1. Write a code to solve a linear system $Ax = b$ using Gaussian elimination with partial pivoting. Use it to solve the system $Ax = b$, where

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 1 & 2 & 2 \\ -2 & 1 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 2 \\ 5 \\ 0 \end{pmatrix}.$$ 

Hand in your code and output.

**Note:** If you do this in MATLAB (and I encourage you to), try to take as much advantage as possible of MATLAB’s array operations to eliminate loops in your code. For example, the loop

$$\text{For } i = k + 1, \ldots, n$$
$$b_i = b_i + a_{ik}b_k$$

can be written in one MATLAB command line as

$$b(k+1:n) = b(k+1:n) + b(k)*A(k+1:n,k);$$

MATLAB code is interpreted, rather than compiled, and stepping through the command lines in a code is a relatively slow process. Since loops often require many repetitions of the lines within them, they are a notorious cause of slow MATLAB code. Replacing loops by array operations wherever possible can often yield much faster code. I won’t grade you on how successful you are in eliminating loops; my goal is just to encourage you to do your best to write good, fast code.

2. (Adapted from problem 4, page 575 of the text.) Consider approximately solving the integral equation

$$x(s) - \int_0^1 \cos(\pi st)x(t) \, dt = 1, \quad 0 \leq s \leq 1$$

by discretizing the integral with the midpoint numerical integration rule (which we will study later) as follows: For a given positive integer $n$, define $n$ grid or mesh points $t_i$ in $[0, 1]$ by setting $h = 1/n$ and $t_i = (i - 1/2)h$ for $i = 1, \ldots, n$. Then obtain approximate solution values $x_i \approx x(t_i)$ by solving the system of linear equations

$$x_i - \sum_{j=1}^{n} h \cos(\pi t_i t_j)x_j = 1, \quad i = 1, \ldots, n.$$ 

Solve this system with $n = 64$. You can use any software you choose, e.g., your code from problem 1 or MATLAB’s “backslash” command. If possible, plot the approximate solution values. (In MATLAB, this is done with `plot(t,x)`, where $t$ and $x$ are the vectors of $t_i$’s and $x_i$’s, respectively.) If you can’t plot the solution for some reason, print out the $t_i$’s and $x_i$’s. Hand in your code and your plot or printout.

**Note:** The linear system can be written in matrix-vector form as $(I - K)x = b$, where $K_{ij} = h \cos(\pi t_i t_j)$ and $b_i = 1$. 