

Mechanics with Matlab

M. Okrouhlík, Institute of Thermomechanics, Dolejskova 5, 182 00 Prague

Abstract

A brief information about an ongoing project of the European Union under the Leonardo da Vinci Programme - dedicated to the creation of the electronic learning tool aimed to the vocational training of engineering mechanics using Matlab - is presented. The activities of the team of experts from Czech Republic, Germany, Belgium and Sweden are resulting in the creation of an 'electronic book' that will be soon available at a web page address. An example from the text devoted to mechanical vibration is given.

Introduction

Although the basic laws of mechanics are known since Galileo and Newton, there are many novel concepts available today in applications of laws of mechanics to complicated problems of engineering practice. At the same time the increasing availability of computers in engineering community is not always fully accompanied by the proper knowledge of numerical and programming techniques needed to employ the mechanical principles efficiently.

To improve the state of affairs the authors of the project employed Matlab as a tool that efficiently deals with matrix algebra mathematics and has a very high level of programming primitives. As such it is almost predestined to be an excellent instrument for learning how to implement mechanical and numerical principles needed for mastering efficient approaches to a wide class of mechanical problems in engineering.

Authors tried to present each topic, accompanied by many examples and worked-out studies, in a more or less self-contained form. The reader, however, is assumed to have an elementary knowledge of mechanics, calculus, differential equations and Matlab programming. The Matlab programs, accompanying the text, are used not only for the computation itself but also for reading. They are not written in the most possible efficient way, the emphasis is given to the explanation of mechanical and programming principles.

An example taken from the Vibration chapter is as follows

Example – determine the natural frequencies of a thin cantilever beam using the finite element method. Compare the results with those obtained analytically, treating the beam as a continuous system, in the paragraph dealing with analytical approach.

A cantilever beam is sketched in Fig. 1. Its left end is free, while the right one is clamped. The length of the beam is $L = 1\text{m}$. Assuming a rectangular cross sectional area with the thickness $b = 0.01\text{ m}$ and the height $h = 0.1\text{ m}$, we have the cross section $A = bh$ and the planar moment of inertia of the cross section $I = bh^3 / 12$. The Young's modulus is $E = 2.1 \cdot 10^{11}\text{ N/m}^2$ and the density is $\rho = 7800\text{ kg/m}^3$. The number of elements is set to 10, initially.

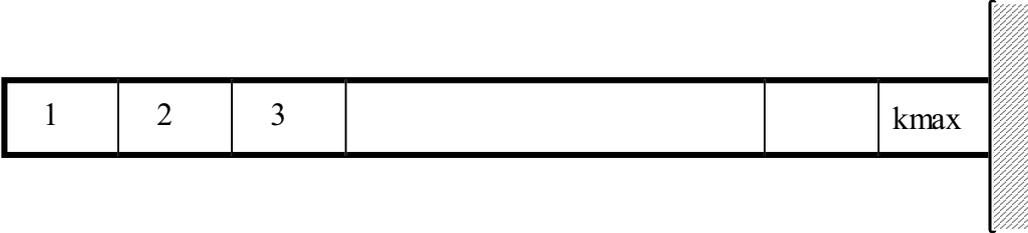


Fig. 1
A cantilever beam discretized by beam elements

For the computation we will employ a four-degree-of-freedom beam element depicted with numbering of its local degrees of freedom in Fig. 2. When modelling the beam we will assume that all the elements are of the same geometry and of material properties.

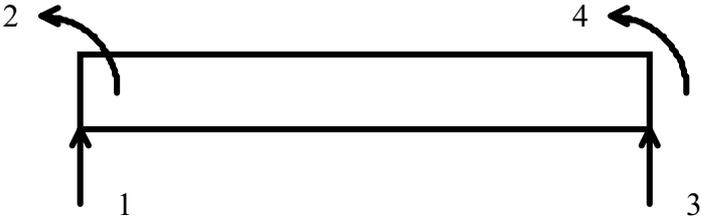


Fig. 2
Planar beam element with four degrees of freedom

Generalized displacements numbered 1 and 3 are transversal displacements of the beam element, those numbered 2 and 4 are rotations of the deflection curve. The length of the element is l .

The mass and stiffness matrices of the element are derived in numerous finite element references as in [1], [2] and [3]. This element corresponds to Bernoulli-Euler theory for thin beams and thus cannot be subjected to an axial force. The consistent mass matrix and the stiffness matrix are as follows.

$$\mathbf{m} = \frac{\rho l A}{420} \begin{bmatrix} 156 & 22l & 54 & -13l \\ & 4l^2 & 13l & -3l^2 \\ & & 156 & -22l^2 \\ \text{sym.} & & & 4l^2 \end{bmatrix}, \quad (1)$$

$$\mathbf{k} = \frac{2EI}{l^3} \begin{bmatrix} 6 & 3l & -6 & 3l \\ & 2l^2 & -3l & l^2 \\ & & 6 & -3l \\ \text{sym.} & & & 2l^2 \end{bmatrix}. \quad (2)$$

The Matlab form of the mass and stiffness matrices could have the following appearance

```
function [xme]=VBEmasbeam(ro,l1,A);
% assemble local mass matrix of a beam
% element with 4 dof's
% ro ... density
% l1 ... element's lenght
% A .... cross sectional area
xme=zeros(4);
l12=l1*l1;
konst=ro*l1*A/420;
xme(1,1)=156; xme(1,2)=22*l1; xme(1,3)=54; xme(1,4)=-13*l1;
xme(2,2)=4*l12; xme(2,3)=13*l1; xme(2,4)=-3*l12;
xme(3,3)=156; xme(3,4)=-22*l1;
xme(4,4)= 4*l12;

% symmetry
for i=1:4,
    for j=i:4,
        xme(j,i)=xme(i,j);
    end;
end;
% constant multiplication
xme=konst*xme;
% end of VBEmasbeam.m

function [xke]=VBErigbeam(ey,l1,jp);
% assemble a local stiffness matrix
% of a beam element with 4 dof's
% ey ... Young's modulus
```

```

% l1 ... element's length
% jp ... planar moment of inertia of the cross section
xke=zeros(4);
l12=l1*l1; l13=l12*l1;
konst=2*ey*jp/l13;
xke(1,1)=6; xke(1,2)=3*l1; xke(1,3)=-6; xke(1,4)= 3*l1;
           xke(2,2)=2*l12; xke(2,3)=-3*l1; xke(2,4)= l12;
           xke(3,3)= 6; xke(3,4)=-3*l1;
           xke(4,4)= 2*l12;

% symmetry
for i=1:4,
    for j=i:4,
        xke(j,i)=xke(i,j);
    end;
end;
% constant multiplication
xke=konst*xke;
% end of VBErigbeam

```

Assumed local element numbering in Fig. 2. and consecutive numbering of global dof's from left end of the beam to the right, results in following code number set which is listed bellow

element	code numbers			
1	1	2	3	4
2	3	4	5	6
3	5	6	7	8
...				
k	2k-1	2k	2k+1	2k+2
...				

Generation of code numbers and the assembling of global mass and stiffness matrices are implemented in the procedure `VBEglobeam.m`.

```

function [xk,xm]=VBEglobeam(kmax,imax,xke,xme);
% assemble global stiffness and mass matrices
% for a thin cantilever beam assembled of kmax
% identical beam elements with 4 DOF
% clear arrays
xk=zeros(imax); xm=zeros(imax);
% loop over elements
for k=1:kmax
% code numbers of k-the element
k2=2*k;
ic=[k2-1 k2 k2+1 k2+2];
% assembling
xm(ic,ic)=xm(ic,ic)+xme;
xk(ic,ic)=xk(ic,ic)+xke;
end
% end of VBEglobeam.m

```

The main program `VBEbefre2.m` calls the above procedures, implements boundary conditions and computes all eigenvalues of the cantilever beam by finite element methods and

compares the results with those obtained analytically by `VCRroots.m` program. Notice, that the number of elements, `kmax`, is a variable that can be easily changed.

```

% VBEbefre2.m
clear
% calculate eigenfrequencies of a thin cantilever beam
% composed of kmax beam elements
% input data
kmax=10;      % number of elements
imax=2*kmax+2; % number of global dof's
L=1;         % beam's length in [m]
l1=L/kmax;   % element's length
ey=2.1e11;   % Young's modulus in [N/m^2]
% a rectangular cross section of a beam
b=0.01;      % thickness
h=0.1;       % height
A=b*h;       % cross sectional area in [m^2]
jp=h*h*h*b/12; % planar mom. of inertia in [m^4]
ro=7800;     % density in [kg/m^3]

% analytically calculated values computed by VCRroots.m
% and expressed in xbar(i) = lambda(i)*L variables
% were copied here
xbar= [1.875104068706770e+000 ...
      4.694091132933031e+000 ...
      7.854757438070675e+000 ...
      1.099554073487850e+001 ...
      1.413716839104652e+001 ...
      1.727875953327674e+001 ...
      2.042035225104147e+001 ...
      2.356194490180130e+001 ...
      2.670353755550106e+001 ...
      2.984513020909291e+001 ...
      3.298672286269287e+001 ...
      3.612831551628358e+001 ...
      3.926990818348858e+001 ...
      4.241150082346221e+001 ...
      4.555309347705200e+001 ...
      4.869468613064180e+001 ...
      5.183627878423159e+001 ...
      5.497787143782138e+001 ...
      5.811946409141117e+001 ...
      6.126105674500097e+001 ...
      6.440264939859077e+001 ...
      6.754424205218055e+001 ];
% auxiliary constants
c0=sqrt(ey/ro); j=sqrt(jp/A);
b2=xbar.*xbar;
% analytically calculated angular frequencies [rad/s]
om=b2*c0*j/(L^2);
%
% assemble local stiffness matrix
[xke]=VBErigbeam(ey,l1,jp);
% assemble local mass matrix
[xme]=VBEmasbeam(ro,l1,A);
% assemble global matrices
[xk,xm]=VBEglobeam(kmax,imax,xke,xme);
%
% boundary conditions - the right hand side is clamped

```

```

% delete the last two rows and columns of global matrices
% pointer to deleted dof's
bound = [imax-1 imax];
xk(bound,:)=[]; xk(:,bound)=[];
xm(bound,:)=[]; xm(:,bound)=[];
%
% calculate eigenvalues by the finite element formulation
ei=eig(xk,xm); % eigenvalues
% sorted natural angular frequencies [rad/s]
ef=sort(real(sqrt(ei)));
% sorted natural angular frequencies [Hz]
f=ef/(2*pi);
ix=1:20; % plotting counters
% plot results
% plot analytical and FE natural frequencies
% for all available FE values
figure(1)
subplot(1,2,1)
plot(ix,ef(ix),'ro', ix,om(ix),'kx','markersize',8);
title(' o - FEM, x - analytical');
xlabel('counter'); ylabel('angular frequencies');
% compute and plot relative errors
% for the first 9 frequencies
ix = 1:9; r=zeros(size(ix));
for i=ix
    r(i)=100*(ef(i)-om(i))/om(i);
end
subplot(1,2,2)
plot(ix,r,'ko','markersize',8);
title('relative errors for FE frequencies [%]');
xlabel('counter');
print VBEbefre2 -deps; print VBEbefre2 -dmeta;
% print results for the first 9 frequencies
format long e
delete befre2.dat
diary befre2.dat
disp(' counter, analytical frequencies, FE frequencies')
disp([ix' om(ix)' ef(ix)])
diary off
format short e
% end of VBEbefre2.m

```

The output of VBEbefre2.m prints the first 9 natural frequencies of a thin cantilever beam computed analytically and by the finite element method.

```

counter, analytical frequencies, FE frequencies
1    5.266504690912090e+002    5.266509194371887e+002
2    3.300462151726965e+003    3.300571391657554e+003
3    9.241389593048039e+003    9.243742518773286e+003
4    1.810943523875022e+004    1.812669270993247e+004
5    2.993619402962561e+004    3.001165614576545e+004
6    4.471949023233439e+004    4.496087393371327e+004
7    6.245945376065551e+004    6.308228786109306e+004
8    8.315607746908118e+004    8.451287572802173e+004
9    1.068093617279631e+005    1.092740977881639e+005

```

The graphical output is in Fig. 3. In the left subfigure there are all natural frequencies obtained by finite element calculation – there are 20 of them in this case, i.e. for 10 beam elements. (Recall that discrete systems have a frequency spectrum that always consists of a finite number of frequencies only.) Frequencies computed by finite element method are compared with the first 20 corresponding frequencies obtained by treating the cantilever beam as continuum. The low frequencies are almost identical – in the right subfigure, where relative differences of the first 9 frequencies are presented, one can observe that the maximum difference is less than 2.5%. This is a marvellous achievement considering that only 10 elements were employed for the discretization and subsequent computation. In the upper part of the spectrum the frequencies are diverging, those due to finite element modelling are higher. This is due to the fact that the discrete finite element model is stiffer than the continuous one.

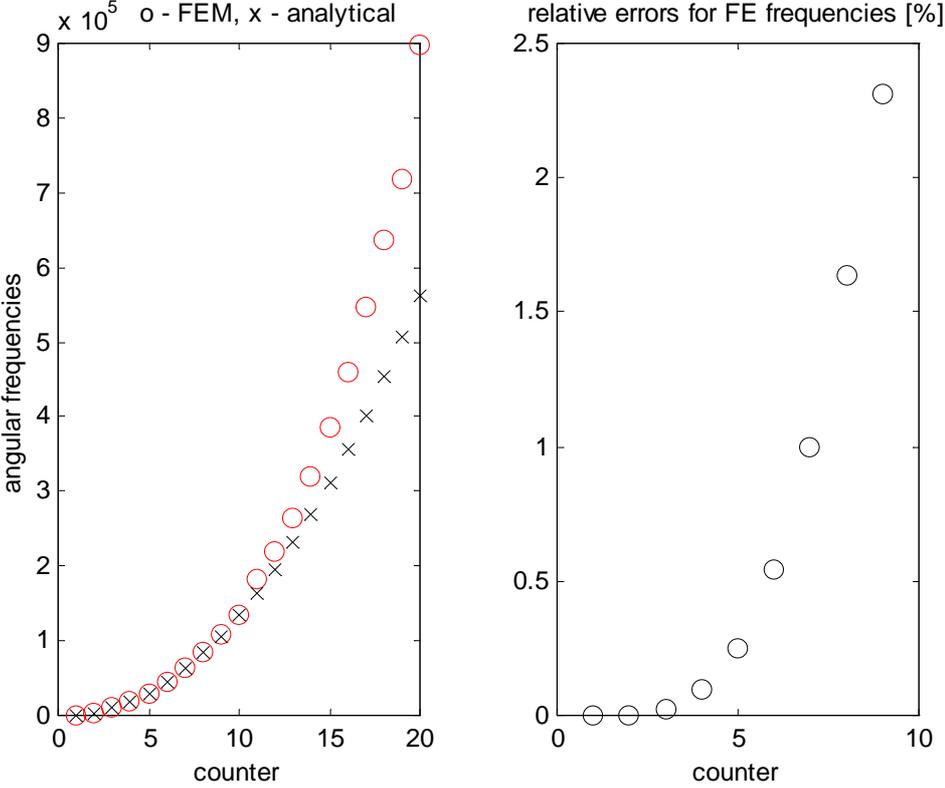


Fig. 3
Comparison of analytical and FE approaches to natural frequency computation of a thin cantilever beam

Since only the lowest frequencies are usually of main engineering interest, we should not be depressed by the fact that the frequencies in the upper part of the spectrum are differing a lot. If we increased the number of elements we would get absolutely more finite element

frequencies that agree well with those of a continuous approach. You should find out, however, that only the frequencies from the lower third of the spectrum would exhibit a good agreement. Try to observe this phenomenon by changing the value k_{\max} in the program.

We are often tempted to think that the continuous model is exact and the finite element one approximate only. That's true, in a sense. We should, however, bear in mind that we are comparing two models here and not our computations with reality and also that continuous model based on Bernoulli-Euler assumptions is as good as far as these assumptions are not violated. Nevertheless checking finite element results with those obtained by continuous models (if it is possible) is a good practice.

References

- [1] Bathe, K.-J., 1996: Finite Element Procedures, New Jersey, Prentice-Hall, Inc., p. 1037.
- [2] Cook, R.D., 1991: Concepts and applications of finite element method, New York, John Wiley
- [3] Przemieniecki, J.S.: 1968, Theory of matrix structural analysis, McGraw Hill, New York