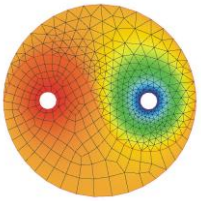
Publishing House
AKAPIT

COMPUTER METHODS IN MATERIALS SCIENCE
Informatyka w Technologii Materiałów

Vol. 18, 2018, No. 2



RULE-BASED CONTROLLING OF A MULTISCALE MODEL OF PRECIPITATION KINETICS

PIOTR MACIOŁ*

Faculty of Metals Engineering and Industrial Computer Science,
AGH University of Science and Technology, al. Mickiewicza 30, 30-059 Krakow, Poland
**Corresponding author: pmaciol@agh.edu.pl*

Abstract

One of the most important obstacles of widening of multiscale modelling is its high computational demand. It is caused by the fact, that each of numerous fine scale models has comparable computational requirements to a coarse scale one. There are several ways of decreasing of computational time of multiscale models. Adaptation of a structure of a model is one of the most promising. In this paper the Adaptive Multiscale Modelling Methodology is described, including Knowledge-Based adaptation of the multiscale model of precipitation kinetics during heat treatment. Core features of the methodology are introduced. The numerical model of heat treatment of an aluminium alloy based on the methodology and the dedicated framework is presented. Besides modelling of macroscopic heat transfer, models of precipitation kinetics based on thermodynamic calculations are included. To decrease computational requirements arising from coupling of the macroscale model and the thermodynamic models, metamodeling and similarity approaches are applied. Computations with several configuration of rules are described, as well as their results. Reliability and time consumption of computations are discussed. Future perspectives of combining of modelling and metamodeling in one, integrated model are discussed.

Key words: Multiscale modelling, Precipitation kinetic, Knowledge-based systems, Knowledge-based optimization, Aluminium alloys, Metamodeling

1. INTRODUCTION

The progress of technology is directly connected to knowledge of materials and processes occurring inside materials during their thermal and mechanical treatment, as well as during further heat treatment and product exploitation. In the past, knowledge had been acquired with a long-lasting searching of proper solutions using the try-and-error approach. Later, planned material investigations had played an increasingly important role. During the last half of the century, besides experimental, also computational methods has become important. In the last ten years, several integrated approaches have appeared. One of them is Integrated Computational Materials Engineering (ICME) (Panchal et al., 2013). In this approach, a capability of developing of new materials and new technological processes with multiscale modelling, supported by previous and concurrent experimental research (durability, chemical, microscopic) is required (Schmitz, 2015). The main problem which must be solved before the ICME and similar approaches become a standard tool for the industry is developing of computational models with high predictive capabilities. They must directly address material behaviour in several time and spatial scales and simultaneously require acceptable computational resources.

If only a single scale problem is investigated, a computing time nowadays is not a serious problem. While the complexity of solved problems increases, capabilities of computer architectures increase as well. Contrary, if multiscale modelling is involved, especially fully coupled or with simultaneous computations in several scales, computing time increase much faster (qualitatively exponential). There are

two main ways to remedy this issue. The first one is decoupling of a multiscale model into several, single-scale ones with the fine scale models used to develop a function or a so-called metamodel of material properties and the coarse scale one, using this function during computations. This approach significantly reduces required computational resources. However, it introduces an additional approximation error, possibly significant. The second way is to use a fine scale model only when and where it is necessary. Presently, identification of critical regions must be performed by a researcher during configuration of the simulation. From the point of view of reliability and efficiency, the obtained configuration is usually far from optimal one. Moreover, it is extremely difficult to keep a robust configuration of a multiscale model when the problem is non-stationay. The problem becomes even more important since ICME and similar approaches are expected to be used also by engineers, not only academic researchers. Currently, designers of multiscale models (researchers) are the only users of the models. In future, while the models will be still developed by researchers, they will be used by engineers. Hence, processes of designing and exploitation should not be iteratively coupled.

An adaptation-capable framework for multiscale modelling had been developed by Delalondre et al. (Delalondre et al., 2010). The system aids adaptive multi-model modelling. The framework supports mainly the design of multi-model systems. Adaptivity is based on a'posteriori error evaluation and simple rules in Prolog language. This framework provides capabilities of an on-line adaptation; however, it is dedicated rather for design-time adaptation and run-time adaptation capabilities are limited.

More recently several frameworks for multiscale/heterogeneous modelling have been developed. In (da Silveira et al., 2012), the object-oriented architecture for structural design software is presented. The framework for multiscale modelling, introduced in (Chopard et al., 2014) significantly simplifies design process of a multiscale model. It introduces a semantic description on different abstraction and scale levels with the Multiscale Modelling Language (MML). In (Shephard et al., 2013), the need of eliminating technical impediments with predefined workflows and supporting frameworks are discussed, following advices of the US Council on Competitiveness. The mentioned frameworks realize the overall concept of standardization of numerical modelling, however, they do not approach multiscale models but rather models with a multi-level abstraction. All referenced frameworks, as well are several others, focus on simplifying of design process, including mapping of information flow. They all lack capabilities of run-time manipulation of a structure of a multiscale model. Such ability was introduced into the framework presented in (Biyikli & To, 2016). However, available rules are very simple and the whole system is fitted to the single configuration and the single domain.

Multiscale models are complex and some compromises between expected reliability and computational requirements are necessary. Hence, supporting a design process is important and the frameworks introduced above can be a helpful tools, decreasing time necessary to develop multiscale models. The main drawback of the design-supporting frameworks is that they relay on a data known before running simulation. It frequently does not allow a proper identification of most important domains. Moreover, a designer must be aware of all possible phenomena occurring in a modelled process, which is sometimes not assured. Finally, redesigning of simulation requires researcher's attention, what is troublesome when multiply, similar simulations are ran (e.g. during optimization, computations governed by design of experiments approaches etc.). An alternative approach, eliminating these disadvantages is automatic, runtime controlling of a configuration of a multiscale model. Design of such a solution was proposed by the author in (Macioł et al., 2013; Macioł et al., 2012a). In these papers, the Adaptive Multiscale Modelling Methodology (AM3) framework is introduced. The approach bases on capability of switching between models representing fine scale processes (called submodels), according to a local state of modelled process, expected reliability and available computational resources. The ultimate goal of this approach is to provide a framework for multiscale modelling, simplifying design and run-time controlling, able to exploit a specificity of optimization with numerical modelling. In the previous publications, authors presented some elements of the AM3 framework: design (Macioł et al., 2013; Macioł et al., 2012a), simple case studies (Macioł et al., 2014; Macioł et al., 2015), knowledge representation (Macioł et al., 2017a; Macioł & Regulski, 2016), design-phase support (Macioł & Michalik, 2018) and development of metamodels (Macioł et al., 2018).

In this paper, the case study combining all mentioned components is presented. The ability to control a balance between reliability and resources con-

sumption is discussed. Finally, weaknesses of the approach, as well as possible remedies, are presented.

2. ADAPTIVE MULTISCALE MODELLING METHODOLOGY (AM3)

2.1 Rationale

The goal of the proposed approach is to reduce the time necessary to conduct a multiscale numerical analysis, keeping reliability on a sufficient (not necessarily the highest) level. The source of the considerable computational requirements of the multiscale models is the necessity of running fine scale submodels for numerous coarse scale calculation points. The classic approach to shortening the calculation time is analogous to the one applied for single scale models. It is based on (i) choosing the best algorithms available and their efficient implementation, (ii) reducing the accuracy (through more scattered calculation points or by neglecting some phenomena) and (iii) parallel computing with an HPC environment. The first has clear benefits, but the possibilities of its use are limited. Most of the currently used numerical models are created with deep awareness of the need to provide optimal algorithms and their correct implementation. Hence, in most cases, attempts to follow this route do not allow a significant reduction in computing time. Using of two further methods has some inevitable negative consequences. Reducing accuracy of calculations by using fewer calculation points or by neglecting potentially important phenomena can lead to a significant reduction of computing time, however, if used carelessly, may lead to unreliable results. A typical example may be neglecting of dynamic recrystallization during hot forming. Parallelization of calculations has two major limitations. First, many algorithms are difficult to parallelize and necessary synchronization of calculations causes additional overheads increasing with the number of processes/threads. Secondly, it is still necessary to use appropriate computational resources. Although the current increase of the availability of HPC resources is impressive, cost of acquiring and exploiting them is important, especially for commercial applications. Analogous arguments refer to a use of more powerful computers (not necessarily parallel). First, the increase of capabilities of computers is limited (most of the efforts are located in the area of parallel computing), and secondly costs of acquiring and operating such computers is also high.

The solution proposed in this paper is based on an alternative approach - limiting a computational complexity of a multiscale model to a minimum necessary to obtain a sufficient accuracy and reliability in a given situation. In a case of calculations for one, uniquely defined system, the need to minimize the demand for computing power is to run accurate models only there (for these coarse scale points), where it is necessary (Macioł et al., 2013; Macioł et al.,, 2012a). When a multiscale model is used inside an optimization process, it is possible to take additional estimates of a current solution's distance from a target one. While a current solution is far from optimum, an accuracy of a solution might be safely decreased (in a reasonable degree).

Reassuming, the key issue is to provide an environment supporting a choice of submodels, basing on predefined knowledge and a current state of a model and supporting switching between various fine scale models in runtime. Presented approach also addresses the need of separating of design and exploitation phases, mentioned in introduction.

2.2 Architecture and methodology

It was assumed that AM3 should support developing of computationally effective and flexible multiscale models. Furthermore, it is expected that AM3 will simplify a design process. It is commonly recognized that efficiency and flexibility are contradictory requirements. Hence, a non-trivial, sophisticated architecture is necessary. The most important issues are:

- separating of knowledge management and a modelling implementation,
- low-level validation of correctness of a multiscale model configuration,
- minimal computational overheads on the framework.

To address these challenges two levels of abstraction are considered in AM3: (i) the abstract 'phenomena' and (ii) the concretized modules. A phenomenon is a chemical or physical process potentially occurring during a simulation; it does not provide any information about how the process is modelled. It is represented by a set of output variables, qualitatively describing the phenomenon. A concretized module is a working numerical entity used to simulate a phenomenon. It must provide at least the same outputs as a modelled phenomenon and define input parameters necessary for a numerical solution. All input parameters must be provided by other modules. Several numerical submodels can

be implemented within a single module, as far as they jointly return the required set of output variables.

AM3's ability to reconfigure models is based on its Object-Oriented design and the Generic Programming paradigm. In each point in time and space, the most suitable submodel is chosen according to a set of rules defined within the Knowledge Base System (KBS). Criteria of suitability can be for example accuracy, reliability, performance, computational requirements or any combination of them. The KBS continuously analyses a configuration of a model and chooses the most suitable submodel for each phenomenon. Detailed description is given in (Macioł & Michalik, 2018).

Knowledge management is based on a combination of KBS, utilizing First Order Logic (FOL) (Macioł et al., 2012b) and semantic description of a model and possible phenomena (Descriptive Logic, DL) (Macioł et al., 2017a; Macioł & Regulski, 2016). Validation of a structure of a multiscale model is based on metaprogramming mechanisms, implemented with the C++ language (Macioł & Michalik, 2018). Computational efficiency is obtained with application of advanced C++ language as the main programming platform and parallelization based on MPI (Macioł & Michalik, 2016). The framework had been designed to combine submodels developed with external modelling platforms (Abaqus, Deform, MatCalc etc.) with in-house code (Internal Variable submodels, metamodels, etc.).

In the previous works, all mentioned components has been developed and tested independently. In this paper, the result of integration of all components together is presented.

Methodology. Two aspects of the methodology can be distinguished: a pre-simulation development of submodels and a during-simulation choice of proper submodels. The first step of the pre-simulation stage is an analysis of a given problem and an identification of possible phenomena (for example precipitation in precipitation hardening, dynamic recrystallization in hot forming etc.). Next, at least one submodel for each of possible phenomena must be provided. If the submodels has the limited ranges of application (e.g. maximal strain rate or temperature), other submodels for the same phenomena must be provided to cover the whole possible range of state variables. Moreover, computational requirements must be taken into consideration. Numerous runs of fine-scale submodels may consume large amounts of computational resources. Hence, if it possible to provide a more efficient submodel, even with limited range of application, an overall computational time would be probably decreased. In this paper, two approaches to developing fast submodels are presented – metamodeling and similarity-based submodels.

Finally, an AM3-based multiscale model consist of a set of submodels, covering all possible phenomena in a whole range of possible states of a modelled process. The last necessary component is a decision-making component. In the proposed approach, the external KBS is applied as a reasoning engine. Contrary, a knowledge base is a part of a model. Design of a knowledge base is the key to the overall quality of a AM3-based multiscale model and governs behavior of a model during a simulation. Part of knowledge is relatively easy to design – it includes mainly obvious relations, like for example submodel of plasticity X includes precipitation hardening. The other important part of knowledge controls balance between reliability and efficiency and is more difficult to be clearly defined. The discussion of the influence of a knowledge design on the overall model quality is discussed in the further part of the paper.

During simulation, a classical multiscale model uses statically predefined fine-scale submodels to calculate variables describing a state of a material. Contrary, an AM3-based simulation in each time step, for each computational point of a coarse scale submodel, asks the KBS which fine-scale submodel should be used to calculate a particular variable. Typically for decision-making systems, employed KBS represents reasoning mechanisms defined for a chosen logical system, hence a model's designer controls its behavior with a proper definition of knowledge in the pre-simulation stage.

2.3 Knowledge representation

Managing of a complex system, constrained with internal compatibility of submodels, requires application of a proper tool. As it was introduced above, Descriptive Logic was chosen for this purpose. Web Ontology Language 2.0 (OWL2) is used to represent multiscale models. A prototype of such description is presented in (Macioł & Regulski, 2016). In last two years, a new approach to semantic description of materials and models arises from the work of European Materials Modelling Council (EMMC) (Ghedini et al., 2018). Currently, European Materials Modelling Ontology (EMMO) is still not completed and released.

Since a DL-based description is not suitable for on-line decision making (Macioł et al., 2017a), a FOL-compliant representation of rules governing choosing of submodels has been developed. Automatic generation of FOL-compliant rules directly from a DL-based description is not possible currently. Hence, compatibility of rules and the ontology must be ensured by a designer.

The fragment of the current semantic description of the AM3-compliant problem is shown in figure 1. It is inspired by the current state of the EMMO. The diagram illustrates the part of hierarchy of concepts (classes) and relations between them. For example, the class AlCrFeMnSi_a is a Phase may exist as a Precipitation in 6082 in the T1 Temper; 6082 is an 6xxx series Aluminium Alloy.

AlCrFeMnSi_a_p0-p3 are so-called 'instances', representing particular type of potentially existing precipitations. The types are identical from the thermodynamic point of view, but precipitates during different stages of a technological process (during casting, during homogenisation etc.).

The rules presented later in the paper are compatible with the ontology, however, as mentioned above, the verification was conducted manually.

3. THE CASE STUDY

The typical case introduced here follows up the author's previous works (Macioł et al., 2018; Macioł et al., 2015). The AM3 multiscale model for Thermo-Mechanical-Treatment (TMT) of metallic alloys was presented. It is composed of two main components: the macroscale module for heat flux during thermo-mechanical treatment and the microstructure module in charge of computing microscopic mechanisms. In this work, annealing of the commercial grade aluminium alloy 6082 is modelled. Temperature of a billet is controlled with resistive heating and enforced gas flow. During annealing, evolution of precipitations is expected. The external materials calculator MatCalc (Kozeschnik, 2012; Kozeschnik et al., 2004) is integrated in the multiscale model as a microscopic submodel that computes precipitation kinetics of second-phase particles. Besides, the metamodel of the precipitation kinetics was developed and integrated with the multiscale model (Macioł et al., 2018).

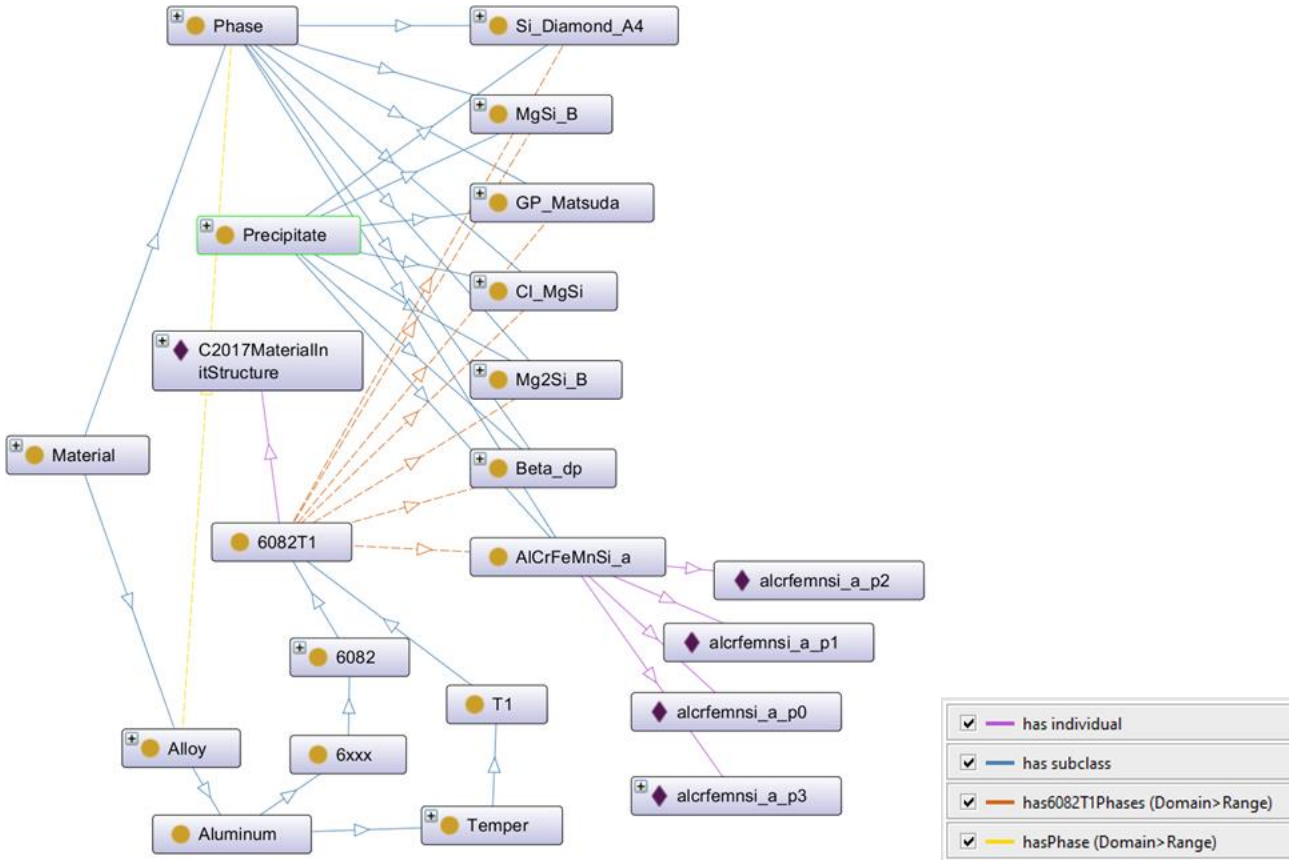


Fig. 1. *Part of the ontology describing the multiscale model of precipitation kinetics; ontology visualized with OntoGraf.*

Important part of design of a multiscale model is identification of interfaces. Since in the discussed case no deformation occurs, temperature is the only one parameter passed to microscale. A choice of parameters computed in microscale must consider both capabilities of the involved submodels and importance of the possible phenomena. As it was mentioned above, precipitation kinetics is the leading one. Many outputs are available by the MatCalc simulator, of which the number of precipitates and their size are the most relevant. The MatCalc outputs detailed distributions of numbers of precipitates and their diameters, which cannot be reproduced by a metamodel. In this work, the output from the precipitation kinetic model is the aggregated mean precipitate diameter, for all second phases, with a lower limit 1×10^{-8} m imposed on the particle diameter.

3.1 Submodels

Macroscale. Usually, for modelling of processes involving microstructure evolution, theoretical parameters of these processes are applied. For example, temperature is described with such parameters as a constant heating rate, an annealing temperature, a time, etc. In real cases, such parameters are never exact. To take into consideration a realistic temperature history, the macroscopic model of heating, keeping in elevated temperature and cooling of a metal billet is modelled. The dimensions and the boundary conditions are shown in figure 2. The macroscale model is developed with the DEFORM 2D commercial software. Since there is no plastic deformation in the current model, only thermal, axisymmetric solution is applied. Initially, the temperature across the whole billet is 40^{o}C. To reproduce a realistic temperature distribution, resistance heating and enforced cooling were included. The current and heat transfer coefficients were chosen to meet the required temperature curve in the "virtual thermocouple" (figure 2); the identified coefficients are shown in table 1. Current is given on the upper surface of the billet (representing the upper die). Convective heat transfer is applied on external surfaces of the billet. As expected, the temperature values in various points differ from the expected ones (figure 3). The maximal temperature difference at the beginning of the cooling phase exceeds 240oC. So large value is caused by a rapid cooling on a surface of a billet and relatively large time step. The mesh consists of 501 elements and 575 nodes. For each node, in each time step, nodal temperature is passed to the AM3 framework to calculate precipitation kinetics.

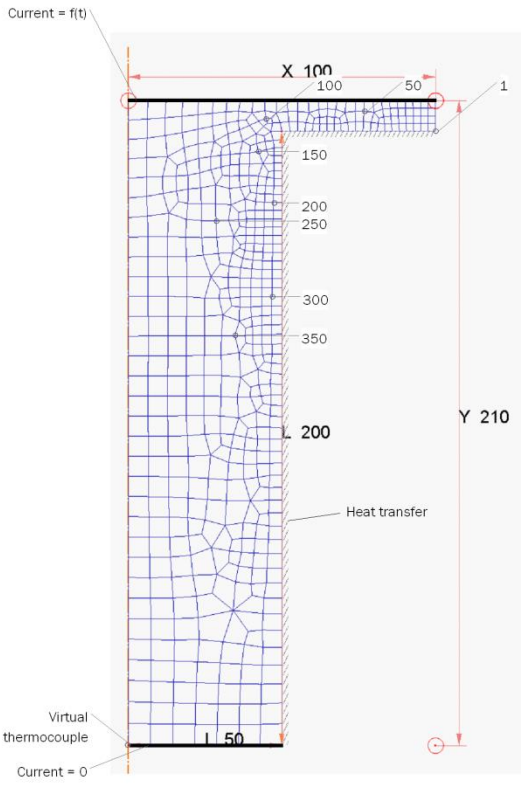


Fig. 2. *Dimensions, selected nodes and boundary conditions.*

Table 1. Parameters of numerical model identified to meet requested temperature profile.

Time, s	Heat transfer coefficient, kW/m^{2}K	Current, A
0.0	0.02	5.412
1350	0.02	5.412
1351	0.02	3.548
5276	0.02	3.548
5277	1.373	0.0
5824	1.373	0.0

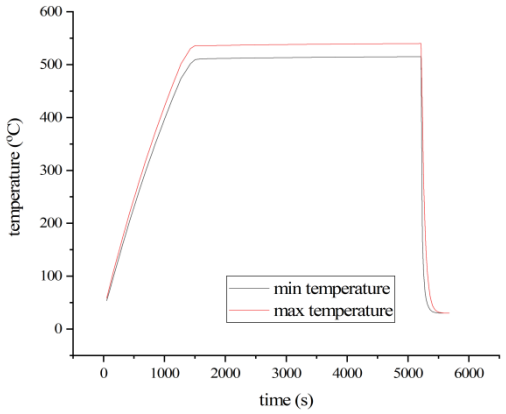


Fig. 3. Lower and upper bounds of temperatures, °C.

Microscale. The precipitation kinetics is the phenomenon modelled in the microscale. Currently the classical. The most reliable approach to modelling of precipitation kinetics is thermodynamic computation, based on CALPHAD databases. The MatCalc, a scientific software toolbox for computer simulation of constrained and unconstrained phase equilibria and multi-component and multi-phase precipitation kinetics is applied. The scientific background of the software is discussed in (Kozeschnik, 2012). The MatCalc software includes CALPHAD-type thermodynamic database for aluminium alloys. It contains data for both, stable and metastable phases. While results are in general reliable, the computational requirements of MatCalc simulations limit its application in multiscale modelling, especially for fully coupled models, when precipitation kinetics must be computed step-by-step. Computational efficiency of the multiscale model might be improved with several measures. First is parallelization, however a large number of fine scale models still requires very large resources (Macioł & Michalik, 2016). Other two possibilities are replacing of MatCalc submodels with metamodels (Macioł et al., 2018; Macioł et al., 2017b) or with results of "similar" MatCalc computations (Macioł et al., 2013).

MatCalc model. The MatCalc has a proved and successful history of applications, both for basic and applied research. However, incorporating of a MatCalc model into a multiscale model has some disadvantages. The main issue is that it had not been designed for on-line collaboration, with step-to-step information passing. The multiscale environment requires concurrent existence of numerous instances of MatCalc, which is not possible with typical use as an embedded library. There is no straightforward way to suspend a simulation and wait for next timestep data. Finally, running a simulation with short timesteps is inefficient

All these issues had to be solved before developing of the multiscale model including a MatCalc submodel. The concurrency issue was solved by enveloping MatCalc instances with MPI protocol. The suspending issue was solved with the suitable, however complex, configuration of a simulation. Finally, inefficiency of short time-steps was partially solved with the ability to switch between a MatCalc and a metamodel.

The MatCalc submodel is adapted from the previous works (Macioł et al., 2014; Macioł et al., 2015). The core of the submodel is the definition of the initial microstructure (the distribution of the precipitating phases), representing the state inherited from previous processing and the definition of the available phases, as well as the characteristic localizations of precipitations (grain boundaries, grain interiors). The third part of the original model, controlling temperature history, was removed and replaced by the interface to the modelling framework.

Metamodel. Metamodeling employs "a model of a model" (Kusiak et al., 2015). Typically, Machine Learning algorithms are used to develop a "black-box" representation of a model. While learned, computational costs of a metamodel answer is neglectable, at least when compared to costs of a model response. In discussed case, the Kriging method was used to develop the metamodel of the MatCalc model of precipitation kinetics. Thorough discussion is included in (Macioł et al., 2018).

From the point of view of a multiscale model, the most important aspect is that a metamodel is significantly less flexible than a model. A cardinality of input variables of a model (in this case the one based on the MatCalc), as well as domains of all input variables, are practically infinite. Developing of a metamodel covering a whole range of input variables is not possible, also with modern machine learning approaches. Hence, a metamodel can be developed only for a significantly restricted number of input variables and restricted domains for all in-

put variables of a metamodel. It must be emphasized that even if a metamodel has been successfully trained for a given range of input parameters, it has no capabilities for extrapolation. In discussed case, the input of the metamodel of precipitation kinetics is restricted to four parameters (table 2).

Table 2. Data used for material description in numerical modeling.

Input parameter	Symbol	Lower bound	Upper bound
annealing temperature, °C	T_i	250	550
heating rate, °C/s	h_r	0.02	0.5
holding time, s	t_i	1000	14000
cooling rate, °C/s	c_r	0.01	2

Training of a metamodel requires prior computing of model outputs for chosen combinations of values of input parameters. For some of requested combinations, computational times of the MatCalc model was extremally long. To limit overall time necessary to generate data for training the metamodel, some combinations of input values were removed from the training set. Hence, the developed metamodel cannot be applied for the whole domain defined in table 2. The definition of the boundaries of the approved range of the input variables are discussed in the section Rules, while more detailed discussion can be found in (Macioł et al., 2018).

Similarity. It can be noticed that, however the upper and lower boundaries of temperatures during the process significantly differs, there are still many nodes with very similar temperature histories. While the metamodel do not require noticeable computational time to return the requested value, the MatCalc submodel does. Hence, using this submodel for nodes with very similar histories is an unacceptable waste of resources. To minimize this effect, a "similarity" concept was introduced (Macioł et al., 2013). Replacing of costly computations with results already existing is very effective, nevertheless, it requires a careful choice of rules of similarity. In the discussed case, the measure of similarity (distance) is the sum of Euclidean distances between temperatures obtained in the same time step. Distance between two histories with the same length can be computed as:

$$dist(m,n) = \sum_{i=0}^{current\ timestep} \sqrt{\left(T_i^m - T_i^n\right)^2} \qquad (1)$$

where: $dist(m,n)$ – distance between m-th and n-th temperature history, T_i^m – temperature in °C, m-th temperature history, i-th timestep.

The set of submodels "similar" to the m-th submodel S contains submodels with the smallest distance to the n-th submodel:

$$S^m = arg\,min_{n\in N, n\neq m} dist(m,n) \qquad (2)$$

where: N – set of all temperature histories.

In the case when there is more than one element in the set Sm, the one with the closest node number to the examined one is chosen. However, while distance is a real number, this situation is neglectable.

3.2 Switching between submodels

In the discussed case, three submodels can be used to calculate the same output value (the MatCalc submodel, the metamodel, the similar submodel). An important feature of the AM3 framework is the capability of switching between submodels in runtime. That requires the ability to pass a material state between submodels. The metamodel is stateless (an output value is calculated only with a current state of input variables), hence switching to it does not require any other actions. The similar submodel is "passively" state-aware. The rules governing switching to similar submodel refer to history, but when the reasoning system chooses it, no further actions on historical values are required. Contrary, a MatCalc submodel requires whole history to compute valid outputs. If a MatCalc submodel is started at the beginning of a multiscale simulation, its state is computed from step to step. However, if a similar submodel or a metamodel have been initially used and a MatCalc submodel is started afterwards, a current state must be computed. One can expect that this algorithm would neglect benefits of metamodeling. However, as it was mentioned above, running a MatCalc submodel step-by-step usually leads to an increase of computational time. In the situation discussed here, it is possible to run a MatCalc submodel not step-by-step, but with whole previous history aggregated into up to three steps (a heating, an annealing, a cooling steps). The ability to replace several coarse-scale time steps with a single MatCalc step will save computational time due to more effective use of the MatCalc.

3.3 Rules

Several steps of the AM3 algorithm require reasoning. Some preliminary requirements must be verified during compilation. In this step, mutual compatibility of inputs and outputs for all submodels must be verified. This process is conducted by metaprogramming and is described in (Macioł & Michalik, 2018). The rules are currently hard-coded, however, they are expected to be replaced with an automatic generator (Macioł et al., 2017b). During runtime, decisions to be made are a choice of submodels for an each coarse scale node in actual state of a model. These decisions are made by the reasoning engine, basing on a set of rules. Typical rules are shown in figure 4. To increase readability the format of the rules is modified and some of precludes are omitted.

IF homogenization temperature <= 325 THEN metamodel_available = TRUE

IF homogenization temperature <= 450 AND cooling rate <= -0.08 AND homogenization temperature > 375 THEN metamodel_available = TRUE

IF metamodel_available AND required_reliability <= metamodel_reliability TRUE submodel = metamodel

IF metamodel_not_available AND similar_model_available AND required_reliability <= similar_model_reliability TRUE submodel = similar

IF metamodel_not_available AND similar_model_available AND required_reliability > similar_model_reliability TRUE submodel = MatCalc_model

IF history_temperature_difference(Node_current, Any_node) <= threshold_similarity THEN similar_model_available = TRUE

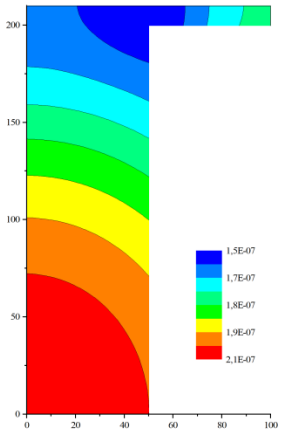
Fig. 4. *Typical rules, representing the range of applicability of the metamodel (rules 1-2) and governing a choice of the best available submodel (3-6).*

The first two rules represent the range of applicability of the metamodel. These rules were generated with the ID3 algorithm during development of the metamodel (Macioł et al., 2018). The remaining rules govern a choice of the best available submodel, fulfilling required reliability.

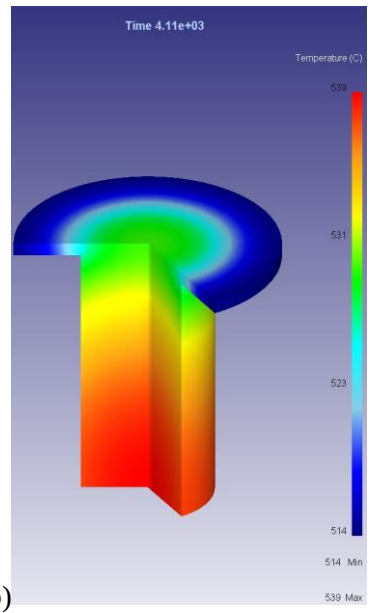
Due to specific of the discussed problem, evaluation of logical values is not sufficient for reasoning. In the presented example, the function history_temperature_difference(Node_current,Any_node) represents the smallest distance (equation (1)) between an investigated node and each other node with the history length equal to an investigated one (in other words, in each time step only nodes computed prior to an investigated one are taken into consideration). Although in theory it is possible to express this condition with logical statements, due to extended expressivity of the employed REBIT system, much more effective procedural functions returning real numbers, are possible.

4. RESULTS

The distribution of mean precipitation diameter in the last step is presented in figure 5a. The temperature distribution is shown in figure 5b. The lower and upper temperature boundaries are shown in figure 2. Rules governing a choice of a submodel refer to further process parameters. They are a heating rate, an annealing time, an annealing temperature and a cooling rate. Upper and lower limits of these variables are shown in figure 6.

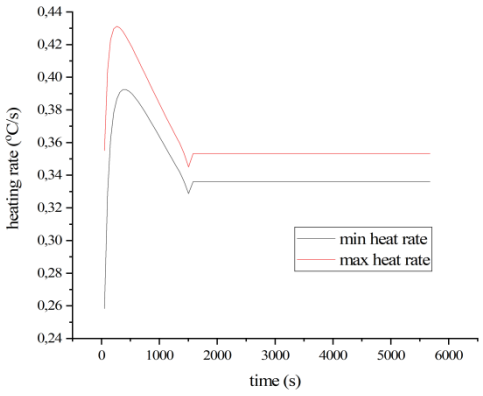


a)

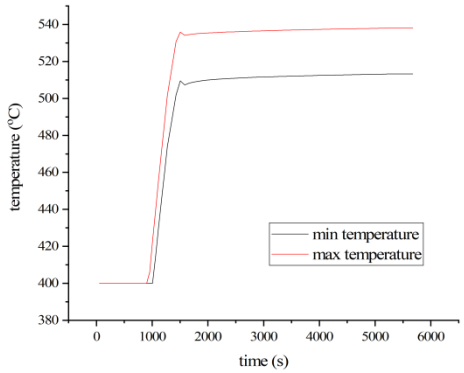


b)

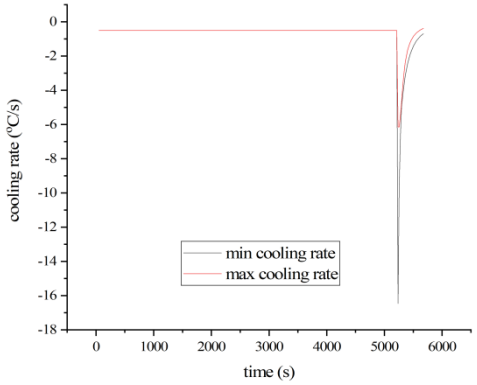
Fig. 5. *Mean precipitation diameter [m] (a) temperature distribution (b).*



a)



b)



c)

Fig. 6. *Upper and lower limits of heating rate (a), annealing temperature (b) and cooling rate(c).*

The most important result of the described work is the comparison of computational times and reliabilities of results. Since the goal of this work is to show the modelling methodology, not to model a particular process, we assume that MatCalc computations are absolutely reliable and valid. To show dependencies between an used submodel and a computational time, the synthetic test with enforced switching between submodels were prepared.

The switching rules, computation times, as well as the numbers of calls of the MatCalc submodels are shown in table 3 (results for node 1). The "first phase" calls are triggered when the heat rate constrain (heating phase) is exceeded and the "second phase" calls are triggered when the cooling rate constrain (beginning of the cooling phase) is exceeded. The distribution of the MatCalc calls is shown also in figure 7. The first obvious conclusion is that using a MatCalc submodel for each time step is significantly more resource-consuming than using only the metamodel (382 seconds vs. 1.67 seconds). More detailed analysis of the dependency between number of calls and computational time is more complex. The reason is that MatCalc computational time of the MatCalc is difficult to predict (it is strongly nonlinear and clear rules were not recognized). However, it seems that overall computational time of the MatCalc submodel depends more on current process conditions than on the number of calls.

Results obtained for all cases listed in table 3 are shown in figure 8. Some differences are clearly visible (figure 9). Firstly, between time 530 s and 1428 s (temperatures circa 300°C to 530°C) the relative metamodeling error is significant, however not disqualifying. Secondly, the cases 6 and 7 show that switching to the MatCalc during cooling phase leads to results more different from the reference case 1 than the metamodel prediction. This effect is caused by the simplified process description used to generate the temperature histories for these MatCalc calls. The worst-case error here is 36%. Finally, besides these two ranges, relative metamodeling error is well below 30%.

Table 3. *Switching rules, computation times and numbers of MatCalc calls for the investigated cases.*

Case	MatCalc model IF	Time, s	MatCalc calls	MatCalc calls in the first phase	MatCalc calls in the second phase
1	Always	381.9	98	77	21
2	Never	1.7	0	0	0
3	$h_r < 0.37$	55.6	14	14	0
4	$h_r < 0.38$	15.9	9	9	0
5	$h_r < 0.39$	18.1	4	4	0
6	$c_r < 2$	31.7	7	0	7
7	$c_r < 6$	28.4	3	0	3

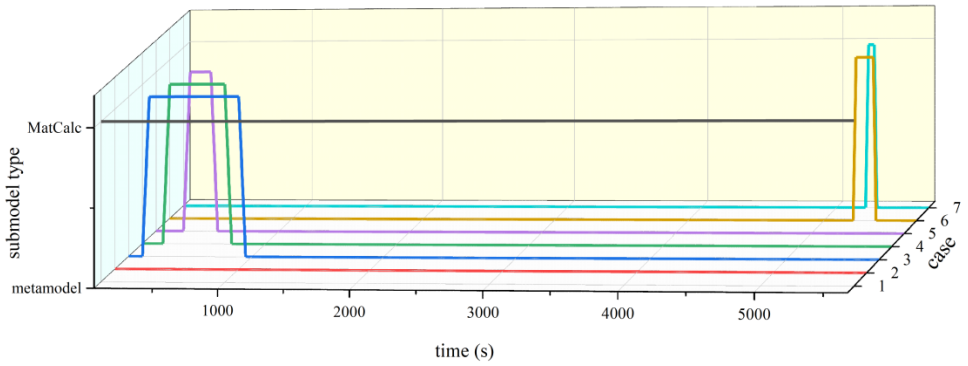


Fig. 7. *Calls of MatCalc submodel; elevated value represents the MatCalc submodel call while lowered represents the metamodel call.*

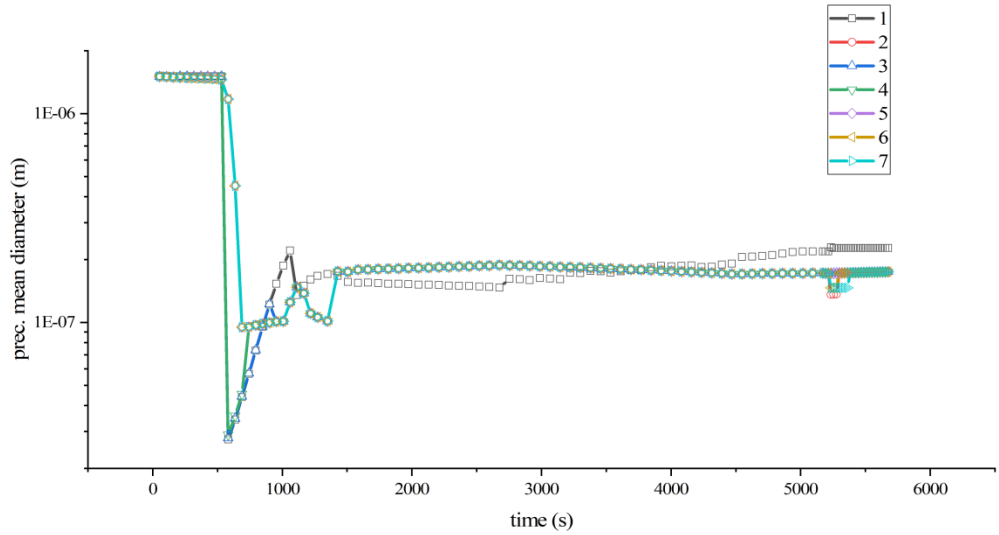


Fig. 8. *Mean precipitation diameter computed with MatCalc or metamodel; the numbers 1-7 refer to the cases described in table 3.*

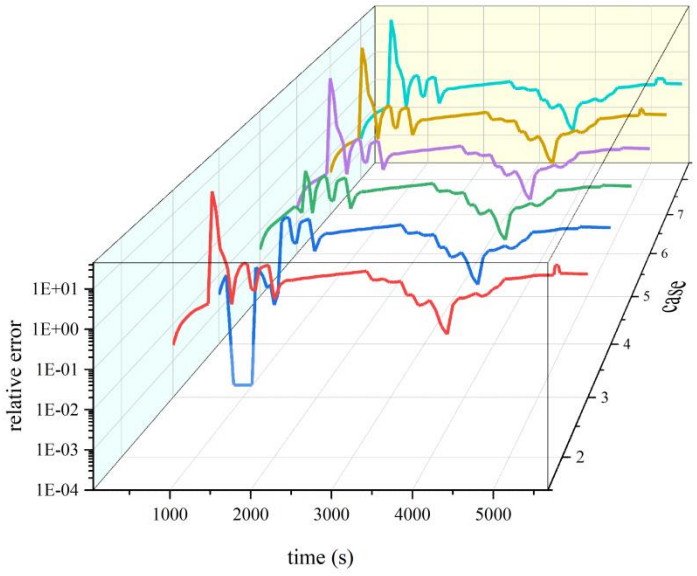


Fig. 9. *Relative error of mean precipitation diameter; the numbers 1-7 refer to the cases described in table 3.*

The second mean to reduce computing requirements is introducing the "similar" submodel. Its influence was tested on the case with 8 chosen nodes (their locations are shown in figure 2). For each node, the knowledge base system choses the submodel type. For each run, different "similarity_threshold" was used. Results are shown in figure 10. Dependency of the computational time on threshold value is difficult to identify. It is clear, that the relation is inversed. However, it is strongly nonlinear and case-dependent.

The "deciding point" is time = 5235 s (the beginning of intensive cooling). Precipitation mean diameters computed by the MatCalc are shown in figure 11 and temperatures at this time are shown in figure 12. Precipitation mean diameters for times between 5172 s and 5298 s, with several different similarity threshold values are shown in figure 13.

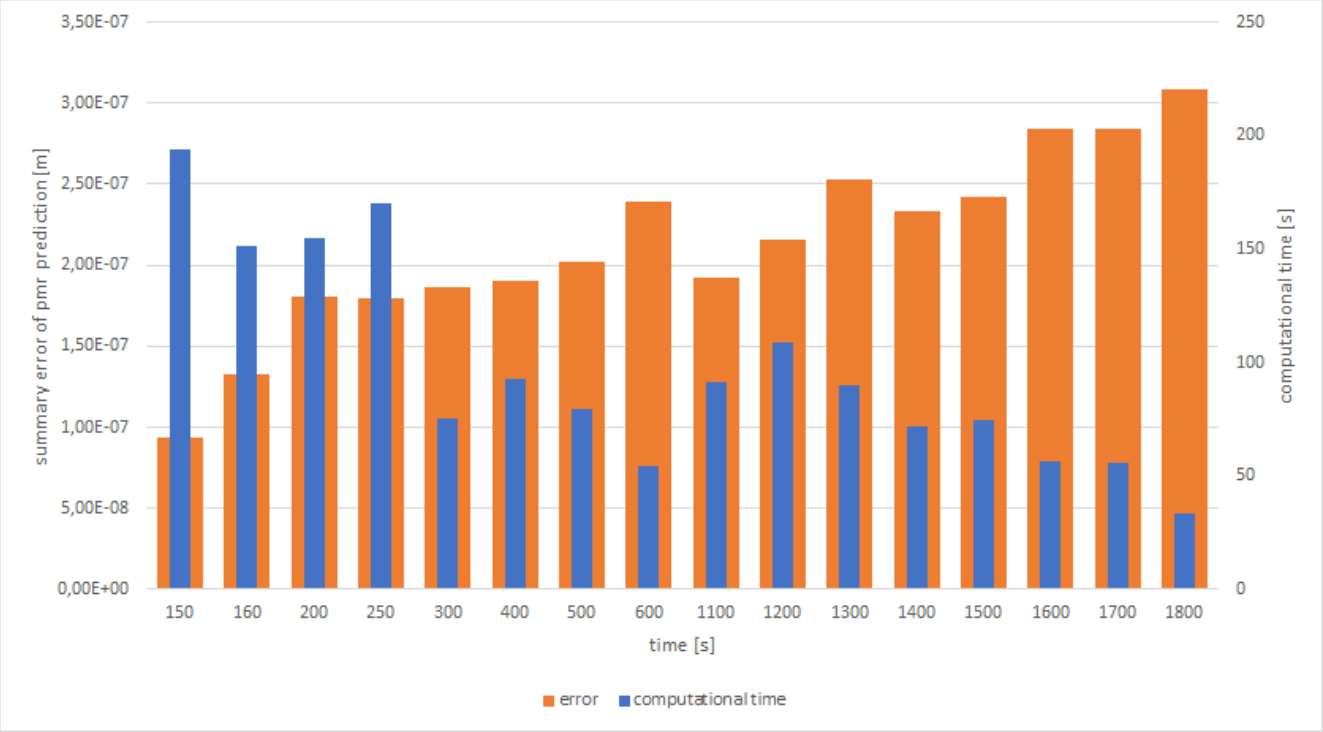


Fig. 10. *Dependency of computational time on similarity threshold.*

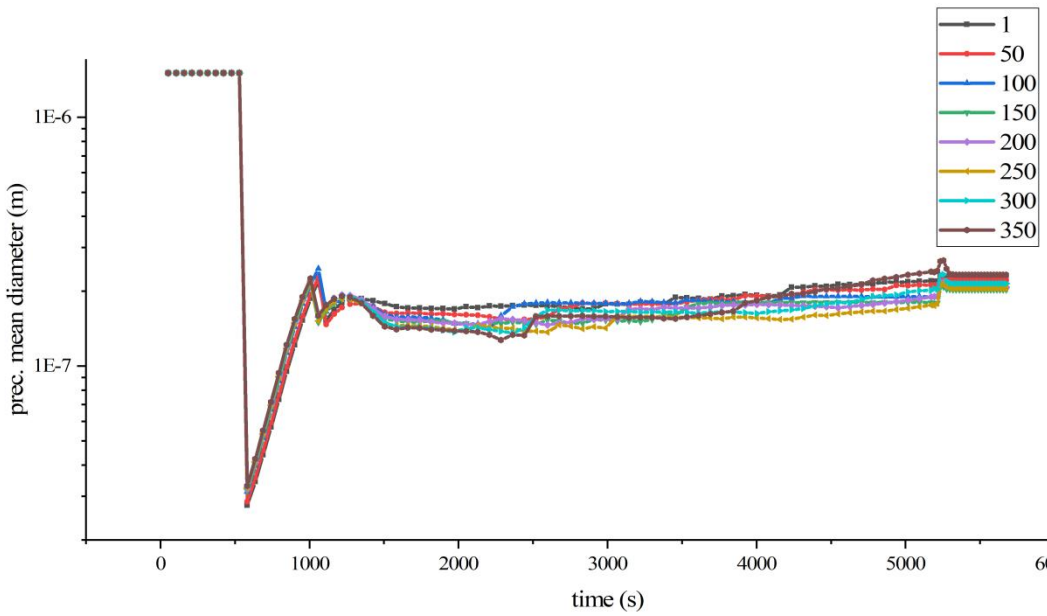


Fig. 11. *Precipitation mean diameters computed for chosen nodes (all MatCalc); the number 1-350 are node numbers (see figure 2).*

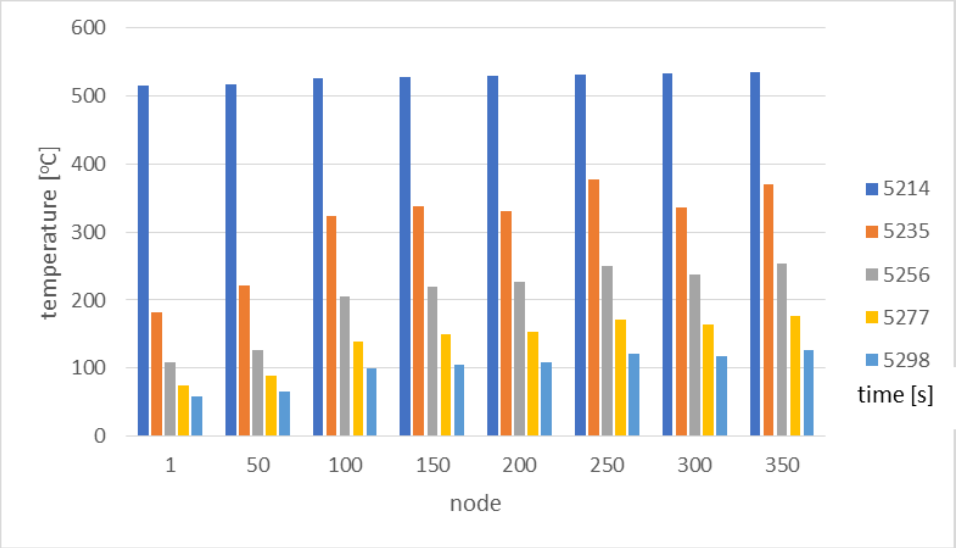


Fig. 12. *Nodal temperatures during "second phase of MatCalc calls" (see table 3).*

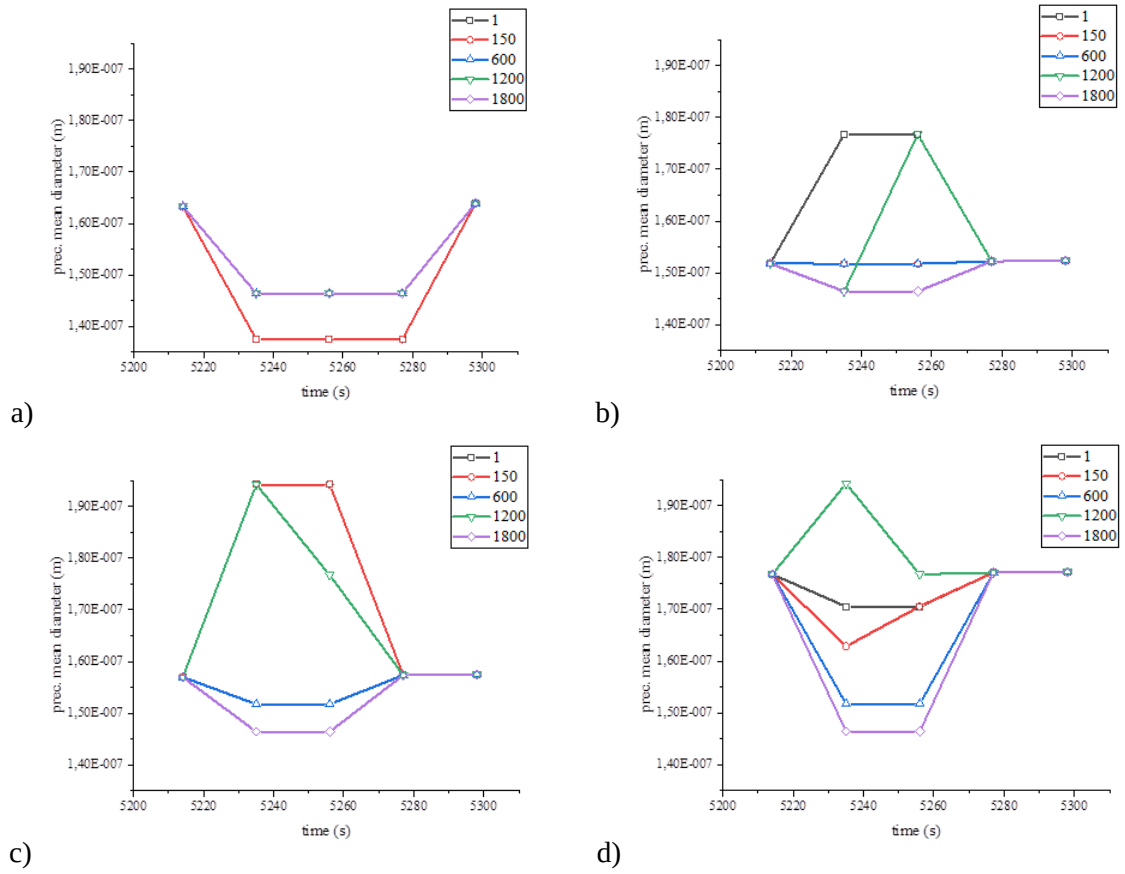


Fig. 13. *Precipitations mean diameter for nodes:50 (a), 150 (b), 200(c), 350(d) (see figure 2).*

5. SUMMARY

Research presented in the paper, as well as in the earlier cited publications, are aimed at developing of the framework for multiscale modelling. The core feature of the framework is the run-time adaptation, based on an autonomous choice of the best fitting submodels. Development of the components of the framework was described in the earlier publications. The goal of this paper was to verify capabilities of the developed solution.

The AM3 framework is aimed at autonomous computations. The role of a researcher is to design a multiscale model, to acquire a knowledge controlling a structure during computations and to provide all submodels. Computations, including reconfiguration of the model, are conducted with no human interaction. This requirement has been positively verified.

An analysis of the speed-up and the introduced metamodeling error is ambiguous. Firstly, the speed-up can be significant (up to 228 times in the analysed case). Contrary, the maximal relative error approaching 4200% is huge (figure 9). However, it must be noticed that such large error appears only in very dynamic processes. In the discussed case, such high values of error appear when the metamodel was late in recognizing the fast drop of mean precipitation diameter when the new phase started precipitating. However, the error rapidly drops when the process stabilizes. It is important that even such a large discrepancy does not influence further results.

Errors introduced by replacing a MatCalc submodel with an already computed, similar solution are significantly smaller (figure 10). When compared with metamodeling, it seems much more promising. In this case, however, at least one instance of MatCalc submodel must be ran for each time step, while the metamodel can completely replace its more time-consuming counterpart.

In our opinion, the usefulness of the rule-based adaptation of multiscale models was proved. The main benefit is ability to control the balance of reliability, computational requirements and computing time. However, the quality of this control is very sensitive on quality of knowledge applied. Faulty knowledge leads to very high computational errors (e.g. when a metamodel is used inadequately), no speed-up (when reliability requirements are too strict) and a poor computational time/reliability balance (e. g. when the similarity condition is used incorrectly, see figure 10, the similarity threshold 600^{o}C).

Reassuming, the crucial issue is knowledge development. Simple, or even naive approach to the development of rules should lead to decreasing of computational time. However, the risk of obtaining erroneous results denies the expected outcome – autonomy of adaptative modelling. Nevertheless, some measures can be taken to improve the success probability. First is careful knowledge development, preceded by computational experiments and followed by suitable tests. The second, important especially in the context of optimization processes, is continuous improvement of both applied submodels and knowledge base.

5.1 Future development

In optimal conditions, the metamodel would be used in all steps and nodes. In the presented case, this goal has not been achieved. The reason is that properties of the heat treatment process considered locally are significantly different from the theoretical values. That led to the too narrow learning set for the metamodel. It should be noticed, that during exploitation of such multiscale model, more and more data computed for different parameters appears and can be stored. These data can be easily used to continuous improving of the range of metamodel's application.

The same data can be used to improve the knowledge base. A continuous analysis of obtained computational results and performance indicators can be used to modify existing switching rules. The most desirable solution is an autonomously learning system, based not only on classical machine learning algorithms, but including also tools generating explicit knowledge.

The other issue observed during research is a large number of rules. Currently their number still allows manual controlling, with the support of auxiliary data management techniques. However, we acknowledge the need of simplifying of the form of the knowledge representation. The solution to be investigated is an application of the fuzzy identification methods. Their main idea is the expansion of the expressiveness of classic IF ... THEN rules, typical for expert rules by treating premises as fuzzy, not crisp facts. Experts are able to determine for each precondition a so-called certainty range, indicating which part of the previously determined range certainly corresponds to the value of the linguistic variable assigned to it.

ACKNOWLEDGEMENTS

This work was supported by the NCN, project No. 2014/15/B/ST8/00187.

MatCalc model of precipitation kinetics was provided by Cecilia Poletti and Romain Bureau, Institute of Materials Science, Joining and Forming, Graz University of Technology. Financial support by the Austrian Federal Government (in particular from Bundesministerium für Verkehr, Innovation und Technologie and Bundesministerium für Wissenschaft, Forschung und Wirtschaft) represented by Sterreichische Forschungsförderungsgesellschaft mbH and the Styrian and the Tyrolean Provincial Government, represented by Steirische Wirtschaftsförderungsgesellschaft mbH and Standortagentur Tirol, within the framework of the COMET Funding Programme is gratefully acknowledged.

REFERENCES

Biyikli, E., To, A.C., 2016, Multiresolution molecular mechanics: Adaptive analysis, *Comput. Methods Appl. Mech. Eng*., 305, 682-702.

Chopard, B., Borgdorff, J., Hoekstra, A.G., 2014, A framework for multi-scale modelling, *Philos. Trans. A. Math. Phys. Eng. Sci*,. 372.

da Silveira, E.S.S., Lages, E.N., Ferreira, F.M.G., 2012, DOOLINES : an object-oriented framework for non-linear static and dynamic analyses of offshore lines., *Eng. Comput*. 28, 149-159.

Delalondre, F., Smith, C., Shephard, M.S., 2010, Collaborative software infrastructure for adaptive multiple model simulation, *Comput. Methods Appl. Mech. Eng*., 199, 1352-1370.

Ghedini, E., Hashibon, A., Friis, J., Goldbeck, G., Schmitz, G., 2018, *EMMO the European Materials Modelling Ontology*, Cambridge.

Kozeschnik, E., 2012, *Modeling solid-state precipitation*, Momentum Press, New York.

Kozeschnik, E., Svoboda, J., Fratzl, P., Fischer, F.D., Fratzl, P., Kozeschnik, E., 2004, Modelling of kinetics in multi-component multi-phase systems with spherical precipitates, *Mater. Sci. Eng. A*, 385, 166-174.

Kusiak, J., Sztangret, Ł., Pietrzyk, M., 2015, Effective strategies of metamodelling of industrial metallurgical processes, *Adv. Eng. Softw*., 89, 90-97.

Macioł, P., Bureau, R., Poletti, C., Sommitsch, C., Warczok, P., Kozeschnik, E., 2015, Agile multiscale modelling of the thermo-mechanical processing of an aluminium alloy, *ESAFORM Conf on Material Forming*, Graz, *Key Eng. Mater*., 651-653, 1319-1324.

Macioł, P., Bureau, R., Sommitsch, C., 2014, An object-oriented analysis of complex numerical models, *Key Eng. Mater*., 611-612, 1356-1363.

Macioł, P., Gotfryd, L., Macioł, A., 2012a, Knowledge based system for runtime controlling of multiscale model of ion-exchange solvent extraction, *ICNAAM, Int. Conf. on Numerical Analysis and Applied Mathematics, Kos, AIP Conf. Proc.*, 125-128.

Macioł, P., Jedrusik, S., Macioł, A., Jędrusik, S., Macioł, A., 2012b, The concept of a rule-based expert system application in multiscale modelling. *Proc.14th Int. Conf. Metal Forming*, Wiley-VCH Verlag GmbH & Co., Kraków, 1327-1330.

Macioł, P., Krumphals, A., Jędrusik, S., Macioł, A., Sommitsch, C., 2013, Rule-based expert system application to optimizing of multiscale model of hot forging and heat treatment of Ti-6Al-4V, *Proc. V Int. Conf. on Coupled Problems*, eds, Idlesohn, S., Papadrakakis, E., Schrefler, B., Ibiza, 1237-1248.

Macioł, P., Macioł, A., Rauch, Ł., 2017a, Ontology dedicated to knowledge-driven optimization for ICME approach, *Proc. 4th World Congr. Integr. Comput. Mater. Eng. ICME 2017*, 113-121.

Macioł, P., Michalik, K., 2016, Parallelization of fine-scale computation in agile multiscale modelling methodology, *19th. ESAFORM Conference on Material Forming, Nantes, AIP Conf. Proc.*, e-book.

Macioł, P., Michalik, K., 2018, Application of metaprogramming and generic programming in multiscale modelling, *Comput. Sci. Eng*., 20(6), 81-94.

Macioł, P., Regulski, K., 2016, Development of semantic description for multiscale models of thermo-mechanical treatment of metal alloys, *JOM, J. Miner. Met. Mater. Soc*., 68(8), 2082-2088.

Macioł, P., Szeliga, D., Sztangret, Ł., 2017b, Substituting of a thermodynamic simulation with a metamodel in the scope of multiscale modeling, *Materials Science Forum, Proc. of THERMEC 2016*, Graz, 1207-1212.

Macioł, P., Szeliga, D., Sztangret, Ł., 2018, Methodology for metamodelling of microstructure evolution: precipitation kinetic case study, *Int. J. Mater. Form*., 11, 867-878.

Panchal, J.H., Kalidindi, S.R., McDowell, D.L., 2013, Key computational modeling issues in Integrated Computational Materials Engineering, *Comput. Des*., 45, 4-25.

Schmitz, G.J., 2015, ICME: Bridging interfaces, *JOM, J. Miner. Met. Mater. Soc*., 68(1), 25-26.

Shephard, M.S., Smith, C., Kolb, J.E., 2013, Bringing HPC to engineering innovation, *Comput. Sci. Eng*., 15(1), 16-25.

STEROWANIE STRUKTURĄ MODELU WIELOSKALOWEGO ZA POMOCĄ SYSTEMU REGUŁOWEGO

Streszczenie

Jednym z najważniejszych ograniczeń w rozpowszechnieniu modelowania wieloskalowego jest jego duże zapotrzebowanie obliczeniowe. Wynika to z faktu, że każdy z licznych modeli skali dokładnej ma złożoność obliczeniową porównywalną z modele w skali zgrubnej. Istnieje kilka możliwych dróg zmniejszania zapotrzebowania modeli wieloskalowych na moc obliczeniową. Jednym z najbardziej obiecujących podejść jest dostosowywanie struktury model W artykule opisano metodologię Adaptive Multiscale Modelling Methodology, w tym opartą na wiedzy adaptację wieloskalowego modelu kinetyki wydzieleń podczas obróbki cieplnej. Przedstawiono podstawowe cechy metodyki. Przedstawiono model numeryczny obróbki cieplnej stopu aluminium oparty na tej metodologii. Oprócz modelowania makroskopowego przepływu ciepła, wykorzystano modele kinetyki wydzieleń oparte na obliczeniach termodynamicznych. W celu zmniejszenia wymagań obliczeniowych wynikających ze sprzężenia modelu skali makro z modelami termodynamicznymi zastosowano metamodelowanie i koncepcję modeli „podobnych". Opisano wyniki obliczeń przy kilku konfiguracjach reguł. Omówiono zagadnienia dokładności i czasochłonności obliczeń. Omówiono przyszłe perspektywy łączenia modelowania i metamodelowania w ramach pojedynczego, spójnego systemu.

Received: October 22, 2018.
Received in a revised form: November 30, 2018.
Accepted: December 21, 2018.