EE 242 Numerical Methods for Electrical Engineering

Project 2: Computing Eigenvalues and Eigenvectors using Normalized Power Iteration Algorithm and Inverse Iteration Algorithm

Project Instructions:

• For this project, you will work alone. No collaborations of any sort will be allowed with others. Any violation, regardless of the scope, will be directly referred to the department's Ethical Commission.

• You will submit your program (fully commented and documented) to Moodle. Late submission penalty is 20% for up to one week after the deadline. No credits will be given for late submissions beyond one week.

• You will write in C/C++. You can use Dev-C++ as a compiler or any other compiler you wish. You can download Dev-C++ from: http://dev-c.en.malavida.com/

• Your project will not only be graded on whether it works or not, but also on whether it has good programming style. Specifically, **you are expected to use Object Oriented Programming concepts** you learned in labs.

• You should turn in:

– *Source code:* Fully commented. You should be explicit in your comments. An educated person reading your code should clearly understand the purpose of each line.
Executable or object files are not accepted. The file name should be **source.cpp**

– *A Readme file:* A short file named **readme.txt** containing information regarding how to compile and run your program including the necessary arguments. If your program is incomplete, this should be indicated in the beginning of the Readme file.

Important: You should upload two files, named source.cpp and readme.txt.

DO NOT upload .zip or .rar files, or you will be penalized.

Project Goals:

In this project, you will be implementing the normalized power iteration algorithm and the inverse iteration algorithm to find the largest and the smallest eigenvalues and corresponding eigenvectors of a given matrix A, where A is a real square matrix. Using normalized power iteration, you can find the dominant eigenvalue of A with corresponding eigenvector. Inverse iteration algorithm is equivalent to the power iteration method applied to A^{-1}. Therefore, using inverse iteration algorithm the smallest eigenvalue of A is the reciprocal of the dominant eigenvalue of A^{-1} and

corresponding eigenvector can be found. You should solve a system of linear equations instead of finding A^{-1} directly. (**Hint: You can use your first projects to solve the linear system of equations, but this time you should write your program as an object oriented one**.)

Your program should read A from an input file and output the dominant eigenvalue, its corresponding eigenvector and the smallest eigenvalue with its corresponding eigenvector as a text file.

Example:

The file "A.txt" contains:

2.7383 -0.5011 0.8817

-0.3039 2.3639 1.4258

0.1285 0.0665 3.8978

Tolerance is given as 1e-6.

Then the output file should contain:

Eigenvalue#1: 4.00

0.38

0.80

1.00

Eigenvalue#2: 2.00

0.78

1,00

-0.09

Programming Details:

Your program should

• have three command line arguments for the parameters. (Command line arguments can be thought of as the inputs of the main function.) The first argument is the name of the file you read the matrix from, the second argument is the tolerance, which will be used in the normalized power iteration algorithm and inverse iteration algorithm, and the third argument is the name of your output file,

• use dynamically allocated memory to store the matrix,

• print out an error message and quit if it detects any problems,

• calculate the eigenvalues and corresponding eigenvector and write them in a text file.

************About Object Oriented Programming************

In this project you will use matrices and matrix operations. In order to implement this project as an object oriented program; for example, you can declare a class (an object) named Matrix, and implement all matrix operations such as multiplication, addition, transpose, and solving a system of linear equations (LU factorization and backward substitution) as methods of this class. This way, you don't need to have a lot of complex loops in your program.