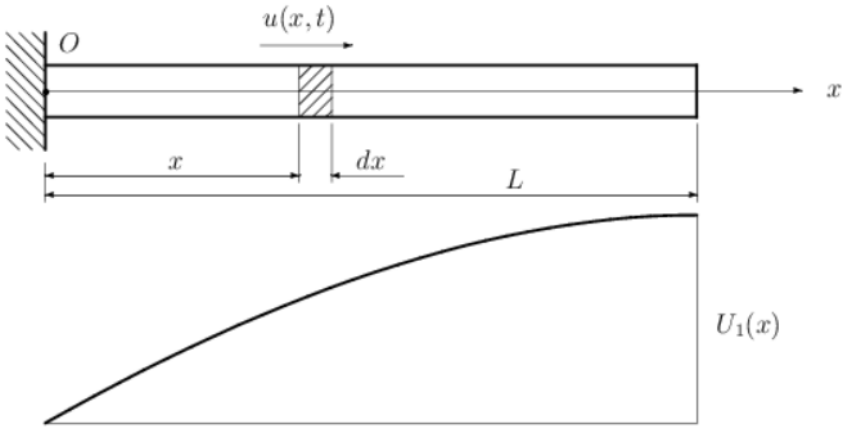


Figure 8.11: The fundamental mode of vibration of a clamped-free rod.

CLAMPED-FREE ROD:

The system natural frequencies are:

$$\omega_r = \frac{(2r-1)\pi c}{2L} = \frac{(2r-1)\pi}{2}\sqrt{\frac{E}{\rho L^2}}, \qquad r = 1, 2, 3, \ldots.$$

In the more general case of free vibration initiated in any manner, the solution will contain many of the normal modes:

$$\begin{aligned} u(x,t) &= \sum_{r=1}^{\infty} \sin\frac{(2r-1)\pi x}{2L}\,(C_r \sin\omega_r t + D_r \cos\omega_r t) \\ \omega_r &= \frac{(2r-1)\pi c}{2L}, \quad r = 1, 2, 3, \ldots. \end{aligned} \tag{8.13}$$

The arbitrary constants C_r, D_r depend on the *initial conditions*.

Stiffness Matrix for a 3D Frame Finite Element:
(FE Local Coordinate System)

$$
[k^e] = \left[\begin{array}{cccccc|cccccc}
\frac{EA}{h} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{h} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{12EI_z}{h^3} & 0 & 0 & 0 & \frac{6EI_z}{h^2} & 0 & -\frac{12EI_z}{h^3} & 0 & 0 & 0 & \frac{6EI_z}{h^2} \\
0 & 0 & \frac{12EI_y}{h^3} & 0 & -\frac{6EI_y}{h^2} & 0 & 0 & 0 & -\frac{12EI_y}{h^3} & 0 & -\frac{6EI_y}{h^2} & 0 \\
0 & 0 & 0 & \frac{GJ}{h} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ}{h} & 0 & 0 \\
0 & 0 & -\frac{6EI_y}{h^2} & 0 & \frac{4EI_y}{h} & 0 & 0 & 0 & \frac{6EI_y}{h^2} & 0 & \frac{2EI_y}{h} & 0 \\
0 & \frac{6EI_z}{h^2} & 0 & 0 & 0 & \frac{4EI_z}{h} & 0 & -\frac{6EI_z}{h^2} & 0 & 0 & 0 & \frac{2EI_z}{h} \\
\hline
-\frac{EA}{h} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{h} & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{12EI_z}{h^3} & 0 & 0 & 0 & -\frac{6EI_z}{h^2} & 0 & \frac{12EI_z}{h^3} & 0 & 0 & 0 & -\frac{6EI_z}{h^2} \\
0 & 0 & -\frac{12EI_y}{h^3} & 0 & \frac{6EI_y}{h^2} & 0 & 0 & 0 & \frac{12EI_y}{h^3} & 0 & \frac{6EI_y}{h^2} & 0 \\
0 & 0 & 0 & -\frac{GJ}{h} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{h} & 0 & 0 \\
0 & 0 & -\frac{6EI_y}{h^2} & 0 & \frac{2EI_y}{h} & 0 & 0 & 0 & \frac{6EI_y}{h^2} & 0 & \frac{4EI_y}{h} & 0 \\
0 & \frac{6EI_z}{h^2} & 0 & 0 & 0 & \frac{2EI_z}{h} & 0 & -\frac{6EI_z}{h^2} & 0 & 0 & 0 & \frac{4EI_z}{h}
\end{array}\right]
$$

Consistent Mass Matrix for a 3D Frame Finite Element:
(FE Local Coordinate System)

$$[m^e] = \frac{mh}{420}\left[\begin{array}{cccccc|cccccc}
140 & 0 & 0 & 0 & 0 & 0 & 70 & 0 & 0 & 0 & 0 & 0 \\
0 & 156 & 0 & 0 & 0 & 22h & 0 & 54 & 0 & 0 & 0 & -13h \\
0 & 0 & 156 & 0 & -22h & 0 & 0 & 0 & 54 & 0 & 13h & 0 \\
0 & 0 & 0 & \frac{140\,I_p}{\mathcal{A}} & 0 & 0 & 0 & 0 & 0 & \frac{70\,I_p}{\mathcal{A}} & 0 & 0 \\
0 & 0 & -22h & 0 & 4h^2 & 0 & 0 & 0 & -13h & 0 & -3h^2 & 0 \\
0 & 22h & 0 & 0 & 0 & 4h^2 & 0 & 13h & 0 & 0 & 0 & -3h^2 \\
\hline
70 & 0 & 0 & 0 & 0 & 0 & 140 & 0 & 0 & 0 & 0 & 0 \\
0 & 54 & 0 & 0 & 0 & 13h & 0 & 156 & 0 & 0 & 0 & -22h \\
0 & 0 & 54 & 0 & -13h & 0 & 0 & 0 & 156 & 0 & 22h & 0 \\
0 & 0 & 0 & \frac{70\,I_p}{\mathcal{A}} & 0 & 0 & 0 & 0 & 0 & \frac{140\,I_p}{\mathcal{A}} & 0 & 0 \\
0 & 0 & 13h & 0 & -3h^2 & 0 & 0 & 0 & 22h & 0 & 4h^2 & 0 \\
0 & -13h & 0 & 0 & 0 & -3h^2 & 0 & -22h & 0 & 0 & 0 & 4h^2
\end{array}\right]$$

END OF LECTURE-3b SLIDES

END OF LECTURES
by Prof P.M.Trivailo

Thank you for your interest, hard work and participation!

LINK