
```

%Kaitlyn Kirt, CMOR 220, Spring 2024, Bridges Project
%Project 6.m
%This script is a project on building a bridge
%Last modified: March 18, 2024

%driver
function Project6
    for NoS=1:3 %runs code for one, two, and three sections
        [adjacency,xc,yc,xcd1,ycd1,W1]=BridgeOperator(NoS,-0.01); %defines
function for -0.01 car weight
        [~,~,~,xcd2,ycd2,W2]=BridgeOperator(NoS,-0.05); %defines function for
-0.05 car weight
        figure
        subplot(2,2,1)
        BasicBridgePlotter(xc,yc,NoS); %plots the basic bridge
        subplot(2,2,2)
        DeformedBridge(xcd1,ycd1,NoS,0.01,W1) %plots the deformed bridge with
smaller weight
        subplot(2,2,3)
        DeformedBridge(xcd2,ycd2,NoS,0.05,W2) %plots the deformed bridge with
heavier weight
        subplot(2,2,4)
        SpyTool(NoS,adjacency) %creates the spy visualization for each bridge
    end
end

function [adjacency,xc,yc,xcd,ycd,W]=BridgeOperator(NoS,CWeight)
%inputs: NoS,CWeight
%outputs: adjacency,xc,yc,xcd,ycd,W
%description: this function defines variables to run other functions with
%the same variables
[adjacency,xc,yc,L]=BridgeAXYL(NoS);
[Force]=forcevector(NoS,CWeight);
[xcd,ycd,W]=BridgeDeformer(adjacency,xc,yc,L,Force,NoS);
end

function [adjacency,xc,yc,L]=BridgeAXYL(NoS)
%inputs: NoS
%outputs: adjacency,xc,yc,L
%description: this function finds the coordinates for the basic bridge
nodes=2*NoS+2; %defines nodes for number of sections
beams=5*NoS+5;%defines beams for number of sections
L=ones(beams,1);%creates column vector of lengths for the adjacency function
s=sqrt(2); %defines length of diagonal beams

adjacency=zeros(beams,2*nodes);
adjacency(1,1)=1;
adjacency(2,[3,4])=[1/s,1/s];
L(2)=s; %defines the length of the second beam
xc=zeros(beams,2); %preallocates xc
yc=zeros(beams,2); %preallocates yc

```

```

%creates coordinates for left end x and y coordinates
xc(1,:)= [0,1];
yc(1,:)= [0,0];
xc(2,:)= [0,1];
yc(2,:)= [0,1];

    for i=1:NoS %runs code for "number of section" times
        column=4*i-3; %defines the column used
        row=5*i-2; %defines the row used
        adjacency(row+0,[column+0 column+1 column+2 column+3])=[0 -1 0 1];
        adjacency(row+1,[column+2 column+3 column+4 column+5])=[-1/s 1/s 1/s
-1/s];
        adjacency(row+2,[column+2 column+3 column+6 column+7])=[1 0 -1 0];
        adjacency(row+3,[column+0 column+1 column+6 column+7])=[1/s 1/s -1/s
-1/s];
        adjacency(row+4,[column+0 column+1 column+4 column+5])=[1 0 -1 0];
        %creates coordinates for middle x and y coordinates
        xc(row+0,:)= [i,i];
        yc(row+0,:)= [0,1];
        xc(row+1,:)= [i,i+1];
        yc(row+1,:)= [1,0];
        xc(row+2,:)= [i,i+1];
        yc(row+2,:)= [1,1];
        xc(row+3,:)= [i+1,i];
        yc(row+3,:)= [1,0];
        xc(row+4,:)= [i,i+1];
        yc(row+4,:)= [0,0];
        L(row+1)=s; %modify the length of the beams
        L(row+3)=s; %modify the length of the beams
    end

rows=5*NoS+3;
columns=4*NoS+1;
adjacency(rows+0,[columns+1 columns+3])=[-1 1];
adjacency(rows+1,[columns+2 columns+3])= [-1/s 1/s];
adjacency(rows+2,[columns+0 columns+1])=[-1 0];
L(rows+1)=s; %modify the length of the beams
%creates coordinates for right end x and y coordinates
xc(rows+0,:)= [NoS+1 NoS+1];
yc(rows+0,:)= [0,1];
xc(rows+1,:)= [NoS+1,NoS+2];
yc(rows+1,:)= [1,0];
xc(rows+2,:)= [NoS+1,NoS+2];
yc(rows+2,:)= [0,0];
end

function [Force]=forcevector(NoS,CWeight)
%inputs: NoS,CWeight
%outputs: Force
%description: this function defines the car force applied to certain nodes
nodes=2*NoS+2; %defines the number of nodes for number of sections
Force=zeros(2*nodes,1); %
    for n=2:4:length(Force) %runs code for length(Force) times by increments
of 4

```

```

    Force(n)=CWeight; %gives every 4th node extra weight/mass
end
end

function [xcd,ycd,W]=BridgeDeformer(adjacency,xc,yc,L,Force,NoS)
%inputs: adjacency,xc,yc,L,Force,NoS
%outputs:xcd,ycd,W
%description: this function finds the coordinates of the defomed (force)
matrix
S=adjacency'*diag(1./L)*adjacency;
D=S\Force;
W=D'*Force;
dx=D(1:2:end); %index vector of x displacements
dy=D(2:2:end); %index vector of y displacements

xcd=zeros(size(xc)); %creates deformed x-coordinate matrix
ycd=zeros(size(yc)); %creates deformed y-coordinate matrix
s=0;
%creates the displaced left end x and y coordinates
xcd(1,:)=xc(1,:)+[0 dx(1)];
ycd(1,:)=yc(1,:)+[0 dy(1)];
xcd(2,:)=xc(2,:)+[0 dx(2)];
ycd(2,:)=yc(2,:)+[0 dy(2)];

    for i=1:NoS %runs code for "number of sections" times
        rows=5*i-2;
        %creates the displaced middle x and y coordinates
        xcd(rows+0,:)=xc(rows+0,:)+[dx(1+2*s),dx(2+2*s)];
        ycd(rows+0,:)=yc(rows+0,:)+[dy(1+2*s),dy(2+2*s)];
        xcd(rows+1,:)=xc(rows+1,:)+[dx(2+2*s),dx(3+2*s)];
        ycd(rows+1,:)=yc(rows+1,:)+[dy(2+2*s),dy(3+2*s)];
        xcd(rows+2,:)=xc(rows+2,:)+[dx(2+2*s),dx(4+2*s)];
        ycd(rows+2,:)=yc(rows+2,:)+[dy(2+2*s),dy(4+2*s)];
        xcd(rows+3,:)=xc(rows+3,:)+[dx(4+2*s),dx(1+2*s)];
        ycd(rows+3,:)=yc(rows+3,:)+[dy(4+2*s),dy(1+2*s)];
        xcd(rows+4,:)=xc(rows+4,:)+[dx(1+2*s),dx(3+2*s)];
        ycd(rows+4,:)=yc(rows+4,:)+[dy(1+2*s),dy(3+2*s)];
        s=s+1; %redefines s after each run
    end
    rows=5*NoS+3;
    columns=4*NoS+1;
    %creates the displaced right end x and y coordinates
    xcd(rows+0,:)=xc(rows+0,:)+[dx(end-1),dx(end)];
    ycd(rows+0,:)=yc(rows+0,:)+[dy(end-1),dy(end)];
    xcd(rows+1,:)=xc(rows+1,:)+[dx(end) 0];
    ycd(rows+1,:)=yc(rows+1,:)+[dy(end) 0];
    xcd(rows+2,:)=xc(rows+2,:)+[dx(end-1),0];
    ycd(rows+2,:)=yc(rows+2,:)+[dy(end-1),0];
end

function BasicBridgePlotter(xc,yc,NoS)
%inputs:xc,yc,NoS
%outputs: none
%description: this function plots the original bridges

```

```

hold on;
plot(xc',yc','Color','b')
title(num2str(NoS),"Section Basic Bridge")
fill([-1 -1 0 0.5],[-1 0 0 -1],"black") %creates the bridges' anchor
fill([NoS+1.5 NoS+2 NoS+3 NoS+3],[-1 0 0 -1],"black") %creates the bridges'
anchor
xlim([-1 NoS+3]) %sets the plot bounds
ylim([-1 2]) %sets the plot bounds
end

```

```

function DeformedBridge(xcd,ycd,NoS,CWeight,W)
%inputs: xcd,ycd,NoS,CWeight,W
%outputs: none
%description: this function plots the deformed bridges

```

```

hold on;
plot(xcd',ycd','Color','r')
title(num2str(NoS),"Section Deformed Bridge")
fill([-1 -1 0 0.5],[-1 0 0 -1],"black") %creates the bridges' anchor
fill([NoS+1.5 NoS+2 NoS+3 NoS+3],[-1 0 0 -1],"black") %creates the bridges'
anchor
xlim([-1 NoS+3]) %sets the plot bounds
ylim([-1 2]) %sets the plot bounds
end

```

```

function SpyTool(NoS,adjacency)
%inputs: NoS,adjacency
%outputs: none
%description: this function creates a spy visualization of each bridge

```

```

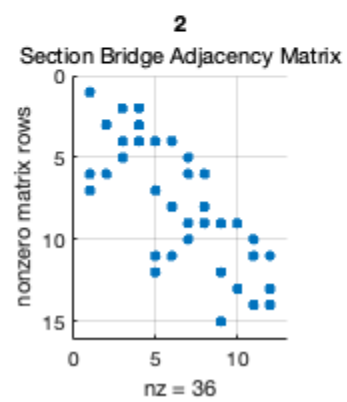
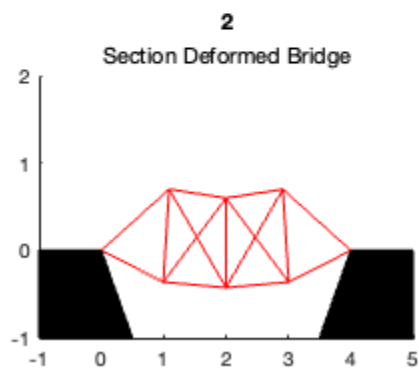
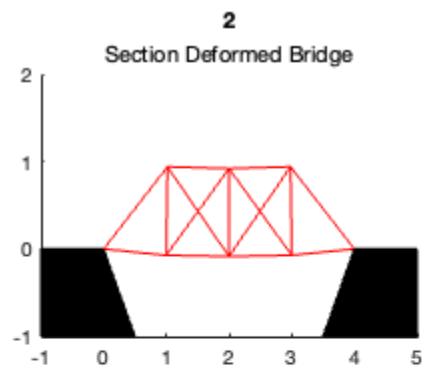
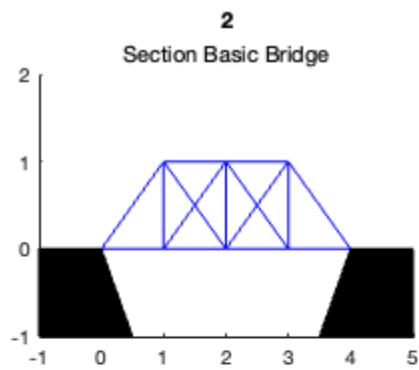
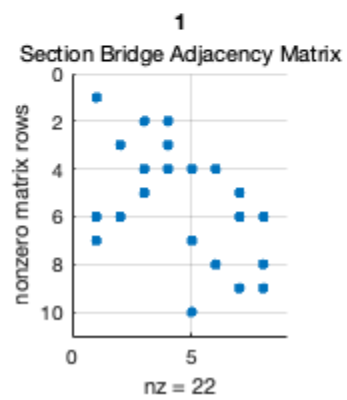
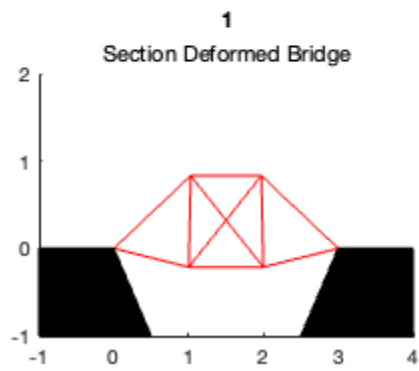
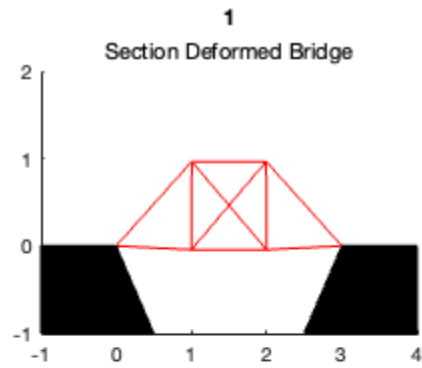
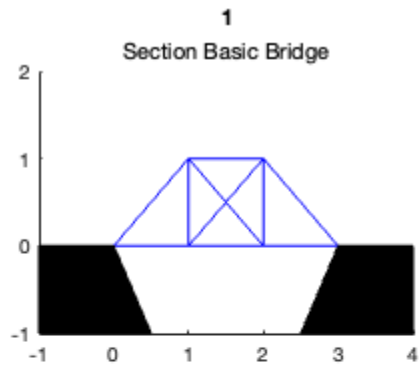
hold on; grid on;
title(num2str(NoS),"Section Bridge Adjacency Matrix")
xlabel("nonzero matrix columns")
ylabel("nonzero matrix rows")
spy(adjacency)
end

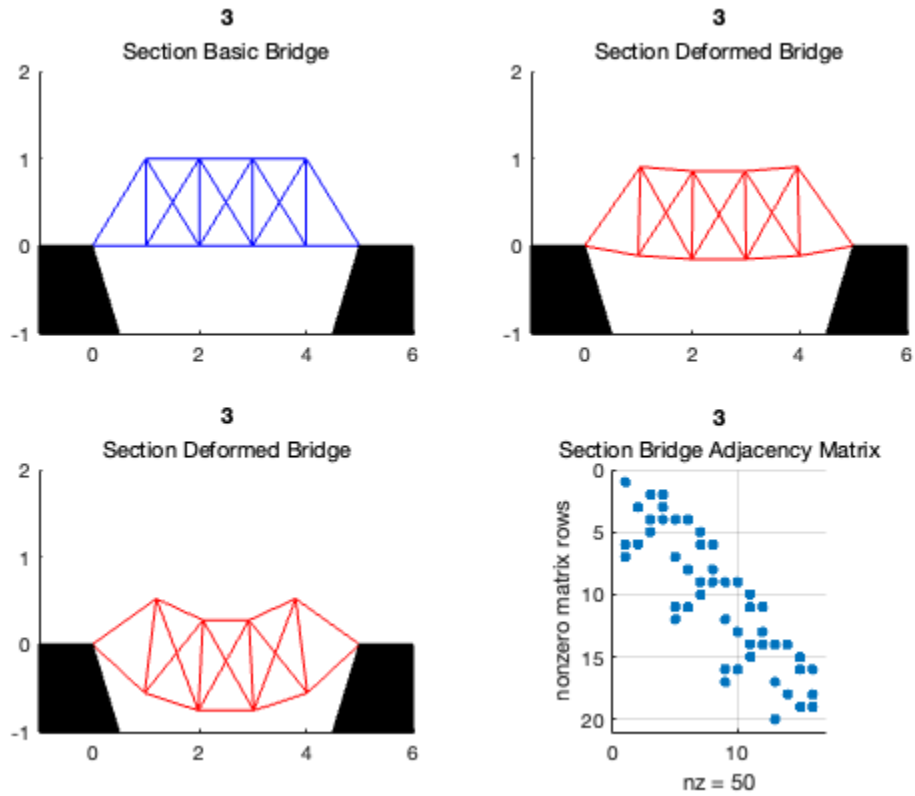
```

```

%1. While driving a light-weight car, the maximum length the bridge would
%be considered safe is 5. My criterion for safe is whether the the
%lowest point of the bridge does not droop lower than -0.5.
%2. While driving an 18-wheeler truck, the maximum length the bridge would
%be considered safe is 2.
%3. The spy for a particular bridge relates to how it deforms because the
%number and location of the stright diagonal lines is where the bridges
%desides to deform and drop.
%4. As nos increases, the length of the bridge increases the bridge is more
%succeptible to collapsing, number of nonzero matrix rows and columns
%increases, and he plots on the spy get more concentrated.

```





Published with MATLAB® R2023b