Acta Materialia 255 (2023) 118988

Contents lists available at ScienceDirect

Acta Materialia

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

Full length article

Kawin: An open source Kampmann–Wagner Numerical (KWN) phase precipitation and coarsening model

Nicholas Ury [a], Raymond Neuberger [b,*], Noah Sargent [c], Wei Xiong [c], Raymundo Arróyave [b,d,e], Richard Otis [a]

[a] Engineering and Science Directorate, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, United States of America
[b] Department of Materials Science & Engineering, Texas A&M University, College Station, TX 77843, United States of America
[c] Physical Metallurgy and Materials Design Laboratory, Department of Mechanical Engineering and Materials Science, University of Pittsburgh, Pittsburgh, PA 15261, United States of America
[d] Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843, United States of America
[e] Department of Industrial & Systems Engineering, Texas A&M University, College Station, TX 77843, United States of America

ARTICLE INFO

Keywords:
Precipitation kinetics
Computer simulations
CALPHAD
Phase transformation
Python

ABSTRACT

Kawin, a new open-source implementation of the Kampmann–Wagner Numerical model of precipitation (concomitant nucleation, growth, and coarsening), is presented. An overview of the organization and capabilities of the program is provided, along with an outline of the constituent physics. Kawin is shown to be able to reproduce the results of state-of-the-art commercial software and experimental data for a variety of alloy systems under multiple precipitation conditions. Kawin is capable of simulating the bulk precipitation behavior of multiphase, multicomponent systems in response to complex heat treatments, and contains numerous innovative features to enhance model stability, improve flexibility and usability, and minimize computational expense. Kawin also incorporates sophisticated elastic energy calculations, traditionally ignored in this type of simulation but shown here to significantly impact the precipitation behavior of some systems. The inclusion of native strain calculations enables Kawin to predict the influence of internal or external stress fields on precipitation, as well as track the evolution of precipitate geometry throughout the course of a heat treatment. It is the hope of these authors that this software will facilitate the advancement of precipitation modeling as a tool for materials design.

1. Introduction

The precipitation of second-phase particles can dramatically alter the material properties of an alloy, allowing for the fine-tuning of properties ranging from tensile strength and ductility to magnetic performance. Smiths and metallurgists have unknowingly made use of this phenomenon since antiquity, with the first written account of tempering steel appearing in Homer's Odyssey circa 800 BCE [1]. The production of Damascus steel, begun as early as 300 BCE and famed throughout the pre-modern period for its superior strength and durability, required a precisely controlled, multi-step heat treatment over the course of several days to produce its distinctive microstructure of folded cementite grains [2]. By the 7th century CE, Japanese swordsmiths were utilizing clay coatings of varying thickness to spatially vary the heat transfer rate during the tempering process, resulting in complex layered microstructure and internal stress patterns that produced swords still held in high regard today [3].

It was not until 1919 that a comprehensive theory for the mechanism behind age hardening – the precipitation of a fine dispersion of second-phase particles – finally emerged, in the seminal paper by Merica, Wallenberg, and Scott [4]. The application of precipitation theory to developments in phase diagrams was a significant driving force in the rapid metallurgical advances of the interwar period, and research into precipitation behavior flourished. By 1935 Classical Nucleation Theory (CNT) had been developed from the work of Becker, Döring, Volmer, and Weber, establishing the mathematics of early-stage precipitation behavior [5]. While capable of accurately predicting particle formation and nucleation rates, CNT was incapable of explaining post-nucleation behavior. In 1961, Lifschitz, Slyozov, and Wagner formalized their theory of grain coarsening, providing a description for the behavior of extant particles at longer time-scales [6].

A unified model of precipitation – combining CNT and LSW coarsening behavior such that nucleation, growth, and coarsening acted

* Corresponding author.
E-mail address: rtneuberger@tamu.edu (R. Neuberger).

https://doi.org/10.1016/j.actamat.2023.118988
Received 14 November 2022; Received in revised form 8 April 2023; Accepted 5 May 2023
Available online 22 May 2023

as concomitant processes – was put forth by Langer and Schwartz in 1980 [7]. The Langer–Schwartz model relied heavily on simplifying assumptions to solve a complex set of differential equations describing the evolution of a particle suspended in a near-critical fluid. In 1984, Richard Wagner and Reinhardt Kampmann proposed a set of modifications to the Langer–Schwartz model that leveraged advances in modern computing to solve for the evolution of an entire particle size distribution, rather than just the mean radius. The Kampmann–Wagner Numerical (KWN) Model, as it came to be known, discretized the particle size distribution into a number of "size bins", within which all particles were assumed to be equal. Each size class was tied to the rest by means of mass balance and continuity equations, applying classical nucleation and growth theories to track the flux of particles in and out of each bin [8].

While the KWN Model is simple in theory, the optimization, mass-balance, and continuity equations required for a stable simulation can rapidly become quite complex, particularly when adapted to model systems that deviate from the original assumptions that governed the model, i.e., homogeneously nucleating spherical particles of pure substances in an isothermal solution. Many such adaptations exist in the literature – allowing for precipitation to be accurately modeled across a wide variety of alloy systems and conditions [9–14] – but due to the diversity in potential KWN-like implementations reconciling these adaptations for use-cases other than the original authors' can be an arduous task. Commercial software packages including implementations of KWN-like precipitation modules are available from Thermo-Calc, CompuTherm, and others, yet these programs are often expensive and their proprietary nature makes adapting and integrating the code difficult. Furthermore, it is difficult to advance the state-of-the-art of precipitation modeling when many implementations of the KWN model are hidden away from the users. To the best of the authors' knowledge, no open-source KWN implementation is currently available.

Recent advances in metals additive manufacturing (AM) have reignited interest in precipitation simulation [15]. The many tunable parameters of AM processes permit a previously unimaginable level of control over the thermal history of a part, with documented potential for using an AM process to produce a spatially varied thermal history — and therefore spatially varied precipitate structures and material properties [16]. Attaining the degree of precipitation control necessary for effective materials design cannot occur without the ability to accurately predict the precipitation response induced by an AM-like thermal history. Much progress has been made in this area through (often expensive) experimentation [17], but the materials design community is in need of a physics-based precipitation model that can be flexibly adapted to the non-standard conditions encountered during additive manufacturing and easily integrated with existing AM process models. It is with this purpose in mind that the authors are releasing Kawin, the first open-source implementation of the Kampmann–Wagner Numerical Model for phase precipitation and coarsening.

2. Methods

2.1. The Kampmann–Wagner Numerical (KWN) Model

At its core, the KWN algorithm is a mean-field model of precipitation, tracking the evolution of average properties of a bulk volume rather than the localized properties surrounding an individual precipitate. Since particles are not tracked individually, the particle size distribution is modeled as a continuous function where particles are placed in uniformly-spaced bins that represent the size of the particles. At each time step, nucleation, growth, and dissolution behaviors are calculated and the particle size distribution (PSD) updated to reflect any changes while still adhering to strict mass balance and continuity equations.

A generalized overview of the core physics underpinning Kawin is provided in the remainder of this section. A more thorough treatment of the concepts and precise equations for the various correction factors and user options can be found in the Kawin documentation [18].

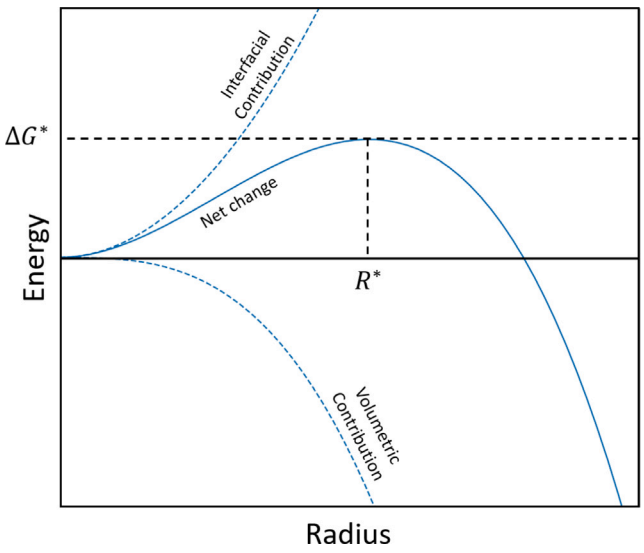

Fig. 1. The sum of the interfacial energy contributions and volumetric contributions for a nucleate creates a nucleation barrier with a height of ΔG^* at critical radius R^*.

2.1.1. Classical nucleation theory

Nucleation in Kawin is modeled according to Classical Nucleation Theory (CNT). In CNT, the formation of second-phase particles occurs due to heterophase fluctuations in a metastable solid solution. These fluctuations are assumed to occur constantly as a result of the thermal instability of the parent phase. Most such fluctuations are unstable—only once the volumetric reduction in free energy resulting from phase dissociation exceeds the energy cost of creating a new interface will the second-phase nucleus become stable and proceed to grow [19]. The net free energy change of nucleus formation as a function of its radius is given in Eq. (2.1.1), depicted graphically in Fig. 1.

$$\Delta G = -\frac{4}{3}\pi R^3 \Delta G_v + 4\pi R^2 \gamma \tag{2.1.1}$$

As depicted in Fig. 1, the critical radius R^*—the radius at which the volumetric and interfacial energy contributions are in a state of unsteady equilibrium—can be found using Eq. (2.1.2). Continued growth of a particle beyond this critical radius will result in a net decrease in system free energy and the formation of a stable nucleus. The energy cost of creating a nucleus of critical size is therefore the energy barrier to nucleation ΔG^*, given in Eq. (2.1.3).

$$R^* = \frac{2\gamma}{\Delta G_v} \tag{2.1.2}$$

$$\Delta G^* = \frac{4}{3}\pi\gamma R^{*2} = \frac{16}{3}\frac{\pi\gamma^3}{\Delta G_v^2} \tag{2.1.3}$$

The proportion of newly formed clusters that will attain this critical size can be found according to a Maxwell–Boltzmann distribution, which can then be adjusted by the Zeldovich factor Z (the probability that stable clusters will dissolve), impingement rate β (the rate at which solutes diffuse toward a cluster), number of available nucleation sites N_0, and an incubation time τ to obtain the steady-state nucleation rate, as given in Eq. (2.1.4) [20].

$$J_{nuc} = N_0 Z \beta \exp\left(-\frac{\Delta G^*}{k_B T}\right) \exp\left(-\tau/t\right) \tag{2.1.4}$$

Notably, the KWN model only tracks stable clusters, thus all newly nucleated particles are assumed to have a size of $R^* + \frac{1}{2}\sqrt{\frac{kT}{\pi\gamma}}$ to ensure growth.

2.1.2. Precipitate growth

Assuming the precipitate is spherical and the composition profile is in a quasi-steady state, the composition of the matrix outside the

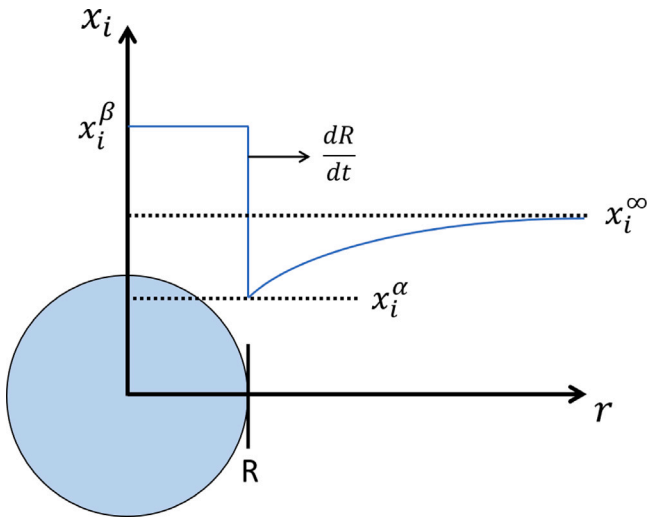

Fig. 2. Composition profile from precipitate to matrix.

precipitate is the solution to the Laplace equation, given in Eq. (2.1.5) (Fig. 2) [21]. Here, the composition (x_i) at a distance r from the precipitate center is dependent on the far-field composition (x_i^∞), the interfacial composition in the matrix phase ($x_{R,i}^\alpha$) and the precipitate radius (R).

$$x_i(r) = x_i^\infty + \frac{(x_{R,i}^\alpha - x_i^\infty)R}{r} \tag{2.1.5}$$

The interfacial velocity is related to the concentration gradient by Eq. (2.1.6), where $x_{R,i}^\beta$ is the interfacial composition in the precipitate phase and D_{ij} is the diffusivity.

$$(x_{R,i}^\beta - x_{R,i}^\alpha)\frac{dR}{dt} = \sum_j D_{ij}\frac{dx_j}{dr}\bigg|_{r=R} \tag{2.1.6}$$

Combining these two equations gives the growth rate as shown in Eq. (2.1.7) [21,22].

$$\frac{dR}{dt} = \sum_j \frac{D_{ij}}{R}\frac{x_j^\infty - x_j^\alpha}{x_i^\beta - x_i^\alpha} \tag{2.1.7}$$

Coarsening arises naturally from the interaction of the nucleation and growth equations used in the KWN model. Continued nucleation and growth will decrease the supersaturation of the solid solution, lowering the driving force for nucleation. As a consequence of Eq. (2.1.2), this decrease in driving force results in an increase in the critical radius. If the critical radius surpasses that of any of the smaller size bins, those particles will become thermodynamically unstable and begin to dissolve. This is handled in Eq. (2.1.7) where any particles smaller than the critical radius will exhibit a negative growth rate and return solute to the matrix, facilitating further growth of large precipitates. For numerical efficiency, particles that drop below a minimum radius are assumed to fully dissolve and are removed from the simulation.

2.1.3. Mass balance

After nucleation and growth have been calculated and the particle size distribution updated, it is necessary to perform a mass balance to update matrix solute concentrations. The number density, average radius and volume fraction can be found by taking the zeroth, first and third moment of the PSD respectively(Eqs. (2.1.8)–(2.1.10)) where n_i is the number of particles per unit volume in size class i, R_i is the radius of size class i and f_v is the volume fraction.

$$N_{total} = \sum_i n_i \tag{2.1.8}$$

$$R_{avg} = \frac{\sum_i n_i R_i}{\sum_i n_i} \tag{2.1.9}$$

$$f_v = \frac{4\pi}{3}\sum_i n_i R_i^3 \tag{2.1.10}$$

To conserve mass, the sum of solute atoms in the matrix and in the precipitate must be equal to the initial number of solute atoms. Thus the current composition in the matrix can be determined by Eq. (2.1.11), where $x_{j,0}$ is the initial composition and p is used to refer to a variable for phase p.

$$x_{j,0} = (1 - \sum_p f_{v,p})x_j^\infty + \frac{4\pi}{3}\sum_p\sum_i n_{p,i}R_{p,i}^3 x_{R_{p,i},j}^\beta \tag{2.1.11}$$

2.2. Kawin implementation and features

Historically, KWN models have faced several barriers to their widespread use that Kawin attempts to address. The original model developed by Kampmann and Wagner was built upon numerous assumptions that limited its applicability outside of the simplest of precipitation scenarios. Later works have vastly expanded upon the capabilities of KWN models through the introduction of numerous correction factors and algorithmic refinements, but a comprehensive compilation of these corrections has never before existed outside of commercial software that cannot be modified to better suit the user's needs or experiment with changes to the correction factors and algorithm itself.

Kawin features a modular design to provide flexibility in development and facilitate user modification of various aspects of the program. The core component of the code is the PrecipitateBase object, which contains all the necessary model parameters and functions to solve the KWN model. The PrecipitateBase object is inherited by the PrecipitateModel object, which is kept separate to allow for easy modification to algorithm structure and model physics. Kawin is currently bundled with a Eulerian solution to the KWN model, in which growth rates are converted into fluxes and particle frequency in each bin is governed by a continuity equation. The relationship between each module in Kawin is shown in Fig. 3, where it is shown that additional features can be added on with little change to the model implementation defined in the PrecipitateBase and PrecipitateModel objects. Other kinetic models can be added easily as the Thermodynamic modules supply diffusivity and mobility calculations that can be used for most kinetic models. In addition, further property calculations such as strength modeling can be added by taking values generated and stored in the PrecipitateModel object.

2.2.1. Thermodynamics and kinetics modules

Thermodynamic and kinetic calculations are handled in the Thermodynamics, Mobility, Free Energy Hessian, and Local Equilibrium modules. Kawin features native integration with PyCalphad, an open source tool for computational thermodynamics. PyCalphad is used to compute both the free energy change of phase dissociation and the composition on both sides of the matrix–precipitate interface, necessary for calculating growth rates [23]. Kawin additionally supports the usage of CALPHAD-based mobility models to determine atomic mobility. The diffusivities of multicomponent systems can be calculated using the free energy curvature and chemical potential gradients.

2.2.2. Correction factors

The general KWN model is predicated on uniformly spherical precipitates. The calculation of driving forces, interfacial compositions, and growth rates are all curvature dependent, and assume a constant radius of curvature across the precipitate surface. Particles that deviate from a spherical shape due to either elastic considerations or heterogeneous precipitation sites can be modeled through the use of various correction factors applied to thermodynamic and kinetic terms, based on previous analysis of single precipitate growth [10–12,24,25]. These correction factors are applied following Eqs. (2.2.1) and (2.2.2), with $f(\alpha)$ and $g(\alpha)$ representing the thermodynamic and kinetic corrections necessary for a given particle shape. In this way, Kawin is capable of modeling a wide range of precipitate geometries including spheres,

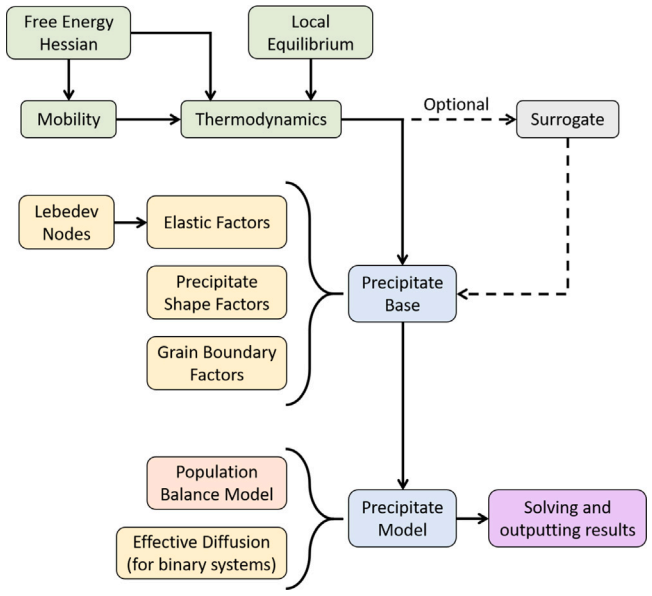

Fig. 3. Object hierarchy in Kawin showing the relationship between each module. Green objects directly interface with PyCalphad, blue objects are implementations of the KWN model, yellow objects are extra factors for the model, orange objects support specific features of a KWN implementation, gray objects are optional features and purple objects relate to output. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

spheroids, needles, plates, and cuboids. Equations for the correction factors of these shapes can be found in the Kawin documentation [18].

$$\frac{dR}{dt} = f(\alpha)\frac{dR}{dt}\bigg|_{sphere} \tag{2.2.1}$$

$$\mu_A^{\alpha} = \mu_A^{\beta} + \left(g(\alpha)\frac{2\gamma}{R} + \Delta G_{el}\right) V_M^{\beta} \tag{2.2.2}$$

Kawin's Grain Boundary Factors module enables calculation of shape factors for precipitates accommodated by dislocations and grain boundaries, edges and corners. For precipitates that nucleate on grain boundaries, the number of available nucleation sites, nucleation barrier, and precipitate morphology, will differ from bulk nucleation.

Nucleation on a grain boundary requires both the formation of a precipitate–matrix interface and the removal of a grain boundary area (matrix–matrix interface). The Gibbs free energy involved in nucleating a precipitate is thus modified, as shown in Eq. (2.2.3) [26,27].

$$\Delta G = -cR^3\Delta G_v + b\gamma_{\alpha\beta}R^2 - a\gamma_{\alpha\alpha}R^2 \tag{2.2.3}$$

$\gamma_{\alpha\alpha}$ is the grain boundary energy, $\gamma_{\alpha\beta}$ is the interfacial energy between the precipitate and the matrix, and R is the radius of curvature. a, b, and c are multiplying factors to the radius to obtain the grain boundary area that is removed, the surface area of the precipitate, and the volume of the precipitate, respectively.

2.2.3. Elastic energy module

Kawin includes extensive methods for calculating the elastic strain energy associated with precipitate formation. As the matrix deforms to accommodate lattice mismatch between the matrix and precipitate, an additional volumetric reduction in free energy must be added to Eq. (2.1.1), slowing or accelerating nucleation depending on the geometry and strain state in question. The elastic strain energy associated with spheroidal precipitates is calculated according to Eshelby's theory [28], while Khachaturyan's approximation is used for spherical and cuboidal precipitates [29].

In many cases, non-spherical precipitate geometry can be attributed to elastic anisotropy in either the matrix or precipitate phase. While all known precipitates are initially spherical to minimize interfacial area, elastic effects can dramatically alter precipitate aspect-ratio as the volume-to-surface-area ratio changes with growth. Kawin's incorporation of a model for elastic energy contributions allows for the equilibrium aspect ratio of such precipitates to be predicted by finding the aspect ratio that minimizes both elastic and interfacial energy contributions. Variable aspect ratio across size classes is tracked and accounted for in the mass balance equation.

$$\alpha = argmin\left(\frac{4}{3}\pi R_{sph}^3 E(\alpha) + 4\pi R_{sph}^2 g(\alpha)\gamma\right) \tag{2.2.4}$$

2.2.4. Population balance model module

The evolution of the particle size distribution (PSD) is handled by a population balance model (PBM). The PBM discretizes the distribution into size classes. Particle growth is handled through a Eulerian implementation, where the growth rates are converted into fluxes and the frequencies in each size class is governed by the continuity equation, where n is the frequency in size class i, G is the growth rate, R is the particle size and J_{ext} represents any external fluxes (such as nucleation).

$$\frac{dn}{dt} + \frac{d(nG)}{dR} = J_{ext} \tag{2.2.5}$$

For discretized size classes, Eq. (2.2.5) is handled by calculating the growth rate at the bounds of each size class. The frequency of size class i is determined by Eq. (2.2.6). With these two equations, the evolution of the particle size distribution can be determined from the nucleation and growth rates. A schematic of this process is shown in Fig. 4

$$\begin{aligned} n_i^{t+1} = n_i^t + \frac{\Delta t}{\Delta R}\big(&G_{i-1}(\mathrm{sgn}(G_{i-1})n_{i-1} + \mathrm{sgn}(-G_{i-1})n_i) \\ &- G_i(\mathrm{sgn}(G_i)N_i + \mathrm{sgn}(-G_i)N_{i+1})\big) \end{aligned} \tag{2.2.6}$$

2.2.5. Data management and visualization

Kawin tracks precipitate volume fraction, average radius, matrix composition, precipitate number density, mean aspect ratio, driving force, nucleation rate, critical radius, interfacial composition, equilibrium composition, equilibrium volume fraction, supersaturation, and the particle size distribution at every step of the simulation. Any of these variables can be visualized using Kawin's built in plotting function, or exported as a CSV file or NumPy binary file for further analysis or use with other tools. Kawin also offers reset functions, allowing multiple simulations to be run consecutively without needing to reload thermodynamic data.

3. Results

3.1. Comparison with state-of-the-art precipitation software

In order to validate the implementation of the KWN algorithm in Kawin, Examples 05 and 08 in the software documentation of commercial KWN implementation TC-Prisma [30], modeling the Ni–Cr–Al and Cu–Ti systems, respectively, are recreated using Kawin. Several additional parameters are defined in these comparisons in order to avoid differences in default parameter values. Table 1 lists the parameters used for each simulation.

Fig. 5 shows the simulation results of the Ni–Cr–Al system comparing TC-Prisma, Kawin (with and without using surrogate models, discussed in Section 3.2.2.) and experimental data found in literature [31]. Different CALPHAD databases had to be used for Kawin and TC-Prisma due to availability and the compatibility of the databases between Thermo-Calc and PyCalphad. For Kawin, the database for Ni–Cr–Al was constructed from data found in the literature [32], while TC-Prisma used the TCNI12/MMOBNI6 database. Despite this, both programs provided comparable results that fit well with the available experimental data.

Fig. 6 shows simulation results of the Cu–Ti system comparing TC-Prisma, Kawin and experimental data from literature [8,33]. Both

Table 1
Parameter values for Ni–Cr–Al and Cu–Ti systems used in comparison of Kawin and TC-Prisma.

Parameter	Value (Ni–Cr–Al)	Value (Cu–Ti)
Parent Phase	FCC_A1	FCC_A1
Precipitate Phase	FCC_L12	CU4_TI
Simulation time	1e6 s	1e5 s
Initial composition	Ni–9.8Al–8.3Cr (at.%)	Cu–1.9Ti (at.%)
Interfacial energy	0.012 J/m^2	0.035 J/m^2
Temperature	1073 K	623 K
Molar volume (parent)	6.566e−6 m^3/mol	7.11e−6 m^3/mol
Molar volume (precipitate)	6.566e−6 m^3/mol	7.628e−6 m^3/mol
Nucleation site	bulk	bulk
Nucleation site density	9.172e28 /m^3	8.470e28 /m^3
Elastic constants (C11, C12, C44)	N.A.	168.4 GPa, 121.4 GPa, 75.4 GPa
Eigenstrain (ε_{11}, ε_{22}, ε_{33})	N.A.	0.022, 0.022, 0.003

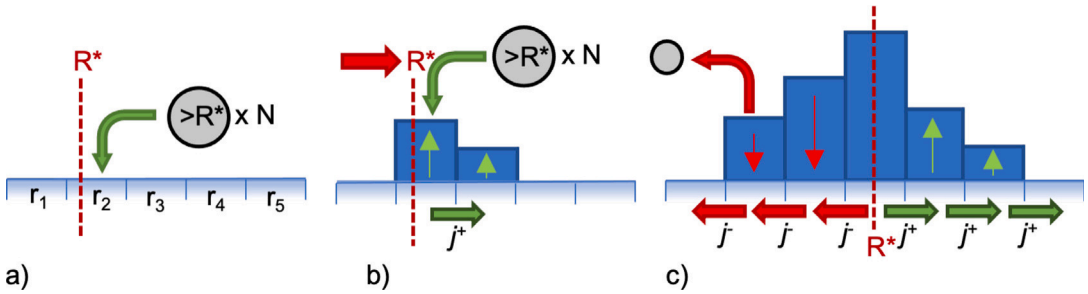

Fig. 4. Schematic of how the particle size distribution is modeled using fixed bin sizes in a Eulerian KWN implementation. In 4a The nucleation of N new particles is calculated via Classical Nucleation Theory and added to the size bin that corresponds to the critical radius. In 4b nucleation continues while the growth of added particles is modeled via the flux between size bins. As supersaturation decreases the critical radius increases. In 4c, continuing nucleation and growth have decreased supersaturation such that the critical radius is larger than particles in the PSD. Particles larger than the critical radius continue to grow while particles smaller than the critical radius shrink and are removed from the simulation, capturing coarsening behavior.

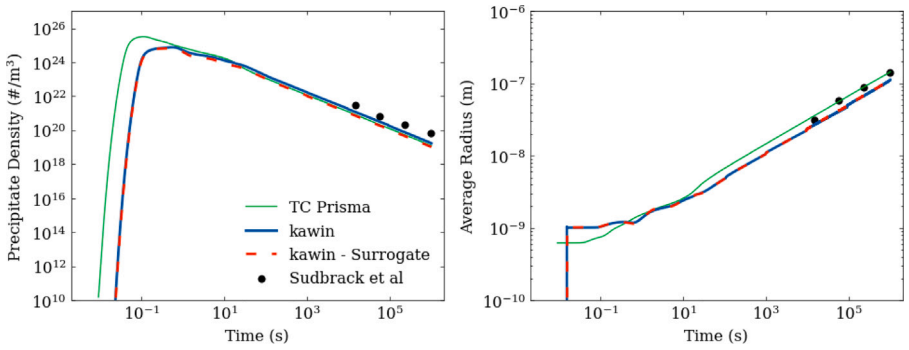

Fig. 5. Precipitation simulation results of the Ni–Cr–Al system comparing TC-Prisma, Kawin and experimental data.

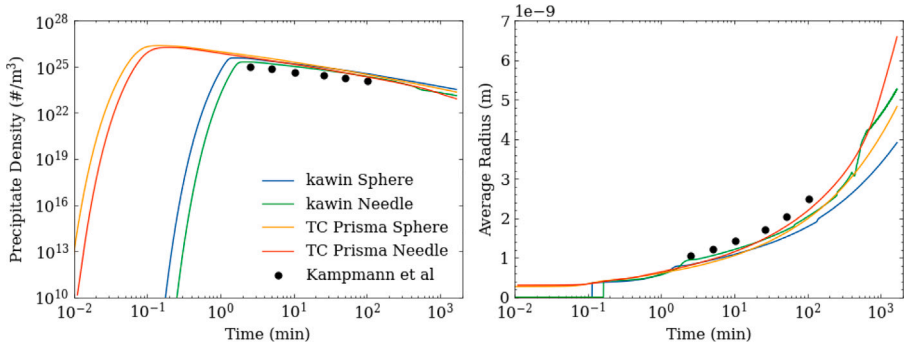

Fig. 6. Precipitation simulation results of the Cu–Ti system comparing TC-Prisma, Kawin and experimental data.

programs here utilized the same thermodynamic database for Cu–Ti, constructed from literature data [34,35]. While TC-Prisma predicts an earlier nucleation onset than Kawin, the two programs converge after about 6 s of simulation time and both align well with available experimental data. Of note, the simulations performed using a spherical particle approximation took 19 s to compute via TC-Prisma and 31 s to compute via Kawin. The needle-shaped particle simulation took 773 s to compute via TC-Prisma and only 56 s to compute via Kawin. The

differences in run time are likely due to the database size, where larger databases may take longer to load, and due to differences in implementation. For example, TC-Prisma may treat the particle size distribution with higher fidelity which could slow down the run time.

3.2. Performance improvements

3.2.1. Caching

When solving the KWN model, the driving force and interfacial composition must be calculated at each time step. This involves at least two equilibrium calculations, which can result in performance issues where the bulk of the time spent per iteration is spent computing PyCalphad's equilibrium function. Equilibrium calculations typically involve sampling the Gibbs free energy surface, then performing a minimization procedure to minimize the Gibbs free energy while subject to mass balance and summation constraints. Because the composition and temperature generally change very little per iteration in the precipitate model, the sampling step of the equilibrium calculation can be avoided by caching the results of the previous calculation. Equilibrium is then calculated by taking the previous equilibrium calculation and performing the minimization procedure subject to the new mass balance constraints. In the aforementioned CuTi example, running the simulation with caching enabled reduced the simulation time from 91.0 s to 39.4 s, a 56.7% decrease. In addition to faster performance, using cached results also improves the stability of the simulation when driving forces are very small. For small driving forces, the precipitate phase may sometimes not be detected if the sampling density of the free energy surfaces are not sufficient. A sufficient sampling density is generally not known beforehand and setting a very high sampling density will also reduce the performance of the model. However, when using cached results, precipitates just below the chemical potential hyperplane of the matrix phase will still be detected and considered stable.

3.2.2. Surrogate modeling

Modeling non-isothermal processes introduces significant variability into the thermodynamic properties required to solve the KWN model, and necessitates repeated recalculation of these variables as the temperature changes. While Kawin includes code to minimize extra calculations, it also comes packaged with surrogate models for driving force, growth rate, diffusivity, and interfacial composition.

For relatively simple multicomponent systems(less than 5 elements), a surrogate of the driving force and the various terms can be derived from the curvature of the free energy surface to enable calculation of growth rate and interfacial composition. Both surrogates require a set of compositions and temperatures for training. In this case, a set of equilibrium calculations are performed beforehand to determine driving force and interfacial composition for various compositions and temperatures. These are then interpolated over the range of the precipitate simulation and sampled directly to avoid constant equilibrium recalculations. The surrogate models use radial basis function (RBF) interpolation, as provided by the SciPy library [36]. The functions in the Surrogate module are designed to be interchangeable with the corresponding functions in the Thermodynamics module. For systems with more than five elements, using surrogate models may not be as effective due to the large degrees of freedom involved.

Without using the caching feature discussed in Section 3.2.1, the Ni–Cr–Al system took 32.9 s to calculate. A surrogate was trained by calculating the driving force and interfacial compositions at 256 grid points over the relevant compositions. The training took 12.6 s that reduced the simulation time to only 6.2 s, for a total computation time of 18.8 s–a 42.8% decrease. The results of the KWN model solved using both PyCalphad and the surrogate are shown in Fig. 5 showing that there is little to no loss in accuracy. However, it is important to note that the accuracy of the surrogate models are dependent on the training data, where more training data will result in more accurate surrogate models while taking longer to train. These surrogates can be trained and saved prior to running any simulation, significantly speeding up if batches of simulations using the same thermodynamic system are needed to be performed.

3.2.3. Adaptive stability measures

A major challenge in the implementation of any KWN model is choosing size bin widths and time-step sizes that will produce a stable simulation. Due to the discretized nature of particle sizes and the large quantities of particles that can nucleate in a short time-span, the KWN algorithm is prone to instability if the choice of bin widths and kinetics of the system results in too significant, or too little, of a change in any time step. This issue is traditionally solved through optimization of model parameters against experimental data, but Kawin incorporates a number of innovative measures that automatically adapt model parameters – the time step and size bin width – to maintain a stable and accurate simulation at minimal computational expense.

During each time step, Kawin checks to ensure both that the maximum observed growth rate is no more than half the width of a size bin and that no more than a configurable limit of phase volume fraction change occurs, subdividing the time step if necessary to maintain these conditions. If the bins are too small, the time interval between each iteration will be severely constrained to account for large fluxes. If the bins are too large, the time increment will not be constrained; however, the low resolution of the PSD may lead to inaccurate results. The bin sizes of the PSD are modified by the following rules:

1. Nucleation: If the size of a nucleate (R_{nuc}) is larger than the maximum bin size (R_{max}), then PSD will be recreated such that $R_{max} = 5R_{nuc}$. The number of bins will be set to the default value of B_0.
2. Adding size classes: If the largest size class has a precipitate density of at least 1 $\text{particle}/\text{m}^3$ then $B_0/4$ will be added at the end of the PSD.
3. Increasing bin size: After adding size classes, if the number of bins is greater than the maximum allowed amount (B_{max}), the PSD will be recreated to have B_{min} bins while keeping the same range of size classes.
4. Decreasing bin size: During dissolution, if the largest bin with a precipitate density greater than 1 $\text{particle}/\text{m}^3$ (R_{filled}) is smaller than $0.5R_{max}$, then the PSD will be recreated to have B_{max} bins with the range extending from $[R_{min}, R_{filled}]$. Dissolution is assumed to occur when the growth rate of all size classes are negative.

Once the PSD bins are adjusted, the frequency of particles within each bin is determined by interpolating the distribution density, then adding a small correction to conserve the volume fraction. A comparison of simulation times for fixed and adaptive bin sizes are shown in Fig. 7. Fig. 8 shows the precipitate density and volume fraction of the example using different fixed bin widths and the adaptive bin sizing. At a fixed bin size of 0.75 nanometers, the simulation time dramatically increases; however, an explanation for this behavior is currently unknown as the interplay between the numerical stability and particle size distribution is quite complex. The smaller bin sizes give higher fidelity simulations while taking longer. As the bin sizes increases, the simulation time decreases with the expense of resolution. Increasing the bin size even further results in an increasing simulation time as the model is no longer able to handle the particle size distribution correctly. The implementation of adaptive bin sizing allows Kawin to reproduce the results of high-resolution models (bin sizes smaller than 1 nanometer) at a minimum simulation time.

The process flowchart for Kawin's implementation of the KWN model is shown in Fig. 9. Several checks are performed per iteration to ensure numerical stability. The first set of checks are to estimate a proper time step for the iteration before any significant calculations are done. This is done by calculating the nucleation rate and growth rate to predict the change in other variables such as precipitate density, composition and volume fraction. The time step is then used to update the particle size distribution. The second set of checks involves the PSD and

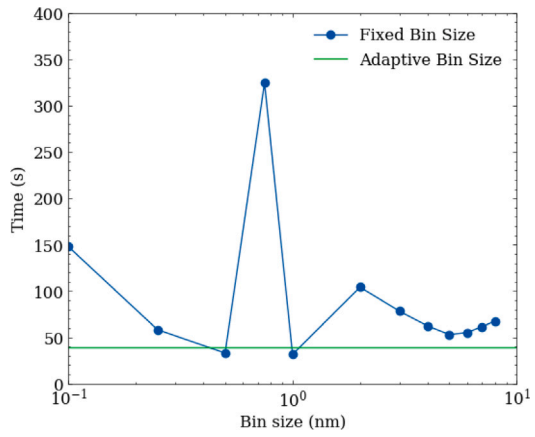

Fig. 7. Comparison of simulation times for different fixed bin sizes and adaptive bin sizes.

will change the bin sizes according to the rules listed in Section 3.2.3. Finally, the composition and volume fraction is calculated and a final set of checks are performed to ensure the actual change in these two variables satisfy the numerical constraints.

3.3. Additional examples

To showcase the multiphase and elastic energy capabilities of Kawin, two additional case studies are shown below. The first is in the Al–Mg–Si system, where a precipitate model is run with five distinct precipitate phases. The second case study is in the Ni-Ti system, showing the drastic effects that elastic strain energy can have on a system.

3.3.1. Multi-phase precipitation in Al–Mg–Si

In the Al–Mg–Si system, several phases can form including: β', β'', B', U1 and U2 [12,37]. Kawin tracks the PSD evolution of each precipitate individually. Nucleation and growth rate are also handled for each precipitate phase independently. Coupling only comes from the mass balance where all precipitates contribute to the overall mass changes in the system. For example, dissolution of one precipitate phase can occur when nucleation and growth of a second precipitate results in the composition in the matrix to be low enough to make the former precipitate phase unstable.

Setting up a precipitate model only requires the thermodynamic database to be loaded along with a few additional parameters. These include the simulation time, initial composition, temperature, molar volumes and interfacial energies. In this case, the temperature is assumed to be a two-stage heat treatment, where the alloy is held at 175 °C for 16 h and heated to 250 °C until 25 h (Fig. 10).

```
from Kawin.Thermodynamics import MulticomponentThermodynamics
from Kawin.KWNEuler import PrecipitateModel
import matplotlib.pyplot as plt

phases = ['FCC_A1', 'MGSI_B_P', 'MG5SI6_B_DP', 'B_PRIME_L', 'U1_PHASE', 'U2_PHASE']
therm = MulticomponentThermodynamics('AlMgSi.tdb', ['AL', 'MG', 'SI'], phases,
    drivingForceMethod='approximate')

model = PrecipitateModel(0, 25*3600, 1e4, phases=phases[1:], elements=['MG', 'SI'],
    linearTimeSpacing=True)

model.setInitialComposition([0.0072, 0.0057])
model.setVmAlpha(1e-5, 4)

gamma = {
    'MGSI_B_P': 0.18,
    'MG5SI6_B_DP': 0.084,
    'B_PRIME_L': 0.18,
    'U1_PHASE': 0.18,
    'U2_PHASE': 0.18
    }

for i in range(len(phases)-1):
    model.setInterfacialEnergy(gamma[phases[i+1]], phase=phases[i+1])
    model.setVmBeta(1e-5, 4, phase=phases[i+1])
    model.setThermodynamics(therm, phase=phases[i+1])

lowTemp = 175+273.15
highTemp = 250+273.15
model.setTemperatureArray([0, 16, 17], [lowTemp, lowTemp, highTemp])

fig, ax = plt.subplots(1, 1, figsize=(6, 5))
model.plot(ax, 'Temperature', timeUnits='h')
ax.set_ylim([400, 550])
ax.set_xscale('linear')
plt.show()
```

Once all the parameters are inserted, the model can be solved and the results are plotted. Kawin features easy plotting methods where tracked variables can be plotted for each phase or as a summation of all phases. The results of these simulation are shown in Fig. 11.

```

model.solve(verbose=True, vIt=5000)

fig, axes = plt.subplots(2, 2, figsize=(10, 8))

model.plot(axes[0,0], 'Total Precipitate Density', timeUnits='h', label='Total', color='k',
linestyle=':', zorder=6)

model.plot(axes[0,0], 'Precipitate Density', timeUnits='h')
axes[0,0].set_ylim([1e5, 1e25])
axes[0,0].set_xscale('linear')
axes[0,0].set_yscale('log')

model.plot(axes[0,1], 'Total Volume Fraction', timeUnits='h', label='Total', color='k',
linestyle=':', zorder=6)

model.plot(axes[0,1], 'Volume Fraction', timeUnits='h')
axes[0,1].set_xscale('linear')

model.plot(axes[1,0], 'Average Radius', timeUnits='h')
axes[1,0].set_xscale('linear')

model.plot(axes[1,1], 'Composition', timeUnits='h')
axes[1,1].set_xscale('linear')

fig.tight_layout()
plt.show()
```

3.3.2. Elastically influenced precipitation in Ni-Ti

Ni-Ti alloys produce Ni_4Ti_3 precipitates at near-equilibrium composition. The metastable Ni_4Ti_3 phase exists in a highly strained state, forming plate-like precipitates oriented along the [111] plane of the Ni-Ti matrix [38]. Kawin's elastic energy module is capable of accounting for non-coincident lattices and calculating the elastic strain energy associated with the formation and growth of these precipitates, and

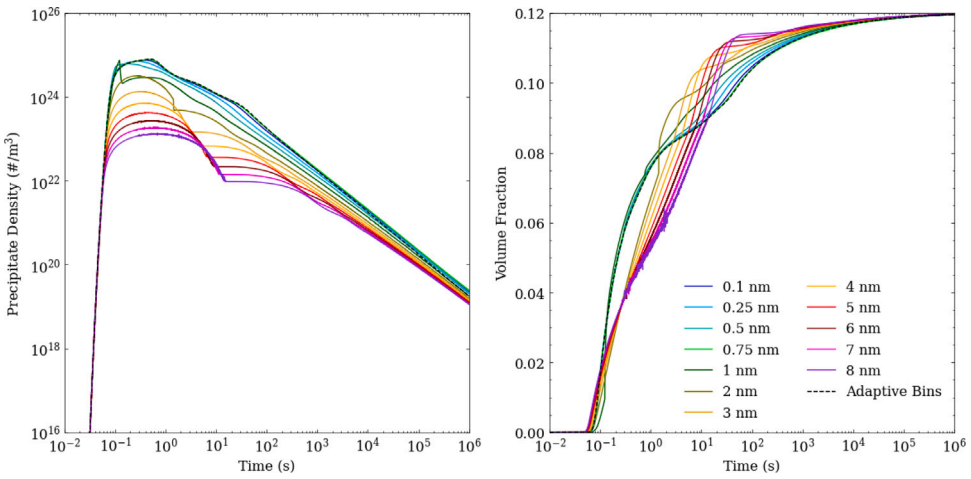

Fig. 8. Comparison of precipitate density and volume fraction over different fixed bin size and adaptive bin sizes.

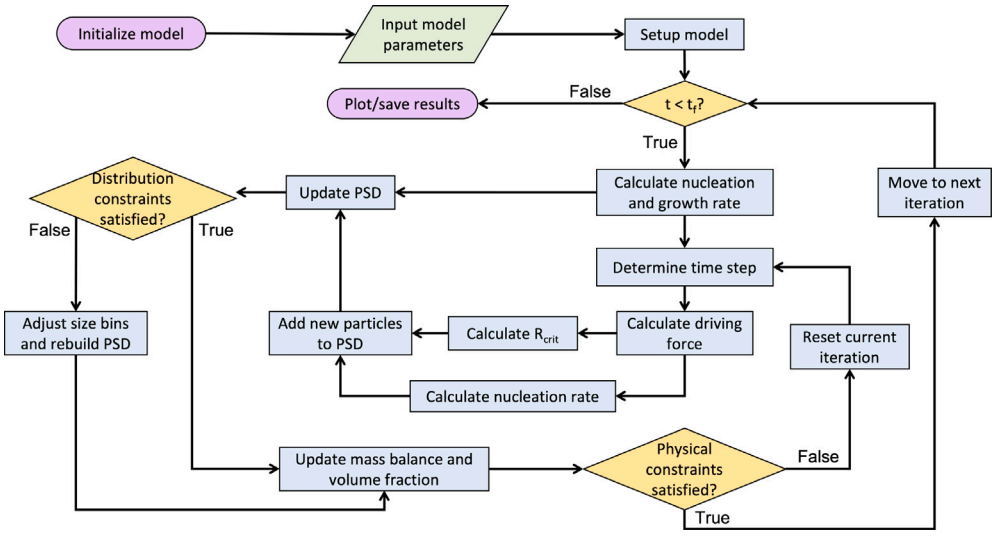

Fig. 9. Flowchart for the KWN model implementation accounting for adaptive bin sizing and numerical constraints.

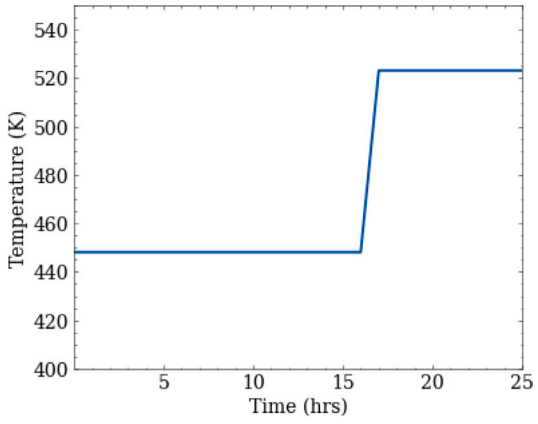

Fig. 10. The two-stage heat treatment to be simulated in the Al–Mg–Si example.

modeling the impact that this strain state has on the evolution of precipitate morphologies across the PSD.

For a precipitate of a given size, the aspect ratio is determined by minimizing the elastic strain energy and interfacial energy [11].

The following script plots out the two contributions and the total contribution for ellipsoid precipitates with a volume-equivalent spherical radius of 20 nm (Fig. 12). Additionally, the relationship between the equivalent spherical radius, equilibrium aspect ratio, and molar strain energy is shown, illustrating the decrease in strain energy that drives the increasing aspect ratio of Ni_4Ti_3 precipitates during growth.

```
import numpy as np
import matplotlib.pyplot as plt

from kawin.Thermodynamics import BinaryThermodynamics
from kawin.KWNEuler import PrecipitateModel
from kawin.ElasticFactors import StrainEnergy
from kawin.ShapeFactors import ShapeFactor

plt.style.use(['science', 'no-latex'])
plt.rcParams.update({'font.size': 14})

#Set up thermodynamics
phases = ['BCC_B2', 'TI3NI4']
therm = BinaryThermodynamics('NiTi_SMA.tdb', ['TI', 'NI'], phases)

#Override guess composition(s) to reduce number of calculations
# when finding interfacial composition
therm.setGuessComposition(0.56)

```

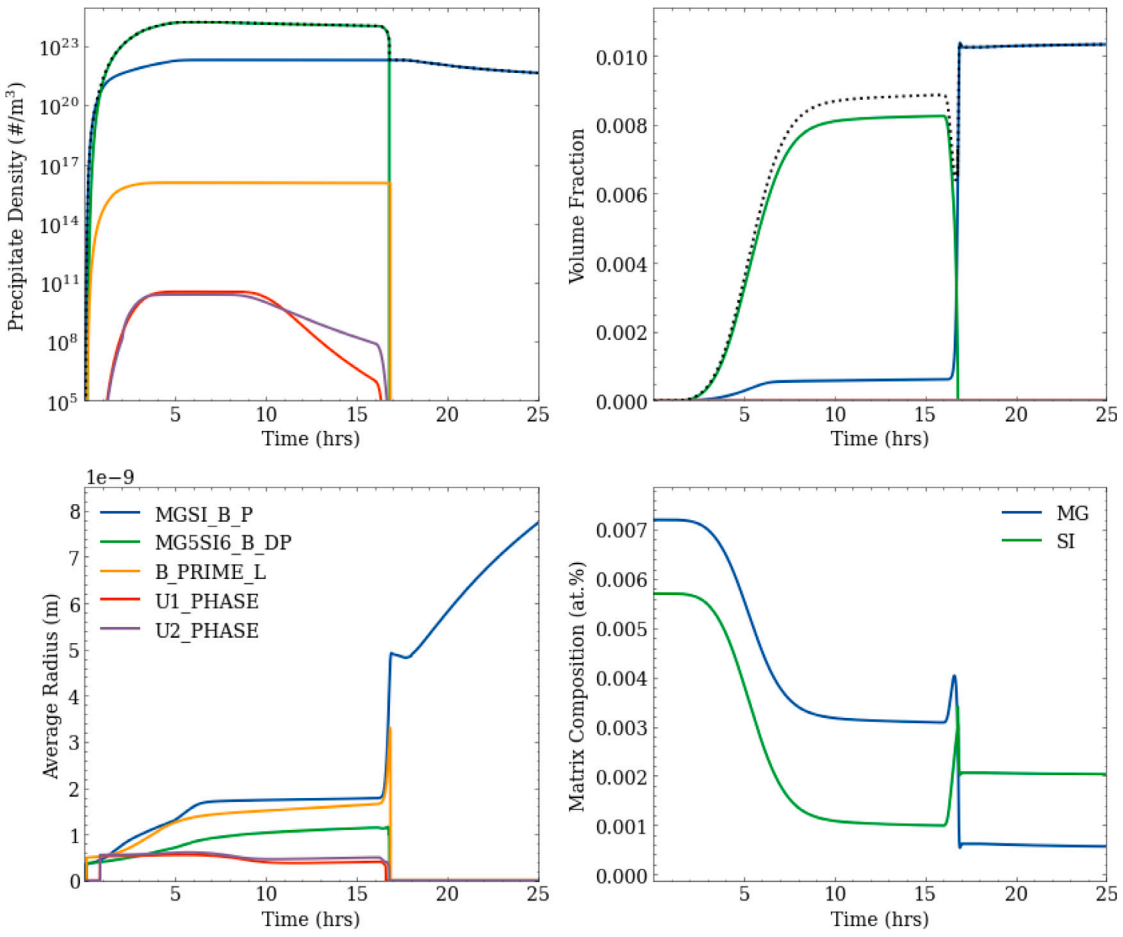

Fig. 11. Results of the Al–Mg–Si simulation showing how (a) precipitate density, (b) volume fraction, (c) average radius and (d) composition changes over time.

```
#Model parameters
xinit = 0.508
gamma = 0.053
T = 450+273.15
Dni = lambda x, T: 1.8e-8 * np.exp(-155000/(8.314*T))
vaBCC, nBCC = 0.0268114e-27, 2
vaNI3TI4, nNI3TI4 = 0.184615e-27, 14

se = StrainEnergy()
B2e = np.asarray([175,45,35]) * 1e9
eigenstrain = [-0.00417, -0.00417, -0.0257]
rotate = [[-4/np.sqrt(42), 5/np.sqrt(42), -1/np.sqrt(42)],
      [-2/np.sqrt(14), -1/np.sqrt(14), 3/np.sqrt(14)],
      [1/np.sqrt(3), 1/np.sqrt(3), 1/np.sqrt(3)]]
se.setEigenstrain(eigenstrain)
se.setElasticConstants(B2e[0],B2e[1],B2e[2])
se.setRotationMatrix(rotate)

fig, axes = plt.subplots(1, 2, figsize=(12,5))
ax2 = axes[1].twinx()

sf = ShapeFactor()
sf.setPlateShape()
se.setEllipsoidal()
se.setup()

#Plot energy contributions for different aspect ratios
# at an equivalent spherical radius of 20nm
Rsph = 2e-8
Vsph = 4/3*np.pi*Rsph**3
Asph = 4*np.pi*Rsph**2

#Strain and interfacial energy
ar = np.linspace(1, 10, 100)
volE = Vsph * se.strainEnergy(sf._normalRadiiEquation(ar))
areaE = Asph * gamma * sf._thermoEquation(ar)
sumE = volE + areaE

axes[0].plot(ar, volE, linewidth=2, label='Strain Energy')
axes[0].plot(ar, areaE, linewidth=2, label='Interfacial Energy')
axes[0].plot(ar, sumE, color='C2', linewidth=2, label='Sum of
    ↪ Energies')
axes[0].set(xlim=[1, 10], xlabel='Aspect Ratio')
axes[0].set(ylim=[2e-16, 12e-16], ylabel='Energy ($J/m^2$)')
axes[0].legend()

#Plot equlibrium aspect ratio and corresponding energies
# for equivalent spherical radii up to 200nm
rs = np.linspace(1e-10, 2e-7, 100)
ars = se.eqAR_bySearch(rs, gamma, sf)
es = se.strainEnergy(sf._normalRadiiEquation(ars))

ln1 = axes[1].plot(rs, es*vaNI3TI4*6.022e23 / nNI3TI4,
    ↪ linewidth=2, color='C1', label='Strain Energy')
ln2 = ax2.plot(rs, ars, linewidth=2, label='Aspect Ratio')
axes[1].set(xlim=[0, 2e-7], xlabel='Equivalent Spherical Radius
    ↪ (m)')
axes[1].set(ylim=[25, 225], ylabel='Molar Strain Energy (J/mol)')
ax2.set(ylim=[0, 25], ylabel='Equilibrium Aspect Ratio')
axes[1].legend(ln1+ln2, [l.get_label() for l in ln1+ln2])

plt.tight_layout()
plt.show()
```

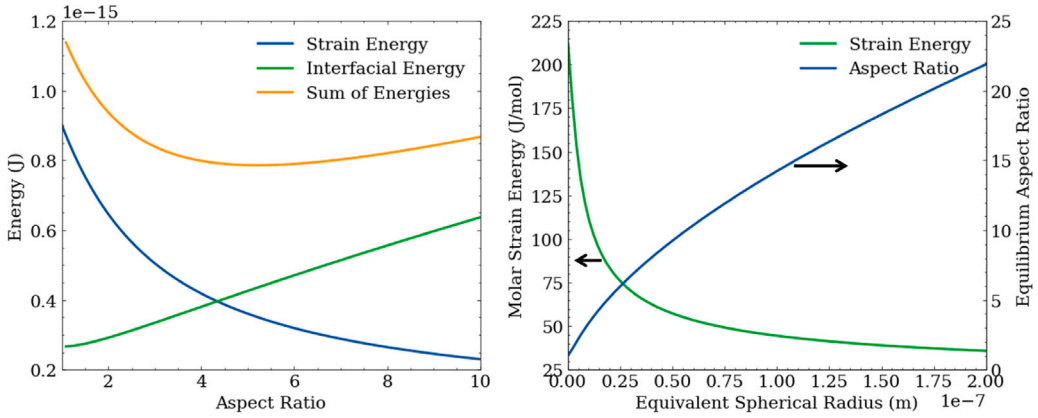

Fig. 12. (a) Strain and interfacial energy vs. aspect ratio for a Ni_4Ti_3 with equivalent spherical radius of 20 nm. (b) Equilibrium aspect ratio and corresponding strain energies (per mole of atoms) vs. equivalent spherical radius.

After inserting the model parameters [39,40] the precipitation behavior can be simulated. Two simulations are run: one assuming spherical precipitates and one accounting for elastic strain energy with variable aspect ratios. Fig. 13 shows the results of the two simulations. Accounting for elastic strain energy slows down nucleation and delays the transition from nucleation to coarsening. This is due to a significant reduction in the net free energy change associated with the phase transformation.

```
#Initialize model
model = PrecipitateModel(0, 1e8, 2e4)
model.setInitialComposition(xinit)
model.setInterfacialEnergy(gamma)
model.setTemperature(T)
model.setDiffusivity(Dni)
model.setVaAlpha(vaBCC, nBCC)
model.setVaBeta(vaNI3TI4, nNI3TI4)
model.setThermodynamics(therm)

saveNames = ['NiTi', 'NiTi_strain']

for i in range(2):
    if i == 1:
        #Input strain energy parameters
        model.reset()
        model.setStrainEnergy(se, calculateAspectRatio=True)

        #Set precpitate shape to plate
        # Aspect ratio is not supplied since
        # it'll be calculated from strain energy
        model.setAspectRatioPlate()

    model.solve(verbose=True, vIt=10000)
    model.save(saveNames[i])

fig, axes = plt.subplots(2, 2, figsize=(12,10))
labels = ['No elastic energy', 'With elastic energy']

for i in range(2):
    model = PrecipitateModel.load(saveNames[i] + '.npz')

    model.plot(axes[0,0], 'Precipitate Density', linewidth=2,
     ↪ timeUnits='h', label=labels[i])
    model.plot(axes[0,1], 'Volume Fraction', linewidth=2,
     ↪ timeUnits='h')
    model.plot(axes[1,0], 'Average Radius', linewidth=2,
     ↪ timeUnits='h')
    model.plot(axes[1,1], 'Driving Force', linewidth=2, timeUnits='h')

axes[0,0].legend()
axes[0,0].set(ylim=[1e15, 1e25], yscale='log')
axes[1,1].set(ylim=[0, 1.5e8])

plt.tight_layout()
plt.show()
```

As shown in Fig. 13, the inclusion of elastic energy effects significantly alters the behavior of the system. While most precipitate systems will not display such a dramatic difference under standard conditions, the complex residual stress profiles produced by rapid cyclic heating during additive manufacturing have been shown to alter precipitation behavior [41], and any attempt at modeling such heat treatments will likely require consideration of elastic energy.

4. Future prospects

Kawin is still in active development. At the time of writing, Kawin implements the features needed to run a variety of precipitation simulations considering factors such as particle shape, elastic energy and grain boundary precipitation. The goal of Kawin is to provide broad capabilities of kinetic-based Calphad modeling, either by implementing new features in the precipitate model or implementing other types of kinetic models.

The future implementation of a Lagrangian solution to the KWN model will enable Kawin to simulate several precipitation scenarios that are not currently possible with the Eulerian solution. The adaptive meshing measures implemented by Kawin's Eulerian solution are not currently capable of handling multi-modal precipitate distributions across multiple length scales, which is necessary to model some multi-stage heat treatments such as primary and secondary gamma prime in Ni-based alloys. Additionally, a Lagrangian solution will enable the modeling of time-dependent precipitate behaviors such as the decomposition of unstable phases.

The open-source nature of Kawin makes it easy to couple with other models, and the wide range of material properties impacted by precipitation means that there is significant potential for the development of additional physics modules. For example, currently in development is a strength model that couples with the KWN model and a diffusion model that can solve single phase diffusion or multi-phase homogenization problems. Further development prospects also include a user-friendly API that allows direct coupling of user-defined models with the kinetic models.

Kawin can also be easily implemented into other process frameworks, and is currently being coupled with thermal history, residual thermal stress, and evaporation models for the simulation of precipitation during additive manufacturing. KWN modeling offers numerous advantages over other methods of precipitation modeling that make it uniquely suited for use in a materials design framework. The KWN

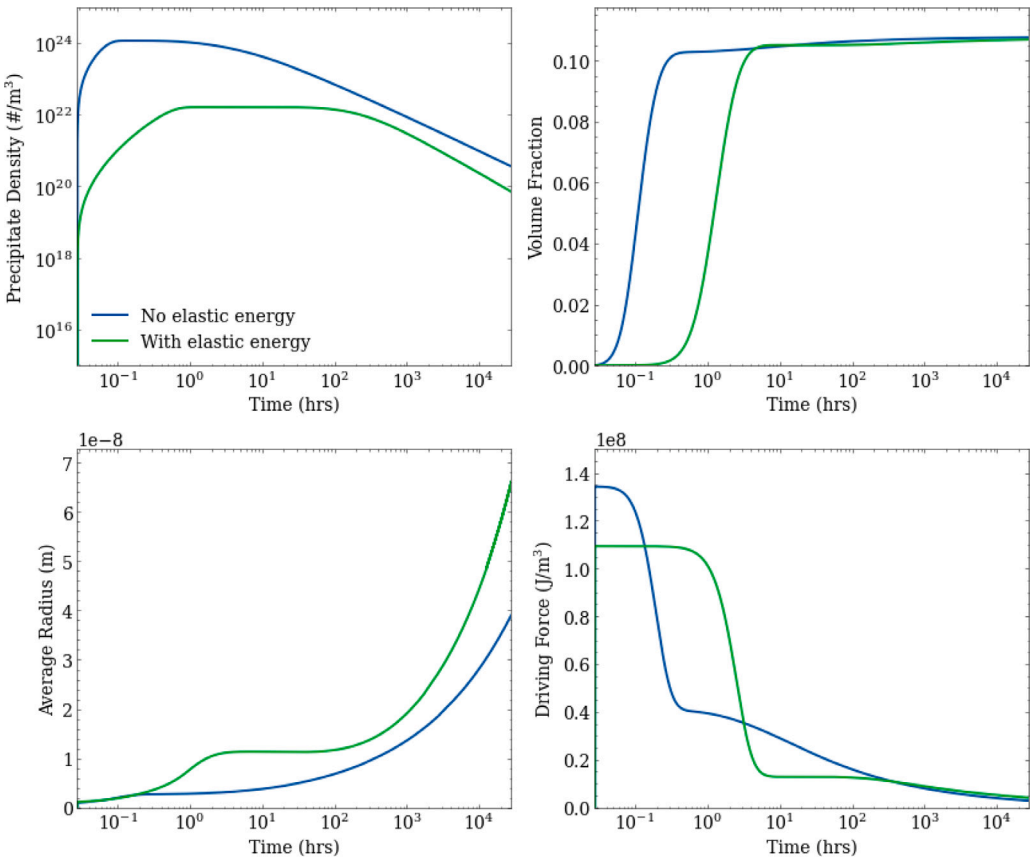

Fig. 13. Simulation results of the Ni-Ti system showing (a) Precipitate density, (b) volume fraction, (c) average equivalent spherical radius and (d) driving force vs. time with and without accounting for strain energy effects.

algorithm allows for the speed and computational efficiency of a traditional mean field model while still calculating the behavior of the entire particle size distribution similarly to a phase-field model. Phase-field modeling has recently been successfully applied to precipitation simulation in additively manufactured parts by running individual precipitation simulations in voxels throughout the part volume [15]. Because only the descriptive statistics of the precipitates in each voxel are relevant to the bulk properties of the part, a more efficient KWN model such as Kawin could be better suited to the task, particularly in materials design applications where many thousands of simulations must be performed.

5. Conclusions

Precipitation modeling is set to become a significant tool for materials and process design in the coming years. With the release of Kawin, an open-source implementation of the Kampmann–Wagner Numerical model of precipitation, a powerful, robust, and highly customizable tool capable of modeling precipitation in a wide variety of systems is freely available for the first time. Forty years of literature on precipitation modeling has been compiled and standardized, under conditions including (but not limited to) multicomponent, multiphase, non-isothermal heat treatments, non-spherical precipitates, heterogeneous precipitation, elastically influenced precipitation, and variable aspect ratios. Kawin has been shown to produce results in close agreement with both commercial software and literature experimental data across a wide range of precipitation reactions. Kawin offers stable simulations, built-in surrogate models, and data management tools, in an easy-to-use package, and with a modular design intended to maximize flexibility and encourage innovation. Download information and other details can be found at https://kawin.org.

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.

Acknowledgments

Part of this research was carried out at the Jet Propulsion Laboratory (JPL), California Institute of Technology, USA, under a contract with the National Aeronautics and Space Administration (80NM0018D0004). This research was supported by the JPL Strategic University Research Partnership (SURP) program, and by a NASA Space Technology Research Fellowship (80NSSC19K1142). RA also acknowledges the support from the National Science Foundation through Grant No. DMREF-2119103.

References

[1] D.S. MacKenzie, History of quenching, Int. Heat Treat. Surf. Eng. (2009).
[2] J.D. Verhoeven, L.L. Jones, Damascus steel, part II: Origin of the damask pattern, Metallography 20 (1987) 153–180.
[3] Hiromu Tanimura, Development of the Japanese sword, J. Metals (1980).
[4] P.D. Merica, R.G. Waltenberg, H. Scott, Heat treatment of duralumin, Sci. Pap. Bureau Standards 15 (1919) 271–316.
[5] V.R. Becker, W. Doring, Einetische Behandlung der Eeimbildurzg in ubersattdgtem Dampfern, Ann. Der Physik (1935).
[6] I.M. Lifshitz, V.V. Slyozov, The kinetics of precipitation from supersaturated solid solutions, J. Phys. Chem. Solids 19 (1–2) (1961) 35–50.
[7] J.S. Langer, J. Schwarz, Kinetics of nucleation in near-critical fluid, Phys. Rev. 21 (1980).
[8] R. Kampmann, H. Eckerlebe, R. Wagner, Precipitation kinetics in metastable solid solutions - theoretical considerations and application to Cu-Ti alloys, in: Materials Research Society Symposia Proceedings, Vol. 57, Materials Research Soc, 1987, pp. 526–542.

[9] D. Den Ouden, L. Zhao, C. Vuik, J. Sietsma, F.J. Vermolen, Modelling precipitate nucleation and growth with multiple precipitate species under isothermal conditions: Formulation and analysis, Comput. Mater. Sci. 79 (2013) 933–943.

[10] Bjørn Holmedal, Elisa Osmundsen, Qiang Du, Precipitation of non-spherical particles in aluminum alloys part I: Generalization of the Kampmann–Wagner numerical model, Metall. Mater. Trans. A: Phys. Metall. Mater. Sci. 47 (1) (2016) 581–588.

[11] Kaisheng Wu, Qing Chen, Paul Mason, Simulation of precipitation kinetics with non-spherical particles, J. Phase Equil. Diffusion 39 (5) (2018) 571–583.

[12] Qiang Du, Bjørn Holmedal, Jesper Friis, Calin D. Marioara, Precipitation of non-spherical particles in aluminum alloys part II: Numerical simulation and experimental characterization during aging treatment of an Al-Mg-Si alloy, Metall. Mater. Trans. A: Phys. Metall. Mater. Sci. 47 (1) (2016) 589–599.

[13] J.D. Robson, M.J. Jones, P.B. Prangnell, Extension of the N-model to predict competing homogeneous and heterogeneous precipitation in Al-Sc alloys, Acta Mater. 51 (2003) 1453–1468.

[14] Jan Herrnring, Bo Sundman, Benjamin Klusemann, Diffusion-driven microstructure evolution in OpenCalphad, Comput. Mater. Sci. 175 (August 2019) (2020) 109236.

[15] Jakub Mikula, Rajeev Ahluwalia, Robert Laskowski, Kun Wang, Guglielmo Vastola, Yong-Wei Zhang, Modelling the influence of process parameters on precipitate formation in powder-bed fusion additive manufacturing of IN718, Mater. Amp Des. 207 (2021) 109851.

[16] Ji Ma, Brian Franco, Gustavo Tapia, Kubra Karayagiz, Luke Johnson, Jun Liu, Raymundo Arroyave, Ibrahim Karaman, Alaa Elwany, Spatial control of functional response in 4D-printed active metallic structures, Sci. Rep. 7 (2017).

[17] Ian D. McCue, Gianna M. Valentino, Douglas B. Trigg, Andrew M. Lennon, Chuck E. Hebert, Drew P. Seker, Salahudin M. Nimer, James P. Mastandrea, Morgana M. Trexler, Steven M. Storck, Controlled shape-morphing metallic components for deployable structures, Mater. Des. 208 (2021) 109935.

[18] N. Ury, R. Otis, R. Neuberger, R. Arroyave, Kawin documentation, 2022, https://kawin.org.

[19] Richard Wagner, Reinhard Kampmann, Peter W. Voorhees, Homogeneous second-phase precipitation, Phase Transformations in Materials, 2005, pp. 309–407.

[20] Emmanuel Clouet, Modeling of nucleation processes, ASM Handb. 203–219.

[21] T. Philippe, P.W. Voorhees, Ostwald ripening in multicomponent alloys, Acta Mater. 61 (11) (2013) 4237–4244.

[22] Qing Chen, Johan Jeppsson, John Ågren, Analytical treatment of diffusion during precipitate growth in multicomponent systems, Acta Mater. 56 (8) (2008) 1890–1896.

[23] Richard Otis, Zi-Kui Liu, Pycalphad: CALPHAD-based computation thermodynamics in Python, J. Open Res. Softw. 5 (1) (2017) 1.

[24] Yue Li, Bjørn Holmedal, Hongxiang Li, Linzhong Zhuang, Jishan Zhang, Qiang Du, Precipitation and strengthening modeling for disk-shaped particles in aluminum alloys: Size distribution considered, Materialia 4 (2018) 431–443.

[25] H.J. Böhm, G.A. Zickler, F.D. Fischer, J. Svoboda, Role of elastic strain energy in spheroidal precipitates revisited, Mech. Mater. 155 (January) (2021).

[26] Ernst Kozeschnik, Precipitate nucleation, in: Precipitation Modeling, 2020, pp. 37–70.

[27] P.J. Clemm, J.C. Fisher, The influence of grain boundaries on the nucleation of secondary phases, Acta Metall. 3 (1) (1955) 70–73.

[28] J.D. Eshelby, The elastic field outside an ellipsoidal inclusion, Solid Mech. Appl. 287–295.

[29] A. G. Khachaturyan, S. Semenovskaya, T. Tsakalakos, Elastic strain energy of inhomogeneous solids 1 DECEMBER 1995-II, vol. 52.

[30] Thermo calc Software, Precipitation module (TC-PRISMA) user guide version 2016a, 2016.

[31] Chantal K. Sudbrack, Tiffany D. Ziebell, Ronald D. Noebe, David N. Seidman, Effects of a Tungsten addition on the morphological evolution, spatial correlations and temporal evolution of a model Ni-Al-Cr superalloy, Acta Mater. 56 (3) (2008) 448–463.

[32] N. Dupin, I. Ansara, B. Sundman, Thermodynamic re-assessment of the ternary system Al-Cr-Ni, CALPHAD 25 (2) (2001) 279–298.

[33] C. Borchers, Catastrophic nucleation during decomposition of Cu-0.9at.% Ti, Philos. Mag. A 79 (1999) 537–547.

[34] J. Wang, C. Leinenbach, L.B. Liu, H.S. Liu, Z.P. Jin, Assessment of the atomic mobilities in fcc Cu-Fe and Cu-Ti alloys, J. Phase Equilibria Diffusion 32 (1) (2011) 30–38.

[35] C.P. Wang, Y.S. Luo, Y. Lu, J.J. Han, Z. Shi, Y.H. Guo, X.J. Liu, Interdiffusion and atomic mobilities in bcc Ti-Ga and Ti-Cu alloys, J. Phase Equilibria Diffusion 38 (2) (2017) 84–93.

[36] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R.J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, SciPy 1.0 Contributors, Scipy 1.0: Fundamental algorithms for scientific computing in Python, Nature Methods 17 (2020) 261–272.

[37] E. Povoden-Karadeniz, P. Lang, P. Warczok, A. Falahati, J. Wu, E. Kozeschnik, Calphad modeling of metastable phases in the Al-Mg-Si system, CALPHAD 43 (12) (2013) 94–104.

[38] C.B. Ke, S. Cao, X.P. Zhang, Three-dimensional phase field simulation of the morphology and growth kinetics of Ni4Ti3 precipitates in a NiTi alloy, Modelling Simul. Mater. Sci. Eng. 22 (5) (2014).

[39] E. Povoden-Karadeniz, D.C. Cirstea, P. Lang, T. Wojcik, E. Kozeschnik, Thermodynamics of Ti-Ni shape memory alloys, Calphad: Comput. Coupl. Phase Diagrams Thermochem. 41 (2013) 128–139.

[40] Gustavo Tapia, Luke Johnson, Brian Franco, Kubra Karayagiz, Ji Ma, Raymundo Arroyave, Ibrahim Karaman, Alaa Elwany, Bayesian calibration and uncertainty quantification for a physics-based precipitation model of nickel-titanium shape-memory alloys, Trans. ASME, J. Manuf. Sci. Eng. 139 (7) (2017) 1–13.

[41] Kefeng Li, Zhi Wang, Kaikai Song, Khashayar Khanlari, Xu Sheng Yang, Qi Shi, Xin Liu, Xinhua Mao, Additive manufacturing of a Co-Cr-W alloy by selective laser melting: In-situ oxidation, precipitation and the corresponding strengthening effects, J. Mater. Sci. Technol. 125 (2022) 171–181.