ECE1254 Assignment 3

Xiao Han Dong
998407834

March 3, 2016

1 part a

For a circuit with only resistors and current sources, the circuit matrix G is constructed by summing up all the resistor stamps. Generally, for a resistor R connecting the two nodes i, j, this is essentially the matrix $R_{i,j}$of zeros except the entries $R_{i,i} = \frac{1}{R} = R_{j,j}, R_{i,j} = -\frac{1}{R} = R_{j,i}$ unless one node is grounded, in which case only the un-grounded i^{th} node element $R_{i,i} = \frac{1}{R}$ is in the $R_{i,j}$ matrix. Clearly, every resistor matrix $R_{i,j}$ is symmetric.

Since

$$G = \sum R_{i,j}$$

summation does not break symmetry; therefore the sum of symmetric matrices is also symmetric.

A positive definite matrix is a matrix G that satisfies $x^T Gx \geq 0$. This means the eigenvalues should be non-negative. We have also seen in class that the nodal matrix is diagonally dominant ($|G_{i,i}| \geq \sum |G_{i,j}|$). Gershgorin's Theorem guarantees that the eigenvalues of G lies within a circle with radius

$$R = \sum_{i \neq j} |G_{i,j}|$$

centered at $G_{i,i}$, but $R \leq G_{i,i}$, so λ_{min} will never be less than 0. Therefore, the matrix is positive semi-definite.

2 part b, c

Condition number of the circuit matrix is 43221, which is too large to expect good convergence for conjugate gradient solver. Also, due to voltage sources, the matrix is not symmetric, although all eigenvalues appear to be positive. Therefore, the result is no longer guaranteed to converge, as can be seen from figure 1.

3 part d

Referring to what we found previously, the circuit element breaking symmetry of the circuit is the voltage source. Resistors stamp the matrix symmetrically, while

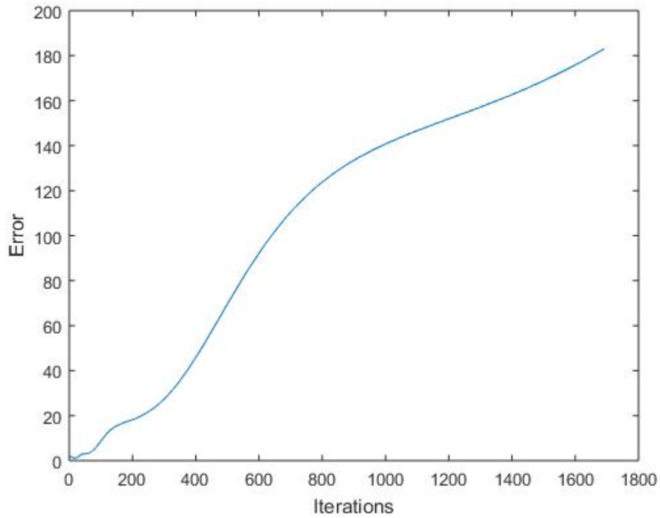

Figure 1: Conjugate gradient diverged for a circuit with a voltage source

current sources affect the b vector only. We can apply source transformation to change the voltage source with series source resistance R_s to a current source with source resistance in parallel with the current source. The current source value is $I_s = \frac{V_s}{R_s}$. This implementation does converge, as can be seen from figure 2

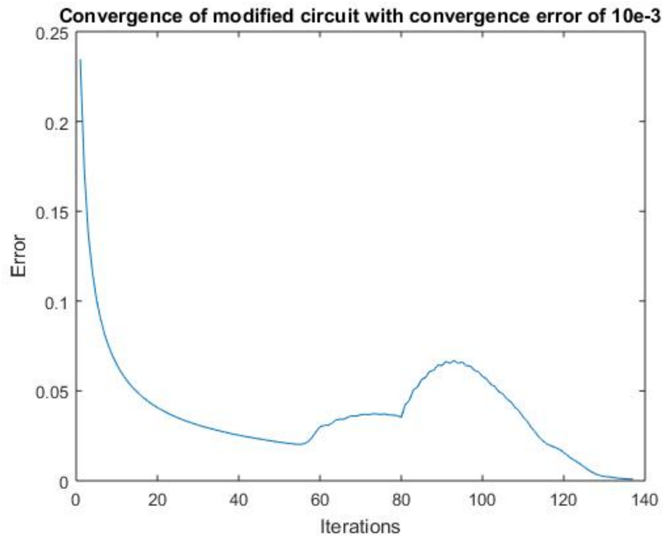

Figure 2: Conjugate gradient converged in 140 iterations with a current source equivalent that replaced voltage source

4 part e

The 2D grid with the source transformation is solved using LU decomposition and Conjugate gradient both with and without pre-conditioning. The results are plotted in figures 3 and 4.

The precondtioning matrix P is taken as a matrix containing only the diagonal of the circuit matrix G and the diagonals one above and one below it. Then, the conditioned problem is

$$P^{-\frac{1}{2}}GP^{-\frac{1}{2}}y = P^{-\frac{1}{2}}b$$

where $y = P^{\frac{1}{2}}x$ and G is the symmetric positive semi-definite matrix.

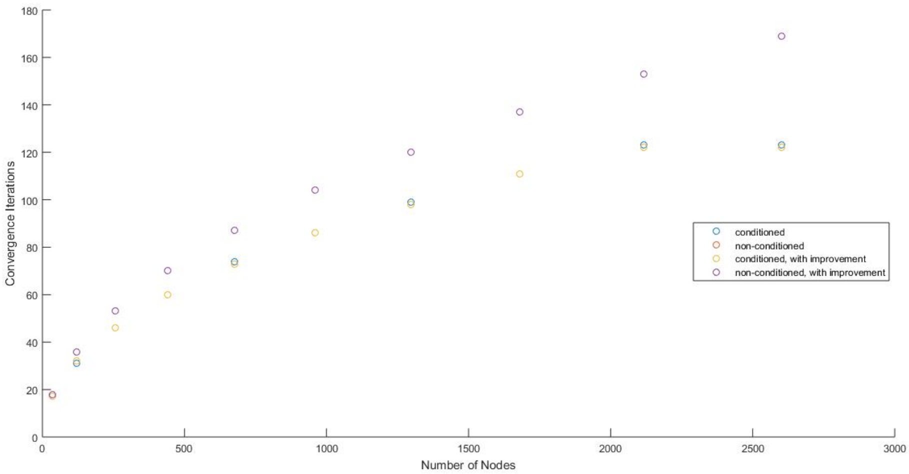

Figure 3: Iterations to reach convergence for the conditioned and non-conditioned cases, both with and without optimizations (see part g)

5 Part f

The timing tests show that iterative methods like conjugate gradient are faster than deterministic methods like LU decomposition. Pre-conditioning reduces the number of iterations to reach convergence, but the overall computation time increases. It seems that although less iterations are required, the extra matrix computations when using pre-conditioning, such as generating the pre-conditioning matrix P and calculating half powers of P are very costly.

6 Part g

To further improve the algorithm, sparse implementations of the circuit matrices are attempted. This requires modifying the code as seen on page 31 of the lecture slides on conjugate gradient method. Taking one half powers of

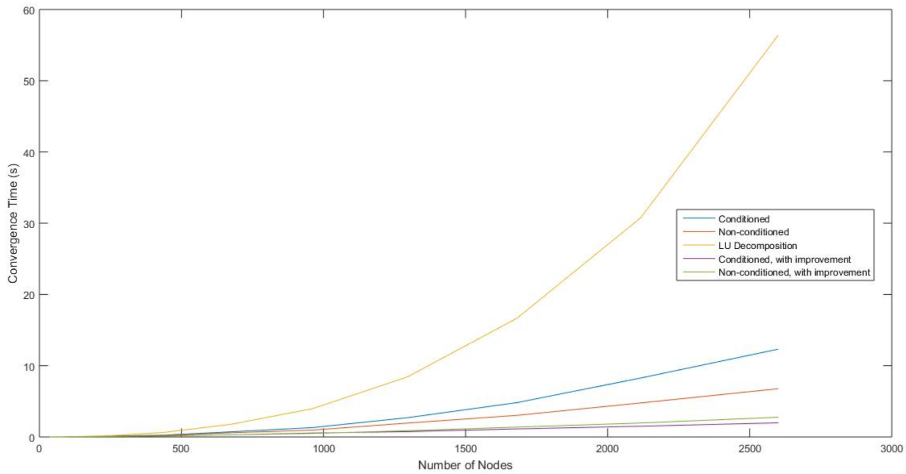

Figure 4: Computation time to reach solution for LU decomposition and conjugate gradient, both conditioned and non-conditioned cases, and with and without optimizations (see part g)

a sparse matrix is difficult and causes MATLAB to throw an error, but this modification results in only needing to compute the inverse of the conditioning matrix, which is done using the back-slash operation in MATLAB. Referring to figures 3 and 4, these improvements had little effect on the convergence iterations, but the convergence time improved several times for both conditioned and non-conditioned. Interestingly, the improvement to the conditioned case is much greater, so that in the improved case it actually runs faster than the non-conditioned matrix. This is probably because the non-conditioned case only benefited from sparsity, while for the conditioned case the improvement is two fold: it removes calculating one half power of a matrix, and it allows for sparse implementations.

7 part h, i

figures 5 to 10 verifies Gershgorin Circle Theorem for both the conditioned and non-conditioned cases: all eigenvalues are inside at least one Gershgorin circle.

We see that the eigenvalues of the circuit matrix is related to the magnitude of the resistances of the grid. As the resistance of the grid increased, the eigenvalues decreased. Most eigenvalues shifted toward the origin, except for an isolated eigenvalue due to the source resistance. Preconditioning seems to apply a correction so that all the Gershgorin circles and the array of eigenvalues centers around $R_{i,i} = 1$; the diagonals of the matrix has all been normalized to one. Normalization also seems to be applied to the radius of the cricles; the size of the eigenvalues and Gershgorin circles of the preconditioned case remained the same while in the non-conditioned case, they decreased in value as resistor values increased.

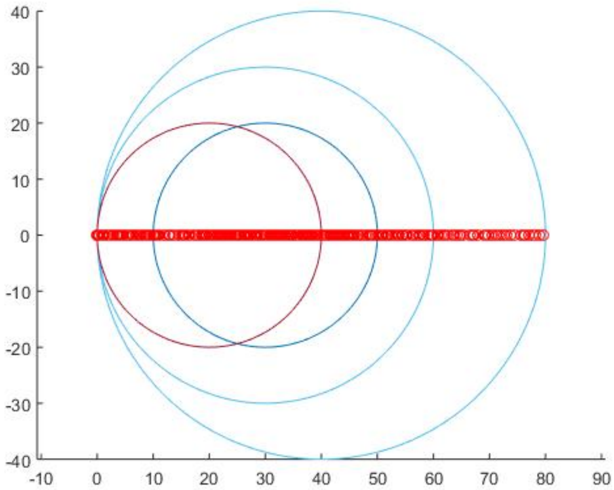

Figure 5: Gershgorin circles for the non-conditioned case with $R = 0.1\Omega$

8 part j

The conditioned case converged in less iterations than the non-conditioned case, as can be seen from figures 11 and 12. This is consistent with our earlier findings. However, the main shape of convergence did not change.

In the previous section, geometrically it was seen that pre-conditioning effectively re-centered the Gershgorin circles around 1, without significantly affecting the behaviour of the actual matrix. By Gershgorin's Theorem, this constrains the eigenvalues to be around 1 also. This is a speed up if the non-conditioned eigenvalues are very small.

Since eigenvalues are a measure of the amount of change applied to vector x by matrix A (since $Ax = \lambda x$), larger eigenvalues allow the system to reach a steady state more quickly, in fewer iterations (assuming the steady state hasn't been changed by the conditioning).

It could also be seen from conditioning number. Since the conditioning number $K = \frac{\lambda_{max}}{\lambda_{min}}$, shifting eigenvalues from near zero to one greatly increases λ_{min}. The new conditioning number $K' = \frac{\lambda_{max}+\delta}{\lambda_{min}+\delta}$. Since values are much more sensitive to changes, the condition number would decrease and improve convergence.

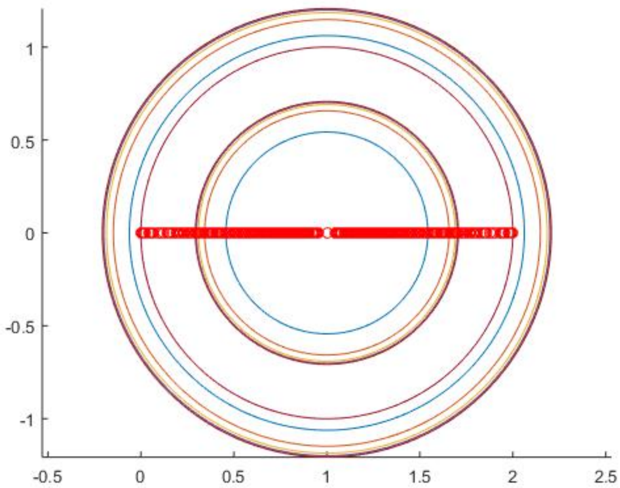

Figure 6: Gershgorin circles for the conditioned case with $R = 0.1\Omega$

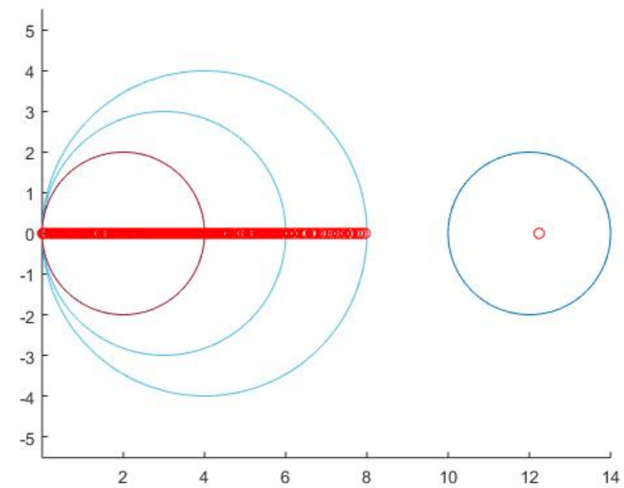

Figure 7: Gershgorin circles for the non-conditioned case with $R = 1\Omega$

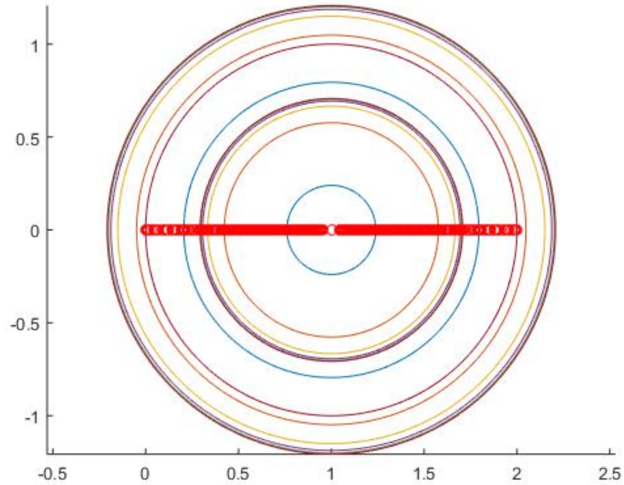

Figure 8: Gershgorin circles for the conditioned case with $R = 1\Omega$

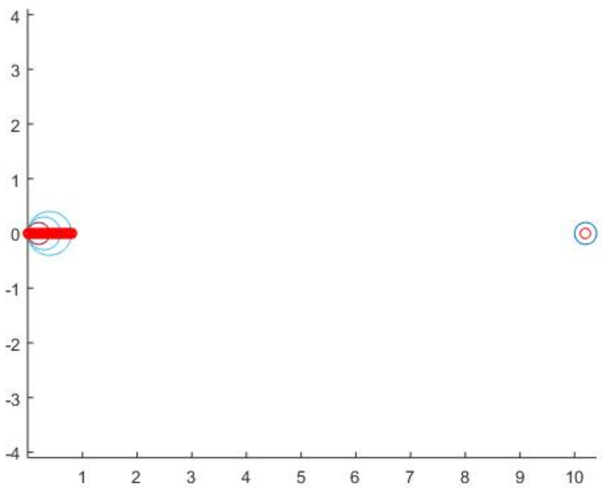

Figure 9: Gershgorin circles for the non-conditioned case with $R = 10\Omega$

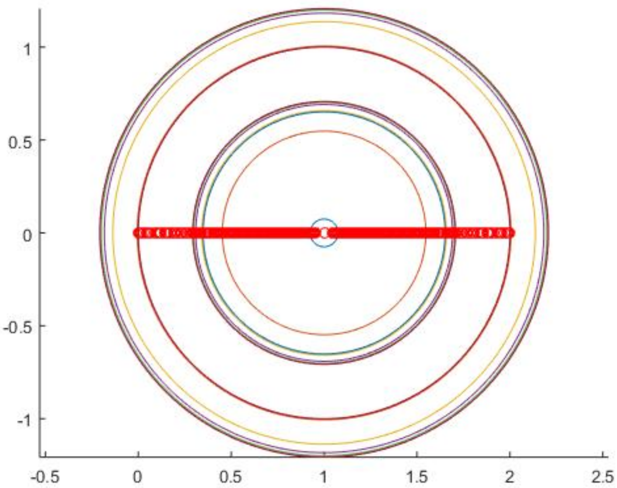

Figure 10: Gershgorin circles for the conditioned case with $R = 10\Omega$

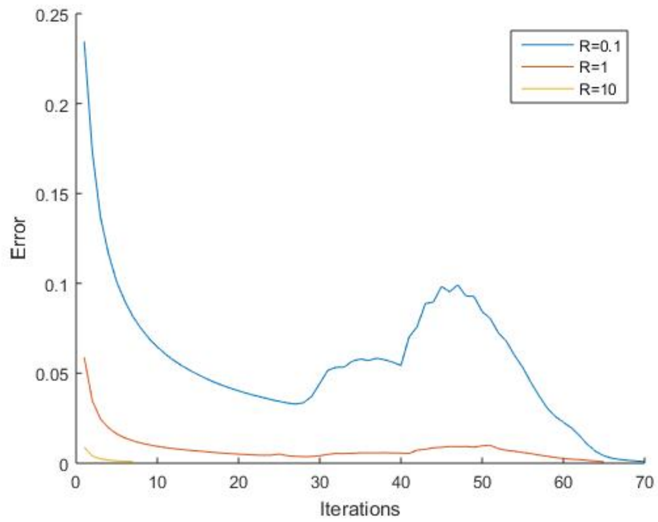

Figure 11: Error convergence for non-conditioned conjugate gradient solution.

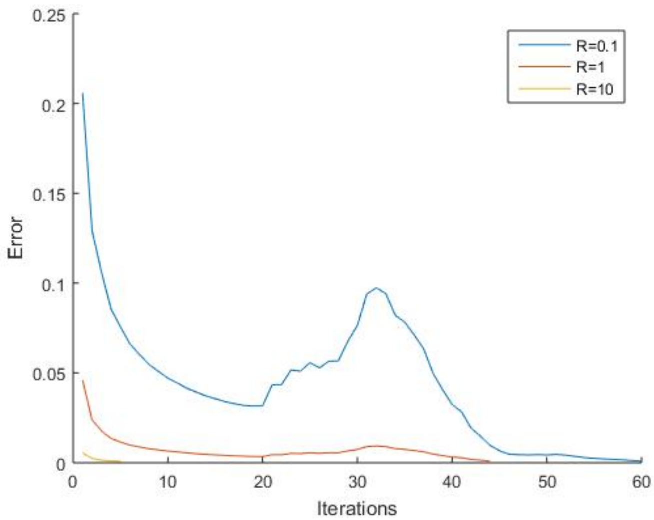

Figure 12: Error convergence for the conditioned conjugate gradient solution.