Modeling the Deflection of a Beam

Given a beam with tension on each end and a uniform weight load on the beam, the deflection of the beam, or how much it “bends” under the weight, can be modeled by the following equation:

.

This boundary value problem has global parameters E=modulus of elasticity, I=moment of inertia and L=length of the beam, which are specific to the type of beam. The parameters T=end tension and w=uniform weight load are the variables that we will use as inputs to test the deflection of the beam under different conditions. In addition, the boundary conditions are always y(0)=y(L)=0, meaning that there will always be zero deflection at the ends of the beam. We were able to find the exact solution to this ODE in our resource, which will allow us to compare our numerical solutions to the exact solution. This exact solution is given by

,

where and and *y(x)* measure the deflection in the beam at point x.

This type of model can be used to determine whether or not a beam will break given a specific load or can help determine the tension needed at the ends of the beam to keep the beam from bending to its breaking point. In other words this model, while simple, has various real world applications. The physical example that we are going to examine with this model is the case of a W12x22 structural steel I-beam, which has length L=120, E=29\*106, and I=121. We will vary the amount of tension on the ends of the beam and the uniform weight load to determine the amount of deflection at each point on the beam.

We used the following three numerical methods to approximate the solution to the boundary value problem: the Collocation Method, the Finite Difference Method, and the Shooting Method.

1. For the Collocation method we followed the basic structure of the code given in class. However, since our boundary conditions are 0<x<120, we had to define A(N+1,k) as 120^(k-1). Then, since the left hand side of our ODE is y''-(y)T/EI, we have A(i,j)=(j-1)\*(j-2)\*x(i)^(j-3)-(T/(3.509\*10^9))\*x(i)^(j-1), where 3.509\*10^9 is the value given by EI. Since this is a nonhomogeneous problem we also have the loop that defines g(m)=(w/(2\*3.509\*10^9))\*x(m)\*(120-x(m)). Since our resource was able to provide the exact solution to the ODE we were able to include it in our code so that the numerical and exact solutions are plotted together. Below is the plot for when w = T= 10,000 using our Collocation Method.



From the plot we can see that our numerical solution is very close to the exact solution. Now we will look at a graph for which the tension, T, will be constant and we will vary *w* to observe the results. Looking at the graph below, we can see that our numerical solution still matches the exact solution for our different inputs. As we can see from the series of graphs that as weight is added to the beam the deflection at each point increases. Here T=10000 was constant and we used w = 1000, 10000, and 50000.



1. For the Finite Difference Method we used the centered difference to approximate y’’ in the equation

where, and

with

Solving for we get

To solve this numerically we initialize our variables a, b, h, w, T and M. Where a = 0 and b=L are our boundary points and y(a)=(b)= 0 are our boundary conditions. Here *h* is our step size and M is our iteration count. We used a for loop to compute our different values of yi, using (\*) from above, at each of our mesh points xi where a < xi < b. This loop was nested inside another loop which would update our approximations for yi for M iterations. We chose M so that our values yi converged to a single solution that was not dependent on our choice of step size. For this problem we chose M = 200 as our iteration count. As we had the exact solution we were able to plot our numerical solution against the exact solution for comparison.

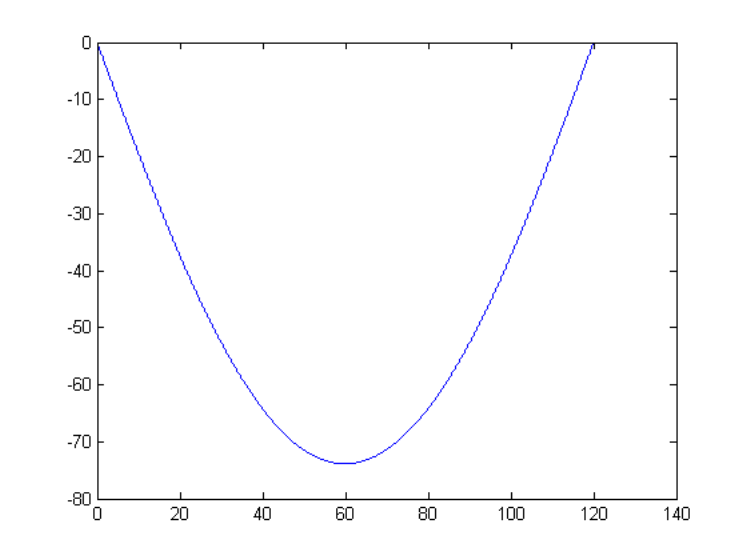


Above is the graph of our solution using the FDM Method with w = 10,000 and T = 10,000. As you can see our numerical solution is very close to the exact solution. Below is a table of values that show the deflection of our beam at each point xi using various values for w and T.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| xi | w = 10,000 | | T=10,000 | T=10,000 | |
| T=1000 | T=100,000 | w = 10,000 | w=1,000 | w=100,000 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | -2.037 | -1.9568 | -2.093 | -0.203 | -20.29 |
| 20 | -3.920 | -3.7624 | -3.902 | -0.390 | -39.02 |
| 30 | -5.512 | -5.2938 | -5.492 | -0.549 | -54.92 |
| 40 | -6.723 | -6.4555 | -6.698 | -0.670 | -66.97 |
| 50 | -7.478 | -7.1797 | -7.450 | -0.745 | -74.50 |
| 60 | -7.734 | -7.4256 | -7.705 | -0.771 | -77.05 |
| 70 | -7.478 | -7.1797 | -7.450 | -0.745 | -74.50 |
| 80 | -6.723 | -6.4555 | -6.698 | -0.670 | -66.97 |
| 90 | -5.512 | -5.2938 | -5.492 | -0.549 | -54.92 |
| 100 | -3.920 | -3.7624 | -3.902 | -0.390 | -39.02 |
| 110 | -2.037 | -1.9568 | -2.093 | -0.203 | -20.29 |
| 120 | 0 | 0 | 0 | 0 | 0 |

One thing we conclude from this is that if our beam has a fixed tension and we increase the weight across the beam, the degree of bend in the beam is gradually increasing. However if the weight is uniformly distributed and you only increase the tension we will see less of a deflection in the beam as the tension is increased.

3. With Shooting Method, recall that it builds upon the basics of the finite difference method. In order to allow for solving along the boundaries, it uses the additional condition of keeping track of y'', referred to as z'. Since we are modeling across the I Beam, we treat the ends of the I beams as zero, that is y(0) = y(L) = 0. From here, we shoot our model out. After we calculate ending values of y for two different values that are on either side of our desired boundary value, we then use a linear approximation to the desired initial slope using our two values for y. After this, we model using the calculated initial slope and check to see if it matches our desired output. Interestingly about this problem, using two very close values for the initial slope create wildly divergent models. Using an initial slope of -2 puts our model close to our desired final y of 0, whereas if we use an initial slope of -1 it puts our final y at nearly 124. Therefore, this method is the least effective method out of the three we used.



Given that we have the exact solution to this boundary value problem, we can compare our numerical solutions to the exact solution and see that our results are accurate. We can also conclude that our results are reasonable based on our knowledge of how a beam would behave with varying amounts of weight and tension applied to it. When the weight on the beam is much greater than the end tension, the deflection of the beam is much greater than when the end tension is greater than the amount of weight on the beam. We expect that this would be the case and overall our numerical methods are producing physically reasonable conclusions.

**Resources**

*Introduction to Numerical Methods and Matlab Programming for Engineers,*

Todd Young & Martin Mohlenkamp, ww.math.ohiou.edu/courses/math3600/lecture33.pdf‎

**MATLAB CODES:**

**Collocation Method**

function [x,y]=CM\_Project2(a,b,h,alpha,beta,w,T)

%Gives solution to how much deflection is occuring at each point x

% a=0,b=120,alpha=0,beta=0

%From initial conditions of a W12x22 structural steel I-beam-->

%L=120,E=29\*10^6,I=121.

%This code allows for different weight loads on beam (w) and end tension

%(T)

%alpha (aa)=T/EI and beta (bb)=w/2EI below

N=(b-a)/h;

x=a:h:b;

A=zeros(N+1,N+1);

A(1,1)=1;

for k=1:N+1

A(N+1,k)=120^(k-1);

end

g=zeros(N+1,1);

g(1)=alpha; g(N+1)=beta;

for i=2:N

for j=1:N+1

A(i,j)=(j-1)\*(j-2)\*x(i)^(j-3)-(T/(3.509\*10^9))\*x(i)^(j-1);

end

end

for m=2:N

g(m)=(w/(2\*3.509\*10^9))\*x(m)\*(120-x(m));

end

c=A\g;

syms t

y=c(1)\*1;

for i=1:N

y=y+c(i+1)\*t^i;

end

ezplot(y,[a,b]);

hold on

aa=(T/(3.509\*10^9)); %alpha=T/EI

bb=(w/(2\*3.509\*10^9)); %beta=w/2EI

xx=a:0.001:b;

yy=(-2\*bb/aa^2)\*(exp(sqrt(aa)\*120)/(exp(sqrt(aa)\*120)+1))\*exp(-sqrt(aa)\*xx)+(-2\*bb/aa^2)\*(1/(exp(sqrt(aa)\*120)+1))\*exp(sqrt(aa)\*xx)+(bb/aa)\*xx.^2-(120\*bb/aa)\*xx+(2\*bb/aa^2);

plot(xx,yy,'Color','r');

title('Numerical Solutions by Collocation Method');

legend('Numerical Solution','Exact Solution','Location','northwest');

xlabel('Point X on Beam');

ylabel('Deflection');

end

**Finite Difference Method using Centered Difference**

function [ x,y ] = FDM\_Project2(a,b,h,alpha,beta,w,T,M)

%Gives solution to how much deflection is occurring at each point x

% a=0,b=120,alpha=0,beta=0

%From initial conditions of a W12x22 structural steel I-beam-->

%L=120,E=29\*10^6,I=121.

%This code allows for different weight loads on beam (w) and end tension

%(T)

N = (b-a)/h;

L = 120;

E = 29\*10^6;

I = 121;

A = T/(E\*I);

B = w/(2\*E\*I);

x = a:h:b;

y = zeros(N+1,1);

y(1) = alpha; y(N+1) = beta;

for j=1:M

for i=2:N

y(i)= (y(i+1) + y(i-1) - h^2\*B\*x(i)\*(L-x(i)))/(h^2\*A + 2);

end

end

plot(x,y,'o-');

hold on

xx = a:0.001:b;

yy =(-2\*B/A^2)\*(exp(sqrt(A)\*L)/(exp(sqrt(A)\*L)+1))\*exp(-sqrt(A)\*xx)+(-2\*B/A^2)\*(1/(exp(sqrt(A)\*L)+1))\*exp(sqrt(A)\*xx)+(B/A)\*xx.^2-(L\*B/A)\*xx+(2\*B/A^2);

plot(xx,yy,'Color','r');

title('Numerical Solutions by FDM Relaxation Method');

legend('Numerical Solution','Exact Solution','Location','northwest');

xlabel('Point x(i) on Beam');

ylabel('Deflection');

end

**Shooting Method Code**

function [x,y,z]=beam\_with\_tension\_shooting(h,w,T,gamma)

%hardcode in teh constants

L = 120;

E = 29\*10^6;

I = 121;

alpha = T/(E\*I);

beta = w/(2\*E\*I);

N = L/h;

x = zeros(N+1,1);

y = zeros(N+1,1);

z = zeros(N+1,1);

x(1) = 0;

y(1) = 0;

z(1) = gamma;

for i = 1:N

y(i+1) = y(i) + h\*z(i);

z(i+1) = z(i) + h \* (beta\*x(i)\*(L-x(i)) + alpha\*y(i));

x(i+1) = x(i) + h;

end

end