In connection with the methods of decomposition of GAUSS type

Summary:

This document presents some related aspects to the method of decomposition of GAUSS.

After a fast presentation, we point out the main advantages and disadvantages related to this direct method. Then, we detail the implementation of algorithm LDL^T implemented in *Code Aster*.

GAUSS (1777 - 1855) is at the origin of all the direct methods of digital resolution of linear systems. That he is thanked here for it.

Contents

1 General information on the methods of the GAUSS type

1.1 Presentation of the method.

On the basis of the observation which it is easy to solve the system $A.x=b$ when A is a triangular matrix lower or higher, one seeks to break up the initial matrix full by a factorization with triangular matrices.

The basic principle is to search a regular matrix P , known as matrix of permutation, such as the product $P.A$ that is to say triangular, then to solve $P.A.x=P.b$

Note:

In practice, P is determined by elementary products of matrices of permutation $P=P^{(k)}...P^{(1)}$.
Matrices $P^{(i)}$ depend on the alternative chosen, but one never calculates explicitly the matrix P but only $P.A$ and $P.b$

The matrix With being factorized in the general form L.U (L lower triangular matrix, U higher triangular matrix), we are brought to solve the two linear systems:

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

Note:

*In the method known as **of elimination of GAUSS** , one carries out simultaneously the factorization of A and the resolution of $L.y=b$*

The following algorithm carries out the elimination of GAUSS and the resolution of $L.y=b$

at the stage (p+1) we have

$$a_{ij}^{(p+1)}=a_{ij}^{(p)}-a_{ij}^{(p)}.\left(a_{pp}^{(p)}\right)^{-1}.a_{pj}^{(p)} \quad \text{for} \quad \begin{matrix} p+1\le i\le n \\ p+1\le j\le n+1 \end{matrix}$$

$$a_{ij}^{(p+1)}=a_{ij}^{(p)} \quad \text{for} \quad \begin{matrix} 1\le i\le p \\ 1\le j\le n+1 \end{matrix}$$

$$a_{ij}^{(p+1)}=0. \quad \text{for} \quad \begin{matrix} p+1\le i\le n \\ 1\le j\le p \end{matrix}$$

Note:

In this writing of the algorithm of elimination of GAUSS, the second member B is regarded as an additional column of the matrix which is then treated like a matrix N X (n+1) .

1.2 Concept of pivot

The preceding implementation of the algorithm of elimination of GAUSS supposes implicitly that the called diagonal terms **pivots** are not worthless in the course of calculation.

Case of worthless pivots : A regular matrix can have a null pivot.

example: the matrix $\begin{pmatrix} 0 & 1 \\ 2 & 1 \end{pmatrix}$

To cure this disadvantage, one can use one **strategy of swivelling**

- **complete swivelling** : this strategy consists in choosing like pivot, the major term in the remaining block then to carry out a permutation of line and column.

  There is then the system $P.A.Q(Q^T x)=P.b$
  where P is the matrix of permutation of the lines and Q that of the columns.

  The found solution is then $y=Q^T x,$ and it is thus necessary to preserve the matrix of permutation Q to obtain the sought solution $x=Q.y$

- **partial swivelling** : the pivot is required as being the term of maximum value, among the not yet treated terms, in the current column (the kth one at the stage K) then one carries out a permutation of line.

1.3 Stability of the method

Definition: A digital method of resolution of system linear is known as mathematically **stable** when “some is the matrix With regular, the algorithm succeeds”.

Theorem: The method of GAUSS with a strategy of swivelling is mathematically stable for any regular matrix.

Corollary: So during a factorization of GAUSS with swivelling, a null pivot is then detected the matrix is singular and this system does not have a single solution.

Theorem: The method of GAUSS (without swivelling) is stable for positive definite real matrices.

For more details, one will consult the basic books which are [bib13] [bib14] [bib6].

1.4 Unicity of the decomposition

Proposal: The decomposition of GAUSS is not single,
but if one specifies the diagonal of L or of U then there is unicity of the decomposition.

1.5 An alternative: the factorization of CROUT

The method of factorization of CROUT [1] is the same algorithm, which requires the same number of operations and carries out the same filling of the matrix but calculations are carried out in a different way.

We place ourselves if the matrix is factorisable, which is always the case with a permutation close to the lines and the columns since the matrix is regular: one thus has $A=LU$

Then one proceeds by identification

$$\begin{cases} \text{pour } i \le j \quad a_{ij} = u_{ij} + \sum_{k=1}^{i-1} l_{ik} . u_{kj} \\ \text{pour } i > j \quad a_{ij} = \sum_{k=1}^{j} l_{ik} . u_{kj} \end{cases}$$

(l_{ij} and u_{ij} are the elements of L and U)

from where values of U_{ij} and L_{ij} according to has_{ij}

$$u_{1j} = a_{1j} \qquad j = 1, \ldots, n$$

$$l_{i1} = a_{i1} / u_{i1} \qquad i = 1, \ldots, n$$

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} . u_{kj} \qquad i \le j$$

$$l_{ij} = \frac{1}{u_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} . u_{kj} \right) \qquad i > j$$

Note:

The order of calculations is not arbitrary, it is necessary to know them L_{ik} located on the left and them U_{kj} with the top of each term to be calculated.

One sees whereas at the kth stage, one defers on the kth line all the former contributions leaving unchanged the lines k+1 with N.

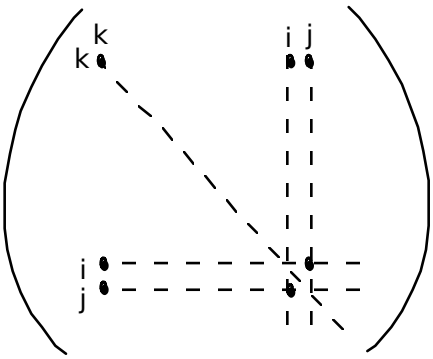
This alternative of CROUT is also called elimination of GAUSS by columns (or columm activates), and prévilégie the scalar operation of product.

1.6 Case of the symmetrical matrices

Proposal: The decomposition of GAUSS respects symmetry.

It is enough to note that with each stage the terms a_{ij} and a_{ji} receive the same contribution.

Indeed:



by assumption of recurrence, one supposes the symmetrical matrix at the stage K and consequently one a:

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - a_{ik}^{(k)} a_{kj}^{(k)} / a_{kk}^{(k)}$$
$$a_{ji}^{(k+1)} = a_{ji}^{(k)} - a_{jk}^{(k)} a_{ki}^{(k)} / a_{kk}^{(k)}$$

from where the proposal.

Consequently, a matrix With symmetrical perhaps factorized in the form $A = L D L^T$

where D is a diagonal matrix and L a unit lower matrix (i.e. with unit diagonal)

This decomposition, single since a diagonal was fixed, applies to any nonsingular symmetrical matrix.

If the matrix A is definite positive, then the terms of the diagonal are strictly positive and one can use the form known as of CHOLESKY $A = L L^T = (L D^{1/2} . D^{1/2} L^T).$

Let us notice that the decomposition of CHOLESKY requires N extractions of square root (which is an expensive operation in time).

In the case of a factorization LDL^T for symmetrical matrices, we can write the algorithm of CROUT in the following form:

Boucle sur les colonnes ic=2,...,n
Boucle sur les contributions im=1, ¼,ic−1
$l_{ic,ic} \neg l_{ic,ic} - l_{ic,im} * l_{im,ic}$
Fin boucle
Boucle sur les lignes il=1,...,ic−1
Boucle sur les contributions im=1,... ,il−1
$l_{il,ic} \neg l_{il,ic} - l_{ic,im} * l_{im,il}$ ¿
Fin boucle
$l_{il,ic} \neg l_{il,ic} / l_{il,il}$
Fin boucle

2 Disadvantages of the methods of the GAUSS type

The disadvantages of the methods of the GAUSS type are primarily of three types:

1) a high number of operations
2) a filling of the matrix
3) a loss of precision in the course of calculation

The first two points are often qualified major defects whereas third is regarded as a minor defect.

2.1 The number of operations

For a system full with size N, at the p-ième stage, we must carry out to calculate the new coefficients of the matrix and the second member:

- (Np) divisions
- (n-p+1) (Np) additions and multiplications

The number of operations is thus:

$\sum_{p=1}^{n-1}(n-p)=\frac{n(n-1)}{2}$ divisions

$\sum_{p=1}^{n-1}(n-\mathrm{p}+1)(n-p)=\sum_{q=1}^{n-1}q^2+\sum_{q=1}^{n-1}q=\left(\frac{n(n-1)(n-2)}{6}+\frac{n(n-1}{2}\right.$ additions and as many multiplications.

That is to say $\frac{1}{3}\ n(n-1)\ \left(n+\frac{1}{2}\right)$ operations for which it is advisable to add them n^2 operations of the resolution of the triangular system.

In short: For a full great system, the algorithm of GAUSS requires about $\frac{1}{3}n^3$ operations.

Note:

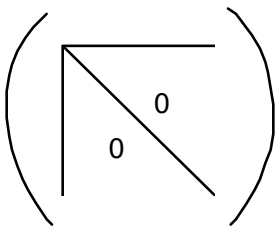
In the case of a stored matrix band, the number of operations is $n.b^2$ where b is the bandwidth.

2.2 Filling of the matrix

Let us start with a classical example of matrix known as “marks with arrows” that one meets for example in chemistry [bib9].

That is to say A the matrix such as $a(1,i)\neq 0, a(i,1)\neq 0, a(i,i)\neq 0$ and all its other terms are worthless, the matrix takes the following form then:

With =

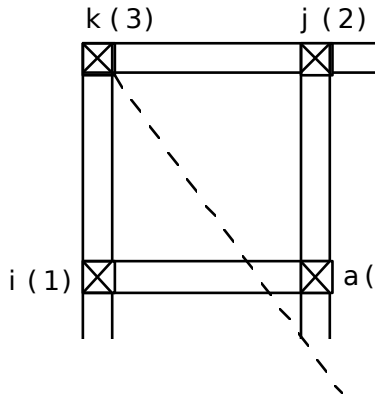


After the 1st stage of factorization (by the algorithm of GAUSS), the matrix is full with the direction where there are no more theoretically worthless terms.

More formally let us look at the phenomenon of filling of to the algorithm; for that let us récrivons the algorithm:

For K varying of 1 to N - 1 to make % buckles on the stages
For I varying of K + 1 to N to make % buckles on the lines
For J varying of K + 1 to N to make % buckles on the columns
$a(i,j)=a(i,j)-a(i,k).a(k,j)/a(k,k)$
end to make
end to make
end to make

What one can schematize graphically, at the kth stage, for the calculation of the term has (I, J) by:




```
a(i,j) = a(i,j) - (1)*(2)/(3)
```

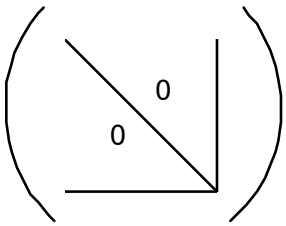
The term $a(i,j)$ is nonnull at the end of the kth stage:

- if it were nonnull at the beginning of this kth stage,
- or if terms $a(i,k)$ and $a(k,j)$ are all two the nonworthless ones at the beginning of the kth stage, and this independently of the initial value of the term $a(i,j)$.

Moreover, it is seen that the method of GAUSS fills the profile wraps matrix during stages of factorization.

In the example of the matrix “marks with arrows”: the profile envelope is the full matrix, from where the noted result.

This example highlights the importance of the classification of the unknown factors of the matrix since the matrix can be récrite, after permutation of the unknown factors, in the form:



whose profile envelope is the matrix it even (there is thus no filling).

We have just seen the importance of the classification of the unknown factors.

We could not insist too much on the fact that algorithms of “optimal” renumerotation must be used to minimize the filling of the matrix.
These algorithms rest on the heuristic ones and are specialized.

Among the algorithms most usually used let us quote

Algorithms	Objectives
CUTHILL - Mc KEE	to minimize the bandwidth
Reverse CUTHILL - Mc KEE	to minimize the profile
Minimum Degree	to minimize the multiplications by 0

Intrinsic formulation of the filling

One can give an intrinsic formulation of the filling during the elimination of unknown factor in terms of graph [bib5].

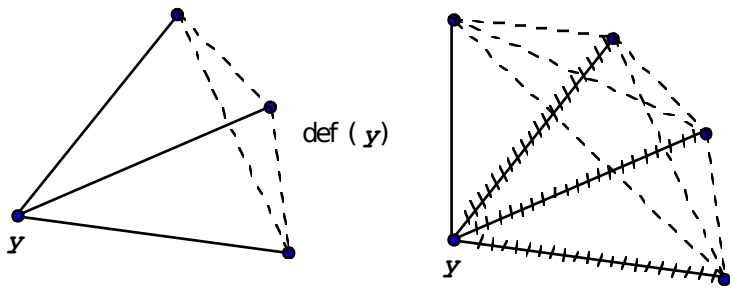
That is to say a matrix `With` to which we associate the graph $G(X,E)$, where X is the whole of the nodes and E the whole of the not directed edges.

The problem of elimination of an unknown factor of the matrix is then equivalent to eliminate a node from the graph.

Definition: Are $x,y \in X$, it will be said that x and y are **adjacent** if and only if $x,y \in E$

If Y is a subset of nodes of $X(X \supset Y)$ we can define the following units:

the whole of the adjacent nodes with `Y`	$Adj(Y)=x/x \in X-Y \text{ et } \exists\ y \in Y \text{ tel que } x,y \in E$
the whole of with dimensions incidents with `Y`	$Inc(Y)=x,y/y \in Y, x \in Adj(Y)$
the whole of definition of `X`	$\mathrm{Def}(X)=y,z/y,z \in Adj(X), y \neq z \text{ et } z \notin Adj(Y)$



- en hachuré : ce que l'on élimine,
- en pointillé : ce que l'on rajoute (remplissage)

Operation of elimination of `there`

The elimination of `there` then consist in considering the subset

$$Elim(y,G)=X-y,(E-Inc(y))U \overset{\text{def}}{=} (y)$$

It is thus necessary well to consider the filling which is related to $\mathrm{Def}(y)$.

To minimize the filling one can use heuristics consisting in eliminating it `there` of minimal degree, the degree of {`there`} being the cardinal of the Adj unit (`there`). It is the governing idea of the use of the algorithm of the minimum degree before a factorization of the GAUSS type.

This approach is used in the multi-frontal method put in work in *Aster* [bib15].

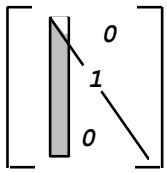
2.3 The loss of precision in the course of calculation

The problem comes owing to the fact that in the course of algorithm the pivots decrease and that they are used as denominator for the following stages [bib13].

2.3.1 Study summary of the loss of precision

Let us note $A^{(k)}$ the matrix at the stage K (i.e. after the elimination of the kth variable); with by convention $A^{(o)}=A$.

We can then write that With(1) check $L^{(1)}.A^{(1)}=A^{(0)}$

by taking $L_{(1)}$ = 

by identification

Then after (n-1) factorizations of this type we obtain

$$(L^{(n-1)}\bullet\ldots\bullet L^{(1)})U=L.U=A^{(0)}$$

Because of the errors E on the decomposition, it is advisable to write: $L.U=A^{(0)}+E$

Coefficients of the matrix of error E can be evaluated [bib12] by taking account of the mistake made on the floating operation which we will note fl :

$$|e_{ij}|\leq 3\,.\,\varepsilon\,.\,m_{ij}.\max_{k,i,j}|a_{ij}^{(k)}|\ \forall_{\ i,j}\quad \forall i,j$$

with m_{ij}: many terms such as $a_{ik}^{(k)}.a_{kj}^{(k)}\neq 0$
ε : relative error of the operations machine

Indeed, while placing itself if the elimination of GAUSS “succeeds” (for example: the matrix is definite positive or one uses a technique of swivelling).

Let us note $\eta_{ik}=\beta\left(a_{ik}^{(k-1)}\Big/a_{kk}^{(k-1)}\right)\equiv\left(a_{ik}^{(k-1)}\Big/a_{kk}^{(k-1)}\right).\left(1+\varepsilon_1\right)$ for $i>k,k=1,\ldots,n-1$

with $|\varepsilon_1| < \varepsilon$

The term $a_{ij}^{(k)}$ is then evaluated by:

$$a_{ij}^{(k)}=\beta\left(a_{ij}^{(k-1)}-\eta_{ik}.a_{kj}^{(k-1)}\right)\quad \text{for}\quad i,j>k,k=1,\ldots,n-1$$

That is to say still $a_{ij}^{(k)}=\beta\left(a_{ij}^{(k-1)}-\eta_{ik}.a_{kj}^{(k-1)}\right)$ with $|\varepsilon_2|,|\varepsilon_3| < \varepsilon$

The “disturbance” e_{ij}^{k} undergone by $a_{ij}^{(k)}$ can then be evaluated; starting from the definition of m_{ij} we deduce the relation from it:

$$|e_{ij}^{k}|\equiv|\varepsilon_1.a_{ij}^{(k-1)}|\leq\varepsilon.|a_{ij}|\ \text{ for }\ i>k,k=1,\ldots,n-1$$

and of the evaluation first of $a_{ij}^{(k)}$ we deduce:

$$|\mu_{ik}.a_{kj}^{(k-1)}|\equiv\left(a_{ij}^{(k-1)}-a_{ij}^{(k)}/\left(1+\varepsilon_3\right)\right)/\left(1+\varepsilon_2\right)$$

and finally

$$|e_{kj}^{(k)}|\equiv|a_{ij}^{(k)}\left(1-\frac{1}{\left(1+e_2\right)\left(1+e_3\right)}\right)-a_{ij}^{(k-1)}\left(1-\frac{1}{\left(1+e_2\right)}\right)|$$

$$|e_{kj}^{(k)}| \leq 3.|a_{ij}|.\varepsilon$$

however decomposition $L.U = A(0) + E$ we indicates that e_{ij} is the sum of the errors.

2.3.2 Estimate of the error on the solution

We solved the system

$$L.U\,x = A(0).x + E.x$$

where $E.x$ is the term of error induced by the errors rounding/truncation in the operations of factorization.
$A(0)x$ is the second member (in fact b).

The found approximate solution $\tilde{x}$ who approximates the true solution X is:

$$\tilde{x} = (A+E)^{-1}b$$

One shows whereas the evaluation of the error on X is related to **conditioning of** With.

Let us pose the problem in the form: $(A+\delta A)(x+\delta x)=b$

While supposing $\delta A.\delta x$ small one a: $\delta x = A-1.\delta A.x$

from where while normalizing $\frac{\|\delta x\|}{\|x\|} \le Cond(A).\frac{\|dA\|}{\|A\|}$

where $Cond(A)$ = $\left(\|A\|.\|A\|^{-1}\right.$ is the conditioning of the matrix With.

Note:

It is noted that the error induced on the second member is weak and the solution disturbs only through one bad conditioning of the matrix With .

Indeed, if the system is considered $A(x+\delta x)=b+\delta b,$

because of the equalities: $\delta x = A-1.\delta b\ et\ Ax=b$

on a $\|\delta x\| \le \|A^{-1}\|.\|\delta b\| \text{et} \|b\| < \|A\|.\|x\|$

et donc $\|\frac{\delta x}{x}\| \le \left(\|A\|.\|A^{-1}\|\right)\|\frac{\delta b}{b}\|$

Note:

If one considers the variations on $A, x,$ *and* B *at the same time, there is the following estimate [bib14]:*

$$\frac{\|\delta x\|}{\|x\|} \le \frac{Cond(A)}{1-\|A^{-1}\|.\|\delta A\|}.\left(\frac{\|\delta A\|}{\|A\|}+\frac{\|\delta b\|}{\|b\|}\right)$$

Some remarks on conditioning

- conditioning is defined only for one regular matrix,
- conditioning depends on the standard chosen on $\mathbb{R}^n$,
- some is the standard chosen, we have $1 \le Cond(A)$ and a matrix of as much is conditioned better than its conditioning is close to 1.

If the standard **Euclidean** is selected for standard, then

- the conditioning of a matrix `With` unspecified is

$$Cond(A) = |\frac{\mu_n}{\mu_1}|$$

where μ_1 and μ_n are the extreme singular values of `With` (i.e. smallest and largest of the eigenvalues of $A^*.A$).

- If the matrix `With` is symmetrical (or square) then

$$Cond(A) = |\frac{\lambda_n}{\lambda_1}|$$

where λ_1 and λ_n are the eigenvalues of minimal and maximum module of A .

2.3.3 Estimate amongst significant figures of the solution

If one has a precision of `p` figures (decimal) significant, one has then:

$$\frac{\|dA\|}{\|A\|} \simeq 10^{-p}$$

If one wishes a precision of S figures (decimal) significant on the solution

$$\frac{\|\delta x\|}{\|x\|} \leq 10^{-s}$$

from where the estimate amongst exact decimal significant figures of the solution

$$s \geq p - \log_{10}(Cond(A))$$

2.3.4 Method to reduce conditioning

The simplest method is that of the scaling:

One "passes" from `With` with $\Phi_1.A.\Phi_2$ with Φ_i diagonal matrix such as `Cond`($\Phi_1.A.\Phi_2$) that is to say better than $Cond(A)$.

This is very theoretical and there does not exist universal method to determine $\Phi_1 et \Phi_2$.

Let us note that if the matrix `With` is symmetrical and that one wishes to preserve this property, it is then necessary to take $\Phi_1 = \Phi_2$.

2.3.5 Example of badly conditioned matrix.

This example, very significant and instructive is with R.S. WILSON.

That is to say the system $Ax = b$ with:

$$A = \begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \text{ and } b = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix} \text{ and whose solution is } x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

- If one disturbs the second member of about 0.5% while taking:

  $\tilde{b} = \left(32.1,\quad 22.9,\quad 33.1,\quad 30.9\right)$

  then the solution is: $\tilde{x} = \left(9.2,\quad -12.6,\quad 4.5,\quad -1.1\right)$.

- If the matrix is disturbed of about 1%:

$$\tilde{A} = \begin{pmatrix} 10. & 7 & 8.1 & 7.2 \\ 7.08 & 5.04 & 6 & 5 \\ 8 & 5.98 & 9.89 & 9 \\ 6.99 & 4.99 & 9 & 9.98 \end{pmatrix}$$

  then the solution is $\tilde{x} = \left(-81,\quad 137,\quad -34,\quad 22\right)$

Remarks on the properties of the matrix `With` :

- *It is symmetrical, definite positive, of determinant `1` and of "nice" reverse.*

$$A^{-1} = \begin{pmatrix} 25 & -41 & 10 & -6 \\ -41 & 68 & -17 & 10 \\ 10 & -17 & 5 & -3 \\ -6 & 10 & -3 & 2 \end{pmatrix}$$

- Its conditioning within the meaning of the euclidian norm is:

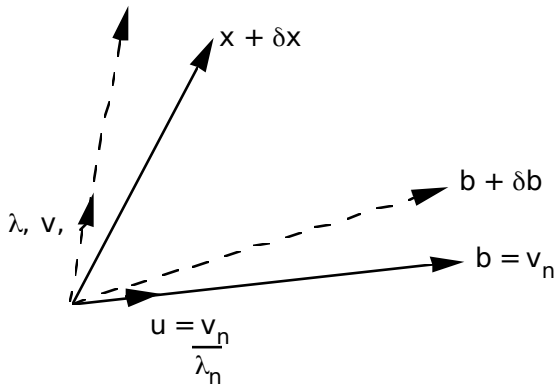
$$\mathrm{Cond}(A) = \frac{l_4}{l_1} = \frac{30.2887}{0.01015} = 2984.11$$

2.3.6 A geometrical Interpretation of the bad conditioning

One can give a very simple interpretation of the bad conditioning of a linear system $Ax = b$ in the typical case where `With` is normal (`i.e.` $A^*.A = A.A^*$).

Are λ_1 the smallest eigenvalue of the matrix A and λ_n its greater eigenvalue and are v_1 and `v_N` associated clean vectors.

for $b = v_n$ et δb, one has $\|\frac{\delta x}{x}\| = \mathrm{cond}(A)\|\frac{\delta b}{b}\|$



from where if $cond(A) = |\frac{\lambda_n}{\lambda_1}|$ is large, a small disturbance δb of b involve a great variation on the solution v .

2.4 Criteria of determination of a null pivot

Definition: One **numerically degenerated system** is a system for which a pivot is null or does not have exact significant figure.

Let us note that a system can be degenerated numerically without the being mathematically.

In these two cases, it is advisable not to solve the system from where need for determining a criterion of stop as soon as one of the pivots does not have any more exact significant figures.

Are `With` the matrix to be factorized and `F` the matrix of free diagonal resulting from factorization.

Criterion 1 : The simplest criterion is to consider that the pivot is null as soon as it is lower, in absolute value with a given threshold.

$$|f_{ii}| < \varepsilon_1$$

ε_1 is a “small number” in lower part of which it is considered that the values are arbitrarily worthless.

Criterion 2 : This criterion applies to the number of significant figures still available. By noting that one cannot have any more p significant figures on a given machine, one will check that the decrease of the pivot is not carried out in a report higher than 10^{-p}.

$$|\frac{f_{ii}}{a_{ii}}| \leq \varepsilon_2 = 10^{-p}$$

Let us note that the report $|\frac{f_{ii}}{a_{ii}}|$ is always lower or equal to 1 because of the reduction of the pivots.

The basic cause of a bad digital conditioning is the rounding error caused by the introduction of great numbers without physical significance.

3 Method LDL^T by blocks put in work in *Aster*

This paragraph details the implementation in *Aster* resolution of the linear system `A.x = B` by the method of factorization LDL^T symmetrical matrix `With`.

The matrix `With` is stored in profile (or line of sky) per block.

Basic principle: A column of the matrix is contained very whole in only one block: we do not segment the columns.

The tables allowing the description of the stored matrix profile per block are:

- `HCOL` height of column of the matrix
  `HCOL (I)` height of the i-eme column
- `ADIA` address of the diagonal term in its block
  `ADIA (I)` return the address of the i-eme diagonal term in its block
- `ABLO` pointer of block
  `ABLO (i+1)` return the number of the last equation in total classification contents in the i-eme block
  By convention `ABLO (1) = 0` and the number of equations in the i-eme block is given by the relation `ABLO (i+1) - ABLO (I) + 1`
  The full number of equation results as being `ABLO (nombre_de_bloc + 1)`

It is also necessary to memorize the full number of blocks used to contain the coefficients of the matrix.

Note:

Formally the table `HCOL` is useless because he results from the tables `ADIA` and `ABLO`, but it makes it possible to carry out calculations more quickly.

3.1 Implementation of factorization

Principal characteristics of the implementation of the factorization of GAUSS by the alternative of CROUT in form LDL^T of a stored symmetrical matrix profile per block are:

- factorization is carried out in place, i.e. crushing the initial matrix,
- perhaps partial factorization,
- the criteria of worthless detections of pivot can be adapted to the factorization of quasi-singular matrices,
- in the event of detection of null pivot, this pivot is replaced by a very great value (10^{40}) what amounts introducing a condition of blocking by penalization.

Note:

Two tables of work are created:

- *a table which will contain the diagonal of the factorized matrix (minimization amongst access to the block),*
- *a table for the current column (minimization amongst calculations carried out).*

```
Algorithm BEGINNING;
    creation of an intermediate table for the current column
    creation of an intermediate table for the current column

    FOR  ibloc VARIANT_DE 1 A nombre_de_bloc TO MAKE
```

- Determination of end and the starting columns for the current block.
- Research of the smallest equation in relation to an equation contained in the current block. This research is done by exploiting the table `HCOL`
- Research of the block of membership of the equation found previously. This research is done by exploiting the table `ABLO`.
- Request in mode `writing` i-eme block.

```
    FOR jbloc VARIANT_DE plus_petit_concerne A ibloc-1 TO MAKE
           Request in mode reading j-eme block
           FOR  iequa  CONTAINED IN the i-eme block TO MAKE
               calculation of the start address of the column in the block
               calculation height of the column
               FOR jequa CONTAINED IN the j-eme block TO MAKE
                   calculation of the start address of the column in the block
                   calculation length of the column
                   With (ibloc, jequa) = A (ibloc, jequa) - < with (ibloc,
                   *), A (jbloc, *) >
               FIN_POUR
           FIN_POUR
           release  j-eme block which was not modified
    FIN_POUR

    FOR  iequa  CONTAINED IN the i-eme block TO MAKE
        calculation of the start address of the column in the block
        calculation length of the column
        FOR  jequa  CONTAINED IN the i-eme block and < iequa TO MAKE
            calculation of the start address of the column in the block
            calculation height of the column
            With (ibloc, lm) = A (ibloc, lm) - < with (ibloc, *), A (jbloc,
            *) >
        FIN_FAIRE

        % use of the column iequa (calculation of the pivot)
        calculation of the start address of the column in the block
        calculation height of the column
        safeguard of the column: wk. (I) ¬ With (ibloc, I)
        standardisation of the column by using the diagonal table:
            With (ibloc, *) ← With (ibloc, *)/diag (*)
        calculation of the diagonal term and actualization of the table of work:
            tabr8 (iadia) = tabr8 (iadia) - < with (ibloc, *), wk. (*) >

        test of the pivot compared to ε
        test of the pivot compared to the number of significant figures

    FIN_POUR

    release of the block running which one has just modified
    FIN_POUR

    release of the tables of work
FINE Algorithm;
```

Determination of end and the starting column for the current block.

This phase is due to the concept of partial factorization.

```
IF (last column of the block < beginning of factorization) THEN
   request in reading mode of the i-eme block
   to fill the table with work containing the diagonal.
   release of the i-eme block
   ALLER_AU following block
SINON_SI (first column of the block > fine of factorization) THEN
   TO LEAVE
IF NOT
   IF (first column of the block < beginning of factorization) THEN
      % to supplement the "diagonal" table
      request in reading mode of the i-eme block
      to fill the table with work containing the diagonal.
   FIN_SI
   IF (last column of the block > fine of factorization) THEN
      modification of the last equation to be taken into account
   FIN_SI
FIN_SI
```

Notice on the obstruction:

It is obligatory that one can have at least simultaneously in memory:
- *two blocks of the matrix,*
- *two vectors of work of size: the number of equations of the system to be solved.*

3.2 Implementation of the resolution

The implementation of the simultaneous resolution of N second members of the system A.x = B, where the matrix With is symmetrical and was factorized in form L D L^T (the resolution is in place)

Notice :

One creates a table of work which will contain the diagonal of the factorized matrix, in order to minimize the access number to the block of the matrix and thus to limit the readings for the large matrices not being able to reside completely in memory.

```
Algorithm BEGINNING;
    Creation of a table to store the diagonal to avoid readings at the time of the stage of diagonal
    resolution.
    FOR ibloc VARIANT_DE 1 WITH the nombre_de_bloc        % downward resolution and
                                                          % filling of the diagonal table
        request in reading mode of the i-eme block
        FOR iequa contained IN the BLOCK
            Calculation height of the column and
            calculation of the start address of the column in its block
            FOR each second member TO MAKE
                xsol (isol) = xsol (isol) - < X (isol), U) >
    FIN_POUR
    safeguard of the diagonal term in the table of work
    FIN_POUR
    release i-eme block
FIN_POUR

FOR each second member TO MAKE        % diagonal resolution
    FOR all the equations TO MAKE
        xsol (iequa, isol) = xsol (iequa, isol)/diag (iequa-1)
    FIN_POUR

    FOR ibloc VARIANT_DE nombre_de_bloc to 1 PAR_PAS_DE -1% going up resolution
        request in reading mode of the i-eme block
        FOR iequa contained IN the BLOCK
            Calculation height of the column and
            calculation of the start address of the column in its block
            FOR each second member TO MAKE
                xsol (ixx+i, isol) = xsol (ixx+i, isol) - xsol (isol) *L
(ide+i)
            FIN_POUR
        FIN_POUR
        release i-eme block
    FIN_POUR

Release working area (i.e. diagonal table)
FINE Algorithm;
```

Notice on the obstruction:

It is necessary that one can have simultaneously in memory:
- *a block of the matrix,*
- *a vector of work of size: the number of equations of the system to be solved,*
- *N second members.*

3.3 Scaling

It is possible to make a setting at the level of the matrix be factorized; this setting on the straight ladder is made in order to obtain a matrix whose diagonal terms are worth 1.

The diagonal matrix is such as:

$$\Phi_i = \begin{cases} \frac{1}{\sqrt{|a_{ii}|}} & si\ a_{ii} \neq 0 \\ 1 & si\ a_{ii} = 0 \end{cases}$$

It should be noted that at the time of the resolution, one obtains the solution of the system only after déconditionnement.

Indeed : the initial system is $Ax = b$

after multiplication on the left by Φ , one a:

$$\Phi Ax = \Phi . b$$

However the solved system is:

$$\Phi A \Phi x = \Phi b$$

from where the solution Φx obtained that it is necessary “to decondition”.

3.4 Tests on the pivot

Two criteria of detections of worthless pivot are implemented:

- the test in absolute value $|a_{ii}| < \varepsilon$, with ε given,
- the test in relative value on the number of exact significant figures.

Let us note that these tests can be reduced to their more simple expression by providing one $\varepsilon = 0$. and by giving, by convention, a number of exact significant figures no one.

This option is made necessary by the fact that algorithms such as the algorithms of search for eigenvalues [R5.01.01] [R5.01.02] seek to factorize matrices quasi - singular.

3.5 Factorization of complex matrices

The algorithm implemented in *Aster* also allows to treat the matrices **symmetrical** with complex coefficients.

The implemented algorithm does not treat the square matrices, although it is theoretically possible.

Annexe 1Methods of classical storage

A1.1 Full matrix

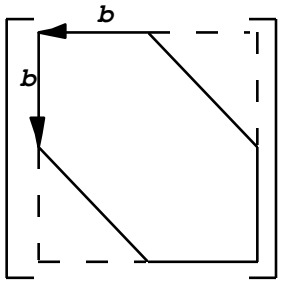
A nonsymmetrical matrix full with size N have N^2 coefficients.

If the matrix is symmetrical, one can store only his triangular the lower or higher is $\frac{n(n+1)}{2}$ values.

No table of description of the matrix is necessary.

A1.2 Matrix bandages

B + 1 = bandwidth



In this case one stores the band (called sometimes rectified matrix) in a rectangular table N (2b + 1) ; they then are included B (B + 1) zero values corresponding to the complements of the points.

In the case of a symmetrical matrix, one can only store N (B + 1) values of which $\frac{b(b+1)}{2}$ zero values (useless).

This method only requires to know the bandwidth.

A1.3 Matrix profile or matrix with line of sky

This technique consists in storing the terms of the matrix by columns and lines variable lengths. The terms external with the “line of sky”, which is the envelope of the tops of the columns being supposed not to have any contribution in calculations, are not stored.

Profile of the i-ème line (resp. column) is determined by:

$$\min \{J \text{ such as } 1 \leq J \leq N \quad has_{ij} \neq 0\}$$
$$(\text{resp } \min \{J \text{ such as } 1 \leq J \leq N \quad has_{ji} \neq 0\})$$

If the profile is symmetrical, one speaks about matrix with symmetrical profile.

This method of storage requires tables of storage which we will detail in the case of a matrix with symmetrical profile.

Classically, in this option of storage, the matrix is arranged in the shape of a dimensioned mono table requiring a table of pointer of entry of column ADIA to explore the matrix: the entry is done by the diagonal terms, the number of terms of the column is obtained by differences in two successive terms: ADIA (i+1) - ADIA (I).

If the matrix is nonsymmetrical, but with symmetrical profile, it is necessary to store the i-ème line and the i-ème column. Classically, one them mets “ends with ends” and the number of terms of the column or the line are (`ADIA (i+1) - ADIA (I))/2.`

A1.4 Storage per block

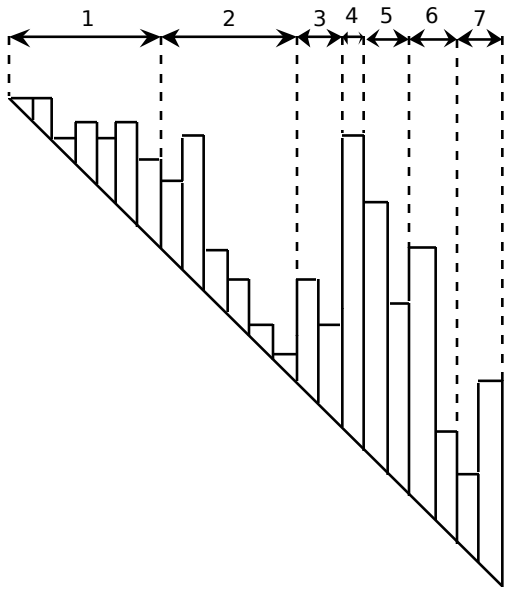
The methods of storage seen previously suppose implicitly that the matrix can reside in main memory, which is not always the case.

From where concept of matrix stored per block (or segmented on disc).

All preceding storages can be segmented, but we will state only the case of the stored symmetrical matrix profile.

Matrix profile stored per block

We consider here only the case of the symmetrical matrices, which does not remove anything with the general information matter.



A1.4-a figure: Maximum size of a block: 20 elements

In this example, we suppose of the same blocks cuts, to use blocks of variable size, it is necessary to introduce an additional table containing the size of each block.

We also consider that a column can belong only to one block: “we do not cut the columns”.

To manage the matrix, it is always necessary to know the address of the diagonal terms; but now, this address is relating to the block of membership of the column.

This table, it is necessary to join a table giving the equations contained in a block.

This table dimensioned with the number of block more `1` the last equation of the block contains.

Annexe 2Variations on the algorithm of GAUSS

As we saw with the alternative of CROUT, there exist several implementations of the algorithm of GAUSS: it consists in carrying out calculations of the coefficients in a different order.

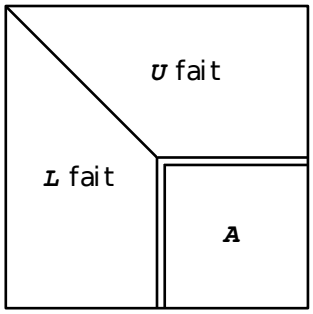
Schematically, it is considered that there are three overlapping loops:

- buckle I on the lines,
- buckle J on the columns,
- buckle K on the stages.

The standard algorithm is characterized by the sequence kij, but there exist 5 other permutations of the indices which cause as many alternatives (or of algorithms).

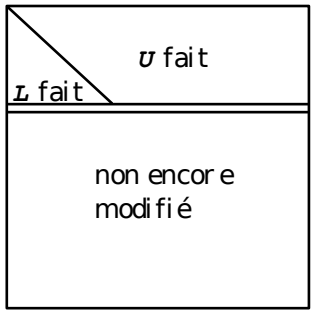
- the algorithm of CROUT is characterized by the sequence jki,
- the algorithm corresponding to the sequence ikj, which works by line, is known under the name of algorithm of “Doolitle”.

Let us give here a chart drawn from [bib10].



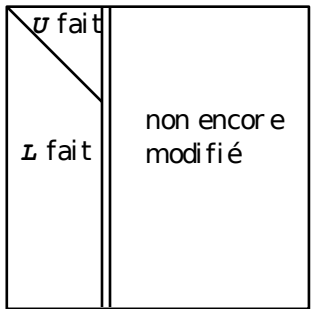
algorithmes kij - kji

on calculates the k-ième colonne et la k + 1 ligne et l’on réactualise la sous-matrice A



algorithmes ikj - ijk

on calculates the i-ième ligne de L et U



algorithmes jki - jik

on calculates the j-ième ligne de L et U

4 Bibliography

[1] P.D. CROUT A shorts method for evaluating determining and solving systems of linear equations with real gold complex coefficients. - AIEE Trans Flight 60.1941, pp 1235 - 1240

[2] E. CUTHILL & J. Mc KEE “Reducing the band with of sparse symmetric matrices” Proc 24th Nat Conf Assoc Comput Mach ACM Publ (1969)

[3] E. CUTHILL “Several strategies for reducing the bandwith of the matrices” in Sparse Matrices and to their applications - D.J. ROSE & R.A. WILLOUGHBY Editeurs, Plenum Near, New York (1972) pp 157 - 166

[4] G. Von FUCHS, J.R. ROY, E. SCHREM broad Hypermatrix solution of sets of symmetric positive definite linear equations - Computer methods in Applied Mechanics and Engineering (1972)

[5] A. GEORGE, minimum D.R. Mc INTYRE “One the application of the dismantles algorithm to finite element systems” SIAM J. Num. Year. Flight 15, (1978) pp90 - 112

[6] G.H. GOLUB and c.f. Van LOAN Matrix computations. Johns Hopkins University Close - Baltimore (1983)

[7] B. Mr. WILL GO Roundoff criteria in direct stiffness solutions - AIAA Newspaper 6 n° 7 pp 1308 -1312 (1968)

[8] B. Mr. WILL GO A frontal solution program for finite elements analysis - Int Num Newspaper. Meth. Eng., 2, 1970

[9] O' LEARY & STEWART, Computing the eigenvalues and eigenvectors of symmetric arrowhead matrices, J. of comp physics 90,497-505 (1990)

[10] J.M. ORTEGA “Introduction to parallel and vector solution of linear systems Plenum Close (1988)

[11] G. RADICATI di BROZOLO, Mr. VITALETTI Sparse matrix-vector product and storage representations one the IBM 3090 with vector facility - IBM-ECSEC G513 Carryforward - 4098 (Rome) July 1986

[12] J.K. REID A notes one the stability of gaussian elimination. J. Int Applies Maths, 1971.8 pp 374 - 375

[13] J.H. WILKINSON Rounding Errors in Algebraic. Processes Her majesty' S stationery office (1963)

[14] J.H. WILKINSON The algebraic eigenvalue problem Clarendon Close Oxford (1965)

[15] C. ROSE “a method multifrontale for the direct resolution of linear systems” Notes EDF - DER HI-76/93/008 (1993)

[16] D. SELIGMANN “Algorithms of resolution for the problem generalized with the eigenvalues” [R5.01.01] - EDF Note - DER HI-75/7815 (1992)

[17] D. SELIGMANN, R. MICHEL “Algorithms of resolution for the quadratic problem with the eigenvalues” [R5.01.02] - EDF Note - DER HI-75/7816 (1992)

5 Description of the versions of the document

Version Aster	Author (S) Organization (S)	Description of the modifications

2,3	D.SELIGMANN EDF-R&D/AMA	Initial text