Chapter 6: Linear Systems

Joseph Sepich

Contents

1 Problem 1 Gaussian Elimination

1.1 a.

Consider the following 3x3 system:

$$3.3330x_1 + 15920x_2 - 10.333x_3 = 7973.6$$
$$2.2220x_1 + 16.710x_2 + 9.6120x_3 = 0.96500$$
$$-1.5611x_1 + 5.1792x_2 - 1.6855x_3 = 2.7140$$

Verify that the solution is $x_1 = 1, x_2 = 0.5, x_3 = -1$.

Solve the following system using naive Gaussian elimination with a calculator. All calculations should use 5 significant digits. Show details of your work.

Starting matrix:

$$\begin{pmatrix} 3.3330 & 15920 & -10.333 \\ 2.2220 & 16.710 & 9.6120 \\ -1.5611 & 5.1792 & -1.6855 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7973.6 \\ 0.96500 \\ 2.7140 \end{pmatrix}$$

Step 1: $a_1 = \frac{1}{3.333} a_1$

$$\begin{pmatrix} 1 & 477.65 & -3.1002 \\ 2.2220 & 16.710 & 9.6120 \\ -1.5611 & 5.1792 & -1.6855 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 239.23 \\ 0.96500 \\ 2.7140 \end{pmatrix}$$

Step 2: $a_2 = a_2 - 2.222a_1$

$$\begin{pmatrix} 1 & 477.65 & -3.1002 \\ 0 & -1059.7 & 16.501 \\ -1.5611 & 5.1792 & -1.6855 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 239.23 \\ -5314.8 \\ 2.7140 \end{pmatrix}$$

Step 3: $a_3 = a_3 + 1.5611a_1$

$$\begin{pmatrix} 1 & 477.65 & -3.1002 \\ 0 & -1059.7 & 16.501 \\ 0 & 7461.7 & -6.5252 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 239.23 \\ -5314.8 \\ 3737.4 \end{pmatrix}$$

Step 4: $a_2 = \frac{1}{-1059.7}a_2$

$$\begin{pmatrix} 1 & 477.65 & -3.1002 \\ 0 & 1 & -0.00156 \\ 0 & 7461.7 & -6.5252 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 239.23 \\ 0.50155 \\ 3737.4 \end{pmatrix}$$

Step 5: $a_3 = a_3 - 7461.7 * a_2$

$$\begin{pmatrix} 1 & 477.65 & -3.1002 \\ 0 & 1 & -0.00156 \\ 0 & 0 & 5.0939 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 239.23 \\ 0.50155 \\ -5.0938 \end{pmatrix}$$

Step 6: $a_3 = \frac{1}{5.0939}$

$$\begin{pmatrix} 1 & 477.65 & -3.1002 \\ 0 & 1 & -0.00156 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 239.23 \\ 0.50155 \\ -0.99999 \end{pmatrix}$$

Step 7: $a_1 = a_1 - 477.65a_2$

$$\begin{pmatrix} 1 & 0 & -2.3551 \\ 0 & 1 & -0.00156 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -0.33536 \\ 0.50155 \\ -0.99999 \end{pmatrix}$$

Step 8: $a_1 = a_1 + 2.3551a_3$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -0.00156 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0.99997 \\ 0.50155 \\ -0.99999 \end{pmatrix}$$

Step 9: $a_1 = a_1 + 0.00156a_3$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0.99997 \\ 0.499996 \\ -0.99999 \end{pmatrix}$$

I get the answer as state above. Not the exact numbers due to rounding errors.

1.2 b.

Consider the 2x2 system of linear equations.

$$1.13x + 1.54y = 4.21$$
$$1.14x + 1.57y = 4.28$$

Verify that x = 1, y = 2 is the solution.

Solve using Gaussian to 3 significant digits.

$$\begin{pmatrix} 1.13 & 1.54 \\ 1.14 & 1.57 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 4.21 \\ 4.28 \end{pmatrix}$$

$$\begin{pmatrix} 1.13 & 1.54 \\ 0 & 0.0164 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 4.21 \\ 0.0327 \end{pmatrix}$$

$$\begin{pmatrix} 1.13 & 0 \\ 0 & 0.0164 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1.14 \\ 0.0327 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1.01 \\ 1.99 \end{pmatrix}$$

The result of the elimination is the answer given above.

2 Problem 2. Gauss in Matlab

When running the script I get a warning for the last vector. The outputted results are similar to the previous ones when using pivoting, but are very small when using naive Gaussian elimination. It is likely that the naive elimination is creating a large loss of significance.

Script:

```
% Construct A and b in the following way

% Case 1
c = [0.2:0.2:1];
A = vander(c);
xsol = ones(size(c'));
b = A*xsol;
disp('Guassian:')
naiv_gauss(A,b)
disp('Pivoting:')
A\b

% Case 2
c = [0.1:0.1:1];
A = vander(c);
xsol = ones(size(c'));
b = A*xsol;
disp('Guassian:')
naiv_gauss(A,b)
disp('Pivoting:')
A\b


% Case 3
c = [0.05:0.05:1];
```

```
A = vander(c);
xsol = ones(size(c'));
b = A*xsol;
disp('Guassian:')
naiv_gauss(A,b)
disp('Pivoting:')
A\b
```

Output:

Problem2Script Guassian:

ans =

- 0.999999999999612
- 1.000000000000889
- 0.999999999999494
- 1.000000000000104
- 0.999999999999993

Pivoting:

ans =

- 1.000000000000032
- 0.999999999999921
- 1.000000000000070
- 0.999999999999974
- 1.000000000000003

Guassian:

ans =

- 0.998774840610394
- 1.005853289548497
- 0.988068244628706
- 1.013568750263305
- 0.990551321010233
- 1.004157217864660
- 0.998851916215852
- 1.000190204544961
- 0.999983061158011
- 1.000000606254654

Pivoting:

ans =

- 1.000000000012049
- 0.999999999936912
- 1.000000000142096
- 0.999999999820241
- 1.000000000139617

- 0.999999999931755
- 1.000000000020688
- 0.999999999996307
- 1.000000000000348
- 0.999999999999987

Guassian:

ans =

1.0e+15 *

- 0.100260367617011
- -0.547611736692488
- 1.322086302655506
- -1.861303854966565
- 1.697314314649383
- -1.052455338465164
- 0.452613654373408
- -0.134745815870952
- 0.027010370505822
- -0.003364454050715
- 0.000193583684468
- 0.000007778825174
- -0.000002060199317
- 0.000000130012968
- -0.000000003878228
- 0.000000000134212
- -0.000000000006845
- 0.000000000000244
- -0.000000000000009
- 0.000000000000001

Pivoting:

Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND = 5.884995e-18.

In Problem2Script (line 33)

ans =

- 1.001948529862383
- 0.980676737577058
- 1.089164834273452
- 0.745765429700010
- 1.501655183245481
- 0.272979675952538
- 1.801520667803391
- 0.313048153146008
- 1.463758890491371
- 0.751644013870925
- 1.105732817783706
- 0.964296230726580
- 1.009498367655909
- 0.998032829037259
- 1.000311369627848

- 0.999963363162325
- 1.000003071985154
- 0.999999828557529
- 1.000000005621326
- 0.999999999919745

3 Problem 3. Application of system of linear equations

Script:

```
% alpha value
alpha = cos(pi/4);

% create and populate matrix
A = sparse(21, 21);

A(1,1) = -alpha;
A(1,4) = 1;
A(1,5) = alpha;

A(2,1) = alpha;
A(2,3) = 1;
A(2,5) = alpha;

A(3,2) = -1;
A(3,6) = 1;

A(4,3) = -1;
A(4,18) = 1;

A(5,4) = -1;
A(5,8) = 1;

A(6,7) = 1;

A(7,5) = -alpha;
A(7,6) = -1;
A(7,9) = alpha;
A(7,10) = 1;

A(8,5) = -alpha;
A(8,7) = -1;
A(8,9) = -alpha;
A(8,19) = 1;

A(9,8) = -1;
A(9,9) = -alpha;
A(9,12) = 1;
A(9,13) = alpha;

A(10,9) = alpha;
A(10,11) = 1;
```

```
A(10,13) = alpha;

A(11,10) = -1;
A(11,14) = 1;

A(12,11) = -1;
A(12,20) = 1;

A(13,13) = -alpha;
A(13,14) = -1;
A(13,17) = 1;

A(14,13) = -alpha;
A(14,15) = -1;
A(14,21) = 1;

A(15,12) = -1;
A(15,16) = alpha;

A(16,15) = 1;
A(16,16) = alpha;

A(17,16) = alpha;
A(17,17) = 1;

% Given forces
A(18,18) = 1;
A(19,19) = 1;
A(20,20) = 1;
A(21,21) = 1;


% Loop over four situations
disp('Forces from f1-f17 then F1-F4.')

% four forces
force =[10, 15, 0, 10;
        15, 0, 0, 10;
        10, 0, 20, 0;
        0, 10, 10, 0];
for i = 1:4
    disp(strcat("Test Case: ",num2str(i)))
    forces = force(i,:);
    b = zeros(1,17);
    b = [b forces];
    b = b';
    disp(A\b)
end
```

Output:

Problem3Script

Forces from f1-f17 then F1-F4.

Test Case: 1

- -26.870057685088803
- 19.000000000000000
- 10.000000000000000
- -28.000000000000000
- 12.727922061357853
- 19.000000000000000
- 0
- -28.000000000000000
- 8.485281374238570
- 22.000000000000000
- 0
- -15.999999999999996
- -8.485281374238570
- 22.000000000000000
- 16.000000000000000
- -22.627416997969519
- 16.000000000000000
- 10.000000000000000
- 15.000000000000000
- 0
- 10.000000000000000

Test Case: 2

- -19.798989873223331
- 14.000000000000000
- 15.000000000000000
- -13.000000000000000
- -1.414213562373095
- 14.000000000000000
- 0
- -13.000000000000000
- 1.414213562373095
- 11.999999999999998
- 0
- -10.999999999999998
- -1.414213562373095
- 11.999999999999998
- 10.999999999999998
- -15.556349186104043
- 10.999999999999998
- 15.000000000000000
- 0
- 0
- 10.000000000000000

Test Case: 3

- -22.627416997969522
- 16.000000000000000
- 10.000000000000000
- -22.000000000000000
- 8.485281374238571

- 16.000000000000000
- 0
- -22.000000000000000
- -8.485281374238570
- 28.000000000000000
- 20.000000000000000
- -14.000000000000000
- -19.798989873223331
- 28.000000000000000
- 14.000000000000000
- -19.798989873223331
- 14.000000000000000
- 10.000000000000000
- 0
- 20.000000000000000
- 0

Test Case: 4

- -14.142135623730953
- 10.000000000000000
- 0
- -20.000000000000000
- 14.142135623730951
- 10.000000000000000
- 0
- -20.000000000000004
- 0.000000000000001
- 20.000000000000000
- 10.000000000000000
- -10.000000000000000
- -14.142135623730949
- 20.000000000000000
- 10.000000000000000
- -14.142135623730949
- 10.000000000000000
- 0
- 10.000000000000000
- 10.000000000000000
- 0

The fact that I get negative forces in my output isn't necessarily a problem, it just means that they were going the opposite direction to which I expected them to be going.