SOLUTION OF SYSTEMS OF LINEAR ALGEBRAIC EQUATIONS

Pascal Joly
Research Engineer, CNRS, France

Keywords: direct methods, iterative methods, conjugate gradient methods, domain decomposition methods

Contents

Summary

In this chapter, we examine with the help of a simple example how a famous formula is not suitable to the computer. This will be an introduction to computational science, which is a new trend of applied mathematics. This new area requires many new methods in order to produce actual calculations with the computers. A lot of research works are developed all around the world to help engineers to face this great challenge: numerical simulation of natural events.

1. An Unusable Formula

I have a great admiration for the power of mathematical language, which encapsulates complex theoretical results in a simple symbolic formalism. Among the many formulas learned by students in sciences, let us have a look at the so-called "Cramer's formulas", which give the expression of the solution of a linear system in terms of the data. Let $A \in \mathbb{R}^{n\times n}$ be a square regular (nonsingular) matrix and $b \in \mathbb{R}^n$ a vector, the n components of $x \in \mathbb{R}^n$, solution of the linear system $Ax = b$ are written as

index k goes from 1 up to n. In these relations, so-called *Cramer's formulas*, $|A|$ stands for the determinant of matrix A, with elements $A_{i,j}$.

$$x_k = \frac{\begin{vmatrix} A_{1,1} & \cdot & b_1 & \cdot & A_{1,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{n,1} & \cdot & b_n & \cdot & A_{n,n} \end{vmatrix}}{\begin{vmatrix} A_{1,1} & \cdot & A_{1,k} & \cdot & A_{n,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{n,1} & \cdot & A_{n,k} & \cdot & A_{n,n} \end{vmatrix}}.$$

The determinant of a squared matrix A of order n is obtained by summation of $n!$ products of n terms each:

$$|A| = \det(A) = \sum_{\sigma\in\mathcal{S}(n)} \varepsilon(\sigma) A_{1,\sigma(1)} A_{2,\sigma(2)} \cdots\cdots A_{n,\sigma(n)}.$$

So the calculation of the solution of the linear system by these formulas requires the evaluation of $n+1$ determinants of order n, which leads to a total cost of $n(n+1)n!$

operations (additions and multiplications). In the limits of a theoretical study, this does not look very expensive (it is just another formula), however this is not true as we can see with an actual (though very simple) example:

Question: A computer performs around 10^{12} operations per second; what is the computational time to solve a linear system of order 20 (power breakdowns or strikes would not be taken into account!).

Answer: As $20! \approx 2,43 \times 10^{18}$, the computational time of the solution of a linear system of order 20 by this method requires $1,02 \times 10^{21}$ operations. A (super) computer performing 10^{12} operations by second will need $(1,02 \times 10^{21}) / (10^{12} \times 365 \times 86400) \approx 32$ years to make it up!

Furthermore, as $100! \approx 9,33 \times 10^{157}$, one can easily check that the computational time to solve a linear system of order 100 by the Cramer's formulas is far above the age of the universe!

But today, engineers have to solve huge linear systems: $n \approx 10^6$; so it is obvious that these elegant formulas are not usable for this work !

This very simple example shows that one has to invent new methods better suited to calculation on the computer. This is one of the aims of numerical analysis, a (rather) new branch of mathematics. The following section is an introduction to some of these methods, which shall be developed later.

1.1. The Case of a Triangular Matrix

If the matrix A is triangular, the solution of the linear system can be obtained in a cheaper way. Triangular matrices are squared matrices with a special pattern:

$$L = \begin{bmatrix} x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ x & x & x & 0 & 0 & 0 & 0 & 0 \\ x & x & x & x & 0 & 0 & 0 & 0 \\ x & x & x & x & x & 0 & 0 & 0 \\ x & x & x & x & x & 0 & 0 & 0 \\ x & x & x & x & x & x & 0 & 0 \\ x & x & x & x & x & x & x & x \end{bmatrix} \quad \text{et} \quad U = \begin{bmatrix} x & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x \\ 0 & 0 & x & x & x & x & x & x \\ 0 & 0 & 0 & x & x & x & x & x \\ 0 & 0 & 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & x \end{bmatrix}$$

Let $L \in \mathbb{R}^{n \times n}$ be a regular lower triangular matrix, the solution of the linear system $Ly = b$ is obtained by the following formulas:

for $i = 1, ..., n$ **do**
$y_i = [b_i - \sum_{j<i} L_{i,j}\ y_j] / L_{i,i}$.

Likewise let $U \in \mathbb{R}^{n \times n}$ be an upper triangular regular matrix, the solution of the linear system $U\,x = b$ is obtained by the following formulas:

for $i = n, ..., 1$ **do**
$x_i = [y_i - \sum_{j>i} U_{i,j}\ x_j] / U_{i,i}$.

These relations are easy to check. So the vector x is computed component by component, beginning with the last one y_n (**backsubstitution procedure**), by the same way y is computed component by component, beginning with the first element x_1. As matrices L and U are supposed to be regular, the diagonals of both matrices have non-zero elements.

Computational cost:

According to these new formulas, the computation of component y_k costs k multiplications, k additions plus one division, that makes altogether $2k+1$ operations. So the total cost of the n components of the vector y is of order $n(n+1)$ operations, much more reasonable than the previous formulas! (We obtain the same cost for the upper triangular system $U\,x = y$).

This analysis leads us to a new approach to the calculation of the solution of linear systems of the general form (with matrix A squared and regular, but not triangular) we try to use this property by a factorization of the matrix A into two triangular matrices: $A = LU$.

2. Direct Methods

2.1. Introduction

In the previous section, we have seen how the triangular pattern of a matrix can be used to simplify the calculation of the solution of a linear system. But how to manage the general case? The classical answer is the factorization of matrix A into two triangular matrices: $A = LU$. The associated algorithms are known as **direct methods** these are Gauss, Crout and Cholesky methods, which are developed in this section. The last part of this section is devoted to the practical use of such methods in the particular (but significant) frame of sparse matrices.

2.2. Factorization Method

The easiest way to introduce the factorization method is to suppose that the problem is partially solved: the matrix $A \in \mathbb{R}^{n\times n}$ is written as

$$A = \begin{pmatrix} [A]_{1,1} & [A]_{1,2} \\ [A]_{2,1} & [A]_{2,2} \end{pmatrix}$$

with $[A]_{1,1} \in \mathbb{R}^{n_1\times n_1}$, $[A]_{2,1} \in \mathbb{R}^{n_2\times n_1}$, $[A]_{1,2} \in \mathbb{R}^{n_1\times n_2}$, $[A]_{2,2} \in \mathbb{R}^{n_2\times n_2}$, and n_1, n_2 are two non zero integers satisfying $n = n_1 + n_2$. Now we suppose a factorization of the first block known, that is, $[A]_{1,1} = [L]_{1,1}[U]_{1,1}$ with $[L]_{1,1}, [U]_{1,1} \in \mathbb{R}^{n_1\times n_1}$ triangular regular matrices (diagonal elements of $[L]_{1,1}$ are equal to 1); then we write

$$A = \begin{pmatrix} [A]_{1,1} & [A]_{1,2} \\ [A]_{2,1} & [A]_{2,2} \end{pmatrix} = \begin{pmatrix} [L]_{1,1} & 0 \\ [L]_{2,1} & I_{n_2} \end{pmatrix} \times \begin{pmatrix} [U]_{1,1} & [U]_{1,2} \\ 0 & \mathcal{S} \end{pmatrix}.$$

Identifying both parts of this relation leads to

$$[A]_{1,1} = [L]_{1,1}[U]_{1,1} \qquad [A]_{1,2} = [L]_{1,1}[U]_{1,2}$$
$$[A]_{2,1} = [L]_{2,1}[U]_{1,1} \qquad [A]_{2,2} = [L]_{2,1}[U]_{1,2} + \mathcal{S}.$$

in which the regular matrices $[L]_{1,1}$ and $[U]_{1,1}$ are known, so

$[U]_{1,2} = [L]_{1,1}^{-1}[A]_{1,2}$ and then $[L]_{2,1} = [A]_{2,1}U_{1,1}^{-1}$,

that is

$$[L]_{2,1}[U]_{1,2} = [A]_{2,1}U_{1,1}^{-1}L_{1,1}^{-1}[A]_{1,2} = [A]_{2,1}A_{1,1}^{-1}[A]_{1,2.}$$

Finally

$$\mathcal{S} = [A]_{2,2} - [A]_{2,1}A_{1,1}^{-1}[A]_{1,2.}$$

This matrix $\mathcal{S} \in \mathbb{R}^{n_2\times n_2}$ comes directly from the partition of matrix A; it is called **Schur complement** associated with this partition. Suppose then that we know how to factorize this matrix $\mathcal{S}$ into $\mathcal{S} = [L]_{2,2}[U]_{2,2}$ with $[L]_{2,2}$, $[U]_{2,2} \in \mathbb{R}^{n_2\times n_2}$ triangular regular matrices, ($[L]_{2,2}$ with ones on the diagonal) then we are at home, because

$$A = \begin{pmatrix} [L]_{1,1} & 0 \\ [L]_{2,1} & I_{n_2} \end{pmatrix} \times \begin{pmatrix} [U]_{1,1} & [U]_{1,2} \\ 0 & \mathcal{S} \end{pmatrix} = \begin{pmatrix} [L]_{1,1} & 0 \\ [L]_{2,1} & [L]_{2,2} \end{pmatrix} \times \begin{pmatrix} [U]_{1,1} & [U]_{1,2} \\ 0 & [U]_{2,2} \end{pmatrix}.$$

Let us summarize the factorization method of the matrix A: We suppose the first diagonal element $[A]_{1,1}$ not equal to zero, then we write $[A]_{1,1} = 1 \times [A]_{1,1}$ and we define $[L]_{1,1} = 1$ and $[U]_{1,1} = [A]_{1,1}$; that is to say that we consider a partition of matrix A with $n_1 = 1$ and $n_2 = n-1$. From the previous study, we get

$$[U]_{1,2} = L_{1,1}^{-1}[A]_{1,2} = [A]_{1,2} \text{ and } [L]_{2,1} = [A]_{2,1}U_{1,1}^{-1} = [A]_{2,1}/[A]_{1,1}$$

$$\mathcal{S} = [A]_{2,2} - [A]_{2,1} \times [A]_{1,2}/[A]_{1,1}$$

This later relation may also be written as

$$\forall_{i,j} \quad 2 \le i, j \le n \qquad \mathcal{S}_{i-1,j-1} = [A]_{i,j} - \frac{[A]_{i,1} \times [A]_{1,j}}{[A]_{1,1}}.$$

So only by using the elements of matrix A, we can start the factorization process:

-the first row of matrix A defines the block $[U]_{1,2}$ by $[U]_{1,2} = [A]_{1,2}$
- the first column of matrix A defines the block $[L]_{2,1}$ by $[L]_{2,1} = [A]_{2,1}/[A]_{1,1}$
- the Schur complement is defined by $\mathcal{S} = [A]_{2,2} - [A]_{2,1} \times [A]_{1,2}/[A]_{1,1}$.

Now it is obvious that the continuation of the process is only related to the factorization of the matrix $\mathcal{S}$, and this is possible if $\mathcal{S}_{1,1} \neq 0$! When this property is satisfied, the previous method is used to factorize the matrix $\mathcal{S}$ of order $n-1$ and so on... By denoting $A^{(k)}$ and $\mathcal{S}^{(k+1)}$ the successive matrices generated by the method (we define $A^{(0)} = A$), the different steps are summarized in the following algorithm

for $k = 1,.....n-1$ **do**
 if $A_{k,k}^{(k-1)} \neq 0$
 $[L]_{k,k} = 1, \quad [U]_{k,k} = A_{k,k}^{(k-1)}$
 for $i = k+1,.....n$ **do**
 $[L]_{i,k} = A_{i,k}^{(k-1)} / A_{k,k}^{(k-1)}$
 $[U]_{k,i} = A_{k,i}^{(k-1)}$
 for $j = k+1,....n$ **do**
 $A_{i,j}^{(k)} = A_{i,j}^{(k-1)} - \dfrac{A_{i,k}^{(k-1)} A_{k,j}^{(k-1)}}{A_{k,k}^{(k-1)}} = A_{i,j}^{(k-1)} - A_{i,k}^{(k)} A_{k,j}^{(k)}$
 $\mathcal{S}_{i-k,j-k}^{(k)} = A_{i,j}^{(k)}$
 end
 end
end

If a diagonal element $A_{k,k}^{(k-1)}$ appears to be equal to zero during the calculation, we look into column k of matrix $A^{(k-1)}$ for a non zero element; let $A_{i,k}^{(k-1)}$ be this element for some row index $i > k$, we exchange then rows i and k. This operation is equivalent to multiply the generic matrix $\mathcal{S}^{(k)}$ by a permutation matrix $P^{(k)}$:

$$P^{(k)}\mathcal{S}^{(k)} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathcal{S}_{k,k}^{(k)} & \cdot & \cdot & \mathcal{S}_{k,i}^{(k)} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathcal{S}_{i,k}^{(k)} & \cdot & \cdot & \mathcal{S}_{i,i}^{(k)} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} = \begin{pmatrix} \mathcal{S}_{i,k}^{(k)} & \cdot & \cdot & \mathcal{S}_{i,i}^{(k)} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathcal{S}_{k,k}^{(k)} & \cdot & \cdot & \mathcal{S}_{k,i}^{(k)} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

In the frame of the matrix of order n, this is equivalent to multiply matrix $A^{(k-1)}$ by a permutation $P(i,k) \in \mathbb{R}^{n \times n}$

$$P(i,k) = \begin{pmatrix} I_{n-k} & 0 \\ 0 & P^{(k)} \end{pmatrix}$$

and finally

$$P(i-k)A^{(k-1)} = \begin{pmatrix} I & 0 \\ 0 & P \end{pmatrix} \begin{pmatrix} [L]_{1,1} & 0 \\ [L]_{2,1} & I \end{pmatrix} \begin{pmatrix} [U]_{1,1} & [U]_{1,2} \\ 0 & \mathcal{S}^{(k)} \end{pmatrix}.$$

This multiplication does not modify rows of index less than k whose elements are known, but it changes the values of rows i and k in matrices L and $A^{(k)}$.

Suppose now, that we cannot find any non zero element $A_{i,k}^{(k-1)}$ in column k. What does it mean? It means that the current factorization has the following pattern

$$A = \begin{pmatrix} [A]_{1,1} & [A]_{1,2} \\ [A]_{2,1} & [A]_{2,2} \end{pmatrix} = \begin{pmatrix} [L]_{1,1} & 0 \\ [L]_{2,1} & I_{n_2} \end{pmatrix} \times \begin{pmatrix} [U]_{1,1} & [U]_{1,2} \\ 0 & \mathcal{S}^{(k)} \end{pmatrix}.$$

Then the first column of matrix $\mathcal{S}^{(k)}$ is null, and

$$\det A = \det[U]_{1,1} \det \mathcal{S}^{(k)} = 0.$$

This relation contradicts the assumption that matrix A is regular ! So we are assured to find at least one non zero element in column k.

When the complete process needs more than one permutation to be successfully concluded, the final relation is written as

$$P\ A = P(i_m, k_m)P(i_{m-1}, k_{m-1})...P(i_1, k_1)A = L\ U,$$

where the matrix P is again a permutation matrix, taking into account the history of all permutations performed during the factorization. So we have established the following result:

Let $A \in \mathbb{R}^{n\times n}$ be a regular matrix, there exists a permutation matrix $P \in \mathbb{R}^{n\times n}$, a lower triangular matrix $L \in \mathbb{R}^{n\times n}$ with unity values on the diagonal, and an upper triangular matrix $U \in \mathbb{R}^{n\times n}$ such that $P\ A = L\ U$.

-
-
-

Bibliography

[1] J.H. Bramble. (1993) *Multigrid methods*. Pitman Research Notes in Mathematics series 294,.

[2] M. Cross and O. Widlund C.H. Lai, P.E. Bjorstad. (1999)11th *International Conference on Domain Decomposition*. Domain Decomposition Press.

[3] P.G. Ciarlet.(1982) *Introduction à l'analyse numérique matricielle à l'optimisation*. Masson.

[4] A. George and W.H. Liu.(1981) *Computer solution of large sparse positive definite systems*. Prentice-Hall.

[5] W.McCormick, Multigrid methods.(1988) *Lectures Notes in Pure and Applied Mathematics*., 110.

[6] T. Chan and al. R. Barrett, M. Berry.(1991) *Templates for the Solution of linear systems: Building blocks for iterative methods*. SIAM.

[7] Y. Saad.(1996) *Iterative methods for sparse linear systems*. PWS Publishing Company.

[8] P. Wesseling.(1992) *An introduction to Multigrid Methods*. Wiley.

[9] D. Keyes and J. Xu.(1995) *Domain Decomposition Methods in Scientific and Engineering Computing*. Contemporary Mathematics N157. AMS.

Biographical Sketch

Pascal Joly was born in Amiens, France. He obtained his Ph.D. in Numerical Analysis from University Paris 6 (*Université Pierre et Marie Curie*). He is now a research engineer at the CNRS, the French National Research Council. His research is devoted mainly to studying algorithms from a theoretical and a practical point of view, in order to solve linear systems arising from various approximation schemes: finite differences, finite elements, and more recently, wavelet bases. This work is part of one of the major French research projects, such as the "Club Modulet", or European ones such as the TMR "Wavelets in Numerical Simulation". Part of this work is related to industrial applications and is carried out in collaboration with the engineers of industrial firms: EADS (fluid dynamics), Elf (oil reservoir simulation), and St Gobian (glass forming).
His expertise in this area is also applied to teaching in Doctoral Studies in Numerical Analysis at the University Pierre et Mary Curie. In this course, students learn all aspects of modern scientific computing and more specifically, the solution of physical problems on a computer. This teaching activity is also related to international programs such as TEMPUS projects.