

Pandat PPF CA Solidification

User Manual

Table of Contents

Tutorial	3
<i>Pandat Batch (PBFX) File</i>	<i>3</i>
<i>Phasefield Database (PFDB) File</i>	<i>6</i>
Single-grain and multi-grain simulations	6
Surface and bulk nucleation	7
Initial grain structure	8
Thermal field	9
Additional thermodynamic properties	10
Time step size	11
Check initial conditions	11
<i>Examples</i>	<i>12</i>
Single grain growth	12
Casting	16
Directional Solidification	18
Welding	20
Theory	23
<i>Nucleation</i>	<i>23</i>
<i>Crystal Orientation</i>	<i>24</i>
<i>Grain Growth</i>	<i>25</i>
<i>Columnar and Equiaxed Grain Structure</i>	<i>25</i>
<i>Sensitivity and Numerical Error</i>	<i>28</i>
References	29

Tutorial

To run a CA simulation, you need to provide the thermodynamic database file (TDB or PDB file), the Pandat Batch file (PBFX file), and the phase-field database file (PFDB file). The TDB/PDB files are provided by Pandat that define the thermodynamic properties for the alloy system of interest. The user needs to provide the PBFX and PFDB files for the specific problem. The PBFX file defines the setup such as geometry, alloy components, and mesh size. The PFDB file defines additional user-controlled material properties that are not available in TDB/PDB files. In this tutorial, we describe how to modify the PBFX and PFDB files and give several examples. We recommend the user follow and reproduce these examples before setting up a new simulation.

Pandat Batch (PBFX) File

The PBFX file contains the problem setup information. The user can either directly edit the PBFX file or modify the PBFX file through Pandat GUI. Here, we use an example PBFX file (named “SingleGrain_AlSi7.pbf” under the example folder “SingleGrain_AlSi7”) to introduce the meaning of each parameter. It is suggested that the user carefully examine the parameters discussed here to correctly define the problem.

Calculation name: define **calculation name** as desired, the **type** must be “phasefield”, and the **app** must be “PPF_SolidificationCA” for the CA addon.

Databases: define the TDB/PDB and PFDB files associated with the CA simulation.

Units: define the unit for pressure (**P**), temperature (**T**), **length**, components (**n**), and **time**. By default, the unit for pressure is Pascal (Pa), temperature is Kelvin (K), length is meter (m), component is weight fraction (w), and time is second.

Alloys: the **alloy name** must be the same as the alloy name in the PFDB file. Also see the section Phasefield Database (PFDB) File.

Components: define the alloy components. **The selected components must exist in the TDB/PDB files.**

Phases: define the phases. **The liquid phase must be included.** The second phase must be either **FCC** (face-centered-cubic) or **BCC** (body-centered-cubic). **The current version does not support other types of solid phases (such as HCP) and does not support a third phase either.** Also see the section Phasefield Database (PFDB) File.

Weight fraction: define the weight fraction of each component. An example is given below:

```
<statespace>
  <n component="Al" value="0.93" />
  <n component="Si" value="0.07" />
</statespace>
```

If the alloy component's unit is weight fraction (w), it means the mass fraction (not mole fraction), and the fraction must be summed up to 1 (not 100).

Note on pure metal:

The current version does not support single-component (pure) metal. To simulate the grain growth of pure metal, you can select two components and make one of the component have a very small weight fraction. In the example folder "SingleGrain_PureAl", we define the weight fraction of Al as 0.9999 and a weight fraction of Si as 0.0001. *However, we do not suggest defining a weight fraction smaller than 0.0001 as it might trigger numerical singularities and instabilities.*

Thermal history: define the temperature history at the reference location specified in the PFDB. Based on the temperature at the reference location, the three-dimensional temperature field is computed as described in PFDB file. Each line specifies a time

moment and a temperature file (see more in the section Phasefield Database (PFDB) File). **The first time moment must be 0**, and the simulation stops at the last time moment. A linear interpolation is used between the two specified time moments. An example is given below:

```
<node time="0" T="892.0" />
<node time="1" T="885.0" />
<node time="11" T="884.0" />
```

Mesh: the mesh can only be a uniform Cartesian mesh. The mesh size Δx is given by “length_scale value.” The number of meshes in X, Y, Z directions are given by “number_of_grids_x,” “number_of_grids_y,” and “number_of_grids_z,” respectively. **The user does not have access to control the origin of the mesh, which is always (0.0, 0.0, 0.0).**

Computation threads: the current version only supports serial computation. The “number_of_threads” must be 1.

Output frequency: define the total number of output files in the “num_of_profile_outputs value.” The output files will be generated evenly in time as the simulation progress from time zero to the last time moment defined in thermal history. Each output file contains the temperature and grain id at each mesh cell $T(x, y, z)$, $ID(x, y, z)$.

Output tables: in the “tables” definition, a table is generated with the columnar names “Temperature” and “Grain_Growth_Velocity.” This table’s name is “Grain Growth Kinetics.” It describes the grain growth velocity V_g as a function of temperature T . V_g is zero if T is above the liquidus temperature, and V_g increases monotonically as T decreases below the liquidus temperature. **We suggest the user keep this output table as default.**

Output graphs: in the “graphs” definition, two graphs are generated. One is under the name “Grain Growth Kinetics,” which is a 2D plot of the grain growth kinetics $V_g(T)$. The X axis is the temperature, and the Y axis is the grain growth velocity. The second graph

is under the name “ppf_vtk_profiles,” which is the temperature $T(x,y,z)$ and grain ID $ID(x,y,z)$ at each output time moment (number of output files is defined by “num_of_profile_outputs value”). **We suggest the user keep these output graphs as default.**

Phasefield Database (PFDB) File

The PFDB file contains the material properties that are not provided in TDB/PDB file. The user needs to manually edit the PFDB file to define these properties. We use an example PFDB file (named “SingleGrain_AlSi7.pfdb” under the example folder “SingleGrain_AlSi7”) to introduce the meaning of each parameter. The parameters discussed here should be carefully examined when setting up a new simulation.

Alloy name: define the alloy name. The user may give any name, but it must be consistent with the alloy name used in the PBFX file.

Phases: define the phases. **The liquid phase must be defined as “Matrix type” and the solid phases defined as “Precipitate type.” Only FCC or BCC solid phases are supported.** The user can define both FCC and BCC phases in the PFDB file, but only one of them can be selected in the PBFX file. **The current version only supports one solid phase.**

Custom parameters: the CA related parameters are all defined under the custom parameter table. At each line, the user can modify the value of the variables, **but the name of the variables must not be modified.** We further detail the parameter definitions below.

Single-grain and multi-grain simulations

The variable **s_single_grain** controls the simulation to be single-grain ($s_single_grain = 1$) or multi-grain ($s_single_grain = 0$). Single-grain simulation is a special type of simulation where only one nucleation seed is defined at the center of the calculation

domain and the nucleation cannot occur elsewhere. The single nucleation has a fixed Euler angle (20°, 20°, 20°) (see definition of Euler angle in the Theory section). Only one grain can grow from the nucleation seed and eventually the grain will occupy the entire domain. The single-grain option is to verify the grain growth for a new alloy. **We suggest the user run a single-grain simulation before multi-grain simulations to examine if the grain grows as expected.**

For multi-grain simulations, the nucleation seeds can be distributed at surfaces and interior of the calculation domain as described below.

Surface and bulk nucleation

Here, we describe the nucleation-related parameters, and more details can be referred to the Theory section. The bulk nucleation occurs in the interior of the domain. The nucleation number density is given by the variable **nmax_b**, which has a unit of m^{-3} . The total number of nucleation seeds is nmax_b multiplied by the volume of the domain. Setting nmax_b to zero means no bulk nucleation will occur. The seeds will be randomly distributed inside the domain. For each seed, the nucleation undercooling is randomly generated based on a Gaussian distribution. The mean undercooling is given by the variable **TN_b**, and the standard deviation is given by the variable **Tsig_b**. Once nucleated, the new grain will have a random Euler angle.

The surface nucleation occurs on the six surfaces of the cuboid domain. As shown in Figure 1, the surfaces are assigned as X_1 , X_2 , Y_1 , Y_2 , Z_1 , and Z_2 planes.

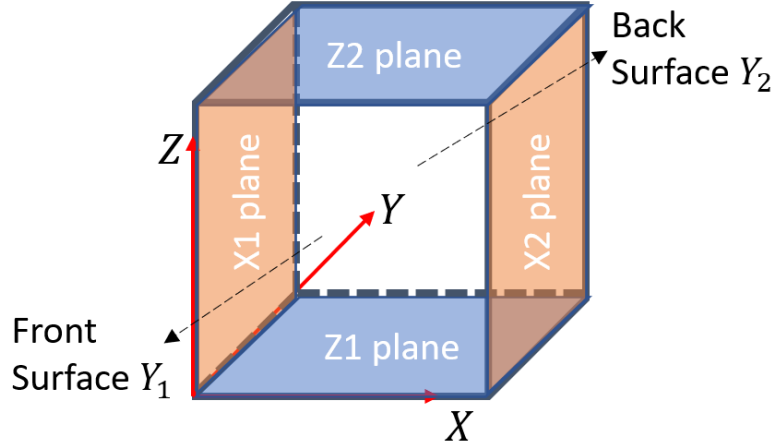


Figure 1. Surface definitions of calculation domain.

For X_1 plane as an example, the nucleation number density is given by the variable **nmax_s_x1**, which has a unit of m^{-2} . The total number of nucleation seeds on the X_1 plane is given by nmax_s_x1 multiplying the area of the X_1 plane. The seeds will be randomly distributed at the X_1 plane. For each seed, the nucleation undercooling is randomly generated based on a Gaussian distribution. The mean undercooling is given by the variable **TN_s_x1**, and the standard deviation is given by the variable **Tsig_s_x1**. Once nucleated, the new grain will have a random Euler angle. The other planes follow the same definition. For example, the nucleation density on Y_2 plane is given by the variable **nmax_s_Y2**, the mean value of nucleation undercooling is given by the variable **TN_s_Y2**, and the standard deviation is given by the variable **Tsig_s_Y2**.

Initial grain structure

The user needs to specify the initial grain structure, which is controlled by the variable **s_init_grain**. **s_init_grain** = 1 means a random equiaxed structure. **s_init_grain** = 0 means a liquid condition, and no initial grain exists in the domain. When **s_init_grain** is set to 1, the user needs to give an averaged grain size for the equiaxed structure, which is defined by the variable **init_grain_size** (unit: m). How to set the initial grain structure depends on the application. We give some general instructions:

In some scenarios such as casting, the entire domain is liquid initially. As the simulation advances in time, surface or bulk nucleation occurs when the temperature drops below the nucleation temperature. This is when `s_init_grain` should be set to 0. **Accordingly, the temperature at time 0 should be above the liquidus temperature in the PFBX file. Otherwise, no nucleation and growth will occur.**

In some scenarios such as welding, the domain is initially solid and should have an initial grain structure. This is when `s_init_grain` should be set to 1 and an averaged grain size should be given by `init_grain_size`. As the simulation progresses, the melting and solidification will alter the initial grain structure. The grain ID will be erased in the molten region, and epitaxial nucleation will occur at the solidification front. **The mechanisms of epitaxial, surface, and bulk nucleation are discussed in detail in the Theory section.**

Thermal field

The three-dimensional thermal field is analytically defined in the current CA simulation. The thermal field type is controlled by the variable `s_thermal_field_type`. Two types of thermal fields can be defined, linear (`s_thermal_field_type = 1`) or Rosenthal (`s_thermal_field_type = 2`).

In the linear thermal field, the temperature field is given by Eq. (1)

$$T(x, y, z) = G_x(x - x_r) + G_y(y - y_r) + G_z(z - z_r) + T_r(t) \quad (1)$$

Here, x_r , y_r , and z_r are the reference point coordinates, defined in the variable `x_ref`, `y_ref`, and `z_ref`, respectively. G_x , G_y , G_z are the temperature gradient along the X, Y, and Z directions, given by the variables `Gx`, `Gy`, `Gz`, respectively. The thermal history at the reference point is $T_r(t)$ is specified in the PFBX file.

In the Rosenthal thermal field, the temperature field is given by Eq. (2):

$$T(x, y, z, t) = T_0 + \frac{Q}{2\pi k r} \exp\left(\frac{V(r - x)}{2\alpha}\right) \quad (2)$$

The Rosenthal thermal field describes the typical thermal field in the welding process [1]. Here, T_0 is a reference temperature, which is fixed as 300 K. Q is the absorbed power

from the weld heat source, given by the variable **Rosenthal_Q** (unit: W). k is the liquid metal thermal conductivity. α is the liquid thermal diffusivity $\alpha = \rho/kc_p$ (unit: m²/s) where ρ is the liquid density, and c_p is the liquid specific heat. Here, ρ and k will be user-defined in the PFDB file, while c_p is provided by the TDB/PDB file. V is the scanning velocity of the heat source. r is the radial distance from the weld center, $r = \sqrt{(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2}$. **In the current version, the scanning direction can only be along the $\pm X$ direction.** The welding center x_c , y_c , and z_c are evaluated as follows:

- $x_c = x_r + Vt$, where x_r is given by the variable x_ref , and V is a constant given by the variable **Rosenthal_V** (unit: m/s).
- $y_c = y_r$, where y_r is given by the variable y_ref .
- $z_c = z_r$, where z_r is given by the variable z_ref .

If the Rosenthal thermal field is used, the reference thermal history defined in the PBFX file will be overwritten. However, the starting time (time = 0) and ending time of the simulation should still be given in the thermal history section of the PBFX file. More discussions of the Rosenthal thermal field will be given in the section Welding Example.

Additional thermodynamic properties

Several additional thermodynamic properties that are unavailable in the TDB/PDB file should be defined in the PFDB file. The density of the liquid metal is given by the variable **k_rho** (unit: kg/m³). The thermal conductivity of the liquid metal is given by the variable **k_kappa** (unit: W/m K). The value of k_rho and k_kappa for common alloys can be found in [2]. The surface energy between the liquid and solid metal is given by the variable **k_sigma** (unit: J/m²). The surface energy for a complex alloy system can be difficult to obtain. We suggest the user use the surface energy of pure metal corresponding to the dominating component of an alloy to define k_sigma . The surface energy for common pure metals can be found in [3, 4]. The thermodynamic properties can be a function of temperature. The current version only supports constant definitions. We suggest the user select the properties at the liquidus temperature to define k_rho , k_kappa , and k_sigma .

Time step size

The CA simulation advances in discrete time steps. Given the simulation result at the n^{th} time step, the $n+1^{\text{th}}$ time step size Δt^{n+1} is given by Eq. (3)

$$\Delta t^{n+1} = \alpha \frac{\Delta x}{V_{g,max}^n} \quad (3)$$

where Δx is the mesh size, $V_{g,max}^n$ is the maximum grain growth velocity in the entire domain calculated from the previous (n^{th}) time step. α is a user-defined constant smaller than 1.0, which is controlled by the variable **dt_alpha**. **We suggest a value of 0.2 is used for dt_alpha to resolve the grain growth.** In the case where the grain growth velocity V_g is small, the time step size Δt can become excessively large. A maximum Δt is set for the entire CA simulation, which is given by the variable **dt_max** (unit: second). We suggest setting dt_max to be around 1/100 of the entire physical duration of the simulation.

Numerical error may result in an unphysically large local grain velocity. This can significantly slow down the simulation due to an excessively small Δt . We can define a maximum V_g for the entire simulation to limit the unphysically large velocity. This maximum cap is given by the variable **CA_Vcap** (unit: m/s). By default, CA_Vcap is set to 1 m/s. If Rosenthal thermal field is used, CA_Vcap is set to be the same as Rosenthal_V. More discussion on the numerical error in CA simulations is given in the Theory section.

Check initial conditions

The user has the option to stop the simulation and only check the initial condition (such as the initial temperature field and grain ID). This is controlled by the variable **s_check_init**. Setting s_check_init = 1 stops the simulation at time = 0. s_check_init = 0 means a full simulation. **We suggest the user set s_check_init = 1 and check the initial condition when running a new simulation.**

Examples

There are four types of examples provided under the installation directory. We suggest the user carefully study these examples before running a customized CA simulation. The folders `SingleGrain_AlSi7`, `SingleGrain_PureAl`, and `SingleGrain_SS304` contain the single-grain simulation examples for Al-Si alloy, pure aluminum, and stainless steel 304 (SS304), respectively. The folders `Casting_AlSi7_Columnar` and `Casting_AlSi7_CET` contain casting simulations with an assumed uniform temperature field. The name “Columnar” and “CET” stand for columnar-dominated grain structure and columnar-to-equiaxed transition. As will be demonstrated, the grain structure (columnar or CET) can be highly dependent on the nucleation conditions. The folders `DS_AlSi7_Columnar` and `DS_AlSi7_CET` contain the directional solidification (DS) simulation with an assumed cooling rate and thermal gradients. Finally, the folder `Welding_SS304` contains a welding simulation example for SS304. In each folder, a TDB file, a PBFX file, and a PFDB file are included. We will describe in detail the problem setup for each example below.

To run a simulation, the user can directly load the PBFX file from the Batch Calc → Batch Run option in the Pandat GUI.

Single grain growth

The motivation of this example is to verify the single grain growth in a 5 mm x 5 mm x 5 mm domain. The mesh size is set as 0.05 mm, giving 100 x 100 x 100 mesh cells. We define the temperature such that the grain growth velocity is approximately 0.25 mm/s. 10 seconds of physical time is simulated for the single grain to approximately occupy the entire domain. We will use Al-7wt%Si (`SingleGrain_AlSi7`) as an example to illustrate the problem definition. The other examples (`SingleGrain_PureAl` and `SingleGrain_SS304`) are defined similarly. The single-grain simulation is a common verification test for CA algorithms. Similar examples have been documented in [5].

In the PBFX file, we define the thermal history as follows:

```
<node time="0" T="892.0" />
```

```
<node time="1" T="885.0" />
```

```
<node time="11" T="884.0" />
```

At time = 0, the temperature (892 K) is above the liquidus temperature. The liquidus temperature can be obtained by running the simulation with `s_check_init = 1`, as will be described later. At time = 1 s, the temperature is set to be lower than the liquidus temperature to trigger nucleation. The undercooling (difference between liquidus and set temperature) is around 5 K to ensure significant grain growth. **The user should carefully check the thermal history. The grain nucleation will not occur if the temperature is always below the liquidus temperature.** From time = 1 s to time = 11 s, the reference temperature linearly decreases with a cooling rate of 0.1 K/s. The simulation will cover a total physical time of 11 seconds.

In the PFDB file, we define a temperature gradient along the Z direction ($G_z = 250$ K/m). This is to verify if the grain grows faster in the -Z direction, which has a smaller temperature. Note that the reference point is selected to be on the top surface of the domain ($z_{ref} = 0.0025$ m). Therefore, the temperature inside the domain will be smaller than the reference point thermal history. The density (k_{rho}) and thermal conductivity (k_{kappa}) of Al-7wt%Si are found in [6]. We use the surface energy for pure aluminum in to define the surface energy (k_{sigma}). For the single-grain simulation, **the variable `s_single_grain` must be set to 1**. All the bulk and surface nucleation densities should be set to zero.

We first set `s_check_init` to be 1 and run the simulation to check the initial conditions, as shown in Figure 2. On the upper left “Workspace window,” we can see that two graphs and two tables are created, corresponding to the output table and graph definitions in the PBFX file. The graph “`ppf_vtk_profiles`” contains the temperature and grain ID. The graph “Grain Growth Kinetics” graph is the grain growth velocity as a function of temperature. The table “`ppf_result_t0000`” contains the ASCII data of the graph “`ppf_vtk_profiles`,” with the number 0000 indicating the first time moment. The table “Grain Growth Kinetics” contains the ASCII data of the graph “Grain Growth Kinetics.” The temperature field is shown on the right window in Figure 2. The user can choose which variable to visualize

by modifying the “Target Property Name” in the lower upper “Property” window. The name “custom_field(@CA_Temperature)” means the temperature field. The name “custom_field(@CA_nucindex)” means the grain ID. As can be seen in Figure 2, the temperature linearly increases in the +Z direction.

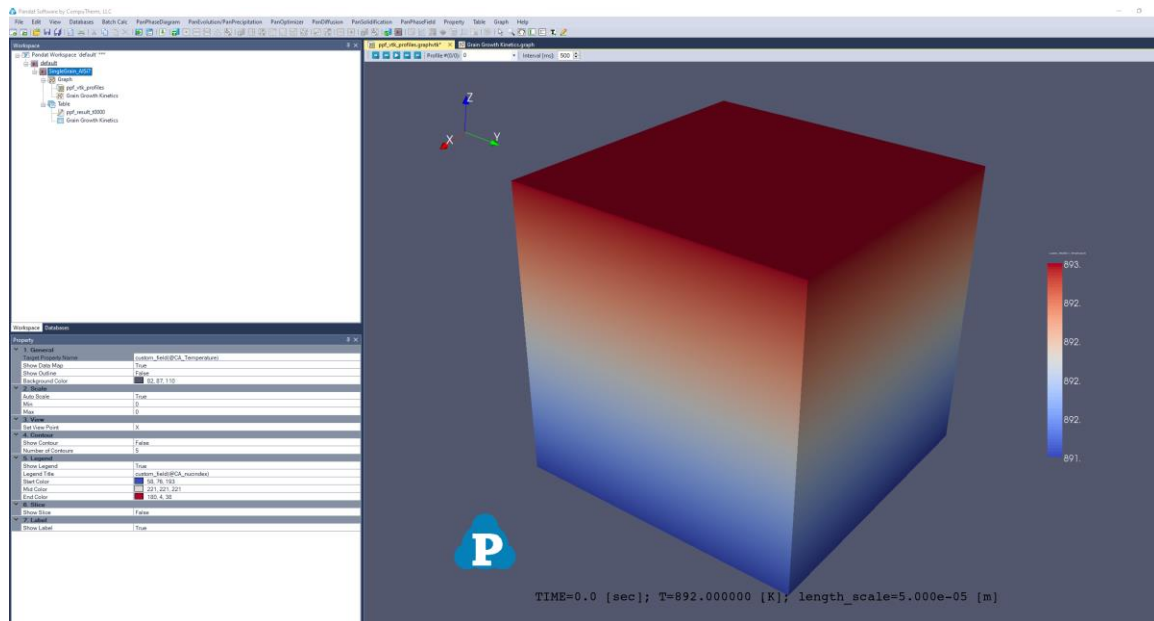
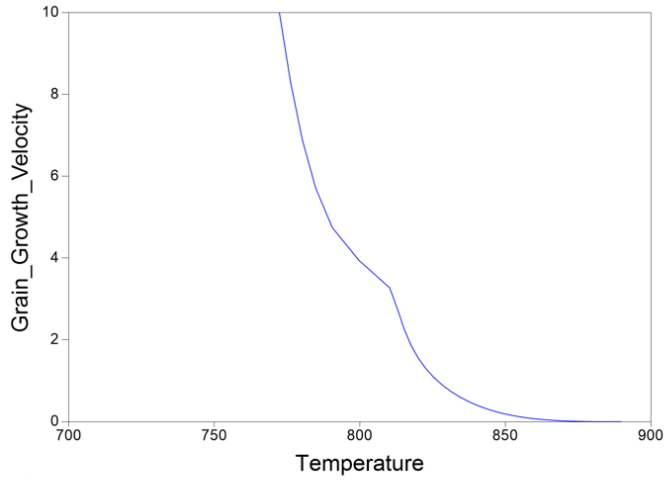


Figure 2. Initial condition check for single-grain simulation.

The grain growth kinetics graph and table are shown in Figure 3. **The first data point in the table always indicates the liquidus temperature (889.772 K) with zero grain growth velocity.** As temperature decreases below the liquidus temperature, the grain growth velocity increases. We identify from the grain growth kinetics that a temperature around 885 K gives a grain growth velocity of around 0.25 mm/s. Therefore, we set the temperature to be 885 K at time = 1 s, so that the grain growth of approximately 10 seconds makes the grain occupy the entire domain. As shown here, the user can run the simulation with s_check_init = 1 to obtain the liquidus temperature and design the thermal history accordingly.



(a)

	Temperature	Grain_Growth_Velocity
1	889.7720	0.000000
2	889.6020	1.000000E-007
3	889.5880	1.204500E-007
4	889.5720	1.450830E-007
5	889.5560	1.747530E-007
6	889.5380	2.104900E-007
7	889.5190	2.535360E-007
8	889.4980	3.053860E-007
9	889.4750	3.678380E-007
10	889.4510	4.430620E-007

(b)

Figure 3. Grain growth kinetics (a) graph and (b) table.

After checking the initial condition, we set `s_check_init` to 0 and rerun the simulation to get the full grain growth result. The grain growth is shown in Figure 4. By default, the graph shows the three-dimensional variable (temperature of grain ID) on the outer surfaces of the cuboid. To visualize the grain morphology, we need to plot the isosurface of the grain ID. In the lower right “Property” window, select “Target Property Name” to be “`custom_field(@CA_nucindex)`”, select “Show Data Map” to be False (which cancels the plot of the outer surfaces), select “Show Outline” to be True (which plots the edges of the cuboid), and select “Show Contour” to be True (which shows the isosurface). As can be seen in Figure 4, the grain has an octahedron shape, with the tip towards the -Z direction growing faster than the tip towards the +Z direction. This result is consistent with the result in [5]. **We suggest the user modify the temperature gradient and thermal history and rerun different single-grain simulations to gain a better understanding of the grain growth.**

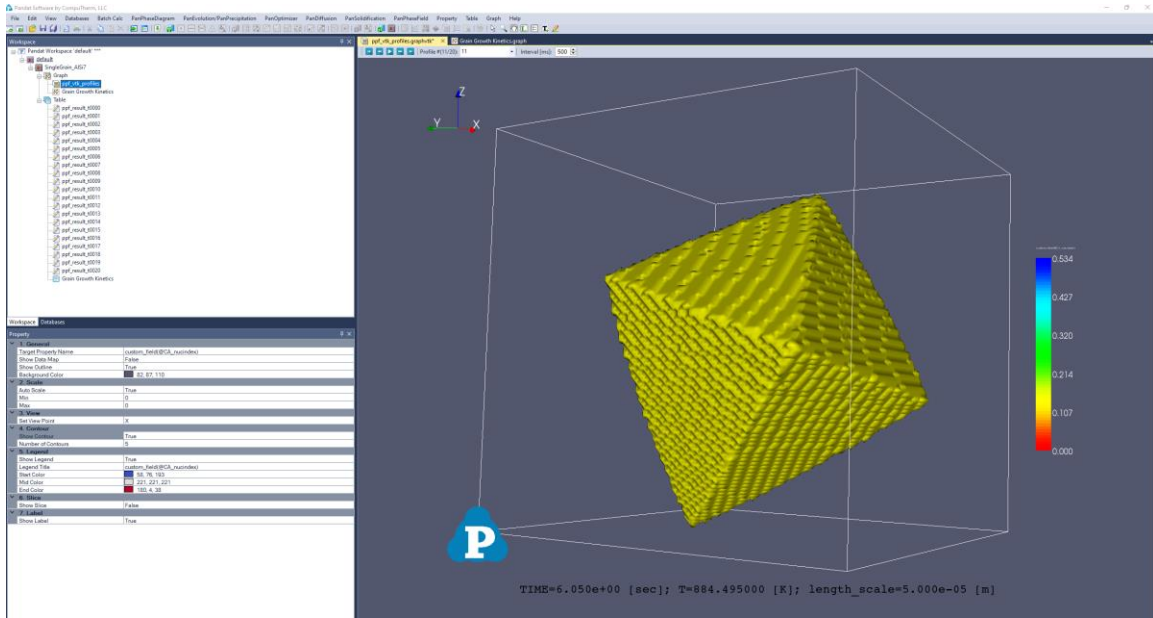


Figure 4. Single-grain simulation for Al-7wt%Si.

Casting

In this example, we simulate the grain growth of Al-7wt%Si in a simplified casting scenario inspired by [7]. The calculation domain is a 4.5 mm x 4.5 mm x 0.15 mm cuboid. The mesh size is 0.015 mm, giving a total of 300 x 300 x 10 mesh cells. The Z dimension is small to make the problem two-dimensional and save computation time. The entire domain is under a uniform temperature field ($G_x = G_y = G_z = 0$). The thermal history is defined as:

```
<node time="0" T="890.0" />
```

```
<node time="6" T="875" />
```

Note the initial temperature is above the liquidus temperature. A cooling rate of 2.5 K/s is applied. A total physical time of 6 seconds is simulated.

In the PFDB file, the bulk nucleation density (n_{max_b}) is defined as $5.5 \times 10^{10} \text{ m}^{-3}$. The mean bulk nucleation undercooling (TN_b) is 10 K in the example “Casting_AISi7_Columnar,” and TN_b is 6 K in the example “Casting_AISi7_CET.” The surface nucleation is defined on the plane X_1, X_2, Y_1, Y_2 (Figure 1). There is no surface nucleation for the plane Z_1 and Z_2 . The surface nucleation density for all of the $X_1, X_2, Y_1,$

Y_2 planes is $2.5 \times 10^8 \text{ m}^{-2}$. The mean surface nucleation undercooling for the X_1 , X_2 , Y_1 , Y_2 planes is 0.5 K. The standard deviation of both bulk and surface nucleation is defined to be 0.1 K. The nucleation parameters are selected based on the values in [7]. As temperature decreases below the liquidus temperature, grains will first be nucleated at the X_1 , X_2 , Y_1 , Y_2 planes. After nucleation, the grains grow towards the center of the domain to form columnar grains. As the temperature further decreases, bulk nucleation can occur inside the domain to form equiaxed grains. Such grain structure is typical in a casting ingot.

The grain structure at the end of the simulation for the example “Casting_AISi7_Columnar” and “Casting_AISi7_CET” is shown in Figure 5. In both cases, the solidification front initiates from the X_1 , X_2 , Y_1 , Y_2 planes and moves towards the center. For the columnar-dominated case (Figure 5(a)), several large columnar grains outgrow the other grains through the competitive growth mechanism. For the CET case (Figure 5(b)), the small bulk nucleation undercooling (6 K) renders equiaxed grains forming ahead of the solidification front. Eventually, the large amount of equiaxed grains completely blocks the growth of columnar grains, leading to a columnar-to-equiaxed transition. The simulation results in Figure 5 reasonably agree with the results in [7]. More discussions of competitive growth and CET are given in the Theory section.

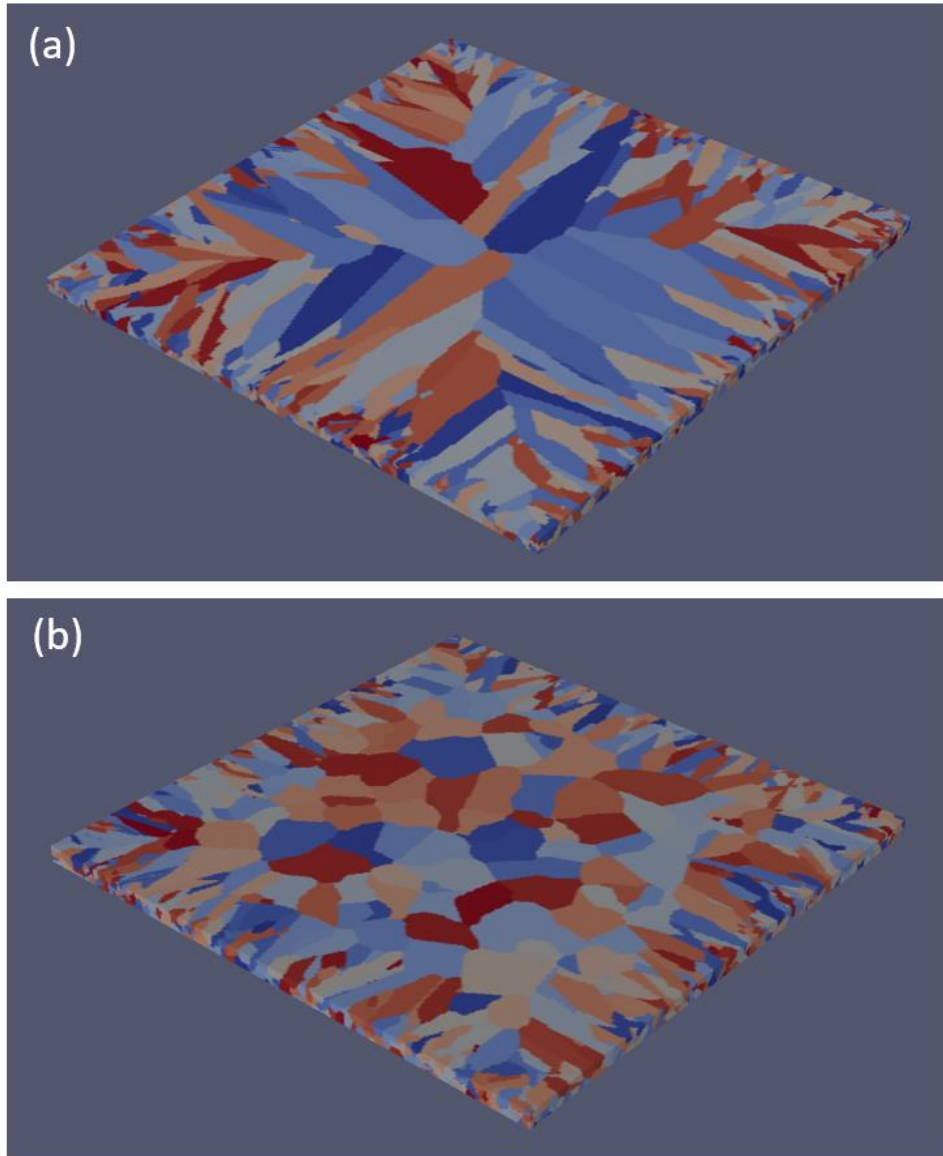


Figure 5. Casting simulation showing (a) columnar-dominated grain structure and (b) columnar-to-equiaxed transition.

Directional Solidification

In this example, we simulate a directional solidification (Al-7wt%Si) scenario which is inspired by [6, 7]. The domain is 0.05 m x 0.05 m x 0.2 m. The mesh size is 1 mm, giving a total of 50 x 50 x 200 mesh cells. A temperature gradient of 250 K/m is applied in the +Z direction (Gz). The following thermal history is defined at the center of the Z_1 plane (reference point):

```
<node time="0" T="892.0" />
```

```

<node time="1" T="884.0" />
<node time="501" T="859" />
<node time="1001" T="759.0" />

```

Initially, the entire domain is in a liquid state. At time = 1 s, the temperature is decreased to 884 K, which triggers surface nucleation at the Z_1 plane. The nucleation is defined in the interior of domain (bulk nucleation) and at the Z_1 plane (refer to the PFDB file in the example folders for the detailed nucleation parameters). From time = 1 s to time = 501 s, a cooling rate of 0.05 K/s is applied. The solidification front initiates from the Z_1 plane and move towards the +Z direction. The velocity of the solidification front can be estimated by $V_{SF} = \dot{T}/G_z$, where the $\dot{T}$ is the cooling rate (0.05 K/s) and G_z is the thermal gradient in the Z direction (250 K/m), giving a V_{SF} of 0.2 mm/s. As studied by the simulation and experiment in [6], the thermal gradient becomes smaller as the solidification front approaches the top surface (Z_2 plane), which leads to a larger V_{SF} . However, the current version does not support a time-varying thermal gradient. To simulate the increase of solidification front velocity V_{SF} , we artificially increase the cooling rate in the time range 501 – 1001 s from 0.05 K/s to 0.2 K/s. Accordingly, the V_{SF} increases from 0.2 mm/s to 0.8 mm/s.

The grain structure at the end of the simulation for the example “DS_AISi7_Columnar” and “DS_AISi7_CET” is shown in Figure 6. In the columnar-dominated case, the bulk nucleation undercooling is set as a large value ($TN_b = 20$ K). Due to the large nucleation undercooling, no bulk nucleation occurs, and large columnar grains dominate the structure. In the CET case, the TN_b is set as 5 K to allow bulk nucleation. According to the CET theory by Hunt [8] (also see the Theory section), an increase in solidification front velocity (with a constant thermal gradient) promotes the bulk nucleation and formation of equiaxed grain. Therefore, the columnar-to-equiaxed transition occurs after the solidification front velocity is increased at time = 501 s. The results in Figure 6 reasonably agree with [7].

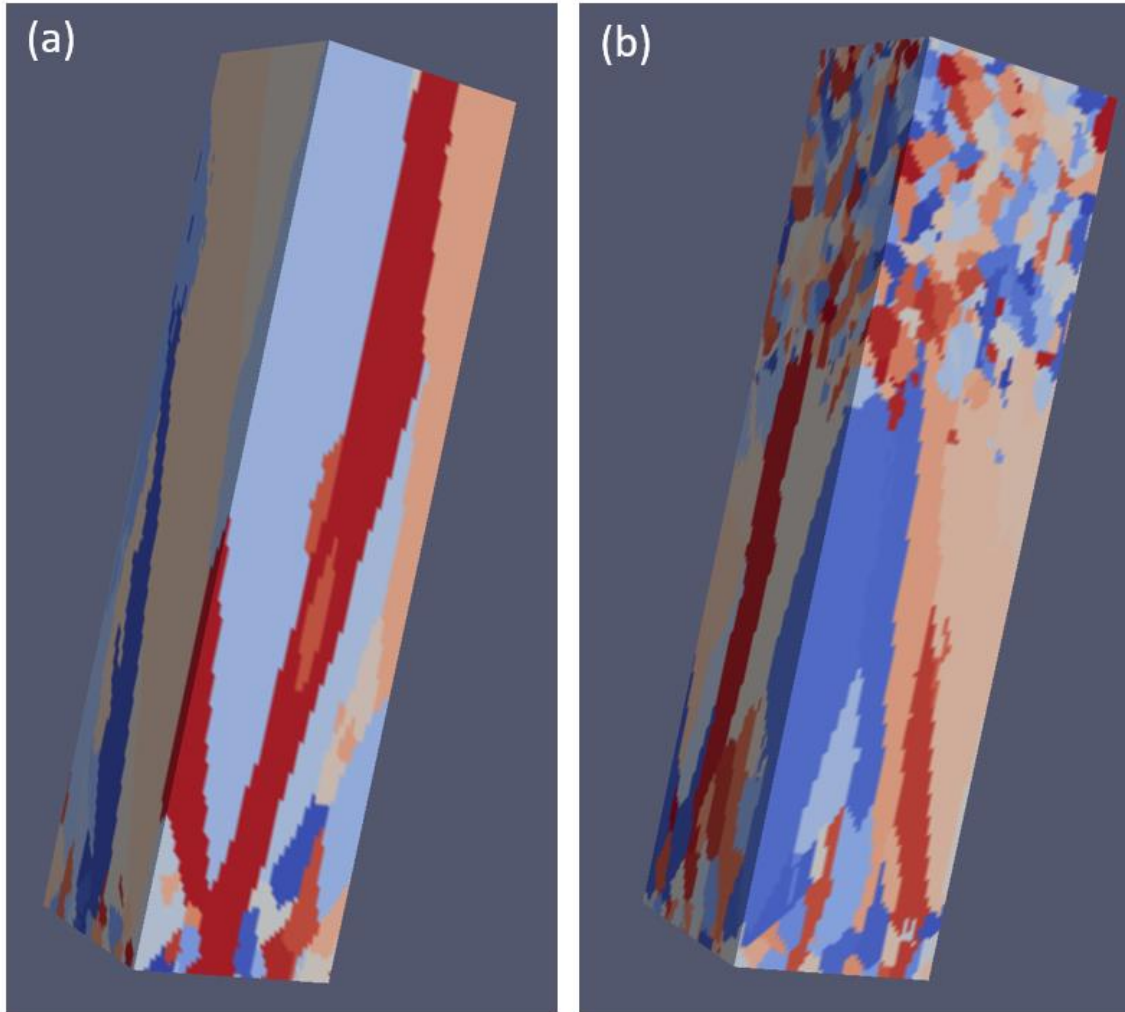


Figure 6. Direction solidification simulation showing (a) columnar-dominated grain structure and (b) columnar-to-equiaxed transition.

Welding

In this example (Welding_SS304 folder), we simulate the epitaxial nucleation and grain growth in a welding situation inspired by [9]. The material is a Fe-Cr-Ni alloy resembling the components of stainless steel 304. The domain is 15 mm x 5 mm x 2 mm. The mesh size is 0.05 mm, giving a total of 300 x 100 x 40 mesh cells. In the PFDB file, the Rosenthal thermal field is used (`s_thermal_field_type = 2`). The absorbed power from the heat source is 1000 W (`Rosenthal_Q`). The heat source center is initially located at (0.0, 0.0025, 0.002) m, defined by `x_ref`, `y_ref`, and `z_ref`. The heat source has a scan speed of 0.01 m/s along the +X direction (`Rosenthal_V`). The current version only supports scanning

direction towards the $\pm X$ direction (Rosenthal_V being negative means scanning along the -X direction). We suggest setting the heat source center at the top surface of the domain ($z_{ref} = 0.002$ m in this example) so that a semi-ellipsoidal temperature contour is achieved to capture the molten pool shape. An example thermal field during the welding simulation is shown in Figure 7. To better visualize the temperature contour, we can modify the Auto Scale in the Property window to be False, and then choose the Min value to be 300 (K) and Max value to be 2000 K. The Rosenthal thermal field gives an unphysically large peak temperature, but the molten pool shape is deemed physical, as shown in Figure 7.

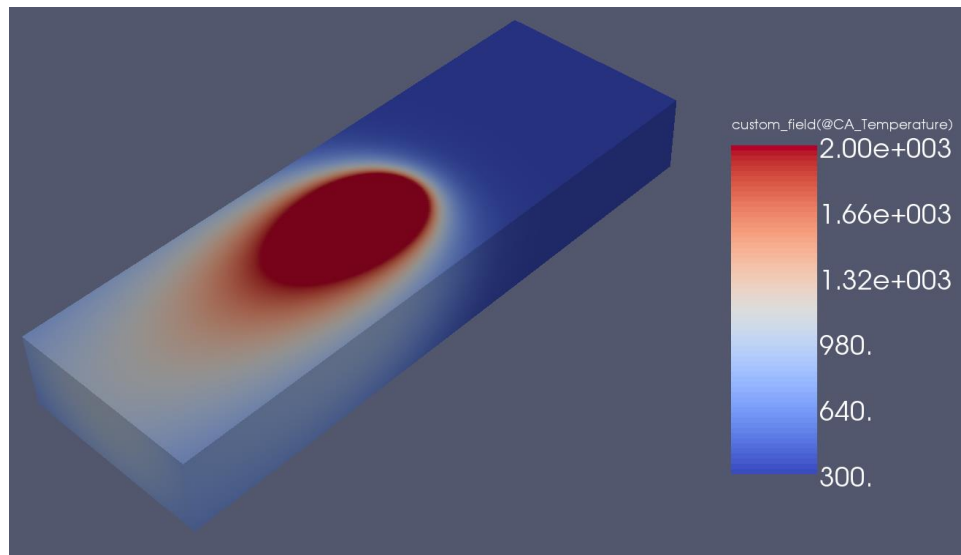


Figure 7. Rosenthal temperature field for welding simulation.

The physical time duration of the simulation is 2 seconds for the heat source to fully scan across the X dimension of the calculation domain. The thermal history is defined as:

```
<node time="0" T="1900" />
```

```
<node time="2" T="1700" />
```

The temperature 1900 and 1700 K will be overwritten by the Rosenthal thermal field. The user only needs to define the initial time and the end time (2 seconds) when using the Rosenthal thermal field.

As an initial grain structure exists before the welding process, the `s_init_grain` option in the PFDB file is set to 1, and an initial grain size of 200 μm is defined (`init_grain_size`).

No bulk nucleation or surface nucleation is considered in this example. The initial grain structure and the grain growth during the welding are shown in Figure 8.

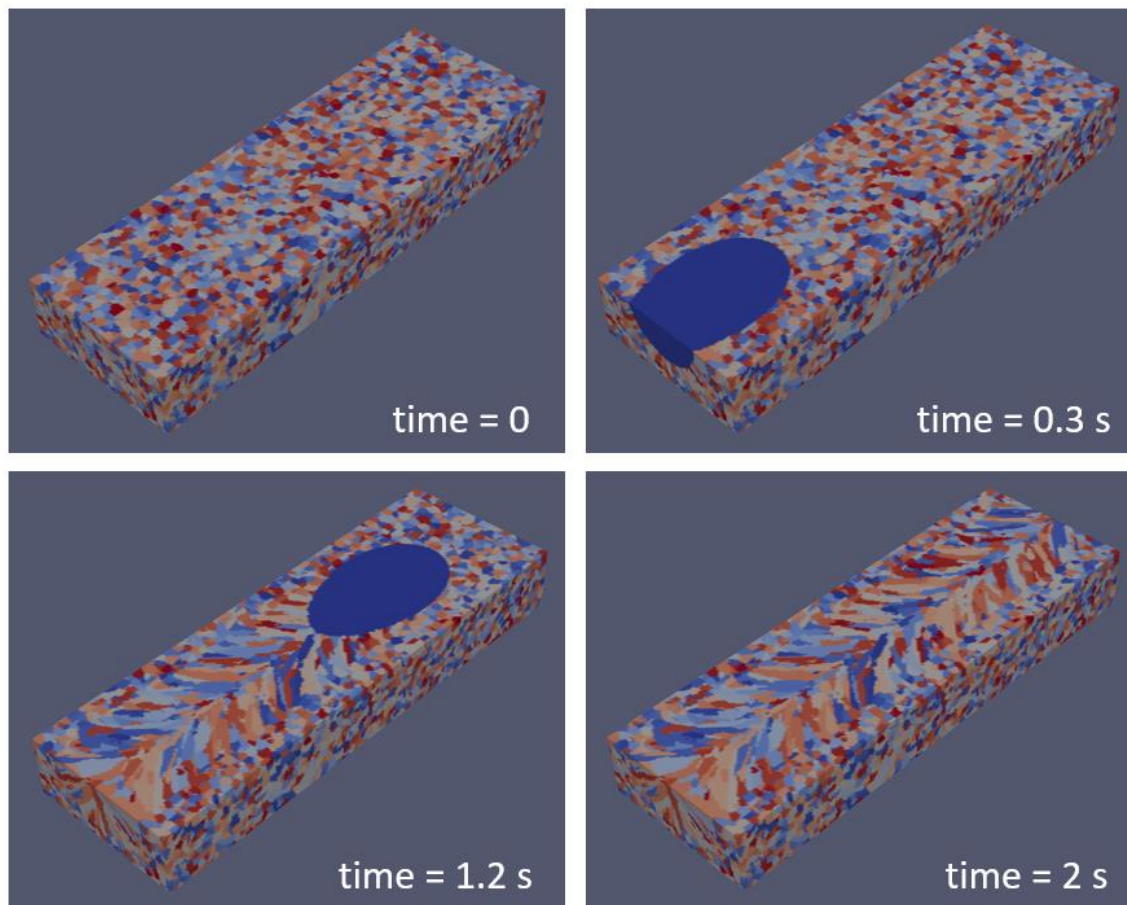


Figure 8. Grain structure evolution in welding.

In the welding simulation, the grain ID inside the molten metal (blue region in Figure 8) is erased. The solidification front follows the molten pool as the latter moves with the scanning heat source along the +X direction. The grain nucleation is not introduced by the surface or bulk nucleation. Instead, the grain nucleates at the fusion line where the existing grains are partially melted (epitaxial nucleation, also see the Theory section). The nucleated grains inherit the grain ID of the existing grains and grow towards the solidification front, as shown in Figure 8.

Theory

Nucleation

There are three types of nucleation mechanisms upon solidification, referred to as the **surface nucleation**, the **bulk nucleation**, and the **epitaxial nucleation**. The new grains can nucleate at any of the six surfaces of the cuboid domain (surface nucleation) or in the interior of the domain (bulk nucleation). For bulk and surface nucleation, the number of nucleation seed is determined by the nucleation densities. The location of the nucleation is randomly distributed at the surface or interior of the domain. A nucleation undercooling ΔT_{nuc} is associated with each nucleation seed. The nucleation event will occur when the temperature drops below a nucleation temperature $T_{nuc} = T_L - \Delta T_{nuc}$ where T_L is the liquidus temperature. The nucleation undercooling ΔT_{nuc} is generated based on a Gaussian distribution, with a mean value $\Delta T_{nuc,m}$ and a standard deviation $\Delta T_{nuc,\sigma}$. The nucleation density, nucleation undercooling mean, and nucleation undercooling standard deviation can be defined in the PFDB file. The newly nucleated grain from the surface and bulk nucleation will have a random crystal orientation (see section Crystal Orientation).

Besides surface and bulk nucleation, the new grain can also nucleate on the fusion line of a partially melted grain. This is the epitaxial nucleation. As shown in Figure 9, a high temperature can partially melt the initial grains, creating a fusion line. Thereafter, a decreasing temperature trigger solidification. The grains preferably nucleate at the fusion line with a large nucleation density and a small nucleation undercooling due to the lowered activation energy. The new grains will inherit the orientation from the existing grains, as shown in Figure 9.

In the CA simulation, the user can define the initial grains with a specified grain size in the PFDB file. The algorithm for generating the initial grain structure is described in [10]. The epitaxial nucleation takes precedence over the surface and bulk nucleation. The epitaxial nucleation is assumed to occur at each mesh cell containing the fusion line with a nucleation undercooling of zero.

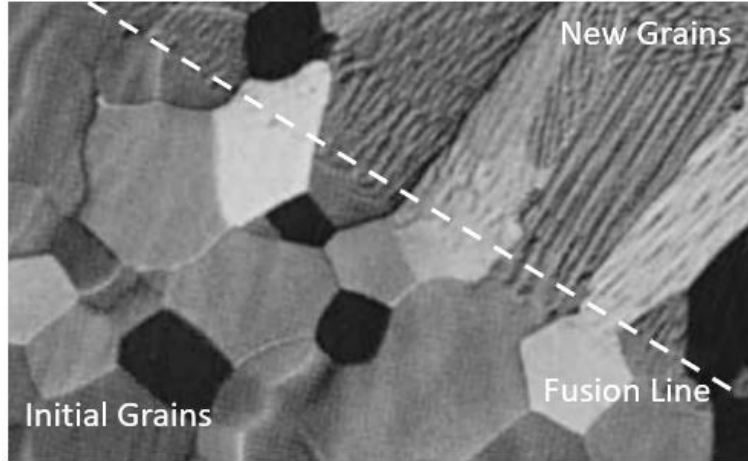


Figure 9. Epitaxial nucleation.

Crystal Orientation

The crystal orientation of a grain refers to a set of rotation operations that map a reference coordinate system (X, Y, Z) to a crystal coordinate system (x, y, z) . Each grain has a unique crystal coordinate and its lattice structure is aligned with its crystal coordinate axis (x, y, z) . The reference coordinate system is the coordinate system (X, Y, Z) in the Cartesian domain (Figure 1), also shown by the blue arrows in Figure 10. The rotation operations from (X, Y, Z) to (x, y, z) is characterized by three Euler angles (ϕ_1, Φ, ϕ_2) . First, the coordinate system is rotated with respect to the Z axis by ϕ . The X axis is rotated to the N axis. Then, the coordinate system is rotated with respect to the N axis by Φ . The Z axis becomes the z axis. Finally, the coordinate system is rotated by the z axis by ϕ_2 . The N axis becomes the x axis, and the y axis is defined accordingly. A unique grain ID (a floating-point number ranging from 0 to 100) is associated with the crystal orientation (ϕ_1, Φ, ϕ_2) .

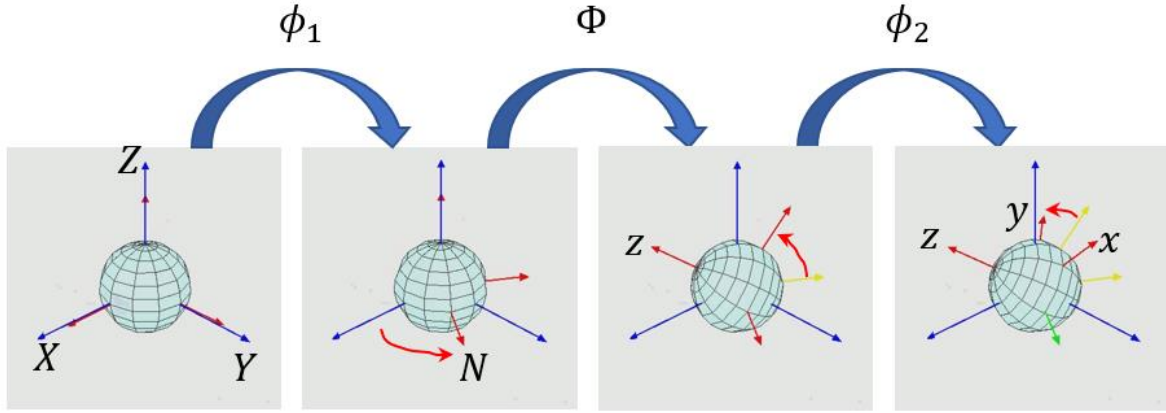


Figure 10. Crystal orientation defined by Euler angles.

Grain Growth

The current CA model considers the grain growth in BCC and FCC metals where the $\langle 100 \rangle$ crystal directions are the preferred grain growth direction. Under a uniform temperature field, a single grain growth leads to an octahedron morphology. The growth velocity of the octahedron contour V_g is determined by the local temperature T , or rather the local undercooling $\Delta T = T_L - T$ where T_L is the liquidus temperature. We compute $V_g(\Delta T)$ based on the thermodynamic and kinetic properties provided in TDB/PDB and PFDB files. The implementation is a modified version of the KGT model in [11, 12].

In the CA simulation, each mesh cell is associated with a grain ID (a floating-point number ranging from 0 to 100). The cells with the same ID constitute one unique grain. The grain ID is set to 0 if the cell is in the liquid state ($T > T_L$). The grain growth is simulated by essentially assigning the nonzero ID to each mesh cell until the entire domain is assigned with a nonzero ID. This process is programmed based on the Cellular Automata method with a de-centered octahedron growth algorithm [5].

Columnar and Equiaxed Grain Structure

Columnar and equiaxed grain morphology refers to the large (columnar) and small (equiaxed) aspect ratio of grains. As the grain growth follows the direction of the solidification front, columnar grains form when there is directionality in the solidification

front movement (e.g., directional solidification), and equiaxed grains form when the solidification moves at the same speed towards all directions (e.g., a single-grain growth under uniform undercooling). For columnar-dominated structure, it is typical that the grains whose $\langle 100 \rangle$ directions better aligned with the solidification front direction outcompetes the less aligned ones, leading to several large columnar grains dominating the structure. This is referred to as the “competitive growth mechanism” [1].

In a directional solidification process, the dominance of columnar or equiaxed grains is determined by the temperature gradient G , the solidification front velocity V , the bulk nucleation density N_0 , and the bulk nucleation undercooling ΔT_N . Based on Hunt’s theory on columnar and equiaxed growth, we can use a “G-V map” to identify if the grain structure is dominated by columnar grains, equiaxed grains, or a mixture of both. Examples of such G-V maps are shown in Figure 11. Two partitioning lines are defined in the G-V diagram for a given set of $(N_0, \Delta T_N)$ to outline the regions of columnar, equiaxed, and mixed grain structure. Small G and large V create a favored condition to trigger bulk nucleation and formation of equiaxed grains. Large G and small V hinders the bulk nucleation and lead to columnar grains. As can be seen in Figure 11, smaller ΔT_N and larger N_0 shifts the two partitioning lines from the lower right to the upper left corner of the G-V diagram, indicating that smaller ΔT_N and larger N_0 favors the formation of equiaxed grains. The detailed derivation of such G-V maps is referred to [8].

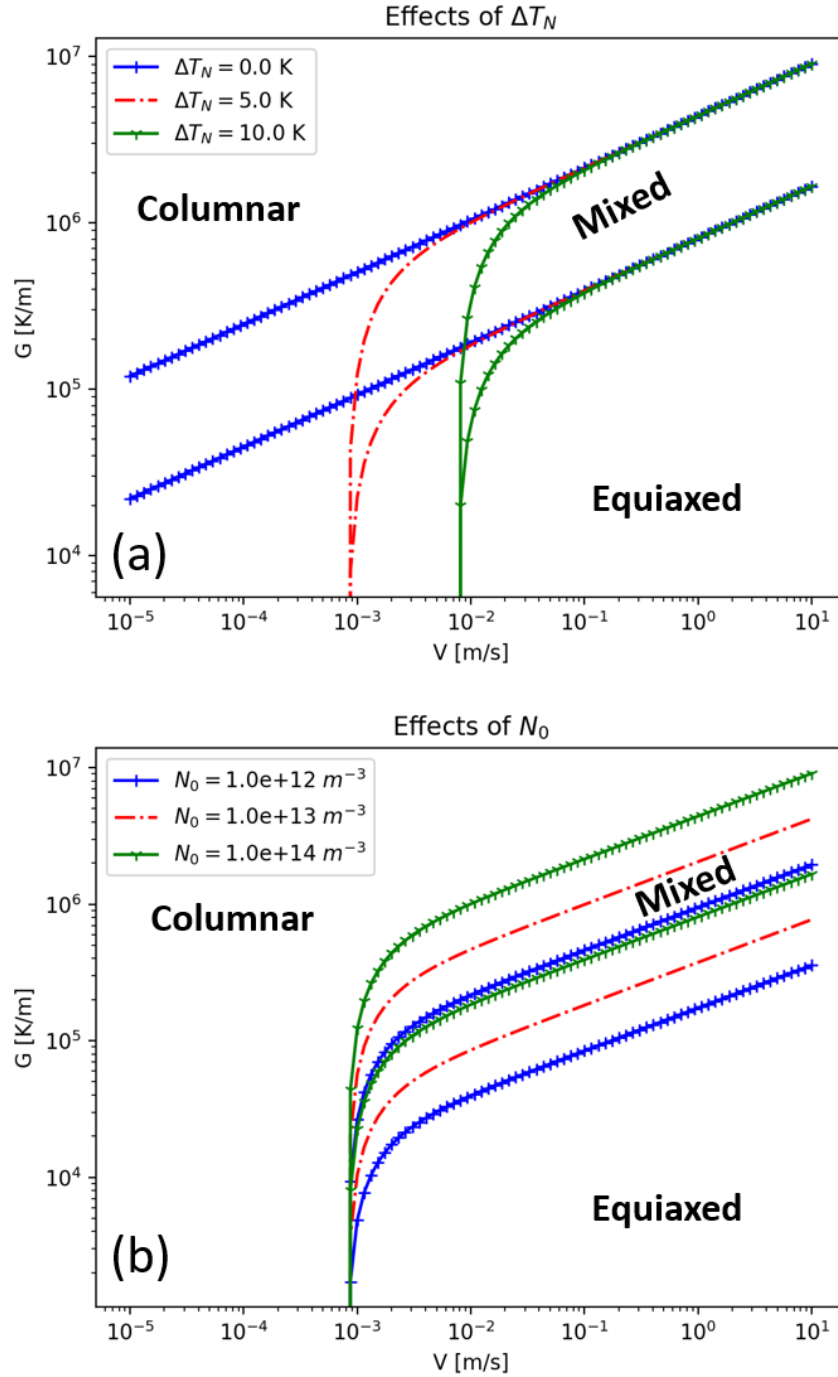


Figure 11. Examples of G-V maps. Given a $(N_0, \Delta T_N)$, two partitioning lines are defined in the G-V map to outline the regions of columnar, equiaxed, and mixed grain structure. (a) G-V map with different ΔT_N . (b) G-V map with different N_0 .

Sensitivity and Numerical Error

The mesh size Δx and time step size Δt can affect the accuracy of the simulation. The user needs to estimate the spatial and temporal scales of the specific problem and define Δx and Δt accordingly. The mesh size needs to resolve the grain geometry and is typically set to be at least 1/10 of a typical grain size. The time step size needs to resolve the grain growth across each mesh cell (Δx). We write the relationship between time step size and mesh size as Eq. (4):

$$\Delta t = \alpha \frac{\Delta x}{V^*} \quad (4)$$

where α is a constant (specified by `dt_alpha` in the PFDB file) and V^* is a characteristic grain growth velocity. V^* can be estimated by $V_g(\Delta T^*)$, where V_g is the growth kinetics function and ΔT^* is the characteristic undercooling. In the CA simulation, V^* is updated as the maximum grain growth velocity at each time step as calculated by Eq. (3). Numerical error can be caused if the user chooses an α value larger than 1 (0.2 is recommended).

Another source of numerical error can be from a large temperature gradient G , such that the spatial resolution Δx cannot resolve the temperature field well. In the de-centered octahedron growth algorithm, there is a temperature error of $\Delta T_{err} \sim G\Delta x$ that contributes to the error of octahedron growth velocity $V_{err} \sim V_g(\Delta T_{err})$. **We suggest the user to check a typical grain growth velocity in the simulation (V^*) and make sure that $V^* > 0.1V_{err}$. If not, the mesh size should be reduced.**

In some cases, the temperature gradient is very large (e.g., welding scenarios) and it may not be feasible to reduce mesh size Δx because of the high computational cost. In such a case, we suggest setting a velocity cap (`CA_Vcap`) in the PFDB file, so that the velocity error V_{err} will not cause unphysically large grain growth velocity and trigger numerical instability. This velocity cap should be defined based on the nature of the specific problem. When the Rosenthal thermal field is used, the velocity cap is automatically set to the welding speed (`Rosenthal_V`).

References

- [1] S. Kou, “Welding metallurgy,” *New Jersey, USA*, pp. 431-446, 2003.
- [2] K. C. Mills, *Recommended values of thermophysical properties for selected commercial alloys*. Woodhead Publishing, 2002.
- [3] N. Eustathopoulos, “Energetics of solid/liquid interfaces of metals and alloys,” *International metals reviews*, vol. 28, no. 1, pp. 189-210, 1983.
- [4] H. Jones, “The solid–liquid interfacial energy of metals: calculations versus measurements,” *Materials Letters*, vol. 53, no. 4-5, pp. 364-366, 2002.
- [5] C.-A. Gandin and M. Rappaz, “A 3D cellular automaton algorithm for the prediction of dendritic grain growth,” *Acta Materialia*, vol. 45, no. 5, pp. 2187-2195, 1997.
- [6] C.-A. Gandin, “From constrained to unconstrained growth during directional solidification,” *Acta Materialia*, vol. 48, no. 10, pp. 2483-2501, 2000.
- [7] Y. Zhang, J. Zhou, Y. Yin, X. Shen, T. A. Shehabeldeen, and X. Ji, “GPU-Accelerated cellular automaton model for grain growth during directional solidification of nickel-based superalloy,” *Metals*, vol. 11, no. 2, p. 298, 2021.
- [8] J. Hunt, “Steady state columnar and equiaxed growth of dendrites and eutectic,” *Materials science and engineering*, vol. 65, no. 1, pp. 75-83, 1984.
- [9] S. Chen, G. Guillemot, and C.-A. Gandin, “Three-dimensional cellular automaton-finite element modeling of solidification grain structures for arc-welding processes,” *Acta materialia*, vol. 115, pp. 448-467, 2016.
- [10] X. Li and W. Tan, “Numerical investigation of effects of nucleation mechanisms on grain structure in metal additive manufacturing,” *Computational Materials Science*, vol. 153, pp. 159-169, 2018.
- [11] W. Kurz, B. Giovanola, and R. Trivedi, “Theory of microstructural development during rapid solidification,” *Acta metallurgica*, vol. 34, no. 5, pp. 823-830, 1986.
- [12] J. Lipton, M. E. Glicksman, and W. Kurz, “Dendritic growth into undercooled alloy metals,” *Materials Science and Engineering*, vol. 65, no. 1, pp. 57-63, 1984.