ENGG 27.01 - Advanced Engineering Math with Numerical Methods, Lecture
ENGG 27.01 M 2025-2
Luisito L. Agustin

# 20260220

## Project 2
## Power Method for Eigenvalues and Eigenvectors

ENGG 27.01 - Advanced Engineering Math with Numerical Methods, Lecture
ENGG 27.01 M 2025-2
Luisito L. Agustin

# Project 2
# Power Method for Eigenvalues and Eigenvectors

Due: start of class time, March 24, 2026

| | |
|---|---|
| Progress Report 1 | Feb. 27 |
| Progress Report 2 | Mar. 6 |
| Progress Report 3 | Mar. 13 |
| Early Work Submission | Mar. 17 |

Numerical Methods Project

# Power Method for Eigenvalues and  Eigenvectors

*Luisito L. Agustin*
*2026 02 20*

## General Specifications

Develop an application in C++ that demonstrates use of the Power Method for finding eigenvalues and eigenvectors of matrices.

## Detailed Specifications

Develop an application in C++ that demonstrates use of the Power Method for finding eigenvalues and eigenvectors of matrices.

Refer to the Matrix File Format specified in a separate document. The application obtains a matrix from a matrix file as specified in the Matrix File Format document. If the matrix obtained is a square matrix, the application attempts to determine eigenvalues and eigenvectors of the square matrix using several test vectors depending on the dimensions of the square matrix.

Provide a means and appropriate guidance for the user to specify the matrix file. Do not modify any filename provided by the user. Do not assume, nor require, nor add a ".txt" extension. Use whatever filename is provided.

Given an *n* x *n* matrix, the application attempts to find eigenvalues and eigenvectors *n* times using the power method with the *n* unit vectors as initial test vectors :
[ 1, 0, 0, ..., 0, 0] (transpose),
[ 0, 1, 0, ..., 0, 0] (transpose),
[ 0, 0, 1, ..., 0, 0] (transpose),
[ 0, 0, 0, ..., 1, 0] (transpose), and
[ 0, 0, 0, ..., 0, 1] (transpose).

The application monitors the iterations of the power method for convergence or divergence. If divergence is detected, the iterations are stopped. If the estimates for the eigenvalue and eigenvector appear to converge, the application continues the iterations until the most precise value of the eigenvalue is obtained.

After each attempt with one of the *n* unit vectors as initial test vector, the application either reports divergence, or reports the most precisely determined value of the eigenvalue found along with a normalized eigenvector scaled such that its largest value is 1.0.  The eigenvalue and the entries in the eigenvector are reported using the maximum precission available for a *double*.

Provide an option where the application would also provide a trace of the intermediate estimates for the eigenvalues and eigenvectors on the console as the Power Method progresses, in addition to reporting the final result for each attempt.

Do not limit the sizes of matrices to be processed.

## Revision History - Power Method for Eigenvalues and Eigenvectors

| | |
|---|---|
| 2026 02 20 | first version |

## Variable Specifications

Specifications and instructions from this point onwards may vary from one use of the document to the next. These are not tracked as revisions to the project specifications.

## Group Work

The implementation of this project in C/C++ may be worked on by groups. Documentation, testing, and all other parts of the project must be done individually.

It is expected that C++ code submitted by groupmates may be similar or exactly the same. Points earned for the entire project will be subject to penalties for unauthorized collaboration if documentation submitted by at least two students contain at least one distinctive phrase, image, or item in common.

## Submission

*Implementation and Testing Environment Report*

Provide details of at least one platform used in implementing and testing the project.

* name and version of compiler
* name and version of integrated development environment ( IDE )
* operating system name/distribution and version

screenshot of compilation

The screenshot must show the files involved and show feedback from the IDE reporting successful compilation and linking; if the project is compiled on the command line, the screenshot must show the commands issued; if a makefile was used, include a copy of the makefile in addition to the screenshot.

screenshot of program run on the console

Copy the executable to a location whose path contains a recognizable version of your name. If working with a group, make sure that the path does not contain any version of the names of any groupmates. Create a new folder named after yourself as needed. Run the executable by typing its name at the command prompt, and take a screenshot as the executable starts. Do not run the application from within an IDE.

ENGG 27.01 - Advanced Engineering Math with Numerical Methods, Lecture
ENGG 27.01 M 2025-2
Luisito L. Agustin

*Documentation*

Document the results obtained when your application is used to determine the eigenvalues and eigenvectors of the 5 x 5 matrix

| | | | | |
|---|---|---|---|---|
| -8.8 | 0.5 | 6.2 | 1.3 | -7.2 |
| 9.6 | 6.7 | -1 | -0.8 | -6.4 |
| -4.3 | 7.3 | 2.2 | -0.7 | 3.1 |
| -9.4 | -7.7 | -0.2 | -8.4 | -4.4 |
| -7.7 | 2.5 | -3 | -6.4 | 4.2 |

Include a matrix file for the given matrix in your submission.

Based on spreadsheet computations, the matrix has at least one known eigenvalue: -14.34809723455140
and eigenvector:
0.378433257542477
0.020798125181845
0.036802278566342
1.000000000000000
0.505299680463325

Obtain screenshots showing the estimated eigenvalues and eigenvectors from at least the first three iterations starting with each of the 5 unit vectors, regardless of convergence or divergence, and at least the last three iterations if the process converges.

Do a self-evaluation using the project evaluation table and evaluation procedures provided. Fill in your scores and add your points correctly.

Place all required documentation and the self-evaluation table together in a single document and export this document to pdf. Label all screenshots accurately.

Submit a single zip file containing
* C++ implementation files and header files for the project
* required documentation
* all other required files

Do NOT include executables in your submission.

ENGG 27.01 - Advanced Engineering Math with Numerical Methods, Lecture
ENGG 27.01 M 2025-2
Luisito L. Agustin

| **Project Evaluation - Power Method for Eigenvalues and Eigenvectors** | | |
|---|---|---|
| Item | Points | Rubrics |
| early work | 8/8 | (all or nothing)<br>8: at most 10% of the code in the final implementation differs from that in early work submission |
| implementation and testing environment report | 4/4 | 4 - all instructions followed correctly |
| user interface: input file | 4/4 | 4- appropriate user interface provided |
| parsing of matrix files | 8/8 | 8 - application properly parses matrix files |
| documentation of Power Method implementation | 32/32 | 32 - properly working and clearly documented implementation of the Power Method |
| convergence | 8/8 | 16 - all instuctions related to convergence are properly implemented |
| divergence | 8/8 | 16 - divergence is always detected properly |
| generality | 16/16 | 16 - implementation of Power Method works for all cases |
| matrix file | 4/4 | 4 - correctly prepared matrix file |
| program output | 4/4 | 4 - output produced according to instructions |
| self-evaluation | 4/4 | 4 - self-evaluation accurate (or evaluating this item leads to an error) |
| total | 100/100 | |

## Evaluation Procedures

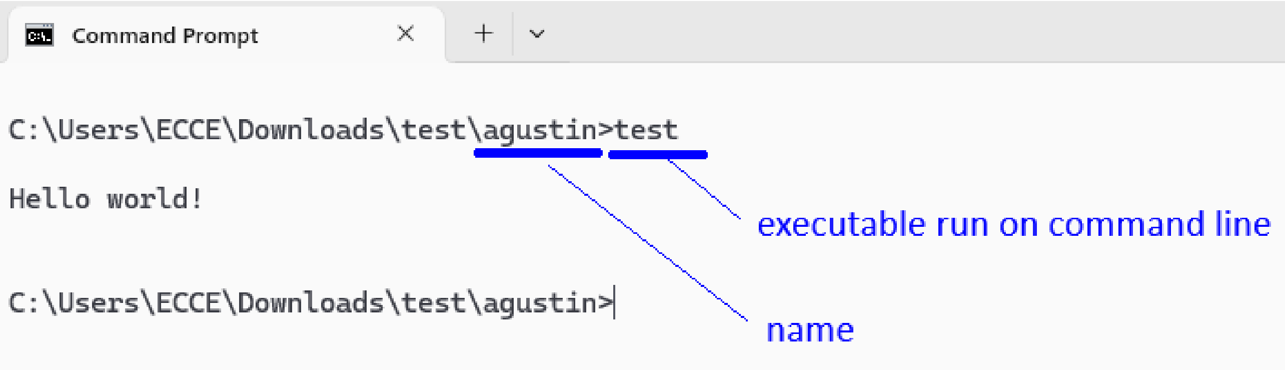
Special Cases

Score for the entire project is 0 if
* no source files are submitted
* the source files, as submitted, will not compile as a C++ project
* the implementation uses resources that are not allowed

early work

Early work credits are based on the final code submitted, assuming the final and early work code use the same style. The total number of lines in the final code that are not in the early work version, or that were modified from the early work version, other than formatting changes that do not modify compiled code, must not be more than 10% of the number of lines in the final code, excluding blank lines.

Implementation and Testing Environment (4 pts)
1 - all details of implementation platform are provided
1 - screenshot of compilation: all instructions are followed correctly
1 - screenshot of program run on the console: all instructions are followed correctly
1 - all filenames appearing in the screenshots are consistent with files in the project submission

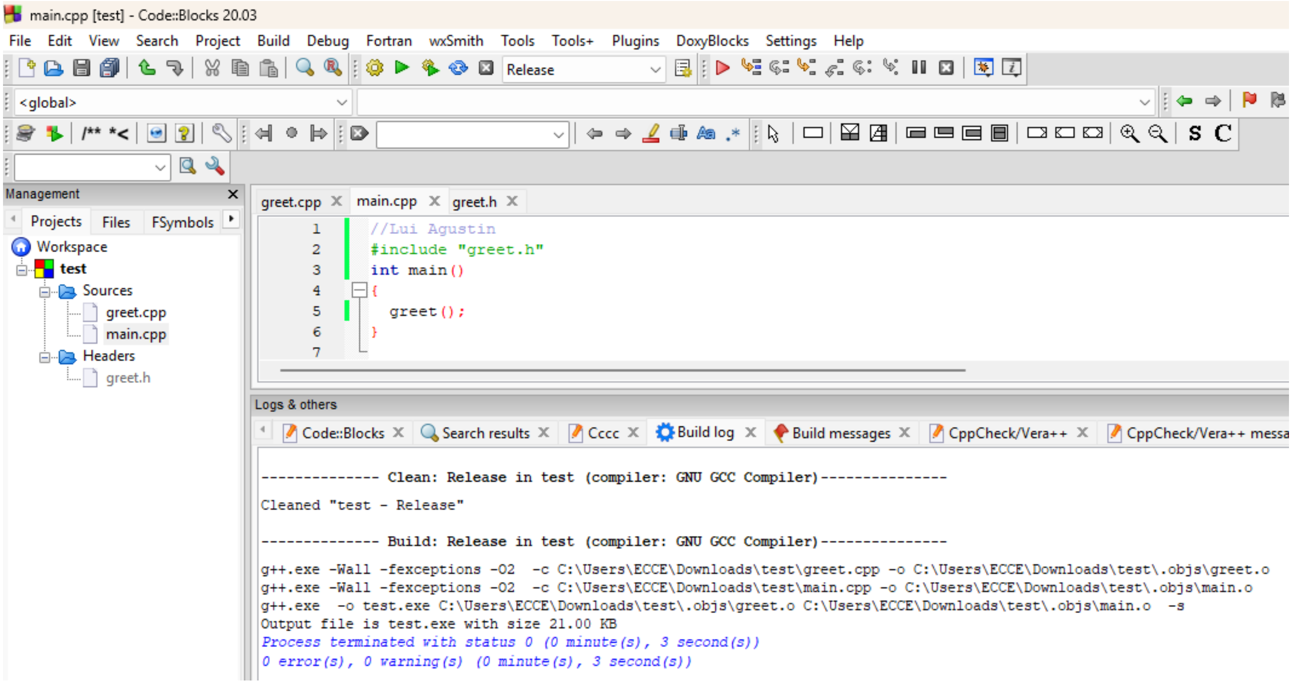


```
C:\Users\ECCE\Downloads\test\agustin>test

Hello world!


C:\Users\ECCE\Downloads\test\agustin>
```

sample screenshot - executable test.exe run by typing its name at the prompt from a path with name of author


main.cpp [test] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Release
<global>
Management
Projects Files FSymbols
Workspace
test
Sources
greet.cpp
main.cpp
Headers
greet.h
greet.cpp main.cpp greet.h
Logs & others
Code::Blocks Search results Cccc Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messa

```
//Lui Agustin
#include "greet.h"
int main()
{
  greet();
}
```

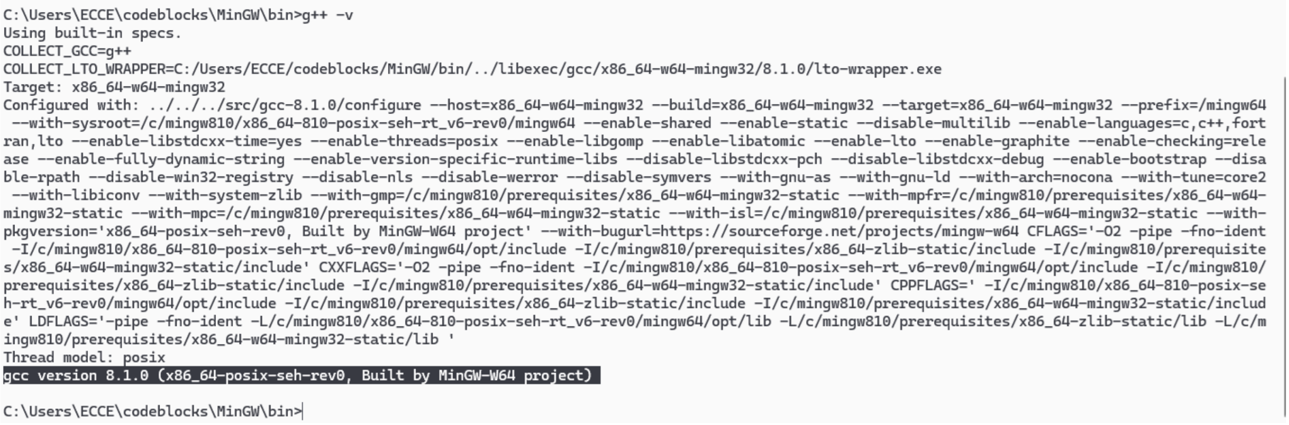
```
------------- Clean: Release in test (compiler: GNU GCC Compiler)---------------

Cleaned "test - Release"

------------- Build: Release in test (compiler: GNU GCC Compiler)---------------

g++.exe -Wall -fexceptions -O2  -c C:\Users\ECCE\Downloads\test\greet.cpp -o C:\Users\ECCE\Downloads\test\.objs\greet.o
g++.exe -Wall -fexceptions -O2  -c C:\Users\ECCE\Downloads\test\main.cpp -o C:\Users\ECCE\Downloads\test\.objs\main.o
g++.exe  -o test.exe C:\Users\ECCE\Downloads\test\.objs\greet.o C:\Users\ECCE\Downloads\test\.objs\main.o  -s
Output file is test.exe with size 21.00 KB
Process terminated with status 0 (0 minute(s), 3 second(s))
0 error(s), 0 warning(s) (0 minute(s), 3 second(s))
```

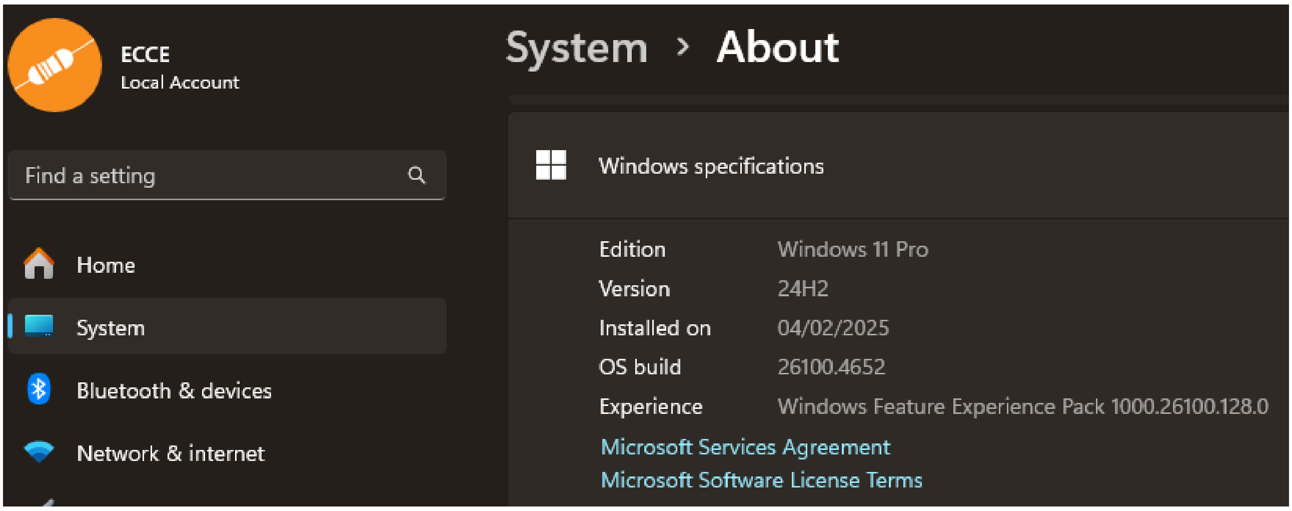
sample screenshot - compilation on Code::Blocks



```
C:\Users\ECCE\codeblocks\MinGW\bin>g++ -v
Using built-in specs.
COLLECT_GCC=g++
COLLECT_LTO_WRAPPER=C:/Users/ECCE/codeblocks/MinGW/bin/../libexec/gcc/x86_64-w64-mingw32/8.1.0/lto-wrapper.exe
Target: x86_64-w64-mingw32
Configured with: ../../../src/gcc-8.1.0/configure --host=x86_64-w64-mingw32 --build=x86_64-w64-mingw32 --target=x86_64-w64-mingw32 --prefix=/mingw64
 --with-sysroot=/c/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64 --enable-shared --enable-static --disable-multilib --enable-languages=c,c++,fort
ran,lto --enable-libstdcxx-time=yes --enable-threads=posix --enable-libgomp --enable-libatomic --enable-lto --enable-graphite --enable-checking=rele
ase --enable-fully-dynamic-string --enable-version-specific-runtime-libs --disable-libstdcxx-pch --disable-libstdcxx-debug --enable-bootstrap --disa
ble-rpath --disable-win32-registry --disable-nls --disable-werror --disable-symvers --with-gnu-as --with-gnu-ld --with-arch=nocona --with-tune=core2
 --with-libiconv --with-system-zlib --with-gmp=/c/mingw810/prerequisites/x86_64-w64-mingw32-static --with-mpfr=/c/mingw810/prerequisites/x86_64-w64-
mingw32-static --with-mpc=/c/mingw810/prerequisites/x86_64-w64-mingw32-static --with-isl=/c/mingw810/prerequisites/x86_64-w64-mingw32-static --with-
pkgversion='x86_64-posix-seh-rev0, Built by MinGW-W64 project' --with-bugurl=https://sourceforge.net/projects/mingw-w64 CFLAGS='-O2 -pipe -fno-ident
 -I/c/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64/opt/include -I/c/mingw810/prerequisites/x86_64-zlib-static/include -I/c/mingw810/prerequisite
s/x86_64-w64-mingw32-static/include' CXXFLAGS='-O2 -pipe -fno-ident -I/c/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64/opt/include -I/c/mingw810/
prerequisites/x86_64-zlib-static/include -I/c/mingw810/prerequisites/x86_64-w64-mingw32-static/include' CPPFLAGS=' -I/c/mingw810/x86_64-810-posix-se
h-rt_v6-rev0/mingw64/opt/include -I/c/mingw810/prerequisites/x86_64-zlib-static/include -I/c/mingw810/prerequisites/x86_64-w64-mingw32-static/includ
e' LDFLAGS='-pipe -fno-ident -L/c/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64/opt/lib -L/c/mingw810/prerequisites/x86_64-zlib-static/lib -L/c/m
ingw810/prerequisites/x86_64-w64-mingw32-static/lib '
Thread model: posix
gcc version 8.1.0 (x86_64-posix-seh-rev0, Built by MinGW-W64 project)

C:\Users\ECCE\codeblocks\MinGW\bin>
```

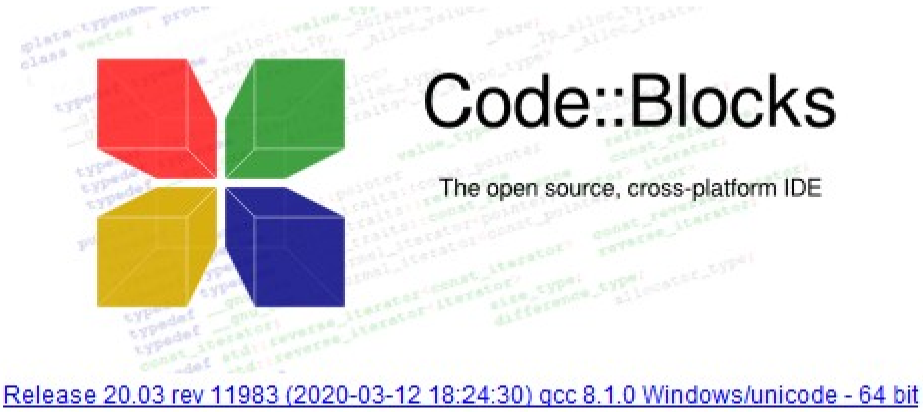
compiler - gcc version 8.1.0 (x86_64-posix-seh-rev0, Built by MinGW-W64 project)



Edition Windows 11 Pro
Version 24H2
Installed on 04/02/2025
OS build 26100.4652
Experience Windows Feature Experience Pack 1000.26100.128.0

Microsoft Services Agreement
Microsoft Software License Terms

operating system - Windows 11 Pro, 24H2, build 26100.4652



Release 20.03 rev 11983 (2020-03-12 18:24:30) gcc 8.1.0 Windows/unicode - 64 bit

IDE -Code::Blocks 20.03
rev 11983 (2020-03-12 18:24:30) gcc 8.1.0 Windows/unicode - 64 bit

self-evaluation
0 - not done
2 - self-evaluation score differs from project score by more than 10 points or scores not tallied properly
4 - self-evaluation accurate (or evaluating this item leads to an error)

Appendix

Matrix File Format
see next page

# Matrix File Format

*Luisito L. Agustin*

*2023 03 23*

The Matrix File Format is a text file format for encoding matrices of floating point numbers or real numbers. There is no required file name extension for matrix files. Compliance with the format is based on the contents of a file.

Matrix data for a matrix with *m* rows and *n* columns are encoded in a text file as follows:

$a_{1,1}$ $a_{1,2}$ ... $a_{1,n}$ optional comments
$a_{2,1}$ $a_{2,2}$ ... $a_{2,n}$ optional comments
.
.
.
$a_{m,1}$ $a_{m,2}$ ... $a_{m,n}$ optional comments
optional comments

Each row of the matrix is encoded in a separate line. The number of floating point numbers that could be extracted in sequence from the first line of the file determines the number of columns of the matrix. The numbers in a row are separated from each other by one or more white spaces. The row ends if either the end of the line has been reached or the next item on the line cannot be interpreted as a valid floating point number.

There may be any amount of white space before the first entry on a line and after the last entry on a line.

The number of rows of the matrix is determined from the number of lines, following the first line of the file,  from which it is possible to extract the required number of floating point numbers, equal to the number of columns determined from the first line of the file.

**Recommendations**

Applications producing matrix files should use a space followed by a tab ( `" \t"` ) as separator between entries on a line.

## Notes

The extraction operator ( >> ) in C++ will skip white spaces before attempting to extract the desired content. The parser stops when white space is encountered.

A valid matrix can be extracted from the file if the first line of the file has at least one valid entry. Clearly the matrix will have at least one row in such a case. If the second row has less valid entries than the first, then the matrix would be a one-row matrix. The matrix would not have a second row.

## Revision History - Matrix File Format

2022 10 25: first version
2023 03 23: revisions for clarity