Solutions to Programming Assignment Four – Simultaneous Linear Equations

1. Using MATLAB

a. Import the workbook data for A, b, and x into MATLAB. Use MATLAB commands to solve for all the x solutions. Find the RMS errors, defined as $\{[\Sigma(x_{correct,k} - x_{MATLAB,k})^2]/N\}^{1/2}$, in this solution.

```
format compact
%Get solution vectors in xs
xs = A\b;
%Get error vectors in e for each x in each solution
e=xs-x;
%Get RMS errors for each solution.  (MATLAB sum gets column sums)
rms = sqrt(sum(e.^2)/100)
rms =
  1.0e-011 *
  Columns 1 through 10
    0.1020    0.0682    0.0762    0.0647    0.1120    0.1269
0.0662    0.0805    0.0980    0.1146
  Columns 11 through 12
    0.0604    0.0918
min(rms)
ans =
   0.0604e-11
max(rms)
ans =
   0.1269e-11
```

b. Determine A^{-1} and the product of AA^{-1}. Discuss the error in this product. Note that you can use the MATLAB command `eye(N)` to construct an NxN unit matrix.

```
%Compute B as A inverse and error in AB product as matrix e
B=inv(A);
e = A*B-eye(100);
%Compute rms error for total solution
rms = sqrt(sum(sum(e.^2))/10000)
ans =
  7.9193e-016
```

c. Determine Det(A). You will be asked below to compare it to the value found with Excel and to discuss any differences.

```
%Compute determinant
format long
det(A)
ans =
    7.588546097841507e+249
```

d. Review section 3.6 in the Rao text on condition numbers.

i. Use the cond function of MATLAB to determine the condition number of the A matrix.

```
Compute condition number of A and B = A inverse
cond(A)
ans =
    5.654396821713309e+002
%As expected cond(A) and cond(B = A-inverse) are the same
cond(B)
ans =
    5.654396821713143e+002
```

ii. Compute your own condition number using the definition of the condition number as ||A|| ||A^{-1}||, using each of the four matrix norms available in MATLAB. Discuss the differences between these norms. Are any of them the same as the one you found in part a?

```
%Compute condition numbers from different matrix norms
norm(A,1)*norm(B,1)
ans =
    2.338658667495540e+003
norm(A,2)*norm(B,2)
ans =
    5.654396821713157e+002
norm(A,inf)*norm(B,inf)
ans =
    2.467951744747787e+003
norm(A,'fro')*norm(B,'fro')
ans =
    1.596977823842042e+003
%We see that the 2 norm gives us the same result as the cond
function
```

iii. Define a new matrix which is a submatrix that consists of the first three rows and the first three columns of the A matrix. What is the condition number of this submatrix according to the cond function? Discuss the differences between the condition number for the A matrix and the one for the submatrix.

```
%Get subarray of A and its condition number
A3=A(1:3,1:3)
A3 =
  60.899999999999999 -63.200000000000003 -86.400000000000006
  16.800000000000001  98.500000000000000 -79.900000000000006
  27.000000000000000  74.799999999999997  84.299999999999997
cond (A3)
ans =
   2.621937384352283
%Smaller size arrays with similar data have lower condition numbers
```

e. Enter the following arrays into MATLAB A = [1 -2 3 -5; 4 8 6 -10; -3 2 -7 8; 6 0 13 -18] and b = [-14; -2; 12; -27]. Discuss the number of solutions (one, infinite, or none) for these matrices in the problem AX = b.

```
%Get new matrix A and rhs vector b
A = [1 -2 3 -5; 4 8 6 -10; -3 2 -7 8; 6 0 13 -18]
A =
     1    -2     3    -5
     4     8     6   -10
```

```
     -3      2     -7      8
      6      0     13    -18
b = [-14; -2; 12; -27]
b =
   -14
    -2
    12
   -27
%Check rank(A) and rank ([A b]) to see available solutions
rank(A)
ans =
     3
rank([A b])
ans =
     3
%rank(A) = rank ([A b]) = 3 < 4 =  number of equations
%have infinite number of solutions
```

f. Use the `pinv` function of MATLAB to obtain a solution to the set of equations from part d, and use the `rref` function to obtain the reduced row-echelon form for [A b]. Use this form to get an algebraic solution for x_1, x_2, and x_3 in terms of any arbitrary value for x_4. Show that the equations that you find here are consistent with the solution you obtained using the `pinv` function. In the discussion show your equations and your verification that both solutions are consistent.

```
%Use pinv to get solution with minimum 2 norm
pinvX = pinv(A)*b
pinvX =
   1.912074624287451
   1.847987562618759
   2.502504750388673
   3.944722750043194
%check answer
e = A*pinvX-b
e =
  1.0e-013 *
  -0.142108547152020
  -0.284217094304040
   0.035527136788005
  -0.426325641456060
%Use rref to get reduced row-echelon form
ref = rref([A b])
ref =
1.000000000           0           0  16.500000000  67.000000000
          0 1.000000000           0  -2.750000000  -9.000000000
          0           0 1.000000000  -9.000000000 -33.000000000
          0           0           0             0             0
%Define anonymous function to compute x(1:3) from any x4 value
%using rref solution
rrefff =
    @(x4)ref(1:3,5)-x4*ref(1:3,4)
%Compute x(1:3) from pinv solution for pinvX(4) using rref equations
res=rreff(pinvX(4))
res =
   1.912074624287300
```

```
   1.847987562618783
   2.502504750388745
%rref calculations match pinv solution
```

2. Using Excel

Use the downloaded workbook for the assignments below. Do all your work in one workbook. Use separate worksheets for the various assignments.

A modified copy of this workbook with all the solutions discussed below is available at the same web site where you found this solution *via* the link Assignment Solutions Four Workbook.

a. Use the data in the workbook and the Excel `minverse` and `mmult` functions to find the solutions for x. For each x find the root mean square error (RMS) in the each solution defined as $\{[\Sigma(x_{correct,i} - x_{Excel,i})^2]/N\}^{1/2}$. Discuss any differences that you find between the errors using MATLAB and the errors using Excel

The errors for all the Excel operations required in this part are summarized in the following table from the Excel workbook for the calculations.

RMS Errors in Solutons of **Ax** = **b**					
Index for **x** solutions	Excel minverse routine	Gaussian Elimination Routines			
		Type Double		Type Single	
		Pivoting	No Pivoting	Pivoting	No Pivoting
1	6.98E-12	2.51E-12	3.07E-11	2.45E-03	1.40E-02
2	5.80E-12	2.25E-12	2.50E-11	1.28E-03	1.35E-02
3	6.18E-12	2.61E-12	2.47E-11	2.47E-03	1.22E-02
4	6.77E-12	5.50E-12	1.86E-11	2.10E-03	1.45E-02
5	5.75E-12	4.77E-12	9.78E-12	1.56E-03	8.67E-03
6	6.78E-12	2.63E-12	1.47E-11	1.19E-03	1.77E-02
7	5.88E-12	2.91E-12	3.84E-11	1.22E-03	1.10E-02
8	6.39E-12	7.12E-12	2.77E-11	1.75E-03	1.28E-02
9	6.61E-12	4.79E-12	3.09E-11	2.70E-03	7.39E-03
10	7.32E-12	2.43E-12	3.35E-11	3.41E-03	1.69E-02
11	6.64E-12	4.94E-12	2.86E-11	2.47E-03	1.40E-02
12	7.54E-12	2.15E-12	1.46E-11	4.18E-03	1.30E-02
Maximum	7.54E-12	7.12E-12	3.84E-11	4.18E-03	1.77E-02
Minimum	5.75E-12	2.15E-12	9.78E-12	1.19E-03	7.39E-03
RMS RMS	7.20E-12	4.42E-12	2.86E-11	2.63E-03	1.46E-02

The VBA type double codes with pivoting produce an overall error that is slightly less than that of the Excel routines. Note that pivoting produces some increase in accuracy of more than a factor of ten,, but the use of type double instead of type single produces a much greater increase in accuracy. The Excel routines produce a more uniform RMS error (the variation in the RMS error among the various solutions). The MATLAB results (minimum/maximum errors of 0.604×10^{-12} to 1.25×10^{-12} were the smallest of all. These errors depend on the computer used. Different computers give different values of these roundoff errors for the MATLAB solution of simultaneous linear equations.

b. In this step you will see the effects of the choice of single or double data types and pivoting or not pivoting. Discuss the differences that you find in the RMS errors for each of these approaches below and the one in part a, using the Excel functions to solve Ax = b.

i. **Obtain the solutions using the code as written. Determine the RMS error in these solutions.**

ii. **Modify the code to obtain solutions with type single instead of type double. Determine the RMS error in these solutions.**

iii. **Modify the code to obtain solutions with type single without pivoting. Determine the RMS error in these solutions.**

iv. **Modify the code to obtain solutions with type double without pivoting. Determine the RMS error in these solutions.**

See the table above and the discussion following it for these results.

c. **Modify the code to compute the determinant of any input matrix and use this to compute Det(A). Compare this value to the one found in MATLAB and the one found using the Excel function mdeterm. Discuss any differences.**

The following three values were found for the determinant

MATLAB:	7.588546097841507e+249
Excel mdeterm	7.588546097841390E+249
VBA Code	7.588546097840920E+249

We see that there is a difference in the last three significant figures.