Finite Element Analysis of Beam Bending Model

Course Project of Computational Science and Engineering

李沐景　晏浩洋

2020 年 12 月 25 日

1 Problem Description

A finite-length beam, with one known boundary on each side, assuming that $Young's\ Modulus$ and $Moment\ of\ Inertia$ of the beam's direction are constants, we can solve the $deflection$ function of the beam from the $stress$ function it recieved (stress is perpendicular to the beam).

2 Modeling the Beam Bending Problem

2.1 Notation Description

x	variable varying along length of the beam
$f(x)$	stress the beam recieved
$u(x)$	deflection of the beam
y	direction of stress
$M(x)$	bending moment of the beam
$F_s(x)$	shear force of the beam
$\rho(x)$	radius of curvature
E	Young's Modulus of the beam
I	moment of inertia of the beam

2.2 Derivation of Euler-Bernoulli Beam Equation

From Hooke's law

$$f(x) = E\epsilon = E\frac{y}{\rho(x)} \tag{1}$$

and the definition of bending moment M and momont of inertia I, we can get

$$\begin{aligned} M(x) &= \int_a y f(x) da \\ &= \frac{E}{\rho(x)} \int_a y^2 da \\ &= \frac{EI}{\rho(x)} \end{aligned} \tag{2}$$

Also, with the definition of radius of curvature

$$\begin{aligned} \frac{1}{\rho} &= \frac{\frac{d^2u}{dx^2}}{(1+\frac{du}{dx})^{\frac{3}{2}}} \\ &\approx \frac{d^2u}{dx^2} \quad when \;\lim \frac{du}{dx} = 0 \end{aligned} \tag{3}$$

with euqation 2 and 3

$$M(x) = EI\frac{d^2u(x)}{dx^2} \tag{4}$$

analyse the balence of a small section of the beam, we can get

$$\begin{aligned} F_s(x) - (F_s(x) + dF_s(x)) + f_s(x)dx = 0 \\ f(x) = \frac{dF_s(x)}{dx} \end{aligned} \tag{5}$$

also

$$\begin{aligned} \sum M_0 = 0 \\ M(x) + F_s(x)dx + f(x)dx - \frac{dx}{2} - (M(x) + dM(x)) = 0 \\ \frac{dM(x)}{dx} = F_s(x) \; with\ omitting\ the\ infintesimal\ of\ higher\ order \end{aligned} \tag{6}$$

with equation 5 and 4

$$\begin{aligned} f(x) &= \frac{d^2M(x)}{dx^2} \\ &= EI\frac{d^4u(x)}{dx^4} \end{aligned} \tag{7}$$

3 Solving deflection of the beam

3.1 Finite Element Analysis solving fourth order differential equation

Firstly, we divide the beam of length L into n sections, each section has length $h = \frac{L}{n}$

Then, we construct the bases and compute their second order differential

$$\begin{aligned} \phi^d(X) &= (|X|-1)^2(2|X|+1) \\ \phi^s(X) &= X(|X|-1)^2 \\ where\ X &= \frac{x-ih}{h} \end{aligned} \tag{8}$$

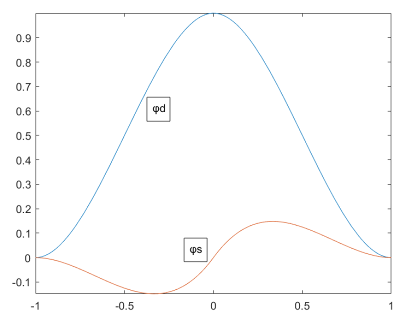

图 1: graph of basis when $i = 0$ in $[-h, h]$

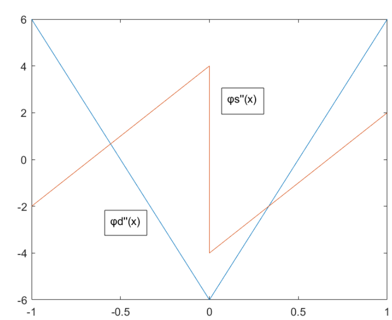

图 2: graph of basis's second order differential when $i = 0$ in $[-h, h]$

To fit these basis's second order differential in vector formation, we use $\frac{1}{h^2}[6, 0, -6]$, $\frac{1}{h^2}[-6, 0, 6]$, $\frac{1}{h^2}[-2, 1, 4]$, $\frac{1}{h^2}[-4, 1, 2]$ to respectively fit fisrt half of ϕ_d'', second half of ϕ_d'', fisrt half of ϕ_s'', second half of ϕ_s''. What's more, we get the interpolation vector $W = h[\frac{1}{6}, \frac{4}{6}, \frac{1}{6}]$ from Simpson's 3-point rule.

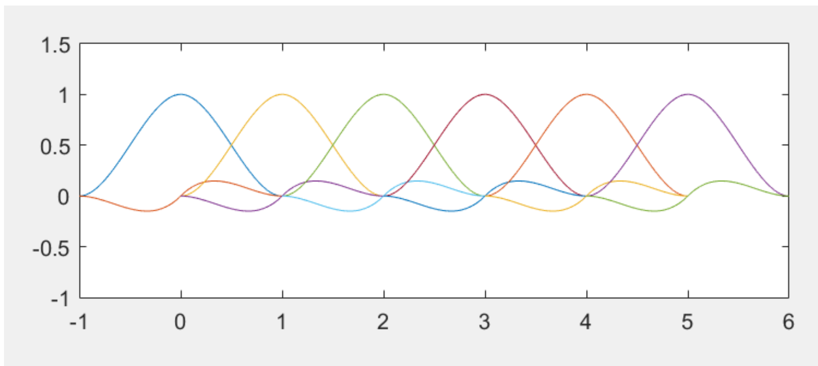

图 3: graph of basis when $i = 0, 1, 2, 3, 4, 5$ in $[-h, 6h]$

From the figures above, for example, in range $[0, h]$, there are $\phi_0^d, \phi_0^s, \phi_1^d, \phi_1^s$ four non-zero functions, so we can construct the 4*4 Element Stiffness Matrix

$$K_i = \begin{bmatrix} \phi_0^{d''} \\ \phi_0^{s''} \\ \phi_1^{d''} \\ \phi_1^{s''} \end{bmatrix} * \begin{bmatrix} \phi_0^{d''} & \phi_0^{s''} & \phi_1^{d''} & \phi_1^{s''} \end{bmatrix} * W \tag{9}$$

Then, we construct the Assembled Stiffness Matrix

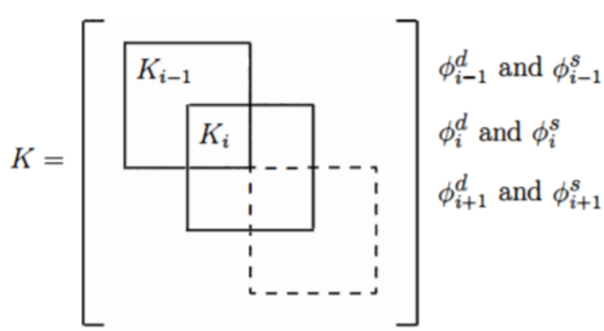

图 4: Assembled Stiffness Matrix

$$F = [\int f\phi_d, \int f\phi_s, \int f\phi_d, \int f\phi_s, \ldots, \ldots] \tag{10}$$

$$U = K \backslash F \tag{11}$$

3.2 Boundary Condition

We consider 3 kinds of boundary conditions, fixed, suppored, free, the expression is as follows

$$\begin{aligned} fixed : u(x) = 0,\ \frac{du(x)}{dx} = 0 \\ supported : u(x) = 0,\ M(x) = 0 \\ free : M(x) = 0,\ \frac{dM(x)}{dx} = 0 \end{aligned} \tag{12}$$

for boundary with limitation $u(x) = 0$

$$\phi_0^d(0) = 0 \quad \phi_0^s(0) = 0$$

which means we need to drop ϕ_0^d, which is the first/last row/column of Stiffness Matrix

for boundary with limitation $\frac{du(x)}{dx} = 0$

$$\phi_0^{d'}(0) = 0 \quad \phi_0^{s'}(0) = 0$$

which means we need to drop ϕ_0^s, which is the second/last but one row/column of Stiffness Matrix

for boundary with limitation $M(x) = 0$ or $\frac{dM(x)}{dx} = 0$

$$beacuse\ \phi_0^{d'''}(0) = \phi_0^{s'''}(0) = \phi_0^{d''''}(0) = \phi_0^{s''''}(0) \equiv 0$$

so there's no row/column need droped

3.3 Uniform Stress and Delta Stress

In order to apply our model in reality, we seperate stress the beam recieved $f(x)$ into two parts, uniform stress f_0, also called load intensity, which uniformly effect on any x, unit in N/m, and delta stress f_δ, just effect on specific point of the beam x_δ $(x, x_\delta \in (0,1))$, unit in N.

$$f(x) = f_0 + f_\delta * \delta(x - x_\delta) \tag{13}$$

To compute the effect of delta stress f_δ, we add $\phi_d(X) * f_\delta$ and $\phi_s(X) * f_\delta$ to respective item of F, and then compute equation 11.

4 Results Displaying

Firstly, we set length $L = 1$, the product of Young's modulus and momoent of inertia $EI = 1$, with only uniform stress $f_0 = -1$, compute the deflection of the beam $u(x)$ in different boundary conditions. The solution by elements $n = 10$ is plot in red. Also, we plot the analysis solution in blue, which derived from

$$-u(x) = \frac{1}{24}x^4 + Ax^3 + Bx^2 + Cx + D \tag{14}$$

Boundary condition of fixed-fixed, fixed-supported, fixed-free, supported-supported is shown as follows, which fit their analysis solution well, while boundary condition of supported-free and free-free have no unique analysis solution. Finite element analysis also meet error that the matrix is closed to singular value.

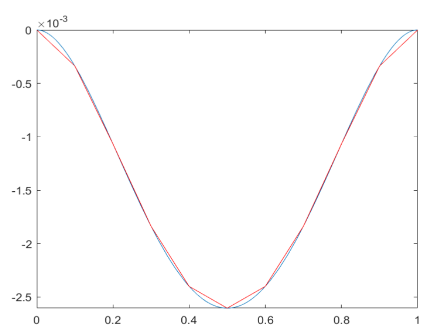

图 5: fixed-fixed

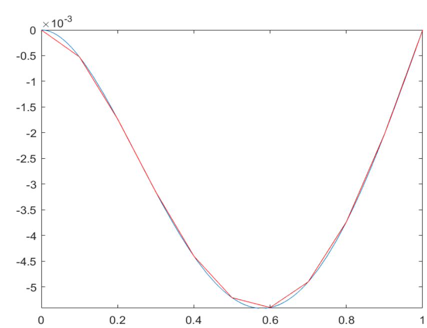

图 6: fixed-supported

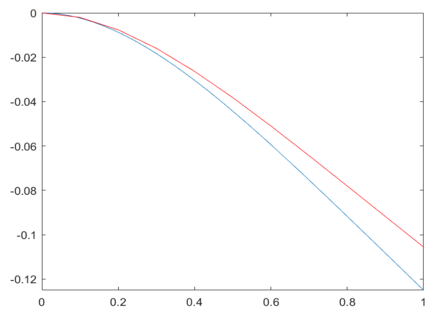

图 7: fixed-free

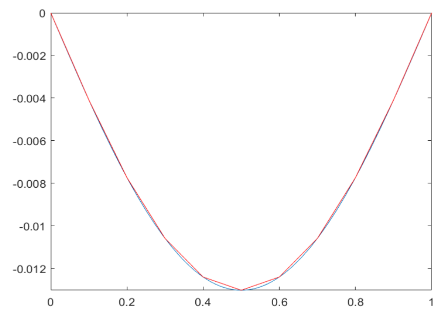

图 8: supported-supported

Then, we add 3 delta stress $f_\delta = [4, -2, 4]$ at the $x_\delta = [0.2, 0.5, 0.8]$ to the fixed-fixed model above, the result of finite element analysis is as follows

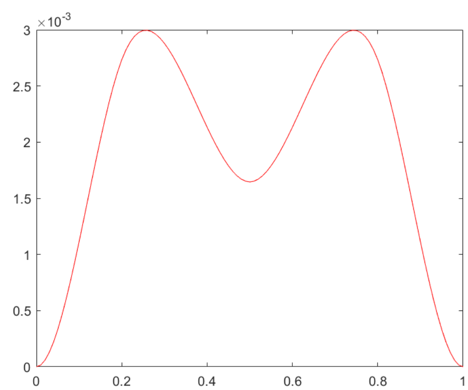

图 9: $f_0 = -1$, $f_\delta = [4, -2, 4]$, $x_\delta = [0.2, 0.5, 0.8]$

A appendix: code

```
clear all;
default = {'1'};
L = inputdlg('input the length of the beam L=','L=',1,default);
L = str2double(L);
EI =inputdlg('input the product of Young''s modulus and momoent of inertia
    of the beam EI=','EI=',1,default);
EI = str2double(EI);
default = {'-1'};
```

```
f = inputdlg('input the uniform stress f0=','f0=',1,default);
f = str2double(f);
default = {'0'};
fdelta = inputdlg('input the delta stress fdelta=','fdelta=',1,default);
fdelta = str2num(fdelta{1});
default = {'0.5'};
xx = inputdlg('input the effect position of delta stress x=','x=',1,default);
xx = str2num(xx{1});
[left,~] = listdlg('ListString',{'fixed','supported','free'},'Name','node','PromptString','what is the status of the left node?','SelectionMode','Single');
[right,~] = listdlg('ListString',{'fixed','supported','free'},'Name','node','PromptString','what is the status of the right node?','SelectionMode','Single');

n = 100; %number of elements
h = L/n;

%fd" and fs"
fd2back = [-6,0,6]/h^2; %(0,1)
fd2forw = [6,0,-6]/h^2; %(-1,0)
fs2back = [-4,-1,2]/h^2; %(0,1)
fs2forw = [-2,1,4]/h^2; %(-1,0)

W=[h/6,h*4/6,h/6]';

%Elementry Stifness Matrix
func = [fd2back;fs2back;fd2forw;fs2forw];
E=zeros(4);
for i = 1:4
  for j = 1:4
    E(i,j) = func(i,:).*func(j,:)*W;
  end
end

%construct the global matrix
K = zeros(2*n+2,2*n+2);
for i = 1:n
  aa = 2*i-1;
  %K(aa:aa+3,aa:aa+3) = E;
  %因为这个是积分 所以是相加 （分区见积分 然后自然就是求和
  K(aa:aa+3,aa:aa+3) =K(aa:aa+3,aa:aa+3)+ E;
end

syms x
X=x/h;
```

```
phi0s= X* (abs(X)-1)^2;
X2=(x-h)/h;
phi1d=(abs(X2)-1)^2*(2*abs(X2)+1);

%Apply BC
 if right == 1
% Built in fixed in Beam
% u =0 u'=0
%drop phid 0 phid N phis 0 phis N
  K(end,:)=[];
  K(:,end)=[];
  K(end,:)=[];
  K(:,end)=[];
end
 if right == 2
% simple supported in Beam
% u =0
  K(end-1,:)=[];
  K(:,end-1)=[];
end
 if left == 1
  K(1,:)=[];
  K(:,1)=[];
  K(1,:)=[];
  K(:,1)=[];
end
 if left == 2
  K(1,:)=[];
  K(:,1)=[];
end

% assume F is a constant function

intphid=int(phi1d*f,[0,2*h]);
intphis=int(phi0s*f,[0,h]);
intphid=double(intphid);
intphis=double(intphis);

for i =1:2:2*(n-1) %from 1 2 3 4 ... N-1
  F(i)=intphid;
  F(i+1)=0;
end

fd=@(X)(abs(X)-1)^2*(2*abs(X)+1);
fs=@(X)X* (abs(X)-1)^2;

```

```
for t=1:length(fdelta)
   ii=floor(xx(t)*L/h); %in the i th interival the index is i+1
%from global to local
  X=(xx(t)/L-ii*h/L)/h;
  F(ii*2-1)= F(ii*2-1)+fd(X)*fdelta(t);
  F(ii*2)= F(ii*2)+fs(X)*fdelta(t);
  % 2 basis function will be influenced
   ii=ii+1;
  X=(xx(t)/L-ii*h/L)/h;
  F(ii*2-1)= F(ii*2-1)+fd(X)*fdelta(t);
  F(ii*2)= F(ii*2)+fs(X)*fdelta(t);
end

%BC
 if right == 2
  F = [F,-intphis];
end
 if right == 3
  F = [F,-intphis,-intphis];
end
 if left == 2
  F = [intphis,F];
end
 if left == 3
  F = [intphis,intphis,F];
end

U=K\F';

%Construct the origin function
fd=@(X)(abs(X)-1)^2*(2*abs(X)+1);
fd0=fd(0); %should be 1

%BC
 if right == 2
  U = U(1:end-1);
end
 if right == 3
  U = U(1:end-2);
end
 if left == 2
  U = U(2:end);
end
 if left == 3
  U = U(3:end);
end
```

```
u=[];
for i =1:2:2*n-2 %from 1 2 3 4 ... N-1
  u = [u,U(i)*fd0];
end

%BC
if right ~= 3
  if left ~= 3
    u = [0,u,0];
  else
    u = [2*u(1)-u(2),u,0];
  end
else
  u = [0,u,2*u(end)-u(end-1)];
end
u = u*EI;

plot(0:h:L,u,'r')
```