Numerical Analysis

Solving linear systems

Introduction

In "numerical linear algebra" we will consider linear systems such as:

$$\begin{cases} 3x_1 & + & 4x_2 & - & 2x_3 & = & 5 \\ 10x_1 & + & 2x_2 & + & x_3 & = & 15 \\ 1x_1 & + & x_2 & + & x_3 & = & 1 \end{cases}$$

We will use matrix notation: this system is written as $A\mathbf{x} = \mathbf{b}$ where

$$A = \begin{bmatrix} 3 & 4 & -2 \\ 10 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 5 \\ 15 \\ 1 \end{bmatrix}$$

Introduction

In this course we will consider matrices $A \in \mathbb{C}^{m\times n}$, or $A \in \mathbb{R}^{m\times n}$, and will denote their entries by

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{m1} & a_{m2} & & a_{mn} \end{bmatrix}$$

A vector $\mathbf{x} \in \mathbb{C}^n$ or $\mathbf{x} \in \mathbb{R}^n$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Methods for solving linear systems

1. **Direct Methods:** Gaussian elimination and LU decomposition.
   In these methods we perform an algorithm, that would have got the exact solution if there were no round-off errors. Since there are round-off errors, we may get significant errors in the solution if we are not careful.

2. **Iterative Methods**
   These methods will work using iterations and will provide an approximate solution that gets improved as we do more iterations. The solution will be approximate regardless of round-off errors. In many applications, direct solution is much more expensive than iterative solution.

Existence of the solution

The existence of a solution. We know that the system $A\mathbf{x} = \mathbf{b}$ above has either of the following:

1. *A unique solution.* In this case the columns or rows of the matrix A are linearly independent, and the matrix is non-singular. In this case the matrix is also said to be of "full-rank".

2. *No solution*–the vector $\mathbf{b}$ cannot be expressed as a linear combination of the columns of A. In this case there must be at least one column which is linearly dependent with the others.

3. *Infinite number of solutions.* In this case, the vector $\mathbf{b}$ can be expressed as a linear combination of the columns of A, but there must be at least one column which is linearly dependent with the others. That means that there is a non-trivial vector $\mathbf{e}$ such that $A\mathbf{e} = 0$, and the matrix is called singular.

Direct solutions – Gaussian elimination

We will assume that the matrix A is of full rank (non-singular) and that a solution to the system exists.

Consider the following linear system

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Our aim is to use elementary operations and transform the system $A\mathbf{x} = \mathbf{b}$ to a system $U\mathbf{x} = \hat{\mathbf{b}}$ that has the same solution $\mathbf{x}$. The matrix U will be upper-triangular: such that

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

Backward Substitution

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \hat{b_1} \\ \hat{b_2} \\ \hat{b_3} \end{bmatrix}$$

Now we can solve the triangular system by solving the 1-variable equation for x_3, then for x_2 using x_3, and then for x_1. This process of solving an upper triangular system is called **Backward Substitution** in which we start solving from the last variable x_n all the way to the first one x_1. That is:

$$x_i = \frac{1}{u_{ii}} \left(\hat{b}_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) \quad i = n, n-1, ..., 1.$$

Example

Example 1. *Solve the following linear system using 4 significant digits using cut-off rounding.*

$$\begin{cases} 3x_1 & + & -x_2 & + & 2x_3 & = & 12 \\ 1x_1 & + & 2x_2 & + & 3x_3 & = & 11 \\ 2x_1 & + & -2x_2 & + & -x_3 & = & 2 \end{cases}$$

$$\begin{bmatrix} 3 & -1 & 2 \\ 1 & 2 & 3 \\ 2 & -2 & -1 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 11 \\ 2 \end{bmatrix}$$

Example

At the first iteration, the pivot will be 3:

$$\begin{bmatrix} \mathit{3} & -1 & 2 \\ 1 & 2 & 3 \\ 2 & -2 & -1 \end{bmatrix}, \begin{bmatrix} 12 \\ 11 \\ 2 \end{bmatrix} \quad \begin{matrix} \Rightarrow \\ \mathbf{r}_2 \leftarrow \mathbf{r}_2 - \frac{1}{3}\mathbf{r}_1 \\ \mathbf{r}_3 \leftarrow \mathbf{r}_3 - \frac{2}{3}\mathbf{r}_1 \end{matrix} \quad \begin{bmatrix} 3 & -1 & 2 \\ 0 & 2.333 & 2.334 \\ 0 & -1.334 & -2.332 \end{bmatrix}, \begin{bmatrix} 12 \\ 7.004 \\ -5.992 \end{bmatrix}$$

At the second iteration the pivot will be 2.333:

$$\begin{bmatrix} 3 & -1 & 2 \\ 0 & \mathit{2.333} & 2.334 \\ 0 & -1.334 & -2.332 \end{bmatrix}, \begin{bmatrix} 12 \\ 7.004 \\ -5.992 \end{bmatrix} \quad \begin{matrix} \Rightarrow \\ \\ \mathbf{r}_3 \leftarrow \mathbf{r}_3 - \frac{-1.334}{2.333}\mathbf{r}_2 \end{matrix} \quad \begin{bmatrix} 3 & -1 & 2 \\ 0 & 2.333 & 2.334 \\ 0 & 0 & -1.000 \end{bmatrix}, \begin{bmatrix} 12 \\ 7.004 \\ -1.993 \end{bmatrix}$$

Example

From here, we apply the backward substitution algorithm for $U\mathbf{x} = \hat{\mathbf{b}}$

$$U = \begin{bmatrix} 3 & -1 & 2 \\ 0 & 2.333 & 2.334 \\ 0 & 0 & -1.000 \end{bmatrix}, \hat{\mathbf{b}} = \begin{bmatrix} 12 \\ 7.004 \\ -1.993 \end{bmatrix} \Rightarrow \mathbf{x} = \begin{bmatrix} 3.007 \\ 1.008 \\ -1.993 \end{bmatrix}$$

$$\textit{where} \quad \mathbf{x}_{exact} = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$$

We got a pretty good approximation of the solution.

Algorithm 1 – Gaussian Elimination

Algorithm: Gaussian Elimination

$\# \; A \in \mathbb{C}^{n \times n}$

Initialize: $U \leftarrow A$, $\hat{\mathbf{b}} \leftarrow \mathbf{b}$.

for $k = 1, \ldots, n-1$ **do**

  for $i = k+1, \ldots, n$ **do**

    $t = \frac{u_{i,k}}{u_{k,k}}$.

    $\#$ *In the next loop we do* $\mathbf{r}_i \leftarrow \mathbf{r}_i - t \cdot \mathbf{r}_k$

    for $j = k, \ldots, n$ **do**

      $u_{i,j} \leftarrow u_{i,j} - t \cdot u_{k,j}$.

    end

    $\hat{b}_i \leftarrow \hat{b}_i - t \cdot \hat{b}_k$

  end

end

Complexity

Now we wish to count the number of multiplications and divisions in GE. In the inner most loop we have one multiplication, and in the second most inner loop we have an additional two. Overall we have

$$
\begin{aligned}
Cost(GE) &= \sum_{k=1}^{n-1}\sum_{i=k+1}^{n}(2+\sum_{j=k}^{n}1) = \sum_{k=1}^{n-1}\sum_{i=k+1}^{n}(2+(n-k+1)) = \sum_{k=1}^{n-1}\sum_{i=k+1}^{n}(n-k+3) = \\
&= \sum_{k=1}^{n-1}(n-k)(n-k+3) = \sum_{p=1}^{n-1}p(p+3) = \cdots \approx \frac{1}{3}n^3 = O(n^3)
\end{aligned}
$$

In the backward substitution stage we have i multiplications for row $n-i+1$

$$
Cost(BS) = \sum_{i=1}^{n} i = \frac{1+n}{2}n \approx \frac{1}{2}n^2 = O(n^2)
$$

Overall for solving the linear system we get $O(n^3)$.

LU decomposition

$$A = LU = \begin{bmatrix} 3 & -1 & 2 \\ 1 & 2 & 3 \\ 2 & -2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ \frac{2}{3} & -\frac{4}{7} & 1 \end{bmatrix} \cdot \begin{bmatrix} 3 & -1 & 2 \\ 0 & \frac{7}{3} & \frac{7}{3} \\ 0 & 0 & -1 \end{bmatrix}.$$

LU decomposition without pivoting

The Gaussian elimination in the previous section can be used to get a powerful tool: LU decomposition. It turns out the the GE process can easily yield a decomposition of A into a product of lower and upper triangular matrices

$$A = LU,$$

where U is the same matrix that is computed in GE, and L is the lower triangular matrix that is decomposed out of the t variables in Alg. 1. That is, in our example

$$L = \begin{bmatrix} 1 & 0 & 0 \\ \frac{u_{2,1}^{(1)}}{u_{1,1}^{(1)}} & 1 & 0 \\ \frac{u_{3,1}^{(1)}}{u_{1,1}^{(1)}} & \frac{u_{3,2}^{(2)}}{u_{2,2}^{(2)}} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0.333 & 1 & 0 \\ 0.666 & \frac{-1.334}{2.333} & 1 \end{bmatrix}$$

where $u_{i,j}^{(k)}$ is the i, j-th coefficient of U in the k-th iteration.

Algorithm 1 – LU decomposition without pivoting

Algorithm: LU

$\# \; A \in \mathbb{C}^{n \times n}$

Initialize: $U = A$. $L = I$.

for $k = 1, \ldots, n-1$ **do**

  for $i = k+1, \ldots, n$ **do**

    $l_{i,k} = \frac{u_{i,k}}{u_{k,k}}$.

    for $j = k, \ldots, n$ **do**

      $u_{i,j} = u_{i,j} - l_{i,k} u_{k,j}$.

    end

  end

end

running time complexity is $O(n^3)$ and its space complexity is $O(n^2)$

LU decomposition advantages

The advantage of having a decomposition: What did we gain out of this? Using the LU decompisition of A, we can solve any linear system $A\mathbf{x} = \mathbf{b}$ easily by two triangular matrix inversions

$$A\mathbf{x} = \mathbf{b} \Rightarrow LU\mathbf{x} = \mathbf{b} \Rightarrow_{\mathbf{y}=U\mathbf{x}} L\mathbf{y} = \mathbf{b}.$$

First by "forward substitution" we solve the lower triangular system

$$\mathbf{y} = L^{-1}\mathbf{b},$$

then, using backward substitution we solve the upper triangular system

$$U\mathbf{x} = \mathbf{y} \Rightarrow \mathbf{x} = U^{-1}\mathbf{y},$$

by "backward substitution" in $O(n^2)$ running time. This is most efficient when we have multiple linear systems to solve with the same matrix A but each time for a different right-hand-sides $\mathbf{b}$.

Calculation of the inverse of A

In particular, the inverse of a matrix can be computed by solving $AX = I$

$$\begin{aligned} Y &= L^{-1}I \\ X &= U^{-1}Y \end{aligned}$$

and the result is $X = A^{-1}$. The process is computed in $O(n^3)$

Partial pivoting

Problem 1

Motivation The process in Algorithm 2 may fail if one of the pivots $u_{i,i}$ becomes zero. For example, if $a_{11} = 0$. Consider the following case.

$$A\mathbf{x} = \mathbf{b} : \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

The naive LU algorithm will divide a_{21} by zero in the first step and an error will occur, even though this matrix is invertible. There is a solution, though. The obvious thing to do is to replace the first row with second one that doesn't have a 0 in the first entry. In other words, we should pick a new pivot by permuting either the rows or the columns of the matrix.

Problem 2

Let us look at a similar case where there is no division by zero. Consider the following case where $\epsilon \approx 0$ but $\epsilon \neq 0$.

$$A\mathbf{x} = \mathbf{b} : \begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

The exact solution is $x_1 = \frac{1}{1-\epsilon} \approx 1 + \epsilon$, and $x_2 = 1 - \frac{\epsilon}{1-\epsilon} \approx 1 - \epsilon$.

Applying the GE process will yield

$$\begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad \underset{\mathbf{r}_2 \leftarrow \mathbf{r}_2 - \frac{1}{\epsilon}\mathbf{r}_1}{\Rightarrow} \quad \begin{bmatrix} \epsilon & 1 \\ 0 & 1 - \frac{1}{\epsilon} \end{bmatrix}, \begin{bmatrix} 1 \\ 2 - \frac{1}{\epsilon} \end{bmatrix}$$

and then $x_2 = \frac{2-\epsilon^{-1}}{1-\epsilon^{-1}} \approx 1$, and $x_1 = \frac{1-x_2}{\epsilon}$.

In the computation for x_1 we will get a severe loss of significance and we will loose accuracy.

Example 2. *Solve the following system using Gaussian Elimination using 4 significant digits of accuracy*

$$A\mathbf{x} = \mathbf{b} : \begin{bmatrix} 1 & 999 \\ 333 & -212 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 7096 \\ -472.9 \end{bmatrix}.$$

The exact solution is $x_1 = 3.1$, *and* $x_2 = 7.1$.

The first step of Gaussian Elimination will

$$\begin{bmatrix} 1 & 999 \\ 333 & -212 \end{bmatrix}, \begin{bmatrix} 7096 \\ -472.9 \end{bmatrix} \underset{\mathbf{r}_2 \leftarrow \mathbf{r}_2 - 333\mathbf{r}_1}{\Rightarrow} \begin{bmatrix} 1 & 999 \\ 0 & -332900 \end{bmatrix}, \begin{bmatrix} 7096 \\ -2363000 \end{bmatrix}$$

That is because, for example, to get $u_{2,2}$

$$round(-212 - round(333 * 999)) = round(-212 - round(332667)) =$$

$$= round(-212 - 332700) = -332900.$$

We get $x_2 = 7.098$, *which is pretty accurate, but*

$$x_1 = 7096 - 999x_2 = 7096 - 7091 = 5,$$

which is far from the right answer 3.1 (60% error).

Partial pivoting

Partial pivoting In partial pivoting we change the order of the rows such that the pivot that we choose is maximal in absolute value. In our case, the maximum between $a_{11} = 1$ and $a_{13} = 333$ is 333. Meaning, we will replace the first row with the second.

$$\begin{bmatrix} 333 & -212 \\ 1 & 999 \end{bmatrix}, \begin{bmatrix} -472.9 \\ 7096 \end{bmatrix} \quad \underset{\mathbf{r}_2 \leftarrow \mathbf{r}_2 - \frac{1}{333}\mathbf{r}_1}{\Rightarrow} \quad \begin{bmatrix} 333 & -212 \\ 0 & 999.6 \end{bmatrix}, \begin{bmatrix} -472.9 \\ 7097 \end{bmatrix}$$

$$x_2 = \frac{7096}{999.7} = 7.1$$

$$x_1 = \frac{-472.9 + 212 \cdot 7.1}{333} = \ldots = 3.099$$

LU factorization with partial pivoting

Suppose that we encounter a case where the pivot is zero, like

$$A = \begin{bmatrix} 0 & 4 & 2 \\ 10 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

we can permute the rows and get away with the division by 0. We will choose the best pivot (largest in magnitude) in every iteration. But - we need a way to keep track of these changes. For this, we will use "permutation matrices".

LU factorization with partial pivoting

A permutation matrix P is a matrix that has exactly one non-zero entry of 1.0 in every row and column. It replaces the locations of the entries of a vector. It For example the matrix

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

replaces the first and second entries, and in this case,

$$PA = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 4 & 2 \\ 10 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 10 & 2 & 1 \\ 0 & 4 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

matrix can also be denoted by a permutation vector $\mathbf{p} = [2, 1, 3]$

Theorem – LU decomposition

Let $A \in \mathbb{C}^{n \times n}$, *then there exists a permutation matrix* P *s.t*

$$PA = LU,$$

where L *and* U *are lower and upper triangular matrices respectively.*

LU decomposition with partial pivoting -example

Init.:

$$A, U = \begin{bmatrix} 3 & 0 & 2 \\ -10 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

1st permut.:

$$U = \begin{bmatrix} -10 & 0 & 1 \\ 3 & 0 & 2 \\ 1 & 1 & 1 \end{bmatrix} \quad L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}$$

LU decomposition with partial pivoting -example

$$U = \begin{bmatrix} -10 & 0 & 1 \\ 3 & 0 & 2 \\ 1 & 1 & 1 \end{bmatrix} \quad L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}$$

Then, we will apply $\mathbf{r}_2 \leftarrow \mathbf{r}_2 + 0.3\mathbf{r}_1$ *and* $r_3 \leftarrow \mathbf{r}_3 + 0.1\mathbf{r}_1$.

$$1^{st} \textit{ outer it.:}$$

$$U = \begin{bmatrix} -10 & 0 & 1 \\ 0 & 0 & 2.3 \\ 0 & 1 & 1.1 \end{bmatrix} \quad L = \begin{bmatrix} 1 & 0 & 0 \\ -0.3 & 1 & 0 \\ -0.1 & 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}$$

LU decomposition with partial pivoting -example

$$U = \begin{bmatrix} -10 & 0 & 1 \\ 0 & 0 & 2.3 \\ 0 & 1 & 1.1 \end{bmatrix} \quad L = \begin{bmatrix} 1 & 0 & 0 \\ -0.3 & 1 & 0 \\ -0.1 & 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}$$

$$2^{nd}\ permut.:$$

$$U = \begin{bmatrix} -10 & 0 & 1 \\ 0 & 1 & 1.1 \\ 0 & 0 & 2.3 \end{bmatrix} \quad L = \begin{bmatrix} 1 & 0 & 0 \\ -0.1 & 1 & 0 \\ -0.3 & 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$$

Note that the 1's in the diagonal of the L matrix are not permuted—we fix them throughout the whole algorithm.

LU decomposition with partial pivoting -example

$$U = \begin{bmatrix} -10 & 0 & 1 \\ 0 & 1 & 1.1 \\ 0 & 0 & 2.3 \end{bmatrix} \quad L = \begin{bmatrix} 1 & 0 & 0 \\ -0.1 & 1 & 0 \\ -0.3 & 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$$

After the permutation we also get the result of the second iteration

$$2^{nd} \text{ outer it:}$$

$$U = \begin{bmatrix} -10 & 0 & 1 \\ 0 & 1 & 1.1 \\ 0 & 0 & 2.3 \end{bmatrix} \quad L = \begin{bmatrix} 1 & 0 & 0 \\ -0.1 & 1 & 0 \\ -0.3 & 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad \mathbf{p} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$$

LU decomposition with partial pivoting -example

At the end of the algorithm we have:

$$PA = \begin{bmatrix} -10 & 0 & 1 \\ 1 & 1 & 1 \\ 3 & 0 & 2 \end{bmatrix} \quad LU = \begin{bmatrix} -10 & 0 & 1 \\ 1 & 1 & 1 \\ 3 & 0 & 2 \end{bmatrix} \quad P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Solving linear system with permutated LU

Solving a linear system with the permutated LU: Note that any permutation matrix P is also an orthogonal matrix, that is $P^T P = I$ (or $P^{-1} = P^T$). So, we can solve any linear system $A\mathbf{x} = \mathbf{b}$ easily by writing

$$PA = LU \Rightarrow A\mathbf{x} = P^T LU\mathbf{x} = \mathbf{b} \Rightarrow_{\mathbf{y}=U\mathbf{x}} L\mathbf{y} = P\mathbf{b}.$$

and performing one permutation and two triangular matrix inversions

$$\begin{aligned} L\mathbf{y} = P\mathbf{b} \Rightarrow \mathbf{y} &= L^{-1}(P\mathbf{b}) \\ U\mathbf{x} = \mathbf{y} \Rightarrow \mathbf{x} &= U^{-1}\mathbf{y}. \end{aligned}$$

Other roles of LU decomposition

The LU factorization plays an important role in calculating the determinant of a matrix or determining whether it is singular or not. It is known that for any two matrices A, B

$$\det(AB) = \det(A)\det(B).$$

Hence, if $PA = LU$ then

$$\det(PA) = \det(L)\det(U).$$

It is easy to show that for such LU decomposition we have $\det(P) = \pm 1$, $\det(U) = \Pi_i u_{ii}$, $\det(L) = \Pi_i l_{ii} = 1$, so $\det(A)$ can be easily computed. Similarly, if U has a zero diagonal entry - then A is singular.