
```

%Ibrahim Al-Akash, CAAM 210, Spring 2022, Project 07: Bridge
%bridge.m
%Driver for modeling the deformation of a bridge
%Last modified: March 14, 2022

function bridge_1_driver()
% This driver computes the adjacency matrix and plots the sparsity
  pattern.
% It also computes and plots undeformed bridge and computes and plots
  the
% deformed bridge.

for i = 1:7
    build_load_plot_basic_bridge(i, 0.01);
    build_load_plot_basic_bridge(i, 0.05);
end

% ANSWERS TO QUESTIONS:
% 1) The maximum number of sections for the bridge to withstand a car
  of
% 0.01 weight is 3. My criterion for safety is the bridge must have a
  work
% less than or equal to 0.01. After 3 sections, the bridges exceed
  this
% threshold of work. I used this threshold since it is the same number
  as
% the weight of the car.

% 1) The maximum number of sections for the bridge to withstand a
  truck of
% 0.05 weight is 1. My criterion for safety is the bridge must have a
  work
% less than or equal to 0.05. After 1 section, the bridges exceed this
% threshold of work. I used this threshold since it is the same number
  as
% the weight of the car.

% 3) More points on the spy indicate more deformation of the bridge.
% A spy that is more sparse indicates the bridge can withstand more
  weight,
% and is more stable. A spy that has a high concentration of points
% indicates there are many points of deformation in the bridge, so it
  is
% less stable and more likely to collapse.

% 4) As nos increases, the spy contains more points and they are more
% concentrated. This makes sense, since adding more sections to the
  bridge
% will result in more points of failure and less rigidity.

end

```

```

function [adj, xc, yc, len] = build_basic_bridge(nos)
% compute the adjacency matrix and store it as adj
% compute the x and y coordinates of two ending points of all fibers
and
% store it as xc,yc
% compute lengths of all fibers and store it as len
% INPUTS:
%   nos - number of sections
% OUTPUTS:
%   adj - adjacency matrix
%   xc - x-coordinates of starting and ending points of fibers
%   yc - y-coordinates of starting and ending points of fibers
%   len - lengths of all fibers

beams = 5*nos + 5;
nodes = 2*nos + 2;
adj = zeros(beams, 2*nodes);
len = ones(beams, 1);
xc = zeros(beams,2);
yc = zeros(beams,2);

[rows, columns] = size(adj);

% Hardcode starting entries
adj(1,1) = 1;
adj(2,[3,4]) = [1/sqrt(2), 1/sqrt(2)];
len(2) = sqrt(2);
xc(1,:)= [0 1];
yc(1,:)= [0 0];
xc(2,:)= [0 1];
yc(2,:)= [0 1];

% Program the section entries
for i = 0:nos-1
    r = 5*i + 3;
    c = 4*i + 1;

    adj(r+0,[c+0, c+1, c+2, c+3])=[0 -1 0 1];
    adj(r+1,[c+2, c+3, c+4, c+5])=[-1/sqrt(2) 1/sqrt(2) 1/sqrt(2) -1/
sqrt(2)];
    adj(r+2,[c+2, c+3, c+6, c+7])=[-1 0 1 0];
    adj(r+3,[c+0, c+1, c+6, c+7])=[-1/sqrt(2) -1/sqrt(2) 1/sqrt(2) 1/
sqrt(2)];
    adj(r+4,[c+0, c+1, c+4, c+5])=[-1 0 1 0];

    len(r+1) = sqrt(2);
    len(r+3) = sqrt(2);

    xc(r+0,:)= [i+1 i+1];
    xc(r+1,:)= [i+1 i+2];
    xc(r+2,:)= [i+1 i+2];
    xc(r+3,:)= [i+1 i+2];
    xc(r+4,:)= [i+1 i+2];
    yc(r+0,:)= [0 1];

```

```

        yc(r+1,:)= [1 0];
        yc(r+2,:)= [1 1];
        yc(r+3,:)= [0 1];
        yc(r+4,:)= [0 0];
    end

    % Hardcode last entries
    adj(rows-2, [columns-3, columns-2, columns-1, columns]) = [0 -1 0 1];
    adj(rows-1, [columns-3, columns-2, columns-1, columns]) = [0 0 -1/
sqrt(2) 1/sqrt(2)];
    adj(rows, [columns-3, columns-2, columns-1, columns]) = [-1 0 0 0];
    len(beams-1) = sqrt(2);
    xc(beams-2,:) = [nos+1 nos+1];
    xc(beams-1,:) = [nos+1 nos+2];
    xc(beams,:) = [nos+1 nos+2];
    yc(beams-2,:) = [0 1];
    yc(beams-1,:) = [1 0];
    yc(beams,:) = [0 0];

end

function [dx,dy,work,X,Y] =
    deform_basic_bridge(nos,adj,xc,yc,len,car_weight)
% compute displacements of all fibers X,Y
% update the coordinates of two ending points of all fibers dx,dy
% compute the work as displacements*force
% INPUTS:
%   nos - number of sections
%   adj - adjacency matrix
%   xc - x-coordinates prior to displacement
%   yc - y-coordinates prior to displacement
%   len - lengths of fibers
%   car_weight - weight of the car applying stress to the bridge
% OUTPUTS:
%   dx - deformation of x
%   dy - deformation of y
%   work - work done on the bridge
%   X - helper matrix associating each node with horizontal
        displacement
%   Y - helper matrix associating each node with vertical displacement

F = loading_vector(nos, car_weight);

D = adj'*diag(1./len)*adj\F;
work = D'*F;
dx = D(1:2:end);
dy = D(2:2:end);

X = zeros(size(xc));
Y = zeros(size(yc));

X(1,:) = xc(1,:) + [0 dx(1)];
Y(1,:) = yc(1,:) + [0 dy(1)];
X(2,:) = xc(2,:) + [0 dx(2)];

```

```

Y(2,:) = yc(2,:) + [0 dy(2)];

for n = 0:nos-1
    r = 5*n + 3;
    X(r+0,:)=xc(r+0,:)+[dx(1+2*n) dx(2+2*n)];
    Y(r+0,:)=yc(r+0,:)+[dy(1+2*n) dy(2+2*n)];
    X(r+1,:)=xc(r+1,:)+[dx(2+2*n) dx(3+2*n)];
    Y(r+1,:)=yc(r+1,:)+[dy(2+2*n) dy(3+2*n)];
    X(r+2,:)=xc(r+2,:)+[dx(2+2*n) dx(4+2*n)];
    Y(r+2,:)=yc(r+2,:)+[dy(2+2*n) dy(4+2*n)];
    X(r+3,:)=xc(r+3,:)+[dx(1+2*n) dx(4+2*n)];
    Y(r+3,:)=yc(r+3,:)+[dy(1+2*n) dy(4+2*n)];
    X(r+4,:)=xc(r+4,:)+[dx(1+2*n) dx(3+2*n)];
    Y(r+4,:)=yc(r+4,:)+[dy(1+2*n) dy(3+2*n)];
end

beams = 5*nos + 5;
X(beams-2,:) = xc(beams-2,:) + [dx(end-1) dx(end)];
X(beams-1,:) = xc(beams-1,:) + [dx(end) 0];
X(beams,:) = xc(beams,:) + [dx(end-1) 0];
Y(beams-2,:) = yc(beams-2,:) + [dy(end-1) dy(end)];
Y(beams-1,:) = yc(beams-1,:) + [dy(end) 0];
Y(beams,:) = yc(beams,:) + [dy(end-1) 0];

end

function build_load_plot_basic_bridge(nos, car_weight)
% invoke build_basic_bridge, deform_basic_bridge, plot_bridge, plot_spy
% to build, deform and plot the bridge, plot the adjacency matrix
% INPUTS:
%   nos - number of sections
%   car_weight - weight of the car applying stress to the bridge
% OUTPUTS:
%   None

[adj, xc, yc, len] = build_basic_bridge(nos);
figure
plot_spy(adj, nos);

figure
plot_bridge(nos,xc,yc);
tit = sprintf("%d-Section Bridge with Zero Load", nos);
title(tit);

[dx,dy,work,X,Y] = deform_basic_bridge(nos,adj,xc,yc,len,car_weight);

figure
plot_bridge(nos,X,Y);
tit = sprintf("%d-Section Deformed Bridge", nos);
subtit = sprintf("Car Weight: %0.2f, Work: %f", [car_weight, work]);
title(tit);
subtitle(subtit);
end

```

```

function plot_bridge(nos,xc,yc)
% plot a bridge using "fill" command
% INPUTS:
%   nos - number of sections
%   xc - the x-coordinates of bridge points
%   yc - the y-coordinates of bridge points
% OUTPUTS:
%   None

% Fill in pylons
x1fill = [-1 0 0.5 -1];
y1fill = [1 1 0 0];
y1fill = y1fill-1;
x2fill = [nos+2 nos+3 nos+3 nos+1.5];
y2fill = [1 1 0 0];
y2fill = y2fill-1;

% Plot the bridge
hold on
line(xc',yc',"Color","b");
fill(x1fill, y1fill, "k");
fill(x2fill, y2fill, "k");
xlim([-1, nos+3]);
ylim([-1,1.1])
set(gca,'XTick',[]);
set(gca,'YTick',[]);
set(gca,'Color',[214/255 214/255 214/255]);

end

function plot_spy(adj,nos)
% plot the sparsity pattern of the adjacency matrix
% INPUTS:
%   adj - adjacency matrix
%   nos - number of sections
% OUTPUTS:
%   None

spy(adj);
tit = sprintf("%d-Section Adjacency Matrix", nos);
title(tit);
xlabel("Nonzero Matrix Columns");
ylabel("Non-zero Matrix Rows");

end

function [F] = loading_vector(nos, car_weight)
% This function finds the force vector on the bridge
% INPUTS:
%   nos - number of sections
%   car_weight - weight of the car applying stress to the bridge
% OUTPUTS:

```

```

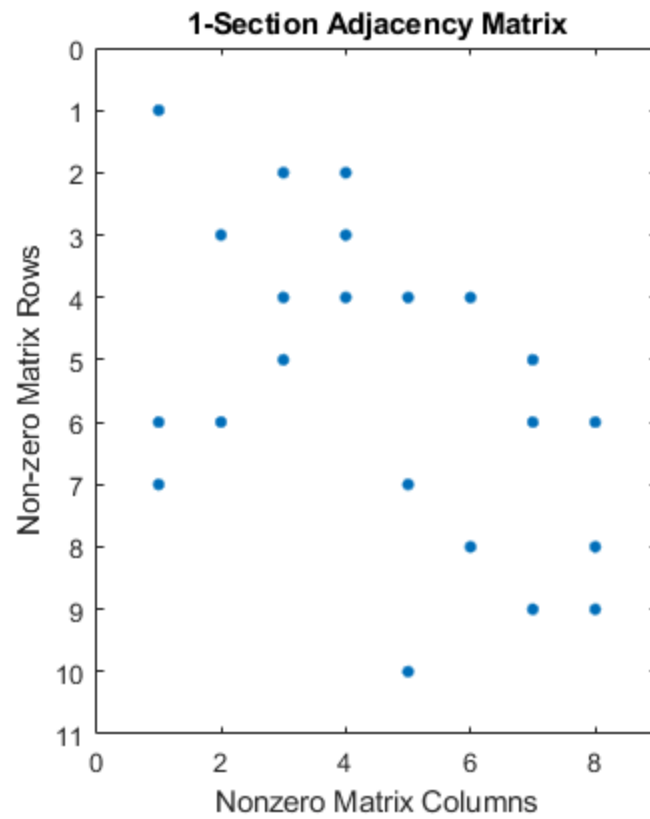
% None

nodes = nos*2+2;
F = zeros(nodes*2,1);

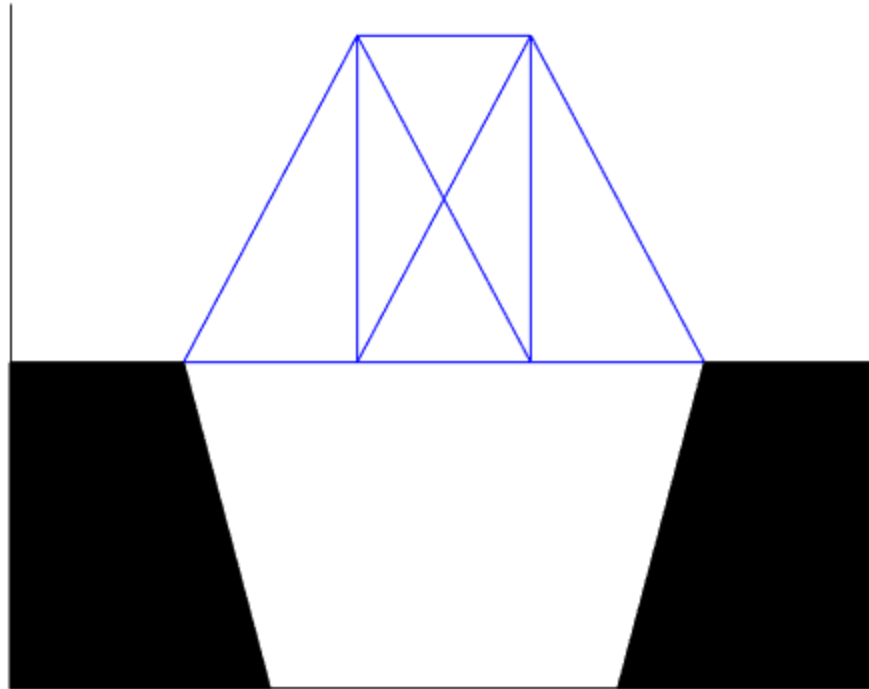
n = 2;
while(n<length(F))
    F(n) = -car_weight;
    n = n+4;
end

end

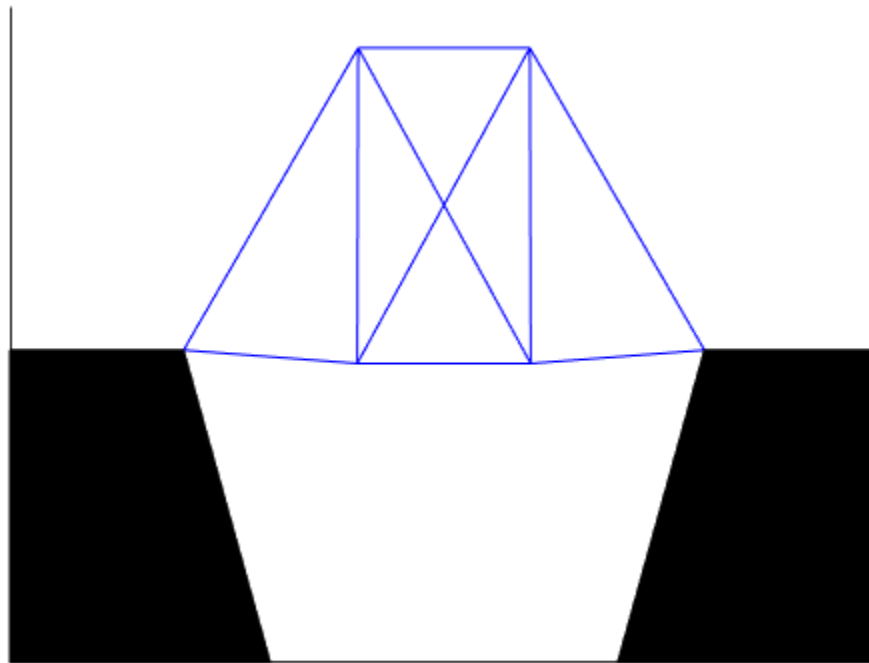
```

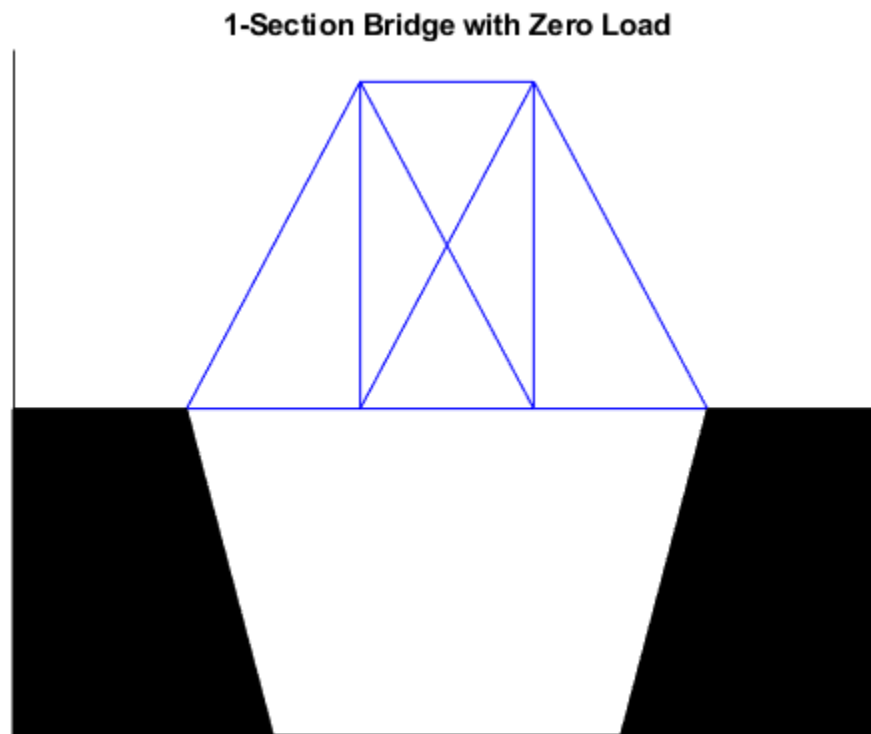
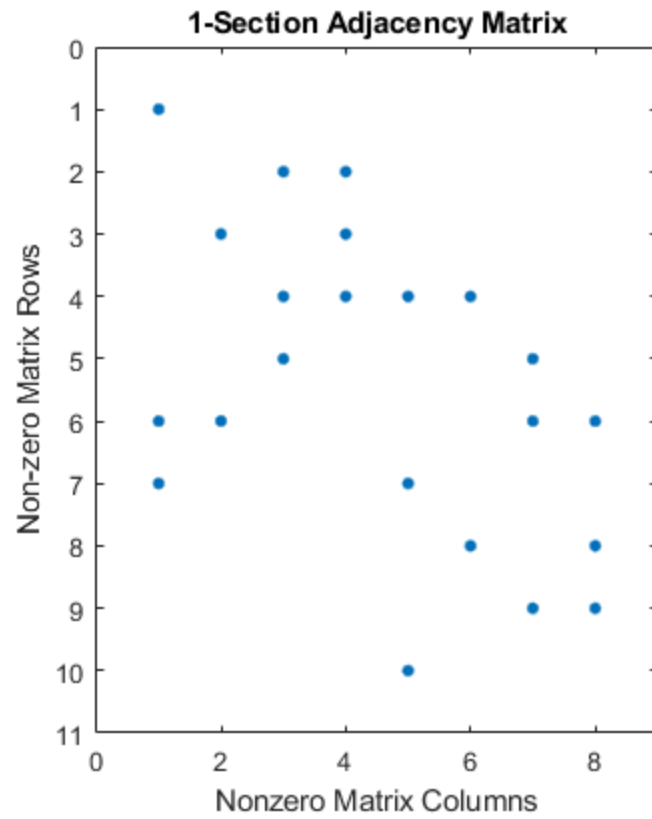


1-Section Bridge with Zero Load

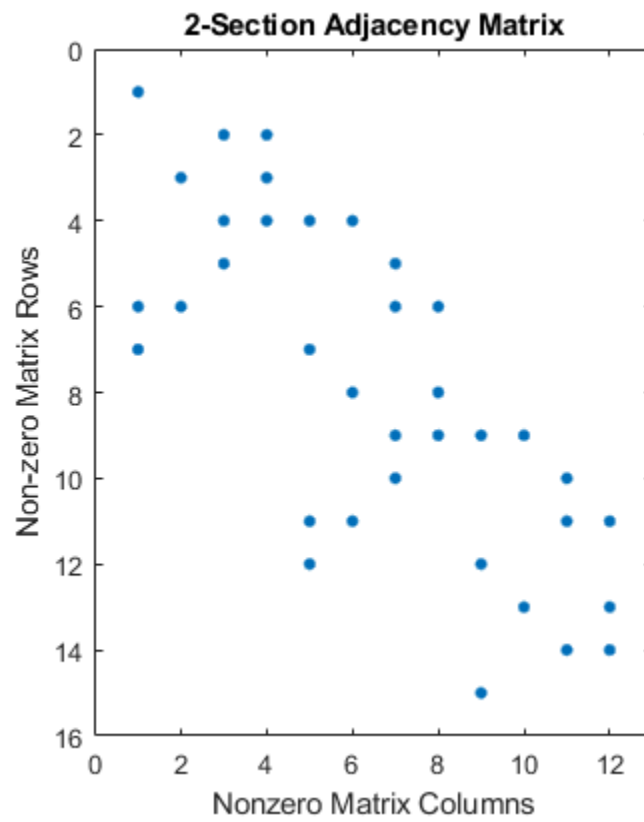
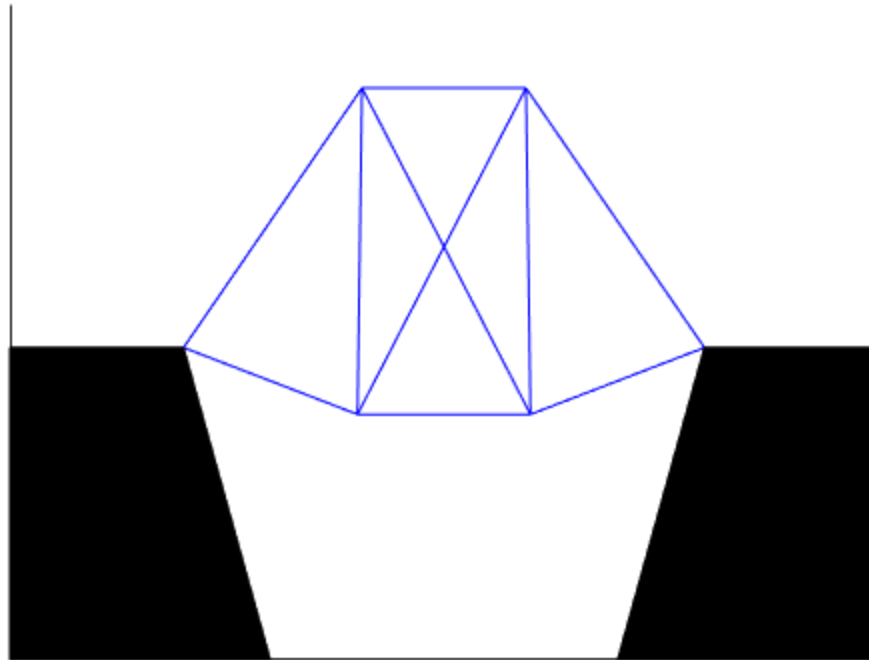


1-Section Deformed Bridge
Car Weight: 0.01, Work: 0.000855

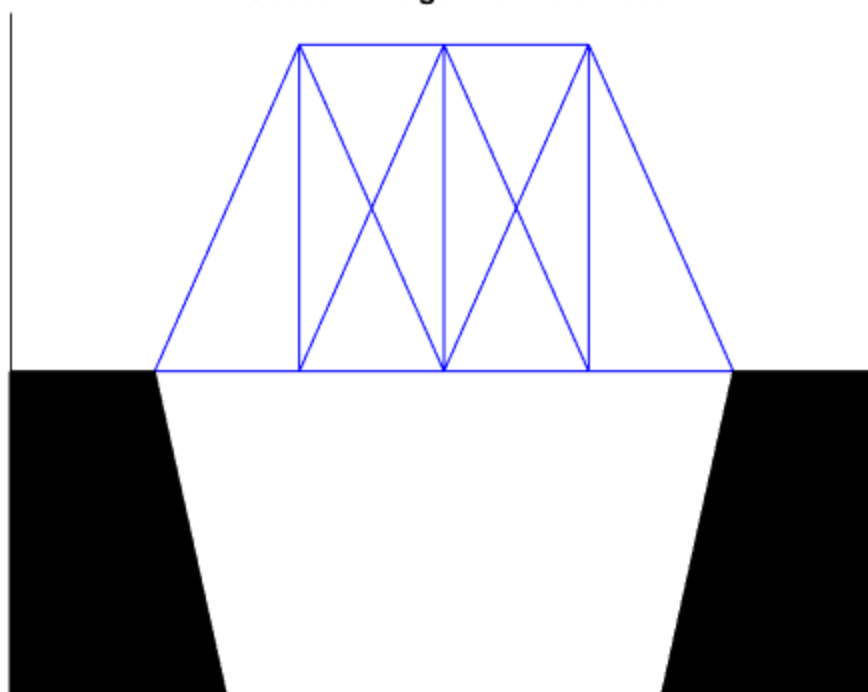




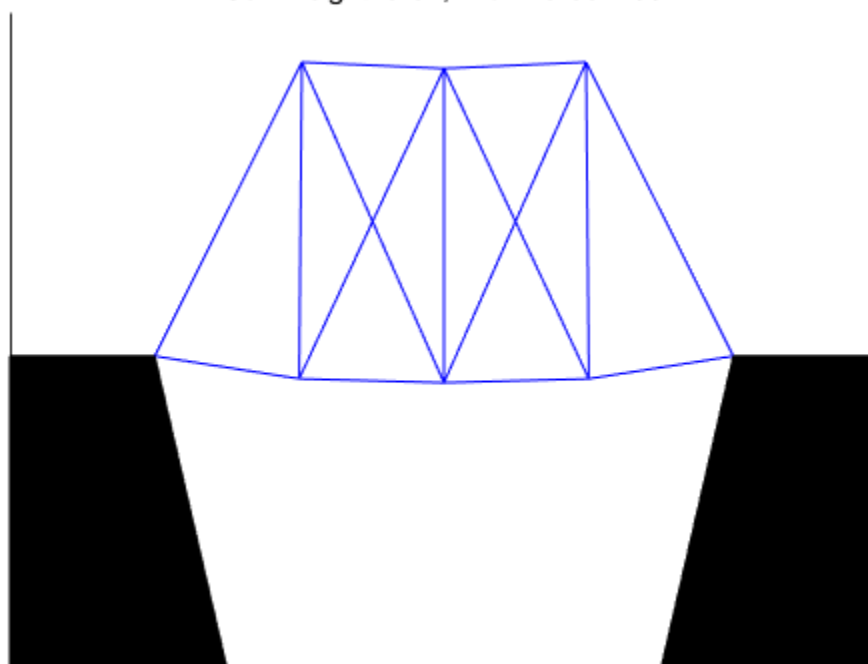
1-Section Deformed Bridge
Car Weight: 0.05, Work: 0.021374

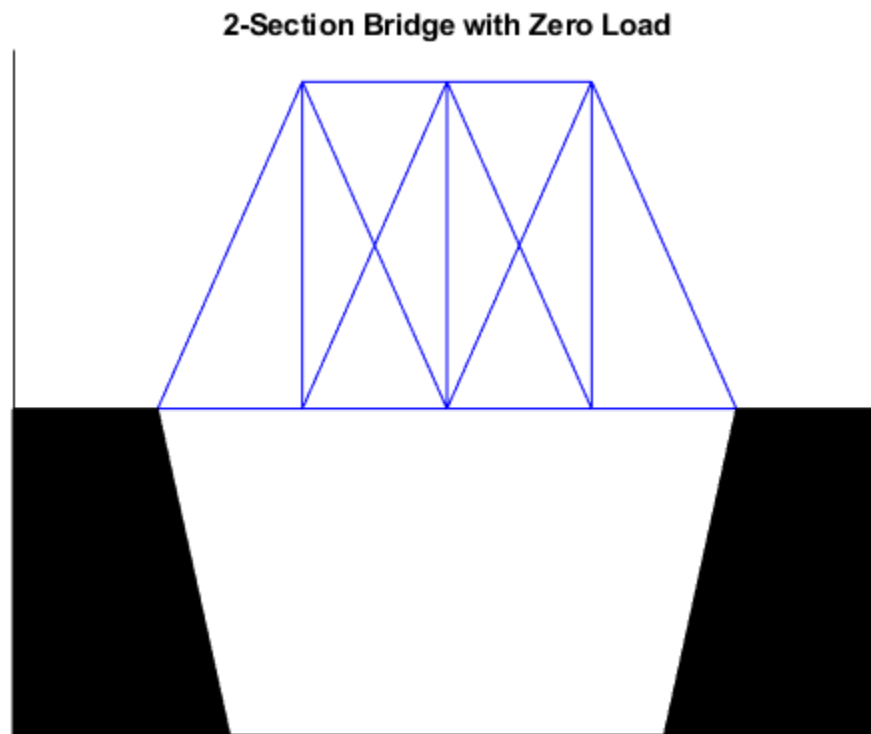
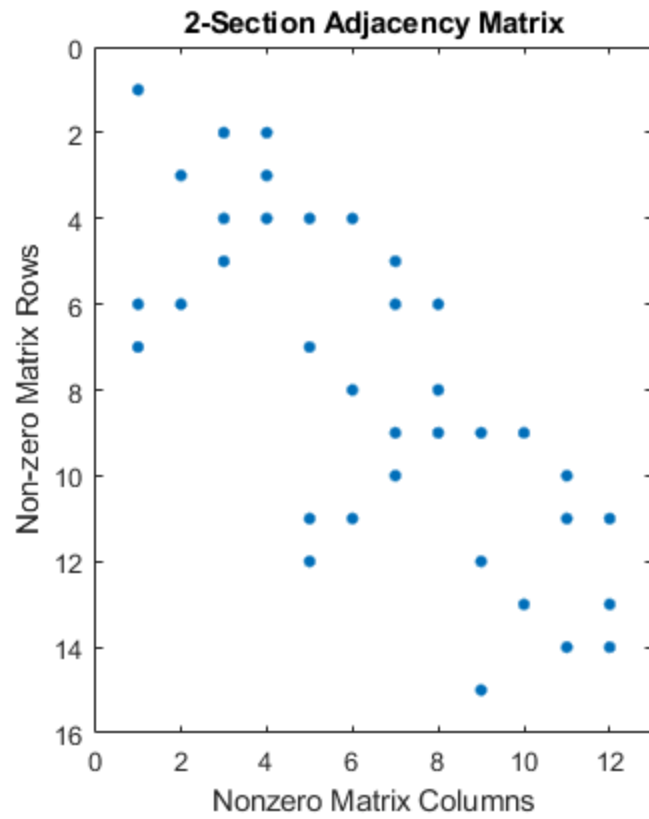


2-Section Bridge with Zero Load

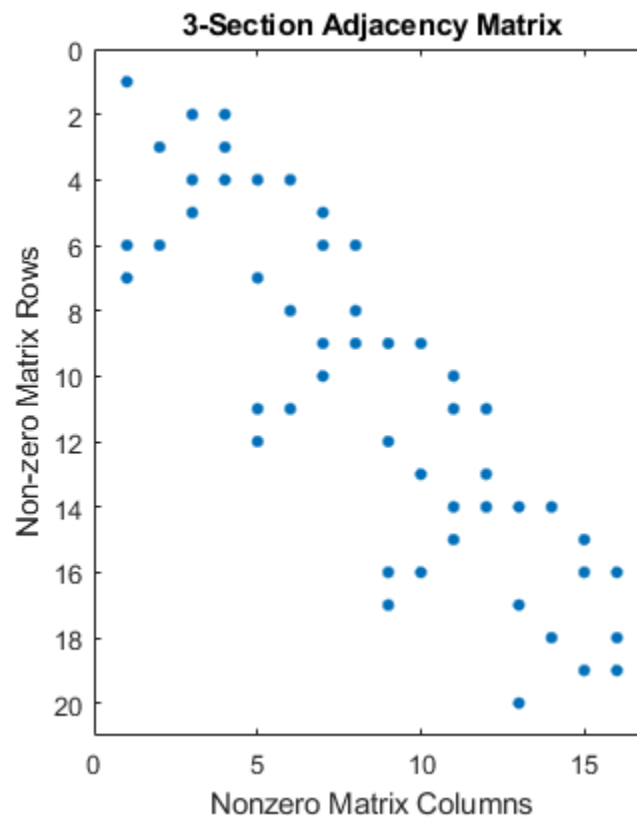
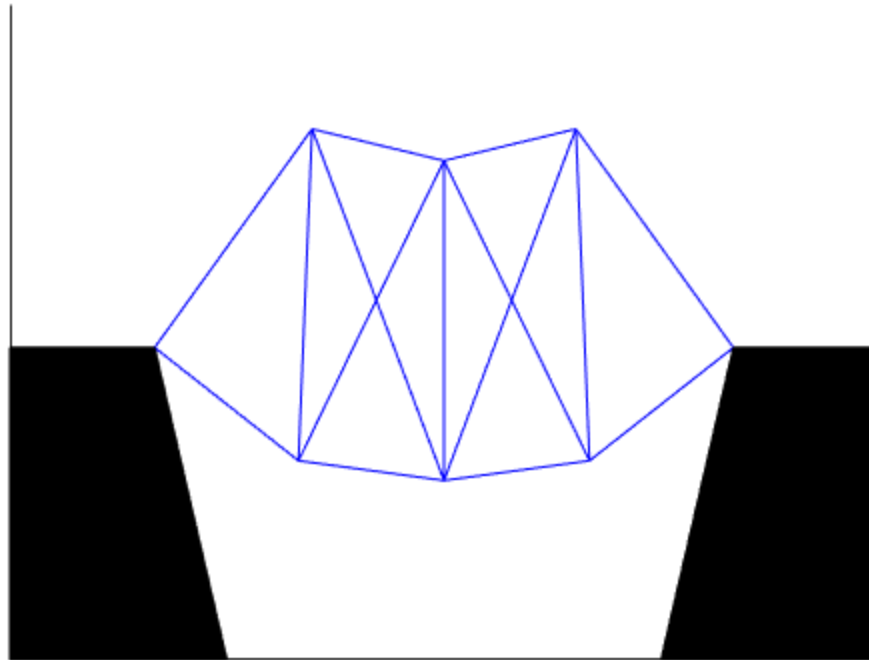


2-Section Deformed Bridge
Car Weight: 0.01, Work: 0.002299

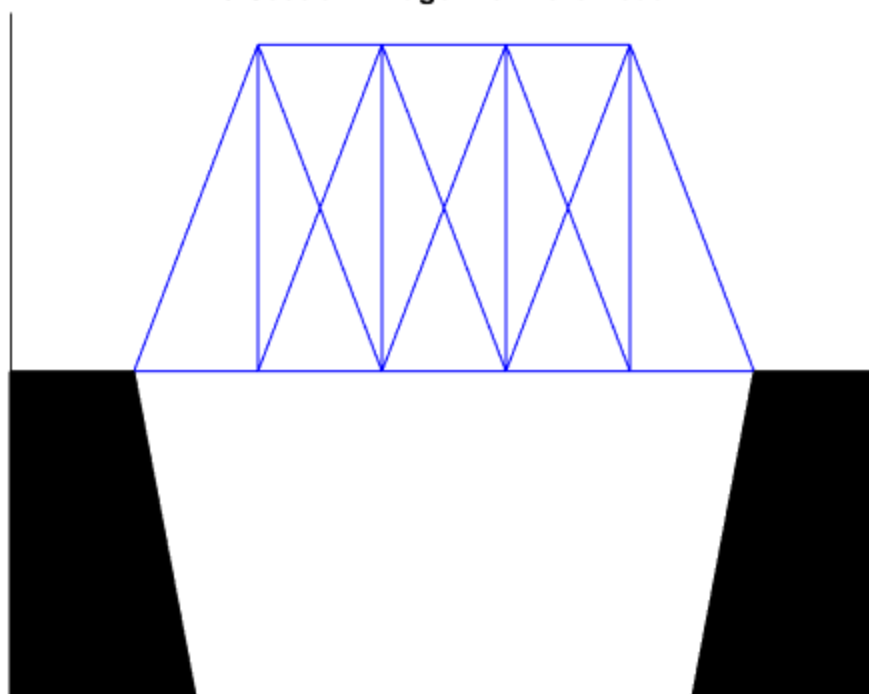




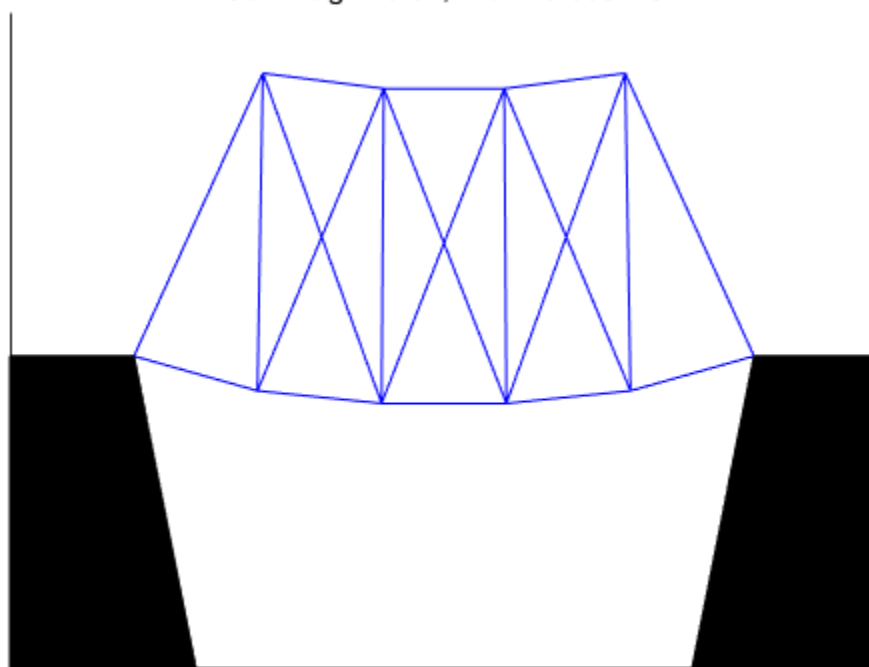
2-Section Deformed Bridge
Car Weight: 0.05, Work: 0.057486

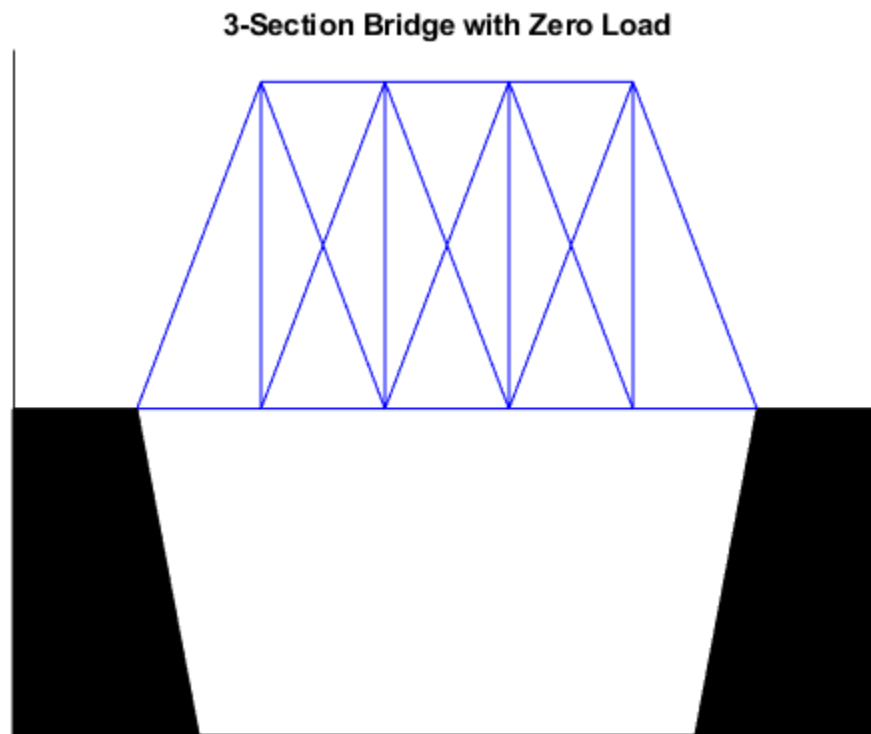
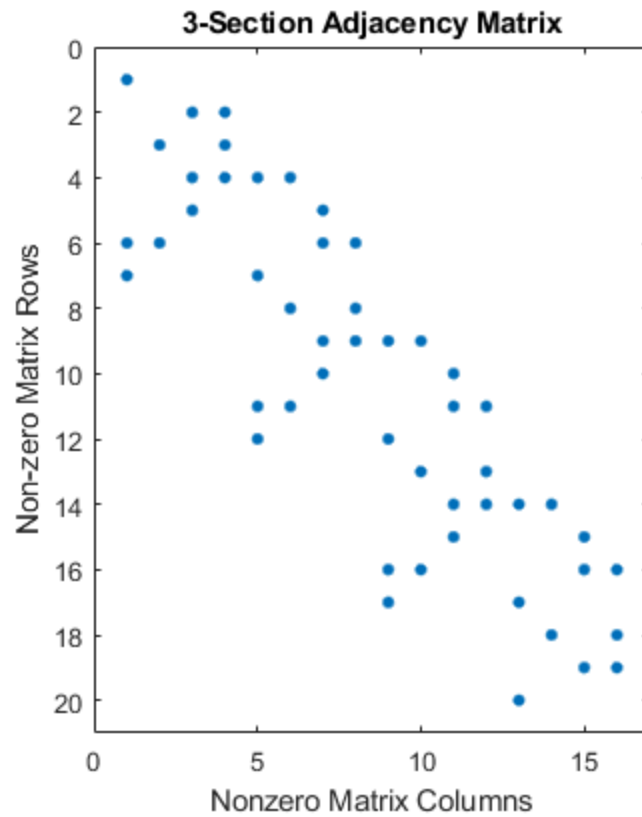


3-Section Bridge with Zero Load

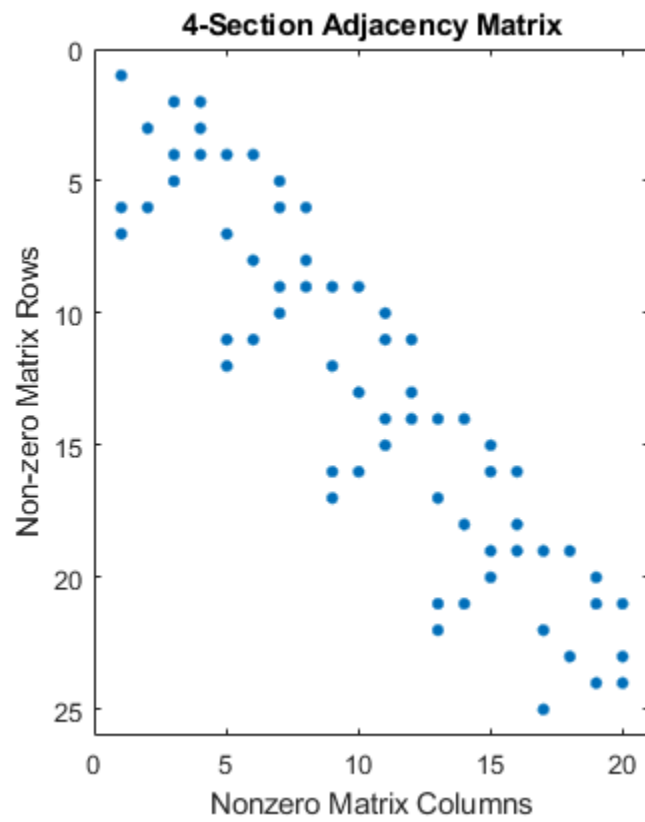
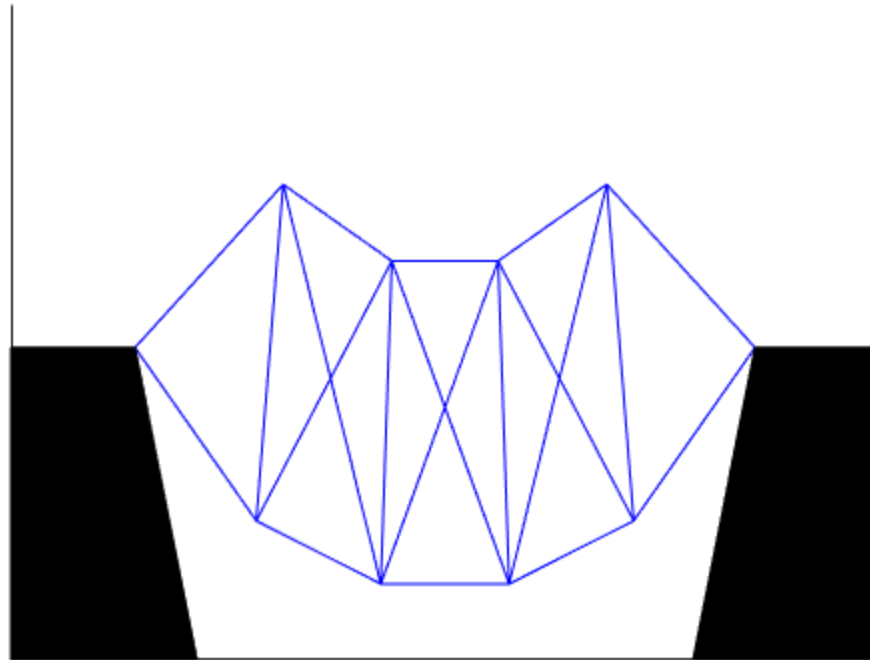


3-Section Deformed Bridge
Car Weight: 0.01, Work: 0.005243

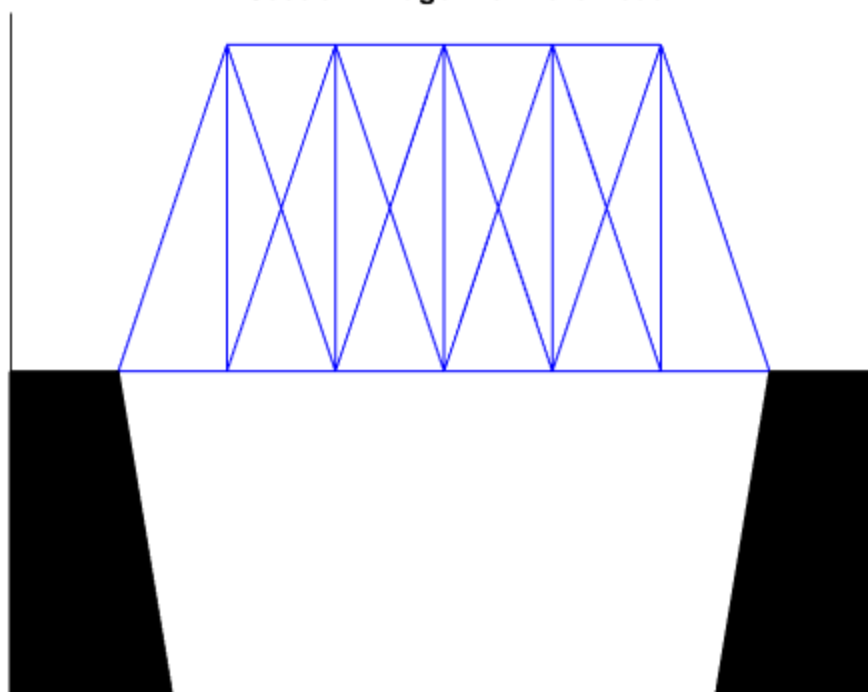




3-Section Deformed Bridge
Car Weight: 0.05, Work: 0.131078

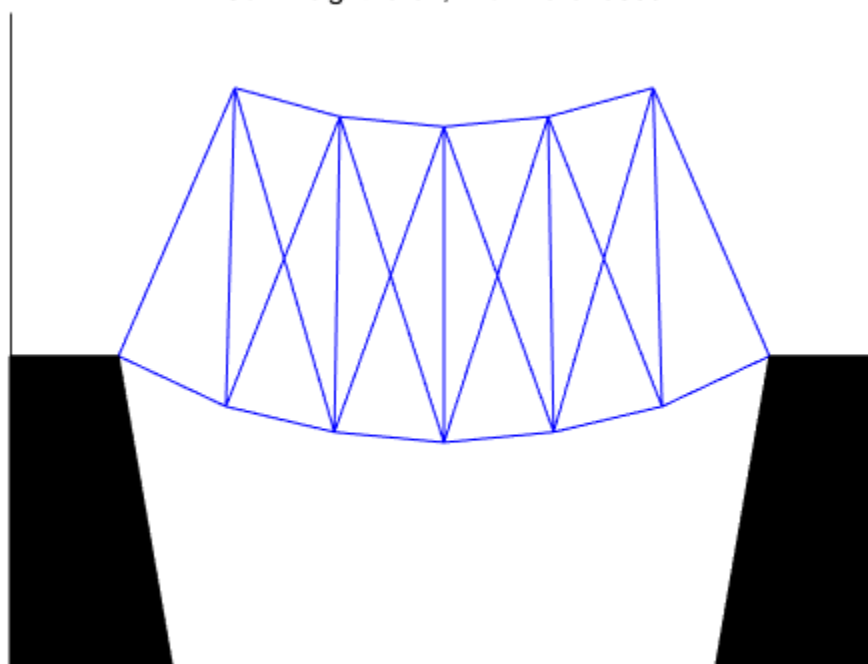


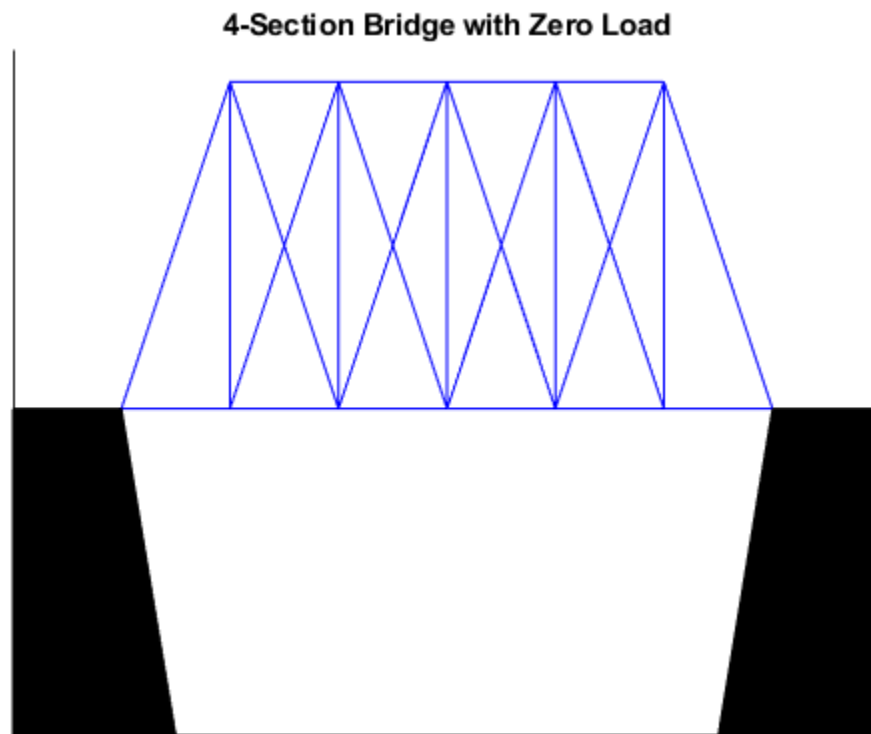
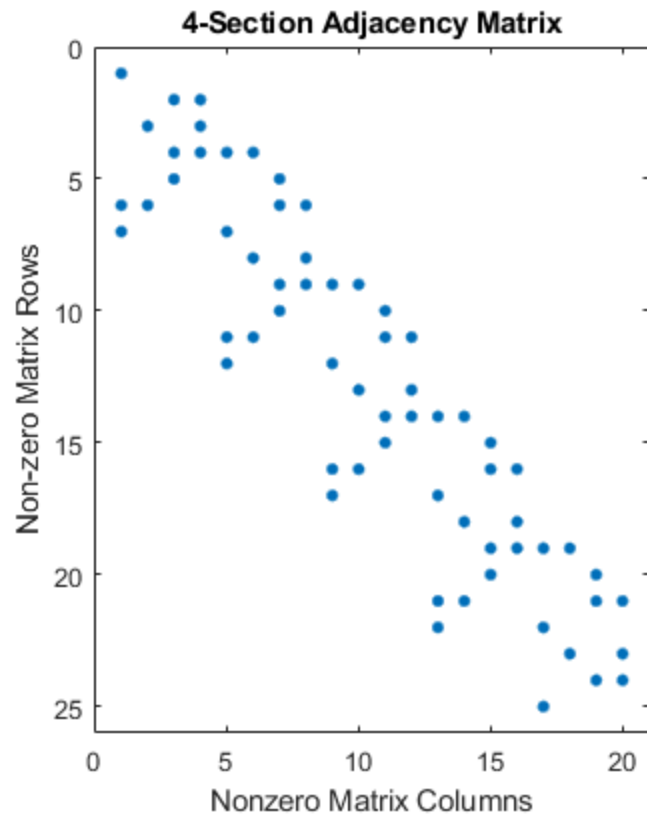
4-Section Bridge with Zero Load



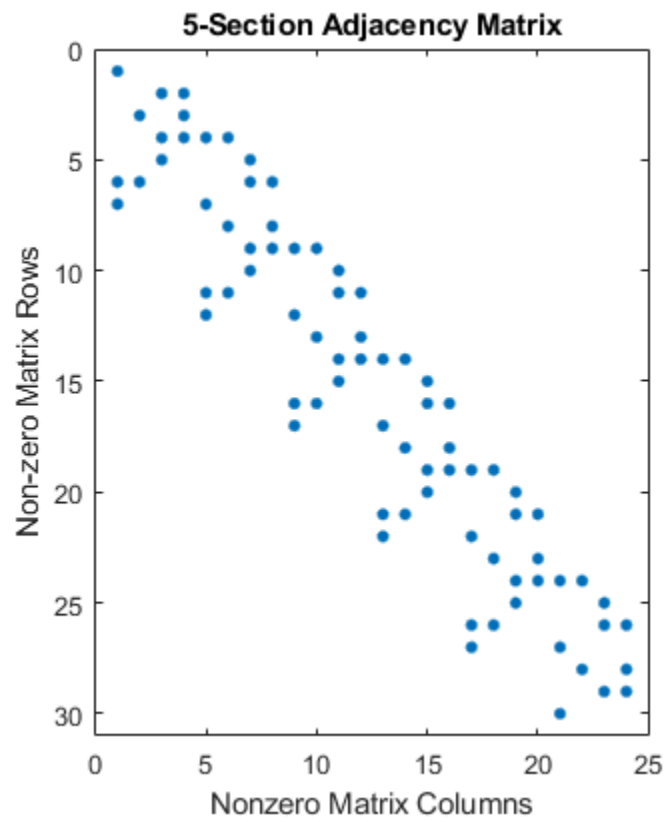
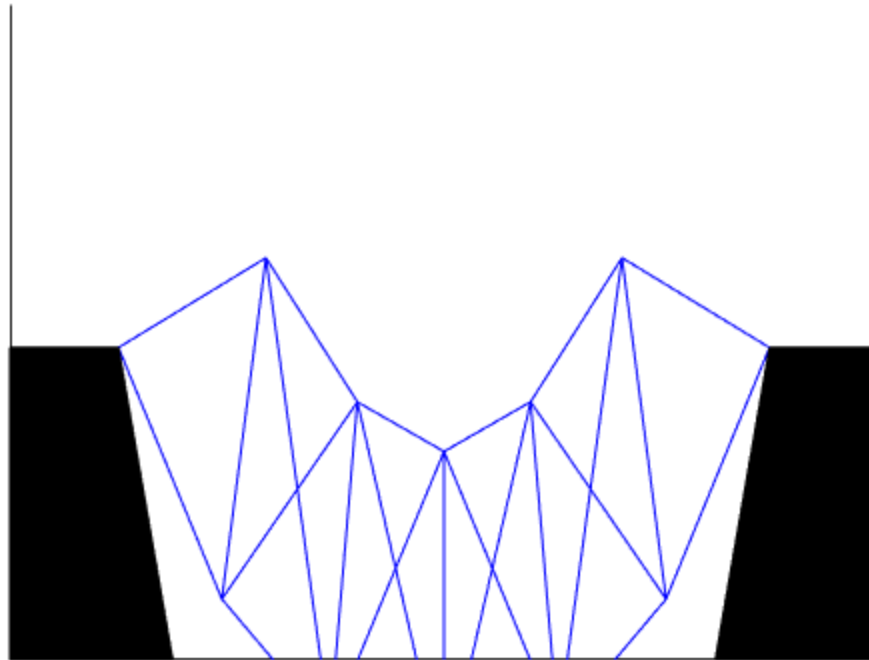
4-Section Deformed Bridge

Car Weight: 0.01, Work: 0.010839

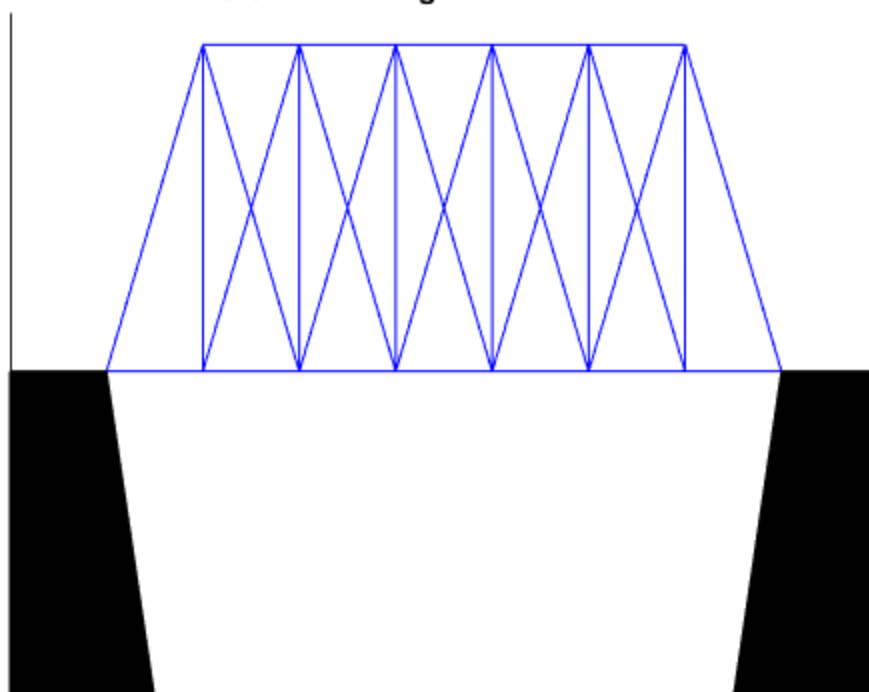




4-Section Deformed Bridge
Car Weight: 0.05, Work: 0.270981

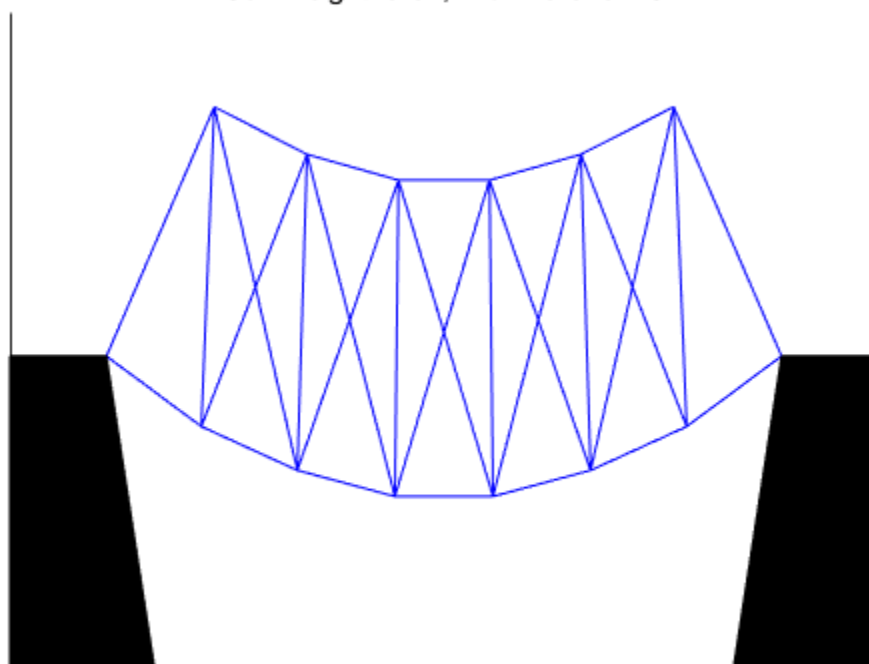


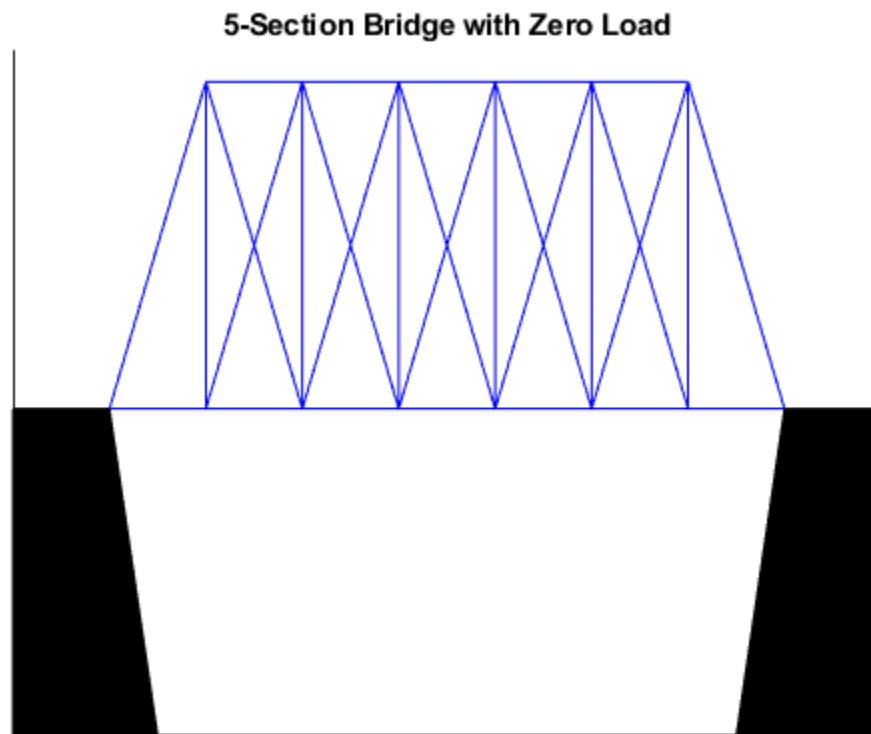
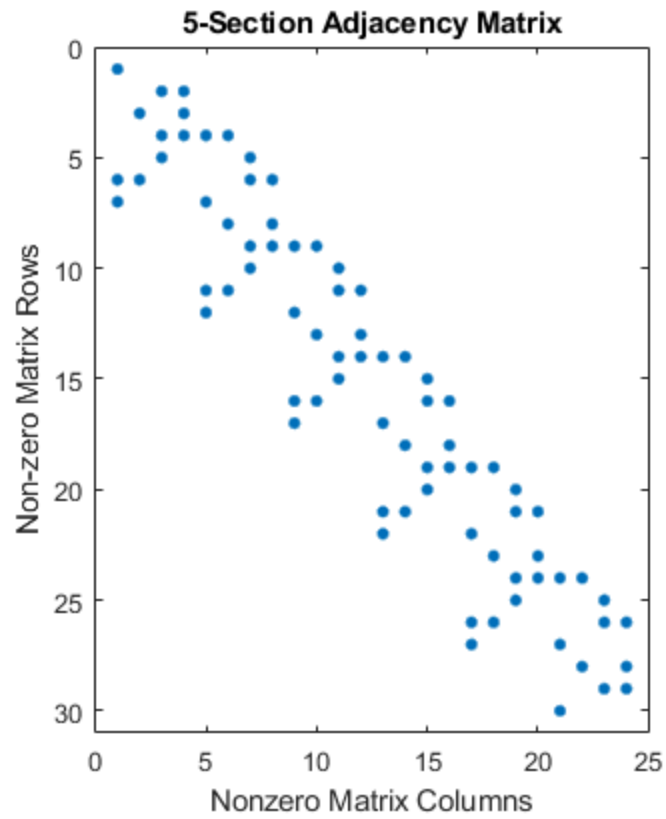
5-Section Bridge with Zero Load



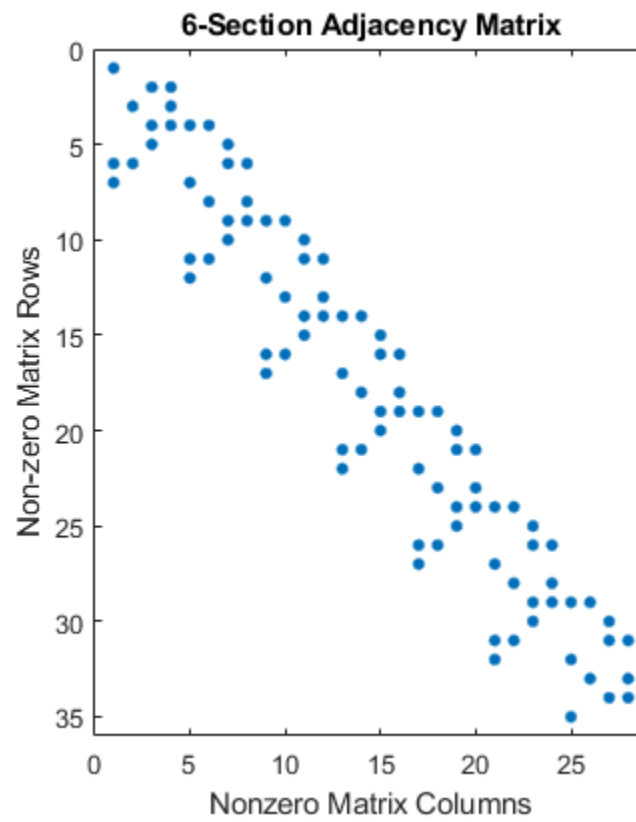
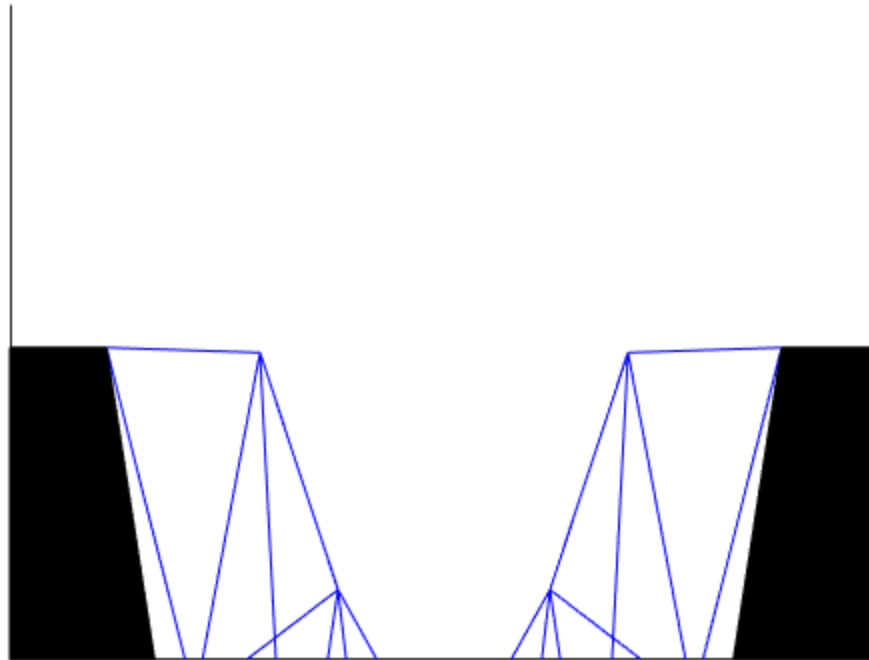
5-Section Deformed Bridge

Car Weight: 0.01, Work: 0.020778

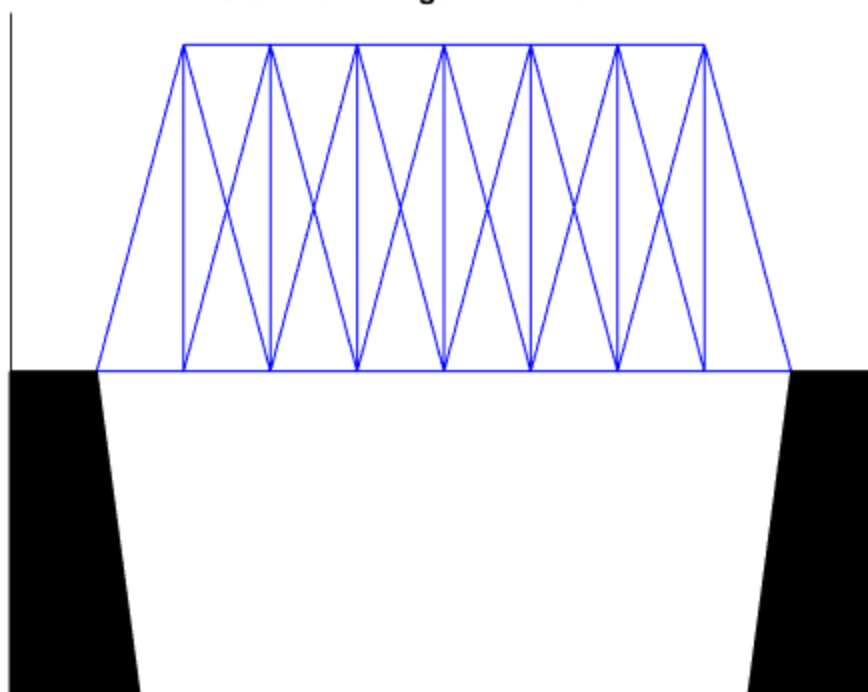




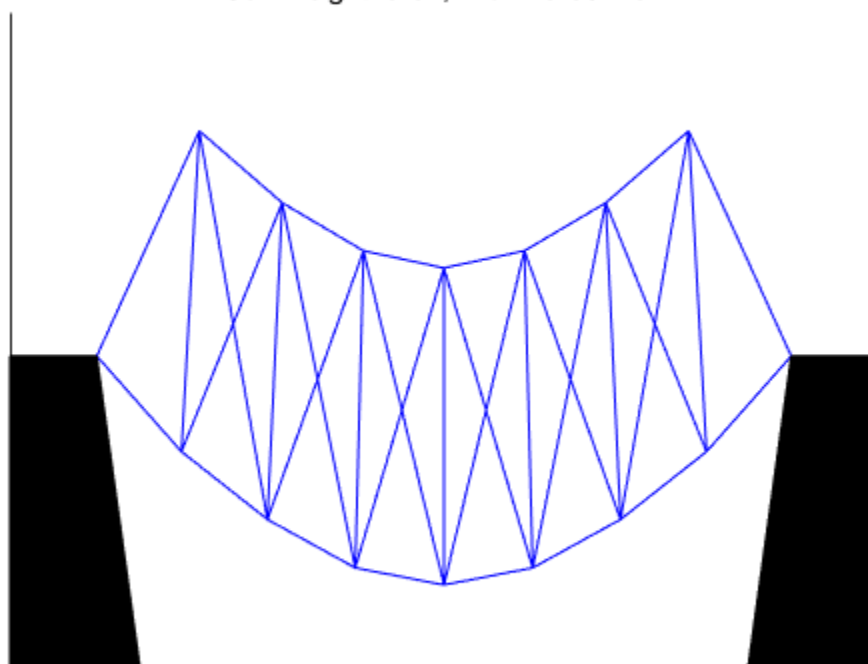
5-Section Deformed Bridge
Car Weight: 0.05, Work: 0.519445

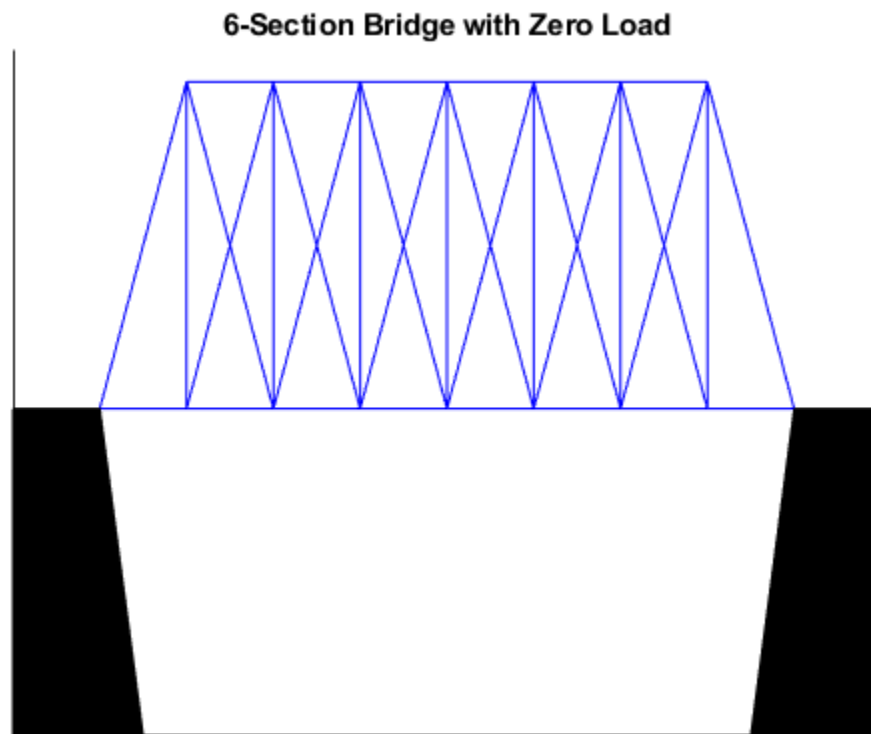
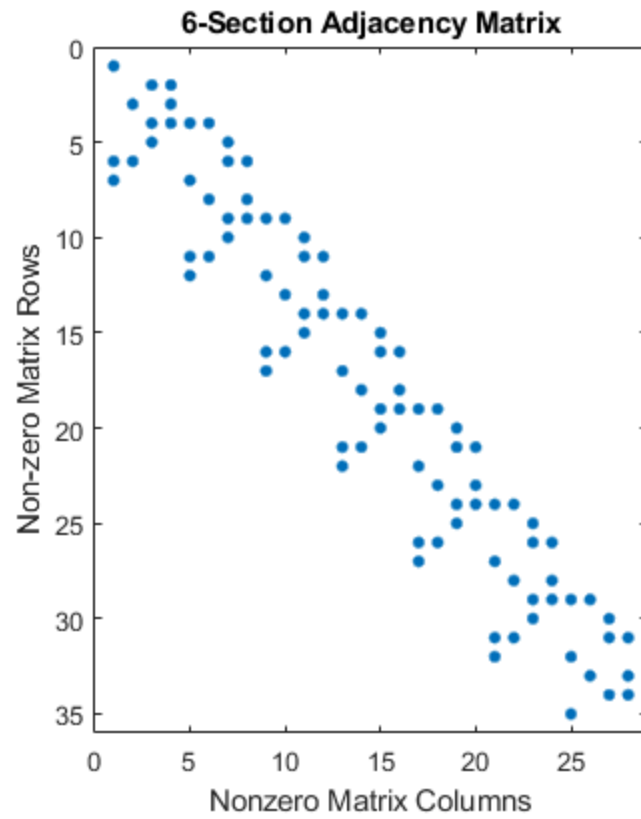


6-Section Bridge with Zero Load

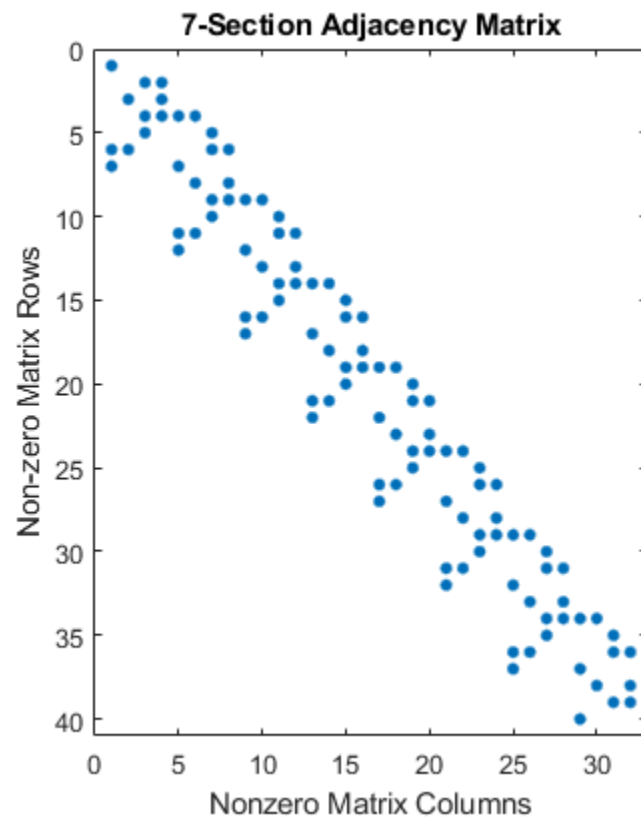
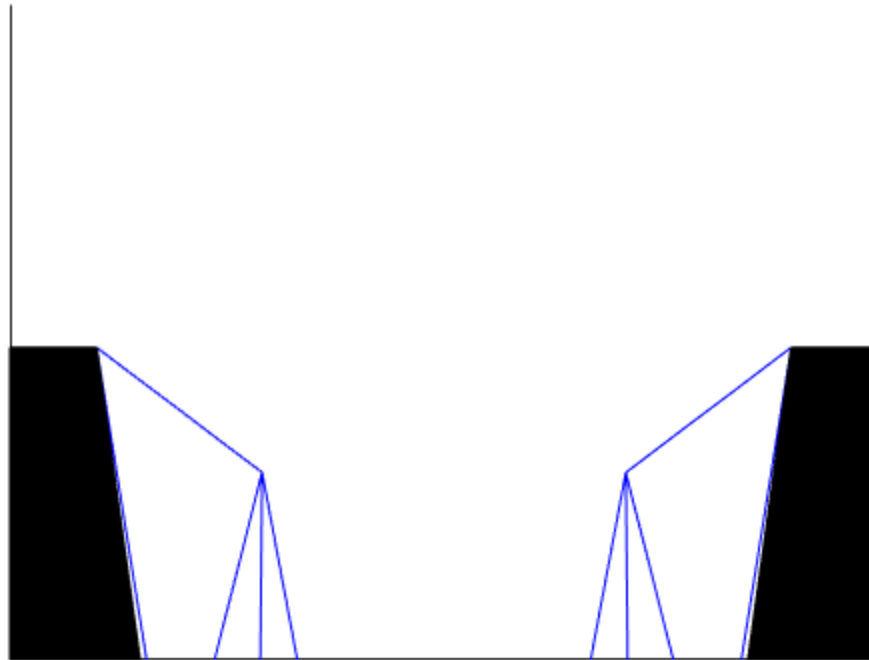


6-Section Deformed Bridge
Car Weight: 0.01, Work: 0.037407

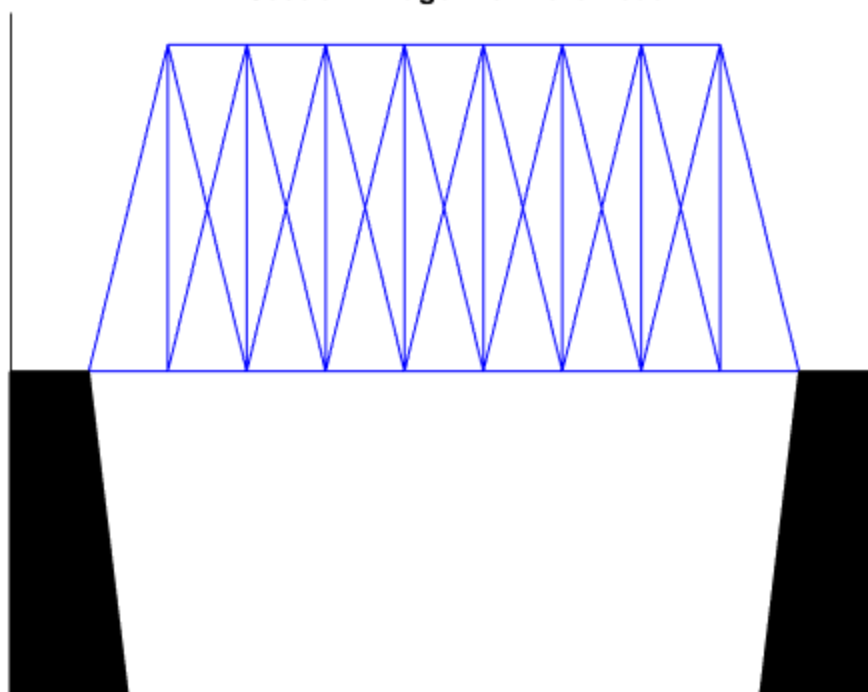




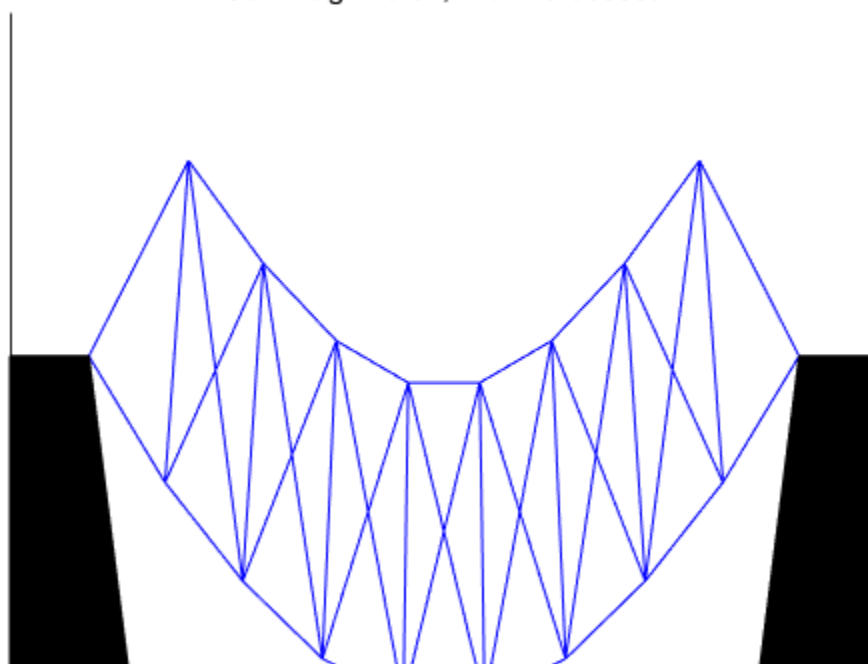
6-Section Deformed Bridge
Car Weight: 0.05, Work: 0.935168

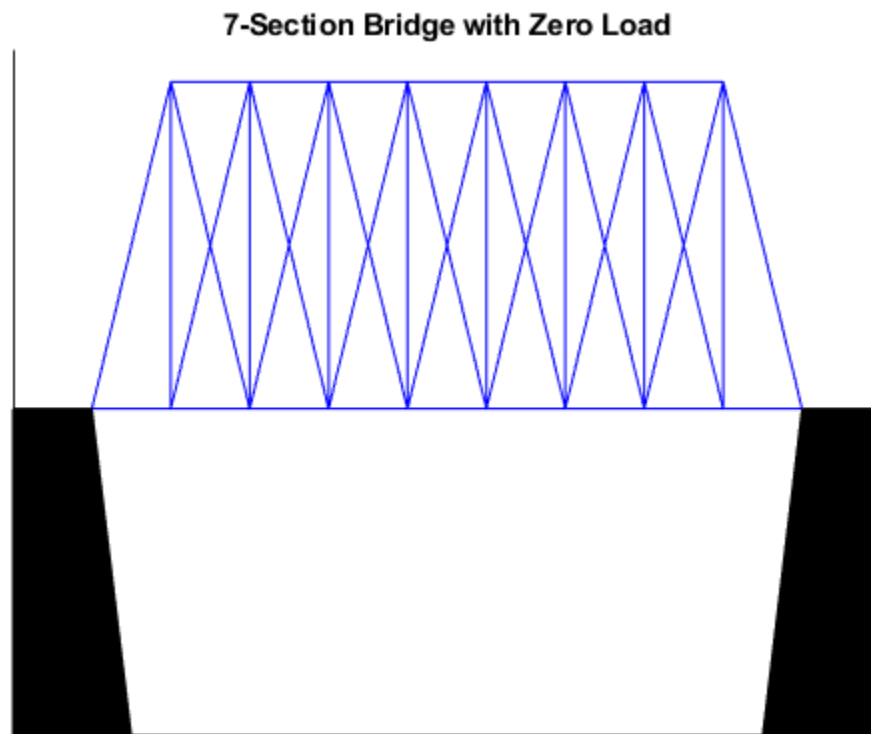
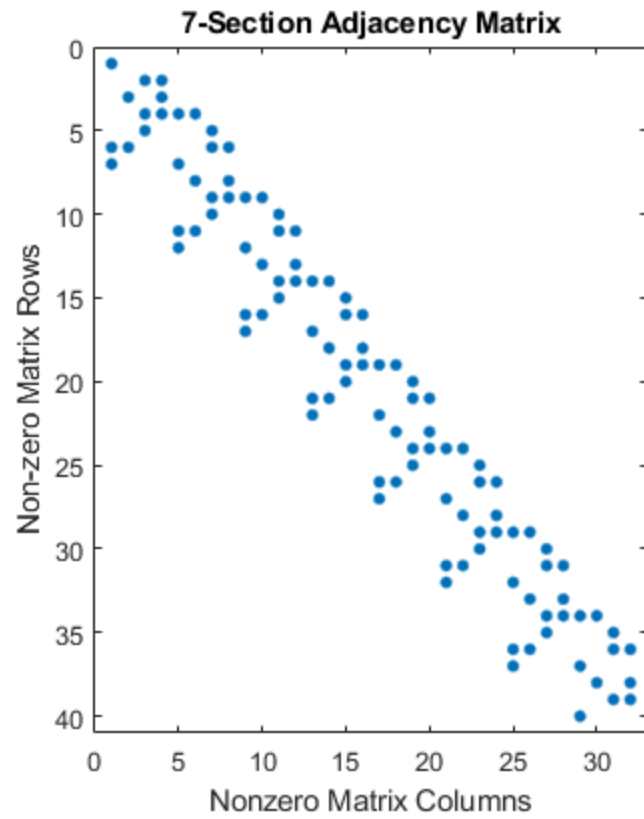


7-Section Bridge with Zero Load

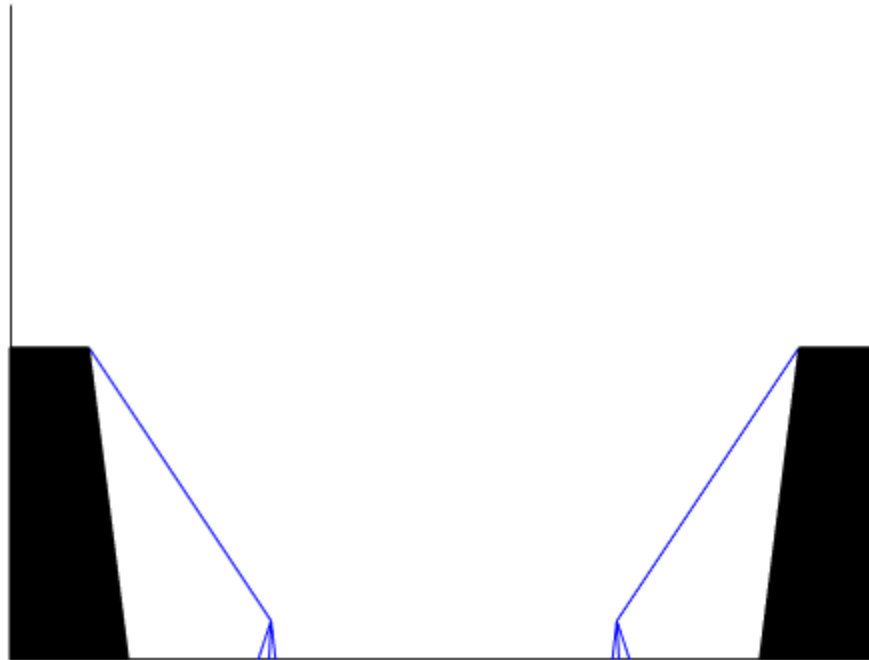


7-Section Deformed Bridge
Car Weight: 0.01, Work: 0.063839





7-Section Deformed Bridge
Car Weight: 0.05, Work: 1.595969



Published with MATLAB® R2021a