Numerical Methods

**Assignment3: Solving a System of Linear Equations**

**Using LU decomposition**

**Name: Hong Se Hyun**  **ID: 21700791**

**Instructions:**

* Check if your function is accepting only square matrices
* You should insert exceptional/error handling(e.g. giving error message when square matrix is not used, div by zero etc)

**Problem: Solve the following linear systems of Ax=b, using Gauss elimination [20pt]**

**Q1. Determine the displacement of the three masses**

They are in the equilibrium states, and u1,u2, u3 are the relative displacement for each mass.

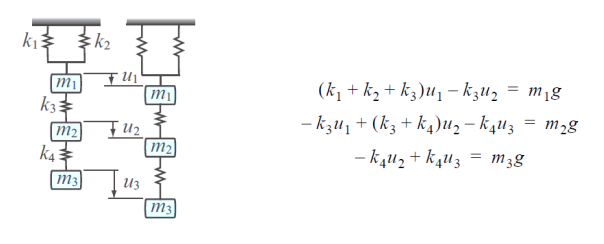


Figure 1.1. Problem

**Procedure**

* Add exceptional/error handling for when A is not square, dimension of A, b are not appropriate, division by zero and so on

|  |
| --- |
|  |
| Figure 2.1. Error Checking |

**1. LU decomposition without partial pivoting[20pt]. If scaled partial pivoting is applied [40 pt]**

Create a C/C++ function that processes the LU decomposition

- Input: matrix A(n x n), vector b(n x1)

- Output: matrix U, matrix L (option)permutation matrix P

- Declare in “myNM.h” and define in “myNM.c”

- See appendix for help

void LUdecomp (Matrix A, Matrix L, Matrix U, Matrix P);

**First write a pseudocode**

**Show your code here**

|  |  |
| --- | --- |
|  |  |
| Figure 3.1. PLU decomposition code with Pivoting | Figure 3.2. PLU decomposition code with Pivoting |
|  |  |
| Figure 3.3. PLU decomposition code with Pivoting | Figure 3.4. PLU decomposition code with Pivoting |

**Scaled Partial Pivoting에 관한 참고자료**

|  |  |
| --- | --- |
|  |  |
| Figure 4.1. Scaled Partial Pivoting Method | Figure 4.2. Scaled Partial Pivoting Flowchart |

**2. Create a function that solves for Ax= LUx=b [20pt].If permulation P is applied [40pt]**

double solveLU (Matrix L, Matrix U, Matrix P, Matrix b);

or void solveLU (Matrix L, Matrix U, Matrix P, Matrix b, Matrix x);

- Input: matrix L, U, P from LUdecomp(A,L,U,P) and vector b(n x1)

- Output: vector x (nx1)

- Declare in “myNM.h” and define in “myNM.c”

LUdecomp(A,L,U,P)

double solveLU(L,U,P,b) // or double solveLU(L,U,b)

{

fwdsub(….) // what should be the input arguments to include P?

backsub(….) // what should be the input arguments to include P?. . .

return vectorX }

**First write a pseudocode**

*[ P = row Exchange Matrix ]*

**Scaled Partial Pivoting이 적용된 LU decomposition에서 해를 찾는 과정에 대한 참고자료**

|  |
| --- |
|  |
| Figure 5.1. Finding Solution from LU decomposition |

**Then, create a C function**

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | |
| Figure 6.1. Function of | | Figure 6.1. Function of | |
|  |  | |
| Figure 6.3. Function of | Figure 6.4. Function of | |

와 행 교환 행렬 를 곱하기 위해서, 또한 아래에 있는 3번 문제에서 역행렬을 이용한 해를 구할 때 과

를 곱함으로써 해를 바로 구하기 위해 라는 함수를 따로 만들어 사용해주었다.

**3. Create a function that finds the inverse of A. [20pt]**

double inv(Matrix A, Matrix Ainv);

- You can use LU decomposition, Gauss-Jordan elimination etc..

- You must check (1)A is square (nxn) and (2) it is full rank. rank(A)=n

- Check your answer by **x=Ainv\*b**

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | |
| Figure 7.1. Inverse Matrix code with Pivoting | Figure 7.2. Inverse Matrix code with Pivoting | |
|  | |  | |
| Figure 7.3. Inverse Matrix code with Pivoting | | Figure 7.4. Inverse Matrix code with Pivoting | |

Inverse Matrix를 구하는 코드에서도 Scaled Partial Pivoting 과정을 추가하여 역행렬이 정상적으로 구해질 수 있도록 코드를 작성하였다.

**4. Show the output results**

|  |
| --- |
|  |
| Figure 8.1. Result of Problem 1 (LU decomposition with Scaled Partial Pivoting) |
|  |
| Figure 8.2. Result of Problem 2 (Solve ) |
|  |
| Figure 8.3. Result of Problem 3 (Find the Inverse of A) |

**5. Check your answer with the output from MATLAB**

|  |  |
| --- | --- |
|  |  |
| Figure 9.1. MATLAB code | Figure 9.2. MATLAB code Result |

Visual Studio를 통해 확인한 결과값과 MATLAB을 통해 얻어진 결과값이 일치함을 볼 수 있다.

**\*\* For LU with pivoting, check your process with the values shown in Appendix.**

**TA will check your library with test matrices that need pivoting.**

|  |
| --- |
|  |
| Figure 10.1. Result of Appendix Matrix 5 by 5 |
|  |
| Figure 10.2. Result of Appendix Matrix 5 by 5 |
|  |
| Figure 10.3. Result of Appendix Matrix 5 by 5 |

|  |  |
| --- | --- |
|  |  |
| Figure 10.4. MATLAB code | Figure 10.5. MATLAB code Result |

Pivoting이 정상적으로 이뤄졌는지 확인하기 위해 Appendix에 있는 5 by 5 행렬을 이용하여 Visual Studio 프로그램을 다시 실행시켰다. 그 이후에 MATLAB의 결과값과 일치하는지 확인함으로써, Scaled Partial Pivoting 기능까지 구현된 프로그램을 완성하였음을 확인할 수 있었다.

참고로, 위에서 의 값은 [ 1 ; 2 ; 3 ; 4 ; 5 ]로 설정해주었다.