Course code : CSC 403

Course title: Numerical Analysis

Group number: 1

PROJECT REPORT AND DOCUMENTATION

Objective of project:

Create a computer program in C or FORTRAN that addresses a certain computational issues.

- Compute a Hilbert matrix of size n
- Compute inverse of matrix and solve x = (A inverse) X b.
- Solve the problem by Crammer's rule.
- Solve the problem by Gauss Elimination with partial pivoting
- Solve the problem by LU decomposition and return the solution
- Find inverse and determinant of Hilbert matrix by LU.
- Compute [(A inverse) x A] and [A x (A inverse)] and report

Project members and Matriculation Numbers

- ✓ Nwafor Happi Yvan Joel Mat: SC14B375
- ✓ Chawa Fabu Elcid Chawa Mat: SC14A242
- ✓ Ngodi Albert Mat: SC14A866
- ✓ Metchezin Franklin Mat:SC14A730
- ✓ Tchassem Guemte Borel Mat: SC14B182
- ✓ Boubakary Wadjiri Mat: SC14A229
- ✓ Ngala Diran
- ✓ Ngoulla Sob Christian Mat: SC14A884

PROJECT APPROACH

The project was approached from a procedural manner in the C programming language, in which each problem to be addressed was viewed as a procedure which could later on call sub-procedures for better understanding of how the procedures functions and ease with debugging of the project. Some layer of abstraction was created to provide a better view of the entire project (i.e. Rather than working directly with 'arrays' in C with various rows and cols passed to procedures, we created a new types called Matrix and Equations). This provided better understanding of our project as a team and hence better productivity.

Project Issues and their solutions

Major issues faced where at the various levels

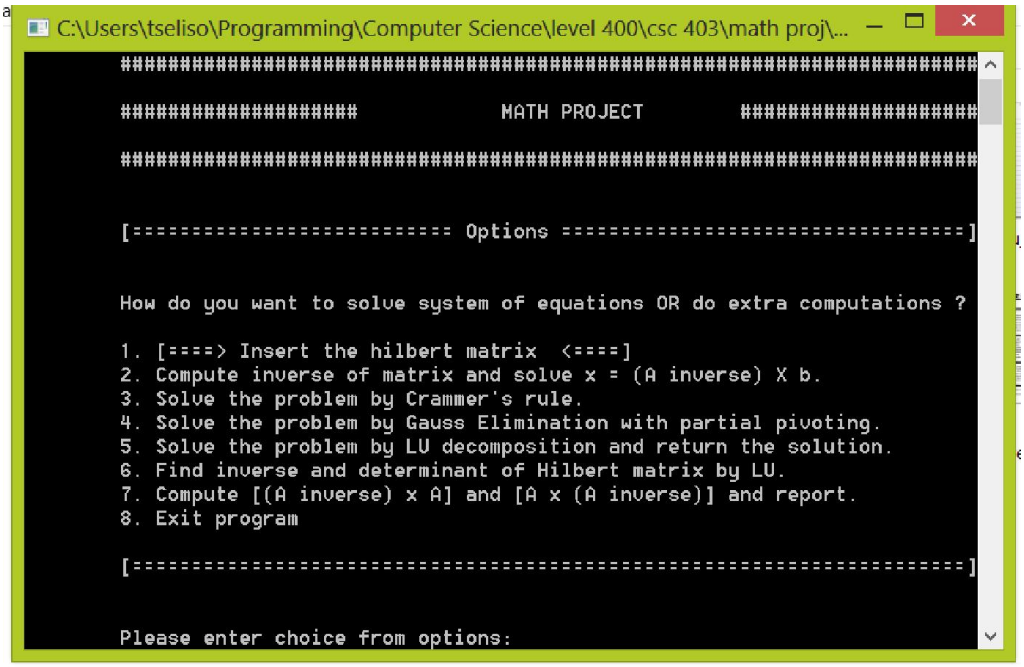
- Abstracting the matrix (with system of equations) for better understanding
- Floating-point arithmetic error which occurs as a result of computations done on the Hilbert matrix for large n, which did exceed the maximum possible floating-point number. We try our best but could not still get a way around it. However, various methods were introduces by replacing the float type to a double type and again to long double, for extensibility.
- Implementing the functionality like Gaussian elimination was/is difficult, so to be honest a little help was gotten from the Internet here. We did not copy and pasted any program; we rather understood better the pseudo-code and hence having a better implementation.
- Issues with memory and space were addressed with dynamic memory management techniques.

MEMBERS CONTRIBUTION

Nwafor Happi Yvan Joel	Worked on 1(a) / Documentation / combined the different subtasks into one file.
Metchezin Franklin	Worked on LU decomposition
Tchassem Guemte Borel	Worked on Crammer's rule.
Ngoulla Sob Christian	Worked on 1(d) & 2(a) run lu_decomposition.exe
Ngondi Albert	Worked on 1(d)
Boubakary Wadjiri	Worked on 2(b)
Chawa Fabu Elcid Chawa	Worked on Gaussian Elimination / Documentation

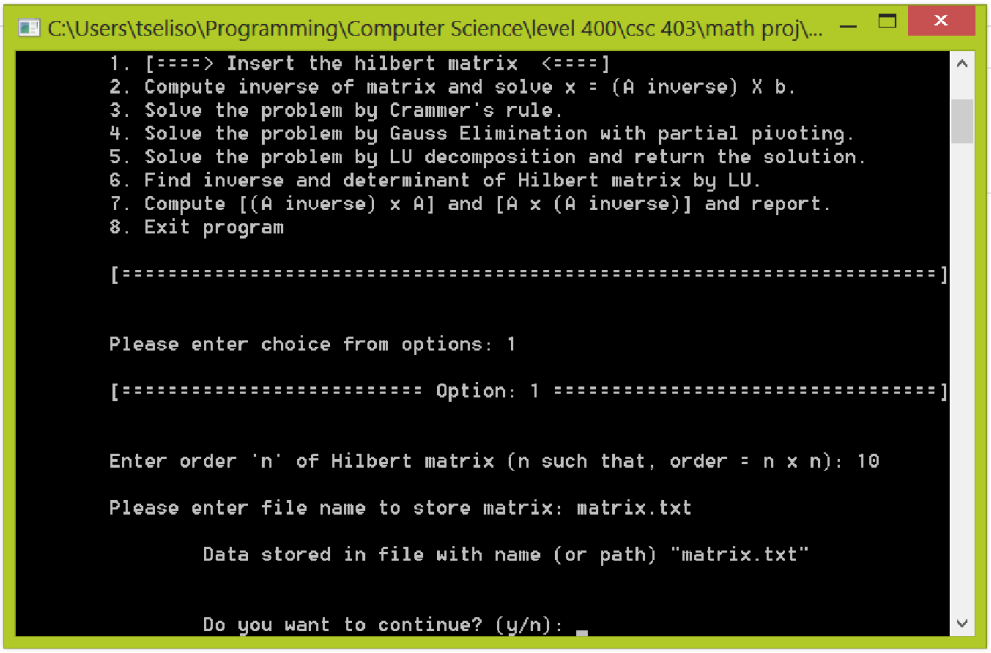
HOW TO RUN PROGRAM

The program in just run by double clicking the 'math_program.exe'. The result is as on the screen shot below.



Options are described as follows:

1. Allows the user to create a Hilbert Matrix with a specified order, for example order 10 as used in the screen shot below.



After that we take the option "y" to continue with the other and "n" to quit the program.

2. The option "2", request the name of the matrix creates (e.g. matrix.txt as in the screen shot above.), then calculates the inverse and solves the Linear system.
3. Option "3", solves the Linear system by Crammers rule.
4. Option "4", solves the Linear system by Gauss Elimination.

Note: Options "5" and "6", where implemented in a different file and uneasy to in cooperate. The LU decomposition is found in the file lu_decomposition.c, and can be tested by running: “lu_decomposition.exe”.

Option "7" computes $A^{-1}A$ and AA^{-1} and Option "8" exits the program.