**1. (a)**

***The code is shown as below:***

% create a master matrix to store top line intensites for each n

t = zeros(100, 4);

% create an index to record the column number of A

column\_index = 1;

% create a vector to store the numbers of n

ni = [4, 8, 16, 32];

% create a vector to store the condition number of each n

condition\_number = zeros(4,1);

% create a vector to store the minimum intensity of each n

min\_intensity = zeros(4,1);

% create a vector to store the maximum intensity of each n

max\_intensity = zeros(4,1);

for n = ni

%for n = 8

% set the number of equations (N) based on the number of loops (n)

N = 3 \* n - 2;

% create a sparse matrix based on the value of N

A = speye(N, N);

% since the left-most loop, the right-most loop and the right-most bottom

% node do not follow the pattern, we set the rows correspoinds to these

% euqtaions in the sparse martix first.

A(1, 1:3) = [1 1 1]; % the left-most loop

A(N-1, N-2:N) = [1 -1 1]; % the right-most loop

A(N, N-2:N) = [-1 0 2]; % the right-most bottom node

% fill rows in the sparse matrix for each top node.

for k = 2: 3: N

A(k, k-1:k+2) = [1 -1 0 -1];

end

% fill rows in the sparse matrix for each bottom node.

for k = 3: 3: N-2

A(k, k-1:k+3) = [1 -1 0 0 1];

end

% fill rows in the sparse matrix for each loop

for k = 4: 3: N-3

A(k, k-2:k+2) = [-1 0 1 1 1];

end

% set the vector b

b = zeros(N, 1);

% change the first element of vector b to the voltage of source

b(1) = 100;

% calculate the intensities

x = A\b;

% record the maximum intensity, minimum intensity and condition number

min\_intensity(column\_index) = min(x);

max\_intensity(column\_index) = max(x);

condition\_number(column\_index) = condest(A);

% get the index of top line intensities

top\_line\_index = 1:3:N;

% replace the column of the matrix by the calculated intensities

t(1:n, column\_index) = x(top\_line\_index);

% after each n loop, the coumn\_index grows by 1

column\_index = column\_index + 1;

% record the matrix A for later on LU factorization

if n == 8

A8 = A;

end

end

% plot the top line intensities versus their normalized index in one plot

plot([1:ni(1)]/ni(1), t(1:ni(1), 1), 'r-', ...

[1:ni(2)]/ni(2), t(1:ni(2), 2), 'g--', ...

[1:ni(3)]/ni(3), t(1:ni(3), 3), 'b-.', ...

[1:ni(4)]/ni(4), t(1:ni(4), 4), 'k-.');

xlabel('normalized index') % x-axis label

ylabel('top line intensities') % y-axis label

legend('n = 4','n = 8', 'n = 16', 'n = 32') % add legend

% print out the outputmatirx to a .txt file.

fileID = fopen('outputAssignment2Q1.txt', 'w');

fprintf(fileID, '%12s %18s %18s %18s %18s\n', ' ','n = 4', 'n = 8', 'n = 16', 'n = 32');

fprintf(fileID, '%12s %18.8f %18.8f %18.8f %18.8f\n', 'maximum intensity', max\_intensity(1), max\_intensity(2), max\_intensity(3), max\_intensity(4));

fprintf(fileID, '%12s %18.8f %18.8f %18.8f %18.8f\n', 'minimum intensity', min\_intensity(1), min\_intensity(2), min\_intensity(3), min\_intensity(4));

fprintf(fileID, '%12s %18.8f %18.8f %18.8f %18.8f\n', 'condition number', condition\_number(1), condition\_number(2), condition\_number(3), condition\_number(4));

fclose(fileID);

% LU factorization for A, when n = 8

[L, U, P] = lu(A8);

spy(A8);

spy(L);

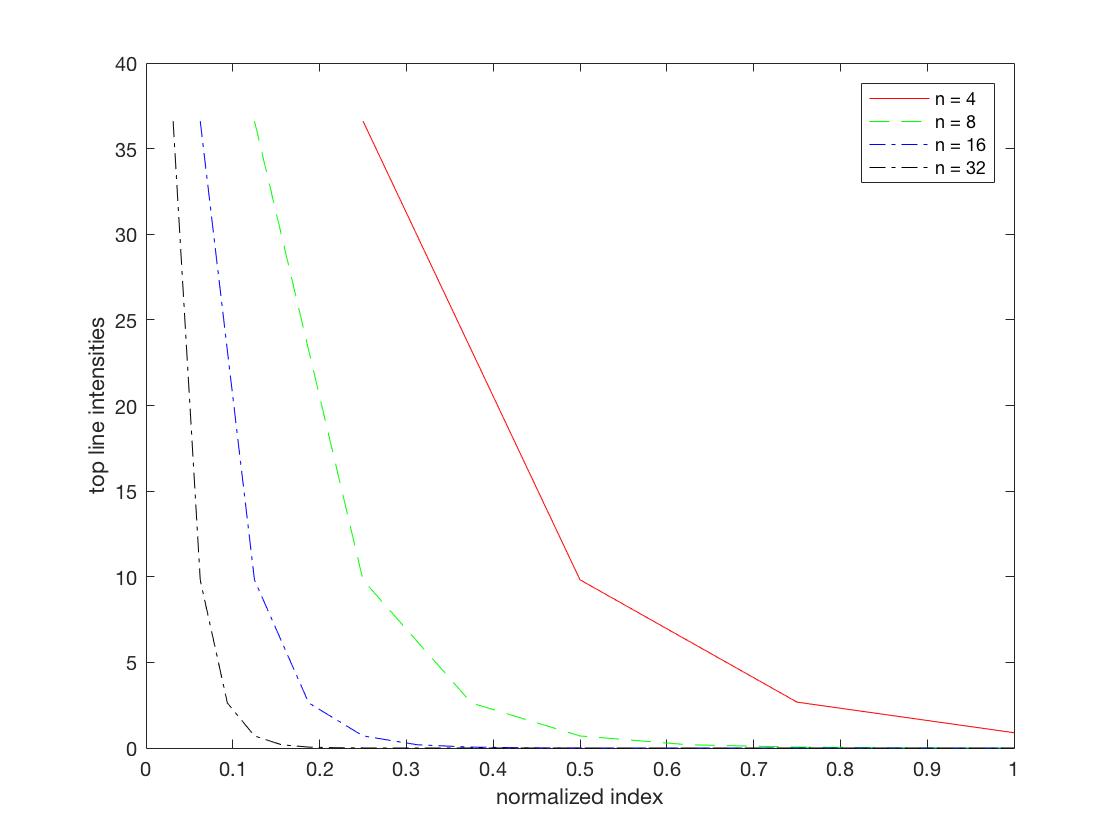
spy(U);

spy(P);

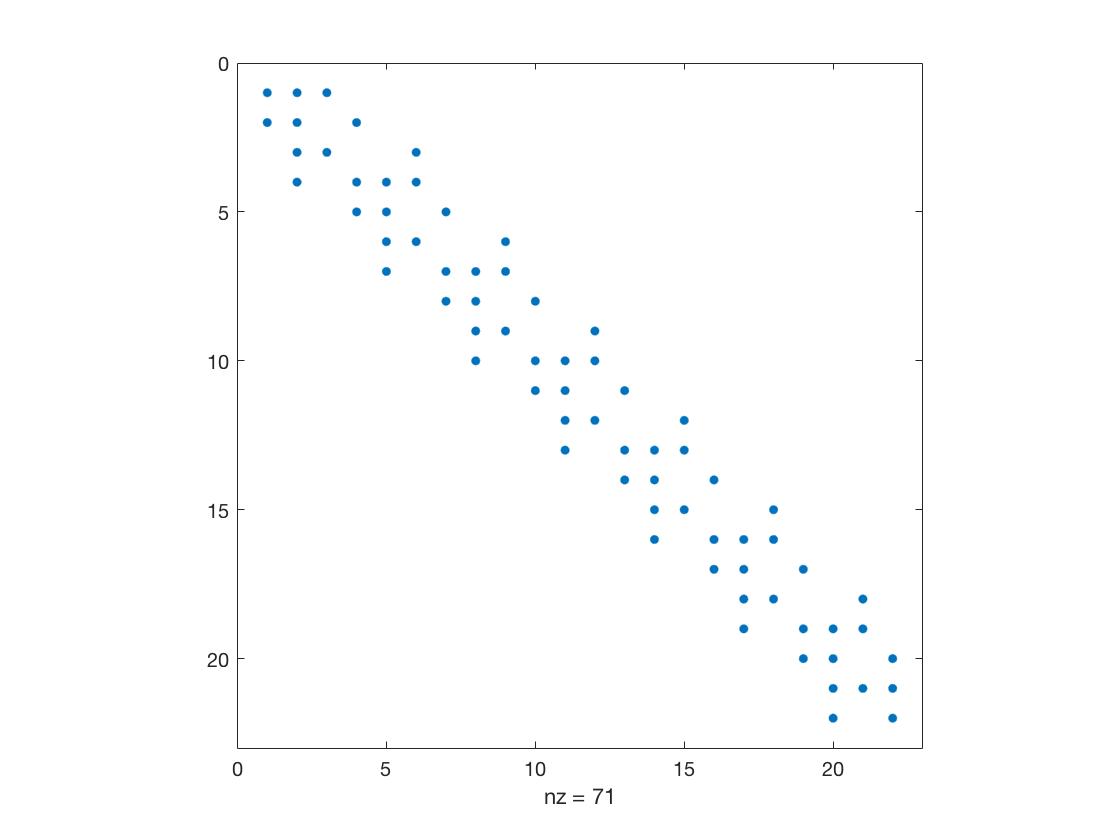
***The outputs for maximum and minimum intensities as well as condition number are shown as below:***

Macintosh HD:Users:lucasminghu:Desktop:Screen Shot 2017-03-12 at 12.22.24 AM.png

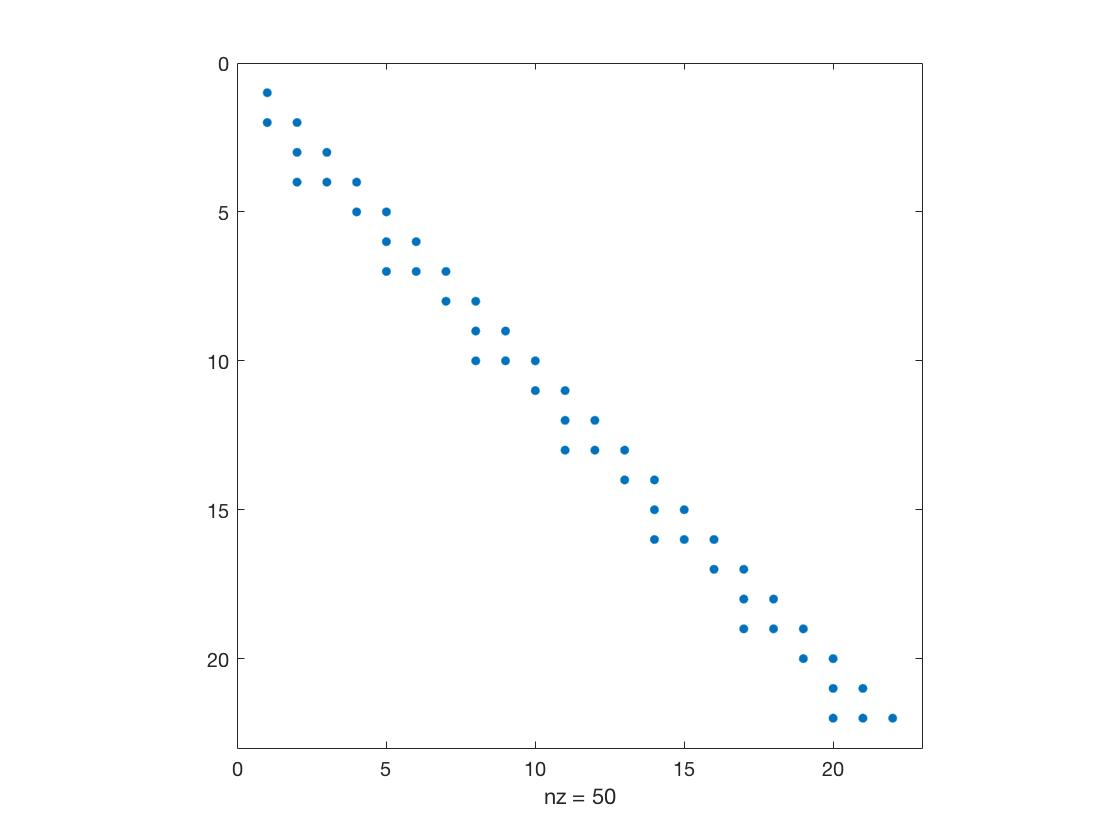
***The plot of top line intensities versus their normalized is shown as below:***



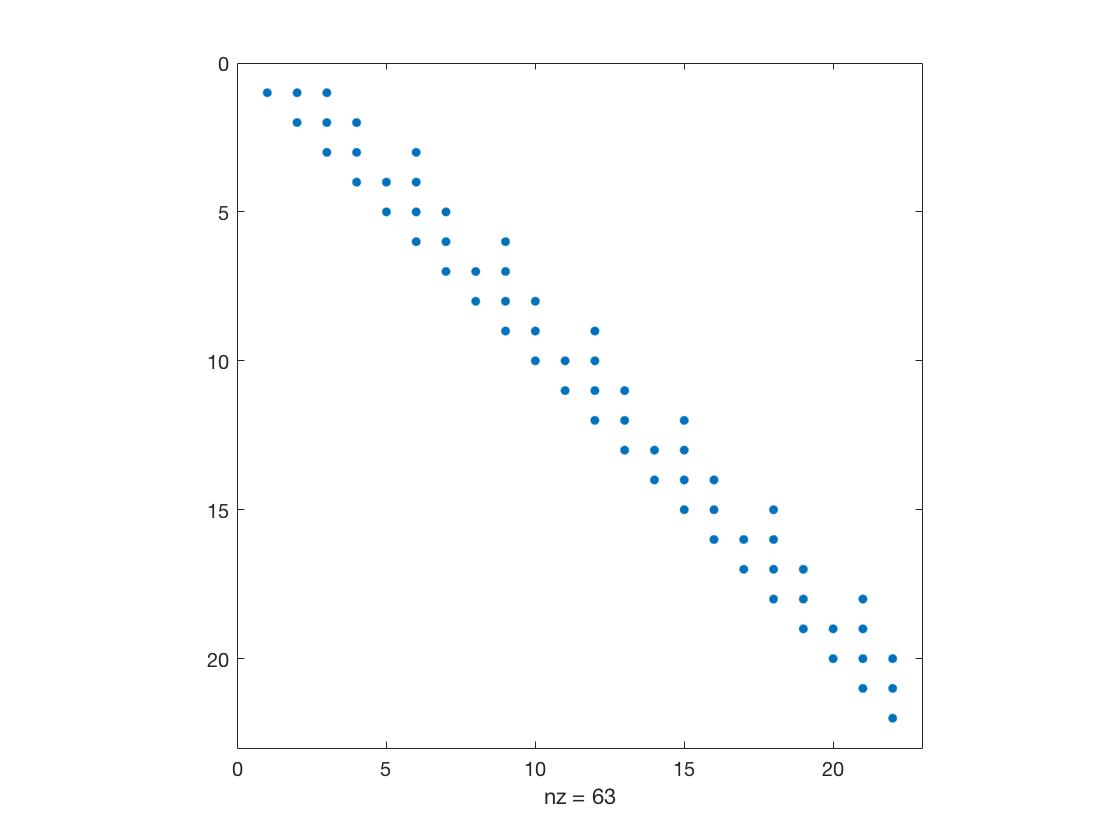
***The sparsity pattern of A when n = 8:***

******

***The sparsity pattern of L when n = 8:***



***The sparsity pattern of U when n = 8:***

******