Solving Linear Equation Systems

Neculai Andrei[1]

Research Institute for Informatics,
Center for Advanced Modeling and Optimization,
8-10 Averescu Avenue, Sector 1, Bucharest, Romania

June 18, 2016

This appendix concentrates on methods for solving systems of linear equations

$$Ax = b,$$

where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. This is a basic problem that arises in many optimization algorithms and is crucial in the efficiency of the algorithms. Assume that A is nonsingular, so the solution is unique for all vectors b and is given by $x = A^{-1}b.$ The matrix A is often called the *coefficient matrix* and the vector b is called the *right-hand side* term. Firstly we present some cases for which $Ax = b$ can be easily solved. In these cases the coefficient matrix has some special structures. Further on we will focus on general systems where A has no structure [Meyer, 2000], [Golub and Van Loan, 1996], [Demmel, 1997], [Higham, 1996] and [Trefethen and Bau, 1997].

1. Systems with diagonal matrices

Suppose that $A \in \mathbb{R}^{n\times n}$ is a diagonal and nonsingular matrix, i.e. for all i, $a_{ii} \neq 0.$ In this case the set of linear equations $Ax = b$ can be written as $a_{ii}x_i = b_i$, $i = 1,\dots,n.$ Therefore, the solution is simply given by $x_i = b_i / a_{ii}$, $i = 1,\dots,n.$

2. Systems with upper triangular matrices. (Back substitution)

Consider the system $Ux = b,$ where the coefficient matrix $U \in \mathbb{R}^{n\times n}$ is an *upper triangular matrix* in which $u_{ii} \neq 0,$ $i = 1,\dots,n,$ i.e. there are no zero pivots.

---

[1] Academy of Romanian Scientists, 54 Splaiul Independenţei, Sector 5, Bucharest, Romania.

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

For solving this system the general *back substitution* is as follows:

1. Firstly compute $x_n = b_n / u_{nn}$.
2. Determine x_i, $i = n-1, n-2, \ldots, 1$, recursively as:

$$x_i = \frac{1}{u_{ii}}(b_i - u_{i,i+1}x_{i+1} - u_{i,i+2}x_{i+2} - \cdots - u_{i,n}x_n) = \frac{1}{u_{ii}}\left(b_i - \sum_{k=i+1}^{n} u_{ik}x_k \right).$$

3. Systems with lower triangular matrices. (Forward substitution)

Consider the system $Lx = b$, where the coefficient matrix $L \in \mathbb{R}^{n\times n}$ is a *unit lower triangular matrix*, that is

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

For solving this system the general *forward substitution* is as follows:

1. Firstly compute $x_1 = b_1$.
2. Determine x_i, $i = 2, 3, \ldots, n$ recursively as:

$$x_i = b_i - (l_{i,1}x_1 + l_{i,2}x_2 + \ldots + l_{i,i-1}x_{i-1}) = b_i - \sum_{k=1}^{i-1} l_{ik}x_k.$$

4. Systems with orthogonal matrices

A matrix $A \in \mathbb{R}^{n\times n}$ is orthogonal if $A^T A = I$, i.e. $A^{-1} = A^T$. In this case the solution of the system $Ax = b$ can be computed by a simple matrix-vector product $x = A^T b$.

5. Systems with permutation matrices

Let $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ be a permutation of $(1, 2, \ldots, n)$. A permutation matrix is a square matrix obtained from the identity matrix by a permutation of its rows. Every row and column of a permutation matrix have a single 1 with 0s everywhere else. If A is a permutation matrix, then solving $Ax = b$ is very simple: x is obtained by permuting the entries of b by π^{-1}.

6. Gaussian Elimination (LU Factorization)

Gaussian elimination (Gauss method) is a *direct method* for solving linear systems of equations $Ax = b,$ where $A \in \mathbb{R}^{n\times n}$ is a real matrix. Let A be a nonsingular matrix, then the end result of applying the Gaussian elimination to A is an upper triangular matrix with nonzero elements on the main diagonal, i.e.

$$A \xrightarrow{\text{Gaussian elimination}} \begin{bmatrix} * & * & \cdots & * \\ 0 & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & * \end{bmatrix}.$$

Gaussian elimination for a general small example

Let us illustrate the Gaussian elimination by means of a general small example:

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

The Gaussian elimination transforms this system to triangular form as follows. Suppose that $a_{11} \neq 0.$ Multiplying the first row by a_{21}/a_{11} and subtracting it from the second row leads to the equivalent system

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2^{(2)} \\ b_3 \end{bmatrix},$$

where

$$a_{22}^{(2)} = a_{22} - (a_{21}/a_{11})a_{12},$$
$$a_{23}^{(2)} = a_{23} - (a_{21}/a_{11})a_{13}$$

and

$$b_2^{(2)} = b_2 - (a_{21}/a_{11})b_1.$$

Now, multiplying the first row by a_{31}/a_{11} and subtracting it from the third row leads to the equivalent system

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2^{(2)} \\ b_3^{(2)} \end{bmatrix},$$

where

$$a_{32}^{(2)} = a_{32} - (a_{31}/a_{11})a_{12},$$
$$a_{33}^{(2)} = a_{33} - (a_{31}/a_{11})a_{13}$$

and

$$b_3^{(2)} = b_3 - (a_{31} / a_{11}) b_1.$$

Finally, assuming that $a_{22}^{(2)} \neq 0,$ multiplying the new second row by $(a_{32}^{(2)} / a_{22}^{(2)})$ and subtracting it from the third row leads to the system

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} \\ 0 & 0 & a_{33}^{(3)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2^{(2)} \\ b_3^{(3)} \end{bmatrix},$$

where

$$a_{33}^{(3)} = a_{33}^{(2)} - (a_{32}^{(2)} / a_{22}^{(2)}) a_{23}^{(2)}$$

and

$$b_3^{(3)} = b_3^{(2)} - (a_{32}^{(2)} / a_{22}^{(2)}) b_2^{(2)}.$$

Observe that the system obtained at the end of this process has the upper triangular form $Ux = c,$ where

$$U = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} \\ 0 & 0 & a_{33}^{(3)} \end{bmatrix} \quad \text{and} \quad c = \begin{bmatrix} b_1 \\ b_2^{(2)} \\ b_3^{(3)} \end{bmatrix},$$

which can be solved by back substitution.

General Gaussian elimination

The above process may be performed in general by creating zeros in the first column, then in the second one, and so forth. For $k = 1, 2, \ldots, n-1$ the Gaussian elimination is defined by the following formulae:

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - (a_{ik}^{(k)} / a_{kk}^{(k)}) a_{kj}^{(k)}, \quad i, j > k$$

and

$$b_i^{(k+1)} = b_i^{(k)} - (a_{ik}^{(k)} / a_{kk}^{(k)}) b_k^{(k)}, \quad i > k,$$

where $a_{ij}^{(1)} = a_{ij},\ i, j = 1, 2, \ldots, n.$

To be well defined, the only assumption required is that $a_{kk}^{(k)} \neq 0,\ \ k = 1, 2, \ldots, n,$ hold. In the Gaussian elimination these entries are called *pivots*. Usually, the following notation is used:

$$A^{(k)} x = b^{(k)},$$

as the system obtained after $(k-1)$ stages, $k = 1, 2, \ldots, n,$ where $A^{(1)} = A$ and $b^{(1)} = b.$ The final matrix $A^{(n)}$ is upper triangular.

Zero Pivots – Row Interchanges

The above described Gaussian process breaks down when a pivot is zero, say $a_{kk}^{(k)} = 0.$ In this case, in order to continue the Gaussian process, row interchanges are needed. We illustrate zero pivots and row interchanges by using a small general example.

Suppose we have executed two stages of the Gaussian elimination on a system of order 5 and at the third stage the system is $A^{(3)}x = b^{(3)}$, in the following form:

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} & a_{15}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{24}^{(2)} & a_{25}^{(2)} \\ 0 & 0 & 0 & a_{34}^{(3)} & a_{35}^{(3)} \\ 0 & 0 & a_{43}^{(3)} & a_{44}^{(3)} & a_{45}^{(3)} \\ 0 & 0 & a_{53}^{(3)} & a_{54}^{(3)} & a_{55}^{(3)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ b_3^{(3)} \\ b_4^{(3)} \\ b_5^{(3)} \end{bmatrix}.$$

In this case, if $a_{43}^{(3)} \neq 0$ or $a_{53}^{(3)} \neq 0$ holds, then the third row is interchanged with either the fourth or the fifth row and we may continue the Gaussian process. This interchanging to obtain nonzero pivots is called *pivoting*.

On the other hand, the Gaussian elimination breaks down if $a_{33}^{(3)} = a_{43}^{(3)} = a_{53}^{(3)} = 0$. In this case the matrix is singular. i.e.

$$\det A^{(3)} = a_{11}^{(1)} a_{22}^{(2)} \det \begin{bmatrix} 0 & a_{34}^{(3)} & a_{35}^{(3)} \\ 0 & a_{44}^{(3)} & a_{45}^{(3)} \\ 0 & a_{54}^{(3)} & a_{55}^{(3)} \end{bmatrix} = 0.$$

Relationship with LU Factorization

For solving the system $Ax = b$, where A is nonsingular, the Gaussian elimination consists of the following 4 steps [Demmel, 1997]:

1. Factorize the matrix A as $A = PLU$, where:
   P is a *permutation* matrix,
   L is a *unit lower triangular* matrix,
   U is a *nonsingular upper triangular* matrix.
2. Solve the system $PLUx = b$ subject to LUx by permuting the entries of b, i.e. $LUx = P^{-1}b = P^T b.$
3. Solve the system $LUx = P^{-1}b$ subject to Ux by *forward substitution*, i.e. $Ux = L^{-1}(P^{-1}b).$
4. Solve the system $Ux = L^{-1}(P^{-1}b)$ subject to x by *backward substitution*, i.e. $x = U^{-1}(L^{-1}(P^{-1}b)).$

The following result is central in the Gaussian elimination:

The following two statements are equivalent:

1. *There exists a unique unit lower triangular matrix L and a nonsingular upper triangular matrix U such that $A = LU$. This is called LU factorization of A.*
2. *All leading principal submatrices of A are nonsingular.*

LU factorization without pivoting can fail on nonsingular matrices and therefore we need to introduce permutations into the Gaussian elimination.

If A is a nonsingular matrix, then there exist permutation matrices P_1 and P_2, a unit lower triangular matrix L and a nonsingular upper triangular matrix U such that $P_1AP_2 = LU$. Observe that P_1A reorders the rows of A. AP_2 reorders the columns of A. P_1AP_2 reorders both the rows and the columns of A.

If A is nonsingular, then it has a nonzero entry. Therefore, we choose the permutations P_1' and P_2' so that the (1,1) entry of $P_1'AP_2'$ is nonzero. Now we write the factorization and solve for the unknown components:

$$P_1'AP_2' = \begin{bmatrix} a_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ L_{21} & I_{n-1} \end{bmatrix} \begin{bmatrix} u_{11} & U_{12} \\ 0 & \bar{A}_{22} \end{bmatrix} = \begin{bmatrix} u_{11} & U_{12} \\ L_{21}u_{11} & L_{21}U_{12} + \bar{A}_{22} \end{bmatrix},$$

where $A_{22}, \bar{A}_{22} \in \mathbb{R}^{(n-1)\times(n-1)}$ and $L_{21}, U_{12}^T \in \mathbb{R}^{n-1}$. Solving for the components of this 2×2 block factorization we get:

$$u_{11} = a_{11} \neq 0, \quad U_{12} = A_{12}, \quad L_{21}u_{11} = A_{21}.$$

Since $u_{11} = a_{11} \neq 0$, we can solve it to get $L_{21} = \dfrac{A_{21}}{a_{11}}$. Finally, $L_{21}U_{12} + \bar{A}_{22} = A_{22}$ implies that $\bar{A}_{22} = A_{22} - L_{21}U_{12}$. Observe that

$$\det P_1'AP_2' = \det \begin{bmatrix} 1 & 0 \\ L_{21} & I_{n-1} \end{bmatrix} \det \begin{bmatrix} u_{11} & U_{12} \\ 0 & \bar{A}_{22} \end{bmatrix} = 1(u_{11} \det \bar{A}_{22}).$$

Since $\det P_1'AP_2' = \pm \det A \neq 0$, it follows that $\det \bar{A}_{22}$ must be nonzero. Therefore, the factorization process may continue.

Indeed, by induction there exist the permutation matrices $\bar{P}_1$ and $\bar{P}_2$ so that $\bar{P}_1\bar{A}_{22}\bar{P}_2 = \bar{L}\bar{U}$, where $\bar{L}$ is a unit lower triangular matrix and $\bar{U}$ is an upper triangular and nonsingular matrix. Substituting this in the above 2×2 block factorization we get:

$$P_1'AP_2' = \begin{bmatrix} 1 & 0 \\ L_{21} & I \end{bmatrix} \begin{bmatrix} u_{11} & U_{12} \\ 0 & \bar{P}_1^T\bar{L}\bar{U}\bar{P}_2^T \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ L_{21} & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \bar{P}_1^T\bar{L} \end{bmatrix} \begin{bmatrix} u_{11} & U_{12} \\ 0 & \bar{U}\bar{P}_2^T \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ L_{21} & \bar{P}_1^T\bar{L} \end{bmatrix} \begin{bmatrix} u_{11} & U_{12}\bar{P}_2 \\ 0 & \bar{U} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \bar{P}_2^T \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \bar{P}_1^T \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \bar{P}_1L_{21} & \bar{L} \end{bmatrix} \begin{bmatrix} u_{11} & U_{12}\bar{P}_2 \\ 0 & \bar{U} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \bar{P}_2^T \end{bmatrix}.$$

Therefore, the desired factorization of A is:

$$P_1AP_2 = \left(\begin{bmatrix} 1 & 0 \\ 0 & \bar{P}_1 \end{bmatrix} P_1' \right) A \left(P_2' \begin{bmatrix} 1 & 0 \\ 0 & \bar{P}_2 \end{bmatrix} \right) = \begin{bmatrix} 1 & 0 \\ \bar{P}_1L_{21} & \bar{L} \end{bmatrix} \begin{bmatrix} u_{11} & U_{12}\bar{P}_2 \\ 0 & \bar{U} \end{bmatrix}.$$

The next two results state simple ways to choose the permutation matrices P_1 and P_2 to guarantee that the Gaussian elimination will run on nonsingular matrices.

<u>Gaussian elimination with partial pivoting</u>
We can choose the permutation matrices $P_2' = I$ and P_1' in such a way that a_{11} is the largest entry in absolute value in its column, which implies that $L_{21} = \frac{A_{21}}{a_{11}}$ has entries bounded by 1 in absolute value. More generally, at step i of the Gaussian elimination, where we are computing the ith column of L, we reorder rows i through n so that the largest entry in the column is on the diagonal. This is called „Gaussian elimination with partial pivoting", or GEPP for short. GEPP guarantees that all entries of L are bounded by one in absolute value.

<u>Gaussian elimination with complete pivoting</u>
We can choose the permutation matrices P_2' and P_1' in such a way that a_{11} is the largest entry in absolute value in the whole matrix. More generally, at step i of the Gaussian elimination, where we are computing the ith column of L, we reorder rows and columns i through n so that the largest entry in this submatrix is on the diagonal. This is called „Gaussian elimination with complete pivoting", or GECP for short.

The following algorithm is an implementation of the results mentioned above by performing permutations, by computing the first column of L and the first row of U, and then by updating A_{22} to get $\bar{A}_{22} = A_{22} - L_{21}U_{12}$.

Algorithm GE (LU factorization with pivoting)

---

for $i = 1$ *to* $n-1$
  apply permutations so that $a_{ii} \neq 0$ *(permute* L *and* U *too)*
   / for example, for GEPP, swap rows* j *and* i *of* A *and of* L
    where $|a_{ji}|$ *is the largest entry in* $|A(i:n,i)|$; *for GECP, swap*
    rows j *and* i *of* A *and of* L, *and columns* k *and* i *of* A *and*
    of U, *where* $|a_{jk}|$ *is the largest entry in* $|A(i:n,i:n)|$ **/*
  / compute column* i *of* L **/*
  for $j = i+1$ *to* n
    $l_{ji} = a_{ji} / a_{ii}$
  end for
  / compute row* i *of* U **/*
  for $j = i$ *to* n
    $u_{ij} = a_{ij}$
  end for

```
/* update A_22 */
for j = i+1 to n
   for k = i+1 to n
      a_jk = a_jk - l_ji u_ik
   end for
end for
end for
```

Remark

Once the column i of A has been used to compute the column i of L, it will never be used later in the algorithm GE. Similarly, row i of A is never used after computing row i of U. This property allows us to overwrite L and U on top of A as soon as they are computed. Therefore, there is no need for extra space to store these matrices. L can occupy the strict lower triangle of A (the ones on the diagonal of L are not stored explicitly). Similarly, U can occupy the upper triangle of A. Therefore, the algorithm can be simplified as:

Algorithm LU (LU factorization with pivoting, overwriting L and U on A)

```
for i = 1 to n-1
   apply permutations so that a_ii ≠ 0
   for j = i+1 to n
      a_ji = a_ji / a_ii
   end for
   for j = i+1 to n
      for k = i+1 to n
         a_jk = a_jk - a_ji a_ik
      end for
   end for
end for
```

7. Cholesky factorization

If $A \in \mathbb{R}^{n\times n}$ is symmetric and positive definite, then it can be factored as

$$A = LL^T,$$

where L is a lower triangular and nonsingular matrix with positive diagonal elements. This is called the *Cholesky factorization* of A and can be interpreted as a symmetric LU factorization with $L = U^T$. The matrix L, which is uniquely determined by A, is called the *Cholesky factor* of A. The algorithm is as follows:

Cholesky factorization

---

for $j=1$ *to* n

$$l_{jj}=\left(a_{jj}-\sum_{k=1}^{j-1}l_{jk}^2\right)^{1/2}$$

for $i=j+1$ *to* n

$$l_{ij}=(a_{ij}-\sum_{k=1}^{j-1}l_{ik}l_{jk})/l_{jj}$$

end for

end for

If A is not positive definite, then the Cholesky factorization will fail by attempting to compute the square root of a negative number or by dividing by zero. This is the cheapest way to test if a symmetric matrix is positive definite.
The Cholesky factorization can be used to solve the system $Ax=b$ when A is symmetric and positive definite.

Solving linear systems by Cholesky factorization

1. *Cholesky factorization*. Factor A as $A=LL^T$.
2. *Forward substitution*. Solve $Lz=b$.
3. *Back substitution*. Solve $L^T x=z$.

8. The factor-solve method

For solving the linear system $Ax=b$ the basic approach is based on expressing A as a product of nonsingular matrices:

$$A=A_1A_2\cdots A_r.$$

Therefore, the solution is given by:

$$x=A^{-1}b=A_r^{-1}A_{r-1}^{-1}\ldots A_1^{-1}b.$$

The solution x is computed working from right to left as:

$$\begin{aligned}
&z_1=A_1^{-1}b,\\
&z_2=A_2^{-1}z_1=A_2^{-1}A_1^{-1}b,\\
&\quad\vdots\\
&z_{r-1}=A_{r-1}^{-1}z_{r-2}=A_{r-1}^{-1}A_{r-2}^{-1}\cdots A_1^{-1}b,\\
&x=A_r^{-1}z_{r-1}=A_r^{-1}A_{r-1}^{-1}\cdots A_1^{-1}b.
\end{aligned}$$

We see that the ith step of this process requires computing $z_i=A_i^{-1}z_{i-1}$, i.e. solving the linear system $A_iz_i=z_{i-1}$. The step of expressing A in factored form is called the *factorization step*. On the other hand, the process of computing $x=A^{-1}b$ recursively by solving a sequence of systems of the form $A_iz_i=z_{i-1}$, is called the *solve step*. The idea of the factor-solve method is to determine the factors A_i,

$i = 1,\ldots,r,$ as simple as possible, i.e. diagonal, lower or upper triangular, permutation, orthogonal, etc.

Factor-solve method and LU factorization.

Assume that in the general Gaussian elimination $a_{kk}^{(k)} \neq 0$ hold for every $k = 1,\ldots,n.$ Referring to the general Gaussian elimination, we see that $l_{ik} = a_{ik}^{(k)} / a_{kk}^{k}$ for $i > k,$ is exactly what is used to multiply the kth row and subsequently subtract it from the ith row in building the new ith row. l_{ik} is called a *multiplier*.

Now, let $L^{(k)}$ be the unit lower triangular matrix which differs from the identity matrix only in the kth column below the main diagonal, where the negatives of the multipliers l_{ik} appear. These matrices are called *elementary lower triangular matrices*. With these matrices, the general Gaussian elimination can be expressed in matrix notation as:

$$A^{(k+1)} = L^{(k)} A^{(k)},$$

where $A^{(1)} = A.$ Using these relations for all values of k we get:

$$U = A^{(n)} = L^{(n-1)} L^{(n-2)} \cdots L^{(1)} A.$$

The inverse of $L^{(k)}$ is very easy to be computed: by changing the sign of the multipliers.

$$L^{(k)} = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & -l_{k+1,k} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & -l_{n,k} & & & 1 \end{bmatrix}, \quad (L^{(k)})^{-1} = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & l_{k+1,k} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & l_{n,k} & & & 1 \end{bmatrix}$$

Therefore, from the above relations we get:

$$A = (L^{(1)})^{-1} (L^{(2)})^{-1} \cdots (L^{(n-1)})^{-1} U.$$

The solution of the linear system $Ax = b$ is very easy to be computed by using the structure of the $L^{(k)}$ and U matrices.

9. Solving underdetermined linear systems

Let us consider the linear system of equations $Ax = b,$ where $A \in \mathbb{R}^{m \times n},$ and $m < n.$ Assume that $rank(A) = m,$ so there is at least one solution for all $b.$ In many applications it is sufficient to know one particular solution $\bar{x}.$ In other situations it is necessary to have a parameterization of all solutions as

$$\{x : Ax = b\} = \{Zy + \bar{x} : y \in \mathbb{R}^{n-m}\},$$

where Z is a matrix whose columns form a basis for the null space of A.

The solution of the underdetermined system $Ax = b$ is very easy to be determined if a $m \times m$ nonsingular submatrix of A is known. Assume that the first m columns of A are linearly independent. The system can be written as

$$Ax = \begin{bmatrix} A_1 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = A_1 x_1 + A_2 x_2 = b,$$

where $A_1 \in \mathbb{R}^{m \times m}$ is nonsingular. Therefore, we can express x_1 as

$$x_1 = A_1^{-1}(b - A_2 x_2).$$

A particular solution for the system $Ax = b$ is $\bar{x}_2 = 0$ and $\bar{x}_1 = A_1^{-1} b.$ All solutions of $Ax = b$ can be parameterized using $x_2 \in \mathbb{R}^{n-m}$ as a free parameter. We can write:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -A_1^{-1} A_2 \\ I \end{bmatrix} x_2 + \begin{bmatrix} A_1^{-1} b \\ 0 \end{bmatrix}.$$

This gives the following parameterization:

$$Z = \begin{bmatrix} -A_1^{-1} A_2 \\ I \end{bmatrix}, \quad \bar{x} = \begin{bmatrix} A_1^{-1} b \\ 0 \end{bmatrix}.$$

10. The QR factorization

The matrix $A \in \mathbb{R}^{n \times m}$ with $m \le n$ and $rank A = m$ can be factored as:

$$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix},$$

where $Q_1 \in \mathbb{R}^{n \times m}$ and $Q_2 \in \mathbb{R}^{n \times (n-m)}$ satisfy

$$Q_1^T Q_1 = I, \quad Q_2^T Q_2 = I, \quad Q_1^T Q_2 = 0$$

and $R \in \mathbb{R}^{m \times m}$ is upper triangular with nonzero diagonal elements. This is called the *QR factorization* of A.

The QR factorization can be used for solving the underdetermined systems of linear equations $Ax = b,$ where $A \in \mathbb{R}^{m \times n}$ with $m < n.$ Consider that

$$A^T = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}$$

is the QR factorization of A^T. Therefore, $\bar{x} = Q_1 (R^T)^{-1} b$ satisfies the equations:

$$A\bar{x} = R^T Q_1^T Q_1 (R^T)^{-1} b = b.$$

The columns of Q_2 form a basis for the nullspace of A. Therefore, the complete solution set of the above system can be parameterized as

$$\{x = \bar{x} + Q_2 z : z \in \mathbb{R}^{n-m}\}.$$

Usually, the QR factorization is used for solving underdetermined systems of linear equations. The main drawback of this method is that it is difficult to exploit the sparsity of the matrix. Even if A is sparse, the factor Q is usually dense.

11. LU factorization of rectangular matrices

The matrix $A \in \mathbb{R}^{n \times m}$ with $m \leq n$ and $rank A = m$ can be factored as

$$A = PLU,$$

where $P \in \mathbb{R}^{n \times n}$ is a permutation matrix, $L \in \mathbb{R}^{n \times m}$ is unit lower triangular (i.e. $l_{ij} = 0$ for $i < j$ and $l_{ii} = 1$) and $U \in \mathbb{R}^{m \times m}$ is nonsingular and upper triangular. If the matrix A is sparse, then the LU factorization usually includes row and column permutation, i.e. A is factored as

$$A = P_1 L U P_2,$$

where $P_1 \in \mathbb{R}^{n \times n}$, $P_2 \in \mathbb{R}^{m \times m}$ are permutation matrices. The LU factorization of a sparse rectangular matrix can be calculated efficiently at a cost that is much lower than for dense matrices.

The LU factorization can be used for solving underdetermined systems of linear equations. Consider the system of linear equations $Ax = b$, where $A \in \mathbb{R}^{m \times n}$ with $m < n$. Suppose that the matrix A^T is LU factored as $A^T = PLU$ and L is partitioned as

$$L = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix},$$

where $L_1 \in \mathbb{R}^{m \times m}$ and $L_2 \in \mathbb{R}^{(n-m) \times m}$. Then the solution set of the system can be parameterized as

$$\{x : Ax = b\} = \{Zz + \bar{x} : z \in \mathbb{R}^{n-m}\},$$

with

$$Z = P \begin{bmatrix} -(L_1^T)^{-1} L_2^T \\ I \end{bmatrix}, \quad \bar{x} = P \begin{bmatrix} (L_1^T)^{-1} (U^T)^{-1} b \\ 0 \end{bmatrix}.$$

The LU factorization of rectangular matrices is used in MINOS and SNOPT packages [Saunders, 2015].

References

Boyd, S., Vandenberghe, L., (2006) *Convex Optimization.* Cambridge University Press, Cambridge (UK), 2nd edition, 2006.

Davis, T.A., (2006) *Direct Methods for Sparse Systems.* SIAM, Philadelphia, PA, 2006.

Duff, I.S., Erismann, A.M., Reid, J.K., (1986) *Direct Methods for Sparse Matrices.* Clarendon Press, 1986.

Demmel, J.W., (1997) *Applied Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.
George, A., Liu, J.W.H., (1981) *Computer Solution of Large-Sparse Positive Definite Systems*, Prentice-Hall, 1981.
Gill, P.E., Murray, W., Wright, M.H., (1981) *Practical Optimization.* Academic Press, 1981.
Golub, G.H., Van Loan, C.F., (1996) *Matrix computation.* Johns Hopkins University Press, Baltimore, MD. Third edition, 1996.
Higham, N.J., (1996) *Accuracy and Stability of Numerical Algorithms*. SIAM, PA, Philadelphia, 1996.
Meyer, C.D., (2000) *Matrix Analysis and Applied Linear Algebra.* SIAM, Philadelphia, PA, 2000.
Nocedal, J., Wright, S.J., (2006) *Numerical optimization.* Springer Series in Operations Research. Springer Science+Business Media, New York, Second ed., 2006.
Luenberger, D.G., (1984) *Introduction to linear and nonlinear programming.* Addison-Wesley Publishing Company, Reading, Second ed., 1984.
Luenberger, D. G.**,** Ye, Y.**,** (2008) *Linear and nonlinear programming*. International Series in Operations Research & Management Science 116 (Third ed.). New York, Springer. 2008.
Osterby, O., Zlatev, Z., (1983) *Direct Methods for Sparse Matrices.* Springer Verlag, New York, 1983.
Saad, Y., (1996) *Iterative methods for sparse linear systems.* PWS Publishing Company, Boston, 1996.
Saad, Y., (2011) *Numerical Methods for Large Eigenvalue Problems.* Second Edition, SIAM, Philadelphia, PA, 2011.
Schendel, U., (1989) *Sparse Matrices – Numerical aspects with applications for scientists and engineers.* Ellis Horwood Limited, Chichester, 1989.
Trefethen, L.N., Bau III, D., *Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.