Journal of Achievements in Materials and Manufacturing Engineering

VOLUME 55
ISSUE 2
December
2012

Determination of rheological properties using hybrid optimisation method

B. Foder, K. Sołek*
Faculty of Metals Engineering and Industrial Computer Science, AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Kraków, Poland
* Corresponding e-mail address: ksolek@metal.agh.edu.pl

Received 10.10.2012; published in revised form 01.12.2012

Analysis and modelling

ABSTRACT

Purpose: of this paper is description of special software developed for identification of parameters values of rheological models used by ProCAST package for simulation of state of stress in the solidifying castings.
Design/methodology/approach: The PyTensionTest uses a hybrid optimization algorithm. This algorithm includes the stochastic method which searches the starting point for direct search optimization method. The identification of parameters values is based on measurement results obtained within the confines of the material tests carried out using the testing machine.
Findings: This paper presents the results of approximation of stress-strain relationship for two grades of steel using the elasto-plastic rheological model with power law hardening where its parameters are linearly dependent on temperature.
Research limitations/implications: This paper presents the special software that allows user to define any rheological model.
Practical implications: Application of obtained values of rheological parameters in the ProCAST software allows to determine stress and strain distribution in continuous castings. It can help to optimize important process parameters.
Originality/value: Developed software can approximate the results of static tension tests using rheological model depended on material temperature. This software makes possible an approximation of several stress-strain curves simultaneously. Application of the hybrid optimization method allows to obtain very precise approximation due to determination of the global minimum of the goal function.
Keywords: Rheological properties; Optimization methods; Computer science

Reference to this paper should be given in the following way:
B. Foder, K. Sołek, Determination of rheological properties using hybrid optimisation method, Journal of Achievements in Materials and Manufacturing Engineering 55/2 (2012) 477-482.

1. Introduction

The main goal of this work was a development of the software for the handling of data obtained during material tests. Such tests are carried out for analysis of rheological properties. The results of the data handling in the form of rheological model parameters can be directly used in numerical simulations. This work was performed in order to simulate the stress and strain distribution in continuous castings. Numerical simulations allow generally to determine proper parameters of the continuous casting processes. The knowledge of the state of stress in the continuous casting can avoid risk of bleedout in the mould area during the casting process. The measurements of the rheological properties in this work were in the form of the tension tests carried out at high temperatures, related to the temperatures of the castings below the mold, where the possibility of the bleedout is the biggest. Developed software, called the PyTensionTest, allows to approximate several curves simultaneously with very good accuracy, thanks to application of the hybrid optimization method. Such possibilities are not often available in another applications.

2. Approximation procedure

2.1. Estimation of rheological model parameters

Each mathematical model that approximates results of rheological tests can be generalised by the following equation:

$$d = F(x, \mathbf{p}, \mathbf{s}), \quad R^1 \rightarrow R^1 , \tag{1}$$

where:
d - output parameter that usually describes stress σ or viscosity,
x - measured input parameter i.e. strain ε or shear rate,
$\mathbf{p} = \{p_1,\ldots,p_k\}$ - vector of rheological model coefficients, where k is a number of model coefficients,
$\mathbf{s} = \{s_1,\ldots,s_k\}$ - vector of process parameters such as temperature or internal variables, where l is a number of considered parameters.

The elasto-plastic rheological model with power law hardening was used for description of relationship between stress and strain. The mathematical equation of this model could be expressed in the following way:

$$\sigma = \sigma_\infty + (\sigma_0 - \sigma_\infty)\mathrm{e}^{-\alpha \cdot \varepsilon^{pl}} , \tag{2}$$

where:
σ_0- yield stress,
σ_∞- ultimate yield stress,
α- strengthening exponent,
$\varepsilon^{\mathrm{pl}}$- plastic strain.

The model (2) is a specific case, where any process parameters are not considered. This equation was modified in order to take an influence of material temperature into consideration (3a). Parameters σ_0, σ_∞ and α can be, for example, linearly dependent on temperature:

$$\sigma(T) = \sigma_\infty(T) + (\sigma_0(T) - \sigma_\infty(T)) \cdot \mathrm{e}^{-\alpha(T) \cdot \varepsilon^{pl}} \tag{3a}$$

$$\sigma_0(T) = a_{\sigma_0} T + b_{\sigma_0} \tag{3b}$$

$$\sigma_\infty(T) = a_{\sigma_\infty} T + b_{\sigma_\infty} \tag{3c}$$

$$\alpha(T) = a_\alpha T + b_\alpha \tag{3d}$$

where:
T - temperature,
$a_{\sigma_0}, b_{\sigma_0}, a_{\sigma_\infty}, b_{\sigma_\infty}, a_\alpha, b_\alpha$ - rheological parameters.

Estimation of the rheological model (2, 3b-3d) parameters (components of vector **p**) can be considered as a problem of the non-linear programming. It is based on the minimization of the goal function written in the following form:

$$\Phi(\mathbf{p}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left| d^c(x_{i,j}, \mathbf{p}, \mathbf{s}) - d^e_{i,j} \right| \tag{4}$$

where:
x_{ij} - input parameter j measured in the test i,
d^e_{ij} - output parameters j measured in the test i,
m - number of experiments carried out for the same material at different temperatures,
n - number of measurements of the load in the single experiment.

Minimization of the goal function (4) can be performed by any non-linear optimization method. It describes the sum of distances between measured values of stress during experiments and computed using equation (3).

2.2. Particle swarm optimization method

The particle swarm optimization method allows to search extremes of multidimensional continuous non-linear functions. This method has a lot of common elements with some evolution techniques, such as the genetic algorithm - GA or the evolutionary programming - EP. It was developed in 1995 by Kennedy and Eberhart [1].

Each step of this method results in determination of the position and velocity of each particle. It gives an information about the best reached position in all swarm. From the swarm one particle, with the best position, is selected as a leader. Start positions and velocities of the particles are assigned randomly.

Actualization of the components of the velocity vector for each particle in subsequent iterations is determined using information about previous state of this particle and state of all society. Determination of the velocities and positions of the particles is carried out with the help of following equations, according to [2]:

$$\mathbf{v}_{t,n} = \omega \cdot \mathbf{v}_{t-1,n} + c_1\eta_1\left(\mathbf{x}^b_n - \mathbf{x}_{t-\mathbf{1},n}\right) + c_2\eta_2\left(\mathbf{x}^b_{leader} - \mathbf{x}_{t-\mathbf{1},n}\right) \tag{5}$$

$$\mathbf{x}_{t,n} = \mathbf{x}_{t-1,n} + \mathbf{v}_{t,n} \tag{6}$$

where:
$\mathbf{v}_{t,n} = \{v_{t,n,1},\ldots,v_{t,n,\mathrm{k}}\}$ - k-dimensional velocity vector for part n in iteration t,
$\mathbf{x}_{t,n} = \{x_{t,n,1},\ldots,x_{t,n,\mathrm{k}}\}$ - k-dimensional position vector for particle n in iteration t,
ω - inertia weight coefficient that keep particle at current trajectory,
c_1, c_2- acceleration constants that represents influence of the experience of the particle and all society,
η_1, η_2- are random values that represents unpredictable movements of each part,
$\mathbf{x}^b_n = \{x^b_{n,1},\ldots, x^b_{n,\mathrm{k}}\}$ - the best position of the particle n during optimisation procedure,
leader - index of the leader, the particle which represent the most optimal value of the goal function.

In the particle swarm optimization method each particle can move either in limited space or in unlimited space with unrestricted conditions [3]. Stop criteria can be defined in many ways, based on improvement of results or movement and distribution of the particles.

2.3. Direct search method

The Hook-Jeeves method is non-gradient optimization method presented in [4]. It is an iterative method. Each iteration consists of sequence of *the exploratory moves* in all direction of multidimensional space. If an improvement (the lower value of the goal function) in selected direction is reached *the pattern move* is executed. The pattern move gives new approximation of the problem solution. Afterwards the new iteration is started.

The goal of the exploratory moves is to learn about values of the goal function Φ in surroundings of current base point $\mathbf{x}_i$ (i - iteration index). For each dimension the movement gives the points $\mathbf{x}_{i,j}$ (j - dimension index) according to expression (5). The step in each dimension *step* is assumed as an input parameter. If no improvement was reached within the confines of the exploratory moves in selected dimension, the increment is changed in the opposite direction.

$$\mathbf{x}_{i,j} = \mathbf{x}_{i,j-1} + step \cdot \mathbf{s}_j \quad \text{where} \quad \mathbf{x}_{i,0} = \mathbf{x}_i \tag{7}$$

where:
$\mathbf{s}_j$ - the versor in the j direction

If the exploratory moves are a success ($\mathbf{x}_{i,k} < \mathbf{x}_i$), the pattern move is executed and new base point $\mathbf{x}_{i+1}$ is determined according to expression (6). If $\mathbf{x}_{i+1} \geq \mathbf{x}_i$, then *step* parameter is reduced and the exploratory moves is executed again. Objective of pattern move is to accelerate searches:

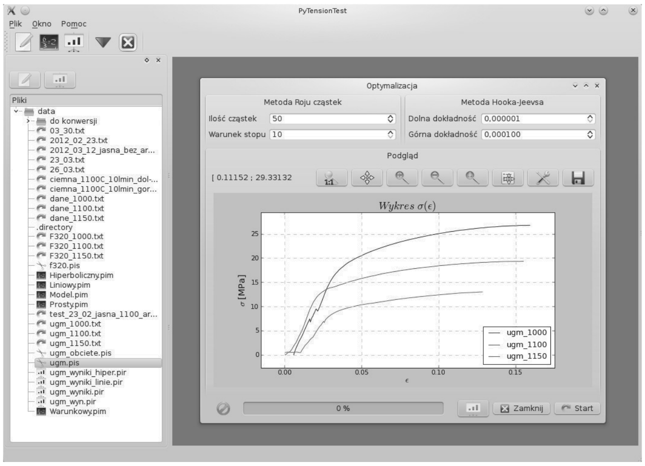
$$\mathbf{x}_{i+1} = \mathbf{x}_i + 2 \cdot \left(\mathbf{x}_{i,k} - \mathbf{x}_i\right) \tag{8}$$

Searching is continued in next iterations based on sequences of successive exploratory and pattern moves [5]. One of disadvantages of this method is ineffectiveness in the case of functions with many extremes. Local extreme can be reached when the step size is too small or the coordinates of the starting point is wrongly chosen.

The most simple approach based on the same value of *step* parameter for each dimension is ineffective when the sensitivity of the goal function value is different in different dimension. In order to avoid such problem the value of *step* parameter is optimized in each searching dimension.

3. Description of PyTensionTest software

The main goal of this work was development of the special software for an analysis of results of material tests. The *PyTensionTest* application is dedicated to estimate parameters of the rheological models, that approximates results of the tension tests. This application will be precisely described in [6]. The main window of this software is shown in Figure 1.



Metoda Roju cząstek

Ilość cząstek: 50

Warunek stopu: 10

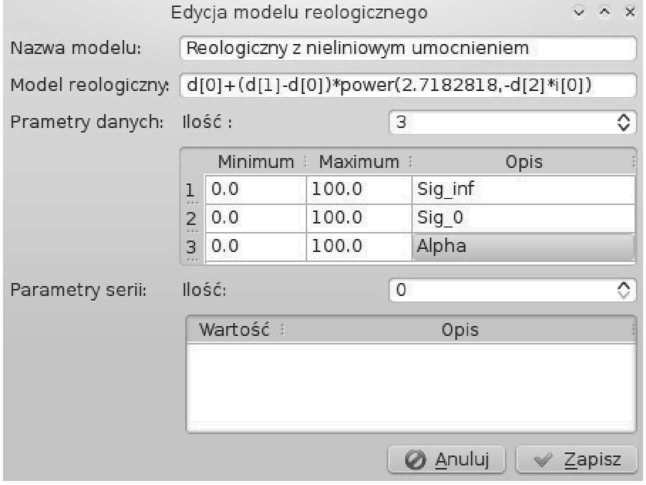
Metoda Hooka-Jeevsa

Dolna dokładność: 0,000001

Górna dokładność: 0,000100

Fig. 1. Main window of PyTensionTest software with optimization sub window

The PyTensionTest application was created in Python script language, usually used to solve the engineering and scientific problems [7]. The application is equipped with the graphical user interface (Figure 1). It can work using different operating system, just like MS Windows, Linux and Mac OS. The main advantage of this application is possibility of definition by users their own rheological models. The models defined by users are treated like normal syntax of the Python programming language, so there is no difference in execution of the software. The window which makes such definition possible is shown in Figure 2.



Nazwa modelu: Reologiczny z nieliniowym umocnieniem

Model reologiczny: d[0]+(d[1]-d[0])*power(2.7182818,-d[2]*i[0])

Prametry danych: Ilość : 3

	Minimum	Maximum	Opis
1	0.0	100.0	Sig_inf
2	0.0	100.0	Sig_0
3	0.0	100.0	Alpha

Parametry serii: Ilość: 0

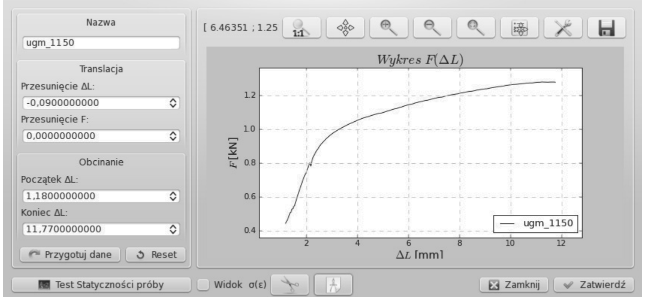
Wartość	Opis

Fig. 2. Window for definition of rheological models describing of material properties

In the window shown in Figure 2 user can define the mathematical equations. In order to insert such equations user uses the syntax of Python programming language. If necessary it is also possible to use the basic arithmetical operations and a lot of functions from the *numpy*, numeric library developed in C language. For easy identification by users and readable presentation of results the model coefficients can be freely called.

Users' mathematical model, located in the source code of the software, is compiled to executable files before optimization, what allows to avoid a decrease of optimization speed.

The application can be used for approximation of both single stress-strain curve and several curves. If the estimation is performed simultaneously for several tests users need to define the temperatures of samples during these tests and also the relationship of rheological parameters and temperature within the rheological model.



Nazwa

ugm_1150

Translacja

Przesunięcie ΔL: -0,0900000000

Przesunięcie F: 0,0000000000

Obcinanie

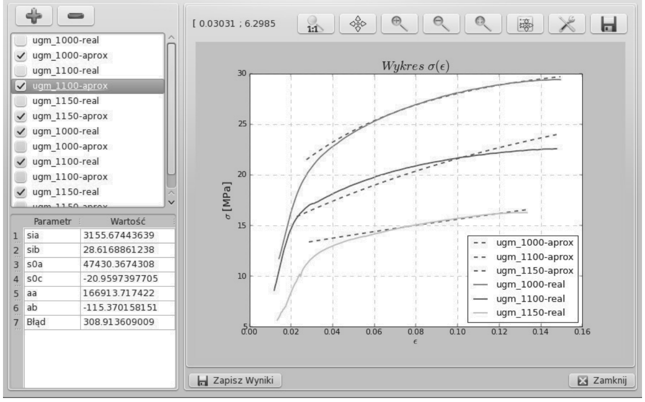
Początek ΔL: 1,1800000000

Koniec ΔL: 11,7700000000

Fig. 3. Window for calibration of the data obtained from the tension tests

The data handling using the PyTensionTest application consists of three steps. The first step is devoted to reading the data from ASCII files with the results created by the measurement system of the testing machine. If necessary, user can calculate the values of stress as a function of strain *σ*(*ε*) on the basis of load-elongation relationship *F*(*ΔL*) and geometric parameters of the samples. The raw data sometimes requires running the offset procedure mainly due to starting the tests with non-zero value of stroke or force. The application allows user both to translate the curves towards stress and strain axis and to limit the range of them to receive good quality data for minimization (window shown in Figure 3).

The second step is devoted to analysis of the material tests conditions. Among other things the PyTensionTest software determines values of the rate of the stress increase and the strain rate during whole tension test (see Figure 4). It could be especially important for the checking if the experiments met the static tension tests requirements according to the standard.



☐ ugm_1000-real
☑ ugm_1000-aprox
☐ ugm_1100-real
☑ ugm_1100-aprox
☐ ugm_1150-real
☑ ugm_1150-aprox
☑ ugm_1000-real
☐ ugm_1000-aprox
☑ ugm_1100-real
☐ ugm_1100-aprox
☑ ugm_1150-real

	Parametr	Wartość
1	sia	3155.67443639
2	sib	28.6168861238
3	s0a	47430.3674308
4	s0c	-20.9597397705
5	aa	166913.717422
6	ab	-115.370158151
7	Błąd	308.913609009

Fig. 4. Window with the graph of the tension process parameters

The approximation procedure is executed in the third step of analysis. The application contains the non-linear optimization solver, that uses the hybrid optimization algorithm, combined the particle swarm optimization with the Hooke-Jeeves search method. In the PSO method particles move within the initial solution space defined by user. In this method user can also define a number of particles which take part in the minimization process. This optimisation stage is to find the starting point for second algorithm in the surroundings of the global minimum. The stop criteria in the PSO method is based on the solution improvement and defined as a number of iterations without progress. The PSO method is paralleled for better performance. The goal of the second stage is exact determination of the global minimum of the goal function (4). During optimization process user can see the progress of approximation in the graphical form.

After optimization process user can see the values of each searched parameter. It is possible to see the graph with stress curves measured in experiments and calculated using estimated parameters (see Figure 5). Such comparison informs about correctness of the performed approximation. User can both edit visual style of graph and save it as image file. The application gives user also some information about optimization process such as optimization time, number of calculations of the objective function value and error of approximation.

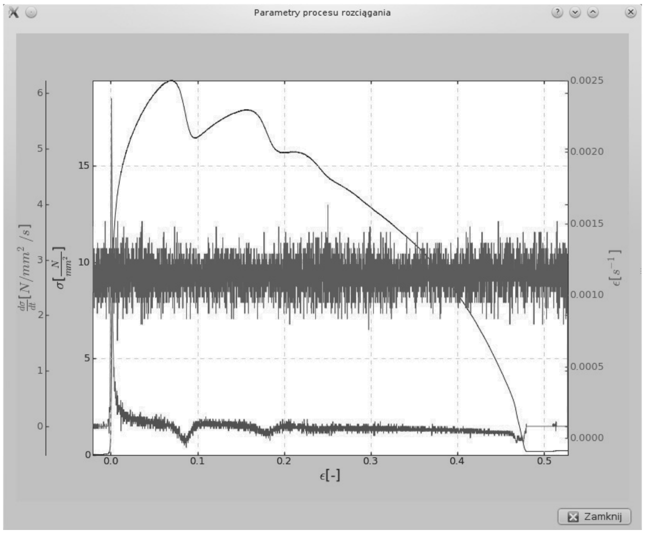


Fig. 5. Window of the PyTensionTest software with results obtained within the confines of the optimization

4. Application of PyTensionTest software

The PyTensionTest software was used for an analysis of results of the tension tests carried out for two different grades of as cast steels: F320 and UMG. The chemical compositions of this steels were presented in Table 1.

The graphs of stress as a function of strain received from tensile tests are presented in Figure 6. The measurements for each steel alloy were carried out at three different temperatures: 1000, 1100 and 1150°C.

Table 1.
The chemical compositions of F320 and UMG steel

Steel	Chemical Composition, %				
	C	Mn	Si	P	S
F320	0.01	0.74	0.02	0.013	0.01
UMG	0.05	0.21	0.007	0.011	0.009
	Cu	Cr	Ni	Mo	
F320	0.02	0.02	0.07	0.007	
UMG	0.03	0.008	0.007	<0.005	

Curves in Figure 6 required preparation. It included translation and limitation of the range of them. Data prepared for minimization process are shown in Figure 7.

The Figure 8 shows the comparison of measured and calculated curves using the values of parameters (see Table 2) estimated for each curve separately (model (2)) and the values of parameters estimated for all curves together for each steel (model (3)). Approximation of each curve separately gives better results. It is an information that linear relationship between models parameters and temperature should by changed into non-linear one.

Table 2.
Parameters values for rheological models (2) and (3) for tested steels

Steel/model	T, K	E, MPa	σ_0 , MPa	σ_∞ , MPa	α
UMG/(3)	1273	806.88	14.68	30.43	18.58
	1373	860.68	12.37	26.24	9.24
	1423	428.14	11.2	24.15	4.57
UMG/(2)	1273		12.8	30.2	21.17
	1373		11.55	23.23	19.55
	1423		8.64	17.28	16.82
F320/(3)	1273	1331.38	18.59	38.36	23.87
	1373	696.95	12.25	25.56	15.85
	1423	796.13	8.05	18	11.13
F320/(2)	1273		24.71	39.15	16.76
	1373		13	24.04	16.16
	1423		8.62	18.75	13.46

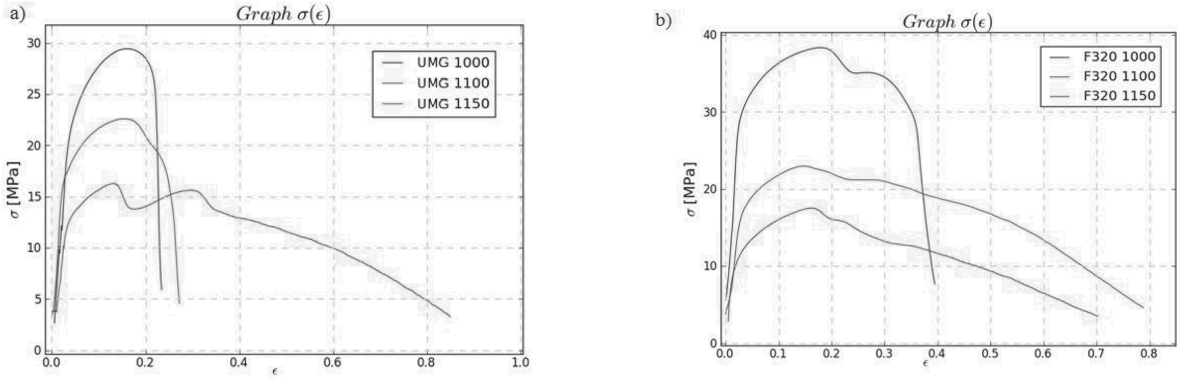


Fig. 6. Graphs of stress - strain curves received from the static tensile tests performed at temperature of 1000, 1100 and1150°C: (a) UMG steel, (b) F320 steel

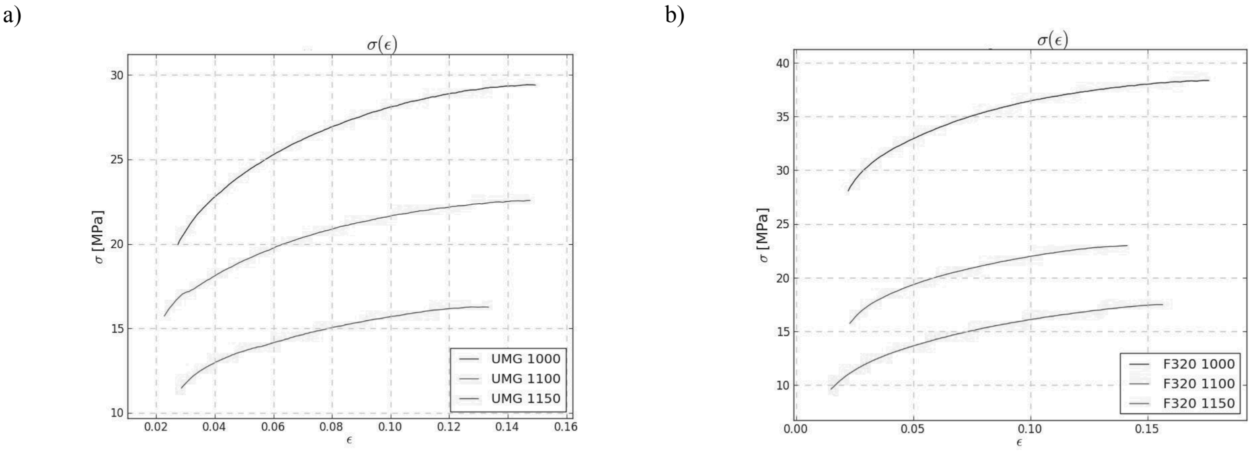


Fig. 7. Graphs of stress - strain curves prepared for identification of rheological parameters: (a) UMG steel, (b) F320 steel

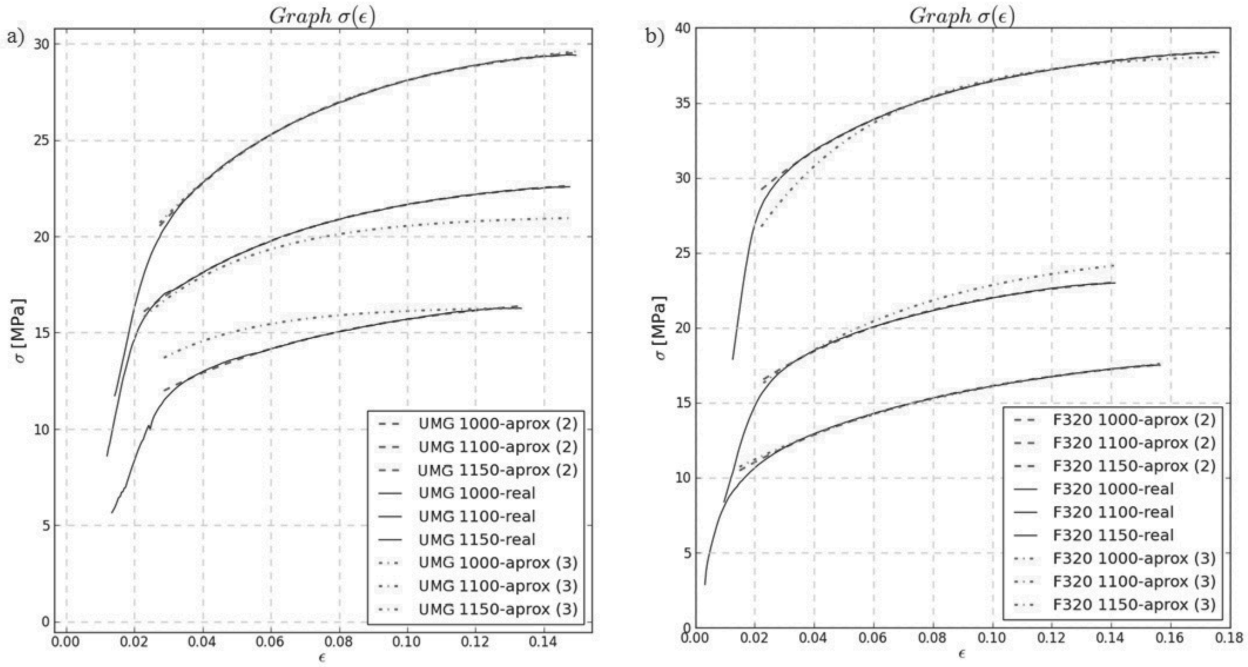


Fig. 8. Graphs of stress-strain curves measured and calculated using rheological models (2) and (3) for steels: a) UMG, b) F320

5. Conclusions

The software developed in this work is an useful tool for the handling of data obtained within the confines of tension tests. It provides a few rare possibilities in comparison with other applications. Among other things user can easy prepare measured data for approximation procedure. The second important advantage of this software is the hybrid optimization method, that gives the possibility of search of the global minimum in the case of non-unimodal goal function. The hybrid method applied in this software is a combination of the stochastic and direct search method. The third important advantage is easy and quick possibility of definition of non-standard rheological models. It allows to test different equations. The PyTensionTest software is also user friendly.

Acknowledgements

Research work was supported by the European Regional Development Fund - the Operational Programme - Innovative Economy, as a project “New Concept for Selection of the Cooling Parameters in Continuous Casting of Steel", POIG.01.03.01-12-009/09.

References

[1] J. Kennedy, R. Eberhart, Particle swarm optimization, Proceedings of the IEEE International Conference on Neutral Networks, Piscataway, 1995, 1942-1948.

[2] Y. Shi, R. Eberhart, A modified particle swarm optimizer, Proceedings of the IEEE Conference on Evolutionary Computation, Anchorage, 1998, 69-73.

[3] S. Xu, Y. Rahmat-Samii, Boundary conditions in particle swarm optimization revisited, IEEE Transactions on Antennas and Propagation 55 (2007) 760-765.

[4] R. Hooke, T.A. Jeeves, “Direct Search“ solution of numerical and statistical problems, Journal of Association for Computing Machinery 8 (1961) 212-229.

[5] L. L. Lay, T. F. Chan, Appendix B: The Method of Hooke and Jeeves, Distributed Generation: Induction and Permanent Magnet Generators, John Wiley and Sons, Ltd, Chichester, 2007.

[6] B. Foder, Application for results analysis of static tension tests and identification of rheological parameters Master’s Thesis, AGH University of Science and Technology, Cracow, 2012 (in Polish),

[7] M. Lutz, Python. Introduction, Third Edition, Helion, Gliwice, 2009 (in Polish).