UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

Lab 1

Implementing a solver for systems of linear equations
Due to October 25th 23:45h

Julen Cayero, Cecilio Angulo

Bachelor's Degree in Video Game Design and Development

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

Outline

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

Outline

It wouldn't be too difficult to implement a naive solver using the Gauss method.
We will need 2 functions:

1. Triangulate a given matrix
2. Backtracking of a triangular matrix

However, in practice there are some problems associated with this way of proceeding related with:

1. Avoid handling large numbers
2. Identify critical cases (Indeterminate or incompatible system)
3. Proceed efficiently (we are not going in deep here)

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

Outline

Naïve Gaussian Elimination

Theory review

Gaussian elimination can be easily implemented to solve a system of equations in a simple two step procedure:

1. **Forward Elimination:** Given the matrix $\mathbf{A}$ and the independent term $\boldsymbol{b}$ as inputs produce a upper triangular matrix $\mathbf{A}_t$ and a independent term $\boldsymbol{b}_t$ such that

$$\mathbf{A}_t \boldsymbol{x} = \boldsymbol{b}_t \quad \text{and} \quad \mathbf{A}\boldsymbol{x} = \boldsymbol{b}$$

   are equivalent, i.e., share the same solution $\boldsymbol{x}$

2. **Back Substitution (a.k.a Backtracking)**: Given an upper triangular and **regular** matrix $\mathbf{A}_t$ and a vector $\boldsymbol{b}_t$ find the solution $\boldsymbol{x}$ such that

$$\mathbf{A}_t \boldsymbol{x} = \boldsymbol{b}_t$$

Naïve Gaussian Elimination

Theory review

Forward Elimination

Convert

$$\begin{array}{c}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n = b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3n}x_n = b_3 \\
\vdots \\
a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n = b_n
\end{array}
\quad \text{into} \quad
\begin{array}{r}
a_{11}x_1 + a_{12}x_2 + \cdots a_{1n}x_n = b_1 \\
a'_{22}x_2 + a'_{23}x_3 \cdots a'_{2n}x_n = b_2 \\
a'_{33}x_3 + \cdots a'_{3n}x_n = b_3 \\
\vdots \\
a'_{nn}x_n = b_n
\end{array}$$

Naïve Gaussian Elimination

Forward Elimination: First Step

The target is to make 0' under a_{11}

$$\left(\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & b_3 \\ \vdots & & \ddots & & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & b_n \end{array}\right)$$

Applying $\boldsymbol{r}_2 = a_{11}\boldsymbol{r}_2 - a_{21}\boldsymbol{r}_1$

$$\left(\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ 0 & a'_{22} & a'_{23} & \cdots & a'_{2n} & b'_2 \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & b_3 \\ \vdots & & \ddots & & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & b_n \end{array}\right)$$

generally applying

$$\boldsymbol{r}_j = a_{11}\boldsymbol{r}_j - a_{j1}\boldsymbol{r}_1 \quad \forall j > 1$$

$$\left(\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ 0 & a'_{22} & a'_{23} & \cdots & a'_{2n} & b'_2 \\ 0 & a'_{32} & a'_{33} & \cdots & a'_{3n} & b'_3 \\ \vdots & & \ddots & & & \vdots \\ 0 & a'_{n2} & a'_{n3} & \cdots & a'_{nn} & b'_n \end{array}\right)$$

Naïve Gaussian Elimination

Forward Elimination: next $n-1$ steps

The second row can be used to make 0's under a'_{22}, followed by the third row to make 0's under a''_{33} and so on.
To make 0's under the pivot a_{pp} located a row p you must modify the row j as:

$$\boldsymbol{r}_j = a_{pp}\boldsymbol{r}_j - a_{jp}\boldsymbol{r}_p \quad \forall j > p$$

Finally achieving

$$\left(\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ 0 & a'_{22} & a'_{23} & \cdots & a'_{2n} & b'_2 \\ 0 & 0 & a''_{33} & \cdots & a''_{3n} & b''_3 \\ \vdots & & \ddots & & & \vdots \\ 0 & 0 & 0 & \cdots & a^{n-1\prime}_{nn} & b^{n-1\prime}_{n} \end{array}\right)$$

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

Naïve Gaussian Elimination

Theory review

Back Substitution: First step

Solve the last equation

$$x_n = \frac{b_n}{a_{nn}}$$

Back Substitution: Next steps

Proceed visiting each row in decreasing order solving one equation at a time to find the associated unknown

$$x_i = \frac{b_i - \displaystyle\sum_{j=i+1}^{N} a_{ij} x_j}{a_{ii}}$$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} \\ 0 & 0 & a_{33} & \cdots & a_{3n} \\ \vdots & & \ddots & & \\ \vdots & & & & \\ 0 & 0 & 0 & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{pmatrix}$$

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

Outline

Handling large numbers
Problems

Computer does not like large numbers [1]. Try:

```
a = 1E99
b = 1E99 + 1
c = a-b
```

- Which value do you expect for c?
- What is the value of c that Python throws?

When triangulating, the operation

$$r_j = r_j Aa(p,p) - r_p Aa(j,p)$$

Do not control if the values of A are bounded.

[1] In case you're interested, there are special ways of dealing with those numbers, however it will be complex and computational expensive

Handling large numbers II

Things we can do:

1. Since we can multiply and divide every row by a non-zero number

$$r_j = r_j - r_p Aa(j,p)/Aa(p,p)$$

   However we must ensure that $A(p,p)$ is far enough from 0.

2. Since we can choose any row to triangulate, before making the 0's, we will swap the actual row for the row with the largest (absolute value) pivot and proceed as normal.

This is called partial pivoting

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

Division by 0

Problems

What is the result of applying the previous forward elimination method to solve the system:

$$x + 2y + 3z = 14$$
$$x + 2y - 2z = -1$$
$$y + 2z = 8$$

The solution is again to use the partial pivoting

Identify critical cases

Indeterminate or Incompatible systems of equations?

By maintaining the strategy of selecting the row with the largest pivot, we can check when the pivot ≈ 0 which leads to critical cases.

Set a **threshold** value (something small $\approx 1E-8$) and compare the pivot value against it. Consider to warn the user if you identify a very small value for the pivot.

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

Outline

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

- Always respect the delivery dates.
- Use the templates when provided and **DO NOT MODIFY THE FILE NAMES OR THE FUNCTION NAMES**
- Verify that your code works before submit.
- Always upload the files in a .zip with the name structure: Surname1_Name1_Surname2_Name2...**.zip**

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

Outline

At the end of the class you must deliver

1. A working function that performs the bactracking.
2. An script making two calls to your function to solve two different (Compatible and Determinate) systems solved in class. One of them must contain at least 4 equations (thus 4 unknowns).

Outline

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

Before October 25th at 23:45 you must deliver:

1. The notebook with the answers to the exercises
2. The `lab1_functions.py` file with your implementations