Numerical Methods

**Assignment2: Solving a System of Linear Equations**

**Gauss Elimination without Pivoting**

**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.

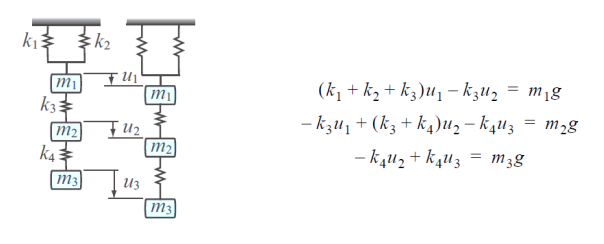


Figure 1.1. Problem

**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 – Square Matrix |
|  |
| Figure 2.2. Error Checking – Square Matrix |

|  |
| --- |
|  |
| Figure 2.3. Error Checking – Dimension of Matrix b |
|  |
| Figure 2.4. Error Checking – Dimension of Matrix b |

|  |
| --- |
|  |
| Figure 2.5. Error Checking – Division by Zero |
|  |
| Figure 2.6. Error Checking – Division by Zero |

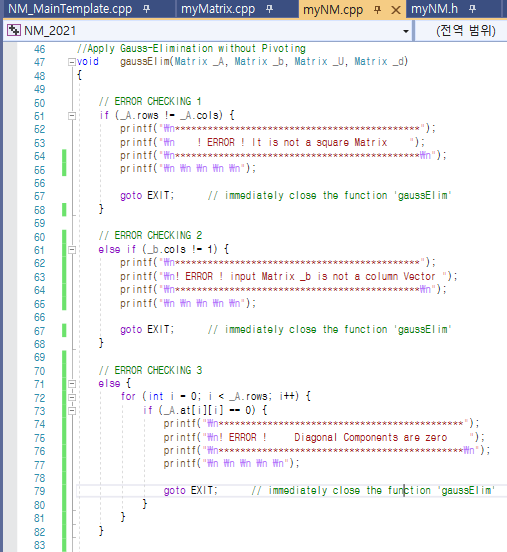


Figure 2.7. Error Checking Code

Gauss Elimination without Pivoting을 수행함에 있어서 정상적인 작동을 방해할 수 있는 총 3가지의 Error Case에 대해 위와 같이 처리해줄 수 있다.

goto EXIT 구문을 이용하여 Error임이 확인이 된다면 곧바로 gaussElim 함수를 빠져나갈 수 있도록 설정해주었다.

**1. Gauss elimination method without partial pivoting**

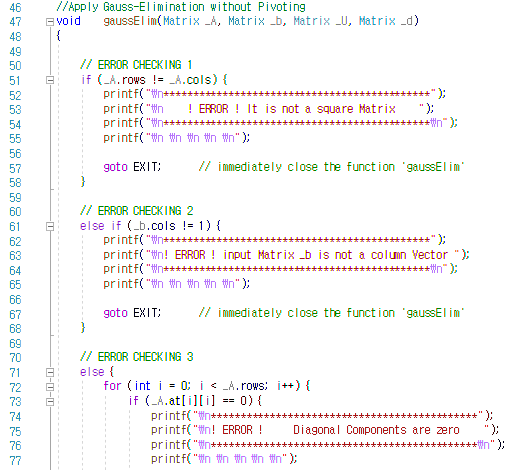
**First write a pseudocode**

Input = Matrix A Vector b

Output = Matrix U vector d

Matrix A의 크기는 M by N이라고 생각하고 pseudocode 작성.

**Show your code here**



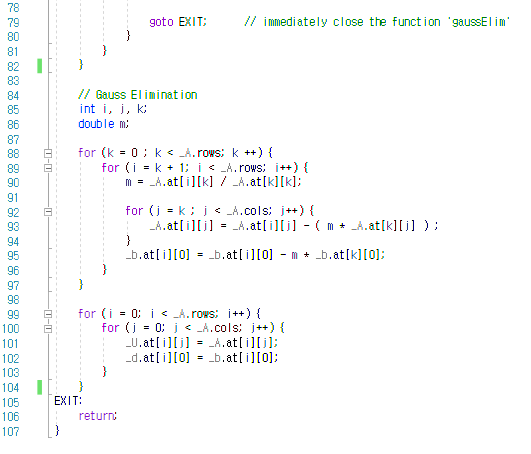


Figure 3.1. Code of Gauss elimination method without partial pivoting

Input으로 받은 행렬A와 벡터b가 Error Checking에서 통과하여 Gauss Elimination 부분에 들어오게 된다면, 위에서 작성한 Gauss Elimination의 pseudocode의 알고리즘이 작동하게 된다.

Gauss Elimination 과정을 거치면서 행렬A는 Upper Triangular Matrix U의 형태로 바뀌게 되고, 이 과정에서 벡터b도 값이 변하게 된다. main함수에서 미리 정의해둔 Matrix U와 Vector d에 각각의 변하게 된 값을 대입시켜주었다.

**2. Create back-substitution function to solve Ux=d**

**First write a pseudocode**

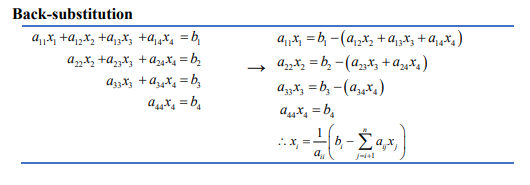


Figure 4.1. Back-substitution Algorithm

Input = Matrix U Vector d

Output = Vector X (=Final Solution)

Matrix U의 크기는 Matrix A와 동일한 M by N

참고: 2~4행은 부분을 코드로 나타낸 것이다.

**Show your code here**

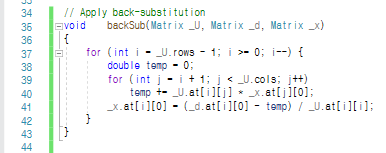


Figure 4.2. Code of Back-substitution

**3. Show the output results**

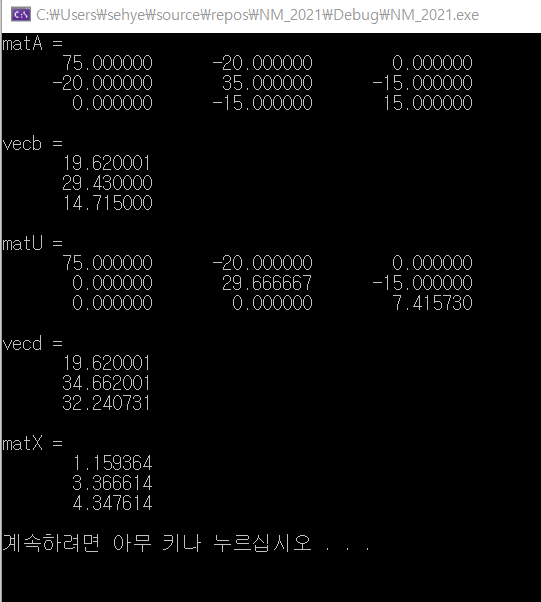


Figure 5.1. Output Results

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

|  |  |
| --- | --- |
|  |  |
| Figure 6.1. MATLAB Code | Figure 6.2. MATLAB Result |

Visual Studio를 통해 확인한 결과값과 MATLAB으로 확인한 결과값이 동일함을 볼 수 있다.