Shota Matsumoto

Professor Ryu

Computer Programming For Engineers

Assignment 3 Report: Mechanical Engineering Case Study - Beam Deflection

10/05/2024

STEP 1: Problem Identification and Statement

Problem statement:
The objective of this assignment is to develop a software on MATLAB that allows users to calculate the vertical deflection of a beam. It prompts the user for valid inputs and returns the beam deflection as output along with a plot charting out the beam deflections as continuous distance (from the left datum of the beam).

STEP 2: Gathering of Information and Input and Output Description

Our goal is to calculate the deflection of the beam with values given by the user. In order to do so, we need to first identify some information and understand its functionality. These key information are listed below:

- x: Distance from the left end of the beam
  This is the distance from the left end of the beam to where you are measuring the deflection. This element is really important as it tells you where along the beam's length you are interested in knowing the deflection.

- P: Load
  This is the applied P on the beam. It could be a force or a distributed load. The magnitude of the load directly affects the amount of deflection the beam experiences.

- L: Length of the beam
  It is the length of the beam between two supports. This plays a crucial role because longer beams experience more deflection under the same load compared to shorter beams.

- a: Location where the load P is applied
  This is the location where the load P is applied. This distance is measured from the left end of the beam. The position of the load affects how the beam bends and where the maximum deflection occurs.

- E: Young's modulus of the beam material

Young's modulus is the measure of stiffness of the material. A higher Young's modulus means the material is stiffer and will deflect less under a given load.

- I: Second moment of area
  This is the second moment of area, which is also known as the moment of inertia, of the cross-sectional shape of the beam. It quantifies how the area of the cross-section is distributed around the axis of bending. A higher moment of inertia means the beam is more resistant to bending.

- V: Deflection of the beam
  This represents the vertical deflection of the beam at a given point along its length, measured perpendicular to the original axis of the beam. It tells you how much the beam is bending downwards under the applied load.

- Formula for two different cases
  1. When $0 < x < a$
     The equation for deflection is given by:

$$V = \frac{Pb}{6EIL}[(-L^2 + b^2)x + x^3], 0 < x < a$$

Figure 1: Equation for deflection when $0 < x < a$

  2. When $a < x < L$
     The equation for deflection is given by:

$$V = \frac{Pb}{6EIL}[(-L^2 + b^2)x + x^3 - \frac{L}{b}(x - a)^3], a < x < L$$

Figure 2: Equation for deflection when $a < x < L$

The diagram below shows an input/output (I/O) diagram, where the black box represents the computer program.

The inputs include: x (distance from the left end), P (Load), L (length of the beam), a (location where P is applied), E (Young's modulus), I (second moment of area)
The outputs include: V (deflection of the beam) and Graph of continuous function

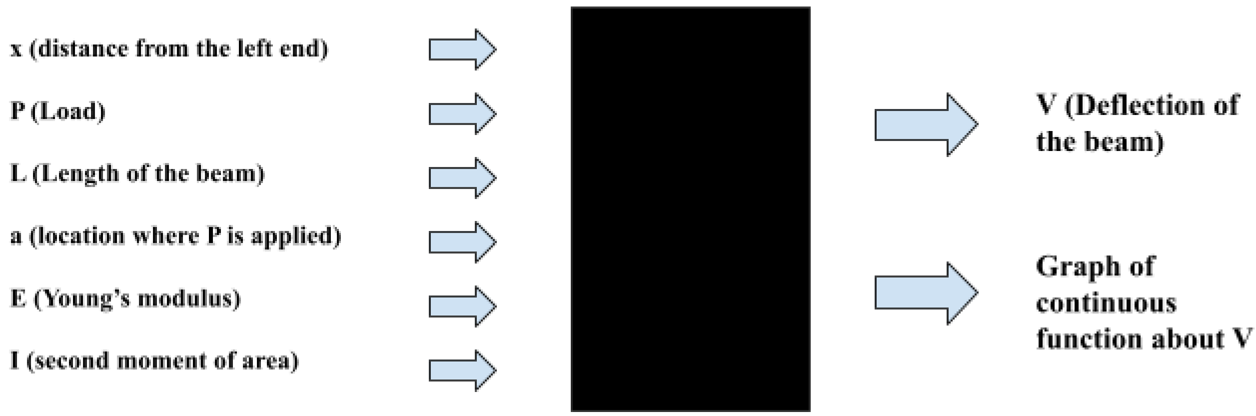


Figure 3: I/O Diagram

Manu Description for I/O Diagram:

1. The user will be prompted to input values for individual variables, including x, P, L, a, E, and I, until they enter valid values.
2. The program will then calculate using the appropriate equation and output the value of the beam's deflection. Additionally, it will plot the graph of its continuous function, with the x-axis representing x and the y-axis representing deflection.

STEP 3: Test Cases and Algorithm Design

There are two distinct situations that vary based on the value of x which we need to consider when calculating the deflection. Let me clearly explain the procedure for finding the deflection in each case, along with an example.

To find the deflection of the beam when $0 < x < a$

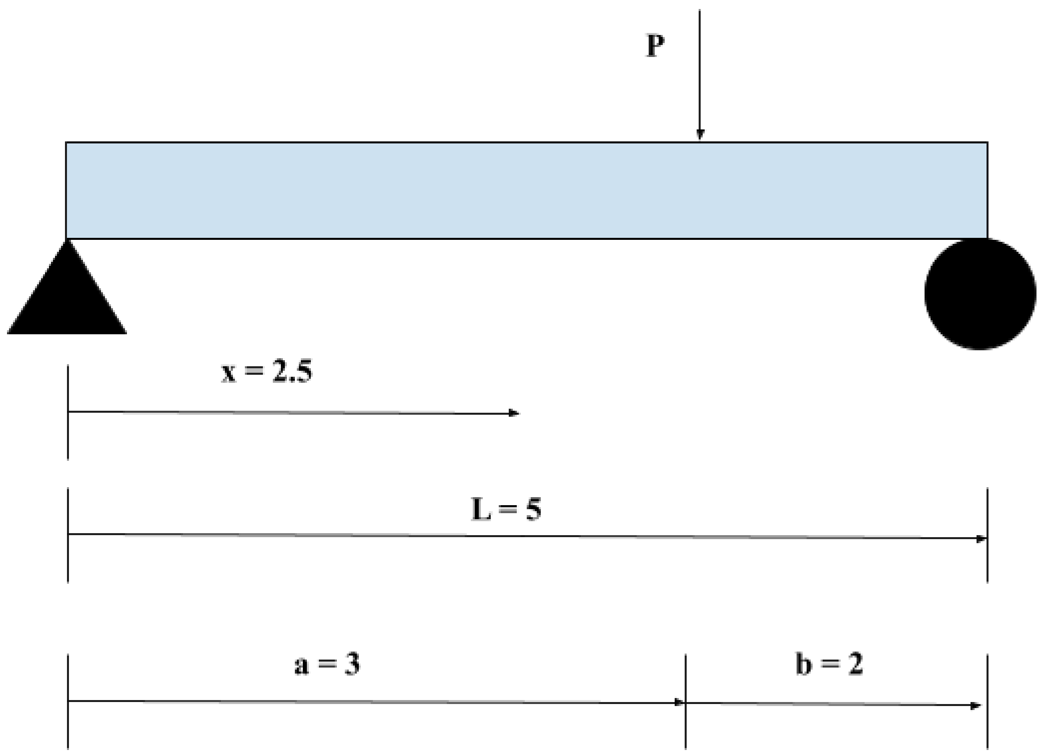
1. Identify given parameters
   Determine the values of the parameters in the equation, including P, E, I, a, x ($0 < x < a$). In this case, we have 2.5 for x, 5 for L, 3 for a, 30e6 for E, 0.0256 for I, and 30000 for P

2. Use given formula to calculate the deflection
   Find the value of the parameter b at first, which is the subtraction of a from L. Substitute the values into the equation to find the deflection V. In this case, V will be -0.0960.

$$V=\frac{Pb}{6EIL}[(-L^2+b^2)x+x^3],0<x<a$$

Figure 4: Formula for finding deflection (0 < x < a)

Dynamic representation of what is going on in this case is illustrated below.



$$V=\frac{Pb}{6EIL}[(-L^2+b^2)x+x^3-\frac{L}{b}(x-a)^3],a<x<L \quad = \quad \mathbf{-0.0960\ (4\ sig\ fig)}$$

Figure 4: Illustration of finding deflection when 0 < x < a

To find deflection when a < x < L

1. Identify given parameters
   Determine the values of the parameters in the equation, including P, E, I, a, x (0 < x < a). In this case, we have 4.05 for x, 5 for L, 3 for a, 30e6 for E, 0.0256 for I, and 30000 for P. Note that x is bigger than a and less than L in this case.

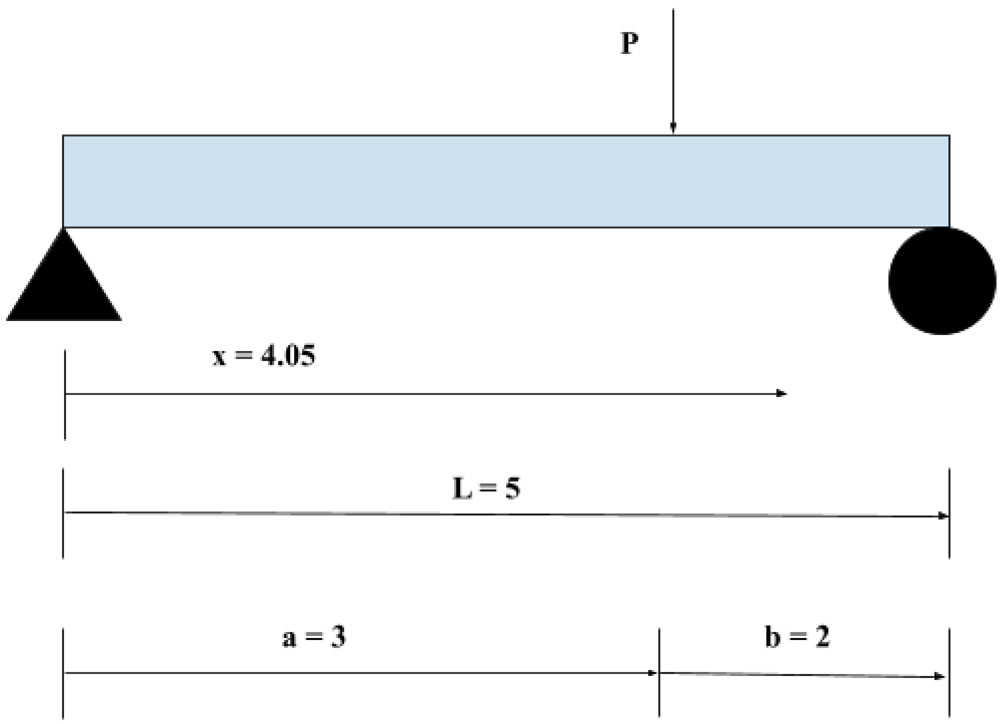
2. Use given formula to calculate the deflection

Find the value of the parameter b at first, which is the subtraction of a from L. Substitute the values into the equation to find the deflection V. In this case, V will be -0.0560.

$$V = \frac{Pb}{6EIL}[(-L^2 + b^2)x + x^3 - \frac{L}{b}(x-a)^3], a < x < L$$

Figure 5: Formula for finding deflection (a < x < L)

Dynamic representation of what is going on in this case is illustrated below.



$$V = \frac{Pb}{6EIL}[(-L^2 + b^2)x + x^3 - \frac{L}{b}(x-a)^3], a < x < L \quad = \quad \textbf{(4 sig fig)}$$

Figure 6: Illustration of finding deflection when a < x < L

*These input/output values serve as one test case.

The following presents a set of test cases that can be used to test the algorithm and software. It is advisable to verify the program using extremely large numbers to assess if the program can handle such inputs. Additionally, we need to check if the program correctly fails when any variable is inputted with a negative value or when a variable is greater than another variable, which should not be the case.

Deflection when $0 < x < a$

Case 1) x = 2.5, L = 5, a = 3, E = 30e6, I = 0.0256, and P = 30000 → The expected outcome is V = -0.0960

Case 2) x = 34, L = 80, a = 50, E = 600e10, I = 0.1543 and P = 40000 → The expected outcome is V = -0.0004

Deflection when $a < x < L$

Case 3) x = 4.05, L = 5, a = 3, E = 30e6, I = 0.0256, and P = 30000 → The expected outcome is V = -0.0560

Case 4) x = 67, L = 100, a = 50, E = 300e6, I = 0.3526, and P = 20000 → The expected outcome is V = -3.3334

Invalid Cases

Case 5) Invalid input for any variable. For example, x = -4 → The expected outcome is “Invalid input. Please enter a valid input.”

Case 6) Invalid input for any variable. For example, L = -10 → The expected outcome is “Invalid input. Please enter a valid input.”

Case 7) Invalid input for any variable. For example, a = 123 (assuming it is greater than beam length) → The expected outcome is “Invalid input. Please enter a valid input.”

Algorithm

The algorithm can be expressed as:

Prompt user to input the length of the beam
Repeat while L is less than or equal to 0
    Prompt user for a valid input for L

Prompt user to input the distance from the left end (x)
Repeat while x is less than or equal to 0 or x is greater than or equal to L
    Prompt user for a valid input for x

Prompt user to input the mass of the load (P)
Repeat while P is less than or equal to 0
    Prompt user for a valid input for P

Prompt user to input the distance from the left where the load is located (a)
Repeat while a is less than or equal to 0, a is greater than or equal to L, or a is equal to x

Prompt user for a valid input for a

Prompt user to input the Young's modulus of the beam (E)
Repeat while E is less than or equal to 0
Prompt user for a valid input for E

Prompt user to input the second moment of area (I)
Repeat while I is less than or equal to 0
Prompt user for a valid input for I

Assign b as L - a

If x is greater than 0 and less than a
Calculate deflection V using formula for x < a
Otherwise x is greater than a and less than L
Calculate deflection V using formula for x >= a

Display the inputs provided by the user

Display the calculated deflection V

Assign function deflection1 to ((P * b / (6 * E * I * L)) * ((-L^2 + b^2) * x + x.^3)) for 0<x<a
Assign function deflection2 to ((P * b / (6 * E * I * L)) * ((-L^2 + b^2) * x + x.^3 - (L / b) * (x - a).^3)) for a<x<L

Plot deflection function for x < a from 0 to a
Plot deflection function for x >= a from a to L

Label the x-axis as "Distance from the left end of the beam"
Label the y-axis as "Deflection (negative sign denotes downward direction)"
Set the title as “Deflection of the Beam”
Display legend for different deflection functions
Enable grid on the plot

STEP 4: Implementation

```
%-------------------------------------------------------------------
%
%Name: Shota Matsumoto, Student Number: sm11745
%Date: May 10, 2024
%Program: CPE 3rd Assignment
%Description: This program computes deflection of the beam by
%considering
```

```matlab
%multiple variables and plots the graph of function.
%-----------------------------------------------------------------------
%

%Prompting the user for individual input and performing validation
L = input('Enter the length of the beam (L)');
while L <= 0
  L = input('Invalid input. Please enter a valid input: ');
end
x = input('Enter the distance from the left end: ');
while x <= 0 || x >= L
  x = input('Invalid input. Please enter a valid input: ')
end
P = input('Enter the load (P): ');
while P <= 0
  P = input('Invalid input. Please enter a valid input: ');
end
a = input('Enter the location where the load P is applied (a): ');
while a <= 0 || a >= L || a == x
  a = input('Invalid input. Please enter a valid input: ');
end
E = input('Enter the Youngs modulus of the beam material (E): ');
while E <= 0
   E = input('Invalid input. Please enter a valid input: ');
end
I = input('Enter the second moment of area (I): ');
while I <= 0
  I = input('Invalid input. Please enter a valid input: ');
end

%Assigning the value for b
b = L - a;

%Conditional Statement which specifies which formula to use
if x > 0 && x < a
  V = ((P*b/(6*E*I*L))*((-L^2+b^2)*x+x.^3));
elseif x > a && x < L
  V = ((P*b/(6*E*I*L))*((-L^2+b^2)*x+x.^3-(L/b)*(x-a).^3));
end

%Display all the inputs and outputs with explanation for each
fprintf('Your Inputs:\n');
fprintf('x = %.2f (distance from left end \n', x);
fprintf('L = %.2f (length of the beam \n', L);
fprintf('P = %.2f (Load) \n', P);
```

```
fprintf('a = %.2f (Location where the load P is applied) \n', P);
fprintf('E = %.2f (Youngs modulus of the beam material) \n', E);
fprintf('I = %.4f (Second moment of area) \n', I);
fprintf('The calculated deflection is %.4f\n', V);

%Defining deflection functions for x < a and x > a
deflection1 = @(x) ((P*b/(6*E*I*L))*((-L^2+b^2)*x+x.^3));
deflection2 = @(x)
((P*b/(6*E*I*L))*((-L^2+b^2)*x+x.^3-(L/b)*(x-a).^3));

% Plotting the deflection along the beam as a continuous distance
from the
% left datum
fplot(deflection1, [0 a]); % Deflection function for x < a
hold on;
fplot(deflection2, [a L]); % Deflection function for x >= a
hold off;

xlabel('Distance from the left end of the beam');
ylabel('Deflection (negative sign denotes downward direction');
title('Deflection of the beam');
legend('Deflection for x < a', 'Deflection for x >= a', 'Location',
'best');
grid on;
```

STEP 5: Tests and Verification

Test Case 1 (above) and Test Case 2 (below):

Input	Output

<table>
<tr>
<td>x = 2.5
L = 5
a = 3
E = 30e6
I = 0.0256
P = 30000</td>
<td>

```
Enter the length of the beam (L)
5
Enter the distance from the left end:
2.5
Enter the load (P):
30000
Enter the location where the load P is applied (a):
3
Enter the Youngs modulus of the beam material (E):
30e6
Enter the second moment of area (I):
0.0256
Your Inputs:
x = 2.50 (distance from left end
L = 5.00 (length of the beam
P = 30000.00 (Load)
a = 30000.00 (Location where the load P is applied)
E = 30000000.00 (Youngs modulus of the beam material)
I = 0.0256 (Second moment of area)
The calculated deflection is -0.0960
>> |
```

</td>
</tr>
<tr>
<td>x = 34
L = 80
a = 50
E = 600e10
I = 0.1543
P = 40000</td>
<td>

```
Enter the length of the beam (L)
80
Enter the distance from the left end:
34
Enter the load (P):
40000
Enter the location where the load P is applied (a):
50
Enter the Youngs modulus of the beam material (E):
600e10
Enter the second moment of area (I):
0.1543
Your Inputs:
x = 34.00 (distance from left end
L = 80.00 (length of the beam
P = 40000.00 (Load)
a = 40000.00 (Location where the load P is applied)
E = 6000000000000.00 (Youngs modulus of the beam material)
I = 0.1543 (Second moment of area)
The calculated deflection is -0.0004
>> |
```

</td>
</tr>
</table>

→The program successfully accepts the inputs, calculates, and displays the deflection of the beam, which agrees with the expected outcomes of -0.0960 and -0.0004, respectively. We can further verify these values using the figure below, which is the graph displayed at the end of each program. The graph plots the results of the beam deflections as a continuous distance from the left datum of the beam.

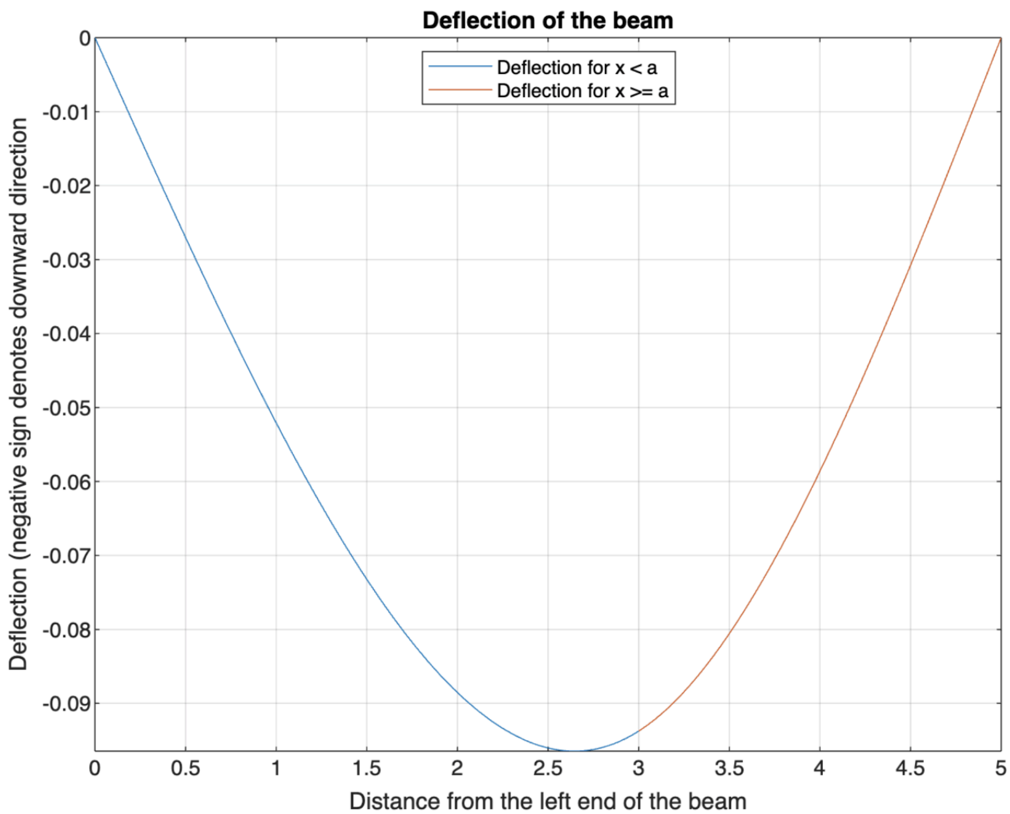


Figure 7: Graph of Test Case 1

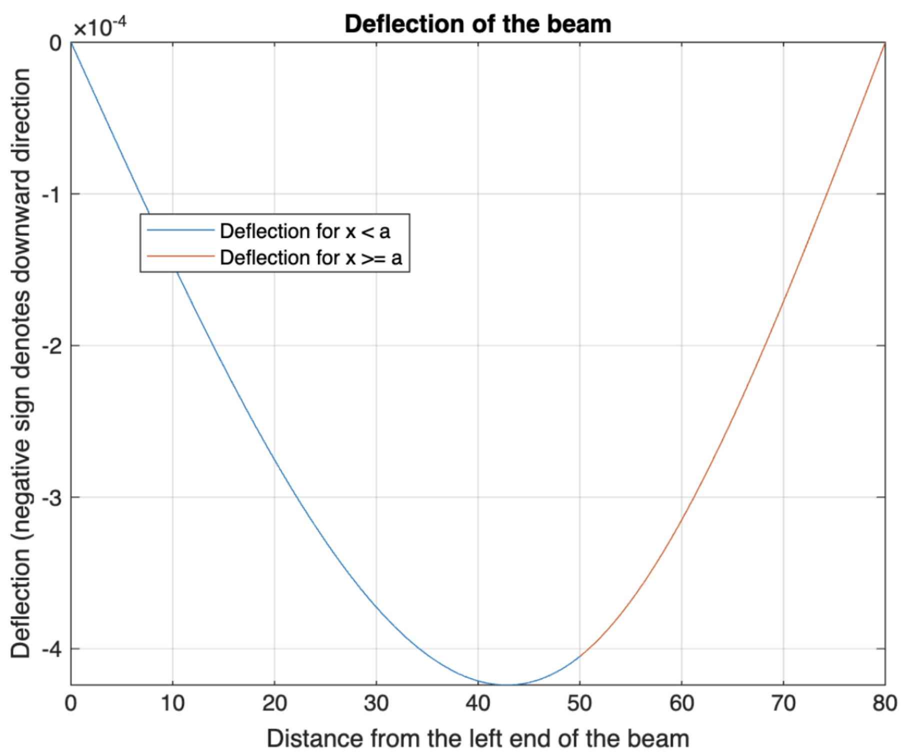


Figure 8: Graph of the Test Case 2

Test Case 3 (above) and 4 (below)

Inputs	Outputs
x = 4.05 L = 5 a = 3 E = 30e6 I = 0.0256 P = 30000	Enter the length of the beam (L) 5 Enter the distance from the left end: 4.05 Enter the load (P): 30000 Enter the location where the load P is applied (a): 3 Enter the Youngs modulus of the beam material (E): 30e6 Enter the second moment of area (I): 0.0256 Your Inputs: x = 4.05 (distance from left end L = 5.00 (length of the beam P = 30000.00 (Load) a = 30000.00 (Location where the load P is applied) E = 30000000.00 (Youngs modulus of the beam material) I = 0.0256 (Second moment of area) The calculated deflection is -0.0560 >>
x = 67 L = 100 a = 50 E = 300e6 I = 0.3526 P = 20000	Enter the length of the beam (L) 100 Enter the distance from the left end: 67 Enter the load (P): 20000 Enter the location where the load P is applied (a): 50 Enter the Youngs modulus of the beam material (E): 300e6 Enter the second moment of area (I): 0.3526 Your Inputs: x = 67.00 (distance from left end L = 100.00 (length of the beam P = 20000.00 (Load) a = 20000.00 (Location where the load P is applied) E = 300000000.00 (Youngs modulus of the beam material) I = 0.3526 (Second moment of area) The calculated deflection is -3.3334 >>

→The program successfully accepts the inputs, calculates, and displays the deflection of the beam, which agrees with the expected outcomes of -0.0560 and -3.3334, respectively. We can further verify these values using the figure below, which is the graph displayed at the end of each program. The graph plots the results of the beam deflections as a continuous distance from the left datum of the beam.

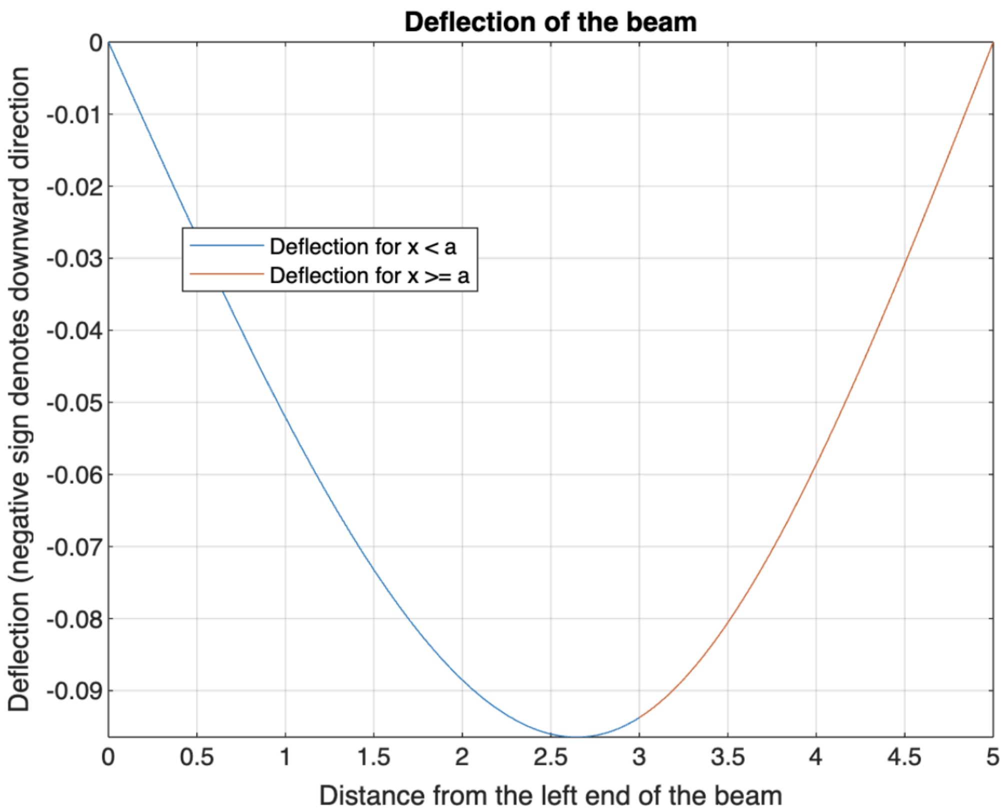


Figure 9: Graph for the Test Case 3

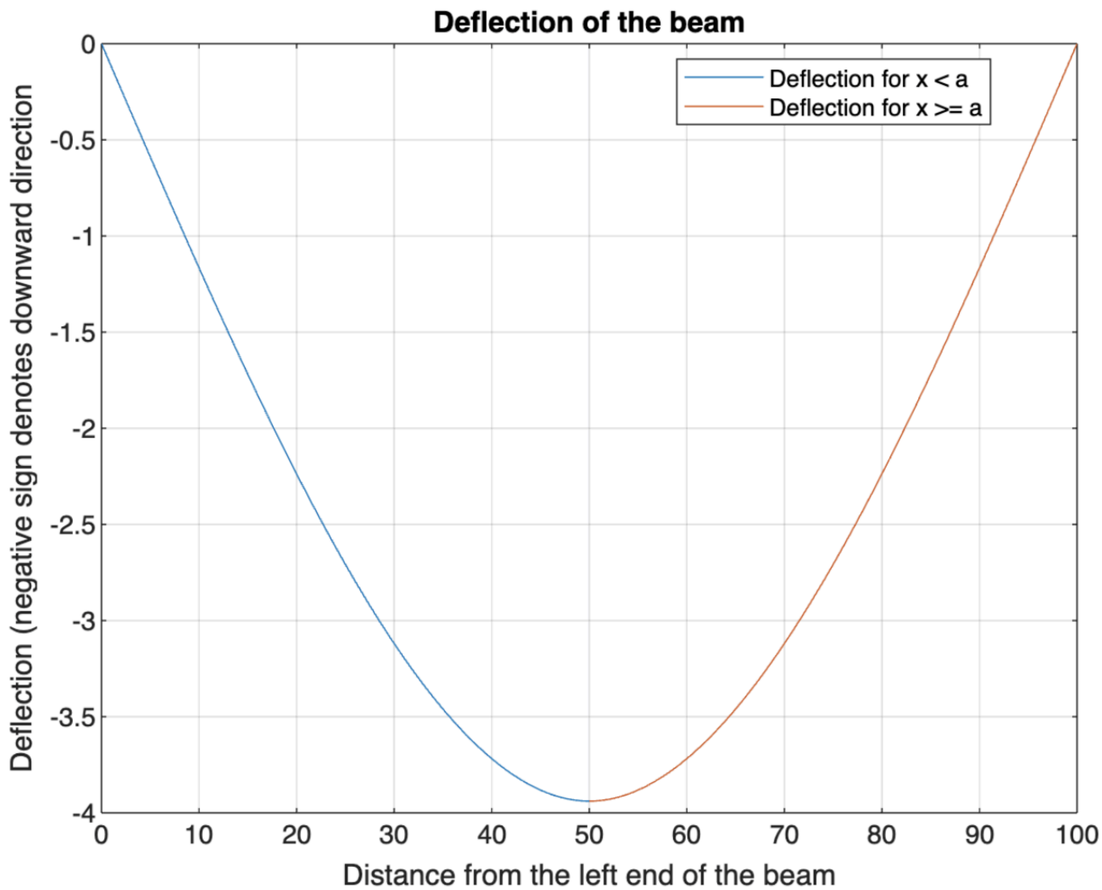


Figure 10: Graph for the Test Case 4

Invalid Cases

Inputs	Outputs

L = -10
L = 0
	`Enter the length of the beam (L)`
	`-10`
	`Invalid input. Please enter a valid input:`
	`0`
	`Invalid input. Please enter a valid input:`

L = 10
x = 0
x = -4
x = 10
x = 20
	Enter the length of the beam (L)
	10
	Enter the distance from the left end:
	0
	Invalid input. Please enter a valid input:
	-4
	
	x =
	
	    -4
	
	Invalid input. Please enter a valid input:
	10
	
	x =
	
	    10
	
	Invalid input. Please enter a valid input:

L = 10 x = 3 P = 0 P = -9	Enter the length of the beam (L) 10 Enter the distance from the left end: 3 Enter the load (P): 0 Invalid input. Please enter a valid input: -9 Invalid input. Please enter a valid input:
L = 10 x = 3 P = 100 a = 0 a = 3 a = 123	Enter the length of the beam (L) 10 Enter the distance from the left end: 3 Enter the load (P): 100 Enter the location where the load P is applied (a): 0 Invalid input. Please enter a valid input: 3 Invalid input. Please enter a valid input: 123 Invalid input. Please enter a valid input:
L = 10 x = 3 P = 100 a = 6 E = 0 E = -23	Enter the length of the beam (L) 10 Enter the distance from the left end: 3 Enter the load (P): 100 Enter the location where the load P is applied (a): 6 Enter the Youngs modulus of the beam material (E): 0 Invalid input. Please enter a valid input: -23 Invalid input. Please enter a valid input:

L = 10
x = 3
P = 100
a = 6
E = 30e6
I = 0
I = -83883
	`Enter the length of the beam (L)`
	`10`
	`Enter the distance from the left end:`
	`3`
	`Enter the load (P):`
	`100`
	`Enter the location where the load P is applied (a):`
	`6`
	`Enter the Youngs modulus of the beam material (E):`
	`30e6`
	`Enter the second moment of area (I):`
	`0`
	`Invalid input. Please enter a valid input:`
	`-83883`
	`Invalid input. Please enter a valid input:`

→ As indicated in the table above, the validation loop repeats until valid inputs are provided for all variables.

In conclusion, the program can successfully calculate the beam deflection when provided with all valid variables. Additionally, it can generate a graph that visually represents the deflection, providing us with a clear visualization of the process.

User Guide:

1. Input Validation:
    - Ensure all inputs are positive and within valid ranges.
    - Length of the beam (L) should be greater than zero.
    - Distance from the left end (x) should be greater than zero and less than the beam's length (L).
    - Load (P) should be greater than zero.
    - Location of the load (a) should be greater than zero, less than the beam's length (L), and not equal to x.
    - Young's modulus of the material (E) should be greater than zero.
    - Second moment of area (I) should be greater than zero.
2. Deflection Calculation:
    - The program calculates deflection based on the input values and applies the appropriate formula based on the position of x relative to a.
3. Output Display:
    - The program displays all input values along with their explanations.
    - It also shows the calculated deflection (V) based on the provided inputs.
4. Deflection Functions:
    - Two deflection functions are defined: one for x < a and another for x >= a.
    - These functions are used to plot the deflection graph.
5. Graph Plotting:
    - The program plots the deflection along the beam as a continuous distance from the left end.

- It uses separate functions for x < a and x >= a to plot the corresponding deflection curves.
- The graph includes labels for axes, a title, and a legend for clarity.

6. Graph Interpretation:
    - The graph visually represents how the beam deflects under the given load and material properties.
    - Negative deflection values denote a downward bending of the beam.
7. Grid and Legend:
    - The graph includes a grid for better visualization.
    - A legend is provided to distinguish between deflection for x < a and x >= a.
8. Graph Interaction:
    - Users can interact with the graph to analyze the deflection pattern along the beam's length.
9. Data Input Considerations:
    - Users are advised to input valid and realistic data to obtain meaningful deflection results and a representative graph.