Scientific Paper

24th International Conference on Material Forming (ESAFORM 2021)

On the optimization efficiency for the inverse identification of constitutive model parameters

M.G. Oliveira[a,b,*], J.M.P. Martins[a,b], B. Coelho[a,b], S. Thuillier[b], A. Andrade-Campos[a]

[a]Centre for Mechanical Technology and Automation (TEMA), Department of Mechanical Engineering, University of Aveiro, 3180-193, Aveiro, Portugal
[b]Univ. Bretagne Sud, UMR CNRS 6027, IRDL, F-56100, Lorient, France

* Corresponding author: Miguel G. Oliveira. E-mail address: oliveiramiguel@ua.pt

Abstract. The development of full-field measurement techniques paved the way for the design of new mechanical tests. However, because these mechanical tests provide heterogeneous strain fields, no close-form solution between the measured deformation fields and the constitutive parameters exists. Therefore, inverse identification techniques should be used to calibrate constitutive models, such as the widely known finite element model updating (FEMU) and the virtual fields method (VFM). Although these inverse identification techniques follow distinct approaches to explore full-field measurements, they all require using an optimisation technique to find the optimum set of material parameters. Nonetheless, the choice of a suitable optimisation technique lacks attention and proper research. Simply, most studies tend to use a least-squares gradient-based optimisation technique, such as the Levenberg-Marquardt algorithm. This work analyses optimisation techniques for the inverse identification of constitutive model parameters, both gradient-based and -free algorithms. To avoid needless implementation and take advantage of highly developed programming languages, the optimisation algorithms available in optimisation libraries are used. A FEMU based approach is considered in the calibration of a thermoelastoviscoplastic model. The material parameters that govern regarding strain hardening, temperature and strain rate are identified. Results are discussed in terms of efficiency and the robustness of the optimisation processes.

Keywords: optimisation algorithms; finite element model updating; heterogeneous thermomechanical tests; thermoelastoviscoplasticity.

1 Introduction

The mechanical behaviour of sheet metals is typically sensitive to the effects of strain, strain rate and temperature. Recently, due to the growth of heat-assisted manufacturing processes [1], strain rate and temperature effects have gained more impact. Hence, accurately predicting sheet metals' mechanical behaviour, under a wide range of strain rates and temperatures, is increasingly relevant to advance manufacturing processes. To predict such behaviour, thermoelastoviscoplastic constitutive models, characterized by their nonlinearity and a large number of material parameters, can be applied. However, based on such models' traditional calibration, it is required an extensive database with tests performed at a broad range of strain rates and temperatures [2].

The use of full-field measurement techniques, heterogeneous mechanical tests, and inverse methods has reduced the number of experimental tests required [3]. Applying full-field measurements techniques, such as digital image correlation (DIC), the entire displacement fields at the specimen's surface can be recorded and directly used to calibrate a constitutive model. The information obtained from these tests can further be enriched using temperature measurements. Later, applying inverse methods, such as the finite element model updating (FEMU), the material parameters used to calibrate constitutive models can be retrieved [4].

Solely using full-field measurements with inverse methods does not guarantee that suitable material parameters are found. An automatic strategy to find these material parameters is required by using optimisation algorithms. However, most studies calibrating constitutive models tend to overlook the importance of optimisation algorithms, by resorting to familiar ones, such as gradient-based least-squares algorithms [5]. These algorithms tend to perform well and are suitable for nonlinear least-squares problems such as the ones formulated using the mentioned inverse methods, but also present disadvantages. A few studies have explored the use of other optimisation algorithms and strategies, such as direct-search and stochastic methods [6,7].

This paper aims to implement different optimisation algorithms in the calibration of a thermoelastoviscoplastic constitutive model. This methodology uses three heterogeneous thermomechanical tests performed at different average strain rates. The finite element model updating method is used as the inverse method, and three optimisation algorithms are applied in the optimisation

procedure. The results obtained with different optimisations algorithms are compared based on the achieved solution, convergence and successful try-outs.

2 Methodology

2.1 Heterogeneous Thermomechanical Test

Three heterogeneous thermomechanical tests are used as reference data to simulate experimental data [8]. Nonetheless, this reference data is based on tests performed on a Gleeble 3500 thermomechanical simulator, that uses a hydraulic servo system able to impose tension or compression forces, as well as a direct resistance heating system [9]. A uniaxial tension loading is imposed on the specimen, with a heterogeneous temperature field. Finally, the tests are performed at different average strain rates 10^{-2}, 10^{-3}, and 10^{-4} s^{-1}. The temperature field is imposed through the direct resistance heating system, controlling and maintaining the temperature at the centre of the specimen during the test. The specimen's remaining part presents a temperature gradient due to the machine's grips' water-cooling system. This procedure's added value is the temperature gradient triggering a heterogeneous deformation, providing information on the material's mechanical behaviour for different temperatures and strain rates.

The reference data is generated by a finite element model of the tests, whose configuration is presented in Fig. 1. For simplicity, the finite element model is restricted only to the region of interest (ROI), defined by a width of 28 mm and length of 60 mm. The finite element model is two-dimensional, assuming a plane stress formulation. Abaqus/Standard [10] software is used in the finite element analysis, and the four-node bilinear plane stress quadrilateral element CPS4 is used, with a large strain formulation. The finite element mesh is composed of 1680 elements. Displacement-driven boundary conditions are imposed on the finite element model's extremities, at -30 and 30 mm of the reference coordinate x (origin at the specimen's centre).

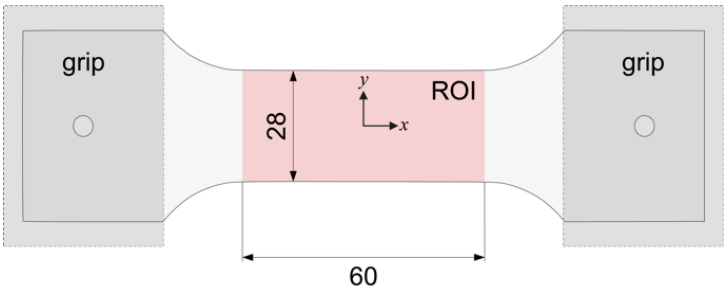

Fig. 1. Schematic representation of the specimen used in the heterogeneous thermomechanical tests (dimensions in mm). The grips are represented on the extremities of the specimen, as well as the region of interest (ROI) defined for the finite element model [8].

The temperature field acquired with Gleeble equipment usually presents a parabolic shape, symmetrical about the specimen's centre. Measurements of three thermocouples placed at -40, 0 and 40 mm of the reference coordinate x, confirm the approximate symmetrical and parabolic shape of the profile along the specimen's length. Variations of the temperature field along the width of the specimen are neglected. Additionally, the temperature field was confirmed to remain constant throughout the deformation process [8]. Because of its parabolic shape, each test's temperature field can be described by a second-order polynomial, as presented in Fig. 2, between -30 and 30 mm of the reference coordinate x. For the three tests, the maximum temperature of approximately 500 ºC is reached at the specimen's centre, decreasing to around 360 ºC at -30 and 30 mm of the reference coordinate x. Therefore, the temperature field is imposed in the finite element model through the second-order polynomial and the nodes' temperatures remain constant throughout the test.

The Johnson-Cook thermoelastoviscoplastic constitutive model is adopted [11]. The model is characterized by a multiplicative formulation, decomposing the flow stress evolution in three terms: strain hardening, temperature and strain rate. It can be written as

$$\sigma_{\mathrm{y}} = [A + B(\overline{\varepsilon}^{\mathrm{p}})^{n}]\left[1 - \left(\frac{T - T_{\mathrm{tr}}}{T_{\mathrm{m}} - T_{\mathrm{tr}}}\right)^{m}\right]\left[1 + C\ln\left(\frac{\dot{\overline{\varepsilon}}^{\mathrm{p}}}{\dot{\varepsilon}_{0}}\right)\right], \tag{1}$$

where $\overline{\varepsilon}^{\mathrm{p}}$ is the equivalent plastic strain and $\dot{\overline{\varepsilon}}^{\mathrm{p}}$ the equivalent plastic strain rate. The first term describes the strain hardening effect, modelled by the material parameters A, B, and n. The temperature effect is modelled by the temperature T, the transition temperature T_{tr}, governing the threshold of temperature effect, the melting temperature T_{m}, and the exponent m. Lastly, the strain rate sensitivity is modelled by the material parameters C and $\dot{\varepsilon}_{0}$, defining the threshold of strain rate dependence.

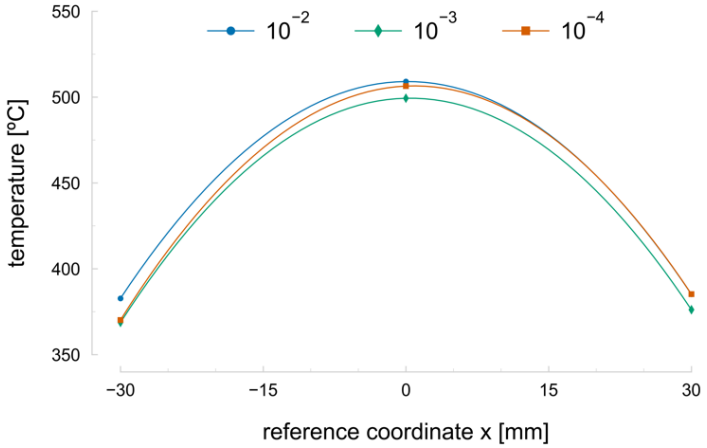

Fig. 2. Longitudinal temperature field of heterogeneous thermomechanical specimens longitudinal temperature field described by second-order polynomials. Each curve is identified from the average strain rate of each test (in s^{-1}).

The reference data is created using a set of reference parameters [8], characteristic of DP980 dual-phase steel (see Table 1). The material is considered isotropic, and Hooke's law describes the elastic behaviour and von Mises yield criterion is adopted. The elastic properties of the material, Young's modulus E and Poisson's ratio ν, are known a priori, assumed to be constant in the temperature range of study and equal to 210 GPa and 0.3, respectively.

Table 1. Reference set of parameters used in the Johnson-Cook thermoelastoviscoplastic constitutive model to generate the reference data [8].

Strain hardening			Temperature			Strain rate	
A [MPa]	B [MPa]	n [-]	T_{tr} [ºC]	T_m [ºC]	m [-]	C [-]	$\dot{\varepsilon}_0$ [s^{-1}]
205.210	1124.000	0.092	25	1000	1.360	0.050	0.001

Then, the reference data is generated using the temperature field and the reference set of parameters. This reference data is composed of the displacement field of the ROI (longitudinal and transversal displacements) and the load for each time instant. The reference load for the tests performed at three different average strain rates is presented in Fig. 3. A different number of time instants composes the tests' reference data, specifically, 47, 58, and 62, respectively, for tests with an average strain rate of 10^{-2}, 10^{-3}, and 10^{-4} s^{-1}. It can be observed that the load is sensitive to the average strain rate, as an increase leads to higher values of maximum load.

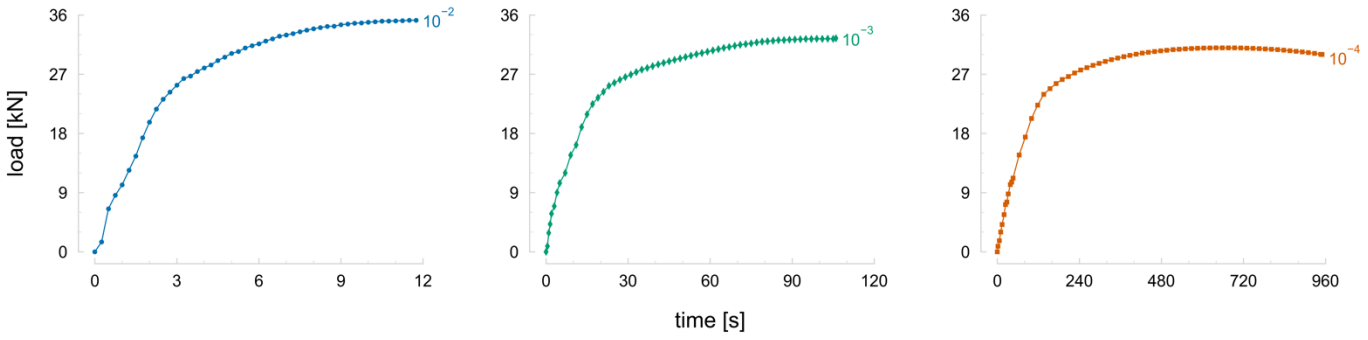

Fig. 3. Load evolution throughout the deformation process of the heterogeneous thermomechanical tests performed at average strain rate of 10^{-2} (left), 10^{-3} (centre), and 10^{-4} s^{-1} (right).

2.2 Finite Element Model Updating

The finite element model updating (FEMU) is used to calibrate the constitutive model. This method is based on the simple idea of iteratively adjusting the finite element model's unknown material parameters to minimise the difference between experimental and numerical results. The FEMU has been largely adopted in many different applications, partly because of its ease of implementation and flexibility. The objective function to be minimised can be composed of different data, such as strain, displacement and load signal. This flexibility has contributed to an increase in the number of formulations presented in the literature. Recently, the use of the FEMU combined with full-field measurements has mainly been used. In that regard, the adopted objective function can be written as

$$\varphi(\boldsymbol{\chi}) = \frac{1}{t}\sum_{\mathrm{i}=0}^{t}\left\{\frac{1}{p}\sum_{\mathrm{j}=0}^{p}\left[\frac{1}{q}\sum_{\mathrm{k}=0}^{q}\left[\left(\frac{\varepsilon_{\mathrm{xx}}^{\mathrm{num}}(\boldsymbol{\chi})-\varepsilon_{\mathrm{xx}}^{\mathrm{exp}}}{\varepsilon_{\mathrm{max}}^{\mathrm{exp}}}\right)^2+\left(\frac{\varepsilon_{\mathrm{yy}}^{\mathrm{num}}(\boldsymbol{\chi})-\varepsilon_{\mathrm{yy}}^{\mathrm{exp}}}{\varepsilon_{\mathrm{max}}^{\mathrm{exp}}}\right)^2+\left(\frac{\varepsilon_{\mathrm{xy}}^{\mathrm{num}}(\boldsymbol{\chi})-\varepsilon_{\mathrm{xy}}^{\mathrm{exp}}}{\varepsilon_{\mathrm{max}}^{\mathrm{exp}}}\right)^2\right]_k+\left(\frac{F^{\mathrm{num}}(\boldsymbol{\chi})-F^{\mathrm{exp}}}{F_{\mathrm{max}}^{\mathrm{exp}}}\right)^2\right]_j\right\}_i, \qquad (2)$$

where $\boldsymbol{\chi} = (A, B, n, m, C)$ is the vector of optimization variables. The variables t, p, and q are the number of tests, time instants, and in-plane measurement points. To distinguish between experimental (reference) and numerical variables, the superscripts "exp" and "num" are used respectively. The variable $\varepsilon_{\mathrm{max}}^{\mathrm{exp}}$ represents the maximum strain value of all in-plane components and time instants for each test. Analogously, $F_{\mathrm{max}}^{\mathrm{exp}}$ represents the maximum load value of all time instants for each test. Because the displacement field represents the raw data, the strain field is computed from the displacement field using a total Lagrangian formulation. The reference strain field is computed before the calibration procedure, and the updated strain field is computed after every finite element simulation, from the extracted numerical displacement field.

2.3 *Optimisation*

The optimisation procedure controls the FEMU method and is performed for different algorithms. The optimisation procedure is implemented in Python programming language, using the optimisation algorithms from the SciPy library [12]. This library has several optimisation algorithms (e.g., gradient-based, stochastic), providing the user with easy implementation in its programs. Three optimisation algorithms of different types are selected for the optimisation procedure: Levenberg-Marquardt, Nelder-Mead and Differential Evolution algorithms.

The Levenberg-Marquardt (LM) is a gradient-based algorithm that uses the approximated Hessian and Jacobian matrices [13]. It is largely used in the calibration of constitutive models [14–16] because it is well suited for solving nonlinear least-squares problems. The LM algorithm requires the user to select an initial solution, and if not well chosen, it can lead to convergence difficulties or to finding local minima, instead of the global minimum. Nonetheless, if the problem is well-conditioned and a suitable initial solution is selected, the algorithm can rapidly converge.

The Nelder-Mead (NM) algorithm is one of the best known and simpler direct-search algorithms used in unconstrained optimisation [17]. The NM algorithm uses a simplex that begins with a set of points for every optimisation variable plus one, considered its vertices. Based on a series of transformations of the simplex, the algorithm iteratively reduces the simplex size. The algorithm is known to achieve satisfactory results in few iterations, but also presents problems of convergence. Comparatively to gradient-based algorithms, this algorithm stands out for not requiring the use of derivatives.

The Differential Evolution (DE) is a population-based stochastic algorithm that generates new solutions from the combination of other solutions [18]. The DE algorithm can be characterised by its ease of implementation, robustness, and finding the global minimum of a problem in most attempts. However, a significant drawback of DE is its high computational cost and low convergence rate compared to other optimisation algorithms. On the other hand, when compared to other population-based stochastic algorithms, DE tends to overperform them. As is common in population-based algorithms, the number of solutions used as the population can significantly impact the convergence and success of the optimisation procedure [19]. Moreover, the DE algorithm requires the definition of variables bounds to generate the initial set of solutions, either manually, randomly or distributed over the search space.

The LM and NM algorithms are suitable for unconstrained problems instead of the DE algorithm, requiring variable bounds. To limit the search space for the three algorithms, lower and upper bounds are defined for all variables (see Table 2). The bounds are arbitrarily defined based on the order of magnitude for each variable and are not very restricted. On the DE algorithm, the bounds are directly imposed, while for both LM and NM algorithms a variables transformation is used. Note that for all the algorithms, the optimisation variables are normalised by its bounds. Considering $\hat{x}_i = x_i/x_0$ the variable x_i normalised relatively to its initial value x_0, the variables bounds are normalised as

$$\hat{x}_i^{\min} = \frac{x_i^{\min}}{x_0} \quad \text{and} \quad \hat{x}_i^{\max} = \frac{x_i^{\max}}{x_0}. \tag{3}$$

Then, the variable transformation for $\hat{X}_i \geq 1$ corresponds to

$$\hat{x}_i = 1 + (\hat{x}_i^{\max} - 1)\left(1 - e^{\frac{1-\hat{X}_i}{\hat{x}_i^{\max}-1}}\right), \tag{4}$$

and for $\hat{X}_i < 1$ corresponds to

$$\hat{x}_i = 1 + (\hat{x}_i^{\min} - 1)\left(1 - e^{\frac{1-\hat{X}_i}{\hat{x}_i^{\min}-1}}\right). \tag{5}$$

3 Results

The material parameters A, B, n, m, and C are defined as the optimisation variables to identify. The melting temperature T_{m}, the transition temperature T_{tr}, and the parameter $\dot{\varepsilon}_0$ are considered known a priori and kept fixed throughout the optimisation procedure. Mainly, these parameters are kept fixed because the first two have specific physical meaning, and the third may increase the problem of non-uniqueness of the solution [20].

To mimic image noise from actual full-field measurements, random noise from a normal Gaussian distribution is added to the reference data's displacement field, while the load signals are kept without noise. To evaluate the robustness of the methodology and performance of algorithms, three data sets are used as reference data for the calibration: (i) without noise; (ii) with random noise of amplitude 10^{-5} mm; and (iii) with random noise of amplitude 10^{-3} mm. The objective function values using the variables reference set are 0.0, 3.130×10^{-9}, and 3.017×10^{-5}, respectively, for each data set.

Because the LM and NM algorithms are sensitive to the initial solution, five different initial sets are generated using the Latin hyperspace sampling method, generating solutions evenly distributed over the search space (see Table 2). This sampling method does not consider the reference set, avoiding an initial bias towards the global minimum. The distribution of the variables of each generated initial set over the search space is shown in Fig 4. These five initial sets are used for the LM and NM algorithms, while for the DE algorithm, a population of 50 solutions is generated using the same sampling method.

The three algorithms are, in general, used with the default settings defined in the SciPy library. The exceptions are the step size for the finite difference approximation of the Jacobian in the LM algorithm, set equal to 10^{-3}, and the adaptive setting in the NM algorithm, which adapts the algorithm parameters to the problem's dimensionality.

Table 2. Upper and lower bound of optimisation variables and initial sets for the Levenberg-Marquardt (LM) and Nelder-Mead (NM) algorithms.

	A [MPa]	B [MPa]	n [-]	m [-]	C [-]
Bound					
Lower	50.000	100.000	0.010	0.500	0.000
Upper	1200.000	2000.000	0.500	4.000	1.000
Set					
1	838.230	258.620	0.340	3.970	0.210
2	1158.140	1704.400	0.060	0.680	0.920
3	707.870	525.550	0.180	2.950	0.700
4	51.600	1368.290	0.440	2.110	0.010
5	397.470	1218.310	0.300	1.400	0.460

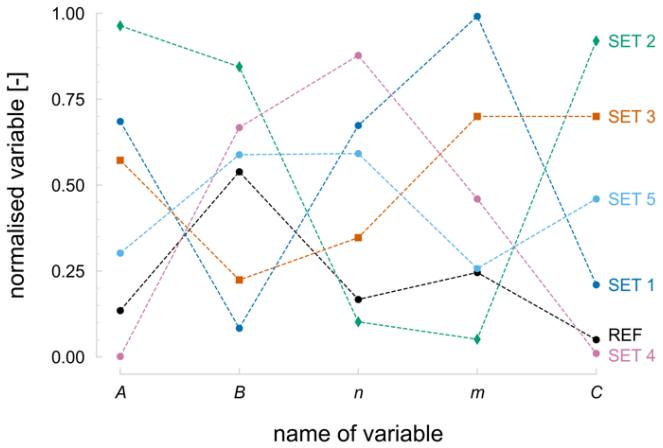

Fig. 4. Distribution in the search space of the optimisation variables, normalised by its bounds, in each initial set for the Levenberg-Marquardt (LM) and Nelder-Mead (NM) algorithms. The reference set is also represented.

The final solutions and objective function values obtained in the optimisation procedures are summarised in Table 3, for the three algorithms. Additionally, in Fig. 5, the evolution of the objective function throughout the function evaluations is represented for the three algorithms.

Considering that for final solutions within 0.1% of error from the reference solution the algorithms achieve the global minimum, it is observed that the LM algorithm presents the worst performance of the three algorithms. Overall, the LM achieves the global minimum only twice, whereas the NM algorithm achieves the global minimum six times, and the DE algorithm can find the global minimum in two out of three attempts. The fact that LM cannot achieve the global minimum more often can be related to the step size for the finite difference approximation of the Jacobian, which is probably not small enough. The algorithm would perhaps benefit from starting a new optimisation procedure, using the final solution as the optimisation's initial solution and reducing the step size.

The results of LM and NM algorithms confirm their sensitivity to the initial set, with results varying depending on the initial set. This is particularly evident when the variable A of the initial set is close to its upper bound, as is the case of set 2. In this situation, the solution stagnates very early, either in the upper and lower bound of variable A. Avoiding initial solutions close to the variables bounds can potentially decrease the chance of the algorithm converging there. In general, variables n, m, and C present higher rates of success in achieving the reference solution. However, set 5 with the LM algorithm shows local minima, which are very susceptible to this algorithm.

The effect of noise in the results is quite interesting, as it is observed that noise can benefit the algorithm. In the case of the LM and NM algorithms, the final solutions achieved are, in general, very similar between the data sets without noise and with random noise of amplitude 10^{-5} mm. However, in the LM with initial set 3, it is observed that the presence of noise benefits the algorithm in achieving a better solution than without noise. When the level of noise is increased to an amplitude of 10^{-3} mm, both the LM and NM algorithms are not as efficient in achieving the reference solution, but the NM algorithm appears to be less negatively affected by the presence of additional noise. The DE algorithm appears to be less affected by noise regarding the final solution, as each variable solution of data set with random noise of amplitude 10^{-3} mm is within 0.1% of error from the reference solution. However, the final solution for the data set with random noise of amplitude 10^{-5} mm is not close to the reference one, but it can be related to the presence of a local minimum. To further investigate the effect of noise in the DE algorithm, more attempts should be performed, as the DE algorithm is not deterministic, and a component of randomness always exists. The objective function convergence is also affected by the presence of noise. For example, for the LM algorithm with initial set 1, where the final solutions are relatively similar, the algorithm converges around 1000, 600, and 50 function evaluations, respectively, for data sets without noise, with random noise of amplitude 10^{-5} and 10^{-3} mm.

In summary, the differences between the LM and NM algorithms are more evident in terms of convergence rate, where the first tends to converge rapidly, whereas the former typically requires a higher number of function evaluations. A single attempt was performed for the DE algorithm with each data set, and it appears to be more robust than the other two algorithms. Nonetheless, it generally requires a high number of function evaluations to converge to the global optimum. Considering the use of multiple initial sets for the LM and NM algorithms, it can be considered a good strategy in substitute to the DE algorithm.

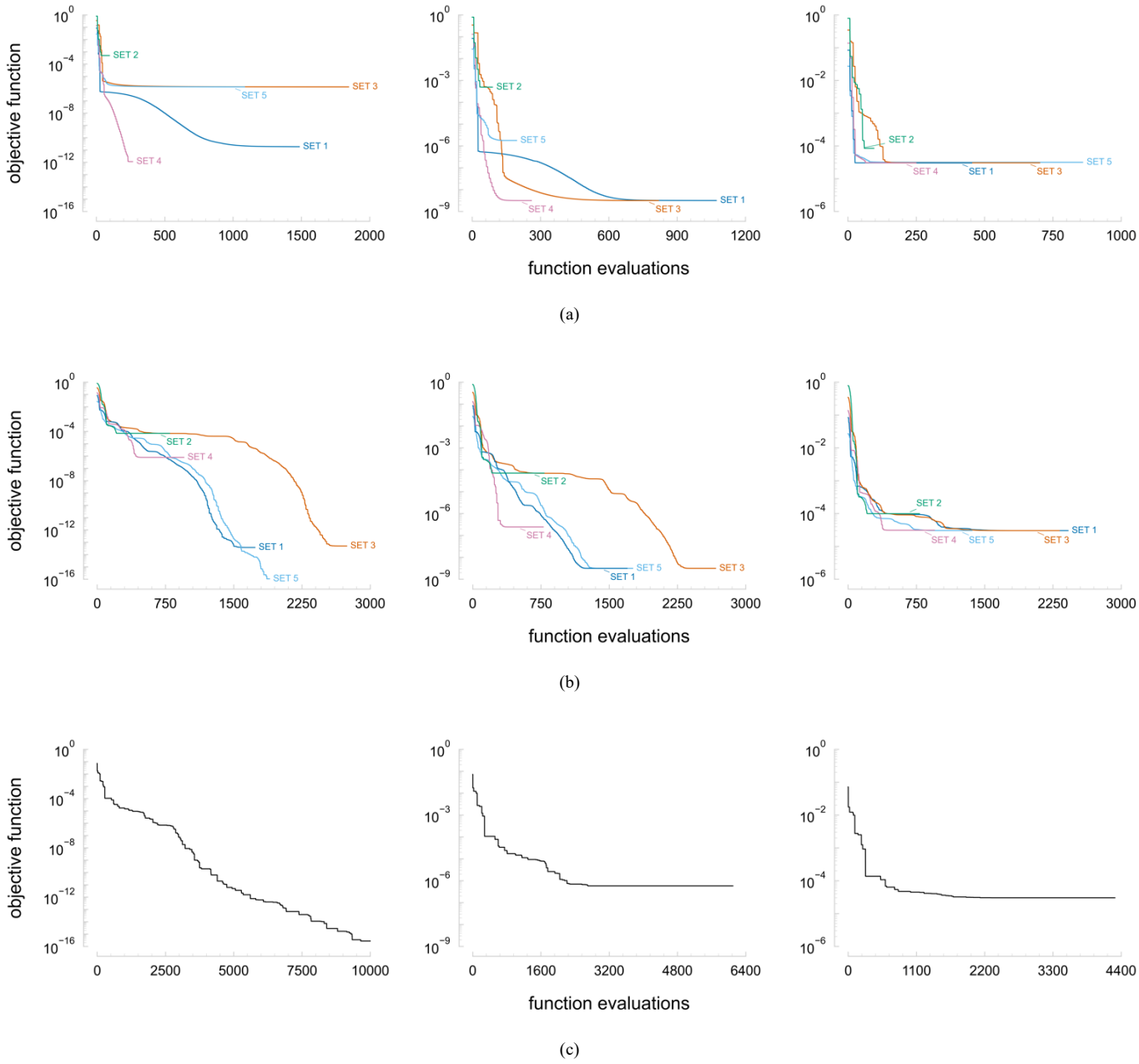

Fig. 5. Evolution of objective function throughout function evaluations using the (a) Levenberg-Marquardt (LM), (b) Nelder-Mead (NM), and (c) Differential Evolution (DE). Results correspond to data sets without noise (left), with random noise of amplitudes 10^{-5} (centre) and 10^{-3} (right).

4 Conclusion

A methodology to calibrate thermoelastoviscoplastic constitutive models based on full-field measurements and the FEMU is considered to reduce the number of thermomechanical tests involved. Three heterogeneous thermomechanical tests performed at different average strain rates are used as reference data required by the calibration procedure. By taking advantage of a modern programming language library, three optimisation algorithms are easily implemented in the calibration process, namely the Levenberg-Marquardt (LM), Nelder-Mead (NM), and Differential Evolution (DE) algorithms. The algorithm's results are compared for three data sets, involving different noise levels introduced in the displacement field of the reference data. The DE algorithm demonstrates to be the most robust algorithm by reaching or being very close to the global minimum in two out of the three data sets. However, it is also susceptible to local minima even though less than the LM and NM algorithms. Moreover, the number of

Table 3. Final solutions and objective function values obtained in the optimisation procedure using the Levenberg-Marquardt (LM), Nelder-Mead (NM) and Differential Evolution (DE) algorithms. Results correspond to data sets without noise, with random noise of amplitude 10^{-5}, and 10^{-3} mm. The smaller font size values represent the relative error (%) of each variable's final value to the reference one. The values in bold indicate that the error to is smaller than 0.1%.

	Set	A [MPa]	B [MPa]	n [-]	m [-]	C [-]	$\varphi(\boldsymbol{\chi})$
LM	without noise						
	1	205.779 0.277	**1123.565 −0.039**	**0.092 0.000**	**1.360 0.000**	**0.050 0.000**	1.860×10^{-11}
	2	1200.000 484.797	2000.000 77.936	0.500 443.478	0.705 −48.162	0.055 10.000	4.900×10^{-4}
	3	269.893 31.520	1147.581 2.098	0.100 8.696	1.235 −9.191	0.051 2.000	1.371×10^{-6}
	4	**205.228 0.009**	**1123.986 −0.001**	**0.092 0.000**	**1.360 0.000**	**0.050 0.000**	1.134×10^{-12}
	5	356.294 73.624	1029.529 −8.405	0.113 22.826	1.292 −5.000	0.051 2.000	1.371×10^{-6}
	with random noise of amplitude 10^{-5} mm						
	1	205.860 0.317	**1123.511 −0.044**	**0.092 0.000**	**1.360 0.000**	**0.050 0.000**	3.153×10^{-9}
	2	1200.000 484.797	2000.000 77.936	0.500 443.478	0.705 −48.162	0.053 6.000	4.900×10^{-4}
	3	**205.202 −0.004**	**1123.965 −0.003**	**0.092 0.000**	**1.360 0.000**	**0.050 0.000**	3.134×10^{-9}
	4	205.530 0.156	**1123.770 −0.020**	**0.092 0.000**	**1.360 0.000**	**0.050 0.000**	3.139×10^{-9}
	5	375.103 82.790	1020.996 −9.164	0.116 26.087	1.280 −5.882	0.051 2.000	1.778×10^{-6}
	with random noise of amplitude 10^{-3} mm						
	1	210.709 2.680	1120.125 −0.345	0.093 1.087	1.358 −0.147	**0.050 0.000**	3.017×10^{-5}
	2	50.000 −75.635	2000.000 77.936	0.082 −10.870	0.729 −46.397	0.053 6.000	8.433×10^{-5}
	3	280.799 36.835	1074.003 −4.448	0.102 10.870	1.328 −2.353	**0.050 0.000**	3.046×10^{-5}
	4	200.753 −2.172	**1127.190 0.284**	0.091 −1.087	1.362 0.147	**0.050 0.000**	3.017×10^{-5}
	5	357.248 74.089	1029.400 −8.416	0.113 22.826	1.291 −5.074	0.051 2.000	3.155×10^{-5}
NM	without noise						
	1	**205.211 0.000**	**1123.999 0.000**	**0.092 0.000**	**1.360 0.000**	**0.050 0.000**	3.781×10^{-14}
	2	1200.000 484.767	1749.794 55.676	0.201 118.478	0.501 −63.162	0.057 14.000	6.991×10^{-5}
	3	**205.211 0.000**	**1123.998 0.000**	**0.092 0.000**	**1.360 0.000**	**0.050 0.000**	4.990×10^{-14}
	4	50.000 −75.635	1242.019 10.500	0.076 −17.391	1.409 3.603	0.049 −2.000	7.780×10^{-7}
	5	**205.210 0.000**	**1124.000 0.000**	**0.092 0.000**	**1.360 0.000**	**0.050 0.000**	1.069×10^{-16}
	with random noise of amplitude 10^{-5} mm						
	1	**205.218 0.004**	**1123.991 −0.001**	**0.092 0.000**	**1.360 0.000**	**0.050 0.000**	3.131×10^{-9}
	2	1200.000 484.767	1749.755 55.572	0.201 118.478	0.501 −63.162	0.057 14.000	6.991×10^{-5}
	3	**205.210 0.000**	**1124.000 0.000**	**0.092 0.000**	**1.360 0.000**	**0.050 0.000**	3.130×10^{-9}
	4	124.777 −39.195	1183.018 5.251	0.083 −9.783	1.388 2.059	**0.050 0.000**	2.440×10^{-7}
	5	**205.206 −0.002**	**1124.002 0.000**	**0.092 0.000**	**1.360 0.000**	**0.050 0.000**	3.131×10^{-9}
	with random noise of amplitude 10^{-3} mm						
	1	206.206 0.485	**1123.279 −0.064**	**0.092 0.000**	**1.360 0.000**	**0.050 0.000**	3.017×10^{-5}
	2	1200.000 484.767	1751.275 55.807	0.200 117.391	0.500 −63.235	0.057 14.000	9.888×10^{-5}
	3	206.326 0.544	**1123.261 −0.066**	**0.092 0.000**	**1.360 0.000**	**0.050 0.000**	3.017×10^{-5}
	4	50.000 −75.635	1241.590 10.462	0.076 −17.391	1.410 3.676	0.049 −2.000	3.094×10^{-5}
	5	257.939 25.695	1082.608 −3.683	0.099 7.609	1.348 −0.882	**0.050 0.000**	3.032×10^{-5}
DE	without noise						
		205.210 0.000	**1124.000 0.000**	**0.092 0.000**	**1.360 0.000**	**0.050 0.000**	1.690×10^{-16}
	with random noise of amplitude 10^{-5} mm						
		308.679 50.421	1043.777 −7.137	0.106 15.217	1.336 −1.765	0.051 2.000	5.824×10^{-7}
	with random noise of amplitude 10^{-3} mm						
		205.152 −0.028	**1123.997 0.000**	**0.092 0.000**	**1.360 0.000**	**0.050 0.000**	3.017×10^{-5}

function evaluations required for convergence by the DE algorithm is higher than the others. A strategy using multiple initial sets could be used to circumvent the sensitivity of LM and NM algorithms to initial sets. In the scope of the present work, a calibration software was developed in Python programming language, which will allow the easy implementation and combination of optimisation algorithms.

Acknowledgements

This project has received funding from the Research Fund for Coal and Steel under grant agreement No 888153. The authors also gratefully acknowledge the financial support of the Portuguese Foundation for Science and Technology (FCT) under the projects PTDC/EME-APL/29713/2017, PTDC/EME-EME/31243/2017 and PTDC/EME-EME/30592/2017 by UE/FEDER through the programs CENTRO 2020 and COMPETE 2020, and UID/EMS/00481/2013-FCT under CENTRO-01-0145-FEDER-022083. The authors also would like to acknowledge the Région Bretagne for its financial support and BPI France under project EXPRESSo. M.G. Oliveira and J.M.P. Martins are, respectively, grateful to the FCT for the PhD grants SFRH/BD/143665/2019 and SFRH/BD/117432/2016.

References

[1] Karbasian, H., Tekkaya, A.E. A review on hot stamping. J Mater Process Tech 210, 2010, 2103–2118. doi: 10.1016/j.jmatprotec.2010.07.019.
[2] Markiewicz, É., Langrand, B., Notta-Cuvier, D. A review of characterisation and parameters identification of materials constitutive and damage models: From normalised direct approach to most advanced inverse problem resolution. Int J Impact Eng 110, 2017, 371–381. doi: 10.1016/j.ijimpeng.2017.01.028.
[3] Kajberg, J., Wikman, B. Viscoplastic parameter estimation by high strain-rate experiments and inverse modelling – Speckle measurements and high-speed photography. Int J Solids Struct 44, 2007, 145–164. doi: 10.1016/j.ijsolstr.2006.04.018.
[4] Martins, J.M.P., Andrade-Campos, A., Thuillier, S. Comparison of inverse identification strategies for constitutive mechanical models using full-field measurements. Int J Mech Sci 145, 2018, 330–345. doi: 10.1016/j.ijmecsci.2018.07.013.
[5] Cooreman, S., Lecompte, D., Sol, H., Vantomme, J., Debruyne, D. Elasto-plastic material parameter identification by inverse methods: Calculation of the sensitivity matrix. Int J Solids Struct 44, 2007, 4329–4341. doi: 10.1016/j.ijsolstr.2006.11.024.
[6] de-Carvalho, R., Valente, R.A.F., Andrade-Campos, A. Optimization strategies for non-linear material parameters identification in metal forming problems. Comput Struct 89, 2011, 246–255. doi: 10.1016/j.compstruc.2010.10.002.
[7] Kowalewski, Ł., Gajewski, M. Assessment of optimization methods used to determine plasticity parameters based on DIC and back calculation methods. Exp Techniques 43, 2019, 385–396. doi: 10.1007/s40799-018-00298-5.
[8] Martins, J.M.P., Andrade-Campos, A., Thuillier, S. Calibration of Johnson-Cook model using heterogeneous thermo-mechanical tests. Procedia Manuf 47, 2020, 881–888. doi: 10.1016/j.promfg.2020.04.274.
[9] Coër, J., Bernard, C., Laurent, H., Andrade-Campos, A., Thuillier, S. The effect of temperature on anisotropy properties of an aluminium alloy. Exp Mech 51, 2011, 1185–1195. doi: 10.1007/s11340-010-9415-6.
[10] Dassault Systèmes. Abaqus 2017 Documentation. 2017.
[11] Johnson, G.R., Cook, W.H. A constitutive model and data for metals subjected to large strains, high strain rates, and high temperatures, in: Proceedings 7th International Symposium on Ballistics. 1983. pp. 541–547.
[12] Virtanen, P., et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods 17, 2020, 261–272. doi: 10.1038/s41592-019-0686-2.
[13] Moré, J.J. The Levenberg-Marquardt algorithm: Implementation and theory. Lect Notes Math 105–116, 1977. doi: 10.1007/bfb0067700.
[14] Güner, A., Soyarslan, C., Brosius, A., Tekkaya, A.E. Characterization of anisotropy of sheet metals employing inhomogeneous strain fields for Yld2000-2D yield function. Int J Solids Struct 49, 2012, 3517–3527. doi: 10.1016/j.ijsolstr.2012.05.001.
[15] Marek, A., Davis, F.M., Rossi, M., Pierron, F. Extension of the sensitivity-based virtual fields to large deformation anisotropic plasticity. Int J Mater Form 12, 2019, 457–476. doi: 10.1007/s12289-018-1428-1.
[16] Martins, J.M.P., Andrade-Campos, A., Thuillier, S. Calibration of anisotropic plasticity models using a biaxial test and the virtual fields method. Int J Solids Struct 172–173, 2019, 21–37. doi: 10.1016/j.ijsolstr.2019.05.019.
[17] Gao, F., Han, L. Implementing the Nelder-Mead simplex algorithm with adaptive parameters. Comput Optim Appl 51, 2012, 259–277. doi: 10.1007/s10589-010-9329-3.
[18] Storn, R., Price, K. Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11, 1997, 341–359. doi: 10.1023/a:1008202821328.
[19] Piotrowski, A.P. Review of differential evolution population size. Swarm Evol Comput 32, 2017, 1–24. doi: 10.1016/j.swevo.2016.05.003.
[20] Notta-Cuvier, D., Langrand, B., Markiewicz, E., Lauro, F., Portemont, G. Identification of Johnson–Cook's viscoplastic model parameters using the virtual fields method: Application to titanium alloy Ti6Al4V. Strain 49, 2013, 22–45. doi: 10.1111/str.12010.