# Algorithms of retiming

**Summary:**

In this document one presents the algorithms of retiming implemented in MACR_RECAL. It is initially of an algorithm of Levenberg-Marquardt with terminals and then about an algorithm évolutionnaire.

For the first one initially describes the general method before specifying certain elements of them. Are detailed the calculation of the functional calculus, Jacobienne matrix, the determination of the initial parameter of regularization like its evolution, the management of the terminals and the convergence criteria.

The principles generals of the genetic algorithms, and in particular the algorithm évolutionnaire, as well as the implementation in *Code_Aster*, are presented in the second part of the document.

# Contents

# 1 Introduction

Before approaching the problems of retiming strictly speaking, it is useful to recall some elements on the identification of parameters. Let us suppose that one wishes to identify $n$ parameters starting from a given mechanical test. Within the framework of this identification, one defines the sizes:

- $\mathbf{c}$ , the vector of $n$ parameters to be identified, pertaining to $\mathbf{O}$ , convex closed $\mathbb{R}^n$ .
- $\mathbf{d}$ , the vector of the sizes calculated during a simulation of the test by using the parameters $\mathbf{c}$ , in opposition to $\mathbf{d}^{\exp}$ , the vector of the sizes measured during an experimental test. Both belong to space $\mathbf{L}$ observable sizes. The simulation of the experimental test, parameterized by the vector $\mathbf{c}$ , can be realized by various methods: finished differences, finite elements, elements of border,…. It is what we will call the direct problem.

The goal of the identification is to determine the set of parameters $\mathbf{c}$ reducing the difference between measured and experimental sizes (by strongly hoping that the reduction of this variation is sufficient to obtain the desired set of parameters…). One thus introduces a cost functional calculus noted $\mathbf{J}$ depending on $\mathbf{c}$ and measuring the distance enters $\mathbf{d}$ and $\mathbf{d}^{\exp}$ .

$$\mathbf{J}(\mathbf{c})=\|\mathbf{d}-\mathbf{d}^{\mathbf{exp}}\| \qquad \textbf{éq 1-1}$$

where $\|.\|$ indicate a standard on $\mathbf{L}$ .

The identification is thus expressed in the form of the problem of minimization according to:

$$\text{To determine } \mathbf{c}^{*}\in\mathbf{O} \text{ such as } \mathbf{J}(\mathbf{c}^{*})=\underset{\mathbf{c}\in\mathbf{O}}{\text{Min}}\ \mathbf{J}(\mathbf{c})$$

Lastly, one defines retiming as the minimization of a kind of functional calculuses individual known as “least squares” which are expressed in the form:

$$\mathbf{J}(\mathbf{c})=\sum_{n=1}^{N} j_n^2(\mathbf{c}) \qquad \textbf{éq 1-2}$$

or $j_n(\mathbf{c})$ represent the component $n$ variation enters the vector of the calculated and experimental sizes.
It is commonly allowed that among the algorithm of the minimization determinists, most effective for this kind of functional calculuses is the algorithm of Levenberg-Marquardt. It is the latter which was historically the first established in the order MACR_RECAL of *Code_Aster* and that we present in the continuation.

# 2 Algorithm of Levenberg-Marquardt

## 2.1 Position of the problem

There exist several families of algorithms of minimization [bib1]. For the relatively regular problems, the most used are the methods of descent. Their principle is to generate in an iterative way a continuation $\left(\mathbf{c}^{\mathbf{k}}\right)_{k\in N}$ defined by:

$$\left(\mathbf{c}^{\mathbf{k+1}}\right)=\mathbf{c}^{\mathbf{k}}+\alpha^{k}\ \mathbf{g}^{\mathbf{k}} \qquad \textbf{éq 2.1-1}$$

such as, for $f\left(x\right)=\mathbf{J}\left(\mathbf{c}^{\mathbf{k}}+x\ \mathbf{g}^{\mathbf{k}}\right),\ x\in R_{+}^{*}$

- $f\left(x\right)$ is decreasing in the vicinity of $0^{+}$
- $f\left(\alpha^{k}\right)=\underset{x>0}{\mathrm{Min}}\, f\left(x\right)$

$\mathbf{g}^{\mathbf{k}}$ is the direction of descent to the step $k$ . It is the method of determination of $\mathbf{g}^{\mathbf{k}}$ who thus conditions nature the effectiveness of the algorithm used, knowing that these techniques are mainly based on approximations of $\mathbf{J}$ with order 1 or order 2. For the algorithm of Levenberg-Marquardt, one handles an approximation with order 2 of the functional calculus.

## 2.2 Resolution

Within the framework of retiming, one handles square least cost functional calculuses of the type:

$$\mathbf{J}\left(\mathbf{c}\right)=\sum_{n=1}^{N} j_{n}^{2}\left(\mathbf{c}\right) \qquad \textbf{éq 2.2-1}$$

where for example $j_{n}\left(\mathbf{c}\right)=\left(F_{n}^{calc}\left(\mathbf{c}\right)-F_{n}^{\exp}\right)$ , with obvious notations.

The characteristic of these cost functional calculuses lies in the fact that one knows the form of their derivative first and seconds:

$$\left(\nabla_{c}\mathbf{J}\left(\mathbf{c}\right)\right)_{i}=2\sum_{n=1}^{N} j_{n}\left(\mathbf{c}\right)\frac{\partial j_{n}}{\partial c_{i}} \qquad \textbf{éq 2.2-2}$$

$$\left(\mathbf{H}\left(\mathbf{c}\right)\right)_{ij}=2\sum_{n=1}^{N}\left(\frac{\partial j_{n}}{\partial c_{i}}.\frac{\partial j_{n}}{\partial c_{j}}+j_{n}\left(\mathbf{c}\right)\frac{\partial^{2} j_{n}}{\partial c_{i}\partial c_{j}}\right) \qquad \textbf{éq 2.2-3}$$

Then, by supposing that the second term of the preceding equation is negligible in front of the first (what is true when them $j_k$ are linear in $\mathbf{c}$ : this term is null), one can rewrite:

$$\left(\mathbf{H}(\mathbf{c})\right)_{ij} \approx 2\sum_{n=1}^{N} \frac{\partial j_n}{\partial c_i} . \frac{\partial j_n}{\partial c_j} \qquad \textbf{éq 2.2-4}$$

It is interesting on this level to introduce the matrix of sensitivity or Jacobienne matrix defined by:

$$\mathbf{A}=\begin{bmatrix} \frac{\partial j_1}{\partial c_1} & \frac{\partial j_1}{\partial c_2} & \cdots & \frac{\partial j_1}{\partial c_n} \\ \frac{\partial j_2}{\partial c_1} & \frac{\partial j_2}{\partial c_2} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial j_N}{\partial c_1} & \frac{\partial j_N}{\partial c_2} & \cdots & \frac{\partial j_N}{\partial c_n} \end{bmatrix} \qquad \textbf{éq 2.2-5}$$

One can thus express the gradient and Hessien by:

$$\nabla_c \mathbf{J}\left(\mathbf{c}^{\mathbf{k}}\right) = 2\,\mathbf{A}^{\mathbf{T}}\,\mathbf{j} \qquad \textbf{éq 2.2-6}$$

$$\mathbf{H}\left(\mathbf{c}^{\mathbf{k}}\right) \approx 2\,\mathbf{A}^{\mathbf{T}}\mathbf{A} \qquad \textbf{éq 2.2-7}$$

with $\mathbf{j}=\left[j_1,\ \ldots,\ j_N\right]^T$ .

Then let us write the development limited to order 2 of $\mathbf{J}$ :

$$\mathbf{J}(\mathbf{c}) \approx \mathbf{J}\left(\mathbf{c}^{\mathbf{k}}\right) + \left(\mathbf{c}-\mathbf{c}^{\mathbf{k}}\right)^T . \nabla_c \mathbf{J}\left(\mathbf{c}^{\mathbf{k}}\right) + \frac{1}{2}\left(\mathbf{c}-\mathbf{c}^{\mathbf{k}}\right)^T \mathbf{H}\left(\mathbf{c}^{\mathbf{k}}\right).\left(\mathbf{c}-\mathbf{c}^{\mathbf{k}}\right) \qquad \textbf{éq 2.2-8}$$

That is to say $\mathbf{g}^{\mathbf{k}}=\mathbf{c}-\mathbf{c}^{\mathbf{k}}$ , the step of descent at the point $\mathbf{c}^{\mathbf{k}}$ , it must check the condition of stationnarity of the quadratic approximation:

$$\nabla_c \mathbf{J}\left(\mathbf{c}^{\mathbf{k}}\right) + \mathbf{H}\left(\mathbf{c}^{\mathbf{k}}\right)\ \mathbf{g}^{\mathbf{k}} = 0 \qquad \textbf{éq 2.2-9}$$

According to the expression of the gradient and Hessien of $\mathbf{J}$ , one can write:

$$\left(\mathbf{A}^{\mathbf{T}}\mathbf{A}\right)\ \mathbf{g}^{\mathbf{k}} = -\mathbf{A}^{\mathbf{T}}\ \mathbf{j} \qquad \textbf{éq 2.2-10}$$

The solution of this equation leads to an algorithm known under the name of Gauss-Newton, very effective but which presents nevertheless some disadvantages:

- $\left(\mathbf{A}^{\mathbf{T}}\mathbf{A}\right)$ can be almost singular and cause the non-existence of solution.
- There is no control on $\mathbf{g}^{\mathbf{k}}$ , which can be too large and thus leave the parameters acceptable space.

To mitigate these disadvantages, one prefers to use the algorithm of Levenberg-Marquardt which proposes a regularization of the algorithm of Gauss-Newton:

$$\left(\mathbf{A}^{\mathbf{T}}\mathbf{A}+\lambda\ \mathbf{I}\right)\ \mathbf{g}^{\mathbf{k}}=-\mathbf{A}^{\mathbf{T}}\ \mathbf{j} \qquad \textbf{éq 2.2-11}$$

where $\lambda$ is a scalar and $\mathbf{I}$ the matrix identity.

It is noticed that if $\lambda=0$ , one finds the direction given by Gauss-Newton and if $\lambda\to+\infty$ , one finds the direction given by the opposite one of the gradient of $\mathbf{J}$ *i.e.* the greatest slope.

The algorithm of Levenberg-Marquardt thus consists, on the basis of a value of $\lambda$ "raised enough", to decrease it by a factor 10 for example, with each decrease of $\mathbf{J}$ . One passes thus gradually from an algorithm of greater slope to the algorithm of Gauss-Newton. One can thus present this procedure in the form:

- Choice of a starting point $\mathbf{c}^{\mathbf{0}}$ and of an initial value of $\lambda$
- With the iteration $k$, to solve
  $\left(\mathbf{A}^{\mathbf{T}}\mathbf{A}+\lambda\ \mathbf{I}\right)\ \mathbf{g}^{\mathbf{k}}=-\mathbf{A}^{\mathbf{T}}\mathbf{j}$
  $\mathbf{c}^{\mathbf{k+1}}=\mathbf{c}^{\mathbf{k}}+\mathbf{g}^{\mathbf{k}}$
- If $\mathbf{J}\left(\mathbf{c}^{\mathbf{k+1}}\right)<\mathbf{J}\left(\mathbf{c}^{\mathbf{k}}\right)$ , then $\lambda=\lambda/10$ if not $\lambda=\lambda*10$
- Test of convergence

**Note:**

*We considered above the algorithm of Levenberg-Marquardt under the angle of the regularization of the algorithm of Gauss-Newton. It is possible to give a lighting different to this algorithm by regarding it as an algorithm from area of confidence [feeding-bottle 2 ]. Indeed, one can show easily that the system [éq 2.2-11] is the condition of stationnarity of the problem of minimization:*

*To determine* $\mathbf{g}^{\mathbf{k}}$ *such as* $\mathbf{g}^{\mathbf{k}}=\mathrm{ArgMin}\left(\mathbf{g}^{\mathbf{k}\ \mathbf{T}}.\mathbf{A}^{\mathbf{T}}\mathbf{j}+\frac{1}{2}\mathbf{g}^{\mathbf{k}\ \mathbf{T}}\mathbf{A}^{\mathbf{T}}\mathbf{A}\ \mathbf{g}^{\mathbf{k}}\right)$ *subjected to*

$$\|\mathbf{g}^{\mathbf{k}}\|\leq D^{k}\ .$$

*Where* $D^{k}=-\left(\mathbf{A}^{\mathbf{T}}\ \mathbf{A}+\lambda\ \mathbf{I}\right)^{-1}\mathbf{A}^{\mathbf{T}}\mathbf{j}$ **et** $\lambda\geq 0$ .

It is a very simple establishment of the algorithm of Levenberg-Marquardt within which various questions are not tackled:

- How to define the functional calculus $\mathbf{J}$ when does one have several tests?
- How to choose the initial value of $\lambda$ ?
- How to make evolve $\lambda$ in a finer way?
- How to define the field of evolutions of each parameters?

We will clarify these various points in the continuation.

## 2.3 Implementation practical

### 2.3.1 Definition of the functional calculus

At the time of a retiming, the user often has several different measurements; they are discrete physical sizes, possibly of different nature, measured during one or several tests. They are functions of a noted given parameter $t$ (time, X-coordinate,…) that one can thus represent by:

$$F_1^{\exp}=\begin{bmatrix} t_1 & f_1^{\exp}(t_1) \\ \vdots & \vdots \\ t_N & f_1^{\exp}(t_N) \end{bmatrix} \quad F_2^{\exp}=\begin{bmatrix} t_1 & f_2^{\exp}(t_1) \\ \vdots & \vdots \\ t_M & f_2^{\exp}(t_M) \end{bmatrix} \quad F_L^{\exp}=\begin{bmatrix} t_1 & f_L^{\exp}(t_1) \\ \vdots & \vdots \\ t_P & f_L^{\exp}(t_P) \end{bmatrix} \quad \textbf{éq 3.1-1}$$

Each one of these experimental measurements has sound during

$$F_1^{\text{calc}}(\mathbf{c^k})=\begin{bmatrix} \tilde{t}_1 & f_1^{\text{calc}}(\mathbf{c^k},\tilde{t}_1) \\ \vdots & \vdots \\ \tilde{t}_I & f_1^{\text{calc}}(\mathbf{c^k},\tilde{t}_I) \end{bmatrix} F_2^{\text{calc}}(c^k)=\begin{bmatrix} \tilde{t}_1 & f_2^{\text{calc}}(\mathbf{c^k},\tilde{t}_1) \\ \vdots & \vdots \\ \tilde{t}_J & f_2^{\text{calc}}(\mathbf{c^k},\tilde{t}_J) \end{bmatrix} F_L^{\text{calc}}(\mathbf{c^k})=\begin{bmatrix} \tilde{t}_1 & f_L^{\text{calc}}(\mathbf{c^k},\tilde{t}_1) \\ \vdots & \vdots \\ \tilde{t}_K & f_L^{\text{calc}}(\mathbf{c^k},\tilde{t}_K) \end{bmatrix}$$

**éq 3.1-2**

calculated for a game of parameter $c^k$ given. Let us notice that the calculated sizes are not inevitably in same number as the measured sizes nor evaluated for the same value of the parameter $t$ . One can then define the functional calculus least squares to be minimized by:

$$\mathbf{J}(\mathbf{c^k})=\frac{\sum_{i=1}^{N}\left(\frac{f_1^{\exp}(t_i)-f_1^{\text{calc}}(\mathbf{c^k},t_i)}{f_1^{\exp}(t_i)}\right)^2+\sum_{i=1}^{M}\left(\frac{f_2^{\exp}(t_i)-f_2^{\text{calc}}(\mathbf{c^k},t_i)}{f_2^{\exp}(t_i)}\right)^2+\ldots+\sum_{i=1}^{P}\left(\frac{f_L^{\exp}(t_i)-f_L^{\text{calc}}(\mathbf{c^k},t_i)}{f_L^{\exp}(t_i)}\right)^2}{\mathbf{J}(\mathbf{c^0})}$$

**éq 3.1-3**

It is important to notice that:

- if a calculated measurement $f_j^{\text{calc}}$ is not defined in one moment $t_i$ , then his value linearly is interpolated
- if an experimental measurement $f_j^{\exp}$ is worthless, one does not divide the quantity $f_j^{\exp}(t_i)-f_j^{\text{calc}}(\mathbf{c^k},t_i)$ and it is present such as it is in the expression of the functional calculus
- the functional calculus $\mathbf{J}$ is standardized so as to be worth 1. at the beginning of the iterations of retiming

### 2.3.2 Form of the Jacobienne matrix

For the calculation of the Jacobienne matrix, one defines the vector $\mathbf{j}$ errors by:

$$\mathbf{j}=\begin{bmatrix} \frac{f_1^{\exp}(t_1)-f_1^{\text{calc}}(\mathbf{c}^{\mathbf{k}},t_1)}{f_1^{\exp}(t_1)} \\ \vdots \\ \frac{f_1^{\exp}(t_N)-f_1^{\text{calc}}(\mathbf{c}^{\mathbf{k}},t_N)}{f_1^{\exp}(t_N)} \\ \frac{f_2^{\exp}(t_1)-f_2^{\text{calc}}(\mathbf{c}^{\mathbf{k}},t_1)}{f_2^{\exp}(t_1)} \\ \vdots \\ \vdots \\ \frac{f_P^{\exp}(t_L)-f_P^{\text{calc}}(\mathbf{c}^{\mathbf{k}},t_L)}{f_P^{\exp}(t_L)} \end{bmatrix} \quad \textbf{éq 3.2-1}$$

That is to say:

$$j_i^K=\frac{f_K^{\exp}(t_i)-f_K^{\text{calc}}(\mathbf{c}^{\mathbf{k}},t_i)}{f_K^{\exp}(t_i)} \quad \textbf{éq 3.2-2}$$

One finds the form of the Jacobienne matrix then of [éq 2.2-5]:

$$\mathbf{A}=\begin{bmatrix} \frac{\partial j_1^1}{\partial c_1} & \frac{\partial j_1^1}{\partial c_2} & \cdots & \frac{\partial j_1^1}{\partial c_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial j_N^1}{\partial c_1} & \frac{\partial j_N^1}{\partial c_2} & \cdots & \frac{\partial j_N^1}{\partial c_n} \\ \frac{\partial j_1^2}{\partial c_1} & \frac{\partial j_1^2}{\partial c_2} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial j_L^P}{\partial c_1} & \frac{\partial j_L^P}{\partial c_2} & \cdots & \frac{\partial j_L^P}{\partial c_n} \end{bmatrix} \quad \text{éq 3.2-3}$$

Where the terms are calculated by direct finished differences:

$$\frac{\partial j_1^1}{\partial c_i}(c_1,\ldots,c_i,\ldots,c_n)\approx\frac{j_1^1(c_1,\ldots,c_i+\alpha c_i,\ldots,c_n)-j_1^1(c_1,\ldots,c_i,\ldots,c_n)}{\alpha c_i} \quad \textbf{éq 3.2-4}$$

### 2.3.3 Regularization of the linear system

We tackle here the problem of the determination and the evolution of the parameter of regularization $\lambda$. One defines with this intention:

- $\lambda_{max} = \mathrm{Max}(\text{Valeurs propres de } \mathbf{A}^T\mathbf{A})$, $\lambda_{min} = \mathrm{Min}(\text{Valeurs propres de } \mathbf{A}^T\mathbf{A})$, $cond = \lambda_{max}/\lambda_{min}$ if $\lambda_{min} \neq 0$
- $\mathbf{Q}(\mathbf{c}) = \mathbf{J}(\mathbf{c}^k) + (\mathbf{c}-\mathbf{c}^k)^T.\mathbf{A}^T\mathbf{j} + \frac{1}{2}(\mathbf{c}-\mathbf{c}^k)^T.(\mathbf{A}^T\mathbf{A} + \lambda\ \mathbf{I}).(\mathbf{c}-\mathbf{c}^k)$

#### 2.3.3.1 Initial value of $\lambda$

Knowing the sizes above, the following algorithm is defined:

- If $\lambda_{min} = 0$, then $\lambda = 1.E-3\,\lambda_{max}$
- If not
  - If $cond < 1.E5$, then $\lambda = 1.E-16\,\lambda_{max}$
  - If not $\lambda = |(1.E5\,\lambda_{min} - \lambda_{max})|/10001$

**Note:**

*In the last case, the value allotted to $\lambda$ causes to bring back the conditioning of $A^T A$ with $1.E5$*

#### 2.3.3.2 Evolution of the value of $\lambda$

To make evolve $\lambda$, the ratio is defined $R^k = \frac{\mathbf{J}(\mathbf{c}^k) - \mathbf{J}(\mathbf{c}^{k+1})}{\mathbf{Q}(\mathbf{c}^k) - \mathbf{Q}(\mathbf{c}^{k+1})}$, which makes it possible to evaluate the validity of the quadratic approximation of $\mathbf{J}$ : the closer it is to 1, plus this approximation is valid. One from of deduced the following algorithm [bib2]:

- If $R^k < 0.25$, then $\lambda = \lambda \times 10$
- If $R^k > 0.75$, then $\lambda = \lambda / 15$

### 2.3.4 Limitations of the field of evolution of the parameters

For various reasons such as guaranteeing the physical validity of the parameters (strictly positive Young modulus, Poisson's ratio understood enters $0$ and $0.5$ ,...), it is necessary to limit their field of evolution. One thus imposes that $\mathbf{c}$ remain in an acceptable field $O$, convex closed $R^n$. This thus imposes constraints on the parameters:

$$\mathbf{c_{sup}} > \mathbf{c}^k + \mathbf{g}^k > \mathbf{c_{inf}}$$

After dualisation of these conditions by introduction of the multipliers of Lagrange $\mu_{inf}$ and $\mu_{sup}$, the system is solved:

*To find* $g^k \quad \mu_{inf} \quad \mu_{sup}$ *such as*

$$\left(A^T A+\lambda\ I\right)\ g^k+\mu_{inf}+\mu_{sup}=-A^T j$$

$$\begin{cases} c^k+g^k>c_{inf} \\ \mu_{inf}>0 \\ (c^k+g^k-c_{inf})_i\ (\mu_{inf})_i=0 \qquad \forall i=[1,n] \end{cases}$$

$$\begin{cases} c^k+g^k<c_{sup} \\ \mu_{sup}<0 \\ (c^k+g^k-c_{sup})_i\ (\mu_{sup})_i=0 \qquad \forall i=[1,n] \end{cases}$$

This resolution is carried out using an algorithm of active constraints. For any precision on this algorithm, to refer to [bib3] or [bib4].

### 2.3.5 Adimensionnement

One is often brought to identify parameters of various physical nature. The orders of magnitude of these parameters can be extremely different. This can generate very strong differences in the orders of magnitude of the components of the gradient and Hessien of the cost functional calculus and compromise the resolution.
To mitigate this difficulty, it is imperative of adimensionner the unknown factors before beginning the resolution. Here a simple and effective procedure.

That is to say $c^0={}^T\left[c_1^0, c_2^0, \ \ldots, \ c_n^0\right]$, the initial vector of the sizes to be rebuilt. One defines the matrix of adimensionnement $D$ :

$$D=\begin{bmatrix} c_1^0 & & & & \\ & c_2^0 & & & \\ & & \ddots & & \\ & & & c_{n-1}^0 & \\ & & & & c_n^0 \end{bmatrix} \qquad \text{éq 3.5-1}$$

Then, if them $c_i^0$ are all nonworthless, one can define the adimensional unknown factors by:

$$\tilde{c}^0=D^{-1}.\ c^0 \qquad \text{éq 3.5-2}$$

In the same way, an adimensional cost functional calculus is introduced:

$$\tilde{J}(\tilde{c})=J(D.\tilde{c})=J(c) \qquad \text{éq 3.5-3}$$

Like its gradient:

$$\nabla_{\tilde{\mathbf{c}}}\tilde{\mathbf{J}}(\tilde{\mathbf{c}})=\frac{\partial\tilde{\mathbf{J}}(\tilde{\mathbf{c}})}{\partial\tilde{\mathbf{c}}}=\frac{\partial\mathbf{J}(\mathbf{D}.\tilde{\mathbf{c}})}{\partial\tilde{\mathbf{c}}}=\frac{\partial\mathbf{J}(\mathbf{c})}{\partial\mathbf{c}}.\frac{\partial\mathbf{c}}{\partial\tilde{\mathbf{c}}}=\mathbf{D}.\nabla_{c}\mathbf{J}(\mathbf{c}) \qquad \textbf{éq 3.5-4}$$

And its Jacobienne matrix:

$$\tilde{A}_{ij}=A_{ij}\times c_j^0 \qquad \textbf{éq 3.5-5}$$

From an algorithmic point of view, the calculation of the Jacobienne matrix is done classically with the functional calculus $\mathbf{J}$, then it is adimensionnée as well as the current parameters $\mathbf{c}$, before being transmitted to the algorithm of minimization. At the exit of this last, parameters $\tilde{\mathbf{c}}$ are redimensionnés to allow the calculation of the functional calculus $\mathbf{J}$.

### 2.3.6 Convergence criteria

Convergence criteria used in MACR_RECAL consist in testing the decrease of the gradient of the functional calculus. It is pointed out that the use of this criterion is naturally justified by the fact that the objective of the algorithm of retiming is to cancel this gradient.

$$\frac{||\nabla_{\mathbf{c}}\mathbf{J}(\mathbf{c}^{\mathbf{k}})||}{||\nabla_{\mathbf{c}}\mathbf{J}(\mathbf{c}^{\mathbf{0}})||}<Prec. \qquad \textbf{éq 3.6-1}$$

Where $Prec$ is by default taken equal to 1.E-3.

## 2.4 Total algorithm

So as to clarify the sequence of the various operations described above, one presents the algorithm of retiming formally:

- Initializations
  - $k=0$
  - calculation of $\mathbf{A}$ , adimensionnement of $\mathbf{A}$
  - Calculation of $\lambda$ initial
  - **Total iterations**
    - Adimensionnement of $\mathbf{c}^{\mathbf{k}}$
    - Resolution of the equation of Levenberg-Marquardt
    - Imposition of the respect of the terminals
    - Redimensioning of $\mathbf{c}^{\mathbf{k+1}}$
    - Calculation of $\mathbf{J}\left(\mathbf{c}^{\mathbf{k+1}}\right)$
    - Actualization of $\lambda$
    - Calculation of $\mathbf{A}$ , adimensionnalisation of $\mathbf{A}$
    - Test of convergence
    - $k=k+1$
- End

# 3 Algorithm évolutionnaire

## 3.1 Principles general

The algorithms évolutionnaires are part of stochastic methods of total optimization based on the principles of the Darwinian evolution of the biological populations, commonly named methods genetic algorithms. One starts besides by recalling principal phases of a simple genetic algorithm:

- *coding* : each parameter to be readjusted is coded in binary base; the population is created;

- *evaluation* : each individual of the population sees himself allotting a measurement of his adaptation, calculated starting from the function cost to minimize;

- *selection* : the individuals who will produce the next generation of the population are selected, one naturally retains the best within the meaning of the adaptation;

- *crossing* : new individuals are created by the parents designated with the preceding phase. In practice it is a question of exchanging part of the bit string between two parents;

- *change* : random mechanism which disturbs one or more bits of the bit string of the children in order to maintain a certain level of diversity in the population;

- *replacement* : a new population, same size, is made up by replacing the parents with the children

Even if, by abuse language, one often uses the genetic term instead of évolutionnaire, it is necessary to point out the few differences which particularize the algorithms évolutionnaires [5]:

- the individuals of the population are represented by vectors of real numbers and either in the binary coding;

- the selection process is different: whereas the genetic algorithms select $n$ parents to create $n$ children who completely replace them in the population, programs évolutionnaires generates $m$

children from $n$ parents then select the new population by keeping them $n$ better among the unit $n+m$ ;

- the probabilities of crossing and change can vary during generations.

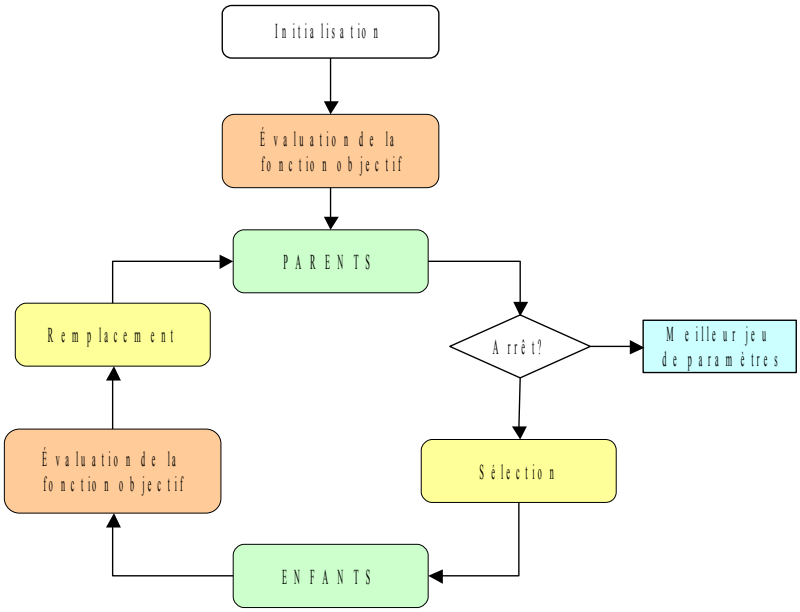
Interest of the introduction of an algorithm évolutionnaire into *Code_Aster* holds in its capacity of multidirectional exploration of the topological space of the parameters by also avoiding that possible the local minima of the functional calculus to be minimized.

The implementation of this algorithm in Code_Aster is the fruit of the partnership between department AMA and Politecnico di Milano.

It is also noted that the defect more criticized algorithms évolutionnaires, that to be greedy in time CPU, can be easily eliminated by the many possible alternatives of parallelization.

## 3.2 Operation of the algorithm

The algorithm is launched while choosing 'GENETIC' METHODE= or 'HYBRID' in the options of the order MACR_RECAL. In the second case, the algorithm évolutionnaire will carry out a followed coarse search for an optimization by the algorithm of Levenberg-Marquardt.




**Logic figure 3.2-a Diagram of the operation of the algorithm évolutionnaire**

The total operation of the algorithm is illustrated on the logic diagram of the Figure 3.2-a and one describes in what follows the stages:

a) **Initialization**: a population of sets of parameters is generated and all the individuals are initialized with the initial values of the parameters to readjust. The size of this population is given by the value of the parameter NB_PARENTS imposed by the user. This value depends on several factors like uncertainty on the solution: more this uncertainty is large plus the population must be large. One thus advises to start a study of retiming while taking for this value 10 and to refine it according to results.

b) **Evaluation of the functional calculus**: it is the functional calculus presented in equation 3.1-3. Initially, after the stage of initialization, only one evaluation is made because all the individuals of the population are identical. Then, in the loop of retiming, one does as many evaluations of the functional calculus at the time an iteration than the number of children (NB_FILS) imposed by the user. The

larger this parameter which manages the renewal rate of the population is, the more the algorithm is greedy in time CPU in this stage. By default the number of children is fixed at 5 (half of the size of the population also imposed by default).

c) **Criterion of stop**: once population of $n$ parents made up and each individuals having his value of the functional calculus, one checks:
- the best value of the functional calculus;
- the iteration count already realized by the algorithm
If the best value of the functional calculus lower than the relative residue is imposed by the user (1.E-3 by default) or if the maximum number of iterations is reached, best the individuals of the population is provided as solution.

d) **Selection**: the best individual of the population is selected and it is **only** (in this implementation) which has the right to reproduce and to generate them $m$ children. This generation quasi-random, around the best individual, with a standard deviation is imposed by the user (keyword ECART_TYPE). At the time of this stage, the algorithm manages the terminals imposed on the parameters by the user. The individuals thus generated are accepted as children only if they are inside the terminals.

E) **Replacement**: after having generated them *m* children, the global population at this stage of the iteration is of $n+m$ (NB_PARENTS + NB_FILS). The operator of replacement carries out here a hierarchy of the individuals according to the values associated with the functional calculus and replaces the individuals of the population PARENTS with $n$ better among $n+m$.

One of the characteristics of this implementation of the algorithm évolutionnaire in *Code_Aster* is the absence, for the moment, of the mechanisms of change and crossing.

## 3.3 Hybrid technique of retiming

While choosing METHODE=' HYBRIDE' in MACR_RECAL, the user has the possibility of combining the advantages of the stochastic algorithms (that évolutionnaire here) with those of the deterministic algorithms (Levenberg-Marquardt in occurrence in our case). It is thus a question of carrying out the first "coarse" research in the topological space of the parameters what will make it possible the algorithm of Levenberg-Marquardt to start optimization with a starting point closer to the solution **total** functional calculus.
Proportioning between stochastic and deterministic is to be fixed by the user taking into account his expert testimony. A great uncertainty on the optimal values of the parameters to be readjusted must impose:
- more iterations of the algorithm évolutionnaire;
- a larger standard deviation for the lotteries;
- a size of the more important population.

These are the three levers which the user has at his disposal to carry out a retiming of quality with a level of performance (time CPU and memory used) satisfactory.

# 4 Bibliography

1. J.C. CULIOLI : "Introduction to optimization", Ellipses, 1994

2. MR. S. BAZARAA, H.D. SHERALI & C.M. SHETTY: "Nonlinear Programming, Theory and Algorithms", Wiley & Sounds, 1979

3. "Discrete formulation of contact-friction", Document *Code_Aster* [R5.03.50]

4. K. KUNISCH & F. RENDL; "Infeasible year activates simple set method for quadratic programming with bounds", SIAM Journal one Optimization, Volume 14, Number 1.2003

5. Z. Michalewicz. "Genetic Algorithms+Data Structures = Programs Evolution". Springer Verlag, 1992--1996.

# 5 Description of the versions of the document

| Version Aster | Author (S) Organization (S) | Description of the modifications |
|---|---|---|
| 7.1 | N.TARDIEU-R&D/AMA | Initial text |
| 10.1 | I.NISTOR - R & D /AMA | |