

PAPER

μ 2mech: A software package combining microstructure modeling and mechanical property prediction

To cite this article: Albert Linda *et al* 2024 *Phys. Scr.* **99** 055256

View the [article online](#) for updates and enhancements.

You may also like

- [Variability and trends of aerosol properties over Kanpur, northern India using AERONET data \(2001–10\)](#)
Dimitris G Kaskaoutis, Ramesh P Singh, Ritesh Gautam et al.
- [A low-profile consolidated metastructure for multispectral signature management](#)
Nitish Kumar Gupta, Gaganpreet Singh, Harshawardhan Wanare et al.
- [Fabrication of a non-wettable wearable textile-based metamaterial microwave absorber](#)
Gaganpreet Singh, Harsh Sheokand, Kajal Chaudhary et al.



PAPER

RECEIVED
24 October 2023

REVISED
26 March 2024

ACCEPTED FOR PUBLICATION
28 March 2024

PUBLISHED
23 April 2024

μ 2mech: A software package combining microstructure modeling and mechanical property prediction

Albert Linda^{1,3}, Ankit Singh Negi^{1,3}, Vishal Panwar¹, Rupesh Chafle¹, Somnath Bhowmick¹ , Kaushik Das² and Rajdip Mukherjee¹

¹ Department of Materials Science and Engineering, Indian Institute of Technology Kanpur, Kanpur- 208016, India

² Department of Metallurgy and Materials Engineering, Indian Institute of Engineering Science and Technology, West Bengal, Shibpur, Howrah, 711103, India

³ These authors contributed equally to this work.

E-mail: rajdipm@iitk.ac.in

Keywords: phase-field model, finite element method, CALPHAD, graphical user interface, spinodal decomposition

Abstract

We have developed a graphical user interface (GUI) based package μ 2mech to perform phase-field simulation for predicting microstructure evolution. The package can take inputs from *ab initio* calculations and CALPHAD (Calculation of Phase Diagrams) tools for quantitative microstructure prediction. The package also provides a seamless connection to transfer output from the mesoscale phase field method to the microscale finite element analysis for mechanical property prediction. Such a multiscale simulation package can facilitate microstructure-property correlation, one of the cornerstones in accelerated materials development within the integrated computational materials engineering (ICME) framework.

1. Introduction

The age-old practice of trial and error-based experimental methods for developing new materials is being replaced by ICME-based methods. The latter reduces the cost and time required for developing new materials. While designing a new alloy, predicting its microstructure is one of the most crucial steps. Although elastic stiffness constants of materials are intrinsic (as they depend on bonding), the microstructure can be controlled by processing conditions, allowing the tuning of effective mechanical properties. Thus, a package combining microstructure modeling and mechanical property prediction can be convenient for alloy design.

Phase field simulation is the most popular technique used for microstructure modeling [1, 2]. In recent years, phase field simulations have been used to simulate complex microstructures evolution during different processes, such as solidification [3–6], spinodal decomposition [7–9], precipitate/grain growth [10–16], coarsening [17, 18], and the effect of an external field [19, 20].

One popular tool researchers utilize is MOOSE (Multiphysics Object Oriented Simulation Environment) [21]. This powerful software offers a range of capabilities for conducting complex multiphysics simulations. MOOSE is a flexible and extensible platform that empowers the development of phase field simulations. Its application extends to the study of microstructural evolution and dendritic growth [22]. Another valuable resource in this field is OpenPhase [23], a library dedicated to implementing the multiphase field method for simulating microstructure evolution during materials processing. OpenPhase excels in various applications, including grain growth and solidification, while offering high-performance capabilities through distributed memory simulation. The ‘Parallel algorithms for Crystal Evolution in 3D’ (PACE3D) [24], offers an integrated solution for addressing multi-physics applications in a parallel and efficient manner. The solver architecture within PACE3D encompasses diffuse interface methods, grain growth, grain coarsening, solidification, fluid dynamics, mechanical interactions, and electrochemistry. Another recently developed tool is MicroSim [25]; it comprises multiple modules, encompassing a grand-potential based solver, Kim Kim Suzuki (KKS) [26] models for OpenCL [27] and CUDA [28], a Cahn-Hilliard model with FFTW, and OpenFoam-based solvers [29] for

multiphysics simulations. MicroSim incorporates a user-friendly graphical interface that enables users to easily create input files, select solvers, and visualize simulation results using Paraview [30]. Despite the advantages provided by these tools, certain limitations hinder their accessibility. MOOSE and OpenPhase, for instance, have steep learning curves, which can pose challenges for programmers or researchers seeking user-friendly interfaces for their simulations. Additionally, the absence of built-in graphical user interfaces (GUIs) to visualize output in these tools can be cumbersome for new users.

In studying microstructures and their impact on material properties, OOF2 (Object-Oriented Finite Element Analysis) [31] stands out as specialized software. By employing finite element analysis techniques, OOF2 enables researchers to simulate the intricate behavior of complex structures at the microscopic level. Its user-friendly interface facilitates the definition of material properties, boundary conditions, and loading conditions. Moreover, OOF2 offers a comprehensive range of features for analyzing and visualizing microstructures, encompassing grain boundaries, phase distributions, and stress/strain fields. By simulating the relationship between microstructure and mechanical properties, OOF2 facilitates investigations into the influence of diverse microstructural features on material behavior. Two other prominent software packages commonly employed for finite element analysis and microstructural studies are ABAQUS [32] and ANSYS [33]. ABAQUS, a widely used commercial software, offers advanced capabilities for investigating the mechanical properties of materials, including microstructures. On the other hand, ANSYS encompasses a suite of simulation tools, including finite element analysis, with specific modules such as ANSYS Mechanical and ANSYS Multiphysics catering to the study of microstructural behavior. These software packages enable researchers to simulate material deformation, phase transformations, and grain boundary effects. Notably, both ANSYS and ABAQUS provide the ability to perform finite element analysis of microstructures, incorporating crystal plasticity through the use of CPFEM (Crystal Plasticity Finite Element Method) [34] by utilizing their respective subroutines.

Microstructure-based mechanical property prediction involves scale bridging, from mesoscale phase-field modeling to macroscale finite element analysis. While stand-alone packages are available for both the length scales, a package combining them does not exist, to the best of our knowledge. Motivated by this, we develop μ 2mech software. It can simulate the microstructure of materials using phase-field techniques. The software uses various open-source libraries for numerical computation, visualization, and data analysis. The package can take inputs from *ab initio* calculations, and CALPHAD [35] software like ThermoCalc [36] for quantitative microstructure prediction. Its features include the ability to simulate binary microstructures for spinodal decomposition and precipitate growth, visualization of microstructures in 2D and 3D, and calculating particle size and volume. Additionally, μ 2mech can be coupled with OOF2 to study the mechanical behavior of the simulated microstructures.

The paper aims to comprehensively describe the μ 2 mech package, including a brief description of CALPHAD integration, numerical implementation of the phase field technique, and installation. We discuss an example of FeCr binary alloy, showcasing the capabilities of μ 2 mech to simulate the microstructure and its integration with OOF2 for mechanical property prediction. The example of FeCr binary alloy also serves as a manual for users, providing guidelines regarding various inputs to the code and outputs generated. Finally, we discuss the impact that μ 2mech can have on the materials science and engineering community.

2. Software description

The software package comprises three main steps: preprocessing, partial differential equation (PDE) solver, and post-processing. Figure 1(a) schematically shows the software's architecture, process flow, and capabilities. During preprocessing, users must select the calculation type (2D or 3D) and other input parameters. Then, the software numerically solves the governing PDEs and generates the microstructure evolution as a function of time. In the post-processing stage, users can visualize the evolution of the microstructure and do many analyses (like particle size distribution and composition profile). Finally, users can create input files for finite element analysis via OOF2, allowing researchers to simulate materials' mechanical behavior under various loading conditions. Coupling with OOF2 is chosen as it can account for the material property's nonlinearity by generating a mesh according to the morphology of the underlying microstructure.

The architecture comprises two main components: GUI (pre and post-processing) and the PDE solver. The latter is entirely written in the programming language C. The discretized equations are solved using the Fourier spectral method, and it employs the FFTW library [37] to perform these calculations. In the following section, we describe some technical details related to the phase-field method, like CALPHAD integration and the numerical scheme to solve the Cahn-Hilliard equation. The GUI component, which provides a user-friendly interface for interacting with the software, is written in Python3 by utilizing the PySide2 framework [38]. The μ 2mech package also incorporates several other libraries such as NumPy for matrix manipulation, Matplotlib

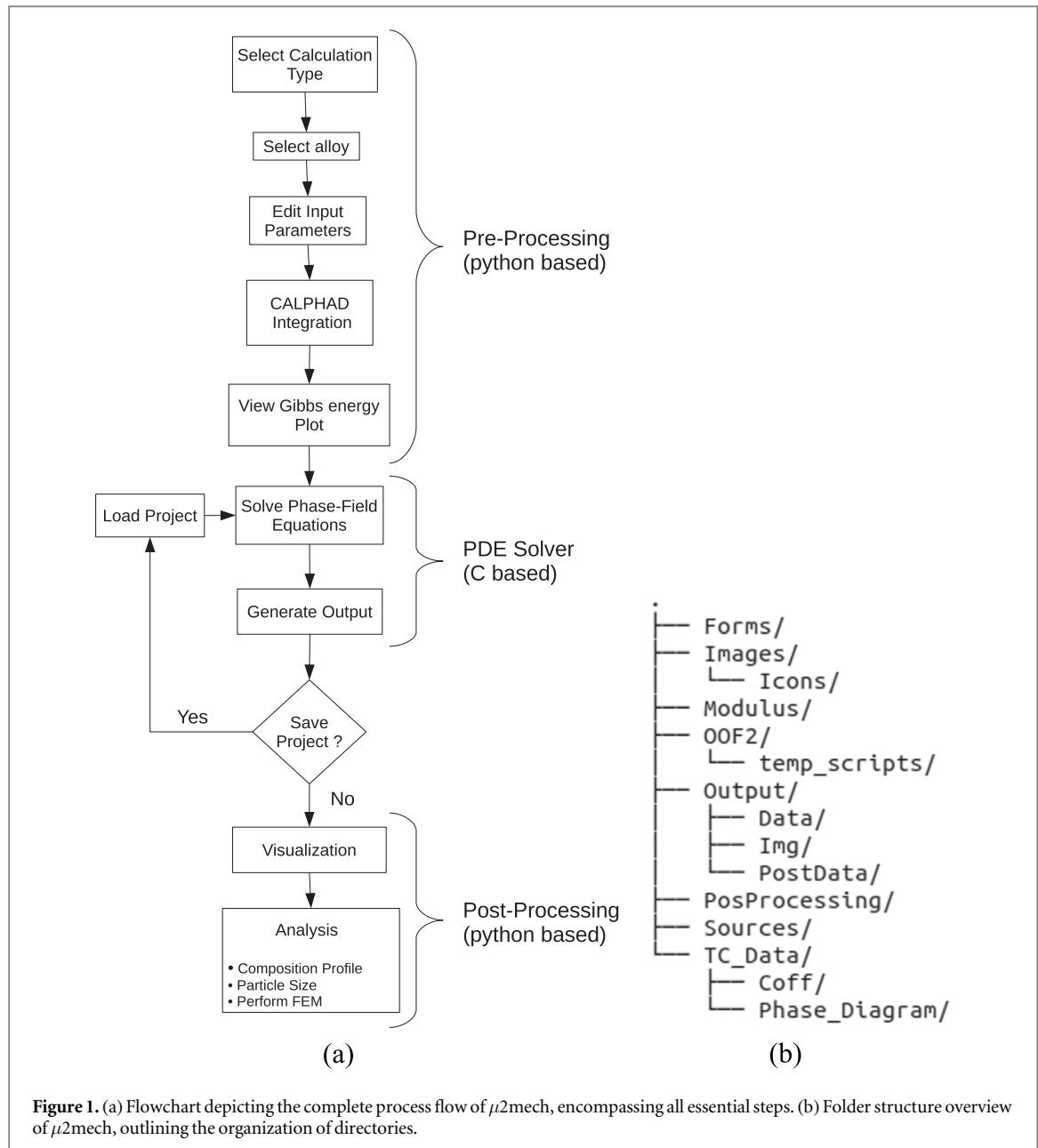


Figure 1. (a) Flowchart depicting the complete process flow of $\mu 2\text{mech}$, encompassing all essential steps. (b) Folder structure overview of $\mu 2\text{mech}$, outlining the organization of directories.

for plotting 2D microstructure intersections, PyVistaQT for visualizing vtk files, PyYAML for project configuration and saving, and FFmpeg [39] for generating microstructure evolution animations. For its coupling with OOF2 [31, 40], $\mu 2\text{mech}$ prepares an input Python script once the microstructure image is passed along with other input parameters.

Detailed installation instructions are given in appendix A. Post-installation, the directory structure is illustrated in figure 1(b). The C source codes and their compiled binaries for the phase field calculations are found in the ‘Sources’ folder. A *Makefile* has been provided to recompile the source files. Typically, recompilation is not needed unless modifications have been made in one of the source files. The ‘Coeff’ sub-folder within the ‘TC_Data’ directory encompasses thermodynamic details pertinent to the binary alloys, which are crucial for conducting the phase field calculations. These details are stored in the csv format, which follows the naming convention ‘A-B.csv’, where A and B are the constituent elements. An example for the Fe-Cr alloy is illustrated in table B1. Contained within the ‘Phase_Diagram’ sub-folder of the ‘TC_Data’ directory are the phase diagrams, serving as a visual reference and not directly involved in the calculation process. The directory labeled as ‘Output’ temporarily holds the resulting data from the computations of phase field. The ‘Post_Processing’ directory temporarily stores the composition plot generated during the post-processing stage. Other folders like ‘Forms,’ ‘Images,’ ‘Modulus,’ and ‘OOF2’ are mainly responsible for the GUI part of $\mu 2\text{mech}$, and it is recommended not to modify these directories.

3. Algorithm: phase-field modeling

The Cahn-Hilliard equation is a fourth-order, nonlinear partial differential equation crucial in studying phase separation and microstructure evolution in multi-component systems. It has been extensively applied to model the time-dependent behavior of various materials, including alloys, polymers, and colloids [41]. One can write the Cahn-Hilliard equation as,

$$\frac{\partial c}{\partial t} = \nabla \cdot \left[M(c) \nabla \left(\frac{\delta F}{\delta c} \right) \right], \quad (1)$$

where c is composition, t is the time, $M(c)$ is the mobility function. F represents the free energy functional, and $\frac{\delta F}{\delta c}$ is the variational derivative of F with respect to c . The free energy functional typically comprises a bulk free energy term $f_b(c)$ and a gradient energy term $\kappa |\nabla c|^2$. In our qualitative model, mobility M is treated as a constant. The variable mobility $M(c)$ is implemented for the quantitative models for real alloys, e.g., Fe-Cu and Fe-Cr alloys. Please see appendix C for the calculation of mobility using self-diffusivity values and numerical implementation of the Cahn-Hilliard equation with variable mobility.

The Fourier spectral method is a powerful numerical technique for solving the Cahn-Hilliard equation in 2D and 3D configurations, offering a robust and efficient approach to studying complex microstructure systems [42]. It relies on the Fourier transform to convert the spatial derivatives into algebraic operations in the frequency domain [42]. This approach allows for efficient and accurate computation of the solution. The Fourier spectral method involves discretizing the time variable using a suitable time-stepping scheme, such as the semi-implicit or fully implicit method [43]. Then, the Fourier transform is taken with respect to the spatial variables x , y , and z (for 3D) or just x and y (for 2D), yielding an equation in the frequency domain (assuming constant mobility M),

$$\frac{\partial \tilde{c}}{\partial t} = -M [k^4 \tilde{c} + k^2 \tilde{g}(c)], \quad (2)$$

where \tilde{c} and $\tilde{g}(c)$ are the Fourier transforms of c and $g(c) = \frac{\partial f_b(c)}{\partial c}$, respectively, and $k^2 = k_x^2 + k_y^2 + k_z^2$ (in 3D) or $k^2 = k_x^2 + k_y^2$ (in 2D), where \mathbf{k} is the vector in Fourier space.

The next step is to solve the frequency-domain equation for $\tilde{c}(t)$ and update the solution at each time step. Starting from the initial condition $c(\mathbf{x}, 0)$ and its discrete Fourier transform $\tilde{c}(0)$ with respect to the spatial variables, one studies the time evolution by updating $\tilde{c}(t)$ using an appropriate time-stepping scheme, like the semi-implicit Euler method,

$$\tilde{c}^{n+1} = \frac{\tilde{c}^n - \Delta t \cdot M \cdot k^2 \tilde{g}(c^n)}{1 + \Delta t \cdot \kappa \cdot M \cdot k^4}, \quad (3)$$

where Δt is the time step, and the superscript n denotes the time level and κ is gradient energy coefficient. The mobility function $M(c)$ can also be incorporated into the time-stepping scheme. Once the updated solution \tilde{c}^{n+1} is obtained in the frequency domain, the inverse Fourier transform is applied to recover the solution $c(\mathbf{x}, t^{n+1})$ in the spatial domain [44]. One repeats this procedure until the desired final time is reached.

One can efficiently implement the Fourier spectral method for solving the Cahn-Hilliard equation using the Fast Fourier Transform (FFT) algorithm, which significantly reduces the computational cost compared to traditional finite difference or finite element methods [45, 46]. Moreover, the spectral method offers high accuracy and preserves the smoothness of the solution, making it well-suited for studying the evolution of complex microstructures [47].

4. Illustrative example

To demonstrate the capabilities of $\mu 2mech$, we present an example of a binary system, Fe-Cr, using Gibbs free energy data from CALPHAD calculations [technical details provided in appendix B]. The main window of $\mu 2mech$ is illustrated in figure 2. As shown in the diagram, users click the ‘New’ submenu under the ‘File’ menu to start a new calculation, which opens the **New** dialog box (figure 3). In this dialog box, the users select either **Cahn Hilliard 2D alloy** or **Cahn Hilliard 3D alloy** from the ‘Calculation type ↓’ drop-down menu, as shown in the top row of figure 3. This action loads the default parameters for calculations. Next, the users can click the **Edit Parameters** button near the top right of the main $\mu 2mech$ window (figure 2). This action opens the **Edit Parameters** dialog box, as shown in the bottom row of figure 3. Next, users select **Fe-Cr** from the ‘Alloy ↓’ drop-down menu (figure 3). Users can visualize the phase diagram for the given system by clicking on the adjacent **View Phase Diagram** button. The users then select the temperature

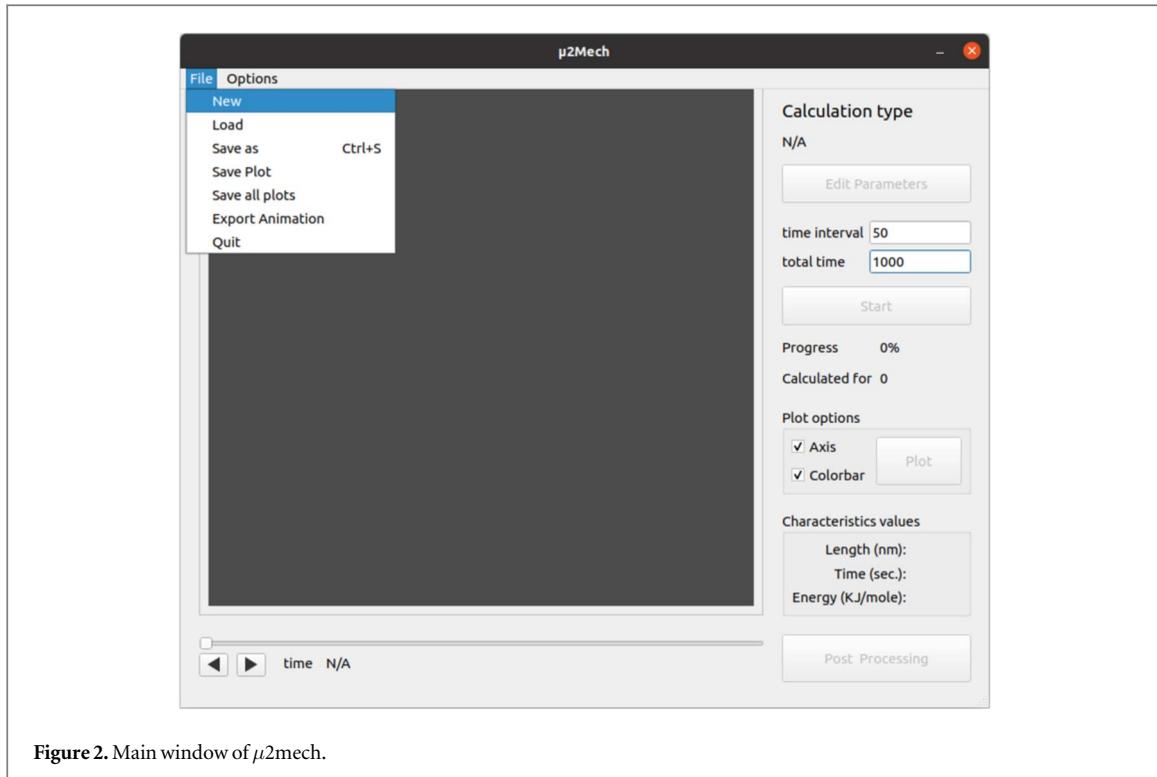


Figure 2. Main window of μ 2mech.

from the ‘Temperature ↓’ drop-down menu and view the Gibbs free energy versus composition plot by clicking the adjacent **View G vs X plot** button (figure 3). A point on the curve fixes the *average composition* for the alloy. When the temperature is 400 °C for the Fe-Cr system, the spinodal phase forms within the $0.27 \leq X \leq 0.75$ (inflection points on free energy curve, figure 4).

Before continuing with the Fe-Cr system, let us briefly discuss how to add a new alloy system, say **Fe-Cu**. First, one has to generate a file named *A-B.csv*, where A and B represent the specific elements in the alloy (A = Fe and B = Cu in this case). The designated directory path for saving this file is outlined in figure 5. The CSV file contains information obtained from CALPHAD as well as some dimensional parameters, structured in the format shown in table B1 in appendix. One can also add a phase diagram in the path ‘TC_Data/Phase_Diagram’ named *A-B.png*, which can be used as a display, although not essential. If the application is already running, one has to restart for the changes to take effect. Figure 5 illustrates that the name of the new alloy FeCu appears after adding the *Fe-Cu.csv* file in the designated directory.

If required, the users can modify other parameters before running the simulation [bottom row of figure 3]. The list of parameters includes ‘Fluctuation’ (a Gaussian noise added on top of the average composition to generate the initial composition field), ‘lx, ly, lz’ (system size), ‘delta x, delta y, delta z’ (grid size), ‘delta t’ (time interval), ‘Mobility’, and ‘Kappa’. The symbols used are consistent with equations (1), (2), (3). All the parameters are self-explanatory and consistent with section 3. In this example, we choose $lx = ly = 400$ with $\delta x = \delta y = 0.4$, which implies a mesh size of 100×100 . ‘Fluctuation’ 0.0001 defines the initial composition field as ‘Average composition’ $0.4 \pm$ a Gaussian noise with width 0.0001 at every grid point of the mesh. In such instances, only equations (2) and (3) need to be modified.

After editing the parameters, users return to the main window of **μ 2mech** and set the ‘total time’ for the phase-field simulation to run and ‘time interval’ after which simulated microstructures are saved for visualization and other analysis. In this example, the total time for microstructure evolution is 1000, and the interval for saving microstructures is 50 (figure 2). Next, the users can choose between two options: (i) click the ‘Save as’ submenu under the ‘File’ menu to save the project, transfer the files and run the calculations on a *remote* machine, such as a high-performance cluster (HPC) or (ii) click the **Start** button to perform the calculations *locally* on a desktop/workstation.

To run the calculations in a remote HPC, the user first transfers the entire project folder (saved in the *.pf format) to the HPC (figure 6). Assuming *slurm* workload manager is installed in the HPC, a job submission command may look like *sbatch slurm.sh*, where *slurm.sh* is a bash file containing the scheduler parameters and commands required for calculation. However, the exact command will depend on the installations in an HPC. Once completed, the user transfers the files back to the local machine and uses the ‘Load’ submenu under the ‘File’ menu in the main **μ 2mech** window to load the finished calculations as a project to do further analysis.

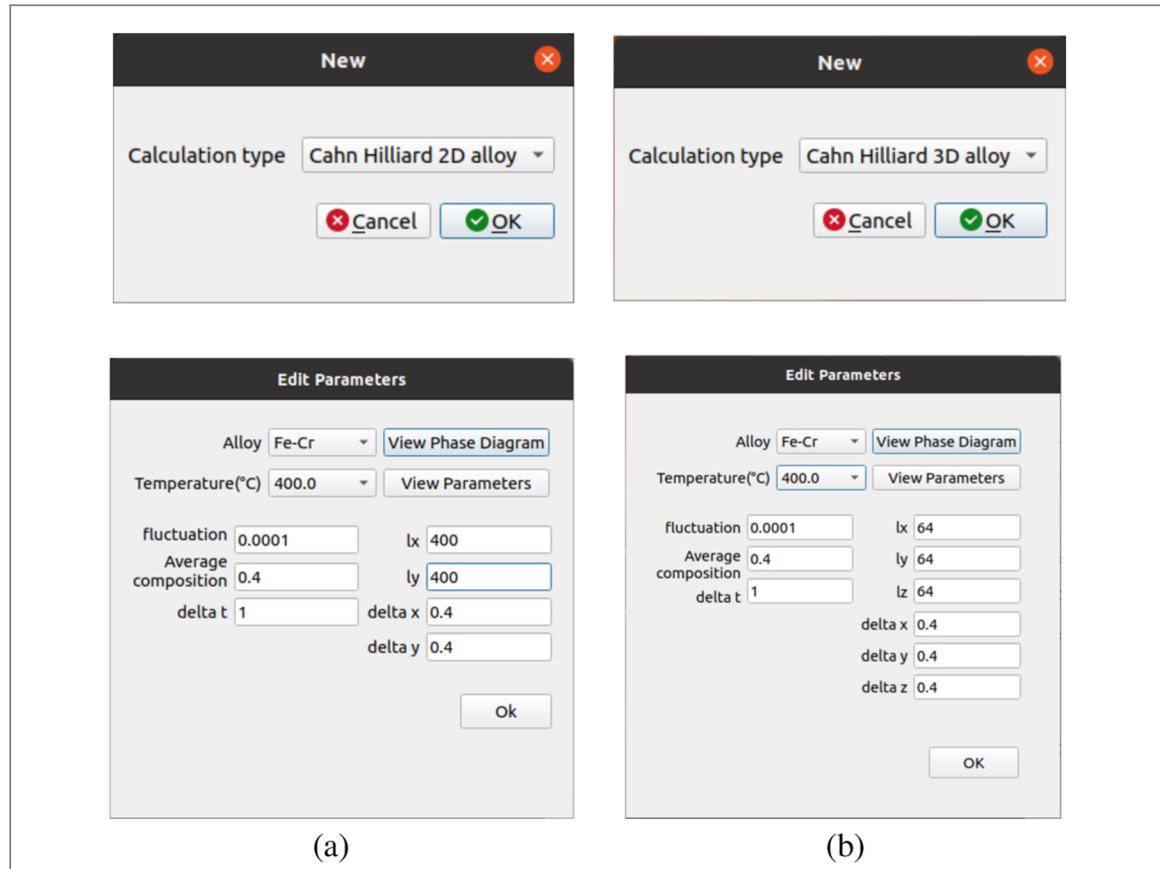


Figure 3. Top panels show the calculation types, (a) 2D and (b) 3D phase field simulations. The bottom panels illustrate the simulation parameters, such as alloy, temperature, average composition, system size, mobility, kappa, and time interval. These parameters play a crucial role in defining the simulation setup and needs to be adjusted carefully.

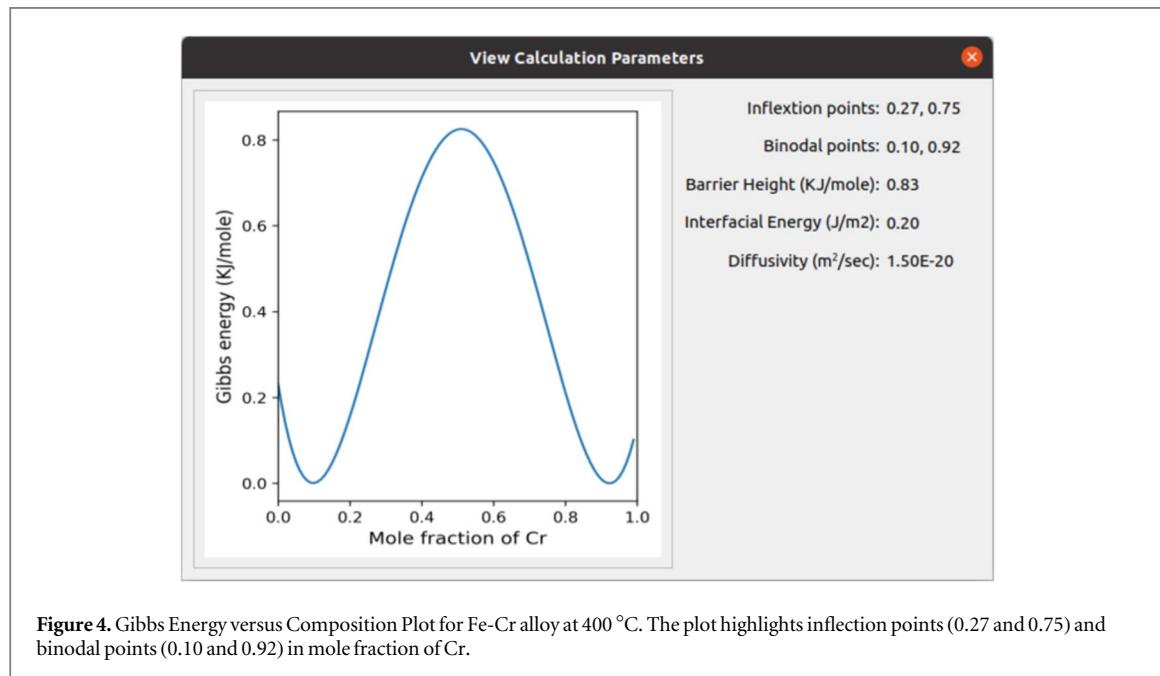


Figure 4. Gibbs Energy versus Composition Plot for Fe-Cr alloy at 400 °C. The plot highlights inflection points (0.27 and 0.75) and binodal points (0.10 and 0.92) in mole fraction of Cr.

To run the calculations locally, the user clicks the **Start** button in the main **μ 2mech** window (figure 2). The status bar below the **Start** button indicates the progress of the calculations, and the calculation can be paused and resumed as needed. ‘Save as’ and ‘Load’ options are available under the ‘File’ menu in the main **μ 2mech** window (figure 2). After the run, the user can save the project files using the

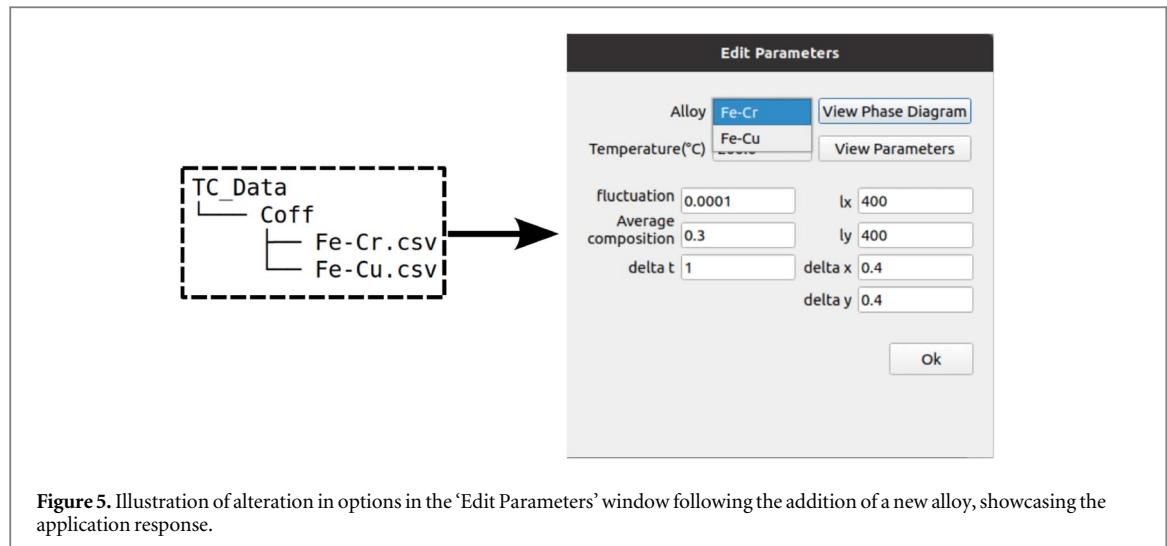


Figure 5. Illustration of alteration in options in the ‘Edit Parameters’ window following the addition of a new alloy, showcasing the application response.

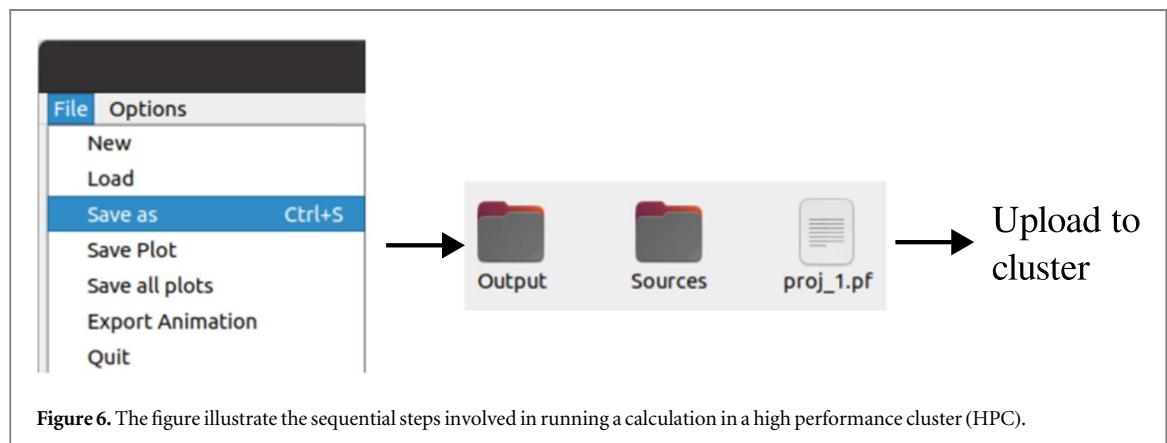


Figure 6. The figure illustrate the sequential steps involved in running a calculation in a high performance cluster (HPC).

‘Save as’ option, which saves the input and output files. The file extension for saving the project is *.pf. When loading the project, the corresponding *.pf file must be selected. After loading the project, the users can either resume the calculations for more time steps or perform post-processing. To perform additional time evolution, one has to increase the total time and click the **Resume** button in the main **μ2mech** window (figure 7).

Users can visualize microstructures in the main **μ2mech** window itself. The microstructures for different time steps are accessible by clicking the left/right arrow key or sliding the slider. Figure 7 depict the 2D and 3D microstructure at 600th time step. Users can customize the color scheme for the microstructure visualization by selecting the ‘Plot Colors’ option within the ‘Options’ menu (figure 7), which opens a new dialog box **Plot Colors**, as depicted in figure 8. The simulated microstructures can be saved (using ‘Save Plot’ or ‘Save all plots’) or exported as an animation (using ‘Export Animation’). These options are available under the ‘File’ menu in the main **μ2mech** window (figure 2).

The users can start the post-processing exercise by clicking the **Post Processing** button located at the bottom right of the primary **μ2mech** window (figure 7), which opens the **Post Processing** dialog box, as shown in figure 9. Available options are (a) particle size distribution, (b) composition profile analysis, and (c) property prediction. Particle size distribution at each time step can be analyzed by selecting the **Particles info.** button. Particle size information appears in a small dialog box, as shown in figure 9. For the composition profile analysis, users can choose a point by clicking on the microstructure in the main **μ2mech** window. For example, the coordinate of the chosen point is (X, Y) = (166,359) in figure 9. The composition profile is drawn along a horizontal or vertical line passing through the selected point. As shown in figure 9, this example draws the composition profile along the horizontal (x) direction. Clicking on the **Generate Plot** button creates the

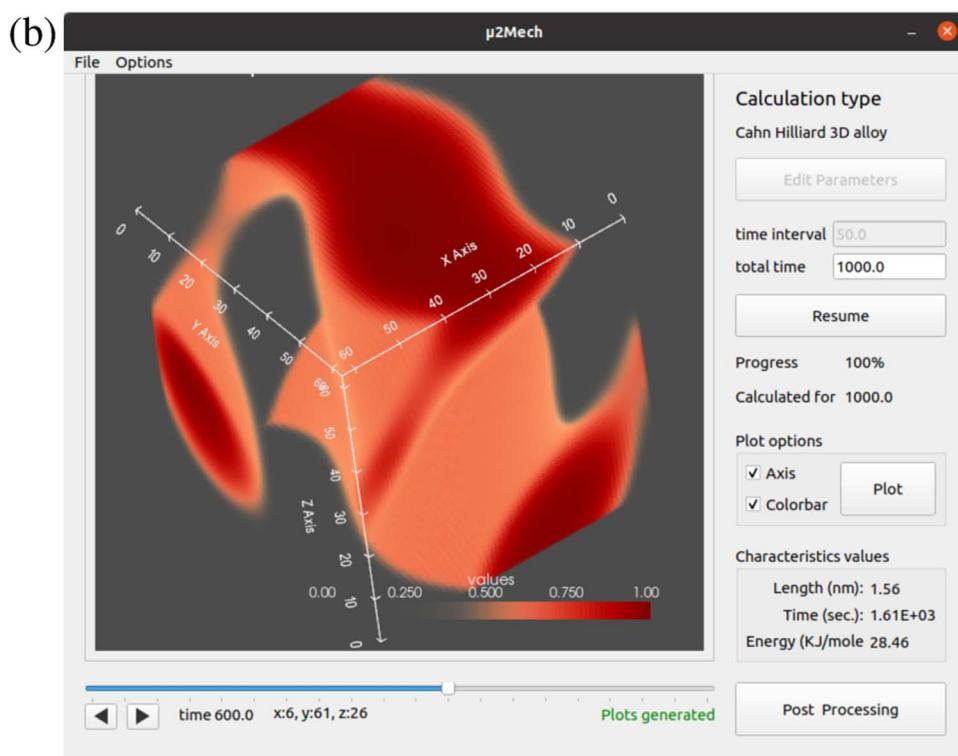
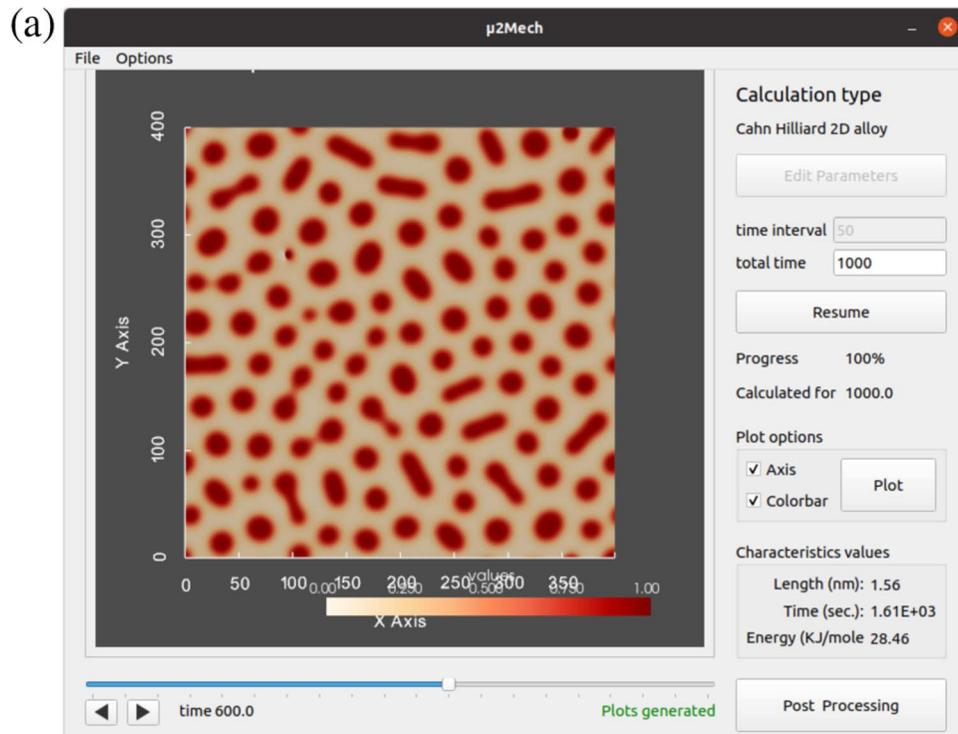


Figure 7. (a) A 2D and (b) a 3D microstructure of a binary alloy are displayed at the 600th time step, following the completion of calculation. Microstructures at different time steps can be accessed using the time slider at the bottom.

composition plot along the designated direction, as illustrated in figure 9. Users can visualize the time evolution of the composition profile by clicking the left/right arrow key or sliding the slider (located at the bottom of the *Post Processing* subwindow). Users can also save the composition profile(s) using the ‘Save Plots’ or ‘Save all plots’ options and make an animation using the ‘Export Animation’ option by clicking the respective **Export** buttons in the **Post Processing** dialog box (figure 9).

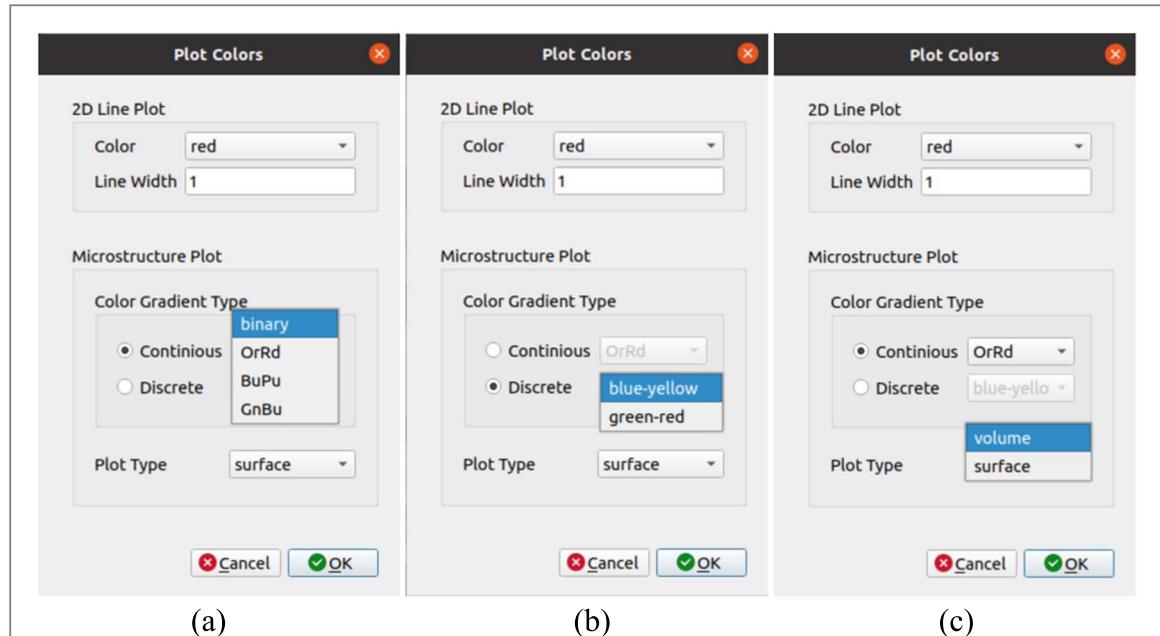


Figure 8. The Plot Colors dialog box for adjusting the appearance of line and microstructure plots. For 2D line plots, color selection is available. In the case of microstructure plots (both 2D and 3D), users can choose between two plot gradient types: continuous and discrete with their color gradient. Additionally, surface/volume plot options are provided for visualizing outer planes and partially transparent isosurfaces, respectively.

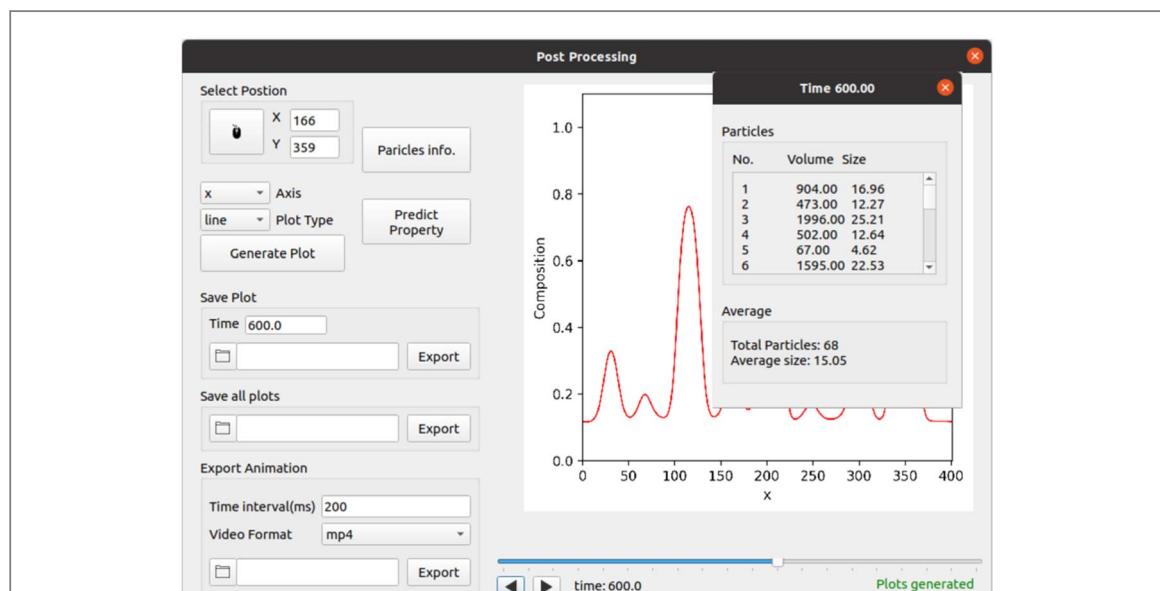


Figure 9. The Post Processing dialog box for 2D microstructure analysis. The main dialog box contains options for composition profile, particle size, and exporting plots and animations related to the composition profile. By clicking the Particles Info. button, a small additional dialog box appears atop the previous one, displaying detailed particle size information.

The property prediction feature of the post-processing module of μ 2mech allows users to predict elastic properties by performing Finite Element Analysis (FEA) directly over the simulated microstructure (obtained via phase field modeling). The current version of μ 2mech allows the prediction of elastic properties from 2D microstructures containing only two phases, which is accomplished by integrating μ 2mech with OOF2, an open-source object-oriented finite element method software developed by NIST, USA [31].

First, let us describe the general operation of OOF2 briefly. OOF2 takes a microstructure and allows users to identify phases based on the constituent phases' pixel values (or gray-scale values). Next, OOF2 needs the elastic properties of the individual phases to be added by the user. If the phases are isotropic, only two material properties, namely elastic modulus and Poisson's ratio, are needed as input, while based on the degree of

anisotropy, multiple components of the elastic stiffness matrix, known as the elastic constants, are needed. Once the phases are identified, it allows the users to build a ‘skeleton’ that identifies the boundaries of the phases so that a mesh with elements and nodes conforming to the phase boundaries can be generated. Building the skeleton requires initial input from the user regarding the number of elements along the horizontal edges (or x-direction) and along the vertical edges (or y-direction). Mesh generation involves further refinement of the elements so that the elements conform to the phase boundaries, as described before. Before meshing, OOF2 allows users to choose triangular or quadrilateral elements, similar to those found in ABAQUS, a commercial finite element software. Mesh generation is followed by the assignment of boundary conditions. Displacement boundary conditions are imposed on the microstructure’s left, right, top, and bottom edges. The displacements are applied in terms of pixels. This is followed by solving the finite element model using the default iterative solver available with OOF2. Post-processing involves a calculation of the area average of the required stress and strain components, from which the elastic properties can be calculated.

These operations are automated via a python code that is integrated within the μ 2mech which calls OOF2 (which should be separately installed), loads the specified virtual phase-field model-generated microstructure, applies the boundary conditions for a simulated tensile test (along the horizontal axis or the vertical axis), solves the finite element model, and performs post-processing operations, calculating the elastic properties and saving these properties in a file at a chosen destination. User inputs are required for users select either material properties of the constituent phases of the microstructure, for the selection of the type of element, for quantifying the displacements applied on the boundaries of the microstructure, and for the choice of destination for saving the output files. It is important to note here that only a small set of features of OOF2 software are being utilized by μ 2mech through a Python code that automates virtual tensile tests in an elastic regime for a microstructure containing two phases, one of which is continuous, while the other is discontinuous. To explore the full features of OOF2, it is recommended that the users export the generated microstructural image files and open these files directly in OOF2, i.e., outside the environment of μ 2mech. The following steps must be followed to initiate the evaluation of the elastic properties (namely elastic moduli and Poisson’s ratio) in μ 2mech.

First, the users click on the  button in the **Post Processing** dialog box, which opens another dialog box **OOF2 Calculation**, as shown in figure 10. In this dialog box, the users click on the mouse-shaped button , which allows the users to click on the continuous phase in the main **μ 2mech** window displaying the microstructure. μ 2mech uses the *burn* and *invert* options associated with pixel-selection methods of OOF2 for phase identification. The *burn* operation works by selecting contiguous pixels of the same color (i.e., RGB or gray-scale pixel values) for one phase until it encounters pixels with different colors (or pixel values). The *invert* option can be used only if the second phase is discontinuous, and it selects pixels other than those stored using the *burn* operation of the continuous phase. These operations happen in the background. On clicking on the continuous phase, only the *x* and *y* coordinate values of the point of mouse-click are displayed in the **OOF2 Calculation** dialog box next to the  button.

Next, the users enter the elastic constants for both phases under the header ‘Elastic Constants (in GPa)’. At present, μ 2mech can only work with microstructures where one or both phases are either isotropic or cubic in terms of material symmetry. For an isotropic phase, the values of the elastic constants can be calculated from Young’s modulus and Poisson’s ratio. Elastic constant values are either obtained from the experimental data or from *ab initio* calculations [48, 49].

Next, the users enter the number of elements along the horizontal and vertical edges of the microstructure in the boxes beside ‘X elements’ and ‘Y elements’ under the header ‘Mesh’. This is followed by choosing the element type by clicking on the drop-down menu for ‘Element type ↓’ under the header ‘Mesh’ and selecting ‘TriSkeleton’ or ‘QuadSkeleton’. OOF2 allows only two geometries of elements, viz. triangular and quadrilateral, with triangular elements (‘TriSkeleton’) as the default option. This initiates the process of building the skeleton and is followed by meshing. OOF2 generates an adapted mesh using its skeleton modification tools to account for the microstructure’s geometrical heterogeneity. An example of meshing has been illustrated in figure 11.

Next, the users enter the boundary conditions for a simulated tensile test (in elastic regime) along the horizontal (or x) direction under the header ‘Boundary Conditions (in Pixel)’. Displacements are applied in terms of number of pixels in OOF2. OOF2 considers displacements along the +x-direction positive and displacements along the -x-direction negative. Hence, for a tensile test along the horizontal direction, normal displacements need to be applied along the outer normals, i.e., perpendicular to the left and right edges of the microstructure. Hence, the normal displacement applied on the left edge has a negative value, while that on the right edge has a positive value. It is recommended that the displacements along the left and right edges should have the same magnitude (i.e., absolute value without the positive or negative signs). The users should enter a

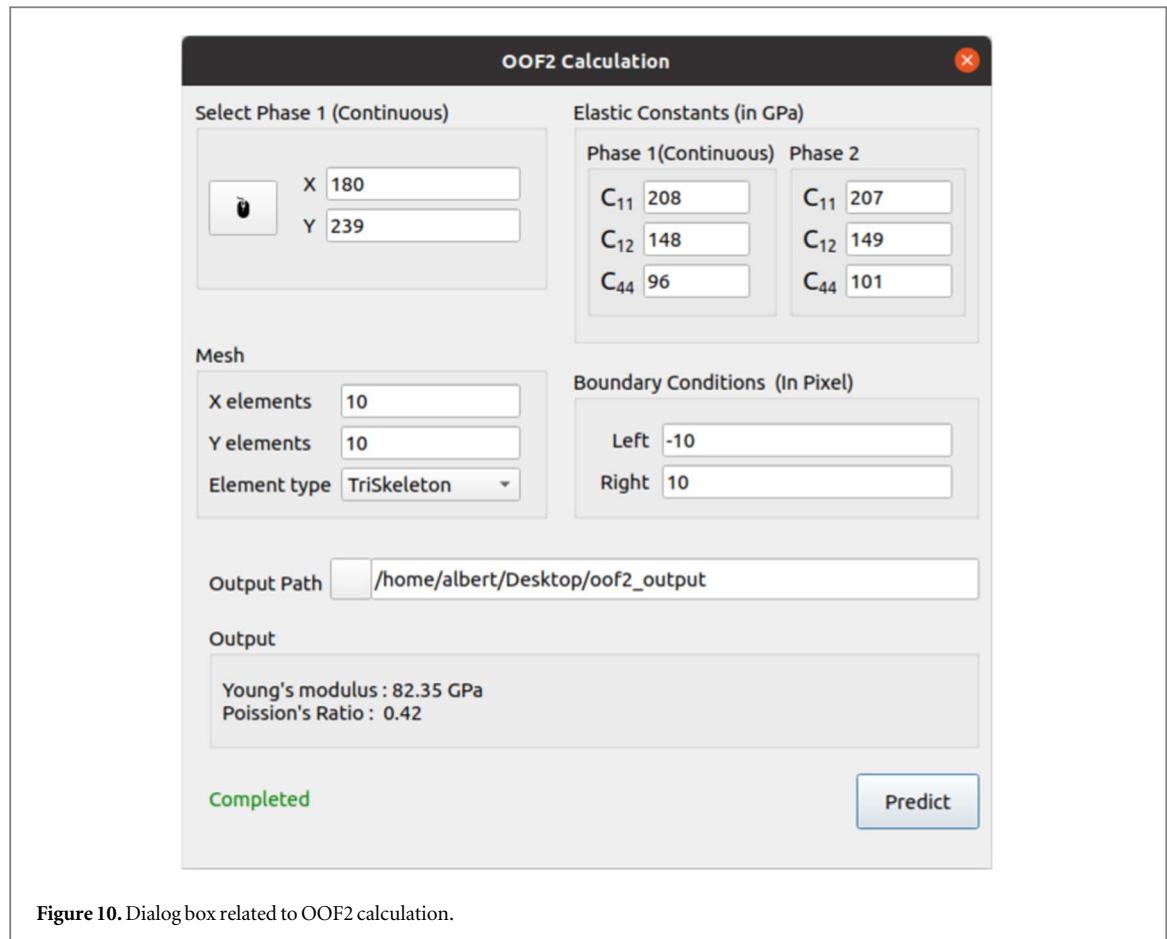


Figure 10. Dialog box related to OOF2 calculation.

negative value for the displacement in the box beside ‘Left’ and a positive value (of the same magnitude) beside ‘Right’.

The users must specify the output path to store the output files from OOF2. OOF2 generates files with area-averaged stress and strain component values, from which the elastic properties are calculated. Finally, clicking on the **Predict** button initiates the FEM calculations. At this point, OOF2 starts running in the background. While the calculation is running, the status bar at the bottom left will show the status ‘Performing calculation’, and upon completion, output files for stress, strain, and mesh are generated at the specified output path. Once OOF2 operations are completed, the computed elastic properties, i.e., Young’s modulus and Poisson’s ratio, are displayed at the bottom of the dialog box.

5. Impact

Replacing the age-old trial-and-error method for developing new alloys using the ICME framework requires the integration of multiple time and length-scale simulations. The μ 2 mech package integrates microstructure prediction via the mesoscale phase field method to the microscale finite element analysis for mechanical property prediction. The package can quantitatively predict by taking inputs from CALPHAD and *ab initio* calculations. Such a multiscale package will be handy for the materials science and engineering community, especially those working in alloy development and microstructure engineering for modulating mechanical properties.

Some salient features of the μ 2 mech package are the following. The software’s architecture, including open-source libraries, makes it easy to use and accessible to a broader range of users. The graphical user interface makes it very easy to run the code, even for beginners. At the same time, it has the option to add a new binary alloy, making it valuable for advanced users as well. With the increasing application of artificial intelligence (AI) in phase-field modeling [50], a package like μ 2 mech will be convenient for relatively effortless microstructure generation in bulk, which can be further used for training machine learning (ML) models.

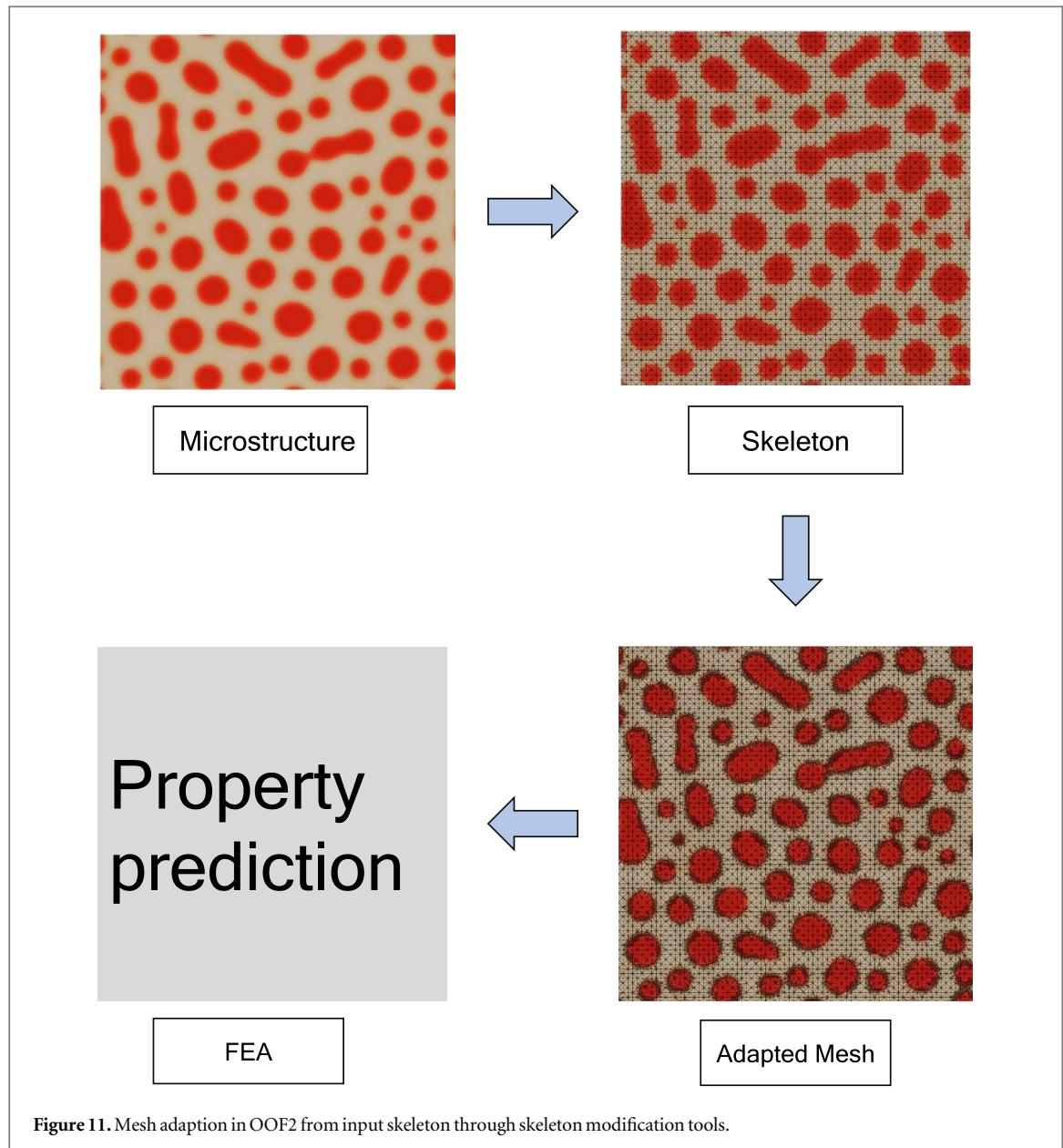


Figure 11. Mesh adaption in OOF2 from input skeleton through skeleton modification tools.

Acknowledgments

AL and ASN equally contributed to this work. RM and SB are thankful for the financial support received from C-DAC Project No. MeitY/R&D/HPC/2(1)/2014. Authors acknowledge the National Super Computing Mission (NSM) for providing computing resources of ‘PARAM Sanganak’ at IIT Kanpur, which is implemented by CDAC and supported by the Ministry of Electronics and Information Technology (MeitY) and Department of Science and Technology (DST), Government of India. We also acknowledge the HPC facility provided by CC, IIT Kanpur, and ICME National Hub, IIT Kanpur.

Data availability statement

The data cannot be made publicly available upon publication because they are not available in a format that is sufficiently accessible or reusable by other researchers. The data that support the findings of this study are available upon reasonable request from the authors.

Appendix A. Installation instructions

Follow the steps below to install and run `mu2mech` on your linux system:

- (i) Download mu2mech
 $git\ clone\ https://github.com/mu2mech/mu2mech$
- (ii) Create and activate a Python environment:
 $python3\ -m\ venv\ mu2mech-env$
 $source\ mu2mech-env/bin/activate$
- (iii) Install packages:
 $pip\ install\ -r\ requirements.txt$
 $sudo\ apt\ install\ ffmpeg$
- (iv) Run the program:
 $python3\ mu2mech.py$

Troubleshooting: Following are some common errors one may encounter during the installation or execution of mu2mech and suggested remedies.

- **ImportError: libOpenGL.so.0: cannot open shared object file: No such file or directory**
Resolve this issue by installing the libopengl0 package:
 $sudo\ apt\ install\ libopengl0\ -y$
- **qt.qpa.plugin: Could not load the Qt platform plugin ‘xcb’ in “” even though it was found.**
Reinstall the libxcb-xinerama0 package to fix this problem:
 $sudo\ apt-get\ install\ --reinstall\ libxcb-xinerama0$

Appendix B. CALPHAD integration

The system's free energy strongly influences the equilibrium compositions of the phases and subsequent microstructure evolution. CALPHAD (Calculation of Phase Diagrams) tools, such as Thermo-Calc, OpenCalphad, and Pandat, have become invaluable for obtaining thermodynamic data for studying alloy systems. These tools employ computational methods to calculate phase equilibria, phase diagrams, and thermodynamic properties based on a combination of experimental data and theoretical models.

One extracts the data points from the free energy curve and constructs a common tangent between individual valleys to determine the equilibrium compositions. The equilibrium points represent the compositions at which the system is in a minimum free energy state. These points are crucial for characterizing the stable phases and predicting phase transformations in the alloy system. One can identify the equilibrium points using mathematical tools such as OCTAVE or Thermo-Calc single-point calculation. One can approximate the free energy by a simple 4th-order polynomial equation:

$$G^* = A(c - c_1)^2(c - c_2)^2 \quad (B1)$$

where c_1 and c_2 are the equilibrium compositions and A is a constant that influences the barrier height of the free energy curve. This polynomial equation provides a mathematical representation of the free energy curve and enables further analysis and calculations. One can express the polynomial equation as:

$$G^* = A_1c^4 + A_2c^3 + A_3c^2 + A_4c + A_5 \quad (B2)$$

where $A_1 = A$, $A_2 = -2A(c_1 + c_2)$, $A_3 = A(c_1^2 + c_2^2 + 4c_1c_2)$, $A_4 = -2Ac_1c_2(c_1 + c_2)$, and $A_5 = Ac_1^2c_2^2$. These polynomial coefficients allow for the direct calculation of the free energy values at different compositions within the alloy system.

The scaled free energy (G) is utilized in our simulation, where the free energy equation above is divided by barrier height $H = A(c_1 - c_2)^4/16$. This scaling is necessary to ensure that the magnitude of the free energy values is appropriate for numerical calculations and simulations. For example, at a specific temperature, the equilibrium compositions c_1 and c_2 are determined based on the analysis of the free energy curve. The polynomial coefficients A_1, A_2, A_3, A_4 , and A_5 can then be calculated using the equation (B2). This approach can be extended to other temperatures of interest, allowing for the construction of polynomial equations that describe the free energy behavior at those temperatures. Integrating these polynomial equations with a phase field model makes it possible to simulate and analyze the alloy system's microstructure evolution and phase transformations under different thermodynamic conditions.

Table B1. Table presenting the crucial ThermoCalc-extracted dataset, indispensable for constructing the Gibbs energy plot in accordance with equation (2), along with the essential values for converting simulation results into dimensional form.

Temp. (°C)	Coefficients					A(kJ/mol)	D(m²/sec)	σ (J/m²)	L*(nm)	T*(sec)
	A ₁ /A	A ₂ /A	A ₃ /A	A ₄ /A	A ₅ /A					
200	1	-2.046	1.0964	-0.051	0.0006	42.38	4.15E-27	0.2	0.6	8.89E+07
220	1	-2.046	1.0964	-0.051	0.0006	39.55	3.26E-26	0.2	0.65	1.29E+07
240	1	-2.046	1.0964	-0.051	0.0006	36.7	2.18E-25	0.2	0.7	2.25E+06
260	1	-2.041	1.1339	-0.0943	0.0021	37.55	1.26E-24	0.2	0.76	4.55E+05
280	1	-2.041	1.1339	-0.0943	0.0021	34.64	6.48E-24	0.2	0.82	1.05E+05
300	1	-2.0314	1.1339	-0.0943	0.0021	35.4	2.95E-23	0.2	0.89	2.71E+04
320	1	-2.0314	1.1339	-0.0943	0.0021	32.25	1.21E-22	0.2	0.98	8.01E+03
340	1	-2.041	1.1789	-0.1403	0.0047	29.04	4.57E-22	0.2	1.09	2.67E+03
360	1	-2.041	1.1789	-0.1403	0.0047	32.45	1.57E-21	0.2	1.22	9.43E+02
380	1	-2.091	1.2779	-0.1933	0.0085	28.86	5.05E-21	0.2	1.38	3.78E+02
400	1	-2.041	1.2214	-0.1837	0.0081	28.46	1.50E-20	0.2	1.56	1.61E+03
420	1	-2.092	1.3212	-0.2375	0.0129	27.85	4.23E-20	0.2	1.81	77.84
440	1	-2.092	1.2633	-0.2255	0.0122	30.63	1.12E-19	0.2	2.1	39.53
460	1	-2.042	1.3599	-0.278	0.0177	35.23	2.81E-19	0.2	2.1	15.75
480	1	-2.092	1.3962	-0.316	0.0228	46.67	6.71E-19	0.2	1.81	4.9
500	1	-2.092	1.4299	-0.3513	0.0282	35	1.53E-19	0.2	1.56	1.58

Appendix C. Variable mobility and its numerical implementation

The variation of mobility depends mainly upon two factors, composition and temperature. One can get the temperature dependent diffusivity using the thermodynamic database (DICTRA, PANDAT etc.), first principle simulations, as well the previous literature [36, 51]. For Fe-Cr system [52–54]:

$$D_{Cr} = 2.0 \times 10^{-5} \exp\left(\frac{-308KJ/mol}{RT}\right) m^2/sec \quad (C1)$$

$$D_{Fe} = 2.0 \times 10^{-4} \exp\left(\frac{-60KCal/mol}{RT}\right) m^2/sec \quad (C2)$$

For Fe-Cu system [54, 55]:

$$D_{Cu} = 5.7 \times 10^{-5} \exp\left(\frac{-57KCal/mol}{RT}\right) m^2/sec \quad (C3)$$

$$D_{Fe} = 1.8 \times 10^{-5} \exp\left(\frac{-64.5KCal/mol}{RT}\right) m^2/sec \quad (C4)$$

Once the self-diffusivity of elements is known, the inter-diffusivity can be obtained through the Darken's equation and combining this with Einstein's equation for mobility we get the following relation for the binary alloy with A and B elements [56, 57]:

$$M^* = \frac{[D_A c + D_B(1 - c)]c(1 - c)}{RT}, \quad (C5)$$

$$M^* = \frac{D^* c(1 - c)}{RT}, \quad (C6)$$

In $\mu 2mech$, the variable mobility Cahn-Hilliard equation is numerically solved using semi-implicit Fourier spectral method following the work of Zhu *et al* for studying coarsening kinetics during spinodal decomposition with variable mobility [58]. The same formulation is also used to study sintering of nano-particle aggregate [59], hole growth in a single crystalline thin film [60], and thermal grooving in a polycrystalline thin film [61, 62]. The following steps are followed for numerical implementation.

- Writing down the Cahn-Hilliard equation for variable mobility as mentioned below:

$$\frac{\partial c}{\partial t} = \nabla \cdot M(c) \nabla \left(\frac{\delta F}{\delta c} \right). \quad (C7)$$

- Calculating the variational derivative of total free energy (F) to solve the Cahn-Hilliard equation and naming it as f_1 :

$$\frac{\delta F}{\delta c} = \frac{\partial f_b}{\partial c} - \kappa \nabla^2 c = g(c) - \kappa \nabla^2 c = f_1. \quad (\text{C8})$$

- Now, the Chan-Hilliard equation in Fourier space becomes:

$$\frac{\partial \tilde{c}}{\partial t} = i\mathbf{k} \cdot \{M(c)[i\mathbf{k}'(f_1)_{\mathbf{k}'}]_{\mathbf{R}}\}_{\mathbf{k}}. \quad (\text{C9})$$

Here, $[..]_{\mathbf{R}}$ represents the real space. $[..]_{\mathbf{k}}$ and $[..]_{\mathbf{k}'}$ represent the Fourier space and hence:

$$(f_1)_{\mathbf{k}'} = \tilde{g}(c) + \kappa k'^2 \tilde{c}. \quad (\text{C10})$$

- Now, calculating $M(c)[i\mathbf{k}'(f_1)_{\mathbf{k}'}]_{\mathbf{R}}$ separately and naming it at f_2 the Cahn-Hilliard equation becomes:

$$\frac{\partial \tilde{c}}{\partial t} = i\mathbf{k}(f_2)_{\mathbf{k}}. \quad (\text{C11})$$

- Discretizing the above equation we obtain:

$$\frac{\tilde{c}^{n+1} - \tilde{c}^n}{\Delta t} = i\mathbf{k}(f_2)_{\mathbf{k}}. \quad (\text{C12})$$

$$\tilde{c}^{n+1} = \tilde{c}^n + i\mathbf{k}\Delta t(f_2)_{\mathbf{k}}. \quad (\text{C13})$$

- For numerical stability, the terms $A\kappa k^4 \Delta t \tilde{c}^{n+1}$ and $A\kappa k^4 \Delta t \tilde{c}^n$ are added to the left and right sides of the equation respectively. A is taken as the average mobility value.

$$\tilde{c}^{n+1} + A\kappa k^4 \Delta t \tilde{c}^{n+1} = \tilde{c}^n + A\kappa k^4 \Delta t \tilde{c}^n + i\mathbf{k}\Delta t(f_2)_{\mathbf{k}}. \quad (\text{C14})$$

- Thus, finally we get the equation mentioned below for the variable mobility:

$$\tilde{c}^{n+1} = \frac{\tilde{c}^n(1 + A\kappa k^4 \Delta t) + i\mathbf{k}\Delta t(f_2)_{\mathbf{k}}}{1 + A\kappa k^4 \Delta t}. \quad (\text{C15})$$

Appendix D. Non-dimensionalization of the parameters

All parameters utilized in our simulations are rendered non-dimensional. One can achieve this non-dimensionalization by employing the characteristic length L^* , energy E^* , and time T^* , as indicated below (note that the * denotes the dimensional nature of these quantities):

$$E^* = H, \quad (\text{D1})$$

$$L^* = \sqrt{\frac{\kappa^*}{H}}, \quad (\text{D2})$$

$$T^* = \frac{D^*}{(L^*)^2}, \quad (\text{D3})$$

$$\kappa = \frac{\kappa^*}{H(L^*)^2}. \quad (\text{D4})$$

Here H is the barrier height of actual free energy curve (G^*) corresponding to equation (B1). κ^* is value of actual gradient energy coefficient corresponding to κ mentioned in equation (3).

The value of gradient energy coefficient (κ^*) can be calculated through the following equation [63]:

$$\sigma = 2 \int_0^1 \sqrt{\kappa^* f_b(c)} dc \approx \sqrt{\kappa^* H}. \quad (\text{D5})$$

Here σ is interfacial energy between two phases and H is the barrier height of actual free energy curve (G^*). The values of interfacial energies are 0.2 J/m^2 [64–66] and 0.45 J/m^2 [67, 68] for Fe-Cr and Fe-Cu respectively.

ORCID iDs

Somnath Bhowmick  <https://orcid.org/0000-0003-4094-5204>
Rajdip Mukherjee  <https://orcid.org/0000-0002-6714-7811>

References

- [1] Provatas N and Elder K 2010 *Phase-Field Methods in Materials Science and Engineering* (Wiley) ch. 1 1–7
- [2] Long-Qing C 2002 Phase-field models for microstructure evolution *Annual Review of Materials Research* **32** 113–40
- [3] Chatterjee S, Subhradeep Chatterjee T A and Chattopadhyay K 2008 Phase-field simulation of fusion interface events during solidification of dissimilar welds: effect of composition inhomogeneity *Metallurgical and Materials Transactions A* **39** 1638–46
- [4] Hötzer J, Jainta M, Steinmetz P, Nestler B, Dennstedt A, Genau A, Bauer M, Köstler H and Rüde U 2015 Large scale phase-field simulations of directional ternary eutectic solidification *Acta Mater.* **93** 194–204
- [5] Zhao Y, Zhang B, Hou H, Chen W and Wang M 2019 Phase-field simulation for the evolution of solid/liquid interface front in directional solidification process *J. Mater. Sci. & Technology* **35** 1044–52
- [6] Fu Y, Michopoulos J G and Song J-H 2017 Bridging the multi phase-field and molecular dynamics models for the solidification of nano-crystals *Journal of Computational Science* **20** 187–97
- [7] Saswata Bhattacharyya and Abinandanan T A 2003 A study of phase separation in ternary alloys *Bull. Mater. Sci.* **26** 193–7
- [8] Ramanarayanan H and Abinandanan T A 2003 Spinodal decomposition in fine grained materials *Bull. Mater. Sci.* **26** 189–92
- [9] Chafle R, Bhowmick S and Mukherjee R 2022 Domain boundary assisted spinodal decomposition in magnetic materials *Mater. Lett.* **324** 132630
- [10] Bandyopadhyay S, Panwar V, Guin S, Anoop C R, Gurao N P and Mukherjee R 2023 Multivariant microstructure evolution in ti-alloys: insights from a quantitative phase-field study *Modell. Simul. Mater. Sci. Eng.* **31** 1–20
- [11] Mukherjee R, Abinandanan T A and Gururajan M P 2009 Phase field study of precipitate growth: effect of misfit strain and interface curvature *Acta Mater.* **57** 3947–54
- [12] Mukherjee R, Abinandanan T A and Gururajan M P 2010 Precipitate growth with composition-dependent diffusivity: comparison between theory and phase field simulations *Scr. Mater.* **62** 85–8
- [13] Chang K, Chen L-Q, Krill C E and Moelans N 2017 Effect of strong nonuniformity in grain boundary energy on 3-D grain growth behavior: a phase-field simulation study *Comput. Mater. Sci.* **127** 67–77
- [14] Krill C E, Helfen L, Michels D, Natter H, Fitch A, Masson O and Birringer R 2001 Size-dependent grain-growth kinetics observed in nanocrystalline fe *Phys. Rev. Lett.* **86** 842–5
- [15] Verma M and Mukherjee R 2021 Grain growth stagnation in solid state thin films: a phase-field study *J. Appl. Phys.* **130** 025305
- [16] Sitompul Y P, Aoki T, Watanabe S and Takaki T 2022 An ordered active parameter tracking method for efficient multiphase field simulations *Journal of Computational Science* **64** 101834
- [17] Mukherjee R, Choudhury A and Nestler B 2013 Composition pathway in fe-cu-ni alloy during coarsening *Modell. Simul. Mater. Sci. Eng.* **21** 1–12
- [18] Molnar D, Mukherjee R, Choudhury A, Mora A, Binkele P, Selzer M, Nestler B and Schmauder S 2012 Multiscale simulations on the coarsening of cu-rich precipitates in α -fe using kinetic monte carlo, molecular dynamics and phase-field simulations *Acta Mater.* **60** 6961–71
- [19] Gururajan M P and Abinandanan T A 2007 Phase field study of precipitate rafting under a uniaxial stress *Acta Mater.* **55** 5015–26
- [20] Chafle R, Bhowmick S and Mukherjee R 2019 Effect of co-existing external fields on a binary spinodal system: a phase-field study *J. Phys. Chem. Solids* **132** 236–43
- [21] Lindsay A D et al 2022 2.0 - MOOSE: enabling massively parallel multiphysics simulation *SoftwareX* **20** 101202
- [22] Jokisaari A M, Voorhees P W, Guyer J E, Warren J A and Heinonen O G 2018 Phase field benchmark problems for dendritic growth and linear elasticity *Comput. Mater. Sci.* **149** 336–47
- [23] Tegeler M, Shchyglo O, Kamachali R D, Monas A, Steinbach I and Sutmann G 2017 Parallel multiphase field simulations with openphase *Comput. Phys. Commun.* **215** 173–87
- [24] Hötzer J, Reiter A, Hierl H, Steinmetz P, Selzer M and Nestler B 2018 The parallel multi-physics phase-field framework pace3d *Journal of Computational Science* **26** 1–12
- [25] MicroSim—Microstructure Simulator—microsim.co.in. <https://microsim.co.in/>. [Accessed 03-Jul-2023].
- [26] Kim S G, Kim W T and Suzuki T 1999 Phase-field model for binary alloys *Phys. Rev. E* **60** 7186–97
- [27] Stone J E, Ghosh D and Shi G 2010 Opencl: a parallel programming standard for heterogeneous computing systems *Comput. Sci. Eng.* **12** 66–73
- [28] Fatica M 2008 *IEEE Hot Chips 20 Symposium (HCS) Cuda toolkit and libraries* 1 1–22
- [29] OpenFOAM — The OpenFOAM Foundation — openfoam.org. <http://openfoam.org/>. [Accessed 03-Jul-2023].
- [30] ParaView - Open-source, multi-platform data analysis and visualization application—paraview.org. <https://paraview.org/>. [Accessed 03-Jul-2023].
- [31] April 15, 2024 Oof2: A Finite Element Simulation of Material Microstructures
- [32] Smith M 2009 *ABAQUS/Standard User's Manual, Version 6.9* (Dassault Systemes Simulia Corp)
- [33] Ansys — Engineering Simulation Software — ansys.com. <https://ansys.com/>. [Accessed 03-Jul-2023].
- [34] *Crystal Plasticity Finite Element Methods* (Wiley) I–XI
- [35] Saunders N and Miodownik A P 1998 *CALPHAD (Calculation of Phase Diagrams): a comprehensive guide* (Elsevier)
- [36] Andersson J-O, Helander T, Höglund L, Shi P and Sundman B 2002 Thermo-calc & dictra, computational tools for materials science *Calphad* **26** 273–312
- [37] Frigo M and Johnson S G 2002 *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 98 (Cat. No.98CH36181). IEEEFFTW: an adaptive software architecture for the FFT*
- [38] The Qt Company. Pyside2, 2023. Accessed: April 11, 2023.
- [39] Tomar S 2006 Converting video formats with ffmpegLinux J.2006
- [40] Langer S A, Fuller E R and Carter W C 2001 Oof: an image-based finite-element analysis of material microstructures *Comput. Sci. Eng.* **3** 15–23
- [41] Cahn J W and Hilliard J E 1958 Free energy of a nonuniform system. i. interfacial free energy *J. Chem. Phys.* **28** 258–67

- [42] Boyd J P 2001 *Chebyshev and Fourier spectral methods* (Courier Corporation)
- [43] Evert V 1996 A comparative study on methods for convergence acceleration of iterative vector sequences *J. Comput. Phys.* **124** 271–85
- [44] Briggs W L and Van Emden Henson 1995 The discrete fourier transform *The DFT: An owner's manual for the discrete fourier transform* Society for Industrial and Applied Mathematics **1** 15–64
- [45] Cooley J W and Tukey J W 1965 An algorithm for the machine calculation of complex fourier series *Math. Comput.* **19** 297–301
- [46] Press W H, Teukolsky S A, Vetterling W T and Flannery B P 2007 *Numerical Recipes 3rd Edition: The Art of Scientific Computing* 3rd edn (Cambridge University Press)
- [47] Claudio Canuto M Y, Hussaini A, Quarteroni and Zang T A 2007 *Spectral methods: Fundamentals in single domains* (Springer) (<https://doi.org/10.1007/978-3-540-30726-6>)
- [48] Reshak A H and Jamal M 2012 Dft calculation for elastic constants of orthorhombic structure within wien2k code: a new package (ortho-elastic) *J. Alloys Compd.* **543** 147–51
- [49] Wang V, Xu N, Liu J-C, Tang G and Geng W-T 2021 Vasplkit: a user-friendly interface facilitating high-throughput computing and analysis using vasplkit code *Comput. Phys. Commun.* **267** 108033
- [50] Ahmad O, Kumar N, Mukherjee R and Bhowmick S 2023 Accelerating microstructure modeling via machine learning: A method combining autoencoder and convlstm *Phys. Rev. Mater.* **7** 083802
- [51] Wang L-P, Yang W, Ma Z-B, Zhu J-H and Li Y-T 2020 First-principles study of chromium diffusion in the ferritic Fe-Cr alloy *Comput. Mater. Sci.* **181** 109733
- [52] Toshiyuki K and Onodera H 2006 Modeling of microstructure changes in fe- cr- co magnetic alloy using the phase-field method *Journal of Phase Equilibria and Diffusion* **27** 22–9
- [53] Metal Data Book 1993 *Metal data book* ed S Nagasaki (Japan Institute of Metals) 21 MetalData Book
- [54] Buffington F S, Ken-ichi Hirano and Cohen M 1961 Self diffusion in iron *Acta Metall.* **9** 434–9
- [55] Anand M S and Agarwala R P 1966 Diffusion of copper in iron *J. Appl. Phys.* **37** 4248–51
- [56] Mohanty R R, Guyer J E and Sohn Y H 2009 Diffusion under temperature gradient: a phase-field model study *J. Appl. Phys.* **106** 1–7
- [57] Andersson J-O and Ågren J 1992 Models for numerical treatment of multicomponent diffusion in simple phases. Journal of applied physics *J. Appl. Phys.* **72** 1350–5
- [58] Zhu J, Chen L-Q, Shen J and Tikare V 1999 Coarsening kinetics from a variable-mobility cahn-hilliard equation: Application of a semi-implicit fourier spectral method *Phys. Rev. E* **60** 3564
- [59] Mukherjee R, Chakrabarti T, Anumol E A, Abinandanan T A and Ravishankar N 2011 Thermal stability of spherical nanoporous aggregates and formation of hollow structures by sintering a phase-field study *ACS Nano* **5** 2700–6
- [60] Verma M and Mukherjee R 2023 Effect of surface energy anisotropy on hole growth in a single-crystalline thin film: a phase-field study *Scr. Mater.* **234** 1–7
- [61] Verma M, Sugathan S, Bhattacharya S and Mukherjee R 2022 A computational analysis of universal behavior of thermal groove in a moving grain boundary *Scr. Mater.* **209** 114383
- [62] Verma M, Sugathan S, Bhattacharyya S and Mukherjee R 2023 Effect of concurrent thermal grooving and grain growth on morphological and topological evolution of a polycrystalline thin film: insights from a 3d phase-field study *Acta Mater.* **261** 1–17
- [63] Long-Qing C 2002 Phase-field models for microstructure evolution *Annual Review of Materials Research* **32** 113–40
- [64] Lu S, Hu Q-M, Yang R, Johansson B and Vitos L 2010 First-principles determination of the $\alpha - \alpha'$ interfacial energy in Fe-Cr alloys *Phys. Rev. B* **82** 195103
- [65] Johnson D F, Jiang D E and Emily E A 2007 Structure, magnetism, and adhesion at cr/fe interfaces from density functional theory *Surf. Sci.* **601** 699–705
- [66] Lu S, Hu Q-M, Johansson B and Vitos L 2011 Composition and orientation dependence of the interfacial energy in fe-cr stainless steel alloys *Physica Status Solidi (b)* **248** 2087–90
- [67] Cui S, Mamivand M and Morgan D 2020 Simulation of cu precipitation in fe-cu dilute alloys with cluster mobility *Mater. Des.* **191** 108574
- [68] Jia Z, Zhao X, Zhang G, Kang Y, Xu H and Zhao Z 2022 A comparable study of fe/cu interfaces by first-principles method: the surface energy, work of adhesion and electronic structures *Physica B* **646** 414348