Numerical Analysis: homework 03

Due on Tuesday, March 21, 2017

102061149 Fu-En Wang

1 Introduction

For modern circult design, the scale of circuit is always such large that human cannot handle it manually. To overcome this, we frequently use a lot of EDA tool such as **hspice** to solve these problems. However, the license of these EDA tool is usualy expensive. As a result, EDA provider such as **Synopsys** benefits a lot from it every year, which means these circuit analysis technology is quite valuable.
In this project, we will implement several simple **Resistor Networks** and use **LU Decomposition** to analyze it.

1.1 Resistor Networks

Each resistor networks only consists of several resistors as shown in Figure 1.

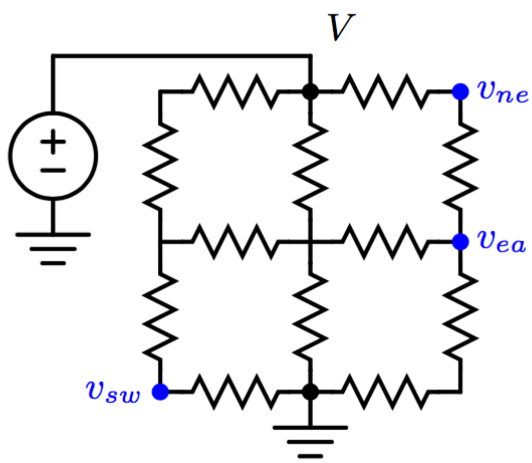

Figure 1: Simple resistor network

In this assignment, there are six networks to implement:

1. 2 resistors each side and resistance is $1K\Omega$
2. 4 resistors each side and resistance is 500Ω
3. 10 resistors each side and resistance is 200Ω
4. 20 resistors each side and resistance is 100Ω
5. 40 resistors each side and resistance is 50Ω
6. 50 resistors each side and resistance is 40Ω

For each of them, we have to show the equivalent resistance and the voltage of V_{ne}, V_{ea} and V_{sw}.

2 Implementation

2.1 Algorithm

To model the problem into A and b, I divide the network into two segments by horizontal and vertical resistor as shown in Figure 1. And then update matrix A and b with **Algorithm 1.4.1** in class material.

Algorithm 1.4.1 System Equation for a Resistor Network

For a network with N nodes in each side, create $N^2 \times N^2$ zero matrix A and N^2 zero vector b.
nodeIndex = 0
G = $resistance^{-1}$
for each i ∈ {1 ... N} **do**
  for each j ∈ {1 ... N-1} **do**
    A[nodeIndex][nodeIndex] += G
    A[nodeIndex][nodeIndex+1] -= G
    A[nodeIndex+1][nodeIndex+1] += G
    A[nodeIndex+1][nodeIndex] -= G
    nodeIndex += 1
  end for
  nodeIndex += 1
end for
nodeIndex = 0
for each i ∈ {1 ... N-1} **do**
  for each j ∈ {1 ... N} **do**
    A[nodeIndex][nodeIndex] += G
    A[nodeIndex][nodeIndex+N] -= G
    A[nodeIndex+N][nodeIndex+N] += G
    A[nodeIndex+N][nodeIndex] -= G
    nodeIndex += 1
  end for
end for
for v,i ∈ voltage and index at fixed voltage point **do**
  A[i] = 0
  A[i][i] = 1
  b[i] = v
end for
LU = luDecompose(A)
Y = forwardSub(LU, b)
X = backwardSub(LU, Y)
X is the voltage of each node.

2.2 Complexity

In all the folowing part, I will refer N as the square of the number of nodes at each side of networks. For the network in Figure 1, N is $3 \times 3 = 9$.
When solving the problem, the most time-comsuming part is the **LU Decomposition**, which is $O(n^3)$. As a result, the system is a $\boldsymbol{O(n^3)}$ problem.

3 Discussion

3.1 Performance Evaluation

In this project, I use five numbers to indicate the efficiency of our method:

1. **Runtime**(total execution time of program)
2. **PROBLEM**(execution time of modeling problem into martrix A and vector b)
3. **LU**(time taken by LU Decomposition)
4. **FWD**(time taken by forward substitution)
5. **BCK**(time taker by backward substitution)