# ME-212 Solid Mechanics - Computer Project 1

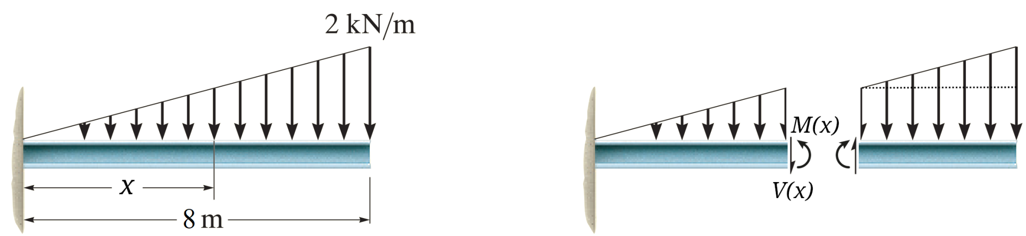
Numerical Analysis of Beam Loading

Spring 2024

Turn in a report detailing the solutions to each part of the project (use word or PDF format). Include all group names in the report. Store the report and all MATLAB files in a single zip archive. Name the zip archive "Project1_TEAMNAME.zip" and submit to blackboard. MATLAB's "publish" feature may be useful in constructing your report.

## Part 1. Method of Sections (10 points)

Consider the cantilever beam shown in Figure 1a. Use the method of sections (Fig. 1b) to determine the internal shear and moment of the beam at any position $x$. Follow the standard sign convention of downward shear force and anti-clockwise moment on a positive face both defined as positive. Recall that the distributed load on the rightmost segment may be described by the sum of a triangular load and a uniform load.




(a) Cantilevered Beam (b) Method of Sections

Figure 1: Cantilever Beam with a Distributed Load

Using MATLAB, create a function handle $P(x)$ which returns the magnitude of the distributed load for any position $x$ on the beam. Then create a method to calculate the internal shear force and moment in the beam for any position $x$. Plot the distributed load, internal shear force, and bending moment at one hundred equally spaced positions along the length of the beam. Label the horizontal and vertical axis of each plot. Include units in your label. Add a title to each plot. Then Use MATLAB's Publish feature to generate a report of the program.

# Part 2. Multiple Point Loads (10 points)

Consider a beam with a single point load. Using MATLAB, write a program to determine the internal shear and bending moment in the beam at any position $x$ given the magnitude and position of the point load. Test your code using the loading shown in Fig. 2a. Notice that the loading shown in Fig. 2a is a point load approximation of the distributed load show in Fig. 1a. Comment on the accuracy of using such an approximation (add your comments to the beginning of your code).


(a) Point Load (b) Multiple Point Loads

Figure 2: Cantilever Beam with Point Loads

Modify your code to use an arbitrary array of applied point loads as input. Using the loads shown in Figure 2b and listed in Table 1, find the internal shear and moment of the beam. Plot the point loads and internal shear and moment in the beam. The MATLAB code shown in Code 1 can be used to generate a quiver plot of loads on the beam. Publish your results using MATLAB's Publish feature.

| Load Position (m) | 2 | 4 | 6 | 7 |
|---|---|---|---|---|
| Load (kN) | -5.0 | -2.0 | 3.0 | -8.0 |

Table 1: Point Loads

```
L = 8;                              %Beam length (m)
P.x     = [ 2  4  6  7];            %Load x positions (m)
P.mag   = [-5 -2  3 -8];            %Load magnitudes (kN), downward (-)
%Plot Loads
quiver( P.x, 0*P.x, 0*P.mag, P.mag )
hold on
plot( [0, L], [0, 0], 'k' )
xlabel('Load Position (m)')
ylabel('Load Magnitude (kN)')
title('Point Loads')
```

Code 1: Point Load Plot

# Part 3. Numerical Analysis (20 points)


(a) Distributed Load (b) Multiple Point Load Approximation

Figure 3: Cantilever Beam with a Uniform Distributed Load

As shown in Figure 3, a distributed load may be approximated by decomposing it into multiple discrete point loads. The total force applied by a distributed load on any discrete beam segment $ab$ can be found by taking the definite integral of the distributed load over the beam segment (Eq. 1). By decomposing a beam into multiple segments of known length, this method can be used to find the point-load equivalent to the distributed load working on each beam segment.

$$P_{ab} = \int_a^b w(x)dx \tag{1}$$

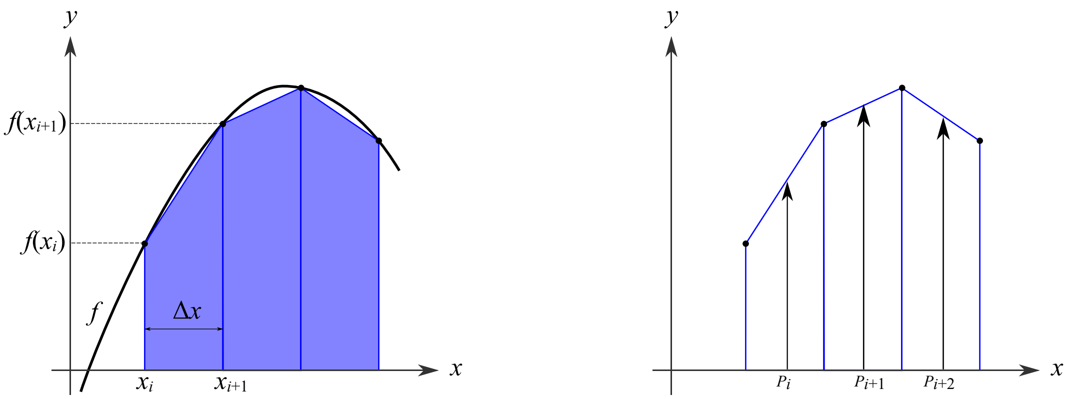
It is often convenient to numerically integrate a function. This allows for analysis of systems not well represented by analytical functions. The Trapezoidal rule (Eq. 2) is a first order numerical method for the approximation of the definite integral of a function. It calculates the area of a trapezoidal shape fit to a function (see Fig. 4a).

$$\int_a^b f(x)dx \approx (b-a) \cdot \frac{1}{2}[f(a) + f(b)] \tag{2}$$

When programming, it is useful to uniquely identify objects or specific elements of an assembly. Let us define $P_i$ as the load on the $i$th beam segment (where $i = 1 : N$) and assume a uniform beam segment width of $\Delta x$. It follows that the starting and ending positions of each beam segment are $x_i = (i-1) \cdot \Delta x$, $x_{i+1} = (i) \cdot \Delta x$. Applying the Trapezoidal rule to Eq. 1 and substituting these definitions yields Eq. 3. This equation provides a numerical approximation for the equivalent point load acting on each beam section.

$$P_i = \Delta x \cdot \frac{1}{2}[w(x_i) + w(x_{i+1})] \tag{3}$$

The principle of replacing a distributed load described by a continuous function by multiple discrete point loads is illustrated in Fig. 4. To best match the true distributed load, ideally each point load would be applied at the centroid of the integrated area. However for an approximate method it is sufficient to consider the load as acting on the midpoint of each beam segment, where $x_{p,i} = \frac{1}{2}(x_i + x_{i+1})$.




(a) Trapezoidal Rule

(b) Approximate Point Loads

Figure 4: Trapezoidal Rule Approximation of a Distributed Load

Using MATLAB, create a function handle which returns the distributed load shown in Fig. 3a. Code 2 shows one possible way to create such a function handle.

```
w = @(x) 0.*x + 2000; %Uniform Distributed Load of 2000 (N/m)
```

Code 2: Uniform Load Function Handle

## a)

Using the created function handle and Eq. 3, create an array of 100 point loads ($N = 100$) distributed uniformly across the beam and acting at the midpoint of each beam segment. Use your method from Part 2 to analyze the beam show in Fig. 3a. Plot the point loads and internal shear and moment in the beam. Publish your results using MATLAB's Publish feature. Verify that the results are correct.

## b)

Repeat this process for a 10m cantilever beam with an applied external load $w(x)$ as shown in Eq. 4, where $w_0' = 10$ kN/m and $L = 10$m. Publish your results using MATLAB's Publish feature.

$$w(x) = w_0' \sqrt{1 - \left(\frac{x}{L}\right)^2} \tag{4}$$