6

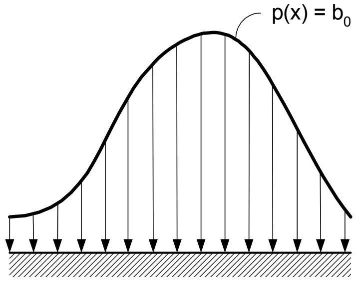
Reduction of a Simple Distributed Loading

Ref: Hibbeler § 4.10, Bedford & Fowler: Statics § 7.3

Loads are often not applied to specific points, but are *distributed* across a region. Design calculations can be simplified by finding a single equivalent force acting at a point; this is called *reduction of a distributed loading*. For loads distributed across a single direction (beam loading, for example) we need to find the both the magnitude of the equivalent force, and the position at which the equivalent force acts.

Example 1: Load Distribution Expressed as a Function of Position, x

The load on a beam is distributed as shown below:



where b_0 through b_4 are coefficients obtained by fitting a polynomial to data values (see Example 2). The x values range from 0.05 to 0.85 meters. The pressure is expressed in Pa. The values of the coefficients are tabulated below.

	$b \times 10^{-4}$
b_0	0.107
b_1	-1.68
b_2	11.9
b_3	-19.9
b_4	9.59

The data represent the pressure on the floor beneath a pile of 12-foot 2x4's, so y = 12 feet or 3.66 m.

Determine:

a) the two-dimensional loading function, w(x)

b) the magnitude and position of the equivalent force.

Solution 1

The two-dimensional loading function is obtained from the pressure function using the length of the boards, y = 3.66 m.

$$\begin{aligned} w(x) &= p(x)\, y \\ &= \left(b_0 + b_1 x + b_2 x^2 + b_3 x^3 + b_4 x^4\right)(3.66) \end{aligned}$$

The units on w are Pa·m.

The magnitude of the resultant force is obtained by integration.

$$F_R = \int_L w(x)\,dx$$

To perform this integration we first need to create an m-file for the loading function, w(x,y,b), and save it in the MATLAB search path. We then use the MATLAB quad() function to carry out the integration on the new function.

Note: For a full explanation of the quad() function type "help quad" at the command prompt.

```
function w = LoadingFunction(x, y, b)
%Saved as LoadingFunction.m in the MATLAB search path.

%Pressure Function (Pa)
p = b(1) + b(2).*x + b(3).*x.^2 + b(4).*x.^3 + b(5).*x.^4;

%Loading Function (Pa m)
w = p .* y;
```

```
» y = 3.66;                                                   % meters
» b = [ 1.07E3    -1.68E4    1.19E5    -1.99E5    9.59E4 ];
» F_R = quad('LoadingFunction', 0.05, 0.85, [ ], [ ], y, b)   %Newtons
F_R =
  6237
```

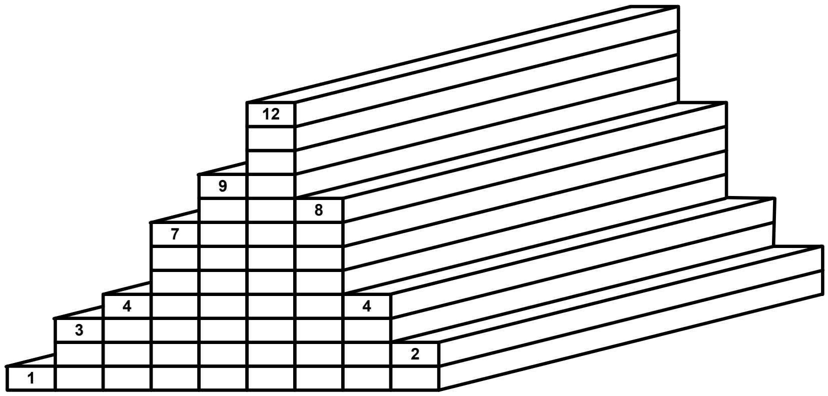
The location at which the resultant force acts is found by calculating the centroid of the area defined by w(x). We again need to create a new function to calculate the 1st moment of w(x,y,b).

```
function FM = FirstMoment(x, y, b)
%Saved as FirstMoment.m in the MATLAB search path.

%Pressure Function (Pa)
p = b(1) + b(2).*x + b(3).*x.^2 + b(4).*x.^3 + b(5).*x.^4;

%Loading Function (Pa m)
w = p .* y;

FM = w .* x;
```

```
» x_loc = quad('MeanForce', 0.05, 0.85, [ ], [ ], y, b) ./ F_R     %meters
x_loc =
    0.4927
```

Example 2: Tabulated Load Distribution

The function used in Example 1 was obtained by polynomial regression of a set of data values, representing the pressure acting on the floor beneath a pile of 2x4 (inch, or 50x100 mm) lumber.



Using the density of wood (approximately 700 kg/m^3) and the lumber dimensions, the pressure exerted by the wood on the floor at the bottom of each stack of lumber was calculated (see table). The reported x value represents the position of the center of each column of boards.

x (m)	p (Pa)
0.05	343
0.15	1029
0.25	1372
0.35	2401
0.45	3087
0.55	4116
0.65	2744
0.75	1372
0.85	686

Use the tabulated pressure, position data to estimate the magnitude and position of the equivalent force.

Solution 2

w values can be calculated from pressure values, as before:

$$w = p \cdot y$$

```
» x = [ 0.05 0.15 0.25 0.35 0.45 0.55 0.65 0.75 0.85];              %meters
» p = [  343 1029 1372 2401 3087 4116 2744 1372  686];               %Pascals
» w = p .* y                                                          %N/m
w =
  1.0e+004 *
  Columns 1 through 9
    0.1255    0.3766    0.5022    0.8788    1.1298    1.5065    1.0043    0.5022    0.2511
```

The integral for force is approximated as a summation:

$$F_R \approx \sum_{i=1}^{N} w_i \, \Delta x$$

where Δx is the width of a stack (0.10 m) and N is the number of stacks (9).

```
» delta_x = 0.10;                                  %meters
» F_R = sum(w .* delta_x)                          %Newtons
F_R =
 6277
```

Similarly, the centroid is approximated as follows:

```
» x_loc = sum(x .* w .* delta_x) / F_R             %meters
x_loc =
   0.4900
```

Annotated MATLAB Script Solution

```
function w = LoadingFunction(x, y, b)
%Saved as LoadingFunction.m in the MATLAB search path.

%Pressure Function (Pa)
p = b(1) + b(2).*x + b(3).*x.^2 + b(4).*x.^3 + b(5).*x.^4;

%Loading Function (Pa m)
w = p .* y;
```

```
function mean = FirstMoment(x, y, b)
%Saved as FirstMoment.m in the MATLAB search path.

%Pressure Function (Pa)
p = b(1) + b(2).*x + b(3).*x.^2 + b(4).*x.^3 + b(5).*x.^4;

%Loading Function (Pa m)
w = p .* y;

mean = w .* x;
```

```
%Length of Boards (meters)
y = 3.66;

%Coefficients obtained by fitting a polynomial to data values
b = [ 1.07E3 -1.68E4 1.19E5 -1.99E5 9.59E4 ];


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                          Solution 1                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The magnitude of the resultant force is calculated.
F_R = quad('LoadingFunction',0.05,0.85,[],[], y, b);
fprintf('\nThe magnitude of the resultant force is = %1.0f N\n', F_R)

%The location of the resultant force is calculated.
x_loc = quad(' FirstMoment',0.05,0.85,[],[], y, b) ./ F_R;
fprintf('The location of the resultant force is = %1.2f m\n', x_loc)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                 Solution 2 Using Tabulated Data                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x = [ 0.05 0.15 0.25 0.35 0.45 0.55 0.65 0.75 0.85]; %meters
p = [  343 1029 1372 2401 3087 4116 2744 1372  686]; %Pa

%Calculate w values from pressure values using the length of the boards.
w = p .* y;

%Calculate the magnitude of the resultant force (the use
%     of a summation instead of an integration
%     makes this an approximate result).
delta_x = 0.10; % meters
F_R = sum(w .* delta_x);
fprintf('\nThe approximated magnitude of the resultant force is = %1.0f N\n', F_R)

%Calculate the location of the resultant force
x_loc = sum(x .* w .* delta_x) / F_R;
fprintf('The approximated location of the resultant force is = %1.2f m\n', x_loc)
```

A Final Note

Both of these solutions are approximations. The second method approximates the integrals using summations, while the first method approximates the solution because the polynomial regression is a "best fit" but not a perfect fit to the data. In the graph shown here, the circles represent the tabulated values in Example 2, while the pressures predicted by the polynomial are indicated by the line.