Communication

Bale Baidi Blaise*, Gambo Betchewe, and Tibi Beda

Optimization of the model of Ogden energy by the genetic algorithm method

https://doi.org/10.1515/arh-2019-0003
Received Dec 01, 2018; accepted Mar 04, 2019

Abstract: The model of Ogden, is a density of energy used in the modeling of hyperelastic materials behavior. This model of energy presents a high number of material parameters to identify. In this paper, we expose a method of identification of these parameters: Genetic Algorithm. This method contrary to the method of Beda-Chevalier, Least Squares, directed programming object method, PSA (Pattern Search Algorithm) and LMA (Levenberg-Marquardt), allows to identify quickly good parameters which give to the Ogden model a very good prediction in uniaxial tension, biaxial tension and pure shear. This prediction is considered to be better because we better bring the experimental curve closer to Treloar one with the parameters optimized by the genetic algorithm.

Keywords: hyperelastic, model, parameters, Genetic Algorithm.

PACS: 02.60.pn, 66.30.hk, 12.39.pn

1 Introduction

In hyperelastic materials characterization, many continuum mechanical approaches have been developed to model the strain energy function in strain invariant-based [1, 2], principal stretch based [3], or hybrid-based [4] modeling. However, enough identification methods have not been developed in the mechanical literature because of the complexity of the problems. Although models are illustrated with estimated rheological parameters, the numerical values evaluation are mostly treated by numerical investigators instead of mechanical ones. The problem presents too many difficulties such as the unknown parameters which are variables of transcendental nonlinear functions. Sometimes, mechanical researchers use empirical methods based on their own experience rather than a methodical approach. Regularly, they use the nonlinear least squares methods, which are all iterative procedures. This requires the user to be a master of numerical calculation for the resolution of problems [5, 6]. Difficulties that come out using the linear or nonlinear least squares [5], lie in the determination of the number of basic functions optimizing the model. This number is established by the investigator, from her own judgment and experiences.

In the field the characterization of hyperelastic incompressible materials: the least-squares method, the Newton-Raphson method, the Gauss-Newton method, the gradient method, the relaxation method, the Newton method [7], the Beda-Chevalier method [8], the Beda-Chevalier method combined to least-squares method [9], directed programming object method [7] and recently PSA (Pattern Search Algorithm) and LMA (Levenberg-Marquardt) methods [10] are methods used in the optimization the material parameters the models. In this paper we use the Genetic Algorithms in its exact description and especially in interactive mode in Matlab 2013 without combination with another method to optimize the Ogden strain energy for 8 materials parameters and four terms.

Algorithm developed in this research has the following original parts: it optimizes a discrete variables, doesn’t require derivative information, it optimizes variables with extremely complex cost function, it provides a list of optimum material parameters not just a single solution, it may encode the variables so that the optimization is done with the encoded, and it is well suited for parallel computers.

2 The strain energy function

2.1 The Ogden energy

In mechanics rubber, the representation of Ogden model depends on principal stretches. The general form of the energy function is given as a convergent series of the princi-
pal stretches [3].

\[ W(\lambda_1, \lambda_2, \lambda_3) = \sum_{p=1}^{+\infty} 2a_p (\lambda_1^p + \lambda_2^p + \lambda_3^p - 3), \]  

(1)

where \( \lambda_j \) are the principal stretches, \( a_p \) are real constants, characteristic of material. Under a different form, this energy function is equally written as follows

\[ W(\lambda_1, \lambda_2, \lambda_3) = \sum_{p=1}^{N} \frac{\mu_p}{\alpha_p} (\lambda_1^{a_p} + \lambda_2^{a_p} + \lambda_3^{a_p} - 3), \]  

(2)

where \( \alpha_p \) and \( \mu_p \) are real numbers characteristic of material such as

\[ \mu_p \times \alpha_p > 0. \]  

(3)

### 2.2 Nonlinear Behavior

For simple tension [8]

\( (\sigma_1 = \sigma; \sigma_2 = \sigma_3 = 0, \sigma_k, \) being the Cauchy-stress), one has

\[ \lambda_1 = \lambda; \lambda_2 = \lambda_3 = \frac{1}{\sqrt{\lambda}}. \]  

(4)

The relative strain energy expression is given as follows.

\[ W = \sum_{p=1}^{N} \frac{\mu_p}{\alpha_p} (\lambda_1^{a_p} + 2\lambda^{-0.5a_p} - 3). \]  

(5)

The behavior law is given by [3]

\[ \sigma = \lambda^0 W, \]  

(6)

\[ \sigma = \sum_{p=1}^{N} \mu_p (\lambda_1^{a_p} - \lambda^{-0.5a_p}). \]  

(7)

For equal-biaxial tension [8]

\( (\sigma_1 = \sigma_2 = \sigma; \sigma_3 = 0), \) one has

\[ \lambda_1 = \lambda_2 = \lambda, \]  

(8)

\[ \lambda_3 = \frac{1}{\lambda^2}. \]  

(9)

Thus, the strain energy function becomes

\[ W = \sum_{p=1}^{N} \frac{\mu_p}{\alpha_p} (2\lambda_1^{a_p} + \lambda^{-2a_p} - 3). \]  

(10)

The behavior law is given by [3]

\[ \sigma = \frac{1}{2} \lambda^0 W, \]  

(11)

\[ \sigma = \sum_{p=1}^{N} \mu_p (\lambda_1^{a_p} - \lambda^{-2a_p}). \]  

(12)

### 3 Genetic Algorithm

#### 3.1 Behavior Law

For simple tension, the nominal stress \( \sigma_s \), which is referred to the unstrained state, Piola-Kirchhoff stress, is given by [4]

\[ \sigma_s = \frac{\sigma}{\lambda}, \]  

(13)

\[ \sigma_s = \sum_{p=1}^{N} \mu_p (\lambda_1^{a_p-1} - \lambda^{-0.5a_p+1}), \]  

(14)

\[ \sigma_{nominal} = \mu_1 (\lambda_1^{a_1-1} - \lambda^{-0.5a_1+1}) \]  

\[ + \mu_2 (\lambda_1^{a_2-1} - \lambda^{-0.5a_2+1}) \]  

\[ + \mu_3 (\lambda_1^{a_3-1} - \lambda^{-0.5a_3+1}) \]  

\[ + \mu_4 (\lambda_1^{a_4-1} - \lambda^{-0.5a_4+1}). \]  

(15)

#### 3.2 Technique of parameters identification

The most important reason for us to use a Genetic Algorithm (GA) for the data fitting problem is that it is a multi modal optimization problem. GAs are stochastic algorithms whose search methods model some natural phenomena such as genetic inheritance and the Darwinian struggle for survival [11, 12]. A population of candidate solutions called individuals, it is bred in a parallel fashion. The population evolves so that it contains more and more fit individuals. An initial population of individuals is generated randomly, and newer individuals are created iteratively until acceptable solutions are found. Each iteration is called generation.

One of the attractive points of GAs is their high probability of obtaining the global optimal or a suboptimal solution for a multimodal problem. In this section, we discuss initial individuals and their population, the fitness function, methods of crossover and mutation, and the automatic determination of parameters. We use an elitism method to select the individuals to make up the next generation. We summarize our genetic algorithm implemented in Matlab 2013 [13] by the flowchart below.

#### 3.3 Individuals and population

We use the binary code strategy [14]. This is a standard GA coding, each gene has a value equal to 0 or 1. The optimization algorithm consists of two parts: The first part
is founded on enumerative method which is the binary searching method. The second part uses the simplicity of genetic algorithms subjecting the population of possible results to process of replication, crossing-over and mutation [11].

In the preliminary stage of search for optimal values of material constants minimizing a nonlinear objective function determining the approximation error, the division of the considered class of values of material constants is made by iterative method, until the values are close to optimal or, which is seldom, until material constants of an ideal model with a zero approximation error are found. On the basis of the graphic presentation of the mean squared errors dependence on received values of material constants, the distribution of constants sets into two groups is achieved. Each of these groups determines proper fitting of the theoretical curve to the experimental one in one of two strain ranges-from small to medium and from medium to large one [12]. The initial populations of models obtained in this way, determined by close to optimal material constants sets are then subjected to optimization processes according to created genetic algorithm.

The genetic processing begins from determining individuals which will be included in mating pool. The first generation subjected to further processing of genetic algorithm [14]. The generation begun with two populations originating from the ones created. The number of individuals enclosed in a mating pool is conditioned upon the program user. All the sets of material constants are then submitted to binary coding. Within the limits of each set one chromosome consisting of three combined with each other binary codes is created. Large sets of genes obtained in this way then undergo processes of replication, crossing-over and mutation. Before the genetic material exchange, the individuals have to be joined in pairs. A single individual is randomly chosen from each population enclosed in a mating pool. This individual becomes one of the parents in newly created pair. This procedure is applied to all the individuals in the mentioned pool.

In order to accomplish all subsequent genetic operations on material constants values in a proper way, all coded forms of material constants are remembered as sequences of symbols. When an initial population is created, values of the parameters are created by chance with a uniform distribution on the specified interval. In our present work, the initial parameters of our algorithm is defined as follows:

\[
\begin{align*}
N &= 512; \text{ (number of bits in a chromosome)} \\
M &= 250; \text{ (population)} \\
M_2 &= M/2; \text{ (choose an even number)} \\
last &= 100; \text{ (number of generations)} \\
mutrate &= 0.8; \text{ (mutation rate)} \\
nmuts &= \text{mutrate} \times N \times (M - 1); \text{ (number of crossover in the population). one considers an initial population made up of 8 individuals in the Ogden model (} \alpha_1, \mu_1, \alpha_2, \mu_2, \alpha_3, \mu_3, \alpha_4, \mu_4 \text{) defines into binary code by } \\
\alpha_1 &= \text{round} (\text{rand} (M, N)); \text{ initial population} \\
\alpha_2 &= \text{round} (\text{rand} (M, N)); \text{ initial population} \\
\alpha_3 &= \text{round} (\text{rand} (M, N)); \text{ initial population} \\
\alpha_4 &= \text{round} (\text{rand} (M, N)); \text{ initial population} \\
\mu_1 &= \text{round} (\text{rand} (M, N)); \text{ initial population} \\
\mu_2 &= \text{round} (\text{rand} (M, N)); \text{ initial population} \\
\mu_3 &= \text{round} (\text{rand} (M, N)); \text{ initial population} \\
\mu_4 &= \text{round} (\text{rand} (M, N)); \text{ initial population} \\
\end{align*}
\]

1 1 0 1 1 ......1 0 0 0 1 0 ......0 1 1 1 1 1 ......1 0 1 0 0 0 ......0 1 0 0 1 1 ......1 1 0 0 0 0 ......1

After having to generate our initial parameters in binary code we pass at the stage of selection.

### 3.4 Selection

One evaluates the performance of each individual using the objective function [14]. The performance is a measurement of the exactitude of the solution obtained with the parameters of an individual. This objective function varies from one problem to other. It also varies if one tries to maximize or to minimize a problem.

In our work we seek to reduce the difference between two curves, the experimental curve of Treloar and that of
the model of Ogden with 8 parameters. In this paper, we use the equation below (16) as our fitness function.

\[
    f_{\text{obj}} = (\sigma_{\text{experimental}} - \sigma_{\text{nominal}})^2. \tag{16}
\]

In simple tension, the nominal stress of Ogden Model is given by equation (15). Let us pose \( \sigma_{\text{experimental}} \) designates the fitting data of Treloar \([6, 18]\). After having to calculate the objective function, we choose the first good parameters of the model of Ogden. \(a_1 = a_1((1 : M),;)\); the best local population

\[
    a_2 = a_2((1 : M),;) \text{; the best local population}
\]

\[
    a_3 = a_3((1 : M),;) \text{; the best local population}
\]

\[
    a_4 = a_4((1 : M),;) \text{; the best local population}
\]

\[
    \mu_1 = \mu_1((1 : M),;) \text{; the best local population}
\]

\[
    \mu_2 = \mu_2((1 : M),;) \text{; the best local population}
\]

\[
    \mu_3 = \mu_3((1 : M),;) \text{; the best local population}
\]

\[
    \mu_4 = \mu_4((1 : M),;) \text{; the best local population}
\]

There is now a whole of individuals who survived the selection and which represent best obtained solutions.

### 3.5 Crossover method

The operator of crossing allows the creation of new individuals according to a strong simple process \([14]\). It allows the exchange of information between the chromosomes. All initially, two individuals, which form a couple then, are drawn within the new population of the reproduction. Then a site of crossing is drawn by chance. Lastly, according to one probability \(P_c\) which the crossing is carried out, the final segments of the two parents are then exchanged around this site. This operator allows the creation of two new individuals.

However, an individual selected at the time of the reproduction does not undergo necessarily the action of a crossing. More the probability \(P_c\) is raised and more the population will undergo change.

At all events, it may be that the joint action of the reproduction and the crossing is insufficient to ensure success of GA. Thus, in the case of the binary coding, some characters can disappear from the population. For example we consider the \(a_1\) parameter, here we have randomly selected a crossover point after the six digits. \(ic = 1 : \text{nmuts}\)

\[
    a_1(\text{ceil}(M * \text{rand}), 1 : \text{cross}(ic)) = a_1(M + ic, 1 : \text{cross}(ic));
\]



\[
    a_1(\text{ma}) = 001011 | 11001000,
\]

\[
    a_1(pa) = 111011 | 00000001,
\]

\[
    a_{\text{news}} = 00101100000001, 11101111000100.
\]

In this representation \(ma\) and \(pa\) are the parents able to couple itself with end to have a new generation. one again obtains a population of \(N\) individuals which, in theory, have a better performance on average than the individuals of the preceding generation.

### 3.6 Mutation method

The role of this operator is to modify by chance, with a certain probability, the value of a component of the individual \([14]\). In the case of the binary coding, each bit pertaining to 0; 1 is replaced according to a probability \(p_m\) by its reverse. The computer code which was useful to us to find the mutated bits is

\[
    \text{ix} = \text{ceil}(M \times \text{rand});
\]

\[
    \text{iy} = \text{ceil}(N \times \text{rand});
\]

\[
    a_1(\text{ix}, \text{iy}) = 1 - a_1(\text{ix}, \text{iy});
\]

\[
    00101100000001 = 00101000000001.
\]

In this example the population is mutated from a 1 to a 0.

The process of selection, reproduction and change is then repeated. Each time that one carries out these three operations, that constitutes a generation. One continues until the maximum number of generations is reached or until it that the performance of the best individual is smaller than a minimal error specified before.

### 3.7 Elitism method

The elitism is the conservation of the best individual in the new generation \([14]\). The better individual is preserved and the other is destroyed. This technique makes it possible to make sure that our objective function decreases with the use of the new parameters. By using the parameter \(a_1\), the elitism has been implemented by the following code:

\[
    a_1(M2 + 1 : \text{end},;) = a_1; \text{ After having decoded our parameters we obtain the results hereafter.}
\]

### 4 Presentation of some methods used in the optimization the parameters of Ogden model

In this part we briefly present the methods used in optimization of the parameters of the Ogden model.
4.1 Beda-Chevalier method [8]

In the difficulty to determine the parameters of the energy of Gent-Thomas, T. Beda and Y. Chevalier consider a function refines not only like one relation of the form \( y = f(x) \), but like a relation between two vectors of components \((y_1, y_2, y_3, \ldots y_n)\) and \((x_1, x_2, x_3, \ldots x_n)\). Thus, they propose the algorithm of identification. Being given N experimental points, one seeks the function of approximation \( f_a \) whose curve is in correlation with the experimental data. The function of approximation is defined by:

\[
f_a(x) = \sum_{i=1}^{N} C_i \phi_i(x)
\]

Where the function \( \phi(x) \) can be polynomial, exponential, trigonometrical, and \( C_i \) are the constants of the model of energy considered.

The disadvantage of the Beda-Chevalier method is that it is a very traditional method based on the determination of the slope of the nonlinear function curve, this slope can sometimes be negative, it is in this direction that Beda combines its method with the method of least squares at end to direct the sign of the solutions obtained [3]. This method remains very tiresome for a high number of parameter.

4.2 Least Squares method [9]

The least squares methods takes into account the particular form of the function \( \phi \). On the basis of the definition of the function \( \phi \), it aims at linearizing the function \( F \). There are \( N \) points of measurement (experimental values in the space of representation \((x_i, y_i)\)), and \( m \) a fixed entirety \( m \leq N \). \( m \) represents the number of parameter of the model of energy. One seeks a function \( F \) of the form: \( F(x) = \sum_{k=1}^{m} a_k \phi_k(x) \) who minimizes the sum of the standard deviations.

The disadvantage of the Least Squares method it is that, it is very traditional method and requires a very good knowledge on the derivative of the function has to use.

4.3 Directed programming object method [7]

In [7] the author combine the great stages of Genetic Algorithm (selection, crossover and mutation) with following stages: heritage, the encapsulation and polymorphism for the identification the materials parameters. Implementation of this method this fact with an old software FORTRAN 90 which requires a great time computing compared to Matlab and much reflexion.

It is significant to note that our present work is of a great difference of that of Gilles marckmann method, because we use the genetic Algorithms in its exact description and especially in interactive mode in Matlab 2013 without combination with another method. The disadvantage of the Directed programming object method it is that, requires a great time computing and much reflexion.

4.4 Pattern Search Algorithm (PSA) and Levenberg-Marquardt (LMA) method [10]

In [10], algorithm PSA (Pattern Search Algorithm) and LMA (Levenberg-Marquardt) method, were used for the optimization of the model of Ogden in the 4 terms and 8 parameters. In this article the author does not describe the principle operation of its algorithms, i.e. how to program them under Matlab (Which are the initial parameters introduce? which are the factors which influences the convergence results?) Moreover, The disadvantage of the PSA and LMA methods it is that, is require a machine have great capacity, with a rather high time of calculation (i7 CPU for 6h), contrary to our Genetic Algorithm (9min on a computer Dell i3 CPU).

5 Results and Analysis

We present in Table 1 the values of the various parameters of the Ogden model obtained by the following methods: Genetic Algorithm, Beda-Chevalier [8], Least Squares(LS) [9], directed programming object method [7], PSA and LMA method [10].

5.1 Comparison the Genetic Algorithm method to Beda-Chevalier method [8] and Least Squares (LS) method [9]

The various values mentioned in the Table 1, enable us to obtain the following curves:

Uniaxial extension

Figure 2, presents the experimental curve of Treloar and the curves of the model of Ogden energy plotted in uniaxial extension, starting from three different methods of optimization to knowing: Genetic Algorithm; Beda-chevalier; Least Squares (LS). From this figure this reveals that the
Table 1: Parameters Values.

| Methods                     | Parameters              | Unit     |
|-----------------------------|-------------------------|----------|
| Genetic Algorithm           | \( \mu_1 = 0, 238 \times 10^{-25} \) | Mpa      |
|                            | \( \mu_2 = 0, 00127 \)   | Mpa      |
|                            | \( \mu_3 = 0, 597 \)     | Mpa      |
|                            | \( \mu_4 = -0, 024 \)    | Mpa      |
|                            | \( a_1 = 26 \)           | /        |
|                            | \( a_2 = 5 \)            | /        |
|                            | \( a_3 = 1, 3 \)         | /        |
|                            | \( a_4 = -2 \)           | /        |
| Beda-Chevalier method       | \( \mu_1 = 0, 3 \times 10^{-25} \) | Mpa      |
|                            | \( \mu_2 = 0, 0013 \)    | Mpa      |
|                            | \( \mu_3 = 0, 6276 \)    | Mpa      |
|                            | \( \mu_4 = -0, 011 \)    | Mpa      |
|                            | \( a_1 = 29, 7 \)        | /        |
|                            | \( a_2 = 5 \)            | /        |
|                            | \( a_3 = 1, 3 \)         | /        |
|                            | \( a_4 = -2 \)           | /        |
| Least Squares (LS) method   | \( \mu_1 = -0, 011 \)    | Mpa      |
|                            | \( \mu_2 = 0, 62995 \)   | Mpa      |
|                            | \( \mu_3 = 0, 00127 \)   | Mpa      |
|                            | \( \mu_4 = / \)           | Mpa      |
|                            | \( a_1 = -2 \)           | /        |
|                            | \( a_2 = 1, 3 \)         | /        |
|                            | \( a_3 = 5 \)            | /        |
|                            | \( a_4 = / \)            | /        |
| directed programming object | \( \mu_1 = 0, 63 \)      | Mpa      |
| method                     | \( \mu_2 = 1, 2 \times 10^{-3} \) | Mpa      |
|                            | \( \mu_3 = -0, 01 \)     | Mpa      |
|                            | \( \mu_4 = / \)           | Mpa      |
|                            | \( a_1 = 1, 3 \)         | /        |
|                            | \( a_2 = 5 \)            | /        |
|                            | \( a_3 = -2 \)           | /        |
| PSA method                  | \( \mu_1 = 0, 64979 \)   | Mpa      |
|                            | \( \mu_2 = 0, 027261 \)  | Mpa      |
|                            | \( \mu_3 = 0, 0032134 \) | Mpa      |
|                            | \( \mu_4 = -0, 0095667 \) | Mpa      |
|                            | \( a_1 = 1, 13221 \)     | /        |
|                            | \( a_2 = 4, 526 \)       | /        |
|                            | \( a_3 = 1, 1313 \)      | /        |
|                            | \( a_4 = -2, 0548 \)     | /        |
| LMA method                  | \( \mu_1 = 0, 806 \)     | Mpa      |
|                            | \( \mu_2 = 0, 0327533 \) | Mpa      |
|                            | \( \mu_3 = 2, 3601 \times 10^{-7} \) | Mpa      |
|                            | \( \mu_4 = -0, 004861 \) | Mpa      |
|                            | \( a_1 = 0, 80939 \)     | /        |
|                            | \( a_2 = 8, 9157 \)      | /        |
|                            | \( a_3 = 3, 1972 \)      | /        |
|                            | \( a_4 = -2, 26312 \)    | /        |

three methods used allow the model energy of Ogden to cover the whole of the incompressible field of deformation of materials hyperelastic in great deformation which extends up to 750%.

Biaxial extension

In biaxial extension Figure 3, the three methods of optimization: Genetic Algorithm; Beda-Chevalier; Least Squares (LS) allow the model Ogden to better study the hyperelatic behavior of rubbery materials on large a zone of deformation until 450%.
Pure Shear

In pure shear, of the three methods of optimizations used, the genetic algorithm method makes it possible to give to the model of Ogden a good correlation with the experimental curve of Treloar.

**Figure 4:** Pure Shear.

5.2 Comparison the Genetic Algorithm method to directed programming object method [7], PSA method and LMA method [10]

Simple tension

In Uniaxial extension, the genetic algorithm gives to the Ogden model a good correlation with the experimental data of Treloar, see Figure 5.

**Figure 5:** Simple tension.

Biaxial tension

In Biaxial tension, the genetic algorithm gives to the Ogden model a good correlation with the experimental data of Treloar, see Figure 6.

**Figure 6:** Biaxial tension.

Pure Shear

**Figure 7:** Pure Shear.
In Pure shear, the genetic algorithm gives to the Ogden model a good correlation with the experimental data of Treloar, see Figure 7.

5.3 Analysis

5.3.1 Discussion

The relative error is defined by

\[
\frac{|\Delta \sigma|}{\sigma} = \frac{|\sigma_{\text{exp}} - \sigma_{\text{th}}|}{\sigma_{\text{exp}}},
\]

where \(\sigma_{\text{exp}}\) is the experimental load and \(\sigma_{\text{th}}\) is the theoretical predicted response of the model.

We present the various relative errors between the experimental curve of Treloar and that of Ogden obtained by using the methods of optimizations: Genetic Algorithm, Beda-Chevalier, Least Squares (LS), directed programming object, PSA and LMA method. This reveals that in uniaxial extension the method of the genetic algorithm gives a maximum relative error about 0.17%, the Beda-Chevalier method presents a maximum relative error of about 0.11% and the method of least-squares presents a maximum relative error of about 0.17%, directed programming object gives a maximum relative error about 0.19%, PSA method gives a maximum relative error about 0.18% and LMA method method gives a maximum relative error about 0.21%.

On the other hand in biaxial extension the method of the genetic algorithm gives a maximum relative error about 0.51%; the Beda-Chevalier method presents a maximum relative error of about 0.57%, the method of least-squares presents a maximum relative error of about 1%, PSA method gives a maximum relative error about 0.21% and LMA method gives a maximum relative error about 0.49%.

In pure shear the method of the genetic algorithm gives a maximum relative error about 0.72%; the Beda-Chevalier method presents a maximum relative error of about 0.7% and the method of least-squares presents a maximum relative error of about 0.82%, directed programming object gives a maximum relative error about 0.89%, PSA method gives a maximum relative error about 0.51% and LMA method method gives a maximum relative error about 0.54%.

The main advantage of the genetic algorithm over other techniques is its capacity to find out and optimize the number of generating basic functions, thus to determine the optimal order of linear or non-linear approximation.

5.3.2 Critical analysis

The use of least squares requires the user to provide the number of generating basic functions, which can be the order of approximation for linear least squares or the choice of an appropriate effective model for non-linear least squares. All usual non-linear methods \textit{i.e.} Newton-Raphson, Gauss-Newton, Marquardt, etc. being iterative procedures \cite{5, 19}, optimal use of non-linear least squares requires good judgment and experience to establish good starting values and to avoid local minima for stationary. The convergence of these procedures depends on many factors: regularity of the Jacobian matrix, choice of the line search, etc. on the other hand with the method of Beda-Chevalier the difficulties arise when by equivalence of function one cannot bring out clearly a distinct, separate basic function at a step of identification, for example the model of Yeoh and Fleming \cite{20}, Takamizawa and Hayashi \cite{19}, Veronda and Westmann \cite{20}, Hart-Smith \cite{21}.

The method of the genetic algorithm as for it requires much time of programming for the execution one calculates and it remains still an approach based on the chance, it is well suited for parallel computers, Provides a list of optimum material parameters, but not just a single solution, Optimizes variables with extremely complex cost function and may encode the variables so that the optimization is done with the encoded parameters.

6 Conclusion

The generalized genetic algorithm treated in this paper is a method with original methodology of approximation of functions. The identification is done by implementation of the codes of program in Matlab. This process permits us to optimize the approximation procedure and therefore reduces the difficulties of resolution. The genetic algorithm approach unlike the well-known methods, takes into account any analytical form of a model and permits only the basic functions that generate the model to emerge. Thus, the choice of basic functions and size of the population remains very important for the best resolution. It also enables us to palliate difficulties inherent in the complexity to identify exponent parameters and weighting coefficients, which is a delicate problem in numerical calculus \textit{i.e.} non-linear identification procedure.
References

[1] Rivlin R. S., Rheology: Theory and Applications, edited by Eirich F. R., Academic Press, NY, Vol. 1, 1956.
[2] Green A. E., Adkins J. E., Large Elastic Deformation and Nonlinear Continuum Mechanics, Clarendon Press, Oxford, 1960.
[3] Ogden R. W., Nonlinear Elastic Deformations, Ellis Horwood, England, 1984.
[4] Beda T., Chevalier Y., Hybrid Continuum Model for Large Elastic Deformation of Rubber, J. Appl. Phys., 2003, 94, 2701-2706.
[5] Madsen K., Nielsen H. B., Tingleff O., Method for Nonlinear Least Squares Problems, IMM, J. No H38. 7.7 HBN/ms, TU Denmark, 1999.
[6] Kirkup L., Principles and Applications of Non-Linear Least Squares: An Introduction for Physical Scientists Using Excel’s Solver, UT Sydney, Science: Applied Physics, Copyright UTS, 2002.
[7] Marckmann G., Contribution à l’étude des élastomères et des membranes soufflées, Thèse de Doctorat PhD, Ecole Centrale de Nantes, 2004.
[8] Beda T., Optimizing the Ogden Strain Energy Expression of Rubber Materials, Journal of Engineering Materials and Technology, 127, 2005, 351-353.
[9] Beda T., Combining Approach in Stages with Least Squares for fits of data in hyperelasticity, Comptes Rendus Mecanique, 2006, 334-628.
[10] Wu Y., Wangand H., Li A., Parameter identification methods for Hyperelastic and Hyper-Viscoelastic Models, Applied Sciences, 2016, 1-13.
[11] David Goldberg E., Genetic algorithms in search, optimization and machine learning. Reading: Addison-Wesley, 1989.
[12] Mitchell M., An introduction to genetic algorithms, Cambridge, Massachusetts: The MIT Press, 1996.
[13] MATLAB, The Language of Technical Computing. Copyright (c) 1984-1999 by The MathWorks, Inc. Version 5.3.0.10183 (R11), 2013.
[14] Haupt R. L., Haupt S. E., Pratical Genetic Algorithms, Wiley interscience, 2004.
[15] Koza J.R., Genetic programming on the programming of computers by means of natural selection, A. Bradford Book, the MIT Press Cambridge, Massachusetts, London, England, 1998.
[16] Trelor L. R. G., Stress-strain data for vulcanized rubber under various types of deformation, Transaction of the Faraday Society, 1944, 40, 59-70.
[17] Steinmann P., Hossein M., Possart G., Hyperelastics models for rubberlike materials: consistent tangent operators and suitability for Trelor’s data, Arch. Appl. Mech., 2012, 00419-0610.
[18] Yeoh O. H., Fleming P. D., A new attempt to reconcile the statistical and phenomenological theories of rubber elasticity, J. Polym. Sci. Part B: Phys., 1997, 35, 1919-1931.
[19] Takamizawa K., Hayashi K., Strain energy density function and uniform strain hypothesis for arterial mechanics, J. Biomech., 1987, 20, 7-17.
[20] Veronda D. R., Westmann R. A., Mechanical characterization of skin finite deformations, J.Biomech 1970, 3, 11-24.
[21] Hart-Smith L. J., Elasticity parameters for finite deformations of rubber-like materials, Z. Angew Math. Phys. (ZAMP)., 1966, 17, 608-626.