# CME211:Project Part1 writeup

Yunjae Hwang

November 26, 2018

The steps below are the order of commands running in main.cpp. Since the commands in the program runs sequentially and not complicated, it is visulized without using pseudo-code.

1. Initialize vectors required for solving system of linear equations.
2. Read sparse matrix data in COO format from a input file.
3. Transform the data format (from COO to CSR).
4. Call the function solves the system of equations using the conjugate gradient(CG) algorithm.
5. Print output (solution vector $\mathbf{x}$) to the output file.

The main task for the first part of the project is to implement CG algorithm in the fourth step. The algorithm below(Algorithm 1) summarizes the CG algorithm developed in my program. Basically the same with the description in the assignment instruction. First it has an intial guess of the solution vector $\mathbf{u}_0$ with all elements are 1.0. It computes residual vector $\mathbf{r}_0$ and its second norm, accordingly.

After the initilaliztion, while loop starts and interates until either the ratio of the two residuals becomes less than the tolerance ($\|\mathbf{r}_{n+1}\|_2/\|\mathbf{r}_0\|_2 < \epsilon$) or the number of iteration exceeds the specified number of iteration ($n_{iter} > n_{iter,max}$). Once the while loop finishes, the $u_{n+1}$ vector will be returned to main function as the solution $x$.

Implementing "matvecops" significantly reduces the complexity in the calculation between a vector and a matrix.

In addition to this for reducing redundant programming, two functions are developed for computing the second norm of a vector and inner product of two vectors.

**Algorithm 1** Conjugate Gradient (CG) algorithm for solving a linear system: $\mathbf{Ax} = \mathbf{b}$

Initialize $\mathbf{u_0}$
$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{u}_0$
compute $\|\mathbf{r}_0\|_2$
$\mathbf{p}_0 = \mathbf{r}_0$
**while** $n_{iter} < n_{iter,max}$ **do**
  $n_{iter} = n_{iter} + 1$
  $\alpha = (\mathbf{r}_n^T \mathbf{r}_n)/(\mathbf{p}_n^T \mathbf{A} \mathbf{p}_n)$
  $\mathbf{u}_{n+1} = \mathbf{u}_n + \alpha_n \mathbf{p}_n$
  $\mathbf{r}_{n+1} = \mathbf{r}_n + \alpha_n \mathbf{A} \mathbf{p}_n$
  **if** $\|\mathbf{r}_{n+1}\|_2/\|\mathbf{r}_0\|_2 < \epsilon$ **then**
    **break**
  **end if**
  $\beta_n = (\mathbf{r}_{n+1}^T \mathbf{r}_{n+1})/(\mathbf{r}_n^T \mathbf{r}_n)$
  $\mathbf{p}_{n+1} = \mathbf{r}_{n+1} + \beta_n \mathbf{p}_n$
**end while**
$\mathbf{x} = \mathbf{u}_{n+1}$