**Question 1:**

To be able to solve for x in the given systems of equations, it was necessary to be able to generate matricies A and b for any given n-size. I implemented generateA\_b.m to do this; the eponymous function takes in an n-size and returns A and b matricies.

The A and b matricies generated are then used as arguments for the pentsolve function, which returns x, the goal of question 1. x is stored in the first column of the results matrix, in preparation for question 2.

Here is an example of the generated matricies:

Table 1: Matricies A and b ( n = 7)

|  |  |
| --- | --- |
| **Matrix A** | **Matrix b** |
| 9 -4 1 0 0 0 0  -4 6 -4 1 0 0 0  1 -4 6 -4 1 0 0  0 1 1 1 1 1 0  0 0 1 -4 6 -4 1  0 0 0 1 -4 5 -2  0 0 0 0 1 -2 1 | 0.000416493127863390  0.000416493127863390  0.000416493127863390  0.000416493127863390  0.000416493127863390  0.000416493127863390  0.000416493127863390 |

**Question 2:**

Using the A and b matricies generated for question 1, the systems of equations was solved using Matlab’s built-in solver; thus, the result of A\b was stored in column 2 of the results matrix. Here are some comparisons between the solutions generate from the Matlab solver and the full solver:

Table 2: solutions for system of equations (n = 7)

pentsolver solution MatLab solution

-0.000219688682829041 -0.000219688682829041

-0.00109844341414520 -0.00109844341414521

-0.00200008238325606 -0.00200008238325606

-0.00187193065160579 -0.00187193065160579

0.000755179847224828 0.000755179847224828

0.00463176972964561 0.00463176972964562

0.00892485273992978 0.00892485273992980

Table 3: solutions for system of equations (n = 100)

pentsolver solution MatLab solution

-5.31914893617020e-09 -5.31914893617021e-09

-2.48226950354610e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163120e-08

-2.80141843971632e-08 -2.80141843971631e-08

5.24822695035458e-08 5.24822695035460e-08

5.17730496453903e-08 5.17730496453901e-08

-2.48226950354610e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163121e-08

-2.80141843971632e-08 -2.80141843971631e-08

5.24822695035459e-08 5.24822695035460e-08

5.17730496453902e-08 5.17730496453901e-08

-2.48226950354609e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163120e-08

-2.80141843971632e-08 -2.80141843971631e-08

5.24822695035460e-08 5.24822695035460e-08

5.17730496453901e-08 5.17730496453901e-08

-2.48226950354609e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163120e-08

-2.80141843971632e-08 -2.80141843971631e-08

5.24822695035460e-08 5.24822695035460e-08

5.17730496453901e-08 5.17730496453901e-08

-2.48226950354609e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163121e-08

-2.80141843971632e-08 -2.80141843971631e-08

5.24822695035459e-08 5.24822695035460e-08

5.17730496453902e-08 5.17730496453901e-08

-2.48226950354609e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163120e-08

-2.80141843971631e-08 -2.80141843971631e-08

5.24822695035459e-08 5.24822695035460e-08

5.17730496453902e-08 5.17730496453901e-08

-2.48226950354609e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163120e-08

-2.80141843971631e-08 -2.80141843971631e-08

5.24822695035459e-08 5.24822695035460e-08

5.17730496453902e-08 5.17730496453901e-08

-2.48226950354610e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163120e-08

-2.80141843971631e-08 -2.80141843971631e-08

5.24822695035459e-08 5.24822695035460e-08

5.17730496453902e-08 5.17730496453901e-08

-2.48226950354610e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163120e-08

-2.80141843971631e-08 -2.80141843971631e-08

5.24822695035459e-08 5.24822695035460e-08

5.17730496453902e-08 5.17730496453901e-08

-2.48226950354609e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163120e-08

-2.80141843971631e-08 -2.80141843971631e-08

5.24822695035459e-08 5.24822695035460e-08

5.17730496453902e-08 5.17730496453901e-08

-2.48226950354609e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163120e-08

-2.80141843971632e-08 -2.80141843971631e-08

5.24822695035458e-08 5.24822695035460e-08

5.17730496453903e-08 5.17730496453901e-08

-2.48226950354609e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163120e-08

-2.80141843971632e-08 -2.80141843971631e-08

5.24822695035458e-08 5.24822695035460e-08

5.17730496453903e-08 5.17730496453901e-08

-2.48226950354609e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163120e-08

-2.80141843971631e-08 -2.80141843971631e-08

5.24822695035458e-08 5.24822695035460e-08

5.17730496453903e-08 5.17730496453901e-08

-2.48226950354609e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163121e-08

-2.80141843971631e-08 -2.80141843971631e-08

5.24822695035458e-08 5.24822695035460e-08

5.17730496453903e-08 5.17730496453902e-08

-2.48226950354609e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163121e-08

-2.80141843971631e-08 -2.80141843971631e-08

5.24822695035458e-08 5.24822695035460e-08

5.17730496453903e-08 5.17730496453902e-08

-2.48226950354609e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163121e-08

-2.80141843971631e-08 -2.80141843971631e-08

5.24822695035458e-08 5.24822695035460e-08

5.17730496453903e-08 5.17730496453902e-08

-2.48226950354609e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163121e-08

-2.80141843971631e-08 -2.80141843971631e-08

5.24822695035458e-08 5.24822695035460e-08

5.17730496453903e-08 5.17730496453902e-08

-2.48226950354609e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163121e-08

-2.80141843971631e-08 -2.80141843971631e-08

5.24822695035458e-08 5.24822695035460e-08

5.17730496453903e-08 5.17730496453902e-08

-2.48226950354609e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163120e-08

-2.80141843971631e-08 -2.80141843971631e-08

5.24822695035458e-08 5.24822695035460e-08

5.17730496453903e-08 5.17730496453902e-08

-2.48226950354609e-08 -2.48226950354610e-08

-4.14184397163120e-08 -4.14184397163121e-08

-2.80141843971631e-08 -2.80141843971631e-08

-4.60992907801422e-09 -4.60992907801419e-09

As seen above, the highlighted portions in the tables note the discrepancies in the solutions obtained using the two methods; I didn’t highlight all of them, as it would be too time consuming. These discrepancies, while small, indicate the the solutions generated are not equal.

**Question 3:**

This question was similar to question 1 in terms of the approached used to solve it. Instead of having only one initial x, 25 iterations must be made for each x initialization beginning from (400,400) to (600,600), inclusively. While the first question only had one for loop to handle the iteration, an outer loop had to be created for this question to iterate through all the initial x-values. Another notable difference is that iteration didn’t terminated upon immediately hitting the approved convergence error threshold. 25 iterations were made for each initial starting value of x. This resulted in a large (2 x 25 x 201)—2 representing x1 and x2, 25 for the number of iterations per initial starting value, and 201 for account all the initial starting values between 400 and 600. Otherwise, the methodology remained the same.

A table was created in the end, to display the values where the system converged toward with respect to the initial starting values.

By setting ε = 10-3, the following result was obtained. Please see below the table to understand how to interpret it:

**(x1, x2)** – because x1 =x2 in this problem, the lone number in this column represents both integers, where (x1, x2) is the **initial value of x**.

**f1(x) and f2(x)** are placed side-by-side to see the value they are converging toward

**Conv?** column is a Boolean, where 1 = true (the system does converge) and 0 = false (the system does not converge)

Thus, viewing the values in the table where Conv? is 1 (true), it is plausible to assume that they are converging toward the value of 0. There are 25 different solutions where the system converges.

**Sources of Information:**

Sources of information include the slides posted on Canvas, as well as MatLab’s documentation on routines such as zero() and fsolve(). Also, I did browse the Discussion forum on Canvas for tips and help, as well as going to TA hours to obtain assistance

I did not collaborate with any classmates on this assignment.

As far as source codes went, I implemented all the functions that didn’t already come with MatLab or were not handed out. In this case, I implemented the function *generateA\_b.m* while *pentsolve.m* was doled out for our use.