Department of Computer Science, UCF

COT 5405: Design and Analysis of Algorithms

Final Project: LU Decomposition via Web Front End

Presented by:
Rajib Dey

Introduction

- A matrix is a rectangular array of numbers arranged in rows and columns
- Decomposition is a process of breaking down a matrix into a product of two or more matrices
- The matrix decomposition which is expressing a matrix as a product of simpler matrices plays an important role in linear algebra and widely used for easy computations of solving Linear equations, image compressions and related Problems in statistics.

What are we trying to do?

- The aim of the project is to make a teaching aid for high school students about the matrix decomposition using LU algorithm based on fundamental steps.

- The code would be open source , readable and reliable.

- It would serve as a source of knowledge for students and those who are interested.

How is it done today?

- Nowadays it is implemented using Matlab, Mathematica, Symbolab, and other computing environments.
- The decomposition can be done by manual calculation.
- Matlab and Mathematica have various functions to return a decomposition of a given matrix that enables us to solve linear systems.

Syntax

```
dA = decomposition(A)
dA = decomposition(A,type)
dA = decomposition(A,type,triangularFlag)
dA = decomposition( __ ,Name,Value)
```

Compute the LU decomposition of a matrix:

```
In[1]:= {lu, p, c} = LUDecomposition[{{1, 2}, {3, 4}}]

Out[1]= {{{1, 2}, {3, -2}}, {1, 2}, 0}
```

What are the limits of current practice?

- The software mentioned earlier has abstract methods to compute the matrix decomposition and are not suitable for demonstration in academia.
- They do not have a heuristic method to pick the Matrix solver.
- Some of the software mentioned are not open source.
- Students need to know how to communicate with the software, but as they are only interested in decomposing a matrix, this is an overhead.

What is new in your approach?

- Our code is open source, reliable and platform independent.
- It also has an easy to use Graphics-User-Interface. It can be used for demonstration using Real-Time inputs.
- Our code is easy to read and does not contain obfuscation or any other kind of protection.
- No need to install any complicated softwares or learn a programming language.
- Using our tool students will actually learn how to perform LU Matrix Decomposition by themselves. As our tool shows every step of the calculation.

Who cares?

- Our tool is aimed at High-School students. It should make it easier for them to understand the steps that they need to follow to get a Lower and Upper triangular matrix from a given matrix.

- People who use computing systems to solve different problems based on the decomposition approach such as statistics, imaging or solving different equations can also get help from our tool.

What were the risks?

- Maybe students might not have the programming or math background to understand the whole process.

- Students may not have the knowledge of using the software required to run our program. Installing the software and using it may be a challenge

- Our tool might contain errors and might not cover all possible scenarios

Were the risks mitigated ?

- As our tool does not require the students to have programming background so we can safely say that we have mitigated that risk. But unfortunately the students will need basic knowledge in math to understand the steps being shown.

- As our tool is a website, so there is no need to install anything.

- It is evident that sometimes our tool will show errors like this. But it is very rare. According to our tests, it happens approximately 2 times out of 100 calculations.

The Upper Matrix is =

5	10	3
0	0.00	5.80
0	0	Infinity

The Lower Matrix is =

1	0	0
0.40	1.00	0
0.40	-Infinity	1.00

Mid-term outcomes - Pseudocode for the LU Algorithm

Input Matrix A

var L = []; var U = []; //This will become the L and U Matrix

function identity(N) {

this function would return a N-by-N identity Matrix

Where N= Length of Input matrix A}

L = identity(N); U = identity(N); // Get two identity matrix of size N and put in L and U

for (i=0; i<N; i++) { //Run for loop for the length of A

j = i;

while (j<N) { // Determine U across row i

q = 0; U[i][j] = A[i][j];

while (q<i) { U[i][j] -= L[i][q]*U[q][j]; q++;}

j++; } j = i+1;

while (j<N) { // Determine L down column I

q = 0; L[j][i] = A[j][i];

while (q<i) { L[j][i] -= L[j][q]*U[q][i]; q++;}

L[j][i] = L[j][i]/U[i][i]; j++;} }

return [L, U]; // This will return the L and U matrix

Mid-term outcomes – Checking Correctness

- We checked one result of our tool by doing manual calculation. It appears to be giving us incorrect results for one out of 9 numbers (for 3 by 3 matrix) after 1 decimal point.

- Which we can fix by taking more than 2 decimal points while Calculating the LU Decomposition.

- For the sake of simplicity we are taking only 2 decimal points. So that it is easier for the students to understand the whole process. For high precision calculations we obviously need to consider more decimal points.

Mid-term outcomes

- Our given matrix is

8	3	3
8	7	6
7	1	8

and the steps to calculate the L and U Matrix are

Upper Matrix Calculation

For the First step, we do [2nd Row - (8/8 x 1st row) ====> Row 2]

8	3	3
0	4.00	3.00 ⇐
7	1	8

For the 2nd step, we do [3rd Row - (7/8 x 1st row) ====> Row 3]

8	3	3
0	4.00	3.00
0	-1.63	5.38 ⇐

For the 3rd step, we do [3rd Row - (-0.41 x 2nd row) ====> Row 3]

8	3	3
0	4.00	3.00
0	0	6.59 ⇐

Lower Matrix Calculation

After the First step, we put 8/8 to Row 2, Column 1 Position

1	0	0
1.00	1	0 ⇐
0	0	1

After the 2nd step, we put 7/8 to Row 3, Column 1 Position

1	0	0
1.00	1	0
0.88	0	1 ⇐

After the 3rd step, we put -0.41 to Row 3, Column 2 Position

1	0	0
1.00	1	0
0.88	-0.41	1 ⇐

The final result is

The Upper Matrix is =

8	3	3
0	4.00	3.00
0	0	6.59

The Lower Matrix is =

1	0	0
1.00	1.00	0
0.88	-0.41	1.00

Mid-term outcomes

- Our manual calculation is shown below:

Lower matrix $= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0.88 & -0.41 & 1 \end{bmatrix}$

Upper matrix $= \begin{bmatrix} 8 & 3 & 3 \\ 0 & 4 & 3 \\ 0 & 0 & 6.59 \end{bmatrix}$

multiplying Lower and upper matrix $\Rightarrow$

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0.88 & -0.41 & 1 \end{bmatrix} \times \begin{bmatrix} 8 & 3 & 3 \\ 0 & 4 & 3 \\ 0 & 0 & 6.59 \end{bmatrix}$$

$$= \begin{bmatrix} 1\times8+0\times0+0\times0 & 1\times3+0\times4+0\times0 & 1\times3+0\times3+0\times6.59 \\ 1\times8+1\times0+0\times0 & 1\times3+1\times4+0\times0 & 1\times3+1\times3+0\times6.59 \\ 0.88\times8+(-0.41\times0)+1\times0 & 0.88\times3+(-0.41\times4)+1\times0 & 0.88\times3+(-0.41\times3)+1\times6.59 \end{bmatrix}$$

$$= \begin{bmatrix} 8 & 3 & 3 \\ 8 & 7 & 6 \\ 7.04 & 1 & 8 \end{bmatrix}$$

Final Outcomes

- Our final exam was to develop an GUI so that the students can easily input their matrices and get them decomposed into L and U matrix.

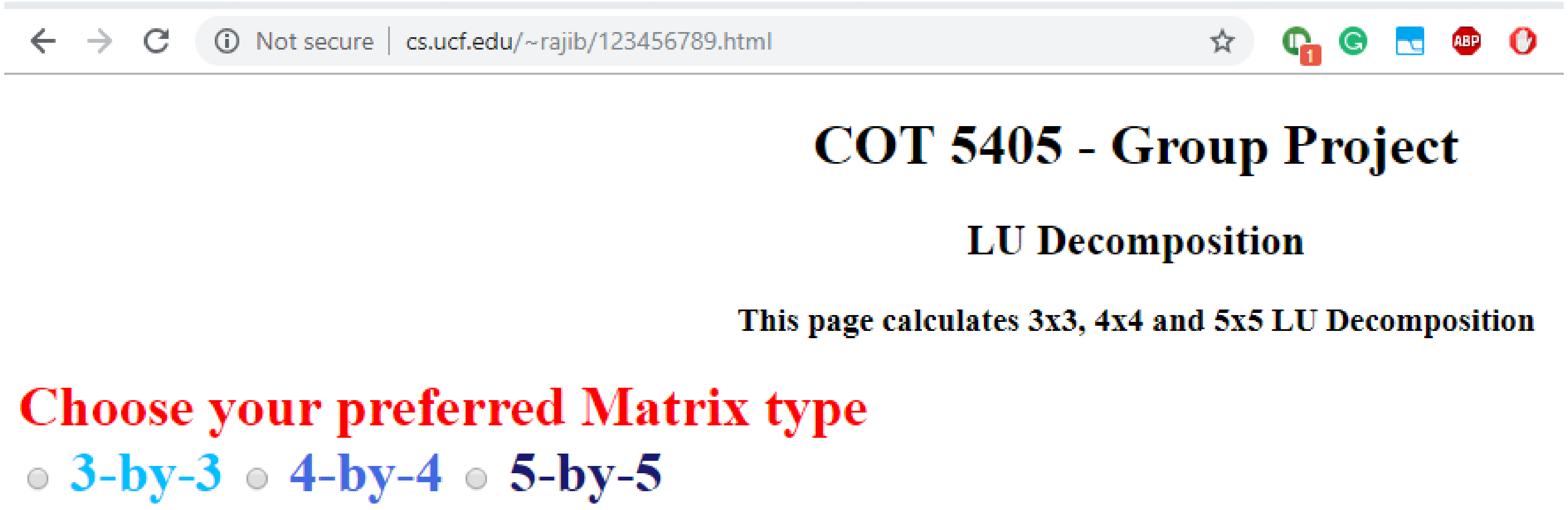
- We have been able to do that successfully.

- Which would be visible if you take a look at the attached HTML file or go to our website. Which is temporarily located at

  http://cs.ucf.edu/~rajib/123456789.html

Final Outcomes

- When you go to our website, you will be able to choose between 3 different matrix types



Final Outcomes

- Depending on which one you chose in the previous step, you can now either use a random matrix or manually enter a matrix in the input box.
- For a randomly generated matrix, each individual member of the matrix will be randomly assigned a number between 1 and 10. We are not considering huge numbers as an input here for the sake of simplicity.
- If you want to enter the matrix manually then you can enter larger numbers.