# Numerical Analysis
# homework 05: Conjugate Gradient Methods

Due on Tuesday, April 4, 2017

**102061149 Fu-En Wang**

# 1 Introduction

To solve a such linear system:

$$Ax = b \tag{1}$$

We have already use **LU Decomposition**, **Jacobi**, **Gauss-Seidel** and **Symmetric Gauss-Seidel** to solve it. However, in the previous homework, we found that the three iterative methods are slower than **LU Decomposition**. As a result, to solve the system more faster, **Conjugate Gradient Descend Method** was introduced. **Conjugate Gradient** can solve Equation 1 faster a lot than **LU Decomposition**.

## 1.1 Resistor Network

To evaluate the performance of algorithm, we will build several simple resistor networks(shown in Figure 1) to test the accuracy and efficiency.

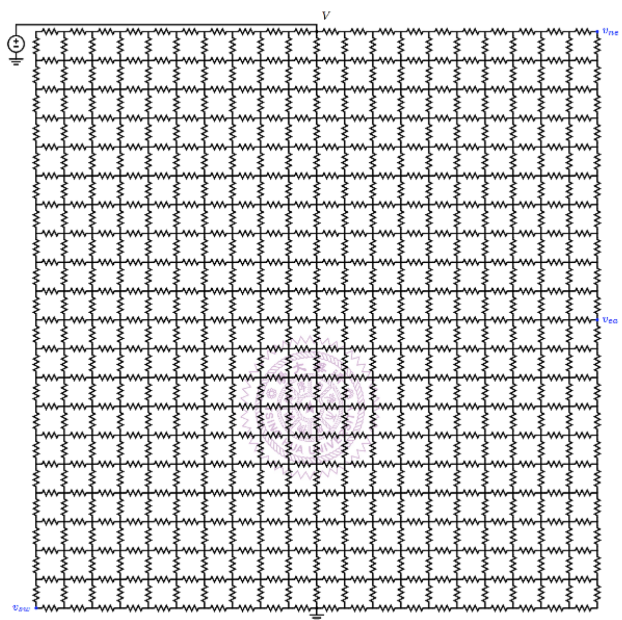


Figure 1: Simple Resistor Network

For **Confugate Gradient Methods**, the error between iteration is defined as Equation 2:

$$Error = \sqrt{\frac{r^T r}{n}} \tag{2}$$

# 2 Implementation

**Algorithm 1 Conjugate Gradient Methods**

$p^{(0)} = r^{(0)} = b - Ax$
**for** each iteration k $\in$ {0, maxIter-1} **do**
 $\alpha_k = \frac{(p^k)^T r^{(k)}}{(p^k)^T A p^k}$
 $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$
 $r^{(k+1)} = r^{(k)} - \alpha_k A p^{(k)}$
 $\beta_k = \frac{(p^{(k)})^T A r^{(k+1)}}{(p^{(k)})^T A p^{(k)}}$
 $p^{(k+1)} = r^{(k+1)} - \beta_k p^{(k)}$
 **if** $\sqrt{\frac{(r^{(k+1)})^T r^{(k+1)}}{n}} < tol$ **then**
  break
 **end if**
**end for**

## 2.1 Complexity

For each iteration of Algorithm 1, the most time-consuming part is the multiplication of Matrix and Vector, which need a double for-loop. As a result, this is a $\boldsymbol{O(n^2)}$ problem.

# 3 Discussion