FlexMM

Groen, Manso; Solhjoo, Soheil; Voncken, Ruud; Post, Jan; Vakis, Antonis

Published in:
Advances in Engineering Software

DOI:
10.1016/j.advengsoft.2020.102876

Publication date:
2020

Citation for published version (APA):
Groen, M., Solhjoo, S., Voncken, R., Post, J., & Vakis, A. (2020). FlexMM: A standard method for material descriptions in FEM. *Advances in Engineering Software*, *148*, [102876]. https://doi.org/10.1016/j.advengsoft.2020.102876

Advances in Engineering Software 148 (2020) 102876

Contents lists available at ScienceDirect

Advances in Engineering Software

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

FlexMM: A standard method for material descriptions in FEM

Manso Groen[a,1], Soheil Solhjoo[b,1,*], Ruud Voncken[a], Jan Post[a,c], Antonis. I. Vakis[b]

[a] Philips, the Netherlands

[b] Computational Mechanical and Materials Engineering, Engineering and Technology Institute Groningen, Faculty of Science and Engineering, University of Groningen, Nijenborgh 4, 9747 AG Groningen, the Netherlands

[c] Advanced Production Engineering, Engineering and Technology Institute Groningen, Faculty of Science and Engineering, University of Groningen, Nijenborgh 4, 9747 AG Groningen, the Netherlands.

ARTICLE INFO

Keywords:
FEM analysis
Digital twin
Material modeling
Multi stage modeling
User subroutines
Zero-defect manufacturing
Industry 4.0

ABSTRACT

This article discusses a number of key issues concerning simulation-based digital twins in the domain of multistage processes. Almost all production processes are multistage in nature, and so most digital twins involve multiple physical phenomena, process steps and different solvers for the simulations. Good interoperability between model solvers and processes are key to achieving a functional digital twin. Passing information between steps can be challenging, complex and time consuming, especially for material data, because the constitutive model interacts with the full modeling environment: material behavior is interdependent with the history of the process, the solver subroutines and the boundary conditions. This work proposes a flexible yet robust standardization approach, called FlexMM, for dealing with material data, constitutive models, measurement data or mathematical models to overcome part of the abovementioned complexity. The implementation of FlexMM consists of a general rule structure in which constitutive behavior is described, as well as its interaction with the subroutines used by the finite element solver. The definition of the constitutive model is stored in a separate file, in which the material behavior can be described in a user selected format, such as look-up tables, standard statistical models, machine learning or analytical expressions. After a calculation step, the new local material properties are mapped to a file to facilitate the next history-dependent step. In this way, the interaction between the different fabrication steps and processes can be incorporated. A material/process case study is presented to demonstrate the flexibility and robustness of FlexMM.

1. Introduction

Industry 4.0, the so-called fourth industrial revolution, is based on the digital transformation of industry, and includes cyber-physical systems, the internet of things or the internet of services, and cloud computing [1,2]. In this transformation, digital replicas of physical assets –referred to as digital twins– play an important role. Digital twins are digital representatives of the real world and can be used to monitor and control their physical counterparts. Therefore, one can study the behavior of the digital twins instead of the physical systems, resulting in cost savings, decreasing development time and increasing manufacturing flexibility toward zero-defect manufacturing. This means that topics like quality control, manufacturing flexibility, predictive maintenance and virtual and augmented reality are important issues to be implemented in future digital production platforms. Fig. 1 presents the position of the digital twin in the concept of zero-defect manufacturing.

Generating and working with digital twins, where FEM plays a central role, presents a unique set of challenges. Summarizing the scope in which they are applied, certain requirements have to be met, requiring digital twins to be:

- dedicated to the process,
- highly robust,
- multi-stage,
- multi-platform,
- multi-process,
- multi-vendor.

Clearly, the interoperability of digital twins is a key element of their operation. Interoperability issues have been known for a long time, and different researchers have worked on possible solutions in the context of multiphysics FEM. The main approach is to develop mapping

[*] Corresponding author.
E-mail addresses: s.solhjoo@rug.nl, soheilsolhjoo@gmail.com (S. Solhjoo).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.advengsoft.2020.102876
Received 22 January 2020; Received in revised form 15 May 2020; Accepted 24 June 2020

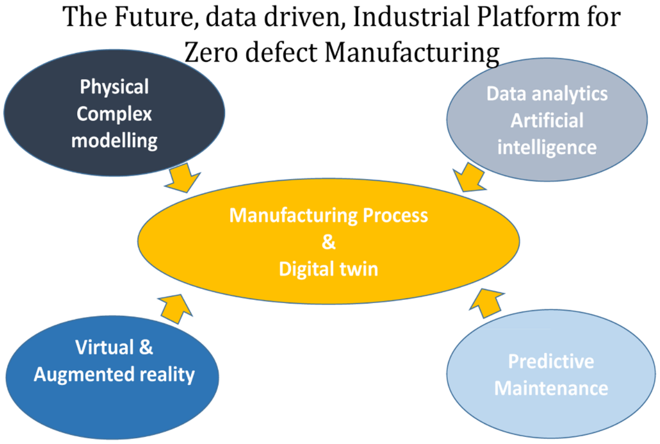

The Future, data driven, Industrial Platform for Zero defect Manufacturing
Physical Complex modelling
Data analytics Artificial intelligence
Manufacturing Process & Digital twin
Virtual & Augmented reality
Predictive Maintenance

Fig. 1. The position of the digital twin in zero-defect manufacturing.

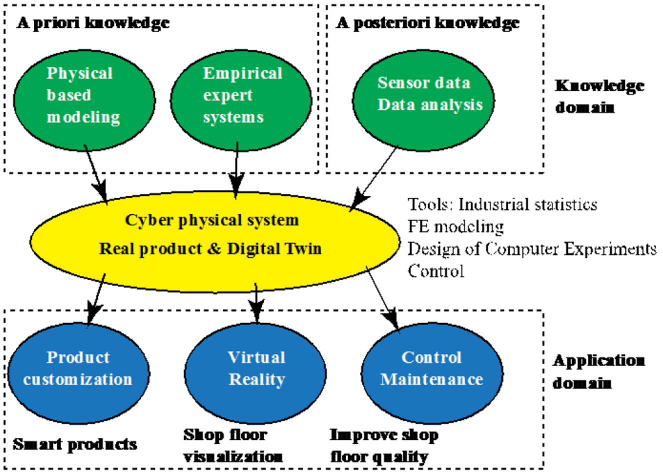


Fig. 3. The position of the digital twin in zero-defect manufacturing.

strategies with which the FE data can be transferred between different mesh types [3–7] or between different simulations for fluid dynamics and structural simulations, either as separate environments, such as MpCCI [8,9], DIGIMAT [10], etc. [11], or as integrated parts of the commercial software packages [12–14]. Fig. 2 shows a schematic of the mapping methods, which allow the data transfer of the FE simulation results from one solver to another. The mapper (see Fig. 2) structures the data. This is important in order to map, for instance, the stress calculated with solver A to the stress used by solver B.

While the FE data can be transferred via either of these mapping strategies, the description of the material(s) behavior is limited to the available models used by the FE solvers. For example, in the work of Young [15], an effort was made to automate the interaction between the material model and the user subroutines of ABAQUS. If another solver has to be used, the work has to be performed again. Hence, the material-solver interaction has to be solved specifically for every case. If the desired material models are not available in all involved solvers, the user has to write specific subroutines to be implemented in the solvers. This task can be exceptionally demanding if, for example: (1) each of the solvers use different names for calling one particular variable, (2) the solvers are written in different coding languages, or (3) the bookkeeping of the data stored in the user defined material data and subroutines requires continuous inspection. Moreover, aside from the names assigned to each particular variable, the definition of a variable in each solver might be slightly different, which can result in errors and wrong estimations.

In this paper, a new standardized method for describing material behavior is proposed, which can serve as a unified method, independent from solvers and interfaces. This Flexible Material Model approach, to be called *FlexMM*, can be used for describing any type of material behavior. In this manner, FlexMM can be used across multiple platforms as the unified material model, while the solving process switches between different solvers. Another advantage of the method is that it can be presented with a user-friendly GUI regardless of its possible behind-the-scenes complex and huge database. The remainder of the paper is organized as follows: firstly, take a look in the background of physical product- and process-based digital twins. Then, we describe the interoperability problem and FlexMM as its remedy. Next, we present the mathematical and computer implementations of the approach. Finally, we discuss the application and potentials of FlexMM through a FEM simulation of a sintering process as a case study, followed by the conclusion section.

2. Background on physical product- and process-based digital twins

A digital twin must be able to predict relevant behaviors of a product or process (or a subset of the process) during its life cycle. The digital twin can be a part of Product Lifecycle Management (PLM) and can be based on different types of data, as shown in Fig. 3.

A digital twin of a process can be used in different ways on a production platform, see e.g. [16,17]. In general, digital twins can serve as a part of:

- the development of new processes and products,
- an industrial control system,
- the predictive maintenance or predictive quality,
- virtual and augmented visualization systems, etc.

The digital twins mentioned above can be established on real production and experimental data or a model of the process, e.g. using the finite element method (FEM). Moreover, it is possible to use real-time production data to update, compare and improve the digital twins.

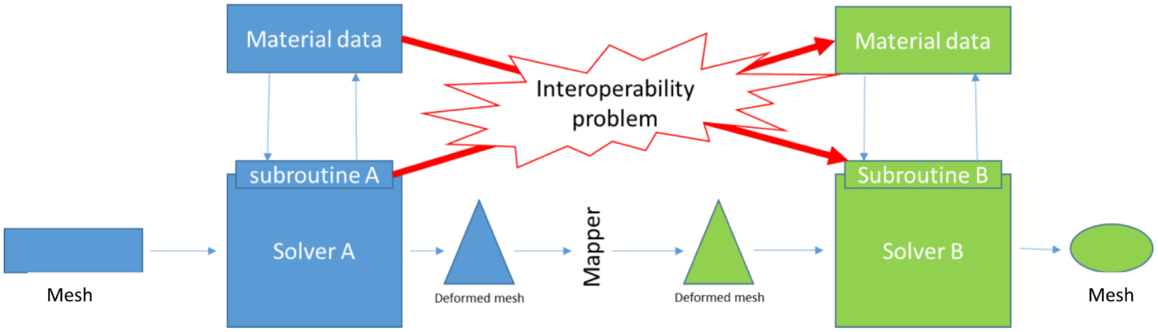


Fig. 2. Overview of the current way of passing information from one calculation to another.

Considering the fact that the digital twins can be generated via different methods, each of them has its own advantages and disadvantages. For example, in most cases of metal forming processes, FE models cannot be used directly as a part of the control system, due to their long calculation times. To solve this problem, a meta-model can be created based on physical or phenomenological models and using statistical techniques, such as DACE (Design and Analyses of Computer Experiments) [18–23]. Furthermore, other data mining techniques are available, leading to an accurate and fast representative model of complex FE models; so, an FE model can indirectly be the core of a digital twin. The following list gives a number of examples of different meta-modeling techniques for generating digital twins of metal forming processes:

- Artificially intelligent models based on statistical analyses of real-time production data, such as different machine learning techniques.
- Simple data processing techniques, such as linear or quadratic regression, or advanced techniques, such as Fourier transformations or signal processing in combination with statistics.
- Adaptive analytical models, which can be recalibrated using real-time production data.
- Analytical models, whether physical or phenomenological, calibrated based on other demanding models, such as complex micro-mechanical nonlinear FE models.
- Combination of different model concepts, e.g. combining physical models and real-time production data.

In the above cases, it is essential that the models capture the relevant behavior of the physical systems under the relevant operating conditions.

Apart from the quality of the model predictions as discussed before, there are also requirements for the usability of the digital twins on the factory floor. If you look at the application of digital twins in product development, process engineering and quality control, some trends can be seen:

- Near net shaping: Increasingly more process steps will be required to produce parts within narrower tolerances. An example of this trend is ship propellers cast from brass. The casting process is a relatively inexpensive; however, the cast propeller subsequently requires a costly milling process to achieve a smooth surface finish. If the casting can be performed to more accurate specifications, the expensive milling process can be avoided.
- Flexibility: A trend for higher quality and smaller features at lower production costs can also be seen. This is combined with decreasing batch sizes. As an example of this trend, consider the variety of car models in car manufacturing, many with lower production volumes. Production lines have to become more flexible to be able to produce different models in one production line.
- Production yield: The approach to increase robustness and predictive quality by process improvement strategies like design for six sigma (DFSS) yields improved results, but is labor intensive because large sampling batches are required for accuracy.

Improvements are needed for tackling the above challenges. One possible improvement is the application of adaptive process control [22] needed in situations where uncontrolled fluctuations occur. Adaptive process control adjusts the process based on real-time measurements to cope with these uncertainties and has been applied successfully in many industries. The main bottlenecks to achieve adaptive process control in the field of metal processing are the following: availability of process control algorithms for complete manufacturing chains (>20 steps), missing in-depth process knowledge and the lack of in-line measurements at the required accuracy and speed [23].

To use complex models real-time in a process line, the output from the models has to be created fast enough to match the pace of the process. There must also be reliable sensors to create feed-forward and backward responses to control the process. One of the challenges is to develop a simulation platform to run the simulations on, in such a way that it creates usable results in the solution space. Another challenge is to create meta-models based on the available information that can be used as an input for the real-time control system. When FE calculations are used as a basis for meta-models, large amounts of calculations have to be performed to create the required accuracy. When performing large amounts of calculations, the building and execution of FE models has to be automated. When performing calculations by an automated script, the risk of failing calculations increases, because there is no interaction with the engineer for final checks [24]. There are many reasons for FE calculations not running to completion or giving erroneous results: typical complexities that result in uncompleted jobs are contact conditions, material subroutines that produce wrong data and convergence problems [25]. When calculations are generated by a script, the final checks cannot be performed, so it is essential that the FE calculations are robust; e.g., if 1000 calculations have to be performed for filling up a meta-model, and these run with a success rate of 50%, 500 calculations have to be debugged, which is an expensive and time consuming process. Moreover, for the same success rate, 250 calculations can remain wrong and therefore unusable, even after the second iteration. However, in order to consider FE calculations as a useful tool for engineering applications, a high success rate is desirable to avoid debugging.

3. Solving the interoperability problem with FlexMM

In the domain of zero-defect manufacturing, one of the issues in using digital twins is the interoperability in material data and constitutive behavior. When different process operations are interacting over a chain of processes, data transfer can be challenging. The material history has to be mapped over the entire duration of the production chain. This difficulty becomes especially visible if different commercial vendors are involved. To create the maturity required by the industry, FE models, as a part of a digital twin, must be multi-stage, multi-process and multi-vendor and the implemented constitutive behavior must be flexible yet robust, so that the chain of a production process can be simulated using different solvers and subroutines. Once a process step requires a different solver than the previous one, the latest results have to be transferred to the new solver of the sequential step, e.g. the geometries and meshes, the boundary conditions, and the material behavior. These items can be categorized into two domains:

- Local information of the state variables on the integration points of the mesh: this data transfer needs to be performed through mapping one simulation step to the next one, even if the mesh is changed dynamically, e.g. due to remeshing, during a simulation process.
- The constitutive models, which describe the behavior of the defined local and global state variables over the simulation steps.

As is mentioned in the previous section, there are a number of methods aimed at addressing the issues of the first domain; however, the second domain is yet to be addressed. Transferring constitutive models plays an important role in the interoperability, where the solver or the process characteristics can change during the process. The FlexMM approach has been developed for transferring the material behavior from one solver to another. Fig. 4 is a schematic of the FlexMM interface: a selected list of state variables with known definitions is transferred to the solver through FlexMM.

Fig. 5 shows an overview of the FlexMM approach of handling the materials data: the integrated FlexMM interface provides the materials behavior for all solvers. The material model can use the state variables defined in the state variable list for the material description. The FlexMM subroutine supports many kinds of material data with minimal boundary conditions. The state variables can be added to the process if

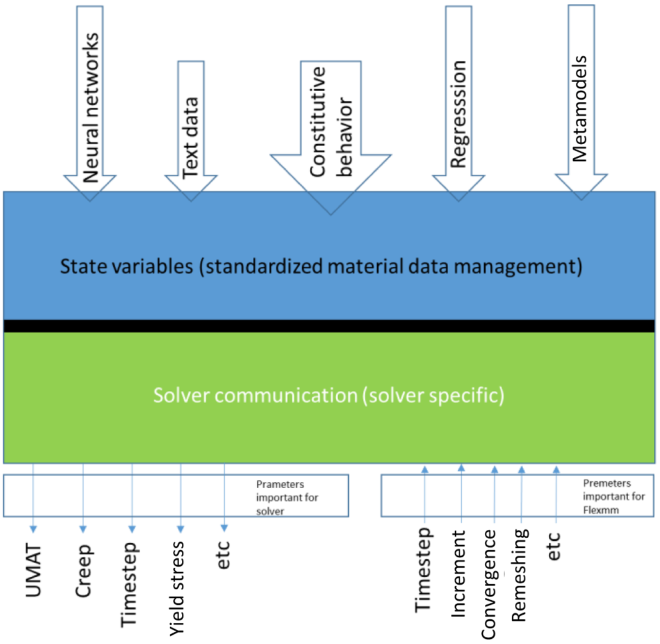

Neural networks
Text data
Constitutive behavior
Regresssion
Metamodels
State variables (standardized material data management)
Solver communication (solver specific)
Prameters important for solver
Premeters important for Flexmm
UMAT
Creep
Timestep
Yield stress
etc
Timestep
Increment
Convergence
Remeshing
etc

Fig. 4. Schematic overview of the FlexMM material modeling program. Blue: a generalized flexible representative of the constitutive behavior decoupled from the solver. Green: a general open structure to solve and maintain and improve the interoperability issue for solver communication between the different processes, production steps and solvers. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

required; however, the user is required to check whether the definitions of the state variables are the same in all FE solvers. In case these definitions are not the same, a transfer function can be programmed in the FlexMM subroutine to make a match between material model and subroutines in the solver. Moreover, if the solver output has some influence on the material model results, FlexMM updates the results and takes care of these changes.

There are many software packages available for structural and thermal analyses where the FlexMM method can be directly implemented. Although there is an overlap in the functionality of these packages, they are not able to share and transfer data without significant input of the user. FlexMM is designed to overcome this problem by defining the material behavior outside the solver and feeding the required information via subroutines of the commercial FEM packages using a standard interface. This approach is new in the modeling world, yet essential as a flexible and robust solution for the interoperability issues of digital twins.

Currently, the approach has been implemented in ABAQUS, Marc and Crystal (the in-house FE solver of Philips) via subroutines; however, the FlexMM approach can potentially be implemented for a much wider range of solvers, e.g. Ansys, COMSOL, LS-DYNA, PAM-STAMP, etc. It should be noted that, despite differences in software packages, their material subroutines are more or less similar.

3.1. The FlexMM approach towards constitutive modelling

One important but challenging aspect of multi-physical FE simulations is the implementation of the material behavior. Depending on the complexity of the selected “material – process” combination, the simulation could be simple, e.g. a tensile test curve, or arbitrarily difficult. For example, material deformation can be divided into elastic (reversible) and plastic (irreversible) regimes. The mechanical plastic deformation of a material causes changes in its internal microstructure. In a simple case, the microstructural changes create a textured material. Complexity can be added to the problem: e.g., grain recrystallization (static and/or dynamic) can be initiated, or there could be a strain-induced or stress-assisted solid phase change occurring in the material. Thus, the evolving microstructure during the forming process affects the elasto-plastic response of the material. The reliability of the results for such a complex process depends on a well-defined, validated and implemented constitutive behavior of the material.

Constitutive models of different processes from large scales, e.g. hot metal forming [26–28], sheet metal forming [29], powder/metal injection molding [30,31], or participation-hardenable grades of stainless steels [32], to the scale of micro-mechanical models [17] are available in the literature. These models could be either physics-based or phenomenological, or even meta-models, e.g. neural networks, based on the scattered experimental data. While academic studies and models help us understand the behavior of metals and alloys under various loading and operating conditions, they are not easily transferable to the industry due to their complexity and the difficulty in creating a robust solution in a commercially available numerical solver. As a functional solution, the FlexMM approach makes a bridge over the gap between cutting-edge models developed in academia and state-of-the-art solvers used in industry, by supporting all of the describing models for the materials behavior.

3.2. The FlexMM approach towards FE implementation

Complex material relations often cause convergence problems for the solvers, which usually require a highly skilled user to solve the problem. For example, if a material property is strongly dependent on the stresses or strains, the solver may never converge. In order to overcome this issue, the user should accept a bigger error and end the convergence iteration manually. Moreover, this problem can occur every time such a material property has to be implemented in an FE solver. The FlexMM approach, however, addresses solver-related convergence issues in a generic way, e.g. by automatically switching to another slower but more robust algorithm if required; therefore, such

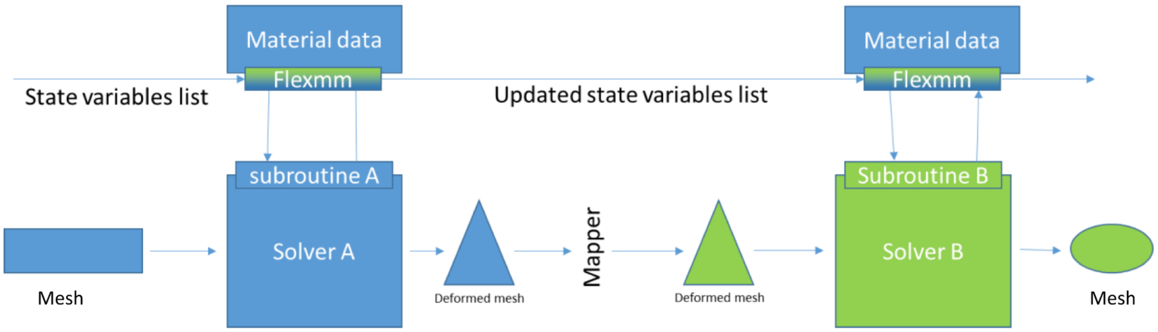


Fig. 5. The FlexMM interface for material model standardization. The FlexMM interface communicates between the material subroutines of the solver and the material data provided by the user.

problems can be simply avoided.

Another capability of FlexMM is the way in which changes of the state variables are treated. It is common to set a maximum change for state variables in the solver settings to minimize the error in the results. For example, if a phase transformation occurs during the process, time stepping should be reduced in the solver for a correct calculation of major stress changes. The time step should be changed so that the amount of phase transformation in one increment does not surpass a threshold, e.g. 10% of the total transformation. While it is not an easy task to control time stepping during the simulation, an adaptive time stepping method can be applied, see e.g. [33–35], although the time step cannot be saved, transferred or shared. A general behavior, such as time step reduction during phase transformation, is hardcoded in FlexMM, so that every time the change in any state variable exceeds a certain threshold, the time stepping is adjusted accordingly.

The FlexMM-based routines include the opportunity to store FE knowledge specifically related to the material, because the state variables are defined and structured separately, and have physical meaning as well as solver related meaning.

4. Mathematical implementation of FlexMM

Running an FE simulation and presenting the results requires three separate stages: pre-processing, solving the problem, and post-processing (most FE software packages provide tools for performing all three stages.) In the pre-processing stage, the geometric model and its consequent mesh are generated. Moreover, setting the initial values of the state variables and defining the materials models should be done in this stage. Once the problem is set, it will be passed on to the solver, which performs the FEA within an incremental-iterative procedure. Once the simulation is finished, the results are available for post-processing. The FlexMM approach implements material modeling in the first two stages by performing constitutive calculations within FlexMM and the equilibrium iterations within the FE solver. The material properties, which are referred to as “state variables” in FlexMM, can be divided into three categories.

1 Pre-defined dependent state variables: e.g. Young's modulus, Poisson's ratio and yield stress. These state variables are needed for calculating strains and stresses. Their behaviors have to be defined in the material model by the user. Note that defining them without a dependency implies that they are constants.
2 Pre-defined independent state variables: e.g. time, local components of stress and plastic strain. These state variables are always needed, but are calculated by the FEM code, and cannot be altered by the user, although they can be used in the material model to affect other state variables.
3 User-defined dependent state variables: e.g. particle size and viscosity. These state variables are dependent on the material model defined by the user. They have indirect influence on the calculation of strains and stresses via the pre-defined state variables. Note that these state variables can be freely defined and selected by the user.

Regarding the naming of the state variables, they are called *dependent* if their values are directly calculated within FlexMM using the defined relationships; however, if the value of a state variable is calculated outside FlexMM, e.g. separately by the FE solver, they are called *independent.* It should be noted that defining the values of the pre-defined dependent state variables is mandatory to perform an FEA, e.g. Young's modulus in a deformation problem.

In order to elucidate the FlexMM approach for modeling material properties, we take the yield stress (σ_y) as an example. For most structural materials that are modeled in the domain of material processing, σ_y is defined as a function of various state variables, regardless of whether the function is phenomenological or physics-based. A general relation of yield stress can be expressed as:

$$\sigma_y = f(\varepsilon_p, \dot{\varepsilon}_p, T, Z), \tag{1}$$

where *f* can be any function of equivalent plastic strain (ε_p), equivalent plastic strain rate ($\dot{\varepsilon}_p$), temperature (*T*), and any other state variable (*Z*), e.g. grain size. The FlexMM approach is based on differentiating the function *f* with respect to the FE solver's time step *t*. Moreover, because the response surface *f* is a multivariable function, the chain rule must be applied. Therefore, the derivative of Eq. (1) with respect to *t* can be written as:

$$\frac{d\sigma_y}{dt} = \frac{df(\varepsilon_p, \dot{\varepsilon}_p, T, Z)}{dt} = \frac{\partial f}{\partial \varepsilon_p}\frac{d\varepsilon_p}{dt} + \frac{\partial f}{\partial \dot{\varepsilon}_p}\frac{d\dot{\varepsilon}_p}{dt} + \frac{\partial f}{\partial T}\frac{dT}{dt} + \frac{\partial f}{\partial Z}\frac{dZ}{dt}. \tag{2}$$

The time increment in FE solvers has a finite value of Δt, during which the state variables exhibit a finite change. Therefore, the expression can be rewritten in terms of total derivatives as:

$$\frac{\Delta\sigma_y}{\Delta t} = \frac{\partial f}{\partial \varepsilon_p}\frac{\Delta\varepsilon_p}{\Delta t} + \frac{\partial f}{\partial \dot{\varepsilon}_p}\frac{\Delta\dot{\varepsilon}_p}{\Delta t} + \frac{\partial f}{\partial T}\frac{\Delta T}{\Delta t} + \frac{\partial f}{\partial Z}\frac{\Delta Z}{\Delta t}. \tag{3}$$

Multiplying both sides of Eq. (3) by Δt results in the following form of the incremental changes of the state variables:

$$\Delta\sigma_y = \frac{\partial f}{\partial \varepsilon_p}\Delta\varepsilon_p + \frac{\partial f}{\partial \dot{\varepsilon}_p}\Delta\dot{\varepsilon}_p + \frac{\partial f}{\partial T}\Delta T + \frac{\partial f}{\partial Z}\Delta Z. \tag{4}$$

This can be rewritten as the following dot product:

$$\Delta\sigma_y = \begin{bmatrix} \frac{\partial f}{\partial \varepsilon_p} & \frac{\partial f}{\partial \dot{\varepsilon}_p} & \frac{\partial f}{\partial T} & \frac{\partial f}{\partial Z} \end{bmatrix} \cdot \begin{bmatrix} \Delta\varepsilon_p \\ \Delta\dot{\varepsilon}_p \\ \Delta T \\ \Delta Z \end{bmatrix}. \tag{5}$$

Furthermore, the vector differential operator can be represented by the Del operator ∇ to shorten the formula. In general, if a dependent state variable *p* can be written as a function of a set of other state variables, i.e. $p = f(\mathbf{v})$, its incremental change can be calculated as the dot product of the del operator of *f* and the incremental changes of those elements, i.e.

$$\Delta p = \nabla f(\mathbf{v}) \cdot \Delta\mathbf{v}. \tag{6}$$

In the FlexMM approach, the increments of the dependent variables are integrated to calculate the final values of those variables[2]. As for a state variable *p*, its value at a time step *t* (p_t) is the summation of its value at the previous time step (p_{t-1}) and the newly calculated change (Δp). Therefore, the mathematical implementation of the FlexMM approach can be concisely written as follows:

$$p_t = p_{t-1} + \nabla f(\mathbf{v}) \cdot \Delta\mathbf{v}. \tag{7}$$

For the calculation of *p* at $t = 0$, the initial value p_0 is required.

4.1. Some notes on the implementation in practice

It should be noted that the FlexMM approach is path-dependent, and the material model can change over the course of the FE simulation. In other words, $f(\mathbf{v})$ can change from one formula to another if certain conditions are met, and such a change would not affect the operability of the FlexMM approach; however, the continuity of the formulae should be assured in such cases. In order to conduct a simulation and define its material model using the FlexMM approach, the solution of the Del operator is required in advance, which can be found based on the formulation of the response surface $f(\mathbf{v})$. Any response surface that can be described as $f(\mathbf{v})$ is suitable for implementation, regardless of its mathematical form. The solution of the gradient, which is the actual response surface in the FlexMM approach, can be formulated as explicit

[2] Referring to the naming of the state variables, they are called *independent* if they are not present on the LHS of Eq. (6), and *dependent* if they are also present on its LHS.

equations, regardless of whether these are physics-based or phenomenological, or are stored in lookup tables, neural networks, or a combination of them.

Moreover, in order to solve Eq. (6) accurately, the most concerning issue is the selected step size. (See Section 6.5. for further discussion on the time step size.) Noting that $\nabla f(\mathbf{v})$ is a pre-defined function, and its value(s) as well as $\Delta\mathbf{v}$ are known at any given step, Δp can be calculated correctly in one call with a properly sized step. On the other hand, if the calculation of Δp requires corrections to meet certain conditions, this can be done via a predictor-corrector method within a mapping algorithm. For example, in the case of plasticity, one should perform a stress update procedure, which can be done by implementing a predictor-corrector method within the radial return algorithm [36].

Furthermore, for an implicit FE solver, a consistent tangent matrix would be required. This issue can be addressed by linearizing the yield stress dependence on the plastic strain (work hardening) to define a consistent tangent matrix within each iteration of the implemented stress update procedure. Note that the established linear relationship must be updated for each such iteration. To increase the robustness of the method, we are using a secant stiffness matrix instead in the current implementation of FlexMM, although it may slow down the calculations.

5. Implementation of FlexMM in Marc

The implementation of Eq. (7) in Marc was performed based on its “model section” data transfer method, which allows users to generate and transfer the simulation data files separately. Using this method, all information in terms of nodal coordinates, element connectivity, material model, etc. is available. Based on the model section capabilities, a subroutine is written in FORTRAN 90 for defining the material model using the described standardized FlexMM approach, which provides the full flexibility of the connected state variables. Each state variable can be connected to other ones using lookup tables or equations; however, the physical meaning has to be described by the user.

In order to write a subroutine for any specific material model, an in-house software has been developed in MATLAB (release 2018b, The MathWorks, Inc., Natick, MA, USA). The GUI of the code is shown in Fig. 6. The top screenshot shows the list of the state variables, their types and initial values. A list of all available state variables in this implementation is provided in Appendix A. By selecting each of the state variables, their corresponding details will be shown in the lower table. For example, as shown in the figure, the variable “Flowstress_austenite” is a function of “Temperature”, “Equiv_plast_strain-rate” and “Y_austenite”. Each of these determining variables can be edited in the integrator menu; see Fig. 6 (bottom). Once the material modelling is done in the code, the FE solver (ABAQUS, Marc or Crystal) should be selected to generate the solver-specific subroutine.

6. Case study

The application of the FlexMM approach has investigated in previous works of the authors:

- Predicting shape defects caused by strain induced transformation in stainless steel [18].
- Predicting delayed Cracking in stainless steel [37].
- Predicting shape defects caused by stress relaxation during heat treatments of steel [38].
- Skin deformation during loading and unloading [39].
- Deformations during sintering caused by viscous flow and shrinkage.

Although the FlexMM approach has been applied to different processes, the detailed description of the method remained to be explained, which is the main focus of this paper. The last-mentioned item in the abovementioned list is selected as the case study in this work.

6.1. Sintering of Tungsten

Metal Injection Molding (MIM) is a metal forming process by which fine-powdered metal is mixed with a measured amount of binder material to comprise a feedstock capable of being handled by plastic processing equipment through a process known as injection molding. The molding process allows dilated complex parts to be shaped in a single step and in high volume. After molding, the powder-binder mixture is subjected to debinding and sintering processes to remove the binder and densify the powders, respectively. Powder rheology allows this shot to be distributed into multiple cavities, thus becoming cost-effective for small, intricate, and high-volume products, which would otherwise be quite expensive to produce by alternate or classical methods.

A typical Metal Injection Molding (MIM) process would involve the following steps:

1- Injection molding
2- Debinding
3- Presintering
4- Sintering

During the sintering process, the powder, with an initially low relative density, is heated and grains become liquid. Then, the material flows into the gaps and surface tension causes the material to shrink, resulting in a final product with a relative density of 95~99% [40].

In a MIM process, the biggest shape defects appear during the sintering stage. Therefore, the challenges in the modelling of such a complicated process involve accurate prediction of the strain tensor, which is composed of viscous, deviatoric and volumetric parts. In the following sections, we describe a model for the sintering process, and simulate the process for producing the Tungsten electrode in a beamer light (see Fig. 7), using FEM performed by Marc (release 2018.1, MSC Software Corporation). The results show how the FlexMM approach assists the simulation process.

6.2. The sintering model and material

On a macroscopic level, there are two driving forces for shape change during the sintering of the injection molded product: deviatoric behavior (viscosity/creep) and volumetric strain (shrinkage). Moreover, these strains are time dependent and usually defined in terms of strain rates as $\Delta\varepsilon = \dot{\varepsilon}\Delta t$. The total strain rate ($\dot{\varepsilon}_{\text{total}}$) during the sintering can be expressed as:

$$\dot{\varepsilon}_{\text{total}} = \dot{\varepsilon}_{\text{c}} + \dot{\varepsilon}_{\text{v}}, \tag{8}$$

where $\dot{\varepsilon}_{\text{c}}$ and $\dot{\varepsilon}_{\text{v}}$ are the creep and volumetric strain rates, respectively. The following equations summarize the model used in the current study, as described in [41]. The readers are referred to the work of Olevsky [42] for a thorough review on the topic.

The deviatoric equation is described as:

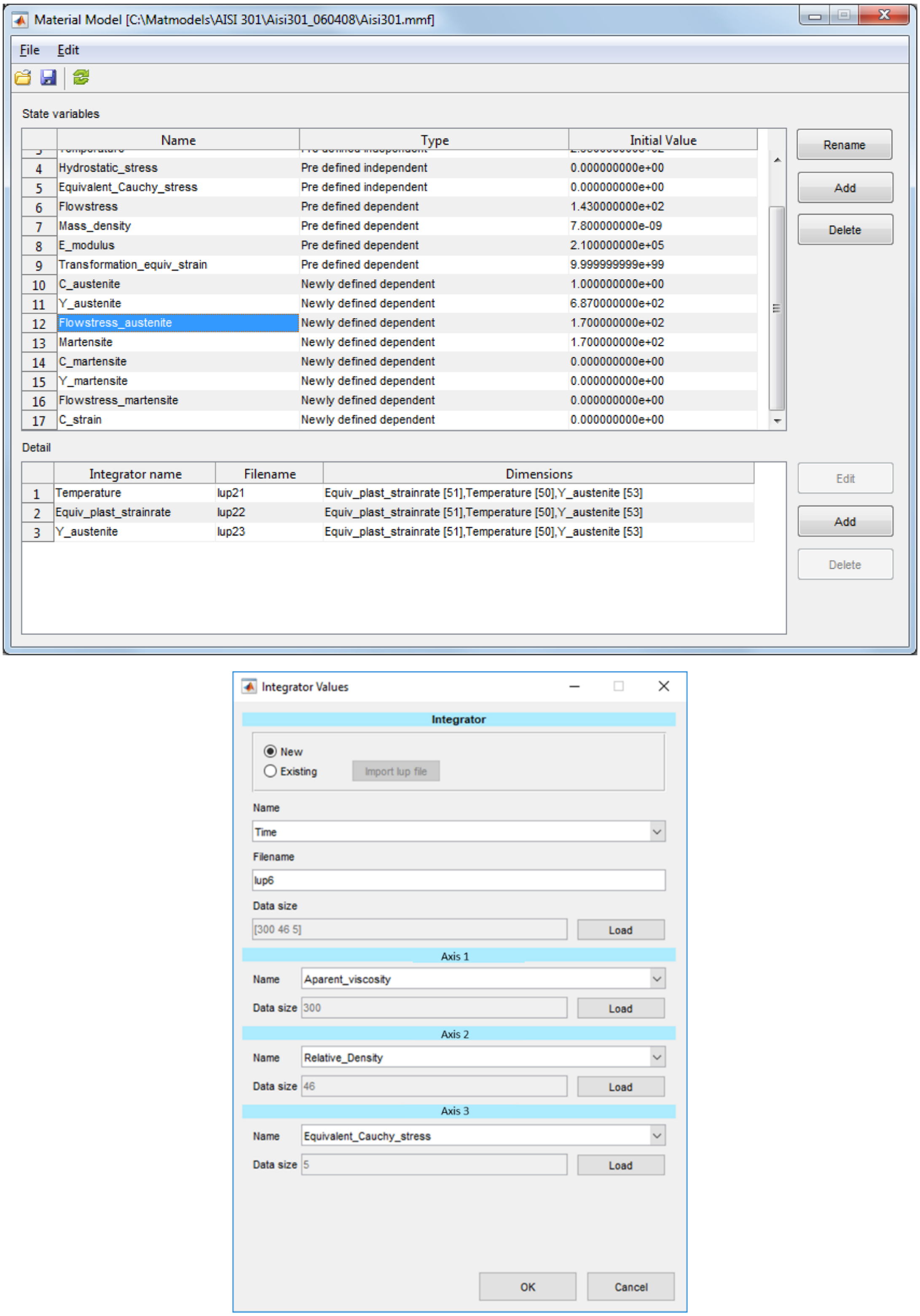
$$\dot{\varepsilon}_{\text{c}} = \sigma_{\text{dev}}/2\varphi\eta, \tag{9}$$

where σ_{dev} is the deviatoric stress, which is caused by the gravitational force in the product, φ is the effective shear viscosity, and η is the viscosity of the fully dense material. The effective shear viscosity is described by:

$$\varphi = (1-\theta)^2\ , \tag{10}$$

where θ is the porosity defined as the ratio of the volumes of voids to the total volume. Moreover, the viscosity of the fully dense material is defined using an Arrhenius formula as follows:

$$\eta = \eta_0 \exp(E/RT), \tag{11}$$



Material Model [C:\Matmodels\AISI 301\Aisi301_060408\Aisi301.mmf]
File
Edit

State variables

	Name	Type	Initial Value
4	Hydrostatic_stress	Pre defined independent	0.000000000e+00
5	Equivalent_Cauchy_stress	Pre defined independent	0.000000000e+00
6	Flowstress	Pre defined dependent	1.430000000e+02
7	Mass_density	Pre defined dependent	7.800000000e-09
8	E_modulus	Pre defined dependent	2.100000000e+05
9	Transformation_equiv_strain	Pre defined dependent	9.999999999e+99
10	C_austenite	Newly defined dependent	1.000000000e+00
11	Y_austenite	Newly defined dependent	6.870000000e+02
12	Flowstress_austenite	Newly defined dependent	1.700000000e+02
13	Martensite	Newly defined dependent	1.700000000e+02
14	C_martensite	Newly defined dependent	0.000000000e+00
15	Y_martensite	Newly defined dependent	0.000000000e+00
16	Flowstress_martensite	Newly defined dependent	0.000000000e+00
17	C_strain	Newly defined dependent	0.000000000e+00

Detail

	Integrator name	Filename	Dimensions
1	Temperature	Iup21	Equiv_plast_strainrate [51],Temperature [50],Y_austenite [53]
2	Equiv_plast_strainrate	Iup22	Equiv_plast_strainrate [51],Temperature [50],Y_austenite [53]
3	Y_austenite	Iup23	Equiv_plast_strainrate [51],Temperature [50],Y_austenite [53]

● New
○ Existing

Name: Time

Filename: Iup6

Data size: [300 46 5]

Name: Aparent_viscosity

Data size: 300

Name: Relative_Density

Data size: 46

Name: Equivalent_Cauchy_stress

Data size: 5

Fig. 6. GUIs of the material modeling code: (top) shows the main page, where the state variables are selected, and (bottom) shows the integrator menu in which the response surface of the selected state variables can be edited.

where η_0 is the proportionality constant, E is the activation energy for viscous flow, R is the universal gas constant, and T is the absolute temperature. The constants of the equation are acquired by data fitting and data optimization processes; see Appendix B for the details of the procedure.

The shrinkage of the material is modelled as follows:

$$\dot{\varepsilon}_{\mathrm{v}} = \frac{\sigma_{\mathrm{m}} - \sigma_{\mathrm{s}}}{6\psi\eta}\delta_{\mathrm{y}}, \tag{12}$$

where σ_{m} is the hydrostatic stress caused by the gravity, which presses the material into the voids, σ_{s} is the sintering stress caused by the capillary forces between the grains, ψ is the effective bulk viscosity, and δ_{y} is a unit vector for the direction of the strains. The sinter stress is

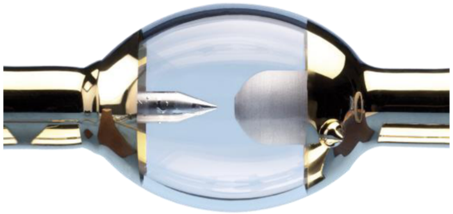


Fig. 7. Metal injection molded tungsten electrode.

calculated by:

$$\sigma_s = \frac{3\gamma}{r_0}(1-\theta)^2, \tag{13}$$

where γ is the surface tension, and r_0 is the initial particle size. Furthermore, the porosity function of Skorohod is used for the calculation of the effective bulk viscosity:

$$\psi = \frac{2(1-\theta)^3}{3\theta}. \tag{14}$$

For working with the model, the material is assumed to be homogeneous throughout the process. Table 1 summarizes the material data used in the experiments and other required parameters for working with the model.

6.3. Implementing the FlexMM approach

The implementation of the FlexMM approach for calculating the volumetric and deviatoric strain rates is done based on Eq. (6), as follows:

$$\Delta\dot{\varepsilon}_c = \begin{bmatrix} \frac{\partial\dot{\varepsilon}_c}{\partial\sigma_{dev}} & \frac{\partial\dot{\varepsilon}_c}{\partial\varphi} & \frac{\partial\dot{\varepsilon}_c}{\partial\eta} \end{bmatrix} \cdot \begin{bmatrix} \Delta\sigma_{dev} \\ \Delta\varphi \\ \Delta\eta \end{bmatrix}, \tag{15}$$

and

$$\Delta\dot{\varepsilon}_v = \begin{bmatrix} \frac{\partial\dot{\varepsilon}_v}{\partial\sigma_m} & \frac{\partial\dot{\varepsilon}_v}{\partial\sigma_s} & \frac{\partial\dot{\varepsilon}_v}{\partial\psi} & \frac{\partial\dot{\varepsilon}_v}{\partial\eta} \end{bmatrix} \cdot \begin{bmatrix} \Delta\sigma_m \\ \Delta\sigma_s \\ \Delta\psi \\ \Delta\eta \end{bmatrix}. \tag{16}$$

Then, the dependent variables, i.e. $\Delta\varphi$, $\Delta\mu$, $\Delta\sigma_s$, and $\Delta\psi$, are defined as a function of pre-defined independent (T) and user-defined dependent (θ) state variables:

$$\Delta\varphi = \frac{\partial\varphi}{\partial\theta}\Delta\theta, \tag{17}$$

$$\Delta\eta = \frac{\partial\eta}{\partial T}\Delta T, \tag{18}$$

$$\Delta\sigma_s = \frac{\partial\sigma_s}{\partial\theta}\Delta\theta, \tag{19}$$

Table 1
Material specifications and model's parameters.

Chemical composition	Tungsten 99.95 %
Particle size (r_0)	8 μm
Binder	Polyethylene
Initial porosity (θ)	0.35
Surface tension (γ)	3.0 N/m [43]
Activation energy for viscous flow (E)	410.1 kJ/mol
Proportionality constant of Eq. (11) (η_0)	67 Pa.s

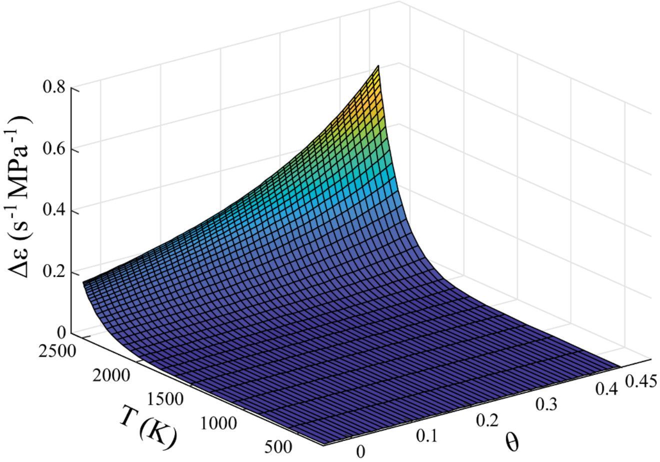


Fig. 8. Graphical representation of material equations of creep $\Delta\dot{\varepsilon}_c$ as a function of temperature and porosity, as used by FlexMM.

$$\Delta\psi = \frac{\partial\psi}{\partial\theta}\Delta\theta. \tag{20}$$

It can be seen that these four dependent variables are functions of two variables, namely temperature and porosity; therefore, the strain rates can be re-formulated as:

$$\Delta\dot{\varepsilon}_c = \begin{bmatrix} \frac{\partial\dot{\varepsilon}_c}{\partial\sigma_{dev}} & \frac{\partial\dot{\varepsilon}_c}{\partial\theta} & \frac{\partial\dot{\varepsilon}_c}{\partial T} \end{bmatrix} \cdot \begin{bmatrix} \Delta\sigma_{dev} \\ \Delta\theta \\ \Delta T \end{bmatrix}, \tag{21}$$

and

$$\Delta\dot{\varepsilon}_v = \begin{bmatrix} \frac{\partial\dot{\varepsilon}_v}{\partial\sigma_m} & \frac{\partial\dot{\varepsilon}_v}{\partial\theta} & \frac{\partial\dot{\varepsilon}_v}{\partial T} \end{bmatrix} \cdot \begin{bmatrix} \Delta\sigma_m \\ \Delta\theta \\ \Delta T \end{bmatrix}. \tag{22}$$

As an example, Fig. 8 shows the dependence of the creep rate on T and θ. Upon establishing the model via FlexMM, the required Marc subroutines for describing the material's behavior are generated with the toolbox described in Section 5.

6.4. Simulation and results

Fig. 9 (left) shows the modeled electrode in its original configuration. The two ends of the sample are fixed in the directions perpendicular to the electrode's shaft, and the gravity is applied as a body force. The sintering process is simulated by changing the temperature from room temperature (293 K) to 2250 K with a rate of 25K/min. Following 1 h of holding time, the temperature is reduced to room temperature with a rate of − 50K/min. It should be noted that the porosity is updated according to the estimated shrinkage ($\dot{\varepsilon}_v$) at every increment.

Fig. 9 (right) shows the sample at the end of the sintering process. The visible bent of the electrode is due to the hydrostatic stress induced by gravity. Fig. 10 shows the collected values of the creep and volume strains calculated from the FEM simulation.

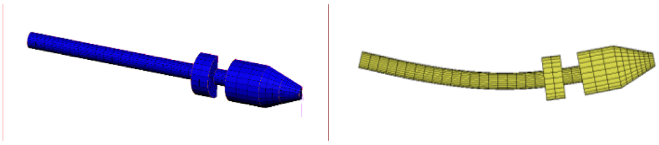


Fig. 9. A 3D beamer electrode (left) before and (right) after the sintering process.

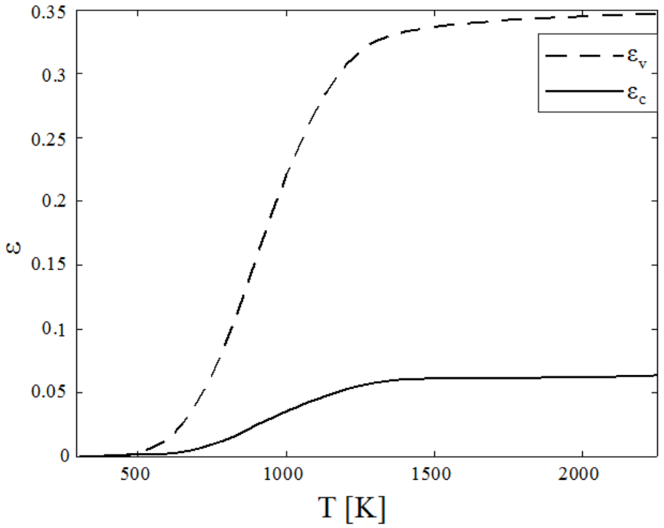


Fig. 10. The calculated volume and creep strains during the sintering process of the electrode.

Table 2
List of simulations.

Test no.	FlexMM	Assigned *N*	Performed *N*	Simulation time (h:mm:ss)
1	No	100	16 (crashed)	Not converged
2	No	295	295	0:41:66
3	No	1000	1000	1:54:12
4-6	Yes	1000-100-10	74	0:09:20

6.5. The advantages of FlexMM

In order to show how FlexMM can assist FEM simulations, a series of simulations for the same problem were performed by varying two parameters: (1) the method for material definition, i.e. either classic or via FlexMM, and (2) the user-defined number of increments (*N*). Table 2 summarizes the performed simulations and their results.

For performing a simulation, the time step (Δt) for an increment is derived from a fixed fraction of the total analysis time, by assigning the number of increments. For test 1, the number of increments was set to 100, but the solver could not finish the simulation. As a solution, the number of increments were increased to 1000, resulting in higher values of both the calculation time and the probability of a successful simulation. Doing so resulted in a successful simulation that took ~ 2 h to run. By varying the number of assigned *N* in a large range, the minimum number of steps for running a successful simulation was found to be 295, as in test 2.

For the other tests, the FlexMM approach was used. Table 2 shows that for these tests, the number of increments for performing the simulation was 74 regardless of the assigned number. This is because an optimized time step (Δt) corresponding to each increment can be adapted based on the gradients of the variable in that increment. In other words, if the gradient of a user-defined state variable is low, a large time step for the increment is assigned, and vice versa. The changes of Δt were controlled by setting a maximum 1% shrinkage strain per increment. Fig. 11 shows the changes of the adapted time steps for each increment: once the gradient of the porosity increases, Δt becomes smaller, and larger values of Δt were adapted as the gradient became negligible.

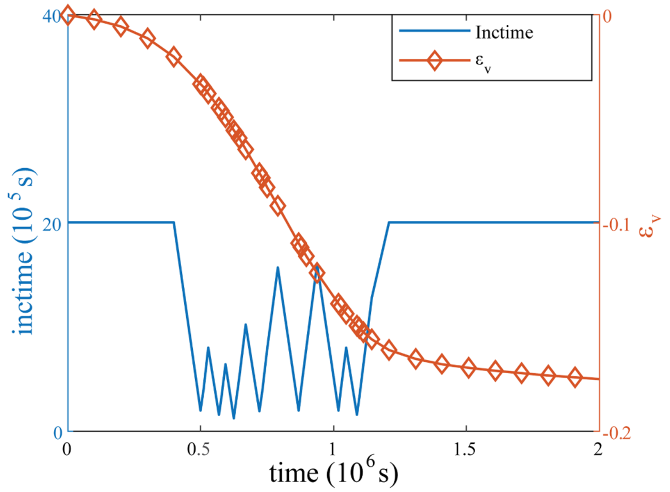


Fig. 11. The time step for each increment (inctime) and shrinkage versus simulation time in test 5 (N = 1000).

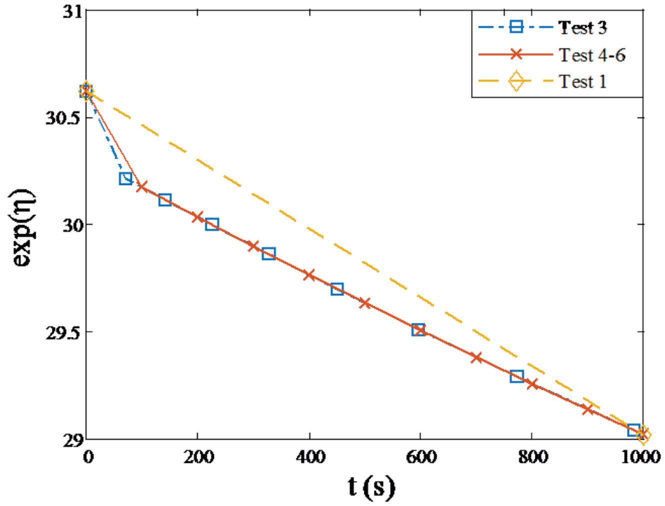


Fig. 12. The calculated viscosity of the first 1000 s of tests 1, 3, and 4-6; see Table 1 for a description of the tests. The lines illustrate the trends.

The ability of performing FEM simulations with adaptive time steps realized via FlexMM ensures that the simulations can be performed without any convergence problem, and within the shortest possible computational time. Simultaneously, this ability eliminates time step-dependent calculation errors, which can occur if the fixed time step is not small enough to capture all relevant physics. For example, Fig. 12 shows the calculated viscosity for tests 1, 3 and 4 for the first 1000 s. The viscosity exhibits some sudden changes at the very first stages of the simulated process, which are visible only when applying small time steps, i.e. in tests 2 and 3. In this example, the solutions of all tests are essentially equal; however, in a general case, e.g. for a function that depends, not only on the current state of the system, but also on the previous values, any missing behavior of the function results in unrealistic deviations. These errors are time step-dependent, and should be avoided by a carefully selected Δt, which is not straightforward, and takes a lot of time to be found. This optimum fixed value of Δt should be small enough to capture all physics, but not too small so that the computation cost will not be wasted. As a remedy, the adaptive time step through FlexMM not only optimizes the simulation time, but also makes sure that all effective changes in the models are captured.

7. Conclusions

In this paper, a standard method for material description in FEM simulations of multistage processes has been proposed and developed. This flexible material modeler, called FlexMM, gives the opportunity to switch between steps, processes and solvers in a structured and robust way.

First, the mathematical implementation of FlexMM with a chain-rule approach is described. It is shown that the incremental changes of an independent state variable, which is a function of other state variables, i.e. $p = f(\mathbf{v})$, can be written as $\Delta p = \nabla f\,(\mathbf{v})\cdot\Delta\mathbf{v}$ therefore, Δp can be calculated for a known $\nabla f(\mathbf{v})$, which can be stored and accessed via look-up tables, trained artificial neural networks, or analytical expressions. In addition, a computer implementation of FlexMM is described to demonstrate the simplicity of transferring data to FEM. Currently, the code can provide subroutines for ABAQUS, Marc and Crystal.

Moreover, the application of FlexMM is demonstrated by simulating a sintering process of tungsten powders. Through this case study, some advantages of FlexMM are discussed, which can be summarized as follows: "FlexMM defines the time step on-the-fly for each increment based on the pre-defined derivatives. This results in an optimized time step, which ensures the capturing of all relevant physics of the used models and guarantees a robust FE simulation."

Furthermore, the FlexMM approach reduces the overall cumulative numerical error of the calculations. The main reason for that is the ability of FlexMM to offer on-the-fly adaptive time stepping based on the defined constitutive models, which ensures an optimum total number of steps to reach the converged solution. This means that the cumulative error caused by the numerical method is minimized accordingly.

Ultimately, an approach such as FlexMM can help create digital twins for manufacturing processes with materials that undergo complex (micro-)mechanical evolutions. FlexMM provides a framework to simplify and generalize the (micro-)mechanical models to build a material's digital twin, which can be used in automation systems for feed-forward and feed-backward real-time process control to achieve zero-defect manufacturing.

CRediT authorship contribution statement

Manso Groen: Methodology, Software, Validation, Formal analysis, Investigation, Visualization, Writing - original draft. **Soheil Solhjoo:** Formal analysis, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization, Supervision. **Ruud Voncken:** Methodology, Software, Validation, Writing - review & editing. **Jan Post:** Conceptualization, Methodology, Investigation, Resources, Supervision, Project administration. **Antonis. I. Vakis:** Resources, Writing - original draft, Writing - review & editing, Supervision, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The development of the presented approach (FlexMM) and its validations are mainly based on two European Projects: (1) the RFCS (Research Fund for Coal and Steel) Press-perfect project (403 RFSR-CT-2014-00005), aimed to develop physics-based simulations of stainless steels, and (2) the ITEA project 16010 called VMAP: a new interface standard for integrated virtual material modelling in manufacturing industry. Further information can be found in the project's webpage https://www.vmap.eu.com/.

Appendix A

The lists of all pre-defined dependent (Table 3) and pre-defined independent (Table 4) state variables available in the current implementation of FlexMM are given below.

Table 3
List of the available predefined dependent state variables.

State variable name	Subroutine	Symbol
Flowstress	Elastic plastic	σ_y
E_modulus	Elastic plastic	E
Poisson_ratio	Elastic plastic	
Mass_density	Volumetric changes	ρ
Thermal_expansion_coef	Volumetric changes	
Thermal_conductivity	Thermal calculation	
Specific_heat	Thermal calculation	
Damage	Damage calculation	
Creep strain	Creep	ε_c
Viscous strain	Creep	ε_{visc}
Transformation equivalent strain	Creep	ε_{tr}
R0	Anisotropy Hill	$R0$
R45	Anisotropy Hill	$R45$
R90	Anisotropy Hill	$R90$
Ratio S45 S0	Anisotropy Barlat	
Ratio S90 S0	Anisotropy Barlat	
Mooney C10	Strain energy density	
Mooney C11	Strain energy density	
Compression modulus	Strain energy density	

Table 4
List of the available predefined independent state variables.

State variable name	Symbol
Time	t
Temperature	T
Local stress components σij	σ_{ij}
Von_Mises stress	σ_{m}
Material angles	θ, φ
Equivalent plastic strain	ε_{p}
Total Lagrange strain	ε_{l}
Hydrostatic stress	σ_{h}
Triaxiality	
Equiv_plast_strainrate	$\dot{\varepsilon}_{\mathrm{p}}$
Local strains εij	ε_{ij}
Hyper elastic stress component εij	

Appendix B

The calibration of the viscosity model (Eq. (11)) was done by fitting solutions of Eq. (12) to experimental data, which were collected from 36 sintering tests. These tests were performed at 4 different temperatures and 3 different sintering times, and 3 tests were done for each pair of time increment and temperature. The weight and volume of the injection molded samples were ∼ 4.85 g and 50 × 3 × 3 mm^3, respectively. After the sintering process, the weight and volume of each sample was measured, their density ρ was calculated, and the shrinkage strain was calculated by:

$$\varepsilon_{\mathrm{v}} = \frac{(\rho - \rho_0)}{3\rho}, \tag{23}$$

with $\rho_0 \approx 10.81 \times 10^3 \mathrm{kg/m^3}$ being the density of the pre-sintered sample. The measured and averaged density over each set of time and temperatures is reported in Table 5

Substituting Eqs. (11), (13) and (14) into Eq. (12), one can find $\dot{\varepsilon}_{\mathrm{v}} = \dot{f}\,(\eta_0, E, T, \gamma, r_0, \theta, \sigma_{\mathrm{m}}, \delta_{\mathrm{y}}) = \dot{f}\,(\eta_0, E, T, \boldsymbol{C})$, where $\boldsymbol{C}$ is the set of known values (Table 1). For simulating these calibrating tests, the hydrostatic stress σ_{m} is disregarded, because the samples are supported against the gravitational forces and no bending takes place. Moreover, the shrinkage is assumed to be isotropic, resulting in $\delta_{\mathrm{y}} = 3^{-1/2}[1\ \ 1\ \ 1]^{\mathrm{T}}$. In order to compare this derivative function with ε_{v} (Eq. (23)), the temperature is required to be expressed as a function of time, which can be calculated simply by means of its derivatives based on the furnace used in the experiments. Table 6 summarizes the required rates. The initial temperature value and the time range were set to $T(t = 0) = 293$ K and $t = [0,\ 500]$ min, respectively.

The derivative equation of $\dot{\varepsilon}_{\mathrm{v}}$ was solved by the Bogacki-Shampine method [44] using its implementation in the ode23 function in MATLAB [45]. The values of η_0 and E were approximated by minimizing a residual function defined as:

$$r = \sqrt{\frac{1}{N}\Sigma(\varepsilon_{\mathrm{v,model}} - \varepsilon_{\mathrm{v,experiment}})^2}, \tag{24}$$

where $N = 12$ is the total number of experimental measurements. The minimization was performed using the fminsearch function of MATLAB [46], which uses the Nelder-Mead simplex search method of Lagarias et al. [47].

The optimized values of η_0 and E were approximated as 69 Pa.s and 410kJ/mol, respectively. Fig. 13 shows the values of shrinkage strain measured from experiments and estimated by Eq. (12).

Table 5
The averaged measured densities [$10^3 \mathrm{kg/m^3}$] for the sintered samples.

	Temperature (K) 1200	1500	2000	2250	
Holding time (h)	1	10.67	11.61	17.41	17.91
	4	10.453	12.35	17.98	18.18
	7	11.05	13.35	17.94	15.99

Table 6
Rate of changes of temperature with time at different stages of the sintering process.

	Heating	Holding	Cooling
dT/dt (K/min)	25	0	− 50

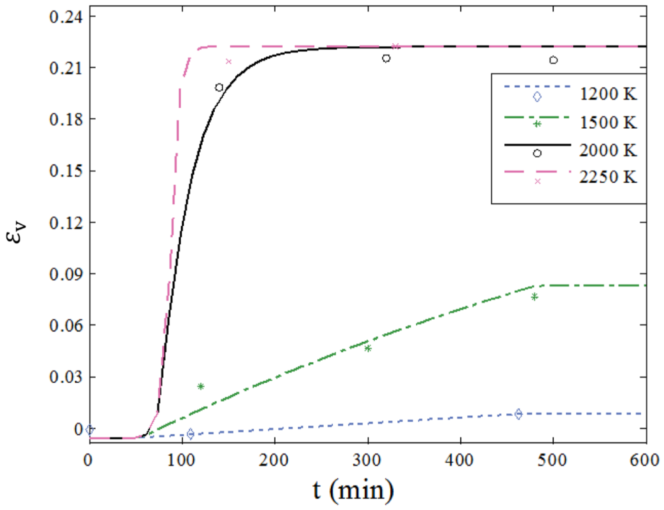


Fig. 13. The values of shrinkage strain (ε_v) obtained from the experiments (symbols) and the model (lines) at different sinter temperatures.

References

[1] Zhong RY, Xu X, Klotz E, Newman ST. Intelligent manufacturing in the context of industry 4.0: a review. Engineering 2017;3:616–30. https://doi.org/10.1016/J.ENG.2017.05.015.

[2] Wang K. Intelligent predictive maintenance (IPdM) system–industry 4.0 scenario. WIT Trans Eng Sci 2016;113:259–68.

[3] Dureisseix D, Bavestrello H. Information transfer between incompatible finite element meshes: application to coupled thermo-viscoelasticity. Comput Methods Appl Mech Eng 2006;195:6523–41. https://doi.org/10.1016/J.CMA.2006.02.003.

[4] Bucher A, Meyer A, Görke U-J, Kreißig R. A comparison of mapping algorithms for hierarchical adaptive FEM in finite elasto-plasticity. Comput Mech 2007;39:521–36. https://doi.org/10.1007/s00466-006-0051-z.

[5] Fernandes JLM, Martins PAF. All-hexahedral remeshing for the finite element analysis of metal forming processes. Finite Elem Anal Des 2007;43:666–79. https://doi.org/10.1016/J.FINEL.2007.02.001.

[6] Scrimieri D, Afazov SM, Becker AA, Ratchev SM. Fast mapping of finite element field variables between meshes with different densities and element types. Adv Eng Softw 2014;67:90–8. https://doi.org/10.1016/J.ADVENGSOFT.2013.08.003.

[7] Scrimieri D, Afazov SM, Ratchev SM. An in-core grid index for transferring finite element data across dissimilar meshes. Adv Eng Softw 2015;88:53–62. https://doi.org/10.1016/J.ADVENGSOFT.2015.06.001.

[8] Ahrem R. Multidisciplinary simulations with the coupling library MpCCI. PAMM 2002;1:39. 10.1002/1617-7061(200203)1:1 < 39::AID-PAMM39 > 3.0.CO;2-0.

[9] Wolf K, Bayrasy P, Brodbeck C, Kalmykov I, Oeckerath A, Wirth N. MpCCI: neutral interfaces for multiphysics simulations. Sci. comput. algorithms ind. simulations Cham: Springer International Publishing; 2017. p. 135–51. https://doi.org/10.1007/978-3-319-62458-7_7.

[10] Digimat - the composite materials modeling software, https://www.e-xstream.com/products/digimat/about-digimat (accessed January 18, 2019).

[11] Mayer N, Van Den Broucke B, Prowe J, Havar T, Hinterhölzl R. Finite element mapping for incompatible FE meshes of composite structures. Adv Eng Softw 2016;99:81–8. https://doi.org/10.1016/J.ADVENGSOFT.2016.05.007.

[12] Digimat - the material modeling platform, http://www.mscsoftware.com/product/digimat (accessed January 18, 2019).

[13] Composites Modeler | ABAQUS CAE - Dassault Systèmes®, https://www.3ds.com/products-services/simulia/products/abaqus/add-ons/composites-modeler-for-abaquscae/ (accessed January 18, 2019).

[14] Composite Simualtion | ANSYS Composite PrepPost, https://www.ansys.com/products/structures/ansys-composite-preppost (accessed January 18, 2019).

[15] Seung-Yong Y. Conversion of ABAQUS user material subroutines. J Comput Struct Eng Inst Korea 2010;23:635–40.

[16] Tao F, Zhang M. Digital twin shop-floor: a new shop-floor paradigm towards smart manufacturing. IEEE Access 2017;5:20418–27. https://doi.org/10.1109/ACCESS.2017.2756069.

[17] Knapp GL, Mukherjee T, Zuback JS, Wei HL, Palmer TA, De A, DebRoy T. Building blocks for a digital twin of additive manufacturing. Acta Mater 2017;135:390–9. https://doi.org/10.1016/J.ACTAMAT.2017.06.039.

[18] Post J. On the constitutive behaviour of Sandvik Nanoflex: modelling, experiments and multi-stage forming. University of Twente; 2004https://research.utwente.nl/en/publications/on-the-constitutive-behaviour-of-sandvik-nanoflex-modelling-exper.

[19] Post J, Klaseboer G, Stinstra E, van Amstel T, Huetink J. Multi-stage metal forming: variation and transformation. J Mater Process Technol 2009;209:2648–61. https://doi.org/10.1016/J.JMATPROTEC.2008.06.061.

[20] Solhjoo S, Vakis AI, Pei YT. Two phenomenological models to predict the single peak flow stress curves up to the peak during hot deformation. Mech Mater. 2017. https://doi.org/10.1016/j.mechmat.2016.12.001.

[21] Solhjoo S. Determination of flow stress under hot deformation conditions. Mater Sci Eng A 2012;552:566–8. https://doi.org/10.1016/j.msea.2012.05.057.

[22] Söderberg R, Wärmefjord K, Carlson JS, Lindkvist L. Toward a Digital Twin for real-time geometry assurance in individualized production. CIRP Ann 2017;66:137–40. https://doi.org/10.1016/J.CIRP.2017.04.038.

[23] van Ravenswaaij R, van Tijum R, Hora P, van den Boogaard T, Engel U. Towards the generation of taylored tempered components: concept definition and process parameters optimization, ETH Zurich. Institute of Virtual Manufacturing; 2013https://www.research-collection.ethz.ch:443/handle/20.500.11850/73989 accessed August 6, 2019.

[24] M. Bäker, How to get meaningful and correct results from your finite element model, (2018). http://arxiv.org/abs/1811.05753 (accessed August 6, 2019).

[25] Elabbasi N, Hong J-W, Bathe K-J. On the Reliable Solution of Contact Problems in Engineering Design. Int J Mech Mater Des 2004;1:3–16. https://doi.org/10.1023/B:MAMD.0000035458.72478.7a.

[26] Solhjoo S. Determination of flow stress and the critical strain for the onset of dynamic recrystallization using a hyperbolic tangent function. Mater Des 2014;54:390–3. https://doi.org/10.1016/j.matdes.2013.08.055.

[27] Solhjoo S. Determination of critical strain for initiation of dynamic recrystallization. Mater Des 2010;31:1360–4. https://doi.org/10.1016/j.matdes.2009.09.001.

[28] Solhjoo S. Analysis of flow stress up to the peak at hot deformation. Mater Des 2009;30:3036–40. https://doi.org/10.1016/j.matdes.2008.12.014.

[29] Wilson DV. Representation of material behaviour in finite element methods of modelling sheet forming processes. Int Mater Rev 1990;35:329–48. https://doi.org/10.1179/095066090790323948.

[30] Grillet AM, Bogaerds ACB, Peters GWM, Baaijens FPT, Bulters M. Numerical analysis of flow mark surface defects in injection molding flow. J Rheol. (N Y N Y). 2002;46:651–69. https://doi.org/10.1122/1.1459419.

[31] Baaijens FPT. Calculation of residual stresses in injection molded products. Rheol Acta 1991;30:284–99. https://doi.org/10.1007/BF00366642.

[32] Konter A, Post J, Lindgren L-E, Jansson T, San Martín D, Somers M. Prediction of stainless steel performance after forming and finishing (PRESSPERFECT). EU publications; 2015. https://doi.org/10.2777/721279.

[33] Li H, Zhao G, He L, Mo Y. Solution of non-linear thermal transient problems by a new adaptive time-step method in quenching process. Appl Math Model 2009;33:329–42. https://doi.org/10.1016/J.APM.2007.11.004.

[34] Valli AMP, Carey GF, Coutinho ALGA. Control strategies for timestep selection in finite element simulation of incompressible flows and coupled reaction-convection-diffusion processes. Int J Numer Methods Fluids 2005;47:201–31. https://doi.org/10.1002/fld.805.

[35] O'Sullivan C, Bray JD. Selecting a suitable time step for discrete element simulations that use the central difference time integration scheme. Eng Comput 2004;21:278–303. https://doi.org/10.1108/02644400410519794.

[36] Krieg RD, Krieg DB. Accuracies of numerical solution methods for the elastic-perfectly plastic model. J Press Vessel Technol Trans ASME 1977;99:510–5. https://doi.org/10.1115/1.3454568.

[37] Guo X, Post J, Bleck W, Groen M. Delayed cracking in meta-stable austenitic stainless steels and its prediction by FEM simulation. Stainless steel world conference & exhibition 2009, November 10-12th, Maastricht, the Netherlands. KCI Publishing; 2009http://publications.rwth-aachen.de/record/100242.

[38] Zijlstra G, Groen M, Post J, Ocelík V, De Hosson JTM. On the role of the residual stress state in product manufacturing. Mater Des 2016;105:375–80. https://doi.org/10.1016/J.MATDES.2016.05.085.

[39] C.O.- Huizer, Computational analysis of cell orientation in response to mechanical stimuli : implications for myocardial repair strategies, (2014). doi:10.6100/

IR780939.

[40] Metal Injection Molding | Powdered Metallurgy | MIM vs PM, https://www.optimim.com/blog-metal-injection-molding-powdered-metallurgy (accessed July 10, 2019).

[41] Blaine DC. The micromechanical influences on the constitutive laws of sintering for continuum model. The Pennsylvania State University; 2004.

[42] Olevsky EA. Theory of sintering: from discrete to continuum. Mater Sci Eng R Rep. 1998;23:41–100. https://doi.org/10.1016/S0927-796X(98)00009-6.

[43] Paradis P-F, Ishikawa T, Fujii R, Yoda S. Thermophysical properties of molten tungsten measured with an electrostatic levitator. Heat Transf Res 2006;35:152–64. https://doi.org/10.1002/htj.20101.

[44] Bogacki P, Shampine LF. A 3(2) pair of Runge - Kutta formulas. Appl Math Lett 1989;2:321–5. https://doi.org/10.1016/0893-9659(89)90079-7.

[45] Shampine LF, Reichelt MW. The MATLAB ODE suite. SIAM J Sci Comput 1997;18:1–22. https://doi.org/10.1137/S1064827594276424.

[46] Find minimum of unconstrained multivariable function using derivative-free method, https://nl.mathworks.com/help/matlab/ref/fminsearch.html (accessed October 14, 2019).

[47] Lagarias JC, Lagarias JC, Reeds JA, Wright MH, Wright PE. Convergence properties of the Nelder-Mead simplex method in low dimensions. SIAM J Optim 1998;9:112–47http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.120.6062 accessed October 14, 2019.