SoftwareX 10 (2019) 100249

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

MaterialModeler—From experimental raw data to a material model

T. Benkert *, C. Hartmann, M. Eder, F. Speckmaier, W. Volk

Chair of Metal Forming and Casting, Department of Mechanical Engineering, Technical University of Munich, Walther-Meisser-Strasse 4, 85748 Garching, Germany

ARTICLE INFO

Article history:
Received 23 January 2019
Received in revised form 6 May 2019
Accepted 6 May 2019

Keywords:
Material model
Material characterization
Metal forming

ABSTRACT

To perform the numerical analysis of metal forming processes, material models are needed. These models are based on experiments such as the tensile test. To generate a material model from experimental results, the test data must be analyzed, possibly smoothed, fitted by mathematical approaches and exported in the format of the simulation software. Currently this is achieved using proprietary software, self-programmed code or spreadsheets. The software MaterialModeler is aimed at closing the gap between the data generated in experiments and simulation software to improve the quality of material models, especially in respect of complex to model materials. Simultaneously, the error rate compared to the current manual process is decreased and every step from the experimental raw data to the material model is reproducible. MaterialModeler has been developed in the field of sheet metal forming, but it should easily be possible to extend its use to other disciplines.

Code metadata

Current code version	0.41
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2019_13
Legal Code License	CC BY-SA 4.0
Code versioning system used	git
Software code languages, tools, and services used	Matlab
Compilation requirements, operating environments & dependencies	Matlab
If available Link to developer documentation/manual	https://gitlab.lrz.de/ne84gam/materialmodeler/tree/master/Documentation https://gitlab.lrz.de/ne84gam/materialmodeler/wikis/home
Support email for questions	tim.benkert@utg.de

Software metadata

Current software version	0.41
Permanent link to executables of this version	https://gitlab.lrz.de/ne84gam/materialmodeler/blob/master/MaterialModeler.exe
Legal Software License	CC BY-SA 4.0
Computing platforms/Operating Systems	Windows, Linux, OS X
Installation requirements & dependencies	Matlab Compiler Runtime, will be installed automatically during setup
If available, link to user manual — if formally published include a reference to the publication in the reference list	
Support email for questions	tim.benkert@utg.de

1. Motivation and significance

Today, the numerical analysis of physical processes is almost entirely based on continuum mechanics, which uses balance equations (mass, linear momentum, energy) for modeling. The formulation of boundary value problems leads to partial differential equations. To solve the proposed boundary value problems, numerical methods, such as the finite element method (FEM) [1], are used. In general, the balance equations provide five equations for 15 unknown field variables. Consequently, ten more equations are needed, which are the so called constitutive equations describing the characteristics of the materials. The constitutive equations, also called material models, are

* Corresponding author.
E-mail address: tim.benkert@utg.de (T. Benkert).

https://doi.org/10.1016/j.softx.2019.100249

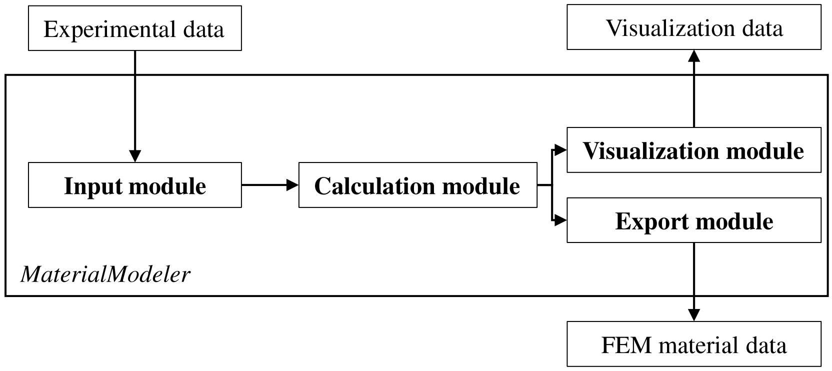

Fig. 1. Global architecture of MaterialModeler.

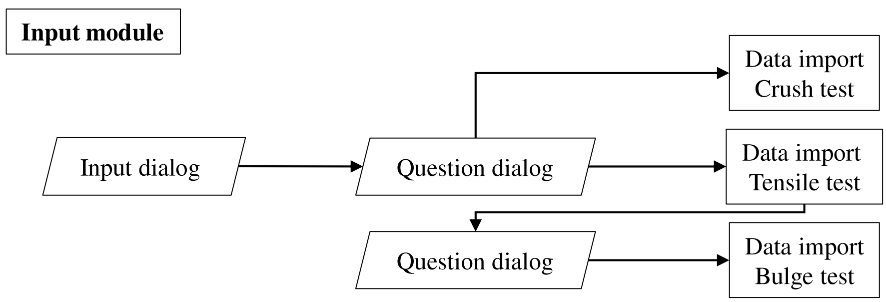

Fig. 2. Architecture of MaterialModeler input module.

parameter-based descriptions of the material behavior. One common way of identifying the material parameters of a given model, is to execute controlled experiments under known conditions. The model parameters are derived from the experimental data and prepared for the transfer to the numerical solver, which is either a manual task or performed using proprietary software. Reproducing a material model is scarcely possible, because the exact settings used to generate the model from the experimental data are usually not published. MaterialModeler aims to overcome these drawbacks by providing a modular open source platform. Its modular structure will allow its functionality to be extended far beyond its current status. Currently, MaterialModeler generates material models for one finite element (FE) program, which includes data handling and treating. This is usually the most time-consuming task in the whole process.

2. Software description

MaterialModeler is able to handle multiple input data from tensile tests, bulge tests and crush tests, which are the most widely used experiments for material characterization in metal forming. Additional tests can easily be added. Originating from a given set of test data, material parameters for different implemented models can be derived and exported for direct application, e.g. in ABAQUS [2]. MaterialModeler's modular structure allows it to be easily extended with new material models or additional export functionality.

2.1. Software architecture

MaterialModeler contains four main modules: the input module, calculation module, export module, visualization module. Fig. 1 shows a schematic diagram of the overall architecture. The input module reads the raw experimental data and asks for the material name and density in the first input dialog, see Fig. 2. The user must decide whether tensile and/or bulge data is provided or crush experiments should be evaluated.

The import module hands the import information over to the calculation module (Fig. 3), where all computations for the determination of the material model are done. Initially, an internal data structure is created from the provided raw experimental data to ensure consistent data handling. The data averaging module makes it possible to edit the experimental data and unifies the redundant experimental files resulting from multiple experiments. The averaged data is used for all further calculations. Before computing material model parameters, preprocessing calculations for the respective tests must be performed. The first calculated material parameter is the Young's modulus (E-Modulus). Since the E-Modulus acts as an input parameter in the following steps, the user may also define a value. Finally, the other parameters for the selected models are calculated.

The visualization and export modules prepare the computed data for documentation and output purposes. The visualization is interactive and all results can be reviewed before the export. If adaptions are necessary, the user can go back to the calculation module for recomputation.

2.2. Software functionalities

MaterialModeler offers commonly used material models in sheet metal forming analysis. The material models can be split into elastic and plastic models. Hooke's law [3] is applied in most cases when describing the elastic domain of metals. The only modeling parameters are the E-Modulus and the Poisson's ratio. The derivation of E-modulus using the regression method presented in [4] is implemented in MaterialModeler. For Poisson's ratio, default values from literature are given for standard material classes (steel, aluminum). Both parameters can also be defined by the user.

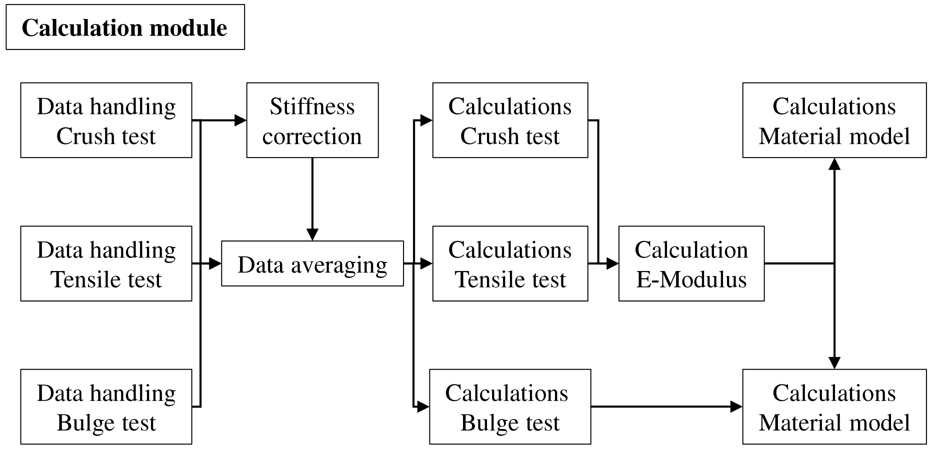

Fig. 3. Architecture of MaterialModeler calculation module.

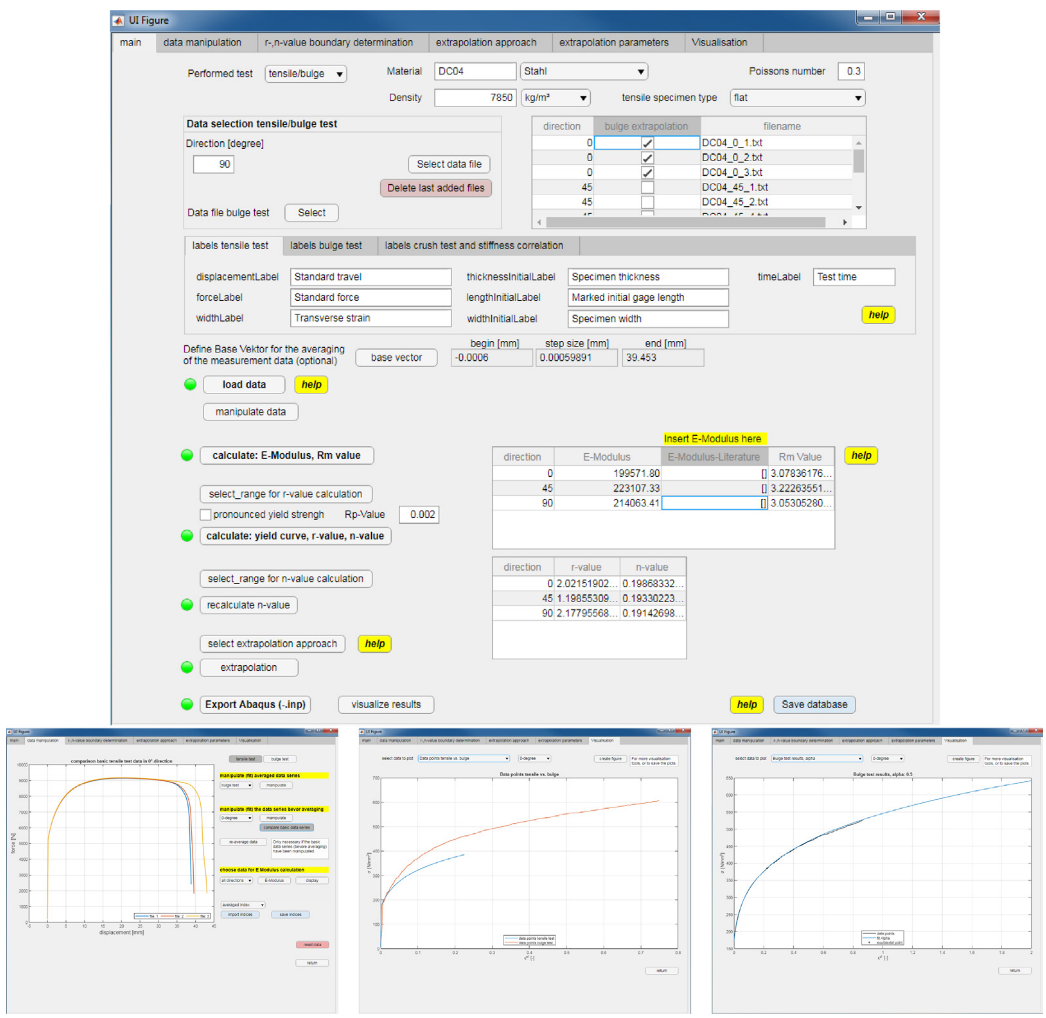

Fig. 4. Overview of MaterialModeler's GUI (top); unprocessed experimental raw data (left); processed experimental data from tensile and bulge tests (middle); final material model (right).

In the plastic domain, the yield curve is normally used to describe the hardening behavior of materials. To deduce yield curve values beyond the limits of the experimental data, extrapolation based on suitable models is possible. In MaterialModeler, the approaches of Ludwik [5], Ghosh [6], Hockett–Sherby [7], Swift [8] and Voce [9] are implemented. In addition to the single model use, weighted summation of two models is possible by introducing a scaling parameter. For the tensile test, the extrapolation can be enhanced by additional incorporation of bulge data, which is adequately transformed by the equivalent deformation power method [10]. Furthermore, n-values can be derived [11]. MaterialModeler can calculate plastic strain ratios [12] if test data from tensile tests in different rolling directions are provided.

The final results from MaterialModeler calculations can be exported for direct use in ABAQUS without further processing steps. in its final form, the material model contains the material name, density, E-Modulus, extrapolated yield curve and plastic strain ratios.

3. Illustrative examples

Fig. 4 shows the GUI of MaterialModeler.

The user works from top to bottom, completing the process by exporting the generated material model. The other tabs are used for data editing, extrapolation and visualization and may be used as necessary. At the beginning of the main workflow, the performed test must be chosen and material, density, specimen geometry and Poisson's number provided. Subsequently, the data files obtained from the test equipment are selected and read. Prior to calculation of the E-Modulus, the user may view and manipulate the data. The final steps prior to export are calculation of yield curve, plastic strain ratios and n-values, the latter being useful in comparing the material data to specifications in standards. In Fig. 4, three screenshots are shown below the main part of the GUI. The data manipulation view is depicted on the left. The view of the averaged experimental results is plotted in the center. The final modeled flow curve and the underlying experimental data are visualized on the right.

4. Impact

As MaterialModeler is open source, it will be easy to compare its results with those of proprietary software. The differences occurring can be analyzed, e.g. with regard to their impact on numerical results. This will trigger new research that on the one hand enhances the quality of material models and on the other hand investigates which experimental data is actually necessary to achieve sufficient simulation results. As there is currently no free software available, the step from experimental data to the final material model is not covered in current papers. Therefore, it is to some extent impossible to compare current papers. In future, it will be possible to provide the settings used in MaterialModeler to make this step transparent to the scientific community. In cases, where experimental data is evaluated several times, MaterialModeler generates reproducible and comparable results. Since the introduction of MaterialModeler at the author's institute, the generation of material models from experimental data has taken a lot less time. The software's GUI guides the user through the necessary steps and provides suitable standard values, which reduces the error rate, especially for inexperienced users, such as students. Furthermore, the visualizations provide a quick and easy way to review results and subsequently change settings if needed.

So far, MaterialModeler has focused on metal forming. Due to its modular structure, its extension to other research fields such as civil engineering (modeling concrete or sand) is easily possible.

5. Conclusions

MaterialModeler provides the maximum possible transparency for the conversion of experimental raw data from standardized tests into material models. Each step of the process is traceable in the code and the extrapolation approaches used can be checked and discussed by the scientific community. The authors are convinced that by letting other scientists evaluate, criticize and contribute to this work, the scientific community will be able to develop and distribute consistent material models more quickly than is currently possible.

Acknowledgments

The authors would like to thank their numerous students who, while they did not contribute directly to this software, made it possible through their work, for the authors to have available sufficient free time to work on this project. Furthermore the authors would like to thank the German Research Foundation (DFG) for the financial support given under grant numbers VO 1487/27 and VO 1487/28.

Appendix A. Supplementary data and video

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.softx.2019.100249.

References

[1] Zienkiewicz Cecil Olgierd, Taylor Robert L, Zhu JZ. The finite element method set. 6th ed. Oxford: Elsevier Butterworth-Heinemann; 2005.
[2] Simulia, Abaqus Extended Portfolio. Version 2018 Providence, Rhode Island, USA: Dassault Systèmes; 2018. Online availabe at https://www.3ds.com/products-services/simulia/products/abaqus, [Last Accessed 05 November 2018].
[3] Hooke R. Lectures de Potentia restitutiva, or of Spring explaining the power of springing bodies. London: 1678.
[4] Vitzthum S, Eder M, Hartmann C, Volk W. Investigation on strain dependent elastic behavior for accurate springback analysis. Journal of Physics: Conference Series 2018;1063. http://dx.doi.org/10.1088/1742-6596/1063/1/012118.
[5] Ludwik P. Elemente Der Technologischen Mechanik. Berlin: Springer; ISBN: 978-3-662-39265-2, 1909.
[6] Ghosh AK. Tensile instability and necking in materials with strain hardening and strain-rate hardening. Acta Metall 1977;12:1413–24. http://dx.doi.org/10.1016/0001-6160(77)90072-4.
[7] Hockett JE, Sherby OD. Large strain deformation of polycrystalline metals at low homologous temperatures. J Mech Phys Solids 1975;23:87–98. http://dx.doi.org/10.1016/0022-5096(75)90018-6.
[8] Swift HW. Plastic instability under plane stress. J Mech Phys Solids 1952;1:1–18. http://dx.doi.org/10.1016/0022-5096(52)90002-1.
[9] Voce E. The relation between the stress and strain for homogeneous deformation. J. Inst. Metals 1948;74:537–62.
[10] DIN EN ISO 16808 .Metallic materials – Sheet and strip – Determination of biaxial stress–strain curve by means of bulge test with optical measuring systems. 2014.
[11] DIN EN ISO 10275 Metallic materials – Sheet and strip – Determination of tensile strain hardening exponent. 2014.
[12] DIN EN ISO 10113 Metallic materials – Sheet and strip – Determination of plastic strain ratio. 2017.