Assignment 2

Morgan Yost

January 22, 2016

1 Exercise 2.2

1.1 Problem Description

a. Determine the Green's function for the two point BVP with $u'' = f(x)$ for $0 < x < 1$ with Neumann boundary condition at $x = 0$ and Dirichlet at $x = 1$.

$$u'' = \delta(x - \bar{x}), u'(0) = 0, u(1) = 0$$

and the functions $G_0(x; \bar{x})$ solving

$$u'' = \delta(x - \bar{x}), u'(0) = 1, u(1) = 0$$

and $G_1(x; \bar{x})$ solving

$$u'' = 0, u'(0) = 0, u(1) = 1$$

b. Write out A and A-1 for h = .25.

1.2 Problem Solution

1.2.1 Part A

$$G(x; \bar{x}) = \int\int G''(x; \bar(x)) = \int\int 0$$

results in

$G(x; \bar{x}) =$

$$\begin{cases} Ax + B & x < \bar{x} \\ Cx + D & x > \bar{x} \end{cases}$$

To solve for A, B C and D, assert the boundary conditions as well at continuity (i.e. $A\bar{x} + B = C\bar{x} + D$) and the definition of the delta function which gives us $C - A = 1$.
From the boundary conditions:

$$u'(0) = 0 \therefore A = 0$$

$$u(1) = 0 \therefore C = -D$$

$$C - A = 1 \therefore C = 1 and D = -1$$

$$A\bar{x} + B = C\bar{x} + D \therefore B = \bar{x} - 1$$

so that
$G(x; \bar{x}) =$

$$\begin{cases} \bar{x} - 1 & x < \bar{x} \\ \text{x-1} & x > \bar{x} \end{cases}$$

Now to find G_0

$$G_0(x; \bar{x}) = Ax + B$$

Asserting boundary conditions we have

$$G_0'(0; \bar{x}) = A = \therefore A = 1$$

$$G_0(1; \bar{x}) = Ax + B = x + B = 0 = \therefore B = -1$$

so that

$$G_0(x; \bar{x}) = x - 1$$

repeating the same process for G_1 we find that

$$G_1(x; \bar{x}) = 1$$

1.2.2 Part B

$$A = \begin{matrix} -4 & 4 & 0 & 0 & 0 \\ 16 & -32 & 16 & 0 & 0 \\ 0 & 16 & -32 & 16 & 0 \\ 0 & 0 & 16 & -32 & 16 \\ 0 & 0 & 0 & 0 & 1 \end{matrix}$$

The $A^{-}1$ matrix was found by implementing part a into a MATLAB function with conditional statements. Please refer to the function getG2_2 in appendix A for the implementation of this function the resulting matrix is:

$$A = \begin{matrix} -1.0000 & -0.2500 & -0.1875 & -0.1250 & 1.0000 \\ -0.7500 & -0.1875 & -0.1875 & -0.1250 & 1.0000 \\ -0.5000 & -0.1250 & -0.1250 & -0.1250 & 1.0000 \\ -0.2500 & -0.0625 & -0.0625 & -0.0625 & 1.0000 \\ 0 & 0 & 0 & 0 & 1.0000 \end{matrix}$$

2 Exercise 2.3

2.1 Problem Description

Determine the null space of A^T and verify equation 2.62.

2.2 Problem Solution

The null space of the transpose of the matrix was found using the MATLAB transpose and null functions. Please refer to section 2.3 of the MATLAB code in the appendix for actual implementation.
The null space of A^T is

$$\begin{matrix} 0.6761 \\ 0.1690 \\ 0.1690 \\ 0.1690 \\ 0.6761 \end{matrix}$$

Equation 2.62 was implemented in MATLAB and found to result in a nonzero value of 3.5496, thus validating the criteria for uniqueness of the solution.

3 Exercise 2.4

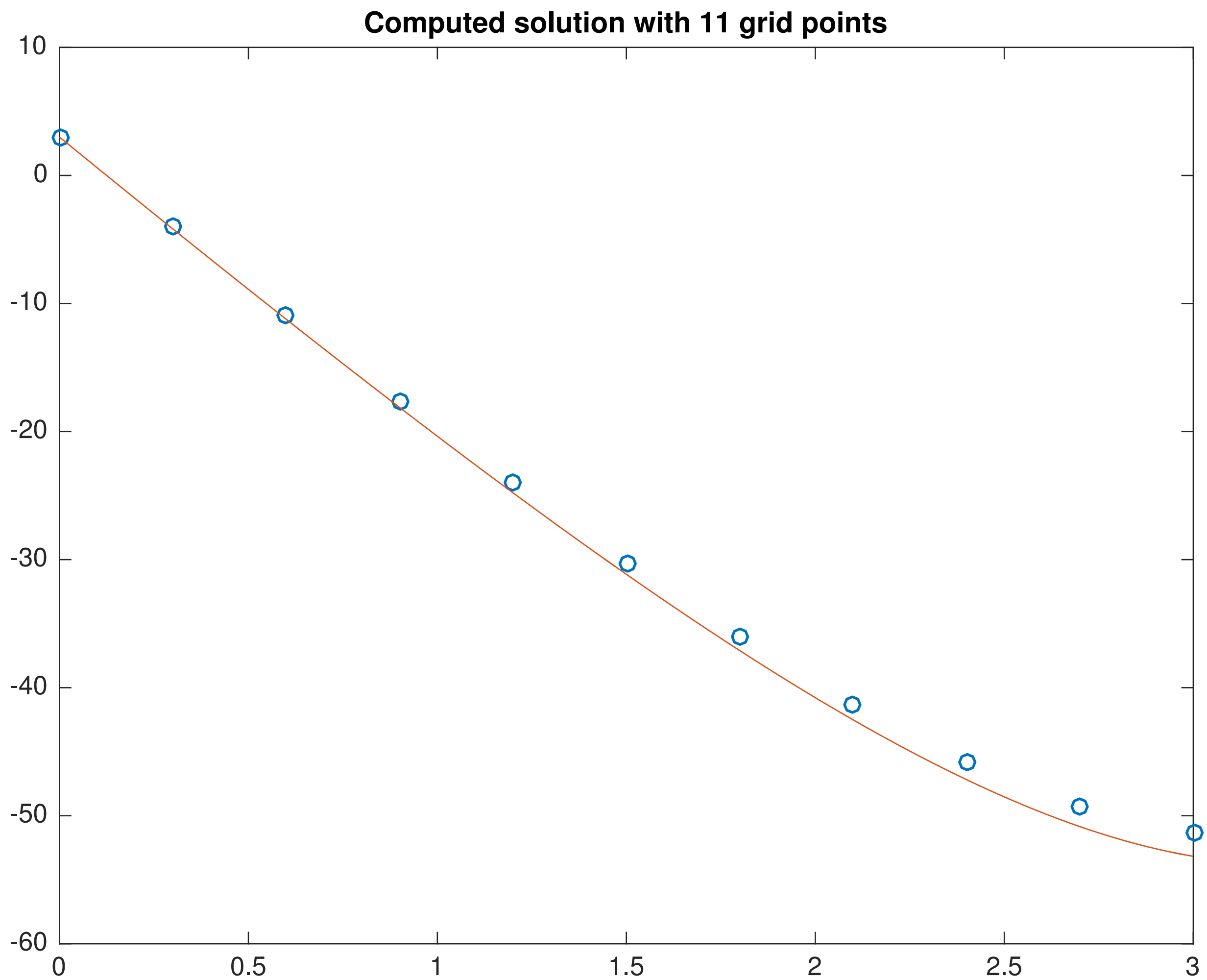
3.1 Problem Description

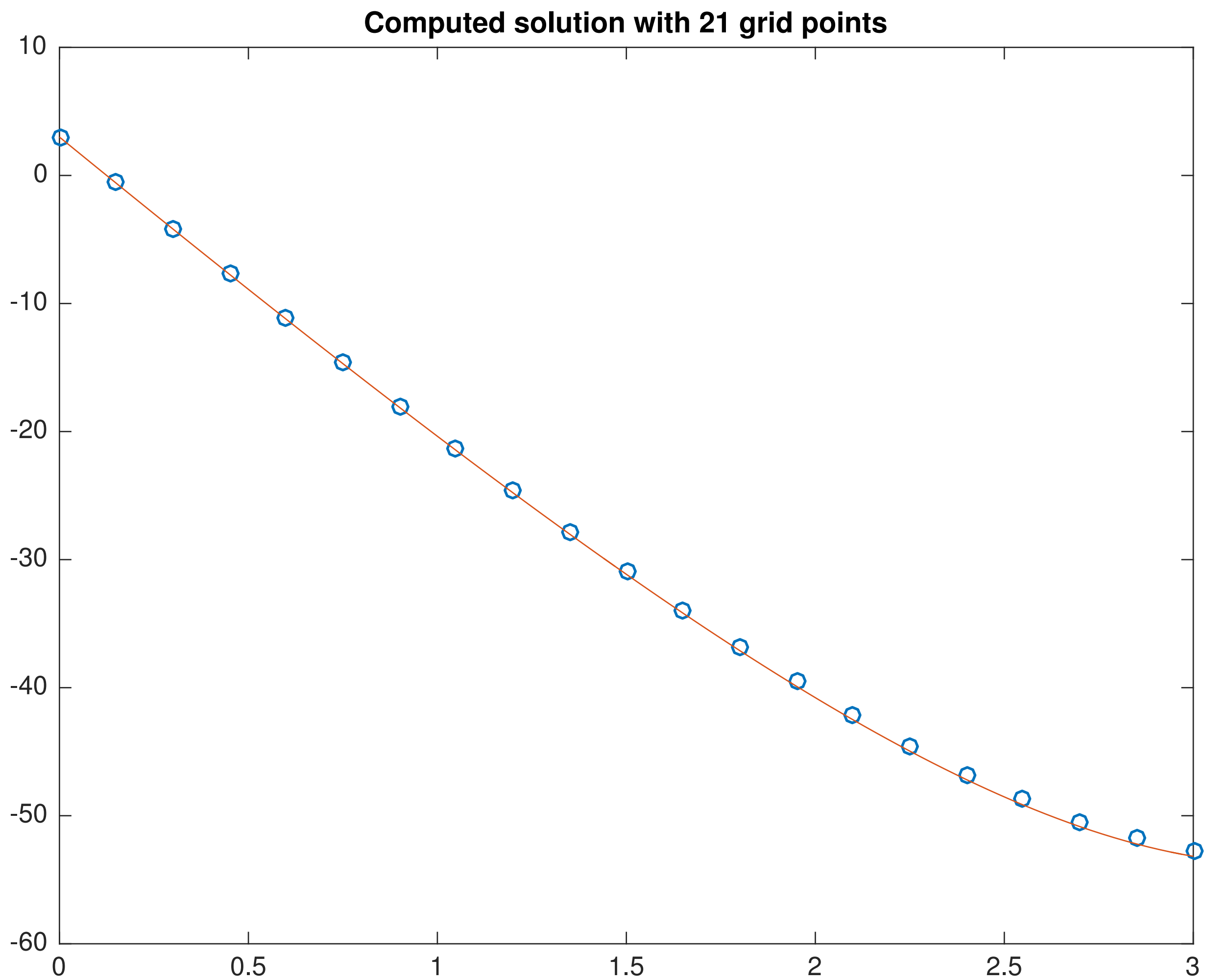
a. Modify bvp2.m for Dirichelet boundary condition at $x = a$ and Neumann at x = b. a. Modify bvp4.m for Dirichelet boundary condition at $x = a$ and Neumann at x = b.

3.2 Problem Solution

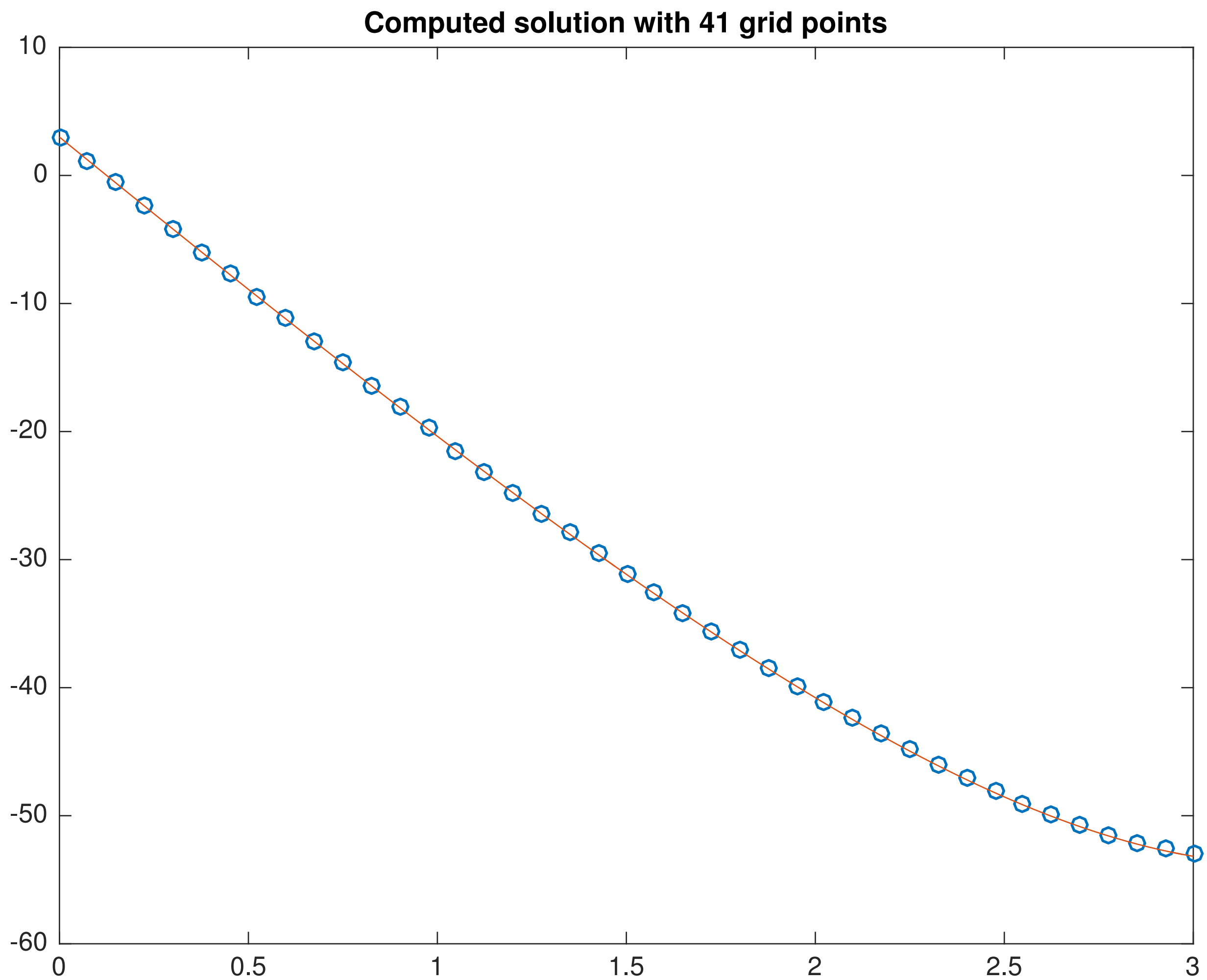
3.2.1 Part A

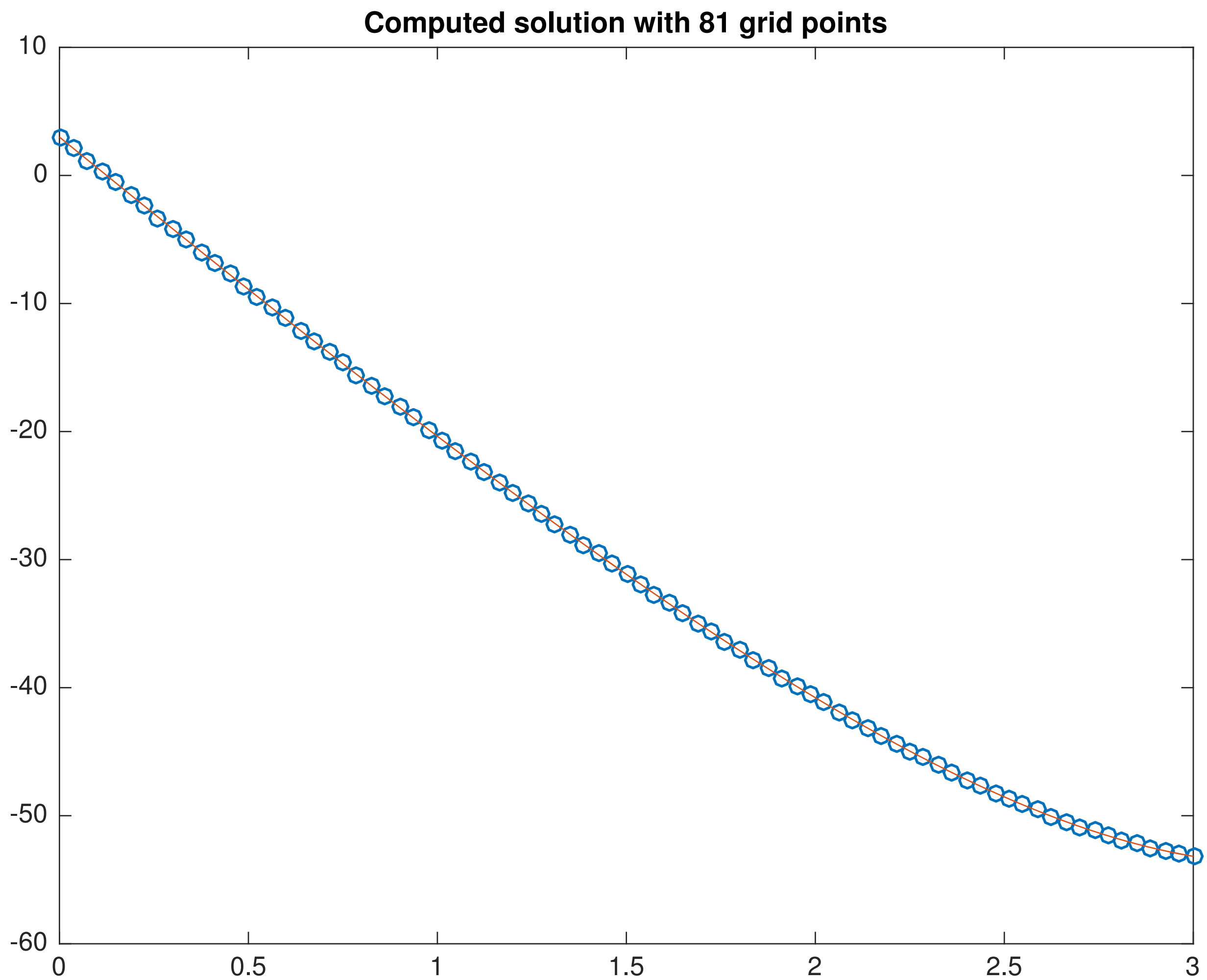
The modified bvp code can be found in bvp_2_2_4 in the MATLAB code in the appendix. The resulting graphs are displayed below.

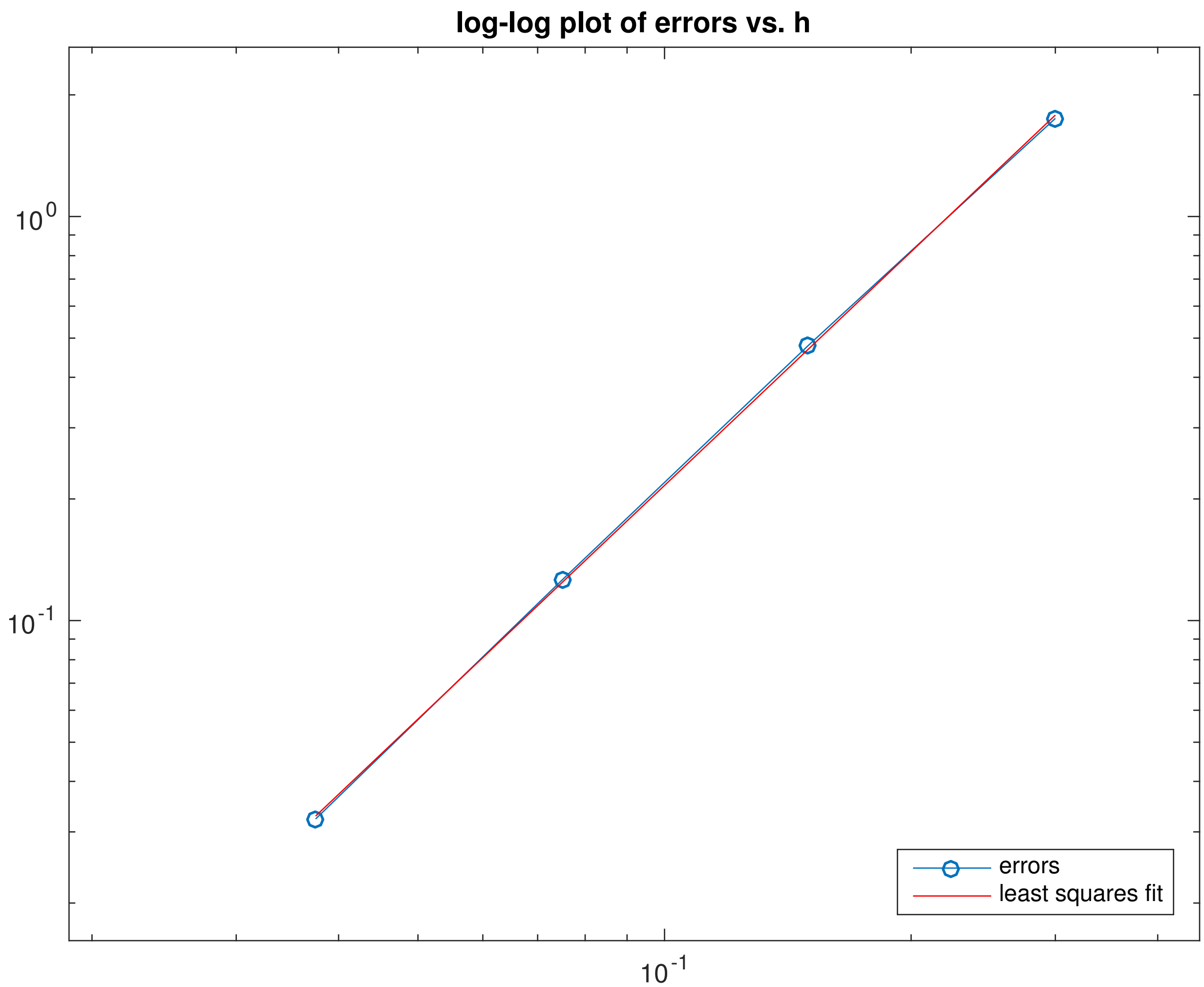

Computed solution with 11 grid points
10
0
-10
-20
-30
-40
-50
-60
0
0.5
1
1.5
2
2.5
3



Computed solution with 21 grid points
10
0
-10
-20
-30
-40
-50
-60
0
0.5
1
1.5
2
2.5
3

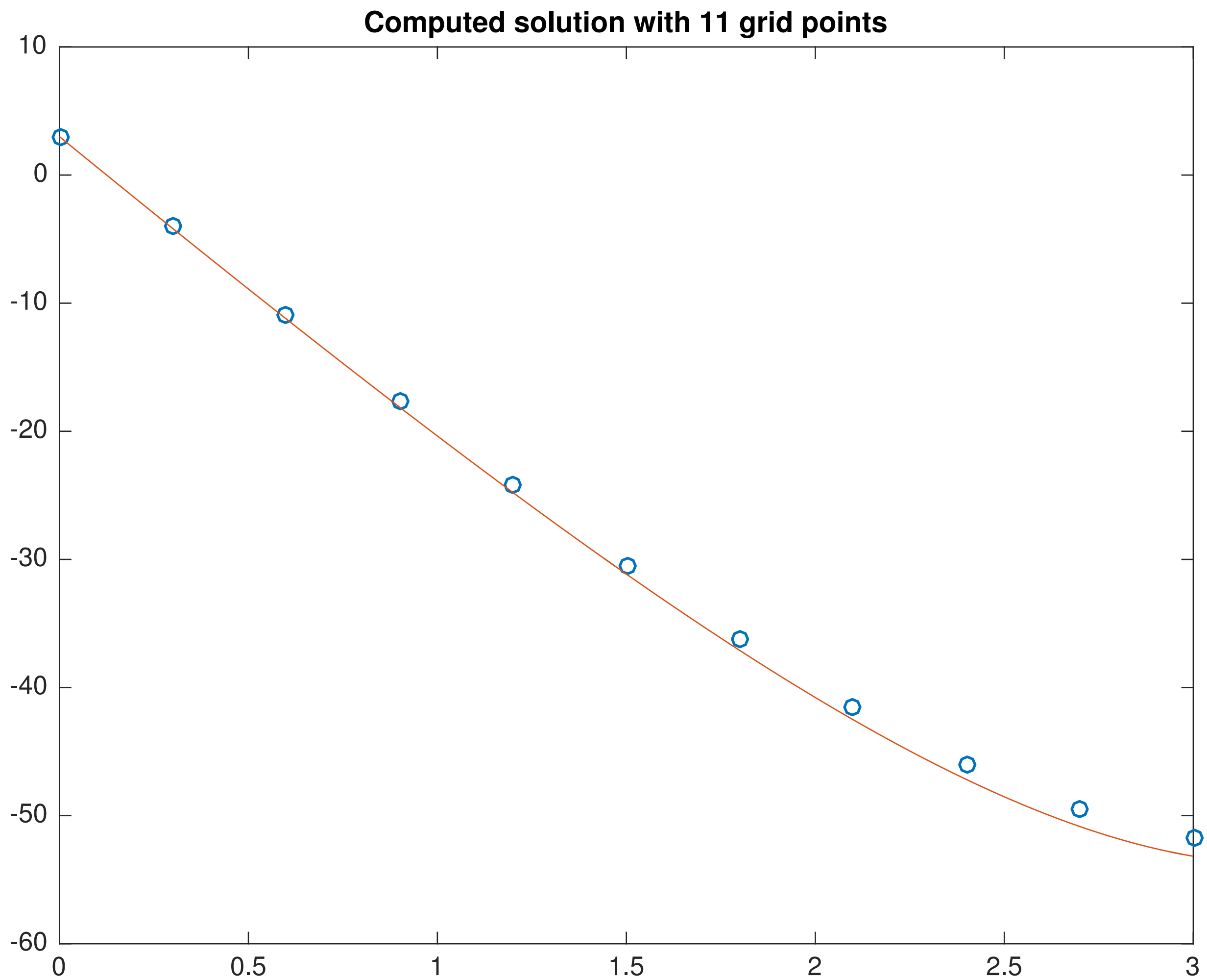

Computed solution with 41 grid points
10
0
-10
-20
-30
-40
-50
-60
0
0.5
1
1.5
2
2.5
3

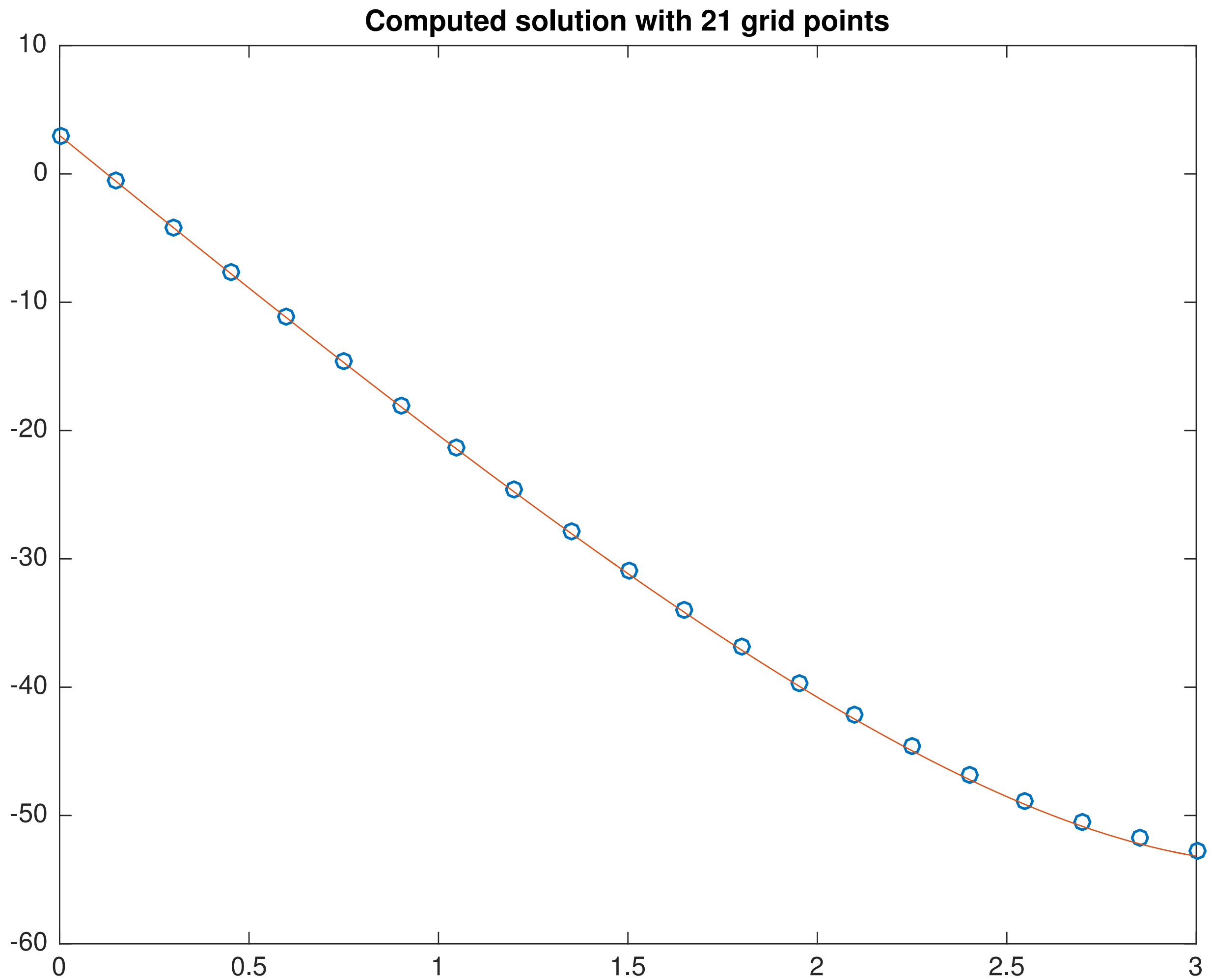

Computed solution with 81 grid points
10
0
-10
-20
-30
-40
-50
-60
0
0.5
1
1.5
2
2.5
3

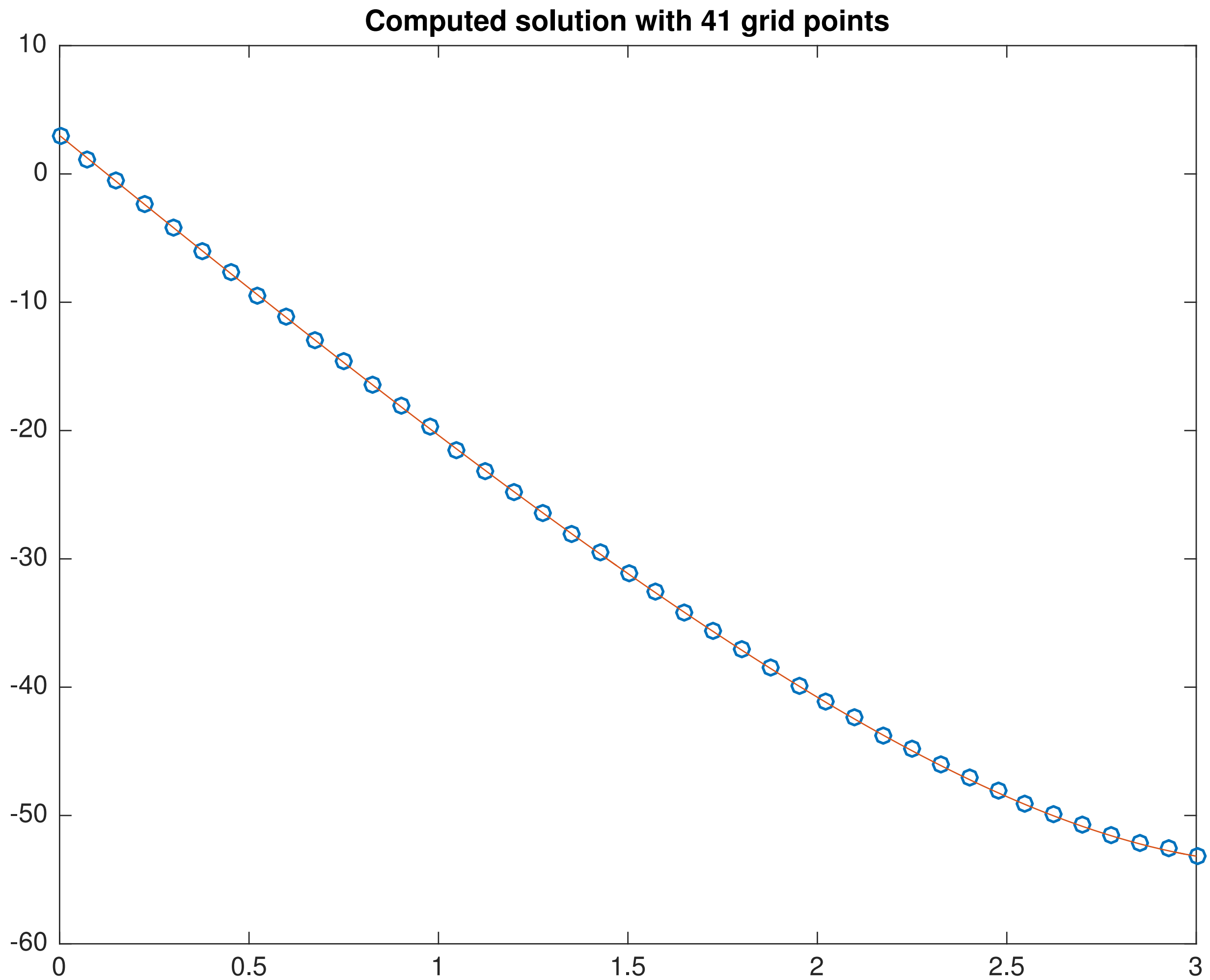

log-log plot of errors vs. h
10^0
10^-1
10^-1
errors
least squares fit

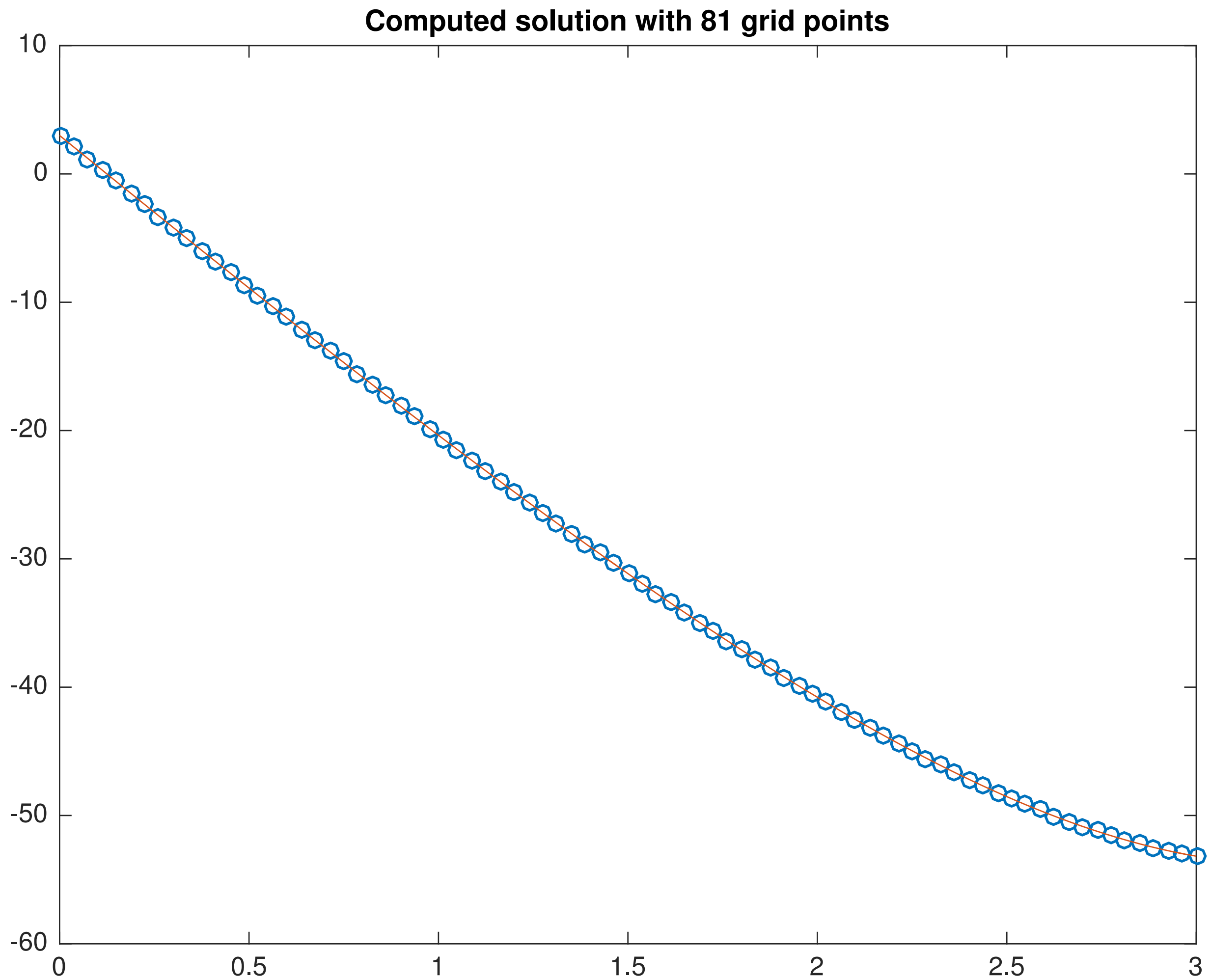
3.2.2 Part B

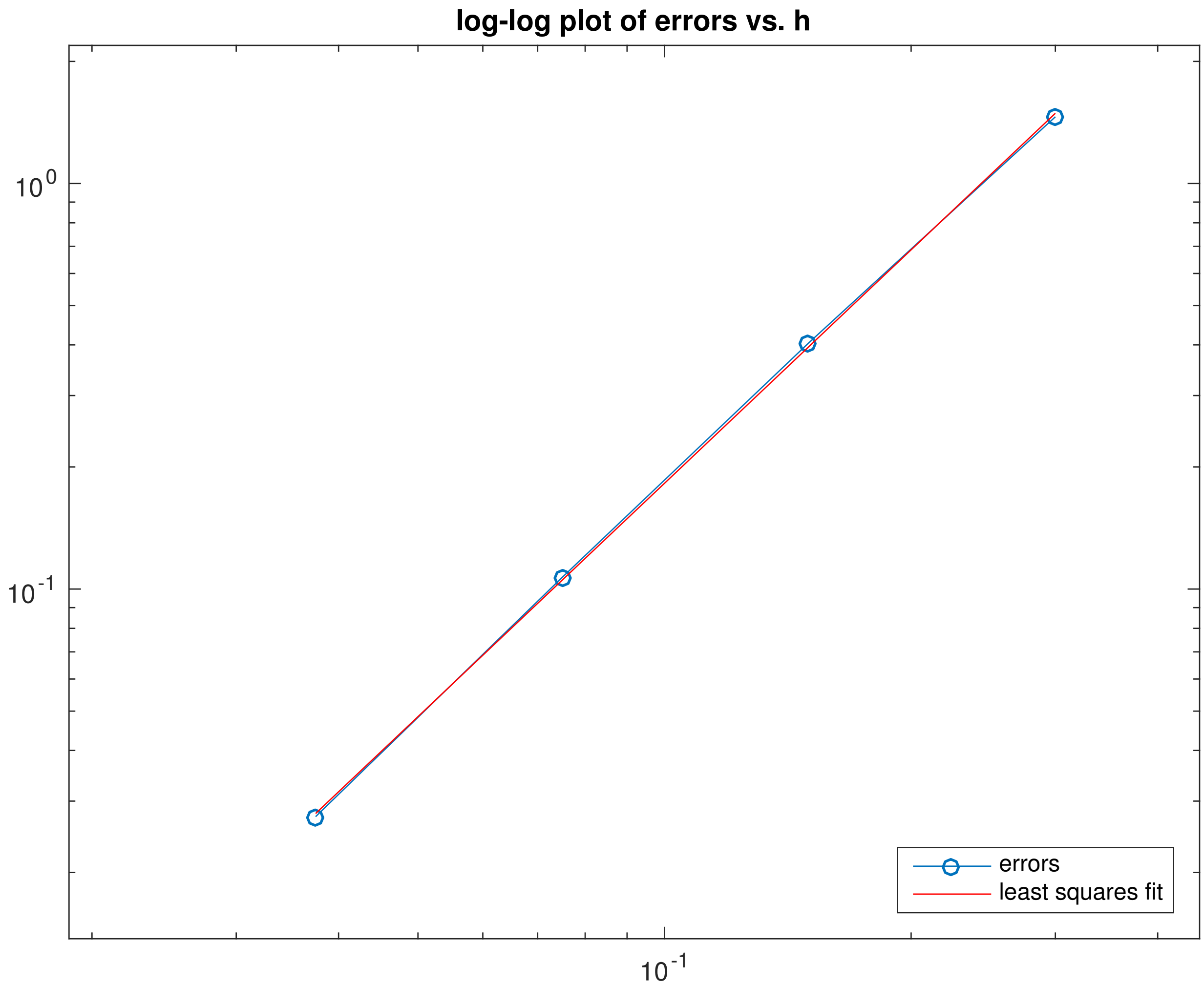
The modified bvp code can be found in bvp_4_2_4 in the MATLAB code in the appendix. The resulting graphs are displayed below.




Computed solution with 21 grid points
10
0
-10
-20
-30
-40
-50
-60
0
0.5
1
1.5
2
2.5
3


Computed solution with 41 grid points
10
0
-10
-20
-30
-40
-50
-60
0
0.5
1
1.5
2
2.5
3


Computed solution with 81 grid points
10
0
-10
-20
-30
-40
-50
-60
0
0.5
1
1.5
2
2.5
3



4 Exercise 2.6

4.1 Problem Description

Consider the following BVP:

$$u''(x) + u(x) = 0 for a < x < b u(a) = \alpha, b(b) = \beta.$$

a. Test the modified bvp with $a = 0, b = 1, \alpha = 2, \beta = 3$.
b. Let $a = 0 and b = \pi$. Sketch the family of solutions for the BVP.
c. Solve the problem with $a = 0, b = \pi, \alpha = 1, \beta = -1$. Also change the boundary condition at π to $\beta = 1$.
d. Compute the eigenvalues of A. Look at the eigenvalues of A as h approaches 0 and the norm of the inverse of A blows up as h approaches zero.

4.2 Problem Solution

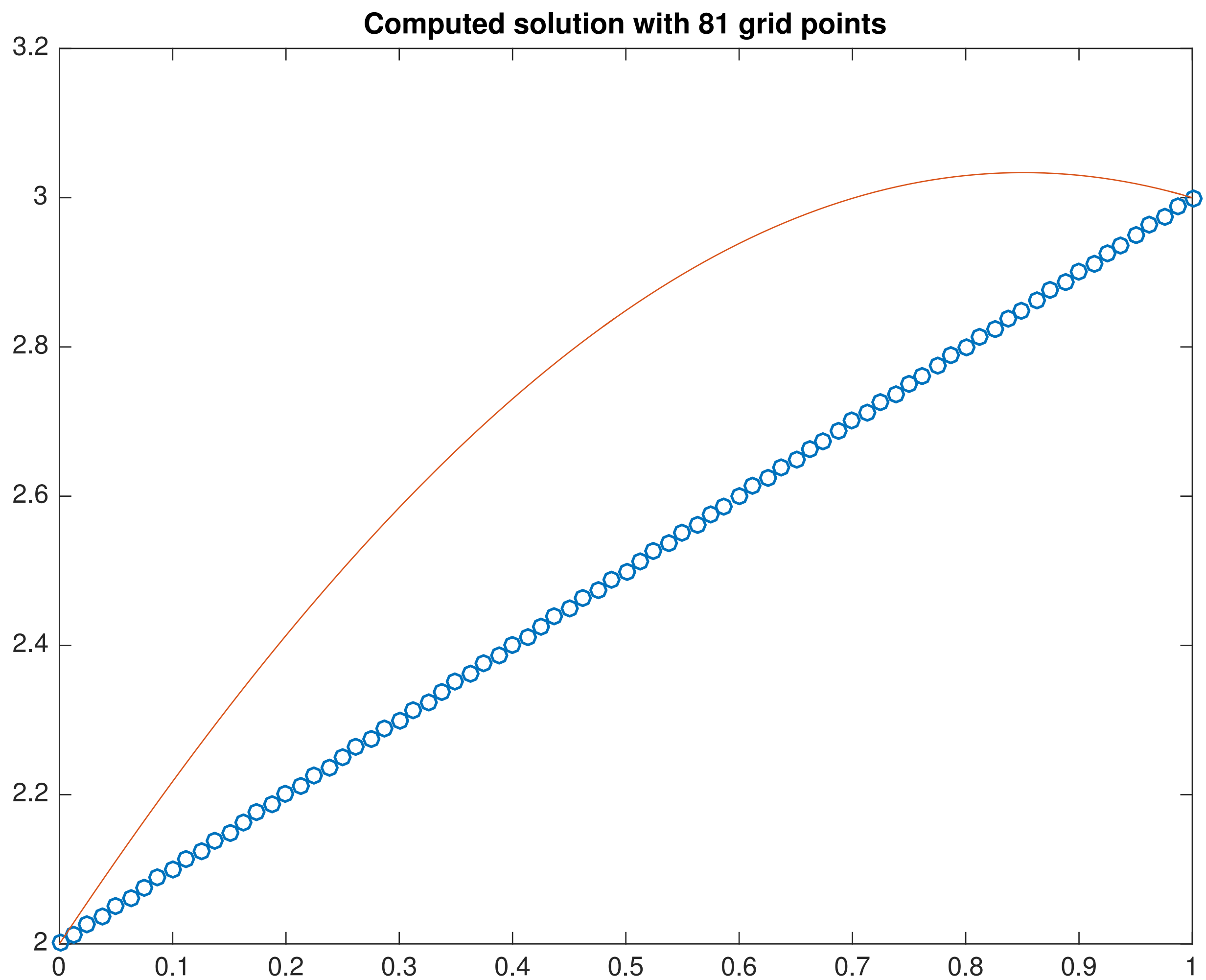
The implementation for this problem can be found in the appendix in the MATLAB code section 2.6.

4.2.1 Part A

The exact solution is of the form

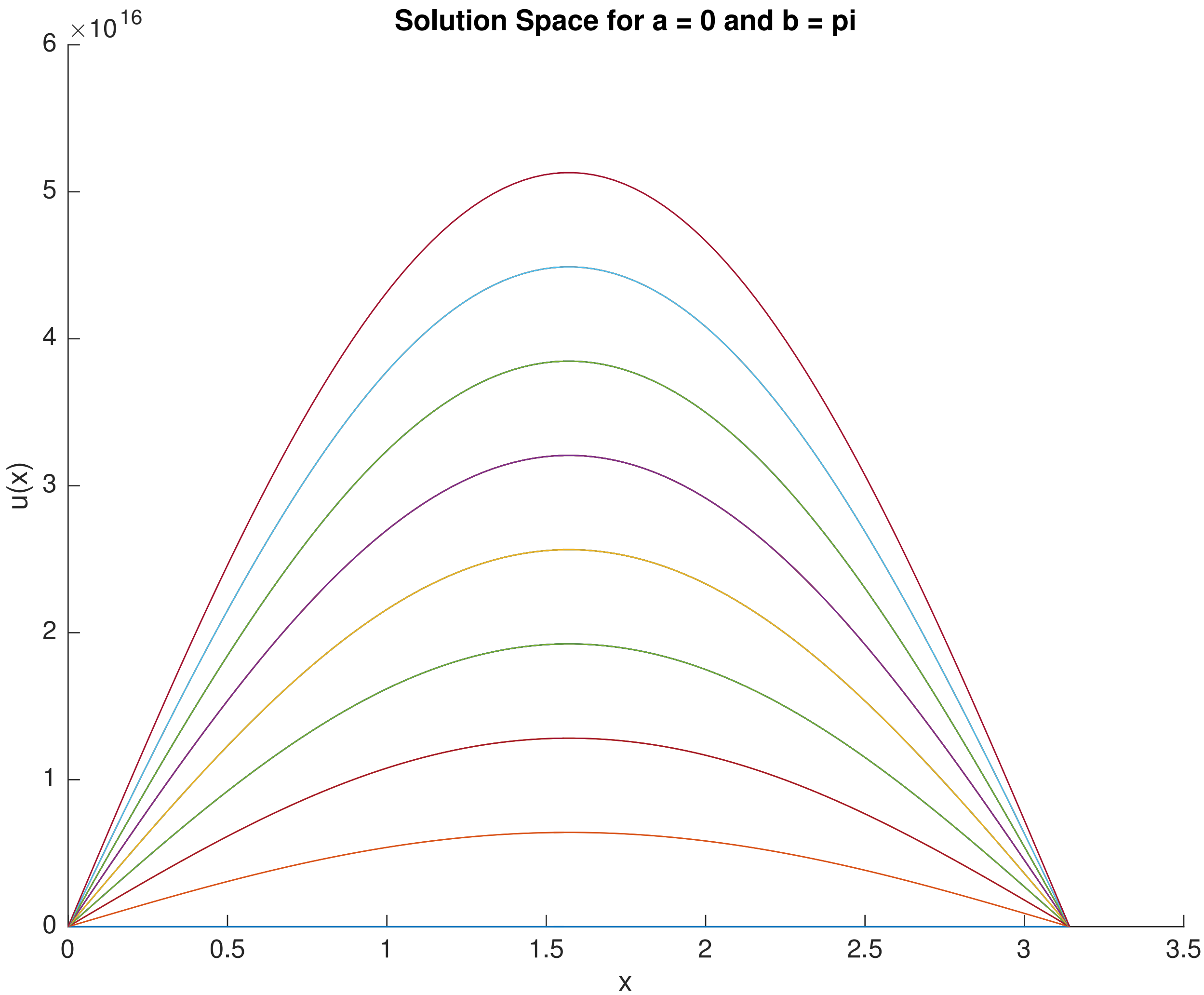
$$u(x) = A\cos x + B\sin x$$

The following plot compares the exact solution to the discretized solution:



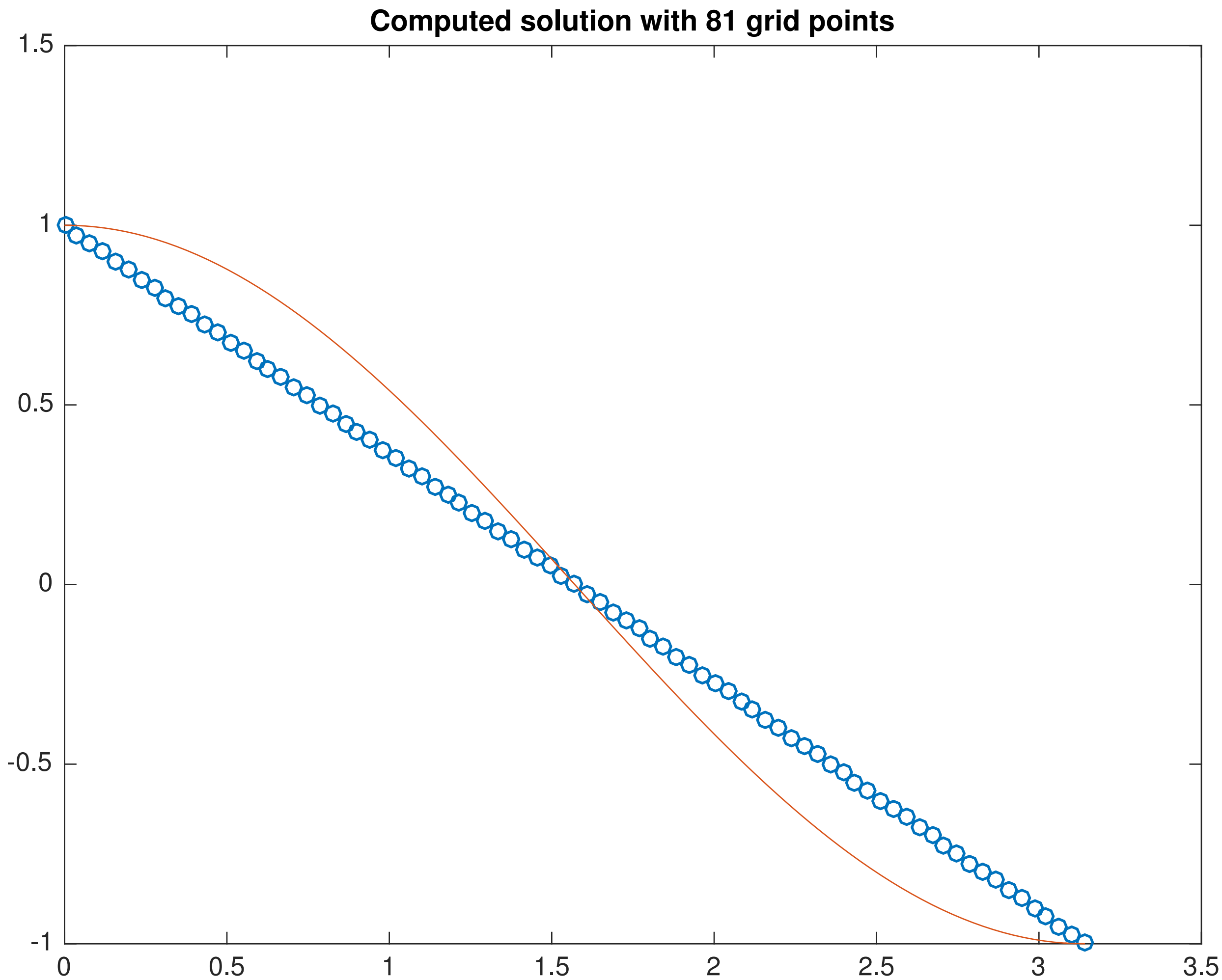
4.2.2 Part B

The solution space is represented by the graph below:

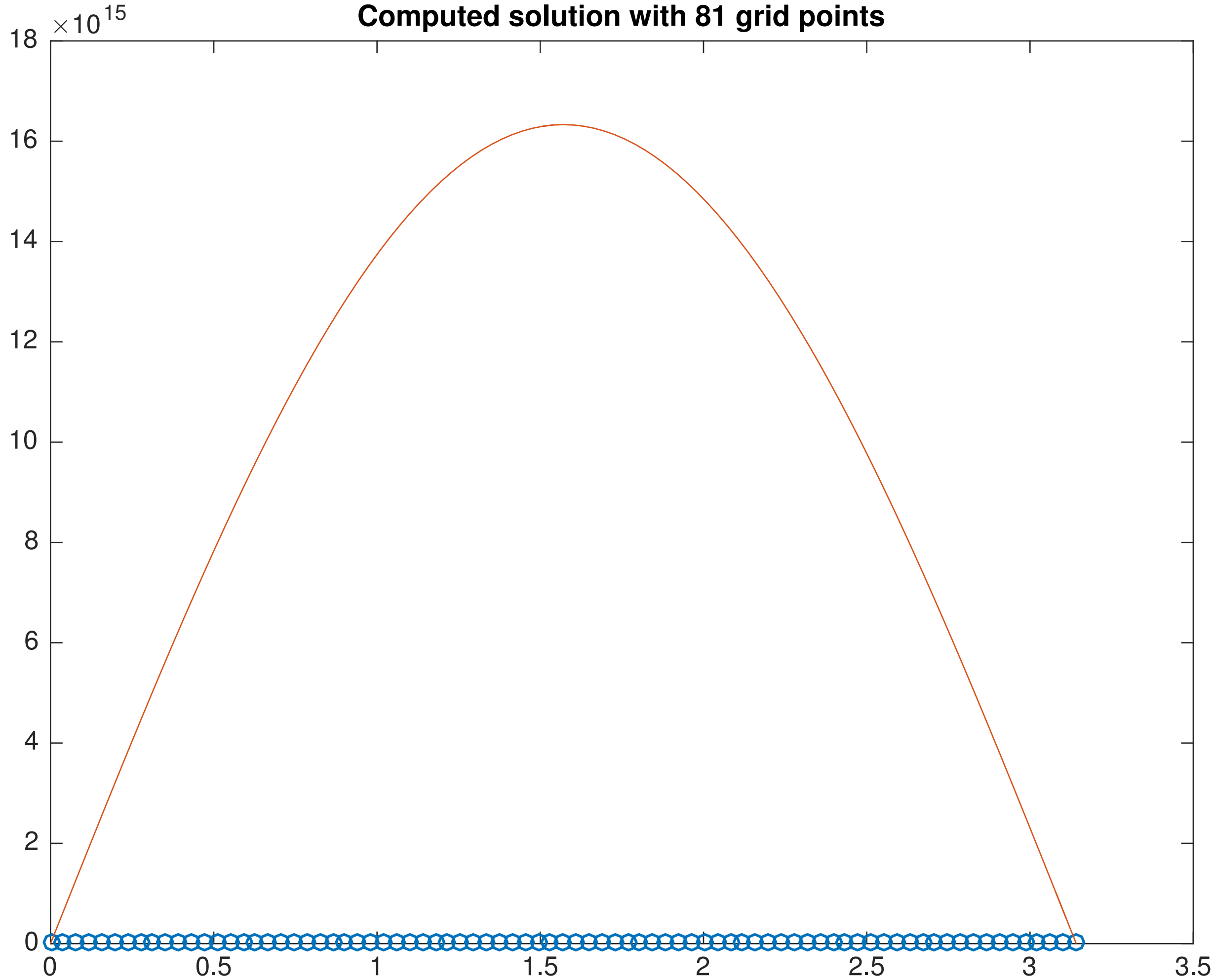


4.2.3 Part C

Solving the problem with $\beta = -1$ at π gives the following result:



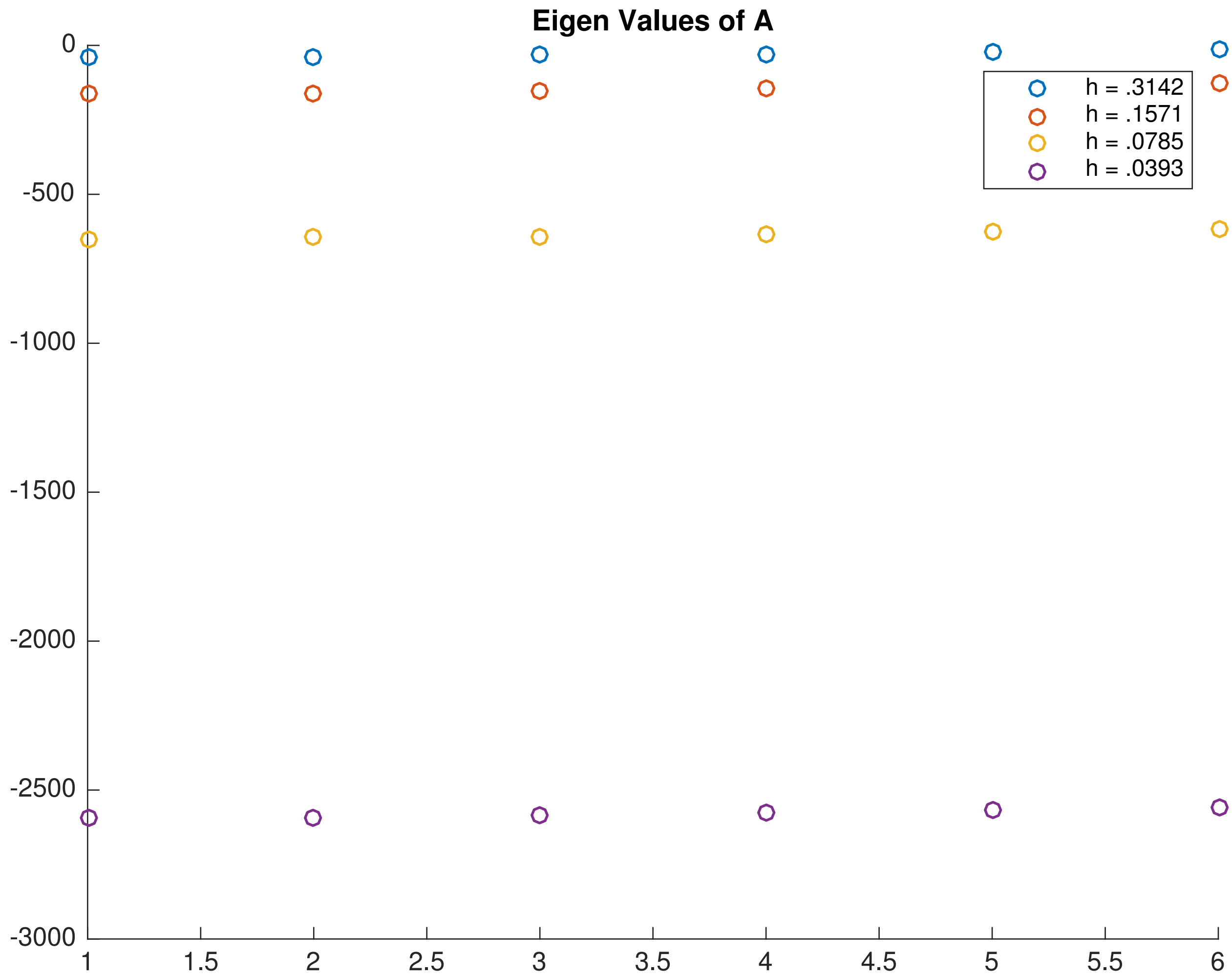
Using $\beta = 1$ at π results in:



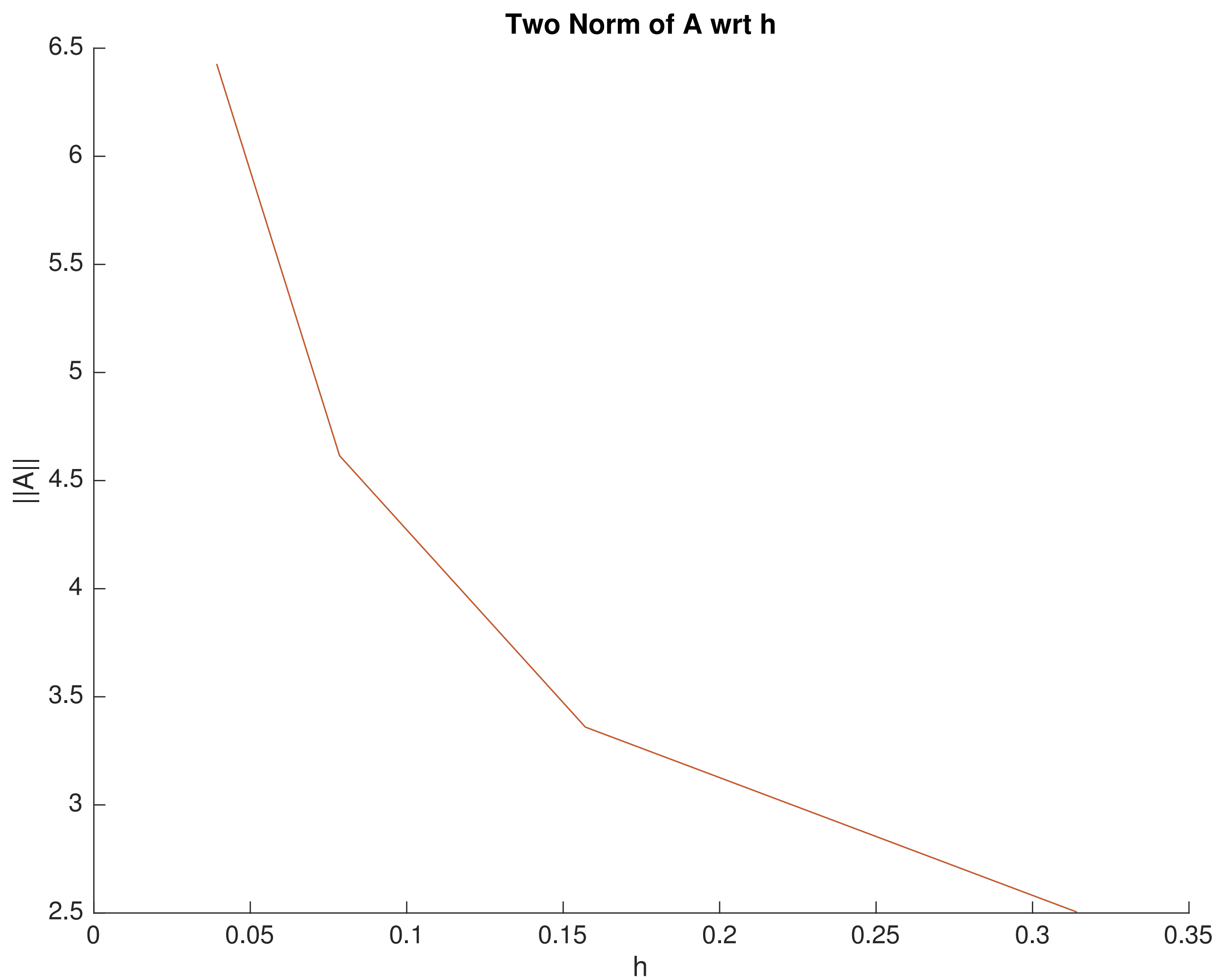
The BVP seems to converge to the linear solution as h approaches zero. When the boundary at b is changed to be the same as the boundary at a the solution does not converge.

4.2.4 Part D

The eigen values of A decrease as h approaches zero as shown in the figure below:



The two norm of A is seen to blow up as h approaches zero in the figure below:



5 Exercise 2.7

5.1 Problem Description

a. Write a program to solve the linear pendulum problem in order to achieve the same results as in figure 2.4 and 2.5. b. Increase T.

5.2 Problem Solution

The program code for the implementation of this program was handed in on polylearn for Professor Choboter to run.

5.2.1 Part A

In order to solve this non linear problem, I started with a guess for θ. I then used θ to determine the J matrix and G vector so that I could compute a δ to be added to the θ vector for the next iteration.

The matrix J was found using the following algorithm: $J =$

$$\begin{cases} 1/h^2 & j = i-1 || j = i+1 \\ -2/h^2 + \cos\theta & i == j \\ 0 & otherwise \end{cases}$$

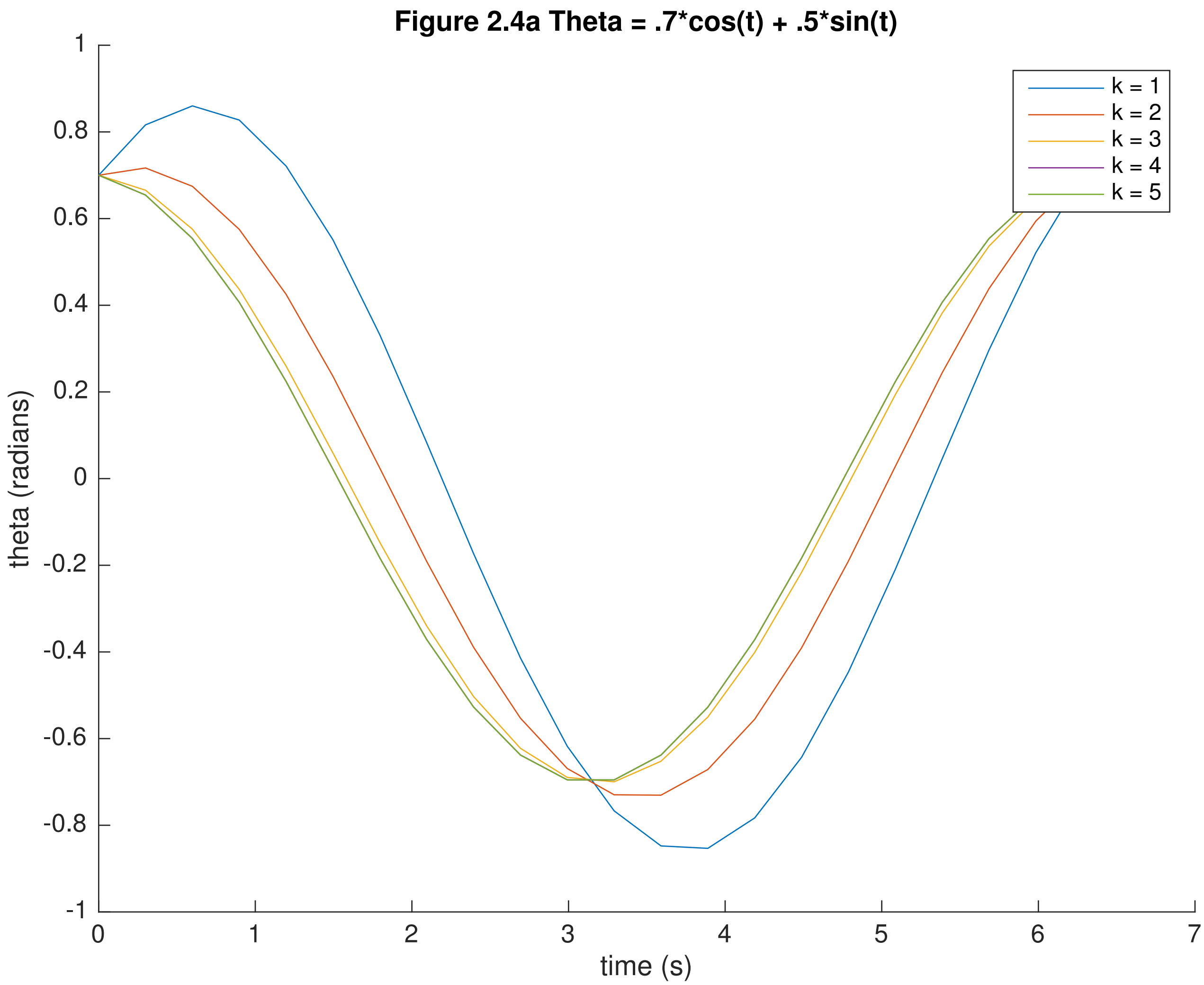
G was calculated with the following equation

$$G_i = 1/h^2 * (\theta_{i-1} - 2\theta_i + \theta_{i+1}) + sin(\theta_i)$$

With J and G found, the following equation could be used to find delta

$$J\delta = -G$$

Then δ was added to θ and the iterations continued until convergence was reached. The following figures demonstrate the success of the implementation as they exactly match the figures in the book.



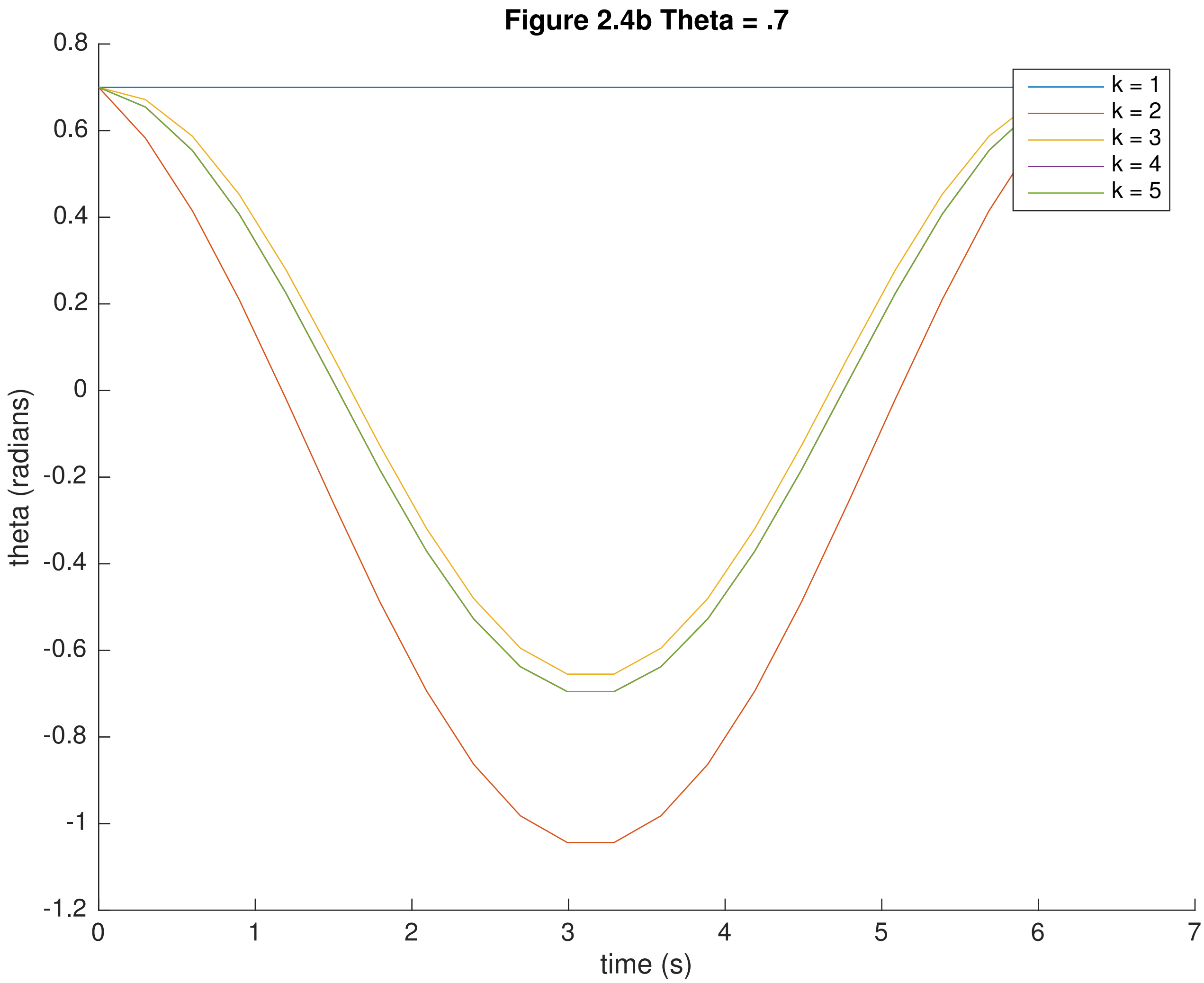


Figure 2.4b Theta = .7
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
theta (radians)
0
1
2
3
4
5
6
7
time (s)
k = 1
k = 2
k = 3
k = 4
k = 5

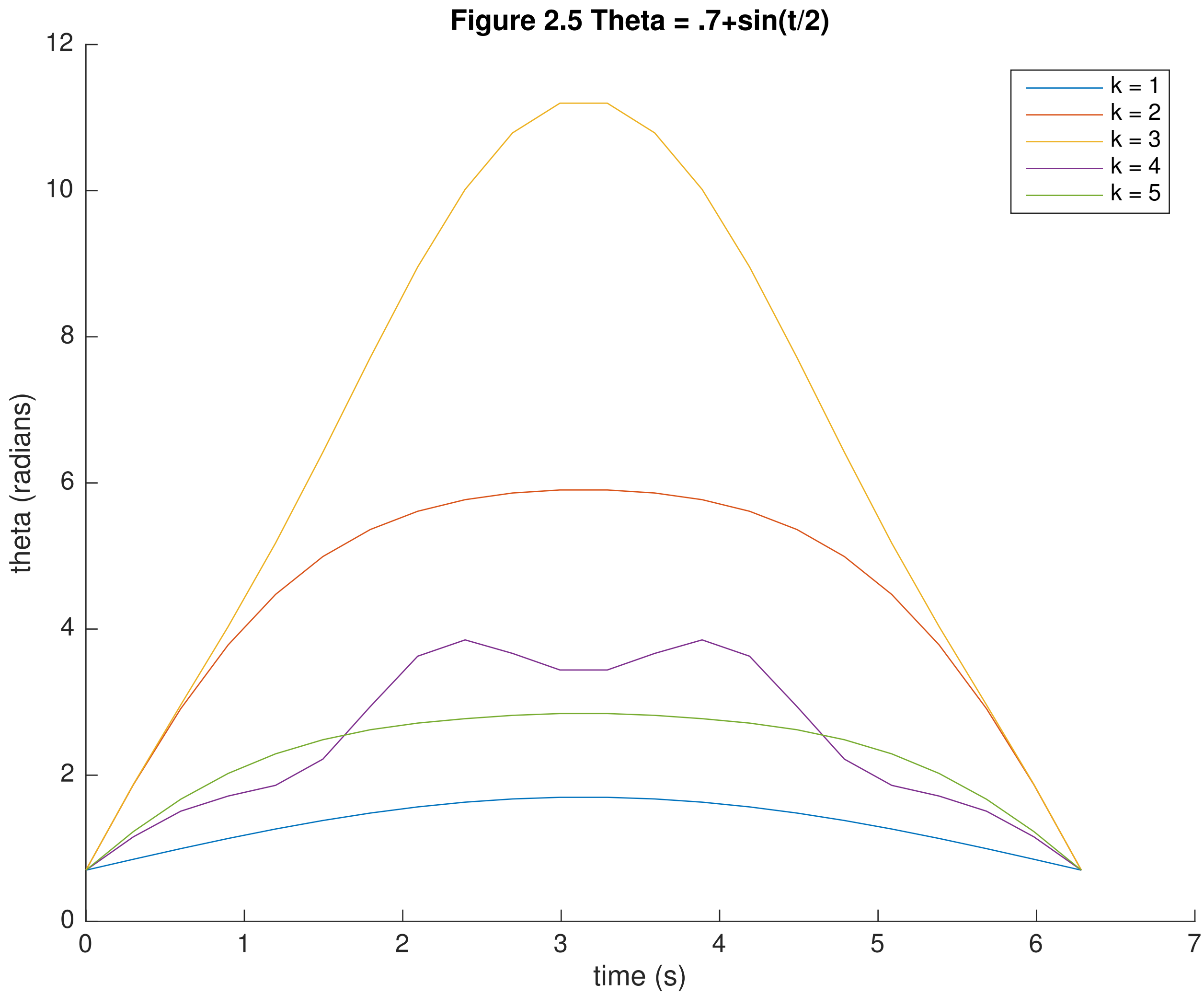
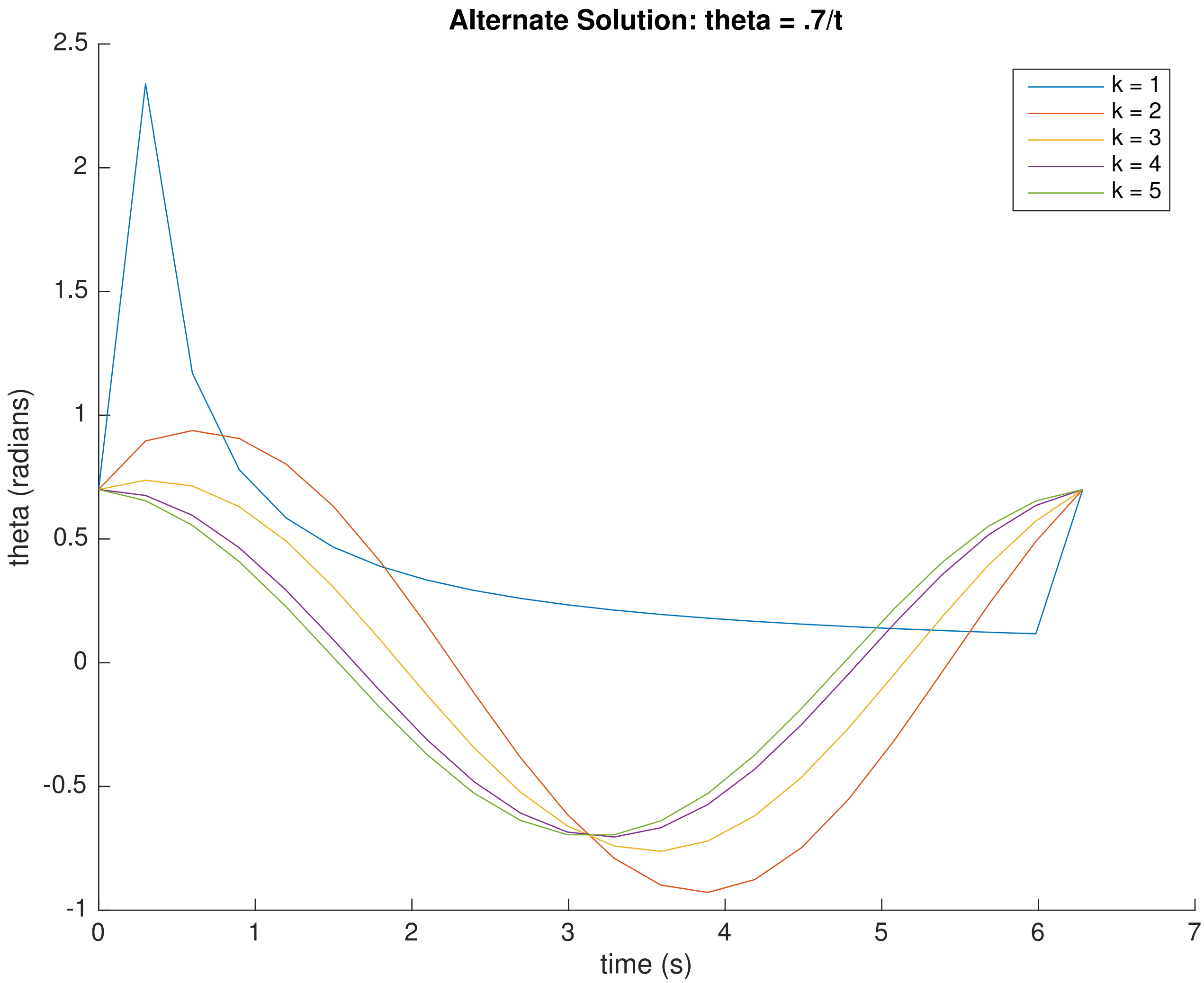
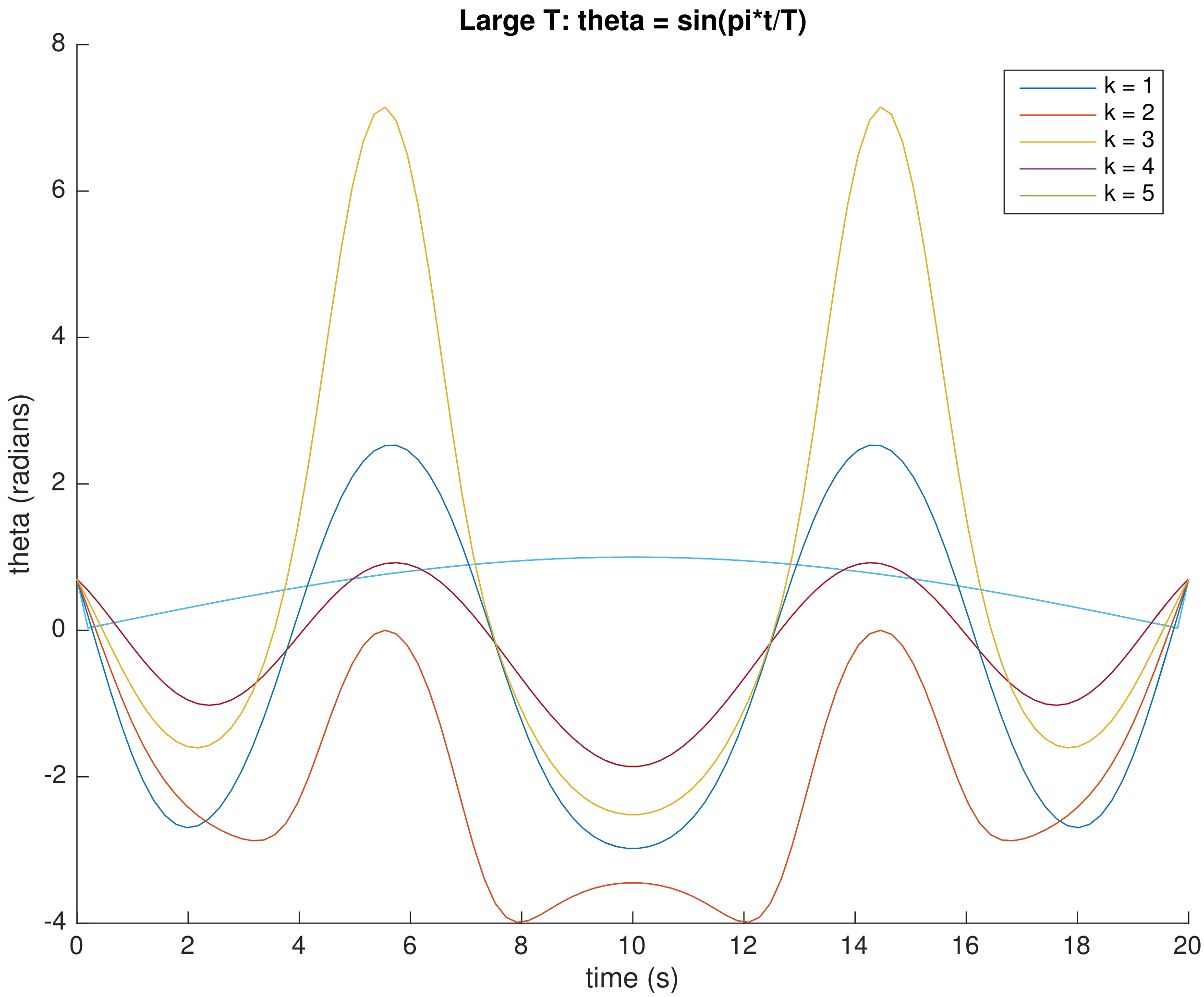


Figure 2.5 Theta = .7+sin(t/2)
12
10
8
6
4
2
0
0
1
2
3
4
5
6
7
theta (radians)
time (s)
k = 1
k = 2
k = 3
k = 4
k = 5

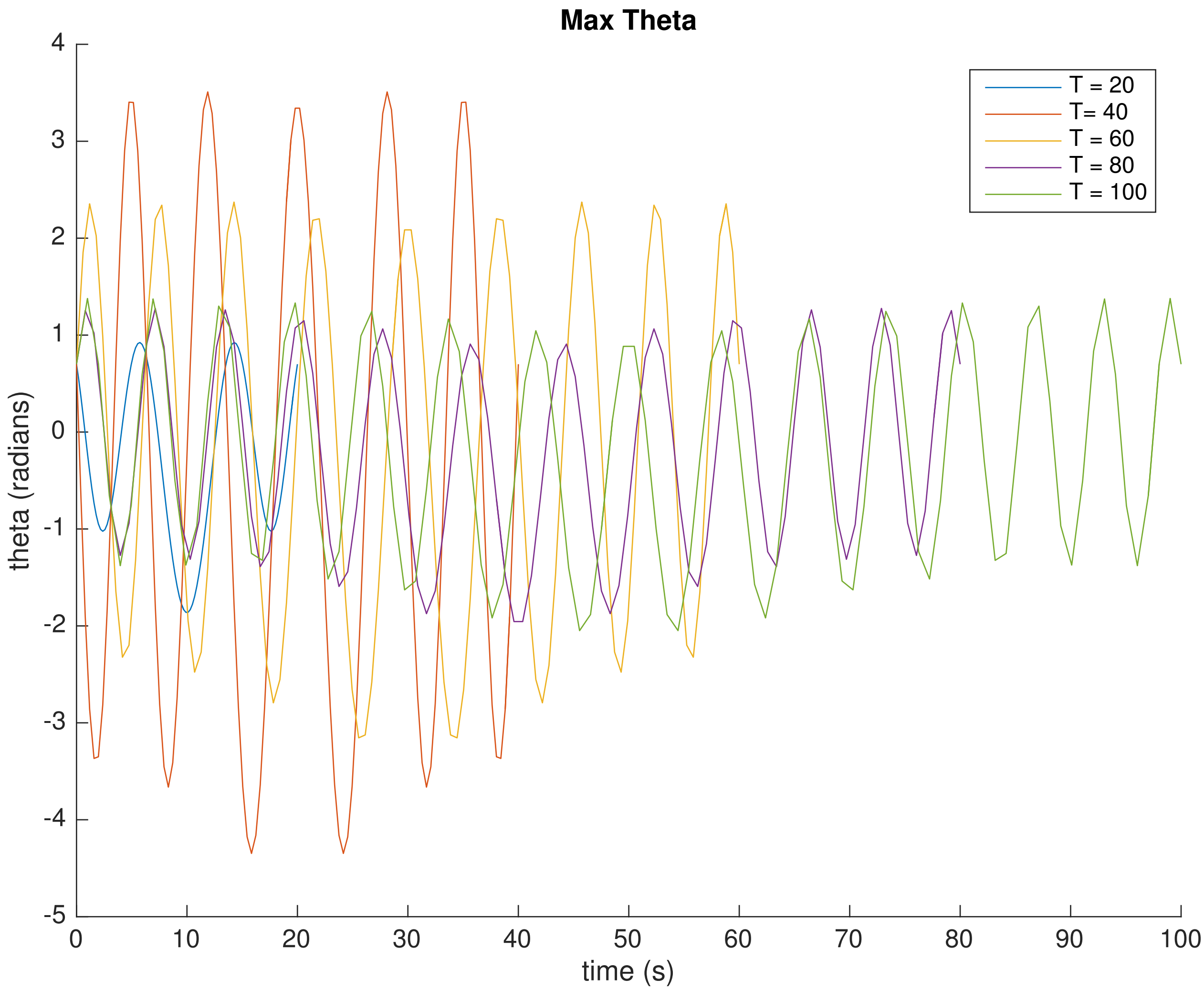


5.2.2 Part B

When T = 20, setting the solution to $\sin \pi * t/T$ results in a similar periodic motion to the motion seen in 2.5 in that the period of the wave is the entire span of the time, T.



The Max theta seems to be bounded by π.


Max Theta
T = 20
T= 40
T = 60
T = 80
T = 100
theta (radians)
time (s)
4
3
2
1
0
-1
-2
-3
-4
-5
0
10
20
30
40
50
60
70
80
90
100

A MATLAB Code