

PHYS 5794 Homework 3

Thomas Edwards

February 16, 2016

1 Problem 1

1.1 Problem Statement

Write a program to solve the following linear algebraic equations using the **LU** decomposition method with implicit partial pivoting. Compare your numerical result with the answer, $x = 224/1545$, $y = -268/1545$, $z = 83/515$, and $u = 589/1545$. Don't forget carrying out the same row permutations (for pivoting) to the right-hand side vector **b**. This step is important for obtaining the solution. In the report, the following six issues should be addressed. (i) Write down the constructed lower **L** and upper triangular matrices **U**. (ii) Check if $\mathbf{L} \cdot \mathbf{U} = \mathbf{A}$. If you have to carry out row permutations or operations, one needs to check if $\mathbf{L} \cdot \mathbf{U} = \mathbf{P} \cdot \mathbf{A}$, where **P** is a permutation matrix. You may use the 3×3 matrix example used in the class or other simple examples for debugging. (iii) Write down a permutation matrix **P** if it was used. (iv) Write down the solution $\mathbf{x} = (x, y, z, u)$. (v) Confirm if your numerical solution satisfies the given answer. (vi) Now turn off the pivoting procedure and see if you obtain the same answer with the pivoting. If so, compare the two answers up to seventh digits. If not, explain the reason. (30 pts)

1.2 Method

The LU decomposition for this homework problem was done using the Dolittle method (the name of which was referenced in the textbook), as described in class. This includes both partial pivoting and non-pivoting methods.

The method takes some square, $n \times n$ matrix **A** and decomposes it in to a lower-triangular matrix **L** and upper-triangular matrix **U**. From these, the solution for **x** may be found given that $\mathbf{Ax} = \mathbf{b}$. Before the decomposition, and optional pivoting procedure can be used to do row swapping in both **A** and **b** to make the diagonal elements of the matrix the largest. This also changed rows of a pivoting matrix **P**, which can be used later to develop final solutions.

L and **U** are found by an iterative relationship, in the following structure:

$$\begin{aligned} &\text{For } i = [0, 1, \dots, n-1] : \\ &U_{i,j} = A_{i,j} - \sum_{k=0}^{i-1} L_{i,k} U_{k,j}, \text{ for } j = [i, i+1, \dots, n-1] \\ &L_{j,i} = \frac{A_{j,i} - \sum_{k=i+1}^{n-1} L_{j,k} U_{k,i}}{U_{i,i}}, \text{ for } j = [i+1, i+2, \dots, n-1] \end{aligned}$$

This solves for a specific row of **U**, and then uses those solutions to find a specific column of **L**, starting from the top left corner and going across and down.

It is important to note that this particular description of the indices is particular to the code written, as the beginning of the matrices are indexed at 0 instead of 1. In addition, some elements are either initialized ahead of time (in the case of **L**) or are done by exploiting looping relationships, whereas the notes suggest that the first row and column of **U** and **L** (respectively) must be done separately.

To finally solve for **x**, we first solve $\mathbf{Ly} = \mathbf{b}$, and then solve $\mathbf{Ux} = \mathbf{y}$. To solve for **y**, we use the following:

$$\begin{aligned} y_0 &= \frac{b_0}{L_{00}} \\ &\text{For } i = [1, \dots, n-1] : \\ y_i &= \frac{b_i - \sum_{k=0}^{i-1} y_k L_{i,k}}{L_{i,i}} \end{aligned}$$

We then solve for **x** similarly, but in reverse:

$$\begin{aligned} x_n &= \frac{y_n}{U_{n,n}} \\ &\text{For } i = [n-2, n-1, \dots, 0] : \end{aligned}$$

$$x_i = \frac{y_i - \sum_{k=i+1}^{n-1} x_k U_{i,k}}{U_{i,i}}$$

It is important to note that x_n is solved first, and then x_{n-1}, x_{n-2}, \dots etc. This is primarily due to the nature of how the code is implemented and the way that the matrices are indexed.

1.3 Verification of Program

To verify this program, a simple test case from the lecture notes was used. The results of this verification are below. It should be noted that the example in the lecture notes did not require pivoting to get the results, and was solved here using both pivoting and non-pivoting methods to verify that the program was working correctly.

```
-----
A simple test case, using pivoting
-----
*****
L and U
*****
L =
[[ 1.          0.          0.          ]
 [ 0.66666667  1.          0.          ]
 [ 0.33333333  0.2         1.          ]]
U =
[[ 3.          2.          1.          ]
 [ 0.          1.66666667  0.33333333]
 [ 0.          0.          3.6         ]]
*****
LU, PA, and Original A for Comparison
*****
LU =
[[ 3.  2.  1.]
 [ 2.  3.  1.]
 [ 1.  1.  4.]]
PA =
[[ 3.  2.  1.]
 [ 2.  3.  1.]
 [ 1.  1.  4.]]
A (Original) =
[[ 1.  1.  4.]
 [ 3.  2.  1.]
 [ 2.  3.  1.]]
*****
P
*****
P =
[[ 0.  1.  0.]
 [ 0.  0.  1.]
 [ 1.  0.  0.]]
*****
y
*****
y =
[ 11.          5.66666667  7.2         ]
*****
```

```

                                x
*****
x (solution) =
[ 1.  3.  2.]
*****
      Comparison to Known Solution
*****
[ 1.  3.  2.]

```

```

-----
      Same test case, without pivoting
-----
*****
                        L and U
*****
L =
[[ 1.          0.          0.          ]
 [ 0.66666667  1.          0.          ]
 [ 0.33333333  0.2         1.          ]]
U =
[[ 3.          2.          1.          ]
 [ 0.          1.66666667  0.33333333]
 [ 0.          0.          3.6         ]]
*****
      LU, PA, and Original A for Comparison
*****
LU =
[[ 3.  2.  1.]
 [ 2.  3.  1.]
 [ 1.  1.  4.]]
PA =
[[ 3.  2.  1.]
 [ 2.  3.  1.]
 [ 1.  1.  4.]]
A (Original) =
[[ 3.  2.  1.]
 [ 2.  3.  1.]
 [ 1.  1.  4.]]

```

```

*****
                                y
*****
y =
[ 11.          5.66666667  7.2         ]
*****
                                x
*****
x (solution) =
[ 1.  3.  2.]
*****
      Comparison to Known Solution
*****

```

[1. 3. 2.]

As we can see from the results above, the solutions are the same as the lectures notes, except some of the terms have been divided by a constant. This is consistent across **L**, **U**, and **b**, so the solution in the end is retained. It also shows that the pivoting method both works and returns the expected result.

1.4 Data

The solution for the problem statement at hand is below.

```
-----
                Problem Statement Solution
-----
*****
                L and U
*****
L =
[[ 1.          0.          0.          0.          ]
 [ 0.66666667  1.          0.          0.          ]
 [ 0.13333333  0.2         1.          0.          ]
 [-0.26666667  0.01176471  0.23821443  1.          ]]
U =
[[ 15.         -3.          1.          3.          ]
 [ 0.          17.         2.33333333  5.          ]
 [ 0.          0.          9.4         -2.4         ]
 [ 0.          0.          0.          10.31289111]]
*****
                LU, PA, and Original A for Comparison
*****
LU =
[[ 15.  -3.   1.   3.]
 [ 10.  15.   3.   7.]
 [ 2.   3.  10.  -1.]
 [-4.   1.   2.   9.]]
PA =
[[ 15.  -3.   1.   3.]
 [ 10.  15.   3.   7.]
 [ 2.   3.  10.  -1.]
 [-4.   1.   2.   9.]]
A (Original) =
[[ 2.   3.  10.  -1.]
 [ 10.  15.   3.   7.]
 [-4.   1.   2.   9.]
 [ 15.  -3.   1.   3.]]
*****
                P
*****
P =
[[ 0.  0.  0.  1.]
 [ 0.  1.  0.  0.]
 [ 1.  0.  0.  0.]
 [ 0.  0.  1.  0.]]
*****
                y
```

```

*****
y =
[ 4.          -0.66666667  0.6          3.93158114]
*****

x
*****
x (solution) =
[ 0.14498382 -0.17346278  0.16116505  0.38122977]
*****
Comparison to Known Solution
*****
[ 0.14498382 -0.17346278  0.16116505  0.38122977]

-----
Problem Statement Solution (no pivot)
-----
*****
L and U
*****
L =
[[ 1.          0.          0.          0.          ]
 [ 0.66666667  1.          0.          0.          ]
 [ 0.13333333  0.2         1.          0.          ]
 [-0.26666667  0.01176471  0.23821443  1.          ]]
U =
[[ 15.          -3.          1.          3.          ]
 [ 0.          17.          2.33333333  5.          ]
 [ 0.          0.          9.4         -2.4         ]
 [ 0.          0.          0.          10.31289111]]
*****
LU, PA, and Original A for Comparison
*****
LU =
[[ 15.  -3.   1.   3.]
 [ 10.  15.   3.   7.]
 [ 2.   3.  10.  -1.]
 [-4.   1.   2.   9.]]
PA =
[[ 15.  -3.   1.   3.]
 [ 10.  15.   3.   7.]
 [ 2.   3.  10.  -1.]
 [-4.   1.   2.   9.]]
A (Original) =
[[ 15.  -3.   1.   3.]
 [ 10.  15.   3.   7.]
 [ 2.   3.  10.  -1.]
 [-4.   1.   2.   9.]]
*****
y
*****
y =

```

```

[ 4.          -0.66666667  0.6          3.93158114]
*****
                                x
*****
x (solution) =
[ 0.14498382 -0.17346278  0.16116505  0.38122977]
*****
      Comparison to Known Solution
*****
[ 0.14498382 -0.17346278  0.16116505  0.38122977]

```

1.5 Analysis

The method runs fairly quickly, but it should be noted that the cases for this homework were fairly small, simple ones. In comparison to the Gaussian Elimination, the method is actually much easier to develop, as the reduction of terms does not need to be done, and can instead be handled iteratively when decomposing \mathbf{L} and \mathbf{U} . The numerical accuracy also appears to be very consistent. This is, of course, partially due to the specific cases used within the program, as a full \mathbf{LU} decomposition is not possible for all cases. This distinction, however, is somewhat outside what was asked for in this homework, and was not included.

1.6 Interpretation

As we can see, the results from both the verification and the actual problem are as expected. $\mathbf{LU} = \mathbf{PA}$ whenever pivoting is used, and both \mathbf{L} and \mathbf{U} have the expected form, including \mathbf{L} being 1 on the diagonal. The solutions, with or without pivoting, both give the results we expect out to 8 decimal places (when appropriate). The intermediate \mathbf{y} is also included for debugging, and to show the process as it calculates.

1.7 Critique

It should be noted that this program is actually pretty fragile. Since these specific test cases were simple and known to behave fairly well, we can skip most of the work of verifying that the input is acceptable and get on with the solution.

1.8 Log

In total, this problem took about 7.5 hours.