EE 430 HW #2

October 24, 2025

Joshua Consenz

Problem 1

$N_{Bus} = 36$
$N_{T\text{-}lines} = 43$

Non-zero elements = $N_{bus} + 2 * N_{T-lines} = 36 + 2 * 43 = 122$

$Size(Y_{bus}) = 36 * 36 = 1296$

$\%\ non - zero\ elements = \left(\frac{non-zero\ elements}{Size(Y_{bus})}\right) \cdot 100\% = \left(\frac{122}{1296}\right) \cdot 100\% = 9.42\%$

Table of Contents

EE 430 HW 2

Joshua Consenz - 10/24/25

```
clc; clear; close all;
```

Problem 4

For an NxN matrix, calculate the vectors DIAG, OFFDIAG, COL, AND ROW

```
% Test matrix and outputs to verify correct functionality
A = [[0 0 7]' [2 5 8]' [3 6 9]'];
DIAG(A)
OFF_DIAG(A)
COL(A)
ROW(A)
```

DIAG function

Function takes an NxN matrix as it's input, and returns a vector of the diagonal elements of the input matrix. DIAG = [values of each diag element]

```
function diag = DIAG(a)

[row, col] = size(a);
diag = zeros(1,row); % creates zero matrix to add elements to

% Loops to check each element of the matrix
for i = 1:row
    for j = 1:col
        if i == j
            % if the matrix indeces are the diagonal elements, then the
            % element is added to the diag output vector
            diag(i) = a(i,j);
        end
    end
end
end
```

```
ans =
```

```
     0     5     9
```

OFF_DIAG function

Function takes an NxN matrix as it's input, and returns a vector of the off diagonal elements OFF-DIAG = [values of each off-diag element]

```
function off_diag = OFF_DIAG(a)

[row, col] = size(a);
num = row*col-row; % calculates the number of non-diagonal elements
off_diag = zeros(1,num);

k=0; % vector position interator
% Loops for evaluating each element of the input matrix
for i = 1:row
    for j = 1:col
        if i ~= j
            % if the matrix indeces are not diagonal then the vector
            % iterator is incremented and the element is added to the
            % output matrix at the position the iterator is indicating.
            k = k+1;
            off_diag(k) = a(i,j);
        end
    end
end
end
```

```
ans =

     2     3     0     6     7     8
```

COL function

Function takes an NxN matrix as it's input, and returns a vector of the column index for each off-diagonal matrix element COL = [column index of each off diag element]

```
function col = COL(a)
    [row, column] = size(a);

    col = zeros(1,column);
    k = 0; % vector position interator

    % loops for assessing each element of the input matrix
    for i = 1:row
        for j = 1:column
            if i ~= j
                % if the indeces of the input matrix refer to an
                % off-diagonal element, then the position iterator is
```

```
                % incremented, and the column index, j, is added to the
                % output vector at position k.
                k = k +1;
                col(k) = j;
            end
        end
    end
end
```

```
ans =

     2     3     1     3     1     2
```

ROW function

Function takes an NxN matrix as it's input, and returns a vector indicating the number of off-diagonal, non-zero elements in each row ROW = [number of each off-diag elements in each row]

```
function row = ROW(a)

[rows, col] = size(a);
row = zeros(1,rows);

% loops for iterating through the entire input matrix
for i = 1:rows
    for j = 1:col
        if i ~= j && a(i,j) ~=0
            % if the loop iterators point to an off-diagonal position, and
            % the element at that position is non-zero, then the row at
            % that position gets +1 added to the outut count.
            row(i) = row(i) + 1;
        end
    end
end
end
```

```
ans =

     2     1     2
```

Published with MATLAB® R2025a