1

Parametric and non-parametric identification of macro-mechanical models

Michèle Sebag and Marc Schoenauer and Habibou Maitournam

Ecole Polytechnique, Palaiseau, France

1.1 INTRODUCTION

Since the very beginning of Engineering Sciences, engineers have been interested in numerical simulations of whatever physical system they were considering, in order to obtain accurate predictions of the behavior of the system they were building under various circumstances: such predictions allow one in turn to optimize the target realization (mechanical structure, chemical plant, ...).

The main causes of errors in simulated models are recognized to be the too large simplification of the underlying physical phenomenon (e.g. using linear elastic models in Structural Mechanics), or the numerical errors during the numerical computations (e.g. due to the non-linearities when using some plastic model). However, a third cause of error should not be neglected: even under given physical hypotheses, some internal laws must be given before actually computing a simulated behavior. Those can reduce to a few coefficients (e.g. Young and Poisson modules in the elastic model), or take more complex form (e.g. the whole plasticity convex for elasto-plastic models of structures).

This paper is concerned with the identification of the constitutive law of materials in the framework of one-dimensional elasto-visco-plastic rheological models. The behavior of such a material can be approached by considering it is well approximated by the reaction of an assembly of elementary modules representing the basic possible

Sample Contributed Book
Editor Jenny Smith

behaviors (i.e. elasticity, viscosity and plasticity). The way these elementary modules are connected gives the shape of the underlying system of differential equations whose resolution gives access to the simulated behavior of the material. Hence different assemblies give birth to a wide range of global behaviors.

The identification of rheological models proceeds from experimental responses of the target material to given loading paths in the time-strain-stress space. It can be made either at the level of the parameters of a given assembly (i.e. elasticity modules, viscosity coefficients and plasticity thresholds), or at the level of the topology assembly itself, when no knowledge about the structure of the model is available. Even in the simplest former case of parametric identification, deterministic methods meets their usual limitations for non-convex functions, and Evolutionary Computation seems a good choice. But in the case where the topology of the assembly is unknown, EC is, as far as we know, the only method that ever proved able to yield a solution [SSJ+96].

The paper is organized as follows: General model identification problems are presented in section 1.2, and basic differences between parametric and non-parametric problems are sketched. Section 1.3 introduces in more details the one-dimensional rheological models for elasto-visco-plastic materials, and details the numerical simulation of such models. Section 1.4 then presents the results of the parametric identification for the polyethylene, in the case where the topology of the assembly is guessed. To handle the general case, the assemblies of elementary mechanical modules are transformed into tree structures: this allows one to use Genetic Programming (GP) techniques to solve the non-parametric identification problem: Section 1.5 first recalls the basic concepts of GP, together with some specific modifications that proved necessary in the case of rheological models identification, then presents and discusses some results of the non-parametric identification. Section 1.6 discusses the issues related to parametric vs non-parametric optimization, and concludes the paper.

1.2 MODEL IDENTIFICATION

1.2.1 Inverse problems

The numerical simulation of a physical, chemical, mechanical phenomenon starts with a modeling phase, during which some mathematical formulation of the studied process is derived, usually at the cost of some simplifying hypotheses. This phase generally relies on an internal *model*, or *constitutive law*, a microscopic description of some underlying phenomenon. By changing this constitutive law, one models as many different processes that satisfy the same simplifying hypotheses.

The overall simulation process can be described graphically by Figure 1.2.1: the general purpose numerical algorithm internally uses the model at hand to derive simulated results from experimental conditions. The simulation of a direct process is said *accurate* if the error between the numerical results and the experimental results for the same experimental conditions is small. It is *robust* if a small change in the experimental conditions results in small modifications of the simulated results.

Though in most cases such models reduce to real-valued functions (see e.g. [FS95]), they can also take different forms, as will be seen in section 1.3.

Suppose now that a new system is to be studied, for which no precise model is

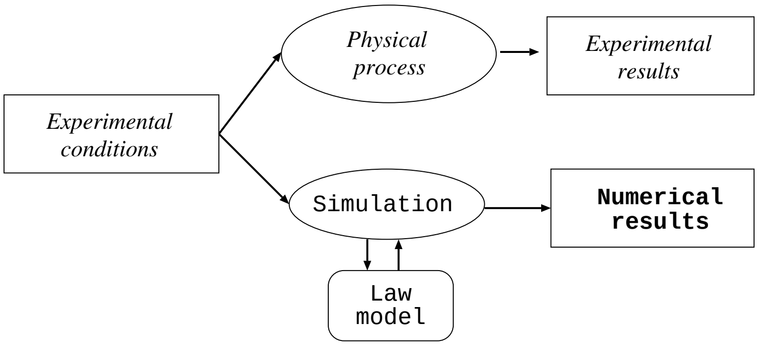


Figure 1.1 *Direct problem: Physical and simulated processes for the same experimental conditions.*

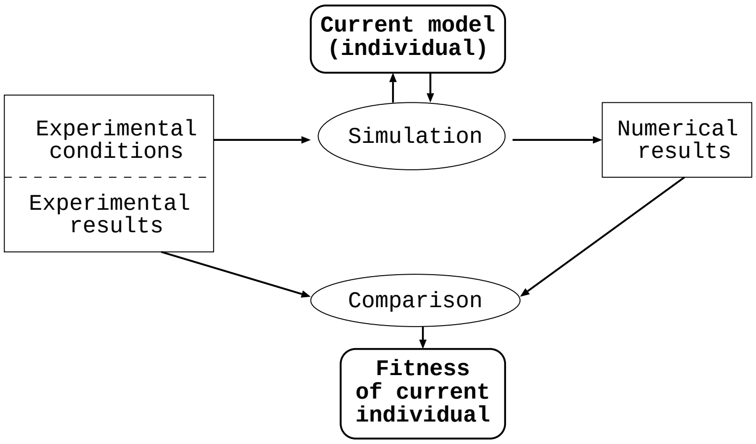


Figure 1.2 *Evolutionary approach for the inverse problem: The fitness of the individual (tentative model) at hand is computed by comparing the actual experimental results with the numerical results obtained using that individual as the model in the simulation.*

available, but which is known to be relevant to the same simplifying hypotheses than another well-known case, and for which some experiments can be made with different experimental conditions. Obtaining the model from those experimental data (experimental conditions plus corresponding experimental responses) constitutes the *inverse problem.*

The quality of a tentative model can be easily derived by comparing, under given experimental conditions, the actual experimental responses with their numerical simulations obtained using that tentative model: a good model should give simulated results close to the experimental ones. Hence, Figure 1.2 represents a possible fitness for evolutionary inverse problem solving.

The critical issue is the choice of the search space in which to look for a solution. It should be large enough to include high quality solutions, but not too large, as the search would then be intractable.

Some constraints can result from the simplifying hypotheses made during the modeling phase: For instance, if some law was assumed linear, only a few coefficients completely describe it.

Some guidelines can be also gathered from the experts in the application domain, resulting in constraints on the search space: For instance, the parameter representing the density of any fluid must be strictly positive.

But another important question also influences the choice of the search space: What do the experts of the application domain expect from the identification? If the emphasis is put on understandability, then only analytical models should be used (for model reducing to real-valued functions), while the choice is larger if only predictive accuracy matters (including, for real-valued models, Neural Networks, for instance).

Another important issue in inverse problem solving is the *generalization capability* of the solution: How good is the resulting model when used with experimental conditions that are different from those used during the identification process? The answer to that question can in turn give some advantage to complex but robust representations vs simpler but unstable ones: Neural Networks are known to better generalize than polynomials, for instance, in the case of data fitting.

The computation of the fitness should take the generalization issue into account: one usually considers during the identification process more than one experimental condition, also termed *fitness cases*. The fitness is then the average of the error over all fitness cases (as in Figure 1.2). Of course, the computational cost is proportional to the number of fitness cases.

1.2.2 Parametric vs non-parametric identification

Among all search spaces, a special attention is paid to those that can be mapped onto the Euclidean vector space $\mathbb{R}^n$, for some integer n, e.g. the set of polynomials of a given degree, the set of Neural Networks of a fixed architecture. When the target of the identification is a vector of real-valued parameters, the inverse problem is termed *parametric identification*.

When parametric models are used, the identification of parametric models is amenable to classical optimization problems. It can thus often be handled by standard deterministic optimization procedures, ranging from linear programming to non convex constrained optimization algorithms (e.g. Uzawa method, the interior point method, ...). In the worst cases, the well understood Euclidean structure of the search space makes it easy to design local search operators, and to use hill-climbing-like methods for local optimization.

Parametric models may however impose too many restrictions on the solution: limiting the degree of the target polynomial, or imposing the topology of the Neural Network might as well forbid the discovery of a good solution. In such cases, one must therefore explore a larger search space (polynomials of any degree, or Neural Networks of any topology). The inverse problem is then termed *non parametric identification*.

Non parametric model identification involves weird spaces, on which no obvious distance generally exists, and where even the basic topological concept of neighborhood cannot be easily defined. These characteristics of non parametric models make them almost intractable for standard deterministic optimization algorithms.

1.3 MACRO-MECHANICAL RHEOLOGICAL MODELS

1.3.1 Constitutive laws

The design of modern structures requires ever more detailed knowledge of the constitutive properties of materials. Such knowledge is needed to predict through numerical simulations the behavior of the structure under external loadings; and reliable predictions allow for meeting the engineering requirements at a lower cost. On the other hand, a number of new materials (composite materials, polymers), has recently come to a wide use and no accurate model of their behavior is so far available. The identification of a constitutive law for new materials therefore becomes a major challenge to mechanical science and industry.

The design of an accurate law requires considerable insight in mechanics. When a new material is expected by experts to resemble a well known material, the model of the latter is adjusted: its numerical parameters are tuned by minimizing a distance between the observed behavior of the material and the behavior predicted from the law for a set of experimental conditions (as described by Figure 1.2). This stage of identification thus amounts to parametric optimization [GZ85].

Otherwise, an ever stronger sense of mechanics is needed: Brand new models are mostly elaborated by trial and error, where the successive models guessed by mechanical engineers are checked against test experiments; these experiments may in turn suggest new models [LC85].

Or one can also start with a thorough analysis of the mechanical behavior of the material at the microscopic scale; a macroscopic law is then derived from the microscopic model, for instance by homogenization [SPZ87]. However, such models often result in tremendously time-consuming numerical simulations because of their complexity.

Much attention has been paid in designing experiments (the fitness cases defined in Section 1.2.2) in order to adequately check a given law [ZFIKN88]. But searching for an accurate law remains a critical issue, as all above approaches fail in many cases: When the target material does not resemble any previous material; When the microscopic analysis does not allow one to build a tractable model; And when the (sometimes highly nonlinear) behavior of the material prevents from fitting any model suggested by the expert.

1.3.2 Rheological models

We restrict ourselves to general (visco-elasto-plastic) one-dimensional models which can be described by the equation which links the stress function $\sigma(t)$ to the strain applied on this material $\epsilon(t)$ and its time derivative $\dot{\epsilon}(t)$:

$$\sigma(t) = \mathcal{F}(\epsilon(t), \dot{\epsilon}(t)).$$

Rheological models [Per60] allow one to describe most mechanical laws in such a one-dimensional frame; they can be thought of as an assembly of elementary com-

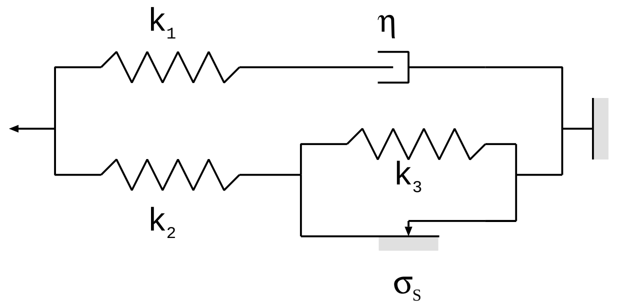


Figure 1.3 *A tentative rheological model of polyethylene*

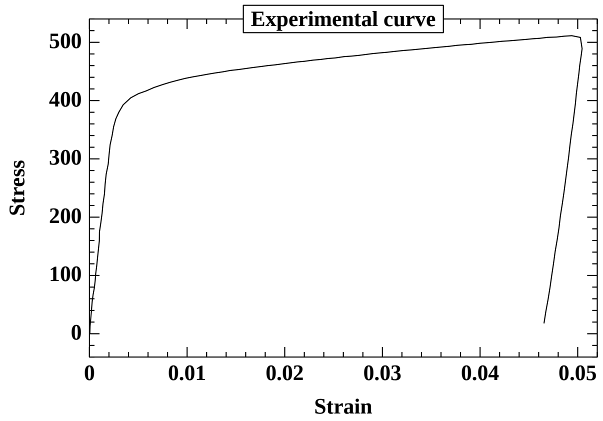


Figure 1.4 *Experimental curve in the strain-stress space. A complete description of the experiment also involves a time dependency, which cannot be seen on the above curve.*

ponents, namely springs, sliders and dashpots, which respectively symbolize elastic, plastic and viscous behaviors. One end of the assembly is kept fixed, and the loading is applied at the other end, whose behavior $\epsilon(t)$,$\sigma(t)$ stands as the response of the model. Figure 1.3 shows an example of such a rheological model, which was proposed for polyethylene [KDVB96].

In particular, a rheological model, far from being a black box, provides a deep understanding of the constitutive properties of the current material and can be rearranged by the expert.

1.3.3 The experimental responses

The response of a rheological material to a given loading is a time-dependent (due to viscosity) irreversible (due to plasticity) phenomenon. Hence the mechanical experiments consists of the loading history ($\epsilon_{exp}(t)$, $t = t_0, ..t_T$ at given times $t = t_0, ..t_T$, and the corresponding observed stress $\sigma_{exp}(t)$). A typical experiment is graphically represented in the strain-stress space (i.e. without the time-dependency information) in Figure 1.4.

1.3.4 Numerical simulation of rheological models

A rheological model ultimately encodes a program mimicking the mechanical behavior of the material. This program takes in input a loading history (the experimental conditions as in Figure 1.2.1), and outputs the corresponding response of the model (the simulated results).

One sequentially executes a two-step process: at each time step, the rheological model is first transformed into a set of partial differential equations. Then, this set of equations plus the the loading at this time-step, is solved numerically, and gives the current response of the model.

Whenever the structure of the rheological model is fixed, the set of partial differential equations can be written once and for all in the algorithm, i.e. *compiled* into an ad hoc program [BOV95]. Hence parametric identification could be done by using such a specialized tool. However, as non-parametric identification involves a new set of differential equations for each new model structure, it is necessary to first *interpret* the model in order to derive the differential equations, before solving it numerically. In this paper, the same interpreter-solver program has been used for both parametric and non-parametric identification to allow for more precise comparisons, It is described in next subsections.

The interpreter

We restrict ourselves to rheological assemblies composed of series and/or parallel branches, which allow for describing most known materials.

The interpreter associates to any rheological model R a set of partial differential equations. Let $\epsilon_i(t)$ and $\sigma_i(t)$ denote the local strain and stress (real-valued functions) attached to component i. The equations attached to the elementary components are:

- The elasticity equation for a spring with stiffness k_i :

$$\sigma_i(t) = k_i \epsilon_i(t)$$

- The viscosity equation for a dashpot with viscosity η_i :

$$\sigma_i(t) = \eta_i \dot{\epsilon}_i(t)$$

- The plasticity equation for a slider with stress threshold σ_i^s, which involves two alternative modes: either the current stress is less than the threshold σ_i^s, in which case the strain derivative is stuck to zero, or the stress is stuck to σ_i^s in which case the strain is undefined:

$$(|\sigma_i(t)| < \sigma_i^s \;\text{ AND }\; \dot{\epsilon}_i(t) = 0) \;\text{ OR }\; (|\sigma_i(t)| = \sigma_i^s \;\text{ AND }\; SGN(\sigma_i(t)).\dot{\epsilon}_i(t) \geq 0).$$

All $\epsilon_i(t)$ and $\sigma_i(t)$ are related due to their connections via series or parallel assemblies. The equation associated to a connecting point relates the stress and strain at this point $\epsilon_p(t), \sigma_p(t)$ to the stresses and strains of the downward connected elements (elementary components or other connecting points) $c_1, c_2, ..c_n$:

- For a series connection, $\begin{cases} \epsilon_p(t) = \epsilon_{c_1}(t) + \epsilon_{c_2}(t) + .. + \epsilon_{c_n}(t). \\ \sigma_p(t) = \sigma_{c_1}(t) = \sigma_{c_2}(t) = .. = \sigma_{c_n}(t). \end{cases}$

- For a parallel connection, $\left\{\begin{array}{lcl} \epsilon_p(t) & = & \epsilon_{c_1}(t) \;=\; \epsilon_{c_2}(t) \;=\; .. \;=\; \epsilon_{c_n}(t). \\ \sigma_p(t) & = & \sigma_{c_1}(t) + \sigma_{c_2}(t) + .. + \sigma_{c_n}(t). \end{array}\right.$

The state of the model, described through $(\sigma_i(t), \epsilon_i(t), \dot{\epsilon}_i(t))_{i=0}^N$, where N is the total number of elements (components and connecting points) in the model, is governed by the above set of partial differential equations.

The main difficulty here is that the structure of the system depends on the loading history: as long as a slider i is not saturated ($|\sigma_i(t)| < \sigma_i^s$), it makes inactive all branches parallel to it and all branches in series below it. The interpreter could proceeds by recursively checking the rheological model (given the underlying semantics of parallel and series assemblies). However, a simpler way consists of sequentially constructing and solving the set of equations at each time step, depending on the state of the model at the previous time step.

The solver

One iteratively computes the response of the model at time t_j, i.e. the global stress $\sigma_0(t_j)$, given the current set of equations plus the equations of the loading at this time step:

1. Derivatives are expressed via finite differences:
$$\dot{\epsilon}_i(t_j) = \frac{\epsilon_i(t_j) - \epsilon_i(t_{j-1})}{t_j - t_{j-1}}.$$
2. The loading history is taken into account:
$$\left\{\begin{array}{l} \epsilon_0(t_j) = \epsilon_{exp}(t_j) \\ \dot{\epsilon}_0(t_j) = \dot{\epsilon}_{exp}(t_j) \end{array}\right.$$
3. These equations, together with the equations written by the interpreter, are handled as a set of linear equations in the unknown $\sigma_i(t_j)$, $\epsilon_i(t_j)$, $\dot{\epsilon}_i(t_j)$ and $\epsilon_i(t_{j-1})$.
4. Solving this set of linear equations gives the state of the model at time t_j, given its previous state at time t_{j-1}, the initial state of the model being $(0, 0, ..0)$.
5. For each slider i, the internal stress $\sigma_i(t_j)$ is compared to threshold σ_i^s; whenever a slider happens to come to or quit saturation ($|\sigma_i(t_j)| < \sigma_i^s$), the system is rebuilt by calling the interpreter.
6. The response $\sigma_0(t)$ for loading $\epsilon_{exp}(t_j), j = 0..T$, is noted $\sigma_R(t, \epsilon_{exp}, t_0, .., t_T)$.

Complexity: The complexity of this resolution amounts to $T \times 2.(3N)^3/3$, where T is the number of time steps of the loading history and N the size of the model (the resolution of a linear system of size n being $2.n^3/3$).

Error: This resolution process involves two kinds of error. First, the mechanical measures are known with a given precision, and at given instants only; and the discretization of the loading history is beyond our control. Second, our handling of derivatives induces numerical errors depending on the current model. We heuristically propose to estimate both kinds of errors *for a given model*, through the difference between the response computed from the whole loading history $\epsilon_{exp}(t), t = t_0, ..t_T$, and the response computed from an excerpt of the same loading history, including only one in two consecutive instants (a loading history typically includes several dozens or a few hundred points):

$$Error = \sum_{t=t_0}^{t_T} |\sigma_R(t, \epsilon_{exp}, t_0, t_1, t_2.., t_T) - \sigma_R(t, \epsilon_{exp}, t_0, t_2, t_4.., t_T)|.$$

This estimate intends to capture both the imprecision coming from the available data, and the error caused by the solver. In what follows, an identification algorithm will be considered successful if the fitness of the best model falls below this estimate of the unavoidable error.

1.4 Parametric identification

In this section, we suppose that the mechanical experts were able to predict the structure of the rheological model for the polyethylene, from some other well-known composite material, but that the real-valued coefficients of that model (i.e. stiffness of springs, viscosity coefficient for a dashpot and threshold for a slider) are to be identified.

The usual method of mechanical engineers is trial-and-error, as the discontinuous form of the differential equations for sliders forbids any calculation of the derivatives of the unknown variables. Hence, Evolutionary Computation is a good candidate here.

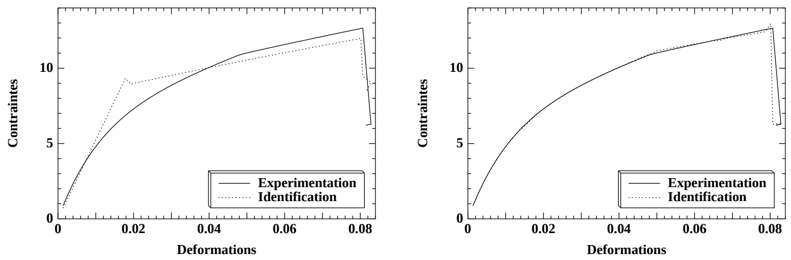
1.4.1 The Evolutionary Algorithm

As the optimization problem reduces here to parametric optimization, a good candidate evolutionary algorithm was Evolution Strategies [Sch81, BS93]. After different tentative settings, our choice was finally a (10+30)-ES: a population of 10 parents produces 30 offspring (3 per parent, regardless of its fitness) by Gaussian mutation only; The standard deviation is self-adapted, i.e. each real-valued parameter "carries" its own standard deviation, which undergoes mutation before the parameter itself gets mutated; The best 10 individuals out of the 40 (parents + offspring) are deterministically selected to become the parents of the next generation. The algorithm stops whenever the fitness of the best individual (error in the time-strain-stress space) becomes lower than the heuristically computed unavoidable error (see above), or after a given number of generations.

1.4.2 The fitness cases

As described in section 1.2.1 (Figure 1.2), a model is evaluated through the difference between the observed stress $\sigma_{exp}(t)$ and the stress $\sigma_R(t, \epsilon_{exp})$ computed from that model according to the experimental loading history:

$$F(R, (\epsilon_{exp}(t), \sigma_{exp}(t))) = \sum_{t=t_0}^{t_T} |\sigma_R(t, \epsilon_{exp}) - \sigma_{exp}(t)|.$$



(a) (b)

Figure 1.5 *Parametric identification: Simulated curves using the best model after 21 and 200 generations.*

	k_1	η	k_2	k_3	σ_S
"Experimental"	790.45	6248.60	150.20	41.60	7.25
Identification	789.989	5947.84	160.243	45.4853	7.00099

Table 1.1 *Results of the parametric identification for the polyethylene*

However, for simplicity reasons, we considered as experimental results the results provided by simulation from accurate numerical algorithm specific to the target rheological model of Figure 1.6 [BOV95].

1.4.3 Results

Out of 21 independent runs, about 70% reached the success criterion (fitness below unavoidable numerical error). The total computation time was in average about 10mn for a run of 500 generations (15000 fitness evaluations) on a Pentium 166 workstation.

Figure 1.4.3 presents, together with the experimental curve, the response curve of the best individual discovered during a typical successful run of the algorithm, at generations 21 (Figure 1.4.3-a) and at generation 200, after "convergence" (Figure 1.4.3-b), while Table 1.4.3 shows the parameters of the best discovered individual, compared to the exact solution. Both curves and coefficients were judged excellent by mechanical engineers, compared to the results of the usual trial-and-error method, even based on the specific numerical algorithm [BOV95].

1.5 Non-parametric identification

In the preceding section, it was assumed that only the parameters of a known rheological model were unknown. But when the new material is very different from any existing material, the structure itself must be looked for, and the identification problem becomes non-parametric.

The first task is to find a representation of the potential solutions covering a large enough subspace of the whole search space, and still amenable to evolutionary search, i.e. where evolution operators like mutation and crossover can be easily and efficiently designed.

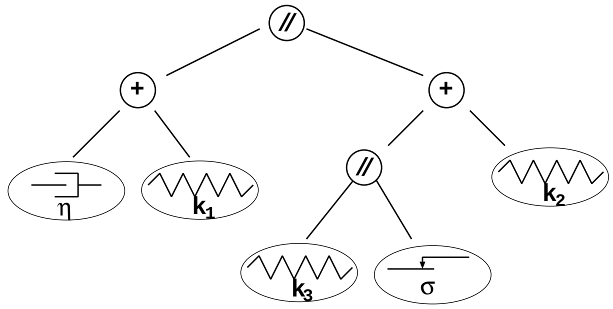


Figure 1.6 *The tree representing the rheological model of polyethylene (Figure 1.3).*

1.5.1 Representation

Going back to the graphical model of rheological models of Figure 1.3 and the considerations detailed in section 1.3.4, and with some knowledge of Genetic Programming [Koz92, Koz94], a first idea is to represent rheological models as trees: the nodes are either the *parallel* or the *serial* connectors, and the terminals are the elementary components of the model (i.e. springs, dashpots or sliders). The model of polyethylene of Figure 1.3 so becomes the tree of Figure 1.6.

This representation of rheological models can represent any rheological model limited to parallel and series assemblies. Moreover, any tree using parallel or series nodes, and spring, dashpot or slider terminals, represents a valid rheological model. Hence, it is straightforward to transpose GP operators to rheological models, and the following were used:

- Initialization up to a given depth, all nodes and terminals chosen randomly;
- Crossover by exchange of randomly chosen sub-trees [Koz92];
- Mutation by random replacement of a sub-tree;
- Mutation by replacing an operator (resp. a terminal) by the (resp. an) other operator (resp. random terminal).

Nevertheless, an important characteristic of the terminals used here (springs, dashpots and sliders) is that they all involve a real-valued parameter that needs to be adjusted precisely. This leads to handle the mutation of real-valued terminal in a specific way, described in next subsection, and called *surface mutation*. By contrast, the mutation operators described above will be called *structural mutations*.

1.5.2 Handling real-valued terminals

Like numerical functions, rheological assemblies allow for values to combine (e.g. the series of two springs with respective stiffness k_1 and k_2 behaves as a spring with stiffness $\frac{k_1 * k_2}{k_1 + k_2}$). The adjustment of numerical values could then be left to random mutation and crossover, as in [Koz92]. However, this leads to a dramatic increase of the size of the trees along evolution; and the fitness computation increases as the cube of the size (section 1.3.4). Such adjustment of constants is therefore much more expensive for rheological identification than for classical regression problems.

	k_1	η	k_2	k_3	σ_S
"Experimental"	790.45	6248.60	150.20	41.60	7.25
Identification	998.892	8698.78	133.085	39.6647	19.0409

Table 1.2 *Results of the non-parametric identification for the polyethylene, when the exact structure has been found.*

A specific mutation operator was thus devised to address the optimization of floating-points values. This operator basically is a random hill-climber, termed surface mutation: Surface mutation modifies all floating-point terminals in a given tree, via the addition of a Gaussian random variable; the standard deviation of the mutation is here again attached to the coefficient, and self-adapted. A surface mutation is considered successful if it results in an improved fitness. The random hill-climber repeatedly performs surface mutations, until a number O of consecutive surface mutations, termed *stubbornness*, is found unsuccessful.

Any crossover or structural mutation is followed by a hill-climbing stage, in order to get the best from the new incoming tree-structure. This strategy transposes in the field of GP the formal memetic approach developed in the frame of GAs in [RS94]: the aim still is to confine the genetic population in the region of local optima with respect to the floating-point values.

This strategy is intermediate between a mutation operator that would modify a single terminal (e.g. by once adding a Gaussian random variable to the coefficient), and the costly complete strategy of running a full parametric optimization (as presented in section 1.4) on any tree after crossover or structural mutation.

1.5.3 Results

For all runs presented here, the population size was set to 200. The initial maximal depth of individuals was set to 12, while offspring could not go beyond depth 50. A standard generational evolution was used: tournament (size 3) selection, crossover rate of 0.4, mutation rate of 0.2, all offspring replacing the parents. The maximum number of generation was set to 300. The total computation time was in average about 40mn on a Pentium 166 workstation.

The stubbornness was set to 2. The influence of population size, stubbornness, and the use of approximated fitness in the early generations are discussed in details in [SSJ$^+$96].

In about 20% of the cases, the success criterion was reached. It is important to notice, though, that the best topologies found in most cases are not the exact ones, and resemble that presented in Figure 1.7: according to mechanical experts, it is due to the lack of *creep* in the experiments, i.e. the part of the experimental curve going back the horizontal axis in Figure 1.4.

Nevertheless, as can be seen in Table 1.5.3, the coefficients found by the best run for which that "good" structure was reached resemble the exact ones, though they are, as expected, not as good as those obtained by the parametric identification. Moreover,

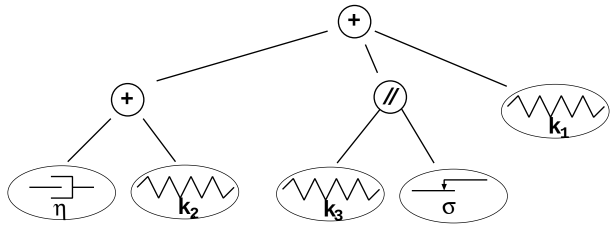


Figure 1.7 *A local optimum for the non-parametric identification: This simpler tree-structure has almost the same behavior than the exact solution when no creep is included in the experimental curve.*

1.6 Conclusion

The point of this paper was the comparison between parametric and non-parametric identification in the case of rheological models. And the first clear conclusion is, of course, that parametric identification gives more accurate results than non-parametric, and at a lower cost!

A first basic remark is that, in any case, and regardless of the "memetic" strategy (see Section 1.5.2), any non-parametric identification solution should in turn become the basis of some parametric identification: the advantages of both approaches would so be combined, and the final accuracy would be that of the parametric part, provided a good structure is found during the non-parametric step.

However, further work in the non-parametric case is concerned with tuning the efficiency of the hill-climbing mutation by allowing the number of surface mutations to increase along evolution, trying to get the same accuracy than the above-mentioned combined approach at the cost of a single non-parametric run.

But non-parametric optimization not only requires more computing power. Actually, the most difficult part of the non-parametric work was the design and implementation of the interpreter-solver program: Hence, for mechanical engineers, the most important, though indirect, consequence of this work is ... the availability of an interpreter able to simulate any rheological model.

As mentioned in Section 1.3.4, the only existing tools can be thought more as "rheological compilers", i.e. program solving the problem for one particular rheological model [BOV95]: As no optimization algorithm was able to handle the identification problem (even in the parametric case), the technical difficulty of designing a "rheological interpreter" probably appeared to be a too large investment to mechanical engineers compared to its potential usefulness.

Yet, the parametric optimization could have been performed by repeated calls (i.e. through file-based argument passing, for instance) to some up-to-date rheological compiler, like the one described in [BOV95] for the polyethylene. The higher accuracy of such *ad hoc* compilers would probably have increased the accuracy of the results. However, though the interpreter was primarily designed for non parametric identification, it now allows straightforward parametric optimization of any further rheological model.

Another perspective of this work is the design of discriminant mechanical experiments in order to decide between candidates rheological models provided by a previous non parametric identification.

REFERENCES

[BOV95] Boytard K., Ouakka A., and Van K. D. (1995) A modelling tool for the mechanical behavior of polyethylene pipe. In Denys R. (ed) *Pipeline Technology.* Elsevier.

[BS93] Bäck T. and Schwefel H.-P. (1993) An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation* 1(1): 1–23.

[FS95] Fadda A. and Schoenauer M. (March 1995) Evolutionary chromatographic law identification by recurrent neural nets. In McDonnell J. R., Reynolds R. G., and Fogel D. B. (eds) *Proceedings of the 4^{th} Annual Conference on Evolutionary Programming*, pages 219–235. MIT Press.

[GZ85] Gittus J. and Zarka J. (eds) (1985) *Modelling small deformations of polycrystals.* Elsevier Applied Sciences Publishers.

[KDVB96] Kichenin J., Dang-Van K., and Boytard K. (1996) Finite element simulation of new two dissipative mechanism models for bulk medium-density polyethylene. *J. of Mat. Sc.* 31: 1653–1661.

[Koz92] Koza J. R. (1992) *Genetic Programming: On the Programming of Computers by means of Natural Evolution.* MIT Press, Massachussetts.

[Koz94] Koza J. R. (1994) *Genetic Programming II: Automatic Discovery of Reusable Programs.* MIT Press, Massachussetts.

[LC85] Lemaitre J. and Chaboche C. (1985) *Mécanique des Matériaux Ssolides.* Dunod.

[Per60] Persoz B. (1960) *Introduction à la rhéologie.* Dunod, Paris.

[RS94] Radcliffe N. J. and Surry P. D. (1994) Formal memetic algorithms. In Fogarthy T. (ed) *Evolutionary Computing: AISB Workshop*, pages 1–16. Springer Verlag LNCS 865.

[Sch81] Schwefel H.-P. (1981) *Numerical Optimization of Computer Models.* John Wiley & Sons, New-York. 1995 – 2^{nd} edition.

[SPZ87] Sanchez-Palencia E. and Zaoui A. (1987) *Homogenization techniques for composite media.* Lecture Notes in Physics, Springer Verlag.

[SSJ+96] Schoenauer M., Sebag M., Jouve F., Lamy B., and Maitournam H. (1996) 23: Evolutionary identification of macro-mechanical models. In Angeline P. J. and K. E. Kinnear J. (eds) *Advances in Genetic Programming II*, pages 467–488. MIT Press, Cambridge, MA.

[ZFIKN88] Zarka J., Frelat J., Inglebert G., and Kasmai-Navidi P. (1988) *A new approach in inelastic analysis of structures.* Martinus Nijhoff Publishers.