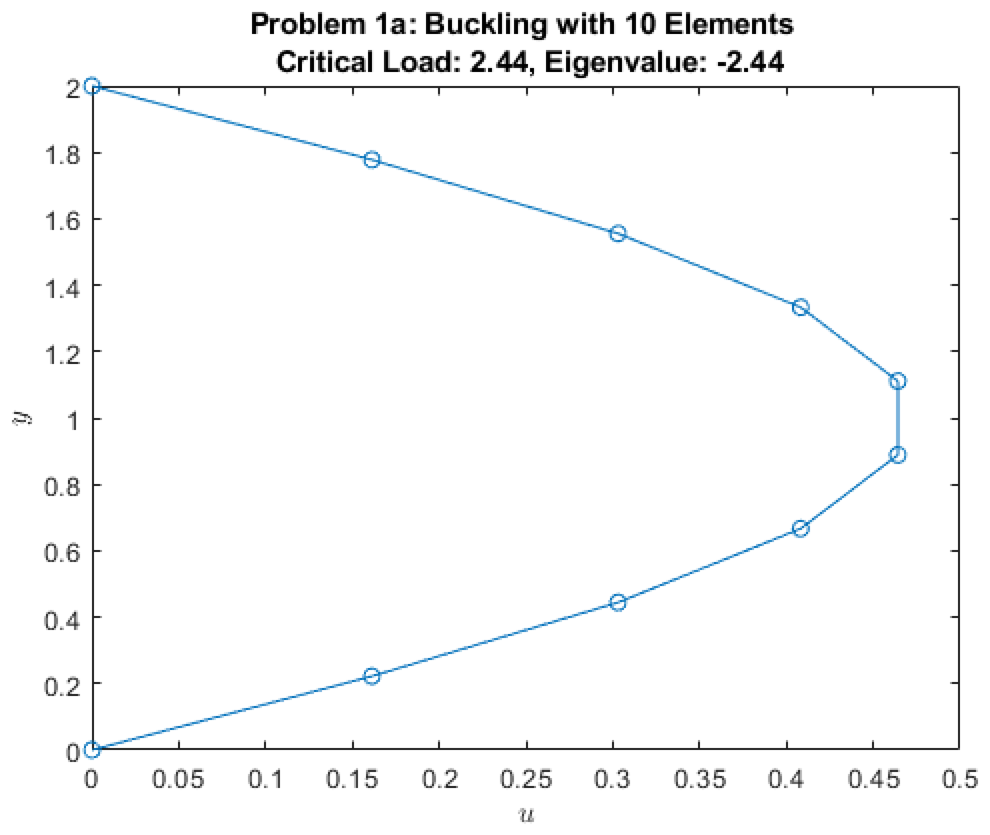
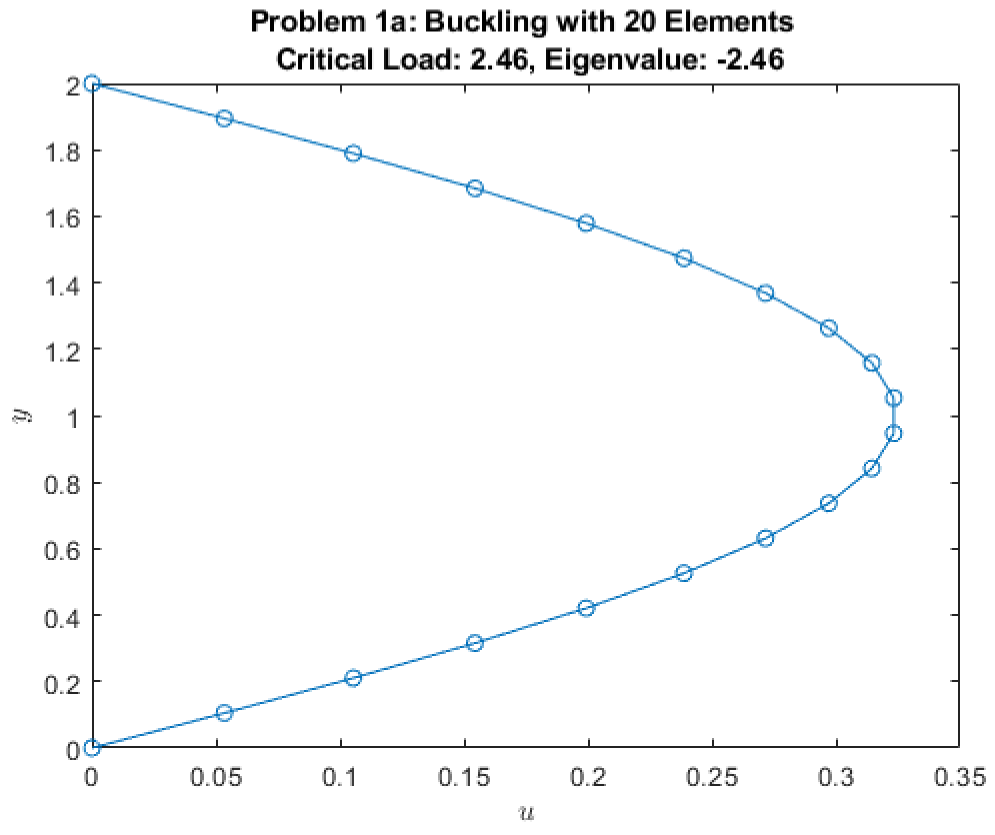
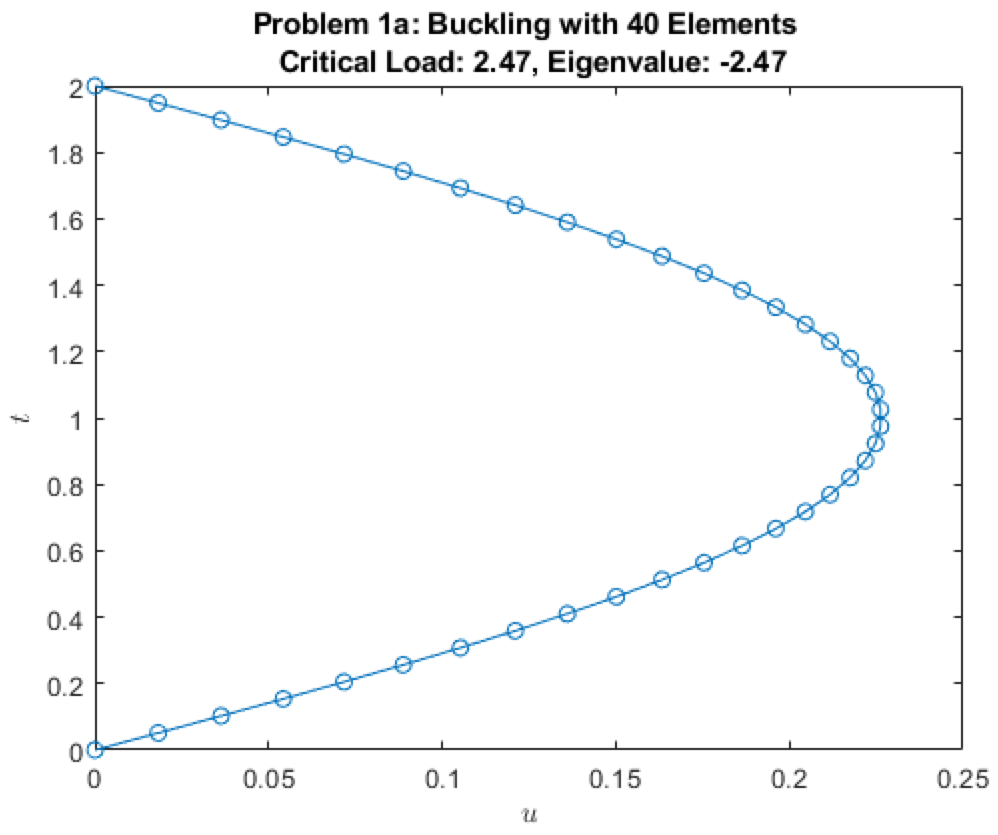
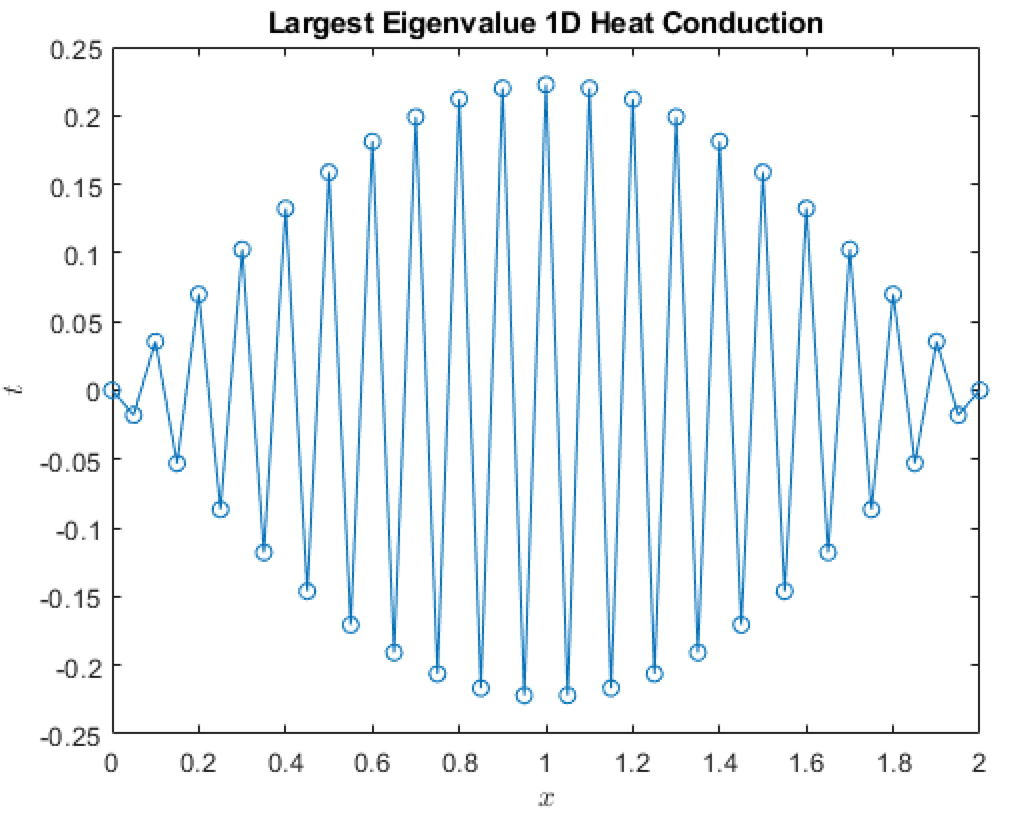
```
close all;
clc, clear;


[lambda10,u10,P10,y10] = strutBuckling(10);
[lambda20,u20,P20,y20] = strutBuckling(20);
[lambda40,u40,P40,y40] = strutBuckling(40);

[lambdaLarge,uLarge,xLarge] = heatConduction1D(41,'largest');
[lambdaSmall,uSmall,xSmall] = heatConduction1D(41,'smallest');

[lambdaVibration, uVibration, frequency, mode, xVibration] =
 stringVibration(40);

figure('Name','Strut Buckling')
plot(u10,y10,'Marker','o')
title(sprintf('Problem 1a: Buckling with 10 Elements \n Critical Load: %.2f,
 Eigenvalue: %.2f', P10, lambda10))
xlabel('$u$',Interpreter='latex')
ylabel('$y$',Interpreter='latex')

figure('Name','Strut Buckling')
plot(u20,y20,'Marker','o')
title(sprintf('Problem 1a: Buckling with 20 Elements \n Critical Load: %.2f,
 Eigenvalue: %.2f', P20, lambda20))
xlabel('$u$',Interpreter='latex')
ylabel('$y$',Interpreter='latex')

figure('Name','Strut Buckling')
plot(u40,y40,'Marker','o')
title(sprintf('Problem 1a: Buckling with 40 Elements \n Critical Load: %.2f,
 Eigenvalue: %.2f', P40, lambda40))
xlabel('$u$',Interpreter='latex')
ylabel('$t$',Interpreter='latex')

figure('Name','Heat Conduction')
plot(xLarge,uLarge,'Marker','o')
title('Largest Eigenvalue 1D Heat Conduction')
xlabel('$x$','Interpreter','latex')
ylabel('$t$','Interpreter','latex')

figure('Name','Heat Conduction')
plot(xSmall,uSmall,'Marker','o')
title('Problem 1B: Smallest Eigenvalue 1D Heat Conduction')
xlabel('$x$','Interpreter','latex')
ylabel('$t$','Interpreter','latex')

figure('Name','String Vibration')
plot(xVibration,uVibration,'Marker','o')
title(sprintf('Problem 1C: String Vibration \n Lowest Frequency: %.2f || Mode:
 %.1f',frequency,round(mode)))
xlabel('$x$','Interpreter','latex')
```

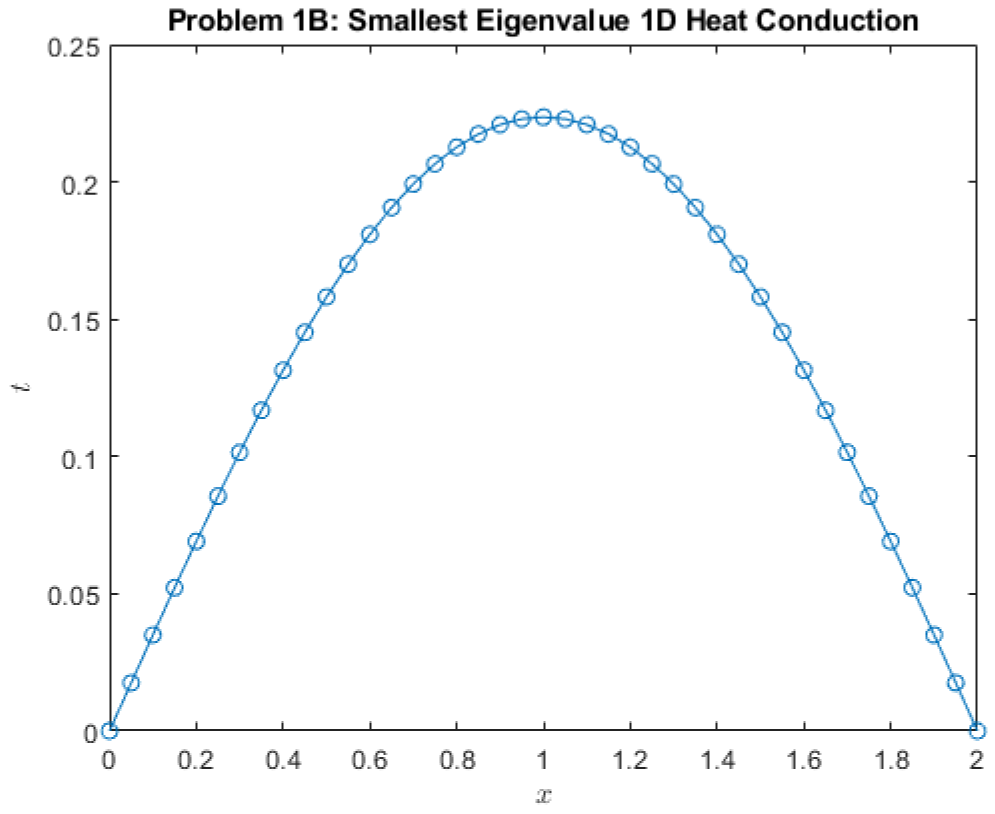
```
ylabel('$y$','Interpreter','latex')

function [eigenvalue, eigenvector_u, critical_load,y] = strutBuckling(n)
    y = linspace(0,2,n);
    delta_y = y(2);
    guessU = ones(n,1); %f(0)
    guessU(1) = 0; guessU(n)=0;

    E = 1;
    I = 1;

    % set up tridiagonal matrix A, which is based on central difference scheme
    alpha = 1/delta_y^2;
    beta = -2/delta_y^2;
    a = alpha.*ones(n,1); a(n) = 0;
    b = beta.*ones(n,1); b(1) = 1; b(n) = 1;
    c = alpha.*ones(n,1); c(1) = 0;

    residuals = 1;
    tolerance = 1e-5;
    newU = guessU;
    lambdaOld = 0;
    while residuals > tolerance
       oldU = newU;
       newU = THOMAS3(a,b,c,oldU,n)';
       lambda = (oldU'*newU)/(oldU'*oldU);
       newU = newU./sqrt(newU'*newU);
       residuals = max(abs(lambda-lambdaOld));
       lambdaOld = lambda;
    end
    eigenvalue = 1/lambda;
    eigenvector_u = newU;
    critical_load = -eigenvalue*E*I;
end

function [eigenvalue, eigenvector_u, x] =
 heatConduction1D(n,powerMethodOption)
    x = linspace(0,2,n);
    delta_x = x(2);
    guessU = ones(n,1); %f(0)
    guessU(1) = 0; guessU(n)=0;

    % set up tridiagonal matrix A, which is based on central difference scheme
    alpha = 1/delta_x^2;
    beta = -2/delta_x^2;
    a = alpha.*ones(n,1); a(n) = 0;
    b = beta.*ones(n,1); b(1) = 1; b(n) = 1;
    c = alpha.*ones(n,1); c(1) = 0;
    A = diag(a(2:n),-1)+diag(b)+diag(c(1:n-1),1);

    residuals = 1;
    tolerance = 1e-5;
    newU = guessU;
    lambdaOld = 0;
    switch powerMethodOption
        case 'largest'
             while residuals > tolerance
               oldU = newU;
               newU = A*oldU;
               lambda = (oldU'*newU)/(oldU'*oldU);
               newU = newU./sqrt(newU'*newU);
               residuals = max(abs(lambda-lambdaOld));
               lambdaOld = lambda;
             end
             eigenvalue = lambda;
        case 'smallest'
            while residuals > tolerance
               oldU = newU;
               newU = THOMAS3(a,b,c,oldU,n)';
               lambda = (oldU'*newU)/(oldU'*oldU);
               newU = newU./sqrt(newU'*newU);
               residuals = max(abs(lambda-lambdaOld));
               lambdaOld = lambda;
            end
            eigenvalue = 1/lambda;
    end
    eigenvector_u = newU;
end

function [eigenvalue, eigenvector_u, frequency, mode,x] = stringVibration(n)
    x = linspace(0,2,n);
    delta_x = x(2);
    guessU = ones(n,1); %f(0)
    guessU(1) = 0; guessU(n)=0;

    c_const = 1.5;
    l = 2;

    % set up tridiagonal matrix A, which is based on central difference scheme
    alpha = 1/delta_x^2;
    beta = -2/delta_x^2;
    a = alpha.*ones(n,1); a(n) = 0;
    b = beta.*ones(n,1); b(1) = 1; b(n) = 1;
    c = alpha.*ones(n,1); c(1) = 0;

    residuals = 1;
    tolerance = 1e-15;
    newU = guessU;
    lambdaOld = 0;
    while residuals > tolerance
       oldU = newU;
       newU = THOMAS3(a,b,c,oldU,n)';
       lambda = (oldU'*newU)/(oldU'*oldU);
       newU = newU./sqrt(newU'*newU);
       residuals = max(abs(lambda-lambdaOld));
       lambdaOld = lambda;
    end
    eigenvector_u = newU;
    eigenvalue = 1/lambda;
    frequency = abs(eigenvalue*c_const);
    mode = abs(2*l/(frequency/c_const));
end

function x = THOMAS3(a,b,c,d,n)

    %initial condition
    bbar(1) = b(1);
    cbar(1) = c(1);
    dbar(1) = d(1);

    %making upper triangle
    for i = 2:n
        multiplier = a(i)./bbar(i-1);
        abar(i) = a(i) - bbar(i-1).*multiplier;
        bbar(i) = b(i) - cbar(i-1).*multiplier;
        cbar(i) = c(i);
        dbar(i) = d(i) - dbar(i-1).*multiplier;
    end

    %initialize x of size n
    x = ones(1,n);

    %initialize end condition
    x(n) = dbar(n)/bbar(n);

    % Upward substitution AKA zip it up
    for i = n-1:-1:1
        x(i) = (dbar(i)-(cbar(i)*x(i+1)))/bbar(i);
    end
end
```

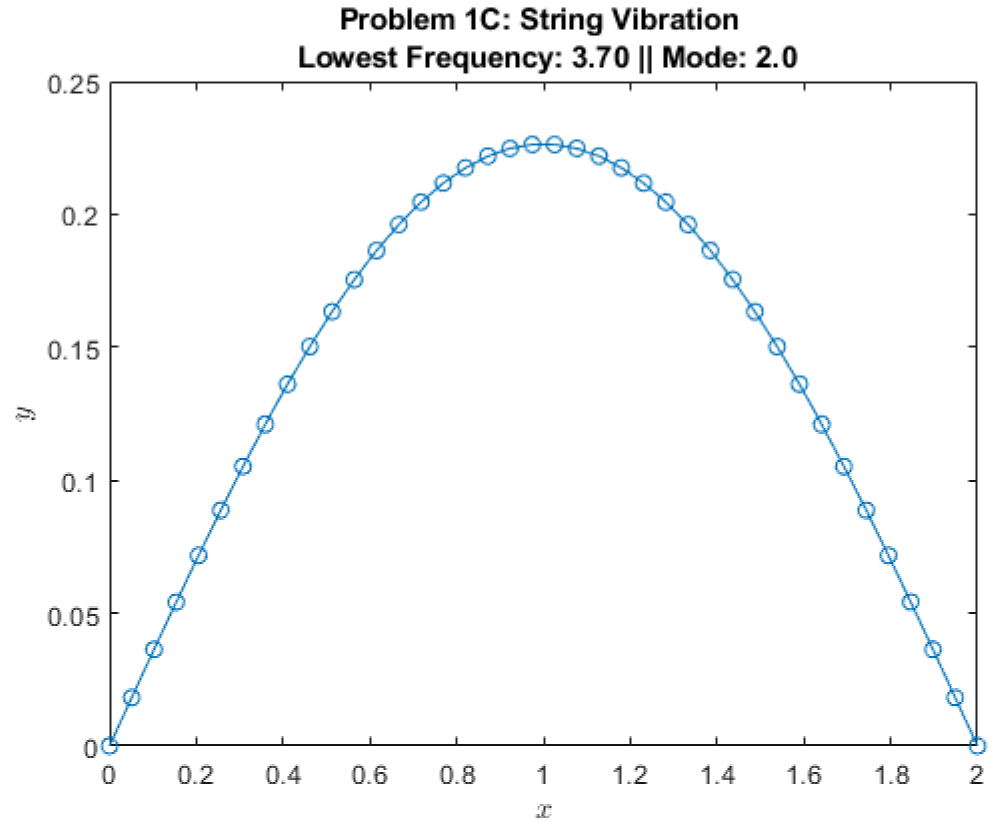

Problem 1a: Buckling with 10 Elements
Critical Load: 2.44, Eigenvalue: -2.44
y
u


Problem 1a: Buckling with 20 Elements
Critical Load: 2.46, Eigenvalue: -2.46
y
u


Problem 1a: Buckling with 40 Elements
Critical Load: 2.47, Eigenvalue: -2.47
t
u
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0
0.05
0.1
0.15
0.2
0.25


Largest Eigenvalue 1D Heat Conduction
t
x
-0.25
-0.2
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
0.2
0.25
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2


Problem 1B: Smallest Eigenvalue 1D Heat Conduction
0.25
0.2
0.15
0.1
0.05
0
t
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
x


Problem 1C: String Vibration
Lowest Frequency: 3.70 || Mode: 2.0
0.25
0.2
0.15
0.1
0.05
0
y
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
x