Linear systems

Francesco Marchetti

Fundamentals of Computational Mathematics

Summary

An introduction

A **linear system** of m linear equations and n variables $x_1, \ldots, x_n$ can be formalized as

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases}$$

where a_{ij} and b_i are parameters of the system, $i = 1, \ldots, m$, $j = 1, \ldots, n$. It is very often more comfortable to use the matrix notation

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

where $\boldsymbol{x} = (x_1, \ldots, x_n)^\intercal$ and $\boldsymbol{b} = (b_1, \ldots, b_m)^\intercal$.

Solutions of a linear system

A linear system may have infinite solutions, a unique solution or no solutions. Moreover,

- If $m = n$, we have a **square** linear system.
- If $m > n$, we have a **overdetermined** linear system.
- If $m < n$, we have a **underdetermined** linear system.

In the following, we will first restrict to the square case and unique solution. We recall that the solution is unique if and only if $\det(A) \neq 0$.
Then, one could think that the solution can be easily found by computing $\boldsymbol{x} = A^{-1}\boldsymbol{b}$. However, computing the inverse of a matrix is a very ill-conditioned operation that we have to avoid!

Structure of the matrix

The strategy to solve a linear system depends on the structure of the system matrix A. We recall that a matrix can be...

- **dense**, when many (or almost all) of its elements are non-zero,
- **sparse**, if only a small portion of its elements are non-zero,
- **banded**, when $a_{ij} = 0$ if $|i - j| > b$, $b \in \mathbb{R}_+$ the bandwidth,
- **triangular**, upper or lower, if the elements below or over the diagonal are zeros.

In some cases, the matrix could present local noteworthy properties, not global, and therefore have a **block structure**.
Recognizing particular structures in the matrix is important, as they have to be exploited in concrete applications!

Properties of the matrix, direct and iterative mehods

Other relevant properties for the matrix A are being

- **symmetric**, if $A = A^\intercal$,
- **positive definite**, if for any $\boldsymbol{x} \in \mathbb{R}^n$ we have $\boldsymbol{x}^\intercal A \boldsymbol{x} > 0$.

There are two main categories of algorithms for numerically solving linear systems:

- **direct methods**, which provide an exact solution (besides rounding errors) and exploit matrix factorizations,
- **iterative methods**, which produce a sequence of approximations of the real solution, but often faster than direct methods.

We start from direct methods.

Diagonal matrices

If A is a diagonal matrix

$$A = \begin{pmatrix} a_{11} & & \\ & \ddots & \\ & & a_{nn} \end{pmatrix},$$

the solution can be easily found with a cost of $\mathcal{O}(n)$ as

$$x_i = b_i / a_{ii}, \quad i = 1, \ldots, n.$$

Inverting a diagonal matrix is not a problem at all!

Back- or forward-substitution

If the matrix A is lower triangular ($A = L$) or upper triangular ($A = U$), we use **forward-** or **back-substitution**, respectively.

Algorithm 1 Forward substitution.

1: $b_1 = b_1 / L_{1,1}$
2: **for** $i = 2 : n$ **do**
3: $\quad b_i = (b_i - L_{i,1:i-1}\, b_{1:i-1}) / L_{ii}$
4: **end for**

Algorithm 2 Back-substitution.

1: $b_n = b_n / U_{n,n}$
2: **for** $i = n-1 : -1 : 1$ **do**
3: $\quad b_i = (b_i - U_{i,i+1:n}\, b_{i+1:n}) / U_{ii}$
4: **end for**

The idea is really simple and efficient, with a cost of $\mathcal{O}(n^2)$. In fact, many direct and also iterative methods aim at solving more general linear systems by reducing to diagonal and/or triangular ones.

LU decomposition

LU factorization decomposes the matrix A as $A = LU$, where L is lower triangular and U is upper triangular. Then, the solution is found by solving two triangular linear systems.

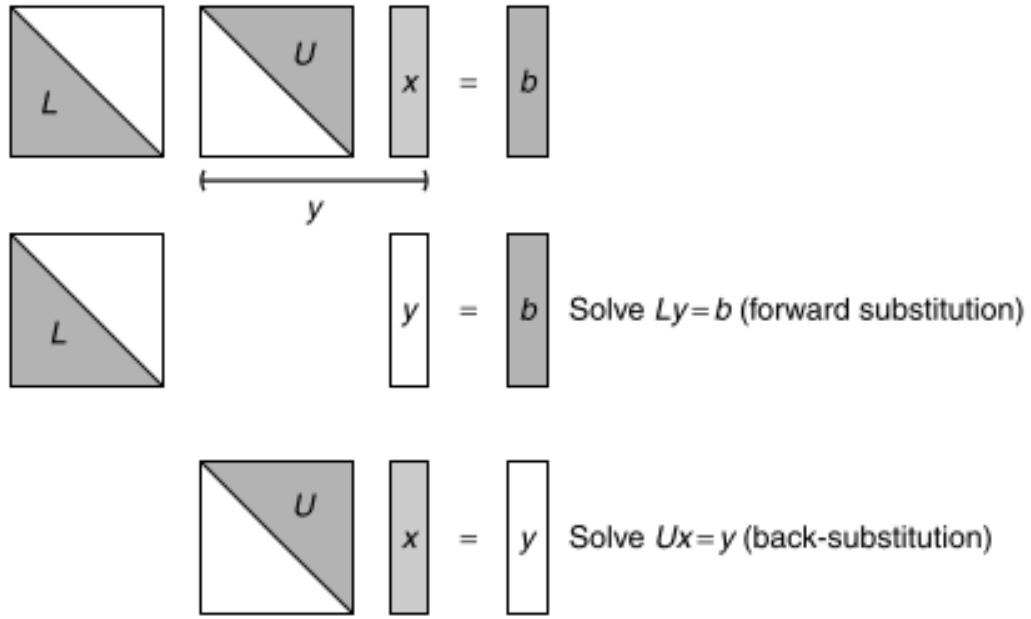

FIGURE 3.1 LU factorization.

Elementary matrices

To understand how to find such matrices L and U, we can use **elementary matrices**. Let I be the $n \times n$ identity matrix.

1. The **permutation** matrix P_{ij} is derived from I by switching the i-th and j-th rows. Its inverse is $P_{ij}^{-1} = P_{ij}$.
2. The **dilation** matrix $D_i(c)$ is derived from I by multiplying the i-th row by a scalar $c \in \mathbb{R}$, $c \neq 0$. Its inverse is $D_i(c)^{-1} = D_i(1/c)$.
3. The **summation** matrix $S_{ij}(c)$ is derived from I by summing c times the j-th row to row i. Its inverse is $S_{ij}(c)^{-1} = S_{ij}(-c)$.

It is important to observe that the inverses of such matrices are elementary too, we know them *a priori* and we don't need to compute them.

Some examples

Let $n = 3$. We have

$$P_{13} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad D_2(3) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$S_{31}(2) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix}, \quad S_{23}(-1) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}$$

Elementary matrices and LU

Our aim is to find a sequence of elementary matrices $E_1, \ldots, E_p$, $p \in \mathbb{N}$, so that

$$E_1 \ldots E_p A = U \Longrightarrow A = E_p^{-1} \ldots E_1^{-1} U.$$

In this manner, if $L = E_p^{-1} \ldots E_1^{-1}$ is lower triangular... we won! But is it actually lower triangular? Well, if all matrices $E_1, \ldots, E_p$ can be chosen to be lower triangular, then yes! Let us comment a concrete example to see if this is always possible.

LU: an example

Let

$$A = \begin{pmatrix} 2 & 0 & 1 \\ 1 & 0 & -3 \\ 0 & 1 & 0 \end{pmatrix}.$$

We have $\det(A) = -7$, therefore it is invertible. The first row is ok, since the element of the diagonal $a_{11} = 2$ is non-zero (U must be invertible!). So we work on the second row, to get rid of that 1 we multiply

$$S_{21}(-1/2)A = \begin{pmatrix} 2 & 0 & 1 \\ 0 & 0 & -7/2 \\ 0 & 1 & 0 \end{pmatrix}.$$

LU: an example (cont.)

Now, we are not happy with the second row, since there is a zero element on the diagonal. Unfortunately, we need to use the permutation matrix P_{23}, which is not lower triangular, to obtain

$$P_{23}S_{21}(-1/2)A = \begin{pmatrix} 2 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & -7/2 \end{pmatrix} = U.$$

Therefore,

$$L = S_{21}^{-1}(-1/2)P_{23}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Even if it is possible to actually get a lower triangular matrix with no need of permutations, this situation is really common.

LU: an example with pivoting

To overcome this issue, we consider the so called LU decomposition **with pivoting** and look for L and U so that $PA = LU$, where we admit a permutation matrix P that multiplies A.
Coming back to our example, we notice that

$$P_{23}L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/2 & 0 & 1 \end{pmatrix} = L^{\star},$$

Therefore $P_{23}A = L^{\star}U$, and the factorization with pivoting is completed.

LU: some comments

- What we presented is in fact the Gauss elimination method.
- The decomposition is not unique *a priori*, but L is usually chosen to have ones on the diagonal.
- When using pivoting, we solve

$$\begin{cases} L\boldsymbol{y} = P\boldsymbol{b} \\ U\boldsymbol{x} = \boldsymbol{y}. \end{cases}$$

- In fact, it is more comfortable to work with permutation and summation matrices only, but dilation matrices could be used too.
- The complexity of LU factorization is $\mathcal{O}(n^3)$. It is a good choice if A is dense with no particular structures.

Cholesky decomposition

When A is positive definite, we have a special case for the LU factorization. Indeed, it can be proved that there exists a unique decomposition

$$A = LL^{\mathsf{T}}$$

where L is lower triangular and $L^{\mathsf{T}} = U$. This is called the **Cholesky decomposition** of A, and the linear system is then solved as in the LU factorization.
The computational cost to construct this factorization is $\mathcal{O}(n^3)$, but slightly smaller than the more general LU decomposition. We omit the details.

Introduction to iterative methods

Iterative schemes represent a valuable choice, since they are often faster than direct methods.
The three methods that we are going to discuss are based on the decomposition

$$A = L + D + U,$$

where L is lower triangular, D is diagonal and U is upper triangular. Moreover, fixed a starting vector $\boldsymbol{x}^{(0)}$, the iterative step is of the form

$$M\boldsymbol{x}^{(k+1)} = N\boldsymbol{x}^{(k)} + \boldsymbol{b},$$

where M and N are matrices derived from the LDU decomposition (see the next slides).

Convergence of iterative methods

Let us investigate on the convergence of the presented framework. First, observe that by setting $A = M - N$ we get

$$A\boldsymbol{x} = \boldsymbol{b} \Longrightarrow M\boldsymbol{x} = N\boldsymbol{x} + \boldsymbol{b}.$$

By defining the error vector at iteration k as $\boldsymbol{e}^{(k)} = \boldsymbol{x}^{(k)} - \boldsymbol{x}$, we write

$$M\boldsymbol{x} = N(\boldsymbol{x}^{(k)} - \boldsymbol{e}^{(k)}) + \boldsymbol{b} = \underbrace{N\boldsymbol{x}^{(k)} + \boldsymbol{b}}_{M\boldsymbol{x}^{(k+1)}} - N\boldsymbol{e}^{(k)},$$

therefore $M(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}) = N\boldsymbol{e}^{(k)}$, which leads to

$$\boldsymbol{e}^{(k+1)} = M^{-1}N\boldsymbol{e}^{(k)} = (M^{-1}N)^k\boldsymbol{e}^{(0)}.$$

Convergence of iterative methods (cont.)

We have convergence if $\|\boldsymbol{e}^{(k)}\| \longrightarrow 0$ as $k \longrightarrow +\infty$, where $\|\cdot\|$ is some matrix norm. This can be proved to be equivalent to the fact that $\rho(M^{-1}N) < 1$, where ρ is the **spectral radius** defined as the maximum of the absolute values of the eigenvalues of the matrix. We point out that this result is similar to the one obtained for the fixed point scheme in solving non-linear equations.

This condition on the spectral radius is often expensive to be verified, therefore we can take advantage of some other results that are related to particular cases.

Jacobi

The **Jacobi** scheme is characterized by

$$M = D, \quad N = -(L + U).$$

This is a natural choice to implement, since M becomes a diagonal matrix and therefore computing M^{-1} in each iteration is not computationally demanding.
A sufficient (but not necessary) condition for the convergence of the Jacobi method is the strictly diagonal dominance of the iteration matrix $P = M^{-1}N$. We recall that a matrix $P = P_{(ij)}$ is strictly diagonal dominant if for any $i = 1, \ldots, n$

$$|P_{(ii)}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |P_{(ij)}|.$$

Gauss-Seidel

The **Gauss-Seidel** method is characterized by the choice

$$M = D + L, \quad N = -U.$$

In this framework, the inversion M^{-1} involve a triangular matrix, and is thus more demanding than the Jacobi case. On the other hand, we have some advantages.

Indeed, the Gauss-Seidel scheme is convergent if the iteration matrix is

- strictly diagonal dominant (as Jacobi), or
- symmetric and positive definite.

Moreover, when converging, the spectral radius of the iteration matrix is smaller in the Gauss-Seidel case with respect to the Jacobi case. As a consequence, the Gauss-Seidel method is faster.

Successive overrelaxation (SOR)

The **SOR** method is a generalization of the Gauss-Seidel framework, with

$$M = D + \omega L, \quad N = (1 - \omega)D - \omega U,$$

where ω is a relaxation parameter. Note that we recover the Gauss Seidel case if $\omega = 1$.
It can be proved that in this case, in the convergence assumptions of Gauss-Seidel, $\rho(P) < 1$ if in addition $0 < \omega < 2$. Moreover, SOR is faster than Jacobi and Gauss-Seidel for appropriate choices of ω. In lab, we will tune this parameter to obtain a fast convergence.

We point out that Jacobi, Gauss-Seidel and SOR are all $\mathcal{O}(n^2)$.