IDENTIFICATION OF ELASTOPLASTIC MODEL PARAMETERS IN LARGE DEFORMATION PROBLEMS

P.-P. Jeunechamps, J. Walmag, V. Mathonet, E. Delhez, A.M. Habraken, J.-P. Ponthot, P. Tossings & P. Duysinx

Institute of Mechanics and Civil Engineering, University of Liège, Liège, Belgium, e-mail: PPJeunechamps@ulg.ac.be

Abstract

This paper reports on preliminary results of parameter identification problems in finite element non-linear analyses such as metal forming simulations. Two approaches are compared. The first one is the classic Levenberg-Marquardt algorithm. The second one is a trust-region algorithm based on a quadratic model. The two algorithms are compared on two cases, one of them being an actual experiment.

Keywords: Parameter Identification, Optimization algorithms, Large deformation simulations, elasto-plastic models.

1. Introduction

Simulation of manufacturing processes, here metal forming, has made major progresses during the last years. The virtual simulation of the fabrication process being now quite well established, a natural step ahead consists in trying to find automatic procedures to optimize the manufacturing process. Another perspective is also to take care of the influence of the manufacturing constraints upon the design, which is the long term objective of the present research project. It aims at optimizing the design subject to both service constraints (stiffness and strength) and fabrication constraints. A preliminary stage is to be able to build high fidelity digital models. Complex models require more and more material parameters for behavior laws (e.g. material constitutive and friction laws), which have to be identified numerically from experimental data.

In this material parameters identification process, three steps can be identified. At first, an experimental testing is carried out. The second step consists in building a simulation model of the experiment. Finally, the unknown model parameters are determined to match the experimental data. A standard identification procedure consists in minimizing a given norm (here the Euclidean norm) of the error between the model predictions and the experimental results.

From many points of view the identification process is similar to a structural optimization. Even if the identification problem is generally quasi-unconstrained, it has the same complexity because of the highly nonlinear and implicit character of the functions, which is especially amplified by the large deformation simulation analysis.

In this paper, we use an approach of elastoplastic calculation by finite elements combined with 2 optimization algorithms: a Levenberg-Marquardt algorithm, which is rather classical in the literature for solving identification problem (e.g. Ghouati and Gelin 1998) and a trust-region one (see Conn et al., 2000), which is a rather novel approach at least for structural problems. The results obtained with these two methods are then compared and discussed on two test cases. The first application is an academic test case to validate the identification method. The second one, compression of a cylinder, takes into account an actual experiment. In this application, the material is assumed to be elasto-(visco)-plastic and described by a Norton-Hoff behaviour law and an isotropic strain hardening law. The parameters to be identified are three coefficients of the Norton-Hoff law.

The outline of this paper is as follows: in section 2, the problem formulation of the inverse formulation of parametric optimization is presented. In section 3, the two optimization algorithms used in this paper –Levenberg-Marquardt and a trust-region algorithms– are described. Section 4 presents 2 numerical applications of the optimization process. Finally, conclusions and perspectives for future works are given in section 5.

2. Problem formulation

Identification problems can be stated as the minimization of the Euclidean norm of the weighted difference between the experimental data and the responses obtained with the finite elements simulation:

$$\min_{x_j,\, j=1\ldots n} f(\mathbf{x}) = \frac{1}{2}\left\|\mathbf{r}(\mathbf{x})\right\| = \frac{1}{2}\sqrt{\sum_{i=1}^{m} w_i (u_i^{FE}(\mathbf{x}) - u_i^{EXP})^2} \tag{1}$$

where

- u_i^{EXP} ($i = 1\ldots m$) are the responses measured during the test
- u_i^{FE} ($i = 1\ldots m$) are the responses predicted by the Finite Element (FE) simulation
- $\mathbf{x}$ is the vector of parameters to be identified (design variables)
- w_i ($i = 1\ldots m$) are positive weight factors for dimensionality and confidence in experimental points
- m is the number of experimental points
- n is the number of design variables

The choice of the Euclidean norm as objective function influences the efficiency of the optimization methods and leads to a fast convergence process for the main optimization methods (Kleinermann, 2000).

The problem stated by Eq.(1) is a quasi-unconstrained one i.e. it is subject to only side constraints on design variables. It's a highly non linear and implicit problem with respect to the design variables, which implies several difficulties to evaluate the objective function and furthermore its sensitivity to the design parameters. This framework forces to use specific finite element codes (e.g. Lagamine, a large deformation FE codes developed in University of Liège, see H. Grober et al., 1985).

In identification problems the number of design variables is relatively low (around 10 design variables) compared to problems of structural optimization, while the dimension of the response vector can be very high (up to 1000 measured data). Moreover, due to the experimental process, the data are polluted by some "noise". It turns out that the residue vector in Eq.(1) can never be equalized to zero. Finally because of the noisy data and of the strongly non linear character of the objective function, attention must be paid to the many local optima of the objective function.

3. Optimization algorithms

The Levenberg-Marquardt algorithm is one of the most widely used method to adress identification problems given by Eq.(1) (e.g. Ghouati and Gélin, 1998). It will therefore be used as a reference in this paper.

In the Levenberg-Marquardt method, global convergence towards a stationary point of the objective function is obtained thanks to a stabilization of the Gauss-Newton method through a regularisation term. Most other optimization algorithms rely on costly line search strategies to ensure convergence. In this paper, we have implemented a trust-region algorithm, which is a rather novel method in structural applications.

3.1 Levenberg-Marquardt algorithm

A quadratic approximation of Eq.(1) is build. At step k, the approximation can be written as follows:

$$m^{(k)}(\mathbf{x}) = f(\mathbf{x}^{(k)}) + \nabla f(\mathbf{x}^{(k)})^T (\mathbf{x} - \mathbf{x}^{(k)}) + 1/2 (\mathbf{x} - \mathbf{x}^{(k)})^T \mathbf{H}^{(k)} (\mathbf{x} - \mathbf{x}^{(k)}) \tag{2}$$

where

- $\nabla f(\mathbf{x}) = \mathbf{J}^T(\mathbf{x})\mathbf{r}(\mathbf{x})$ is the gradient of the objective function with the Jacobian matrix of the residue vector **J** defined as:

$$\mathbf{J}(\mathbf{x})\big]_{ij} = \frac{\partial r_i(\mathbf{x})}{\partial x_j} = \frac{\partial u_i^{FE}(\mathbf{x})}{\partial x_j} \tag{3}$$

  In the used finite element code (Lagamine, see H. Grober et al, 1985), the sensitivity is computed with a semi-analytical method based on a direct differentiation method (Tortorelli et al. 1994) adapted to large deformation problems by Kleinermann (2000).

- **H**(**x**) is the Hessian matrix or its approximation. The exact expression of the Hessian matrix is given by:

$$\mathbf{H}(\mathbf{x}) = \nabla^2 f(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) + \sum_{l=1}^{m} r_l(\mathbf{x}) \nabla^2 r_l(\mathbf{x}) \tag{4}$$

  Estimations of **H**(**x**) can be constructed using quasi-Newton techniques such as DFP, BFGS… But a usual approximation is obtained by neglecting the contribution of second order terms of the residue (Gauss-Newton approximation):

$$\mathbf{H}(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) \tag{5}$$

  This approximation has the advantage that no additional computation is necessary.

Using the Gauss-Newton approximation, the quadratic model can be written as:

$$m^{(k)}(\mathbf{x}) = 1/2 \left\| \mathbf{J}(\mathbf{x}^{(k)})\mathbf{d}^{(k)} + \mathbf{r}(\mathbf{x}^{(k)}) \right\|^2 \tag{6}$$

where $\mathbf{d}^{(k)} = \mathbf{x} - \mathbf{x}^{(k)}$. Optimality conditions of the minimum problem of Eq.(6) lead to the Gauss-Newton classical iteration scheme:

$$\mathbf{J}(\mathbf{x}^{(k)})^T \mathbf{J}(\mathbf{x}^{(k)})\mathbf{d}^{(k)} = -\mathbf{J}(\mathbf{x}^{(k)})^T \mathbf{r}(\mathbf{x}^{(k)}) \tag{7}$$

The drawback of this method is that the convergence is not guaranteed. The Levenberg-Marquardt method consists in the addition of a parameter $l^{(k)}$ to stabilize the Gauss-Newton method in order to penalize the step size. The objective function has now the form:

$$m^{(k)}(\mathbf{x}) = 1/2 \left\| \mathbf{J}(\mathbf{x}^{(k)})\mathbf{d}^{(k)} + \mathbf{r}(\mathbf{x}^{(k)}) \right\|^2 + l^{(k)}/2 \left\| \mathbf{d}^{(k)} \right\|^2 \tag{8}$$

and the step $\mathbf{d}^{(k)} = \mathbf{x} - \mathbf{x}^{(k)}$ is given by the iterative solution of:

$$(\mathbf{J}(\mathbf{x}^{(k)})^T \mathbf{J}(\mathbf{x}^{(k)}) + l^{(k)}\mathbf{I})\mathbf{d}^{(k)} = -\mathbf{J}(\mathbf{x}^{(k)})^T \mathbf{r}(\mathbf{x}^{(k)}) \tag{9}$$

For large $l^{(k)}$, the parameters variations are smaller. The stabilization of the Gauss-Newton method is then achieved with a sub-iteration process. At each sub-iteration, Eq.(9) is solved to find a new set of design variables. Then the objective function is calculated and if its value is inferior to the previous one, then the sub-iteration is convergent and one proceeds to next iteration. If not, $l^{(k)}$ is increased and a new approximation of the parameters is obtained. Then the convergence of the sub-iteration is checked. The procedure is repeated until convergence.

3.2 Trust region algorithm

3.2.1 Principle (see Conn et al, 2000)

The trust-region method is an iterative procedure in which the objective function $f(\mathbf{x})$ is approximated in a suitable neighbourhood (the trust-region of radius $D^{(k)}$) of the current iteration point by a model $m^{(k)}(\mathbf{x})$ that is easier to handle than $f(\mathbf{x})$ itself. In this neighbourhood, one searches for design variables increment $\mathbf{s}_k$ that sufficiently reduces the value of the model while satisfying the bound $\|\mathbf{s}^{(k)}\| \le D^{(k)}$. The objective function is then computed at the resulting trial point $\mathbf{x}^{(k)} + \mathbf{s}^{(k)}$. This trial point is accepted as the new iterate if it corresponds to a reduction of the objective function. Comparison of the expected (based on the model $m^{(k)}(\mathbf{x})$) and the actual (based on $f(\mathbf{x})$) reductions of the objective function provided is used to update the radius of the trust-region. If the fidelity of the model is good, the radius of the trust-region is increased or left unchanged. If not, the trust-region is contracted, in the hope that the model provides a better prediction in this smaller region.

This algorithm is a globally convergent one, meaning that convergence to a stationary point of the objective function is guaranteed. (the gradient tends to zero).

A difficulty of this method is the choice of the shape and size of the initial trust-region especially when variables are badly scaled. It can be partially solved by choosing appropriate values for the characteristic variations of the design variables. So, a variable scaling gives the same relative weight to each direction of the design space.

3.2.2 Trust region algorithm

We define the trust region as the ball $B^{(k)} = \left\{ \mathbf{x} \in \Re^n \,\middle|\, \left\| \mathbf{x} - \mathbf{x}^{(k)} \right\| \le D^{(k)} \right\}$

The principle of the trust-region algorithm is described in the following scheme (see Walmag, 2003):

Step 0: Initialization. An initial point $\mathbf{x}_0$ and an initial trust-region radius $D^{(0)}$ are chosen. Compute $f(\mathbf{x}^{(0)})$.

Step 1: Model definition. Define a model $m^{(k)}$ within the trust region $B^{(k)}$. In the present case, the model is quadratic and has the form given by Eq.(2) where the Hessian matrix is approached using Eq.(5).

Step 2: Step calculation. Compute a step $\mathbf{s}^{(k)}$ that "sufficiently reduces the model" $m^{(k)}$ and such that $\mathbf{x}^{(k)} + \mathbf{s}^{(k)} \in B^{(k)}$. Solve the minimization problem and let:

$$\mathbf{x}^* = \underset{\|\mathbf{x}-\mathbf{x}^{(k)}\| \le D^{(k)}}{\arg\min}\ m^{(k)}(\mathbf{x}) \tag{10}$$

The restriction forces the solution to remain within the trust-region defined by the trust-region radius $D^{(k)}$.

Step 3: Acceptance of the trial point. Compute $f(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})$ and define the ratio r that measures the discrepancy of the quadratic model:

$$r^{(k)} = \frac{f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})}{m^{(k)}(\mathbf{x}^{(k)}) - m^{(k)}(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})} \tag{12}$$

If $r^{(k)} \ge 10^{-2}$, there is at least a reduction of the objective function and $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$ is accepted. The iteration is said to be *successful*. Otherwise, it is rejected and $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$. The iteration is said to be *unsuccessful*.

Step 4: Trust-region radius update. If the model has a high fidelity (resp. low fidelity), then the trust-region is expanded (resp. contracted)

$$\begin{array}{lll} \textit{if} & 0.95 \le r^{(k)} \le 1.05 & \textit{then} \quad D^{(k+1)} := 2D^{(k)} \\ \textit{if} & r^{(k)} < 0.01 & \textit{then} \quad D^{(k+1)} := D^{(k)} / 2 \\ \textit{else} & D^{(k+1)} := D^{(k)} & \end{array} \tag{13}$$

Step 5: Go to next step. Increment k by one and go to Step 1.

4. Numerical results

All the numerical simulations were carried out with Lagamine, a large-deformation computer code developed at University of Liège (see P. Moureaux et al, 2002). Two test-cases were analysed.

4.1 Academic test-case

4.1.1 Description of the problem

The first application is a very simple test-case that allows validating the two proposed optimization methods. The actual experiment is replaced by a numerical one from which the force-displacement curve and the pseudo experimental points are generated. Then the reference parameters are perturbed to generate a starting point and the identification problem is used to retrieve the reference force-displacement curve.

In this benchmark the simulated material is a simple elastoplastic material with linear hardening similar to steel at room temperature. The one-dimension behaviour law "*s*-*e*" has thus the form given by Eq.(14):

$$\begin{array}{ll} s^{el} = Ee^{el} & s < s_0 \\ s^{pl} = s_0 + h(e^{pl} - s_0/E) & s \ge s_0 \end{array} \tag{14}$$

with E the Young's modulus, h the hardening coefficient and σ_0 the elastic limit. The elastic limit s_0 is equal to 700 MPa.

The geometry for this test-case is represented in Fig.1. It is a plane strain triangle. Its inferior part is fixed and a horizontal force is applied at the upper corner. Fig.1a is the initial configuration while Fig.1b gives the Von-Mises stress J_2 after deformation. A coarse mesh is used (11 nodes and 7 elements) in order to be able to perform fast simulation runs and to make the set up of the methods. The CPU time for one simulation is around 10 seconds on SGI Origin 3800-NIC.

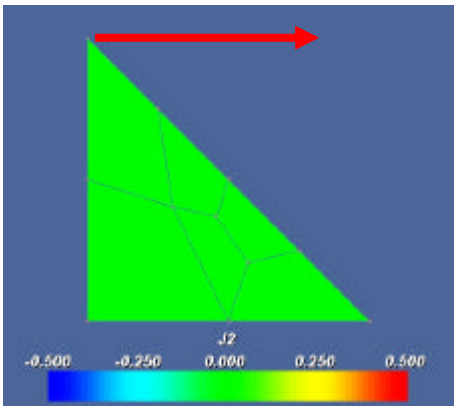

Figure 1a: initial configuration

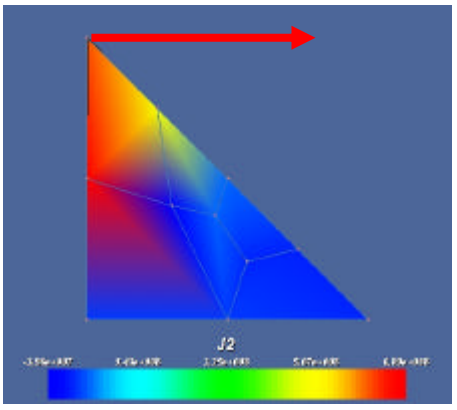

Figure 1b: Von-Mises stress

4.1.2 Optimization results

The two design variables X_1 and X_2 are the Young's modulus E and the hardening coefficient h. Their optimum values are respectively 200,000 and 300 MPa.

The convergence results with the Levenberg-Marquardt trust-region algorithms are presented in Tab.1a and Tab.1b. The histories of the objective function and of the parameters are plotted in Fig.2a and Fig.2b.

The trust-region algorithm converges to a local minimum closer to the global one than the Levenberg method as suggested by Fig.2a and Fig.2b. In The value of the objective function obtained with Levenberg-Marquardt is larger than that of the trust-region method and the relative value of the second design variables has not converged to the reference value of 1 for the Levenberg-Marquardt algorithm. Our explanation of the good convergence properties of trust-region algorithm are is that an adequate choice for the parameter variation scaling prevents the method from falling into a local optimum.

Table 1a: convergence history for the Levenberg-Marquardt method

Iteration	X_1	X_2	Objective function
0	185,000	250.000	0.72724
1	194,243	259.540	0.39889
2	196,376	272.059	0.34319
3	196,377	272.060	0.34312

Table 1b: convergence history for the trust-region method

Iteration	X_1	X_2	Objective function
0	185,000	250.000	0.727240
1	194,816	259.540	0.398894
2	199,378	269.783	0.338630
3	201,156	278.570	0.308183
4	201,658	290.815	0.303525
5	202,189	296.475	0.292012
6	202,189	296.474	0.286654

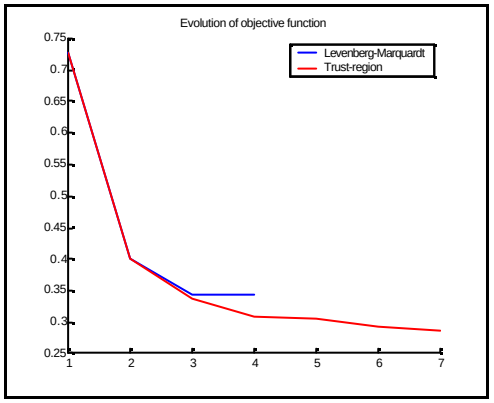

Figure 2a: History of the objective function

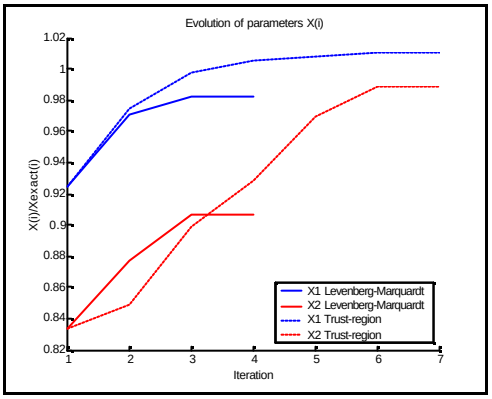

Figure 2b: History of the parameters

4.2 Identification of the parameters of a Norton-Hoff model

4.2.1 Description of the problem

The second test-case is based on an experiment carried out at the University of Liège. It consists in a compression test at high temperature (800°C). The problem is represented in Fig.4a (geometry before deformation). It is a 60% compression problem of a cylinder. Because of symmetry only a quarter of the structure is modelled. So on the bottom line symmetry conditions are applied, as well as on the vertical axisymmetrical axis and the displacement of the punch on the upper part of the cylinder is such that we have a constant strain rate of $5s^{-1}$. An elasto-visco-plastic model is selected for the formulation. The one-dimension behaviour law "*s*-*e*" for such a material has the form given by Eq.(15) and is shown in Fig.3:

$$\begin{aligned} s^{el} &= E e^{el} && \text{in the elastic part} \\ s^{viscopl} &= e^{p_4} \exp(-p_1 e)\sqrt{3}\, p_2 (\sqrt{3}\dot{e})^{p_3} && \text{in the viscoplastic domain} \end{aligned} \qquad (15)$$

with E the Young's modulus, $p_1 \ldots p_4$ the Norton-Hoff coefficients, and $\dot{e}$ the strain rate. The model introduces 4 parameters to be determined by optimization. However it is impossible to determine p_3 since a single experimental curve has been used in this preliminary study. Fig.4b shows the deformed configuration and the final Von-Mises stress (J_2) distribution. The finite element model has 204 nodes and 202 elements and the CPU time for one simulation is 15 minutes on SGI Origin 3800-NIC.

4.2.2 Optimization results

The design variables X_1, X_2 and X_3 are the Norton-Hoff parameters p_1, p_2 and p_4 of Eq.(15). Since the data are perturbed by noise, remark that finding a perfect curve fitting is impossible except in a minimization way. The results obtained the Levenberg-Marquardt algorithm (resp. trust-region algorithm) are presented in Tab.2a (resp. Tab.2b). The evolutions of the objective function and of the parameters are represented in Fig.5a and Fig.5b. The fitting of the experimental curve is shown in Fig.6.

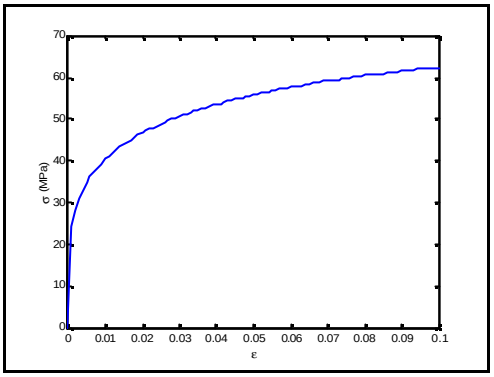

Figure 3: Behaviour law of the material.

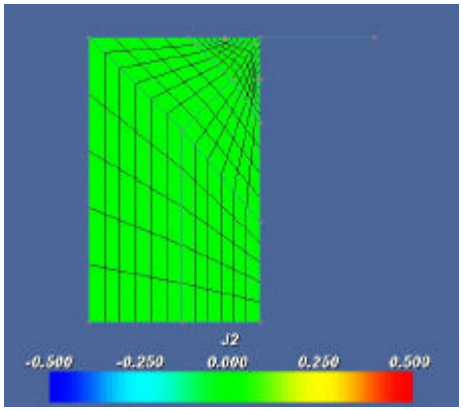

Figure 4a: Initial configuration

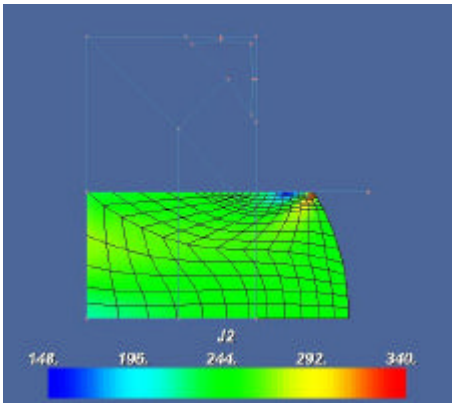

Figure 4b: Von-Mises stresses

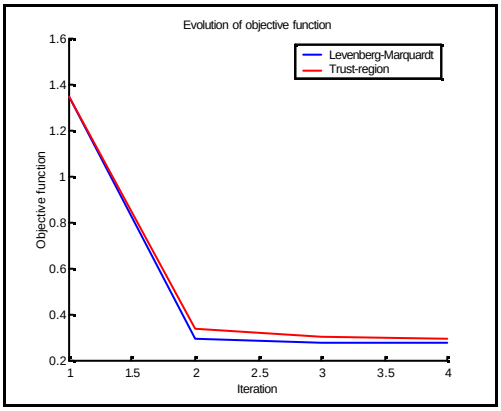

Figure 5a: History of the objective function

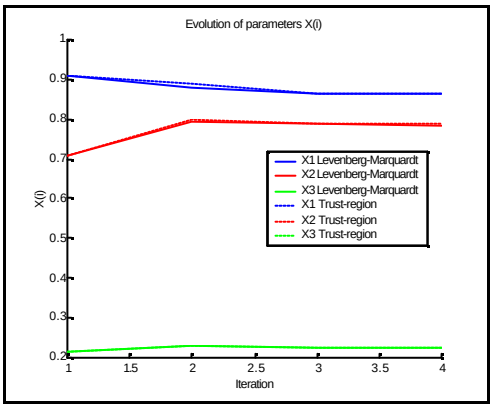

Figure 5b: History of the parameters

Table 2a: convergence history for the Levenberg-Marquardt method

Iteration	X_1	X_2	X_3	Objective function
0	0.9100	0.7097	0.2160	1.34497
1	0.8778	0.79488	0.2280	0.290567
2	0.8636	0.7860	0.2232	0.272296
3	0.8611	0.7851	0.2226	0.272292

Table 2b: convergence history for the trust-region method

Iteration	X_1	X_2	X_3	Objective function
0	0.910000	0.709700	0.216000	1.34497
1	0.889574	0.799966	0.230830	0.336316
2	0.866621	0.787157	0.223757	0.306930
3	0.866589	0.787218	0.223733	0.300467

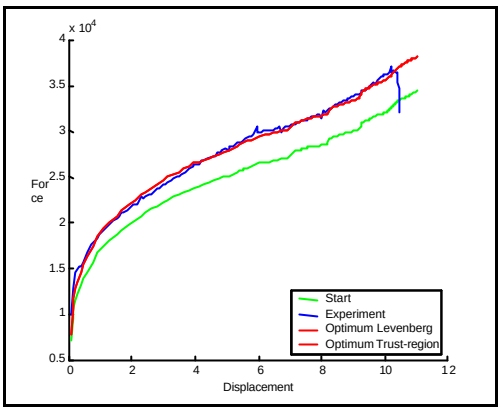

Figure 6: Curve force-displacement

For this problem, the convergence of both methods is very fast (convergence in 3 iterations) and the obtained values for the objective function and the optimal values of the parameters are very close. Remark that for both solutions, the minimization approach yields a nice filtering of the experimental noise.

4.3 Comparison of the cost of two methods

Tab.3 compares the costs of the two methods. The cost of a method is evaluated in terms of number of evaluations of the objective function, i.e. the number of finite element simulations. Most of the CPU time (more than 95%) is indeed spent in the finite element analysis in iterations and in sub-iterations. The extra resources requested by optimization algorithms proper are negligible. Tab.3 compares the number of iterations and the number of objective function evaluations.

Tab.3 shows that the two methods have very similar costs for the second test case. However for the first case, even if the trust region method had a higher cost, but provided a more accurate solution. So the larger cost reflects the improvement of the solution (see Fig.3 and Fig.4).

Table 3: comparison of the cost of the two methods

	Levenberg-Marquardt method		Trust region method	
	Test case 1	Test case 2	Test case 1	Test case 2
Iterations	3	3	6	3
Evaluations of the objective function	6	5	12	5

5. Conclusions

For the identification problems presented in this paper, the trust region method is superior to Levenberg-Maquardt method from different points of view. Firstly, by means of an adequate choice of variation scale of the design variables, the trust region method is less sensitive to local optima. Because of its global convergence properties, it is essentially robust method. The most important disadvantage with respect to use than Levenberg-Marquardt is that it needs an a priori knowledge of the problem to give the initial trust region adequate shape and size. Future works will be devoted to find automatic strategies for the initial trust-region radius to avoid unnecessary sub-iterations (and then finite element simulations). Additional work will be devoted to improve the trust-region adaptation strategy to explore the best region of the design space and so accelerate the convergence.

Finally, we would like to compare the trust-region method with GBMMA solver (see Bruyneel and al, 2002) which offers more flexible and general approach and is very efficient for structural optimization problems.

Acknowledgements

The work presented in this paper has been sponsored by the Research Council of University of Liège (Impulse project 2002 OPFA - Optimization of Product and Fabrication Processes).

References

1. Ghouati O and Gelin J.-C. 1998. Identification of material parameters directly from metal forming processes. *Journal of Materials Processing Technology*, 80-81:560-564.
2. Kleinermann, J.-P., 2000. Identification paramétrique et optimisation des procédés de mise à forme par problèmes inverses, Thèse de doctorat, université de Liège, Belgique.
3. Conn, A. R. Gould, N. I. M. and Toint, P. L. 2000. *Trust-Region Methods.* MPS/SIAM Series on Optimization. SIAM.
4. Walmag, J., 2003. *Méthodes numériques d'optimisation appliquées à l'identification des paramètres d'un modèle dynamique*, travail de fin d'études, université de Liège, Belgique.
5. Bruyneel, M., Duysinx, P. and Fleury, C., 2002. *A family of MMA approximations for structural optimization*. Structural and Multidisciplinary Optimization. 24: 263-276.
6. D.A. Tortorelli, P. Michaleris, C.A. Vidal., 1994. Tangent operators and design sensitivities for transient non-linear coupled problems with application to elastoplasticity. *Int J. num. Meth. In Eng., 37:2471-2499.*
7. Numerical Simulation of Metal Forming Processes.H. Grober, S. Cescotto, R. Charlier, M. Bourdouxhe, A.M. Habraken. Proc. Intl. Conf. on Nonlinear Mechanics, Shanghai, (1985)
8. Identification d'une loi thermo-élasto-viscoplastique en vue de la modélisation du laminage à chaud du cuivre. P. Moureaux, S. Moto Mpong, M. Remy, C. Bouffioux, J. Lecomte, A.M. Habraken, Journal de Physique IV France 12 (2002), Pr11-361-368