1 Problem

In this assignment we are given an ordinary second order differential equation (Poisson's Equation in one dimension)

$$-u''(x) = f(x) \tag{1}$$

with boundary conditions $u(0) = u(1) = 0$ and we seek to find a discrete approximate solution. We will achieve this by rewriting (1) in the approximate form

$$-v_{i+1} + 2v_i - v_{i-1} = h^2 f_i, \tag{2}$$

by using the discrete analog of the second order difference quotient. Here, $x_i = ih$ and $h = 1/(n+1)$, for $i = 1, 2, \dots, n$.

By substituting successive values of i, we can express (2) as the set of linear equations

$$\begin{array}{rcrcl} i=1: & & -v_0 + 2v_1 - v_2 & = & h^2 f_1 \\ i=2: & & -v_1 + 2v_2 - v_3 & = & h^2 f_2 \\ i=3: & & -v_2 + 2v_3 - v_4 & = & h^2 f_3 \\ \vdots & & \vdots & \vdots & \vdots \\ i=n: & & -v_{n-1} + 2v_n - v_{n+1} & = & h^2 f_n \end{array} \tag{3}$$

Which we can then write in matrix $Av = f$ format as (taking $v_0 = u(0) = v_{n+1} = u(1) = 0$)

$$\begin{bmatrix} 2v_1 & -v_2 & 0 & 0 & 0 & 0 & \dots & 0 \\ -v_1 & 2v_2 & -v_3 & 0 & 0 & 0 & \dots & 0 \\ 0 & -v_2 & 2v_3 & -v_4 & 0 & 0 & \dots & 0 \\ 0 & 0 & -v_3 & 2v_4 & -v_5 & 0 & \dots & 0 \\ 0 & 0 & 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 & -v_{n-2} & 2v_{n-1} & -v_n \\ 0 & 0 & 0 & \dots & 0 & 0 & -v_{n-1} & 2v_n \end{bmatrix}$$

$$= \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & \dots & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & \dots & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & \dots & 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ \vdots \\ \vdots \\ v_{n-1} \\ v_n \end{bmatrix} = h^2 \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ \vdots \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix}$$

Thus, we see that the problem is reduced to that of solving the tridiagonal linear system above. We will accomplish this by decomposing the coefficient matrix into upper and lower bidiagonal matrices U and L and solving the associated (much easier to work with) systems $Lg = f$ and $Uv = g$ for the n dimensional vector v.

Note that the vector v is exactly that which has components v_i satisfying the ith equation of (3). That is, v_i is an approximation of the value $u(x_i)$. Hence, the vector v is a collection of n points approximating the function $u(x)$.

For this assignment we will be assuming that $f(x) = 100e^{-10x}$. With this choice of f, (1) has closed solution $u(x) = 1 - (1 - e^{-10})x - e^{-10x}$, which we will be comparing our approximations to for various choices of n.

To verify that $u(x)$ as given is a solution of (1), refer to the following calculation.

$$\begin{aligned} -u''(x) &= -[1 - (1 - e^{-10})x - e^{-10x}]'' \\ &= -[(1 - e^{-10}) + 10e^{-10x}]' \\ &= -(-100e^{-10x}) \\ &= 100e^{-10x} \\ &= f(x) \end{aligned}$$

2 Source Code

```
% File Name: main_hw3.m
% Assignment: Project 3
% Student: Joseph Free
% Course: MATH3261
%
% Purpose: This function is the driver for project 3. The functions
% tridiag and triSolver are utilized here to find the solution to
% to the Ax = f, were A is the tridiagonal matrix resulting from the
% system generated by -v_i+1 + 2v_i - v_i-1 = h^2*f_i and f is the vector
% with components f(x_i). The function returns a tuple with the grid
% points x_i and the solution vector X.
%
% Required input:
%
%   n         -- The number of points used in partitioning [0,1].
%
function [ x_i,X ] = main_hw3( n )
    h = 1/(n+1);
    x_i = h:h:n*h;
    f_i = (100*exp(-10*x_i)) * h.^2;
    u_i = 1-(1-exp(-10))*x_i -exp(-10*x_i);

    A = diag( 2* ones(1,n) );
    for (i = 2:n)
        A(i, i-1) = -1;
        A(i-1, i) = -1;
    end
    [L,U] = tridiag_hw3(n, A);
    X = triSolver_hw3(L,U,f_i);
end
```

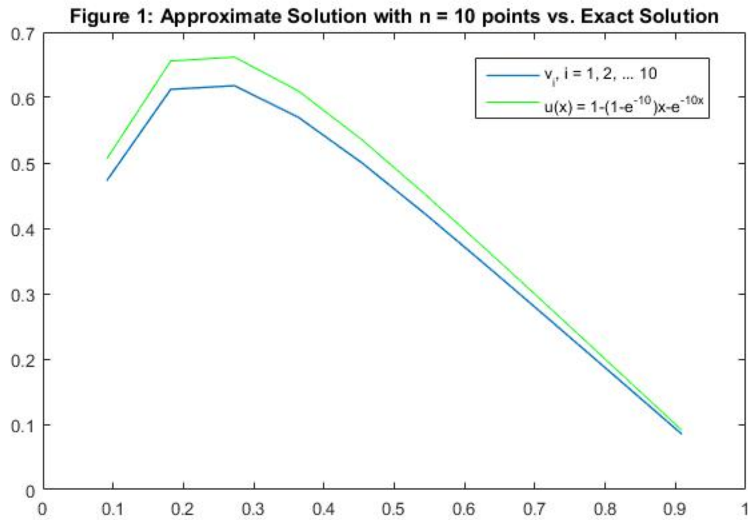
```
% File Name: tridiag_hw3.m
% Assignment: Project 3
% Student: Joseph Free
% Course: MATH3261
%
% Purpose: This function takes a tridiagonal matrix A and its dimensions
% and returns the individual factors of the LU decomposition. That is,
% this function returns a tuple (L,U) where L is a lower bidiagonal
% matrix, U is an upper diagonal matrix, and the product LU = A.
%
% Required input:
%
%   A         -- A square tridiagonal matrix.
%   n         -- The dimension of the square tridiagonal matrix.
%
%
function [ L, U ] = tridiag_hw3( n, A )
    c = diag(A,1);
    U = diag(c,1);
    L = diag( diag(ones(n)), 0 );

    U(1,1) = A(1,1);
    for (j = 2:n)
        L(j,j-1) = A(j-1,j)/U(j-1,j-1);
        U(j,j) = A(j,j)-L(j,j-1)*U(j-1,j);
    end

end
```

```
% File Name: triSolver_hw3.m
% Assignment: Project 3
% Student: Joseph Free
% Course: MATH3261
%
% Purpose: This function takes the LU decomposition of a coefficient
% matrix A of a triangular linear system Ax = f, in addition to the vector
% f, and returns the vector x that satisfies the system.
%
% Required input:
%
%   L         -- A lower bidiagonal matrix.
%   U        -- An upper bidiagonal matrix.
%   f         -- The vector f such that LUx = f.
%
function [ x ] = triSolver_hw3( L, U, f)
    n = length(L);
    g = zeros(1,n);
    x = zeros(1,n);

    g(1) = f(1);
    for (j = 2:n)
        g(j) = f(j)- L(j,j-1)*g(j-1);
    end

    j = n-1;
    x(n) = g(n)/U(n,n);
    while( j > 0 )
        x(j) = (g(j) - U(j,j+1)*x(j+1))/U(j,j);
        j = j-1;
    end

end
```

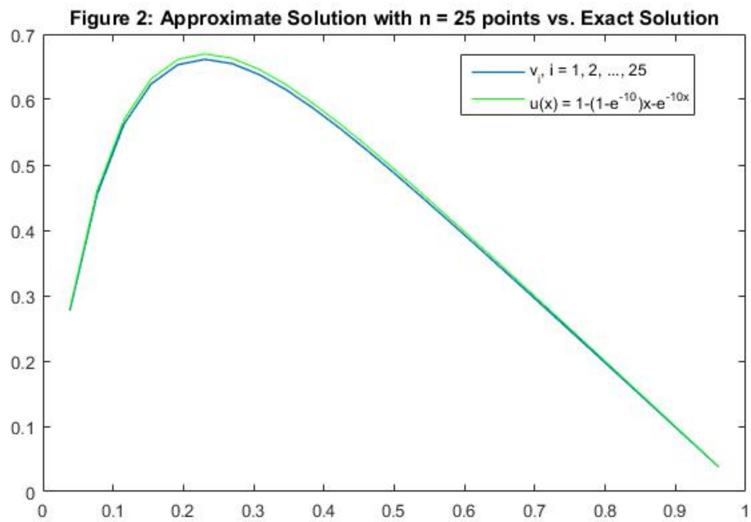
3 Results

Using the code from section two, five sets of approximations were made for n = 10, 25, 40, 50, and 1000. For each approximate solution, plots were made against the exact solution $u(x) = 1 - (1 - e^{-10})x - e^{-10x}$ and relative error E calculated in order to determine accuracy.

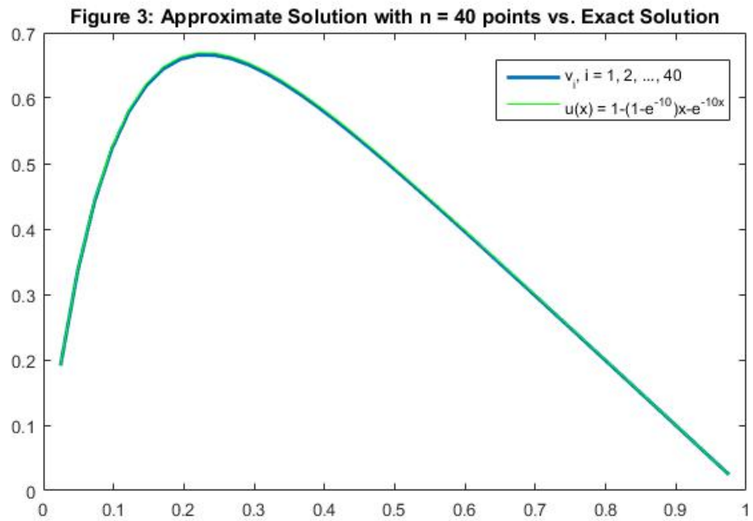
As was to be expected, the approximation using n = 10 points was hardly accurate. However, for larger values of n, v tended to approach $u(x_i)$ very rapidly. In fact with just 25 points, v does an admirable job of approximating the solution of (1).



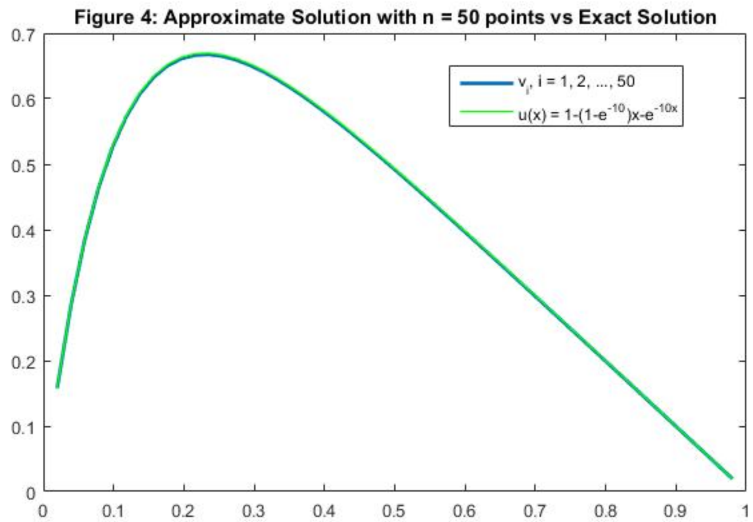
Relative error, $E = -1.17969$



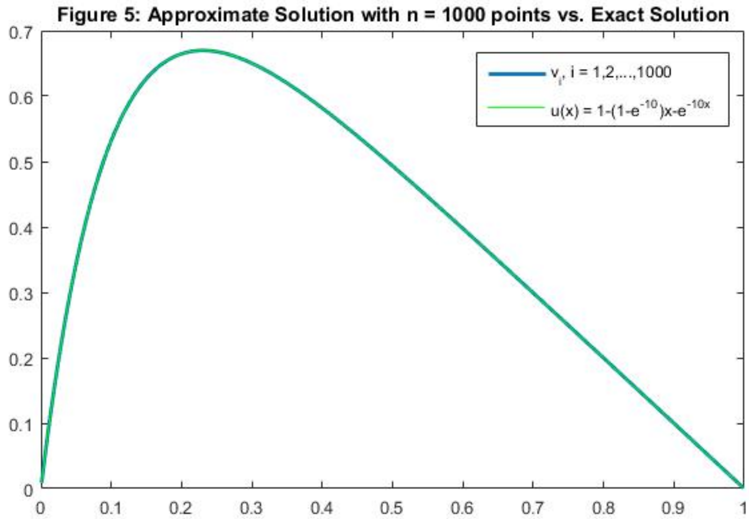
Relative error, $E = -1.91233$



Relative error, $E = -2.30603$



Relative error, $E = -2.49515$



Relative error, $E = -5.08005$