Project One Template

MAT350: Applied Linear Algebra

Student Name: Edward Garcia

Date: 1/25/2024

Problem 1

Develop a system of linear equations for the network by writing an equation for each router (A, B, C, D, and E). Make sure to write your final answer as A**x**=**b** where A is the 5x5 coefficient matrix, **x** is the 5x1 vector of unknowns, and **b** is a 5x1 vector of constants.

Solution:

I was able to determine the linear equations by analyzing the data flow in the network diagram. After analyzing Router A I can see that it is receiving data from the sender as 100 Mbps, it is sending data to Router B as x1, it is sending data to Router C as x2, and it is sending data to Router E as x2, so the equation for Router A = 2x2 + x1 = 100. Looking at Router B in the diagram I can see that it is receiving data from Router A as x1, it is receiving data from Router D as x2, it is sending data to the receiver as x3, and it is sending data to Router E as x5 so the equation for Router B = x1 + x2 = x3 + x5. Analyzing Router C in the diagram I can verify that it is receiving data from Router A as x2, it is receiving 50 Mbps from the sender, it is also sending data to Router E as x3, and it is sending data to Router D as x5. The equation for Router C is 50 + x2 = x3 + x5. After looking at Router D I have verified that it is receiving data from Router E as x4, it is receiving data from Router C as x5, it is sending data to Router B as x2, and it is sending data to the receiver as 120 Mbps, the equation for Router D = x4 + x5 = x2 + 120. After analyzing Router E from the diagram I can see that it is receiving data from Router A as x2, it is receiving data from Router C as x3, it is receiving data from Router B as x5, and it is sending data to Router D as x4, the equation for Router E = x2 + x3 = x5 + x4. I changed the equations so that all the x terms are on the left side and the constant terms are on the right side for Ax = b form.

Linear Equations:

Router A: $2x_2 + x_1 = 100$

= $x_1 + 2\,x_2 + 0x_3 + 0x_4 + 0x_5 = 100$

Router B: $x_3 + x_5$ = $x_1 + x_2$

$= x_1 + x_2 - x_3 - x_5 = 0$

Router C: $50 + x_2 = x_3 + x_5$

$0x_1 + x_2 - x_3 + 0x_4 + x_5 = 50$

Router D: $x_4 + x_5 = x_2 + 120$

$= 0x_1 - x_2 + 0x_3 + x_4 + x_5 = -120$

Router E: $x_2 + x_3 = x_5 + x_4$

$= 0x_1 + x_2 + x_3 - x_4 - x_5 = 0$

Final Answer: A = 5 x 5 coefficient matrix, x = 5 x 1 vector of unknowns, b = 5 x 1 vector of constants.

$$A = \begin{bmatrix} 1 & 2 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 & 1 \\ 0 & -1 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 & -1 \\ 0 & -1 & -1 & 1 & -1 \end{bmatrix}, \text{ x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}, \text{ b} = \begin{bmatrix} 100 \\ 0 \\ 50 \\ -120 \\ 0 \end{bmatrix}$$

Problem 2

Use MATLAB to construct the augmented matrix [A **b**] and then perform row reduction using the rref() function. Write out your **reduced matrix and identify the free and basic variables of the system**.

Solution:

```
%code
% Create the matrices A and b
% Matrix A represents the coefficients of the variables in the system of
equations
% Vector b represents the constants on the right side of the equations
A = [1 2 0 0 0; -1 -1 1 0 1; 0 -1 1 0 1; 0 1 0 -1 -1; 0 -1 -1 1 -1];
b = [100; 0; 50; -120; 0];

% Create the augmented matrix [A b]
% The augmented matrix is used to combine the coefficient matrix A
% and the constants vector b into one matrix
augmented_matrix = [A b];

% Perform row reduction to get the RREF
% This function returns pivot_columns which indicates the indices of the
% pivot columns in A
[reduced_matrix, pivot_columns] = rref(augmented_matrix);

% Display the RREF of the matrix
% This will show the final simplified form of the augmented matrix which can
% be used to interpret the solutions to the system of equations
disp(reduced_matrix);
```

```
     1     0     0     0     0    50
     0     1     0     0     0    25
     0     0     1     0     0    30
```

```
     0     0     0     1     0   100
     0     0     0     0     1    45
```

```
% Identify the basic variables

% The basic variables are the variables that are linked to
% pivots which are the first non-zero entry.
% The basic variables are x1, x2, x3, x4, x5.
basic_vars = pivot_columns;

% Identify the free variables

% Free variables do not link to the pivots to a leading 1
% in the (rref).
% Since all variables are basic, there are no free variables
all_vars = 1:size(A, 2);
free_vars = setdiff(all_vars, basic_vars);

% Display the basic and free variables
disp('Basic Variables:');
```

```
Basic Variables:
```

```
disp(basic_vars);
```

```
     1     2     3     4     5
```

```
disp('Free Variables:');
```

```
Free Variables:
```

```
disp(free_vars);
```

Problem 3

Use MATLAB to **compute the LU decomposition of A**, i.e., find A = LU. For this decomposition, find the transformed set of equations L**y** = **b**, where **y** = U**x**. Solve the system of equations L**y** = **b** for the unknown vector **y**.

Solution:

```
%code
% The MATLAB backslash operator is used to solve systems of
% linear equations of the form Ax = b,
% where A is a matrix and b is a vector.

% Create the matrices A and b
% Matrix A represents the coefficients of the variables in the system of
equations.
```

```
% Vector b represents the constants on the right side of the equations.
% These matrices will be used in the LU decomposition to solve the system.
A = [1 2 0 0 0; -1 -1 1 0 1; 0 -1 1 0 1; 0 1 0 -1 -1; 0 -1 -1 1 -1];
b = [100; 0; 50; -120; 0];

% Compute the LU decomposition of A
% LU decomposition factors matrix A into L and U,
% where L is lower triangular and U is upper triangular.
[L, U] = lu(A);

% Solve Ly = b for y using forward substitution
% Forward substitution is an algorithm that can solve a system of equations
% when the coefficient matrix is lower triangular.
y = L\b;

% Display the output of Matrix L, Matrix U and Vector y
disp('Matrix L:');
```

```
Matrix L:
```

```
disp(L);
```

```
    1.0000         0         0         0         0
   -1.0000    1.0000         0         0         0
         0   -1.0000    1.0000         0         0
         0    1.0000   -0.5000    1.0000         0
         0   -1.0000         0   -1.0000    1.0000
```

```
disp('Matrix U:');
```

```
Matrix U:
```

```
disp(U);
```

```
     1     2     0     0     0
     0     1     1     0     1
     0     0     2     0     2
     0     0     0    -1    -1
     0     0     0     0    -1
```

```
disp('vector y:');
```

```
vector y:
```

```
disp(y);
```

```
   100
   100
   150
  -145
   -45
```

Problem 4

Use MATLAB to **compute the inverse** of U using the inv() function.

Solution:

```
%code
% The inv function in MATLAB computes the inverse of a square matrix if it
exists.
% U is the upper triangular matrix obtained from the LU decomposition of
matrix A.
% Compute the inverse of U
U_inv = inv(U);

% Display the inverse of U
disp('Inverse of U:');
```

```
Inverse of U:
```

```
disp(U_inv);
```

```
    1.0000   -2.0000    1.0000         0         0
         0    1.0000   -0.5000         0         0
         0         0    0.5000         0    1.0000
         0         0         0   -1.0000    1.0000
         0         0         0         0   -1.0000
```

Problem 5

Compute the solution to the original system of equations by transforming **y** into **x**, i.e., compute **x** = inv(U)**y**.

Solution:

```
%code
% Compute x using inv(U)y
% Solve Ux = y for x
X = inv(U)*y
```

```
X = 5×1
    50
    25
    30
   100
    45
```

```
% Here, we are computing x using the system Ux = y directly utilizing the
backslash
% operator
x = U\y;

% Display and output x
disp(x);
```

```
    50
    25
    30
   100
    45
```

Problem 6

Check your answer for x_1 using Cramer's Rule. Use MATLAB to compute the required determinants using the det() function.

Solution:

```
%code
% Create the matrices A and b
% Matrix A represents the coefficients of the variables in the system of
equations.
% Vector b represents the constants on the right side of the equations.
A = [1 2 0 0 0; -1 -1 1 0 1; 0 -1 1 0 1; 0 1 0 -1 -1; 0 -1 -1 1 -1];
b = [100; 0; 50; -120; 0];

% Compute determinant of A
% The determinant of matrix A is calculated because Cramer's Rule involves
the use of determinants.
% The determinant must be non-zero for Cramer's Rule to be valid.
det_A = det(A);

% Initialize solution vector
% A vector of zeros with the same size as vector b is created to store the
solutions of the system.
x_cramer = zeros(size(b));

% Compute solution using Cramer's Rule
% Cramer's Rule solves for each variable of the system individually
% by replacing the corresponding column of A with vector b,
% and then computing the determinant of the modified matrix.
for i = 1:length(b)
    % Create modified matrix A_i
    A_i = A;
    A_i(:,i) = b;
    % Compute determinant of A_i
    det_A_i = det(A_i);
    % Compute i-th variable of x
    x_cramer(i) = det_A_i / det_A;
end

% Display the full vector x computed by Cramer's Rule
% The solution vector x_cramer is displayed to provide
% the solution to the system of equations.
disp('Vector x using Cramer’s Rule is:');
```

```
Vector x using Cramer’s Rule is:
```

```
disp(x_cramer);
```

```
   50.0000
   25.0000
   30.0000
  100.0000
   45.0000
```

```

% Display x1 answer to verify it matches x1 value
disp('The value of x1 is:');
```

```
The value of x1 is:
```

```
disp(x_cramer(1));
```

```
    50
```

```

```

Problem 7

The Project One Table Template, provided in the Project One Supporting Materials section in Brightspace, shows the recommended throughput capacity of each link in the network. Put your solution for the system of equations in the third column so it can be easily compared to the maximum capacity in the second column. In the fourth column of the table, provide recommendations for how the network should be modified based on your network throughput analysis findings. The modification options can be No Change, Remove Link, or Upgrade Link. In the final column, explain how you arrived at your recommendation.

Solution:

Fill out the table in the original project document and export your table as an image. Then, use the ***Insert*** *tab in the MATLAB editor to insert your table as an image.*

Southern New Hampshire University

MAT 350 Project One Table Template

Complete this template by replacing the bracketed text with the relevant information.

Network Link	Recommended Capacity (Mbps)	Solution	Recommendation	Explanation
x_1	60	[50]	[No Change]	[The link is operating within bounds and does not require an upgrade.]
x_2	50	[25]	[No Change]	[The link is operating within bounds and does not require an upgrade.]
x_3	100	[30]	[No Change]	[The link is operating within bounds and does not require an upgrade.]
x_4	100	[100]	[Upgrade Link]	[The capacity (Mbps) is at the maximum recommended capacity, so I recommend upgrading the link]
x_5	50	[45]	[No Change]	[The link is operating within bounds and does not require an upgrade