20251017

Homework
Numerical Methods in Undergraduate Theses

Homework
Course Digest and Self-Assessment

Project 3
Gauss Elimination

ENGG 27.01 - Advanced Engineering Math with Numerical Methods, Lecture
ENGG 27.01 M 2025-1
Luisito L. Agustin

Homework
Numerical Methods in Undergraduate Theses

(on Canvas)

Luisito L. Agustin

Homework
Course Digest and Self-Assessment

(on Canvas)

Project 3
Gauss Elimination

Due: start of class time, November 21, 2025

Progress Report 1	Oct. 24
Progress Report 2	Oct. 28
Progress Report 3	Nov. 7
Early Work Submission	Nov. 18

ENGG 27.01 Project

Gauss Elimination

Luisito L. Agustin
2025 03 22

Reference

[Kreyszig] (book) Erwin Kreyszig: *Advanced Engineering Mathematics*, 10th ed, 2011.
(Part E Numeric Analysis, Chap 19-21).

General Specifications

Use C++ to write an application for solving linear systems of equations using Gauss Elimination.

Detailed Specifications

Use C++ to write an application for solving linear systems of equations using Gauss Elimination.

The application imports data from a matrix file specified by the user. The application provides at least one option for the user to specify the matrix file, either thru the command line or thru an interactive user interface.

The required scope of the application only covers systems of equations with exactly the same number of equations as unknowns. The application should check that the matrix obtained is a possible augmented matrix of a system of equations that the application could handle.

Implement Gauss Elimination as specified in [Kreyszig].

Table 20.1 Gauss Elimination

ALGORITHM GAUSS ($\tilde{\mathbf{A}} = [a_{jk}] = [\mathbf{A} \quad \mathbf{b}]$)

This algorithm computes a unique solution $\mathbf{x} = [x_j]$ of the system (1) or indicates that (1) has no unique solution.

INPUT: Augmented $n \times (n + 1)$ matrix $\tilde{\mathbf{A}} = [a_{jk}]$, where $a_{j,n+1} = b_j$

OUTPUT: Solution $\mathbf{x} = [x_j]$ of (1) or message that the system (1) has no unique solution

For $k = 1, \cdots, n - 1$, do:

1 $m = k$

For $j = k + 1, \cdots, n$, do:

If $(|a_{mk}| < |a_{jk}|)$ then $m = j$

End

If $a_{mk} = 0$ then OUTPUT "No unique solution exists"

Stop

[*Procedure completed unsuccessfully*]

2 Else exchange row k and row m

3 If $a_{nn} = 0$ then OUTPUT "No unique solution exists."

Stop

Else

4 For $j = k + 1, \cdots, n$, do:

$$m_{jk} := \frac{a_{jk}}{a_{kk}}$$

5 For $p = k + 1, \cdots, n + 1$, do:

$$a_{jp} := a_{jp} - m_{jk}a_{kp}$$

End

End

End

6 $$x_n = \frac{a_{n,n+1}}{a_{nn}}$$ [*Start back substitution*]

For $i = n - 1, \cdots, 1$, do:

7 $$x_i = \frac{1}{a_{ii}}\left(a_{i,n+1} - \sum_{j=i+1}^{n} a_{ij}x_j\right)$$

End

OUTPUT $\mathbf{x} = [x_j]$. Stop

End GAUSS

from [Kreyszig]

Implement Gauss Elimination in a single C++ function with appropriate parameters. This function must not call on any other functions except those available thru the C++ standard library. Implement partial pivoting. Include the back-substitution process in the function. This function must be general enough to work on systems of any size.

The application must provide appropriate feedback at all times, including information about errors that might occur, and clear information on how to use it.

Revision History - ENGG 27.01 Project :: Gauss Elimination

2025 03 22 first version

Variable Instructions

Sections marked "(variable)" and 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 exercise.

Group Work

Work on the C++ code may be done by groups. Documentation and testing must be done individually.

Submission

Basic Test

Encode the augmented matrix in a matrix file:

-2.8	-4.4	8.8	-1.7	2.0
-3.3	-9.2	-2.4	-8.6	1.5
-7.0	-6.7	-2.6	-9.3	1.7
0.2	-4.8	-9.7	-7.3	-6.9

Take a screenshot of the solution computed by your application.

The solution should have these values:

-3.15224383
-3.3074943
-1.4668796
4.982774792

Obtain a screenshot of the output of the application on the console in response to basic test data. Be sure the screenshot shows the same results as those expected for the basic test. Place the screenshot in a document with your name.

Submit a single zip file containing:

* the pdf document containing the required screenshot
* C++ code worked on
* the self-evaluation .

Project Evaluation		
Item	Points	Rubrics
early work	8/8	(all or nothing) 8: at most 10% of the code in the final implementation differs from that in early work submission
basic test	50/50	0 - fails basic test 50 - computes the correct result from basic test data and displays the correct result
	10/10	0 - no screenshot of correct basic test result 10 - screenshot of correct basic test result submitted in a pdf document
input validation	4/4	4 - floating point numbers are fully validated when extracted from files
partial pivoting	8/8	0 - not implemented or fails basic test 8 - partial pivoting works correctly at all times
generality of implementation	6/6	6 - valid linear systems are solved correctly at all times
C++ function implementation of Gauss Elimination	6/6	0 - no function or no general implementation 6 - general implementation in a C++ function
user interface and robustness of application	4/4	0 - lacks feedback or application crashes or hangs 4 - application is robust, feedback is correct and appropriate at all times
self-evaluation	4/4	0 - no self-evaluation 2 - self-evaluation score differs from project score by more than 10 points 4 - self-evaluation accurate (or evaluating this item leads to an error)
total	100	

Notes

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