

OpenPhase: Multi-Phase-Field Solidification with Pandat

A hands-on tutorial for the
OP_Solidification_Multicomponent plugin

This tutorial covers the three input files a simulation needs, then works through the seven examples that ship with the plugin. Every figure was produced by running the corresponding example unmodified.

Part I describes the file formats. Part II presents the examples. Part III covers designing your own case. Read Part I first even if you only intend to run an example, because the later parts refer back to it.

Contents

1	What this plugin does	3
1.1	The three files	3
1.2	What you need	3
	Part I: Writing the input files	4
2	The .pbfx file: alloy, domain, thermal history	4
2.1	Databases	4
2.2	Components and phases	4
2.3	Composition	4
2.4	Thermal history	4
2.5	Grid and output	5
2.6	Phase-field resolution	5
2.7	Optional output fields	9
3	The .pfdb file: physics and numerics	9
3.1	Interface properties (@InterfaceProperties)	10
3.2	Thermodynamics (@ThermodynamicPropertiesEQP, @EquilibriumPartitioning) . .	10
3.3	Line compounds (@Composition)	10
3.4	Diffusion (@ChemicalDiffusion)	11
3.5	Nucleation (@Nucleation)	11
3.6	Time stepping and boundaries	11
4	The .pfin file: the initial microstructure	12
4.1	Crystal orientation in a quasi-2D domain	12
	Part II: The examples	14
5	Overview	14
6	Example 1: the Fe-C peritectic	14
7	Example 2: columnar growth in a thermal gradient	17
8	Example 3: the Al-Cu eutectic	17
9	Example 4: ternary Fe-Cr-Ni	18
10	Example 5: nine grains meeting	20

11 Example 6: one grain growing	22
12 Example 7: columnar growth with a second phase	24
Part III: Designing your own case	27
13 A working recipe	27
14 Sizing the run	27
15 Extrapolation mode	29
16 What to expect	29

1 What this plugin does

The plugin couples two solvers that are normally used separately:

- **OpenPhase** evolves the interfaces, using a *multi-phase-field* formulation in which every phase and every grain carries its own order parameter. It governs how grains grow, impinge and coarsen, and how nuclei appear.
- **Pandat** supplies the thermodynamics. At every grid point it answers: given this local composition and temperature, what is in equilibrium, and what is the driving force towards it?

The practical consequence is that **you do not supply a phase diagram, a partition coefficient or a liquidus slope**. You supply a CALPHAD database and a composition, and the thermodynamics of the alloy is evaluated directly at every point. This is what makes multicomponent problems tractable. A seven-element superalloy is no harder to set up than a binary, and each element partitions in whichever direction the database dictates.

1.1 The three files

File	Answers	Format
.pbfx	Which alloy, how large a domain, what thermal history?	XML
.pfdb	What are the physical and numerical parameters?	XML
.pfin	What is the microstructure at $t = 0$?	one command per line

A CALPHAD database (.pdb or .rtdb) is also required, which you do not write. The .pbfx names the other two files, so it is the one you pass to Pandat.

1.2 What you need

Pandat with the plugin installed, a database covering your alloy, and a text editor. No compiler and no source code are required. For post-processing, use ParaView (<https://www.paraview.org>), which reads the VTK output directly.

Part I: Writing the input files

2 The .pbfx file: alloy, domain, thermal history

This is the top-level file. It is XML with five blocks of interest.

2.1 Databases

Names the CALPHAD database and the parameter file. Paths are relative to the .pbfx.

```
<databases>
  <database type="pdb" file_name="PanNi2026_TH+MB.pdb" />
  <database type="pfdb" file_name="SingleNucleus.pfdb" />
</databases>
```

2.2 Components and phases

List every element to be carried and every phase permitted to exist. Suspend all phases first, then enter the ones you want. This avoids picking up phases present in the database that you did not intend to model.

```
<phases>
  <phase name="*" status="Suspended" />
  <phase name="Liquid" status="Entered" />
  <phase name="Fcc" status="Entered" />
  <phase name="Laves_C14" status="Entered" />
</phases>
```

A phase that is Entered but never nucleates still carries a cost, because the solver continues to query equilibria for it. Enter only the phases you expect to observe.

2.3 Composition

Given as one `statespace` block. Units come from the `<units>` block above it. With `x%` the values are mole percent and must total 100; with `w%` they are weight percent, which is the usual convention for steels.

```
<statespace>
  <T value="1625" /> <P value="1" />
  <n component="Al" value="1" /> <n component="Cr" value="21" />
  <n component="Fe" value="17" /> <n component="Mo" value="2" />
  <n component="Nb" value="3" /> <n component="Ni" value="55" />
  <n component="Ti" value="1" />
</statespace>
```

2.4 Thermal history

A list of (time, temperature) nodes, linearly interpolated. These set the temperature at the *gradient reference point*. If the .pfin declares a temperature gradient, every other point is offset from that reference.

```
<thermal_history>
  <node time="0" T="1624" />
  <node time="0.125" T="1524" /> <!-- 800 K/s -->
</thermal_history>
```

Beyond the last node the temperature is held constant rather than extrapolated, so let the last node cover the whole intended run. Place it at the temperature where you want solidification to be complete.

2.5 Grid and output

```
<phasefield>
  <number_of_grids_x value="1" /> <!-- 1 = quasi-2D -->
  <number_of_grids_y value="80" />
  <number_of_grids_z value="400" />
  <length_scale value="1.0e-06" /> <!-- cell size [m] -->
  <number_of_threads value="8" />
  <initial_mode value="Additive.pfin"> <input_folder_path value="" /> </initial_mode>
  <num_of_profile_outputs value="200" />
  <profiles>
    <profile time="0" /> <profile time="0.03" /> <profile time="0.125" />
  </profiles>
</phasefield>
```

num_of_profile_outputs distributes snapshots evenly over the whole thermal history. The explicit <profiles> list adds named times in addition. Use the explicit list when the window of interest is much shorter than the run, which is common.

Two stopping conditions are also set in this block:

```
<start_from_liquidus_surface value="yes" />
<end_when_no_more_liquid value="yes" />
<liquid_amount_min value="0.01" />
```

Size the thermal history so that it ends where you want the run to end. The schedule is the primary control on run length; liquid_amount_min and end_when_no_more_liquid act as a safety net on top of it.

2.6 Phase-field resolution

OPS_GridResolution in the .pfdb chooses whether the phase field is solved on the same grid as everything else or on a grid refined by a factor of two:

```
<Parameter name="OPS_GridResolution" value="1" /> <!-- 0 = Single, 1 = Dual -->
```

The default is Single. Whether it is adequate depends on the diffuse interface thickness, `Interface_Width` $\times$ `length_scale`, against the width of the features you are trying to resolve. An interface as wide as the features cannot represent them: a dendrite arm no thicker than the interface around it cannot carry a side branch, whatever the interface anisotropy says.

The difficulty is that you do not know the feature width in advance, and you cannot measure it from a Single-resolution run, because that run has already smoothed away whatever it could not resolve. Figure 1 shows the trap concretely. In 05_EquiaxedAlMgZn, at $dx = 1 \mu\text{m}$ with `Interface_Width` = 5, the Single result looks like a set of clean four-lobed grains about $15 \mu\text{m}$ across. Against a $5 \mu\text{m}$ interface that is a ratio of one third and looks comfortable. Run the same case at Dual and the grains are branched dendrites whose arms are $5 \mu\text{m}$ wide, so the interface was in fact the same size as the real features. The reassuring $15 \mu\text{m}$ measurement was itself the artifact.

So treat the ratio as an explanation rather than a test.

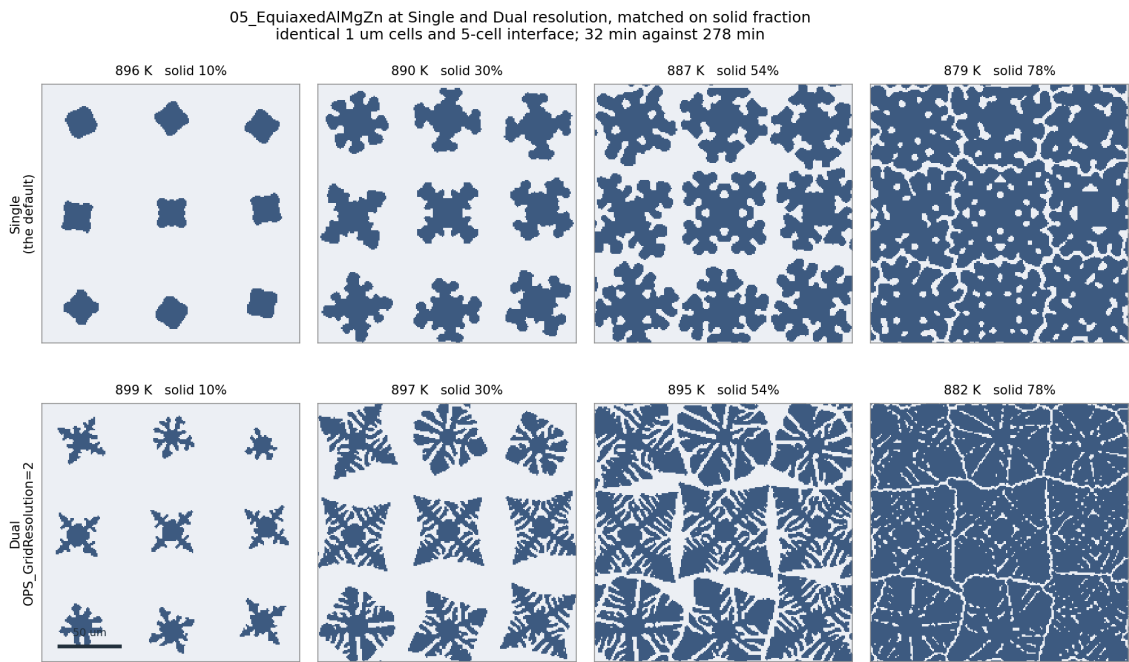


Figure 1: 05_EquiaxedAlMgZn at Single and Dual resolution, compared at matched solid fraction, over the whole run. The cell size and every physical parameter are identical and only `OPS_GridResolution` differs, but note that this does NOT hold the physical interface width fixed: see the warning below. Dual also reaches the same solid fraction 3 to 8 K warmer, because a sharper interface needs less undercooling to advance, and it ends at 81.8% solid against 78.7%. The integral quantities move by a few percent; the microstructure changes completely.

Switching `OPS_GridResolution` at a fixed `Interface_Width` is NOT a convergence test. `Interface_Width` is counted in *grid cells*, and Dual solves the phase fields on a grid of half the spacing, so the same value of 5 gives a physical interface half as wide. A thinner interface is a different model, not a better-resolved one.

Best practice for dual resolution

Three of the plugin's inputs are counted in *grid cells* rather than in metres, so changing the cell size or the resolution changes the physics they describe unless you rescale them:

Input	What it means physically
<code>Interface_Width</code>	The interface is this many cells wide, so its physical width is the value multiplied by the phase-field spacing. Dual halves that spacing.
<code>NUC_ShieldingRadius</code>	Minimum separation between nuclei, in cells of the nucleation grid. The physical exclusion distance is the value multiplied by dx .
nucleus size	A planted nucleus takes its size from the interface width, not from a physical radius, so it shrinks with the interface. A smaller nucleus carries more curvature and needs more driving force to survive.

Recommended recipe. To refine the phase field while leaving the model alone, switch to Dual *and* double dx , halving the grid dimensions so the domain stays the same size. The phase fields then sit on the spacing you had before, the interface keeps its physical width, and only the solute grid is coarsened. Halve `NUC_ShieldingRadius` at the same time, from 5 cells to 2.5, so the exclusion distance is unchanged. Doubling `Interface_Width` to 10 instead achieves the same interface width on the finer grid, at the full cost of the finer grid.

Same temperature (808.0 K), same 892 nucleation sites: the structural length scale follows the physical interface width, not the resolution

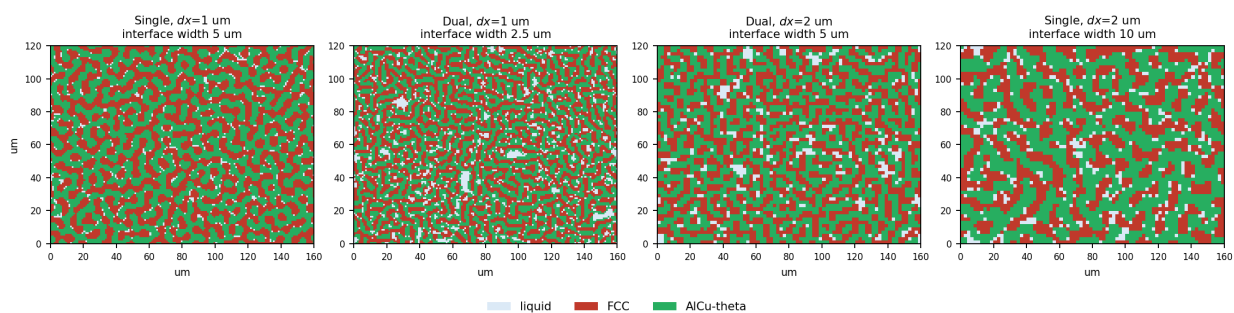


Figure 2: Example 3 at 808 K in all four runs, from the same nucleation sites. The structural length scale follows the physical interface width: halving it halves the lamellar spacing, doubling it coarsens the structure. Compare the first and third panels, which have the same interface width on different grids and produce the same scale of structure, against the second, which is the same grid as the first but half the interface width.

What this looks like in practice. The morphology is what moves first. Figure 2 shows the same alloy at the same temperature from the same nucleation sites: the lamellar spacing scales with the interface width, so a case run at half the intended width returns a structure twice as fine as the one the inputs describe. The integral quantities follow. Figure 3 is the same four runs under

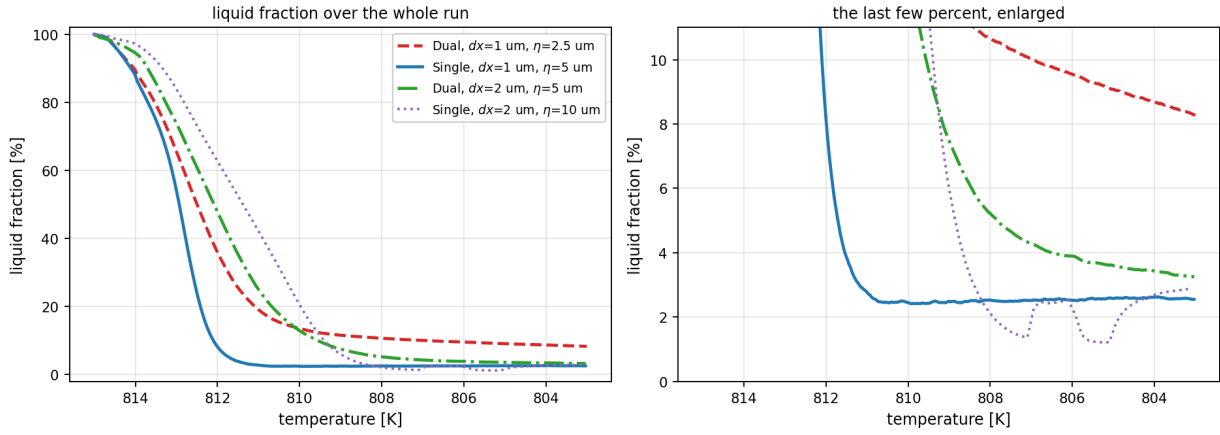


Figure 3: Example 3 on one schedule, with `Interface_Width` = 5 cells in all four runs, so the physical interface width η differs between them. The curves group by η , not by resolution: $\eta = 5$ and $10 \mu\text{m}$ all reach 2.6 to 3.2 % residual liquid, while $\eta = 2.5 \mu\text{m}$ stalls at 8.3 %. The outlier is the only Dual run that did not have dx doubled to compensate.

one schedule with identical seeding. At 803 K the residual liquid is 2.6 % at $\eta = 5 \mu\text{m}$ (Single, $dx = 1 \mu\text{m}$), 3.2 % at the same η reached the recommended way (Dual, $dx = 2 \mu\text{m}$), 2.9 % at $\eta = 10 \mu\text{m}$, and 8.3 % at $\eta = 2.5 \mu\text{m}$. Only the halved interface departs, and it departs by a factor of three: it preserves small grains that a wider interface absorbs, which keeps the structure fragmented, the curvature high, and the last liquid trapped between grains.

Dual is not the expensive part. Those four runs took 74, 26, 13 and 233 minutes respectively. The recommended Dual case was *three times faster* than the Single case it matches, because the phase fields are resolved identically while the CALPHAD work runs on a quarter as many cells. The 233-minute run is the one that gets the different answer. Dual with a halved interface width is what costs; Dual itself can save.

One caveat. The seeded nucleus count is not strictly independent of the nucleation grid: the same density gave 892 nuclei at $dx = 1 \mu\text{m}$ and 686 at $dx = 2 \mu\text{m}$, because the shielding test is applied on discrete cells. The effect here is small next to the interface width, but it means a resolution change is never perfectly neutral for a nucleation-driven case.

With the physical width held fixed, a genuine comparison becomes possible: if the morphology still changes, Single was suppressing structure; if only the arm edges sharpen, Single is adequate.

Dual is not free. It multiplies the phase-field work per step by four in 2D, and it also shortens the time step. The mechanism is not the diffusive dx^2 limit, which is evaluated on the coarse grid and is unchanged: what tightens is the cap on how far a phase field may move in one step, since the finer grid resolves faster local rates. That is why the step count rises by about a factor of three rather than the factor of four the cell count alone would suggest. Measured on the case in Figure 1: 49 100 steps and 32 minutes at Single against 162 200 steps and 278 minutes at Dual, so 3.3 times the steps and 8.6 times the wall clock. That is a real trade-off: a converged morphology at Dual may be worth less to you than three more alloy compositions at Single. But it should be a decision made after looking, not an assumption.

Resolution can also change the answer outright, not just its sharpness. Example 1 shows this most clearly in this set. Note that switching its resolution also changes the physical interface width, as above, so the two runs differ in both. Run at Dual, as it ships, the peritectic goes to completion: δ -ferrite peaks at 62.5% and is then fully consumed, leaving 0.0% ferrite, 98.7% austenite and 1.3% liquid. The same deck at Single leaves 8.8% ferrite untransformed, with austenite reaching only 87.5%. That is not a blurrier picture of the same result, it is the wrong phase fraction: a reader taking the Single number would conclude that this steel retains ferrite at the end of solidification, which on this composition and database it does not. The cost of the difference is 204 700 steps

and 30 minutes against 417900 steps and 101 minutes.

0 selects Single and 1 selects Dual. Any value above 1 is also read as Dual.

2.7 Optional output fields

By default each VTK file carries the phase fractions `eta(@<phase>)`, the total mole fractions `x(<comp>)`, and TK, PhaseID and GrainID. Four further groups can be switched on in the `.pfdb`:

```
<Parameter name="OUT_WritePhaseCompositions" value="1" /> <!-- x(<comp>@<phase>) -->
<Parameter name="OUT_WriteDiffusivity" value="1" /> <!-- D(<comp>@<phase>) -->
<Parameter name="OUT_WriteInterfaces" value="1" /> <!-- phase-field count -->
<Parameter name="OUT_WriteDrivingForce" value="1" /> <!-- dGavg / dGraw -->
```

`x(<comp>@<phase>)` gives the per-phase mole fractions, which is what shows partitioning directly rather than through the mixture value. `D(<comp>@<phase>)` gives the diagonal interdiffusion coefficients actually used, which is the only way to check what the mobility database is supplying. `Interfaces` counts the phase fields present at a point: 1 in the bulk, 2 across a two-phase interface, 3 or more at a junction, which makes it easy to mask analysis to the interface region.

`dGavg(<A>_<B>)` and `dGraw(<A>_<B>)` give the chemical driving force between each pair of phases, averaged over the interface and raw respectively. They are written per phase pair rather than per phase, and only for the distinct pairs: the quantity is antisymmetric, so a pair appears once and the reverse ordering is its negative. Use them when a phase grows or dissolves more slowly than expected, since they show where in the domain the driving force actually sits rather than the single console figure.

They are opt-in because the field count grows as the number of phases times the number of components, so a seven-component alloy with three phases adds eighteen diffusivity fields and twenty-one composition fields to every file. All of them are written into the *same* file on the *same* grid as the phase fractions, so a phase can be correlated with its own composition and diffusivity point by point without resampling.

3 The `.pfdb` file: physics and numerics

Also XML, organised into named groups. Every entry carries a `description` attribute, so the file is self-documenting. The parameters below are the ones you are most likely to adjust. Parameters that name a phase pair follow the pattern `<PREFIX>_<A>_<B>`, for example `IP_Sigma_LIQUID_FCC`.

3.1 Interface properties (@InterfaceProperties)

Parameter	Typical	Meaning
IP_Sigma_A_B	0.24	Interface energy [J/m ²]. Sets the capillary length, and therefore the tip radius and arm spacing.
IP_Mu_A_B	1e-8	Interface mobility [m ⁴ /(J s)]. Higher values make growth less interface-limited and more diffusion-limited.
IP_EnergyModel_A_B	2	0-Ext, 1-Iso, 2-Cubic, 3-CubicFull. Cubic provides the four-fold anisotropy that produces dendrites.
IP_MobilityModel_A_B	2	Same enumeration, for mobility anisotropy.
IP_EpsilonE/M_A_B	0.0	Anisotropy strength for the model selected above.
IP_RegularizationFactor	1.0	Triple-junction regularisation. A value of 1.0 leaves it inactive.

Between two *solid* phases, set IP_Mu several orders of magnitude below the solid/liquid value, typically 1e-11. Two solids growing from a melt are each in local equilibrium with the liquid rather than with one another, and solid/solid interface migration is negligible on solidification timescales.

3.2 Thermodynamics (@ThermodynamicPropertiesEQP, @EquilibriumPartitioning)

Parameter	Use	Meaning
TP_ExtrapolationMode	2	0-None, 1-Local, 2-Global. See §15.
DP_ExtrapolationMode	2	As above, for diffusion data. Keep equal to TP_ExtrapolationMode.
EQP_PartitioningModel	3	0-Optimization, 1-Eiken, 2-Steinbach, 3-TwoStep. Determines how the composition of a cell is divided between coexisting phases.
TP_DrivingForceModel	0	0-Standard, 1-LowerSlope, 2-Weighted, 3-User.
TP_MaxTemperatureDeviation	1.0	How far [K] the local state may drift before the stored tie line is recomputed. Smaller values are more accurate and considerably slower.

3.3 Line compounds (@Composition)

An intermetallic with a fixed sublattice ratio, such as a Laves or sigma phase, does not have a freely variable composition. Solving for one allows the partitioning step to move it away from stoichiometry. Declare it as a line compound instead:

```
<Parameter name="CX_Stoichiometric_LAVES_C14" value="1" />
```

This fixes the phase composition at its equilibrium value and excludes the phase from the partitioning redistribution. In Example 7 the Laves composition then holds to within a few tenths of an atbeing the diffuse interface rather than genuine variation.

The declaration pins the composition of each component individually, so it suits a phase with a narrow composition range. Leave it off for a phase with wide substitutional solubility, where the composition should follow the database.

3.4 Diffusion (@ChemicalDiffusion)

Parameter	Typical	Meaning
DF_DiffusionModel	1	0-Diagonal, 1-DICTRA, 2-Pairwise. DICTRA uses the full mobility matrix.
DF_EnableAntitrapping	1	Corrects artificial solute trapping by the diffuse interface. Leave enabled.
DF_EnableLimitsChecking	1	Guards against out-of-range compositions.

3.5 Nucleation (@Nucleation)

Nucleation is disabled by default for each phase pair. Enabling it makes the whole NUC_* set for that pair mandatory.

```
<Parameter name="NUC_Allowed_LAVES_C14_LIQUID" value="1" />
<Parameter name="NUC_Location_LAVES_C14_LIQUID" value="0" /> <!-- 0-Bulk -->
<Parameter name="NUC_Density_LAVES_C14_LIQUID" value="1e16" /> <!-- 1/m^3 -->
<Parameter name="NUC_Center_LAVES_C14_LIQUID" value="4e-6" /> <!-- mean size [m] -->
<Parameter name="NUC_Tmin_LAVES_C14_LIQUID" value="1300" />
<Parameter name="NUC_Tmax_LAVES_C14_LIQUID" value="1480" />
<Parameter name="NUC_ShieldingRadius_LAVES_C14_LIQUID" value="5" /> <!-- CELLS -->
```

NUC_Center is the centre of a size *distribution*, not a critical radius. A nucleus is introduced without a size and is treated in stages, with the full curvature contribution applied once the grain exceeds roughly five grid cells. There is therefore no threshold below which a nucleus is automatically lost.

NUC_ShieldingRadius is the minimum separation enforced between nuclei, and it is counted in GRID CELLS, not in metres. The physical exclusion distance is therefore NUC_ShieldingRadius multiplied by dx , so changing the cell size silently changes the seeding geometry unless the value is rescaled. It defaults to 5 cells.

Set NUC_Tmax from the temperature at which the phase becomes stable, which a Scheil calculation gives directly. Place it a little below that temperature: each phase pair receives essentially one attempt at its site density, and a small undercooling gives the new grains the driving force to establish.

3.6 Time stepping and boundaries

RTC_dt	1e-7	Initial time step [s].
AdaptiveTimeModel	1	0-ConstantDT, 1-AdaptiveDT. Leave adaptive.
BC_BC0Y etc.	1	Per face: 0-Periodic, 1-NoFlux, 2-Free, 3-FreeGradient, 4-Fixed, 5-Mirror.

4 The .pfin file: the initial microstructure

Commands, one per line, executed in the order written. A `#` starts a comment that runs to the end of the line, and blank lines are ignored, so a case with several seeds can be laid out and annotated rather than packed onto one line.

```
# fill the domain, then place one seed
SINGLE:LIQUID
SPHERE:3:(0,100,100):(15,0,0,XZX):FCC
```

Semicolons separate commands as well, so an existing single-line file keeps working unchanged and the two styles may be mixed in one file:

```
SINGLE:LIQUID;SPHERE:3:(0,100,100):(15,0,0,XZX):FCC
```

Command	Meaning
SINGLE:PHASE	Fill the domain with one phase. Always the first command.
SPHERE:r:(x,y,z):ORIENT:PHASE	Place a sphere of radius <code>r</code> cells at a grid position.
NUCLEUS:(x,y,z):ORIENT:PHASE	Place a zero-volume nucleus.
TEMPGRADIENT:(x,y,z):(gx,gy,gz)	Impose a gradient [K/m] about a reference point.

ORIENT is either RANDOMPLANE or explicit Euler angles such as (0,25,0,XZX).

A NUCLEUS is planted through the same code path the nucleation module uses, so the grain gets the normal bookkeeping and `PF_NucleusVolumeFactor` applies to it. It starts at zero volume and grows normally once there is a driving force: Example 1 grows a single NUCLEUS into the whole primary dendrite. The choice matters when the driving force is marginal, and the two fail differently. A nucleus has no volume to lose, so if the driving force is not there it disappears immediately and does not come back. A sphere shrinks slowly instead, so a seed placed slightly too warm can survive long enough for continued cooling to bring it into the growth regime, and recover. If you are not certain the seeding temperature is far enough below the liquidus, SPHERE is the forgiving choice; a NUCLEUS will simply be gone, and an empty domain is easy to mistake for a thermodynamic or database problem. SPHERE is also the one to use when you want a seed of a chosen radius already present at $t = 0$, to skip an incubation period or to start several grains at a known equal size.

4.1 Crystal orientation in a quasi-2D domain

These cases are quasi-2D, with `number_of_grids_x = 1`. The simulation plane is therefore **y–z** and the axis normal to it is **x**. An in-plane rotation, which is the one that changes the visible orientation of a dendrite, must be a rotation **about x**.

In the XZX convention the three angles are $R_x(\alpha) R_z(\beta) R_x(\gamma)$, giving:

Angles	Rotation	Effect on a y–z domain
$(\theta, 0, 0, XZX)$	about x	in-plane: dendrite arms visibly rotate
$(0, \theta, 0, XZX)$	about z	out-of-plane: tilts the crystal through the plane

Write the angle first to rotate a grain within the simulation plane: $(\theta, 0, 0, XZX)$. This is the form used by the examples that show differently oriented grains, and it is what makes each grain present its own arm directions.

Orientation matters most where it *selects* something. In a directional case the mechanism of interest is that grains aligned with the gradient outgrow misaligned ones, so in-plane angles are

what drive the competition. Orientation does not enter the equations at all when every interface is isotropic (IP_EnergyModel = Iso throughout).

With cubic symmetry the in-plane angle is distinct only modulo 90° , so distribute a set of seeds over 0 to 90° rather than 0 to 360° .

A directional case with competing seeds and a gradient:

```
SINGLE:LIQUID

# three seeds at the cold base, angles spread over 0 to 90 deg
SPHERE:4:(0,7,2):(2,0,0,XZX):FCC
SPHERE:4:(0,23,5):(21,0,0,XZX):FCC
SPHERE:4:(0,37,3):(38,0,0,XZX):FCC

TEMPGRADIENT:(0,0,0):(0,0,2.5e5)
```

Part II: The examples

Each example ships in its own folder under `Examples/`, complete with its database, and runs unmodified. Launch one by passing its `.pbfx` to `PandatX.exe` from within the example folder. A climbing solid fraction in the console output is the indication that the case is running correctly, and snapshots are written as the run proceeds, so early frames can be examined while the remainder accumulate.

5 Overview

The examples are ordered by increasing complexity: binary alloys first, then ternary, then a seven-component superalloy. Within each group, a single given seed comes before multiple seeds, a thermal gradient, and finally microstructure that arrives entirely by nucleation.

That ordering also separates them by database. Examples 1 to 5 use the **demo databases distributed with Pandat**, in `Pandat 2026/Pandat 2026 Example book/` (`Al_Demo.rtdb`, `Fe_Demo.rtdb`), so they need no database licence and are considerably cheaper. Examples 6 and 7 use the seven-component database `PanNi2026_TH+MB.pdb`.

The plugin licence is a separate question from the database licence. If your dongle does not carry an entitlement for this plugin, the plugin still runs, but only for systems of up to three components, counted the way the model counts them so that a binary is two and a ternary is three. A larger system is refused before the first time step, with the component count stated in the message. That boundary falls in the same place as the database one: examples 1 to 5 run either way, and examples 6 and 7 need an entitled plugin.

Folder	System	Comp.	Initial state	Solids
01_PeritecticFeC	Fe-0.26C (w%)	2	one nucleus	2
02_DirectionalAlCu	Al-5Cu	2	seeds + gradient	1
03_EutecticAlCu	Al-17.4Cu	2	nucleation only	2
04_TernaryFeCrNi	Fe-Cr-Ni	3	one particle	1
05_EquiaxedAlMgZn	Al-Mg-Zn	3	nine seeds	1
06_SingleNucleus	Ni superalloy	7	one seed	1
07_AdditiveTwoPhase	Ni superalloy	7	seeds + gradient	2

Examples 1, 3 and 7 carry two solid phases and are the ones to read when a second phase forms in your own alloy, whether by a peritectic reaction, coupled eutectic growth, or precipitation from enriched interdendritic liquid.

6 Example 1: the Fe-C peritectic

`Examples/01_PeritecticFeC`. One BCC (δ -ferrite) nucleus at the centre of a 151×151 domain at $2 \mu\text{m}$, in Fe-0.26 wt% C, with FCC (austenite) nucleating on its interfaces once the peritectic is passed. Compositions are in weight percent, so note `<unit name="n" value="w%" />`.

The carbon level is written as 0.26055 wt% rather than a round number, because that is the exact weight-percent equivalent of the mole fraction $x_C = 0.012$. Rounding it to 0.261 makes the alloy 0.17% richer in carbon. That is far too small to matter here, but it is worth knowing that a weight-percent field carries a conversion: when a case has to reproduce a composition specified in mole fraction, convert with enough digits rather than to three decimal places.

For this composition and database the peritectic is at 1767.90 K. The case runs from 1790 to 1740 K at -10 K/s so the window brackets it with room on both sides, and NUC_Tmax for both FCC pairs is set at 1767.9 K.

This case sets OPS_GridResolution to 2 (Dual), and it is one of the cases where the choice changes the answer rather than just sharpening it. The cell size here is $2\text{ }\mu\text{m}$ with Interface_Width = 6, so at Single resolution the diffuse interface is $12\text{ }\mu\text{m}$ across, comparable to the dendrite arms themselves. The grain then grows as a compact, faceted blob with no side arms at all, whatever the interface anisotropy says. The other examples run at $1\text{ }\mu\text{m}$ with a 5-cell interface and are sensitive to this too, so read §2.6 before assuming the default is safe for your own case.

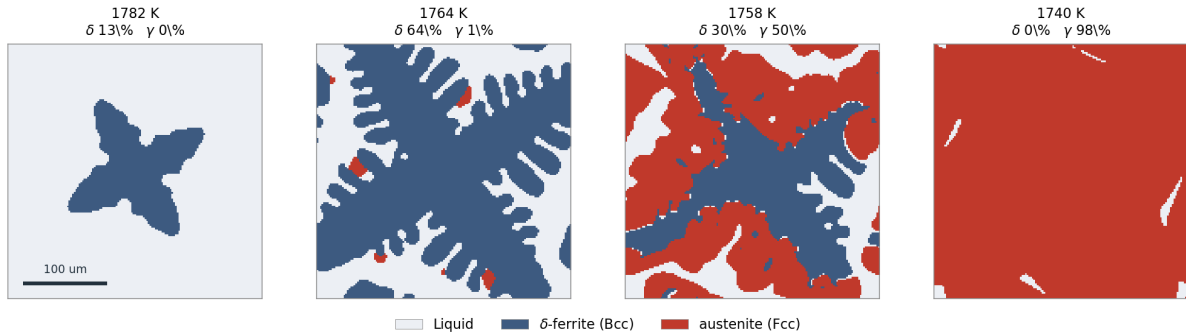


Figure 4: One δ -ferrite nucleus grows into a four-fold dendrite carrying its 15° in-plane orientation, with well developed secondary arms. Once the temperature passes the peritectic, austenite nucleates on the δ /liquid boundaries and spreads along them, consuming the ferrite entirely by the end of the run.

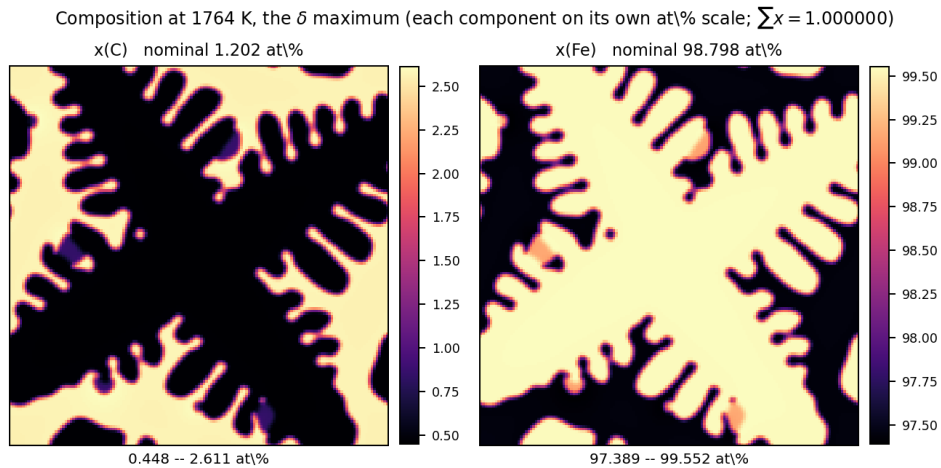


Figure 5: Carbon and iron at the δ maximum, each on its own at% scale because their absolute levels differ by two orders of magnitude. Carbon is rejected into the interdendritic liquid and depleted in the dendrite cores; iron mirrors it exactly, since in a binary the two mole fractions sum to one.

Expected result. Austenite appears at 1767.9 K, essentially at the peritectic. Delta-ferrite rises to a maximum of 62.5% at 1766.0 K and is then consumed; the run finishes with δ fully transformed at 0.0%, austenite at 98.7% and 1.3% liquid remaining. The full schedule takes 417 900 steps and about 101 minutes on eight threads.

The reaction temperature and the endpoint are not specific to the domain size, but the ferrite maximum is mildly so. Measured on the same configuration: the δ maximum is 62.5% here, 63.4% in a 602 μm box and 63.8% in an 802 μm box, so a small frame under-predicts it by about a point. Every size ends at the same place, with ferrite fully consumed and austenite near 98.6%. What does change sharply is whether the dendrite fits at all: a 101×101 box is too small, its arms reach the periodic boundary before the peritectic, and the δ maximum is inflated to 75%.

When adapting a case to a new alloy, take its structure but set the thermal window from a fresh Scheil calculation. The window should open just below the liquidus and close below the reaction of interest, which for a peritectic means bracketing the peritectic temperature itself.

A larger variant

`Examples/01_PeritecticFeC_Large` is the same case at 401×401 in an 802 μm box. It places no seed: bulk nucleation is left on, so it grows five or six nucleated grains into a proper equiaxed mush rather than a single dendrite, and it resolves finer secondary arms.

Expected result. A δ maximum of 63.8% at 1764.4 K, then full consumption of the ferrite: the run finishes at 0.0% δ , 98.6% austenite and 1.4% liquid. 564300 steps and about 13 hours on eight threads.

Use it when the question is about the grain structure. The 151×151 case above reaches the same reaction temperature and the same endpoint in about an hour and a half, and its ferrite maximum is only about a point lower.

This is the case to enable checkpointing on. At `OP_CheckpointEvery = 1` it writes a checkpoint every output profile, roughly every eight minutes, for about 0.2 s and 54 MB each – under 0.04% of the run. A thirteen-hour job then survives a reboot at the cost of a few minutes' work.

Why the solid diffusivity matters here

The peritectic reaction is not limited by thermodynamics but by how fast carbon can cross the austenite layer that forms between the liquid and the ferrite. That makes this case unusually sensitive to the diffusion data, and worth looking at with `OUT_WriteDiffusivity` switched on.

Carbon is interstitial in iron, so its diffusivity in the solid is far higher than substitutional intuition suggests. At 1789 K the database gives about $7 \times 10^{-9} \text{ m}^2/\text{s}$ in δ -ferrite, which is comparable to the value in the liquid, and 1.6×10^{-9} in austenite. Over the 50 K window these fall by only 15% and 26% respectively, following an Arrhenius law with effective activation energies of 82 and 158 kJ/mol.

The consequence is that carbon back-diffuses into the dendrite cores while solidification is still in progress, so the final segregation is milder than a Scheil calculation with no back-diffusion would predict, and the reaction completes at a higher temperature than it would with slower solid transport. If you substitute hand-picked diffusivities for the database values, expect both of those to move.

Diffusivities come from the mobility data in the CALPHAD database, not from anything you write in the `.pfdb`. Switch on `OUT_WriteDiffusivity` once for any new alloy and check the values are what you expect. Under `DP_ExtrapolationMode = Global` they are uniform across the domain, one value per phase, refreshed when the temperature drifts past `MaxTemperatureDeviation`.

7 Example 2: columnar growth in a thermal gradient

Examples/02_DirectionalAlCu. Four seeds at the cold base of a 60×200 domain with a 2.5×10^5 K/m gradient, hypoeutectic Al-5 at% Cu, base cooled from 902 to 852 K over 0.49 s (102 K/s). This is the geometry of Example 7, the additive-manufacturing case, using a demo database.

The 83 K freezing range (liquidus 904.37 K, terminal eutectic 821.37 K) is what makes this a good columnar case, since there is ample Cu partitioning to build a boundary layer ahead of the front. AlCu_Theta is left suspended, because the run does not take any part of the domain down to the eutectic.

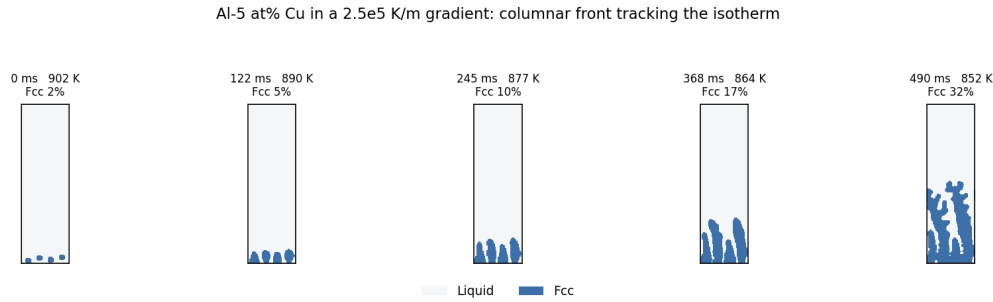


Figure 6: The columnar front advances up the gradient as the base cools, with the interdendritic channels closing behind it.

Matching the cooling rate to the front velocity

The liquidus isotherm sweeps through the domain at $\dot{T}/G$. The solidification front advances at whatever velocity the local driving force sustains, which for these conditions is of order 0.4 mm/s. If the isotherm is faster than the front, undercooling accumulates behind it, the driving force saturates against what the diffuse interface can represent, and additional steps do not help.

With $G = 2.5 \times 10^5$ K/m, a cooling rate of 102 K/s places the isotherm at 0.41 mm/s, which matches the front. The result is a solid fraction of 31.7% at the end of the schedule, with a driving-force ratio whose median is 0.46.

Size a directional case from $\dot{T}/G$ against the front velocity you measure, not from a step budget. Raising the cooling rate beyond what the front can follow increases the undercooling rather than the growth rate.

8 Example 3: the Al-Cu eutectic

Examples/03_EutecticAlCu. Al-17.4 at% Cu at the eutectic composition, 160×120 cells at $1 \mu\text{m}$, isothermal, cooled from 815 to 811.25 K. The domain starts as SINGLE:LIQUID with no seeds at all, so both solid phases arrive by nucleation.

Both interfaces use the same IP_Mu_LIQUID_*, and both phases nucleate within the first timestep and grow together for the remainder of the run.

Expected result. A Scheil calculation on this composition and database gives a liquidus of 821.94 K, the eutectic at 821.38 K, and equilibrium fractions of FCC 48.83% and AlCu_Theta 51.17%. The phase-field run reaches FCC 47.2% and AlCu_Theta 49.7% with 3.2% liquid remaining, which is 48.7 to 51.3 of the solid: agreement with the equilibrium ratio to within 0.2 percentage points. The run takes 17 800 steps and about 16 minutes.

Equal mobilities suit this alloy because the Al-Cu eutectic is close to 50/50 by volume. Where the two phases have very different equilibrium fractions, scale IP_Mu_LIQUID_* in proportion, so that each phase grows at the rate its share of the solid requires. Read the intended ratio from a Scheil calculation first.

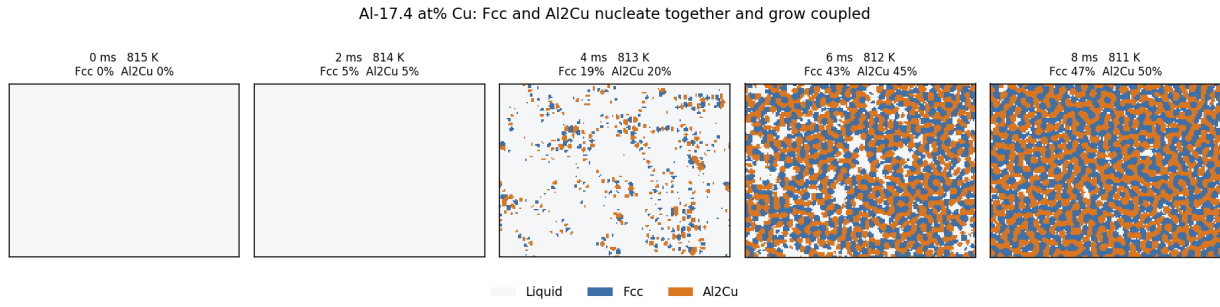


Figure 7: FCC and Al₂Cu nucleate together and grow as an interpenetrating two-phase structure. The two phases remain in nearly equal proportion throughout, which is the signature of coupled eutectic growth rather than of two phases growing independently.

9 Example 4: ternary Fe-Cr-Ni

Examples/04_TernaryFeCrNi. A single BCC particle growing into ternary liquid on a 301×301 grid. This is the case to start from when building a new ternary problem, since it is the simplest configuration that still involves genuine multicomponent partitioning.

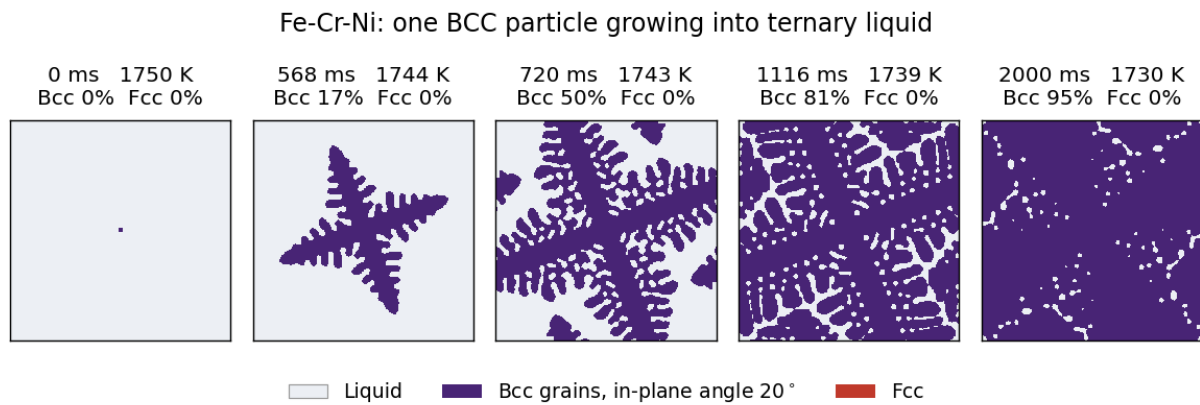


Figure 8: A single BCC particle becomes a four-fold dendrite and fills the domain. The arms follow the grain's 20° in-plane orientation, and the periodic boundaries let the arms re-enter on the opposite face.

Expected result. 402 100 steps and about 253 minutes on eight threads, finishing at BCC 95.0% with 5.0% liquid. This case is also the best-conditioned in the set, with a driving-force ratio whose median is 0.62 and whose maximum is 8.9, so it is a useful reference when judging whether a new case is behaving.

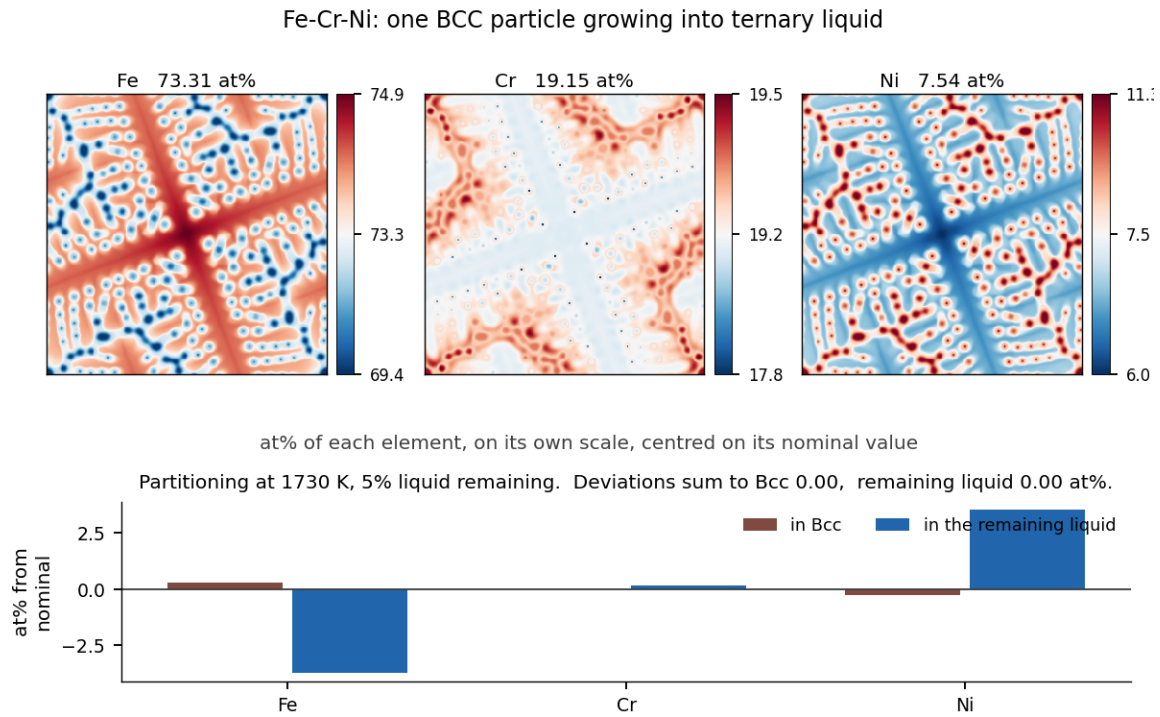


Figure 9: All three elements at the final frame. Each panel is in at%, on its own colour scale spanning that element's own measured range and centred on its nominal value, so blue is depleted and red enriched while the tick labels stay in real mole fraction. A 51 at% solvent and a 1 at% addition have nothing to say to each other on a shared axis. Ni is the element that partitions here, spanning 6.0 to 11.2 at% against a nominal of 7.54, while Fe and Cr move by a couple of percent. The bars give the absolute deviation in at%: Ni gaining 3.5 at% in the liquid is balanced by Fe losing 3.7, and the deviations sum to zero over the elements in both phases.

10 Example 5: nine grains meeting

Examples/05_EquiaxedAlMgZn. Nine FCC seeds on a jittered 3×3 layout in a 200×200 domain at $1 \mu\text{m}$, each given a different in-plane orientation, cooled uniformly from 904 K at 929 K/s. There is no gradient and only one solid phase, so the only thing shaping the result is grains growing into one another.

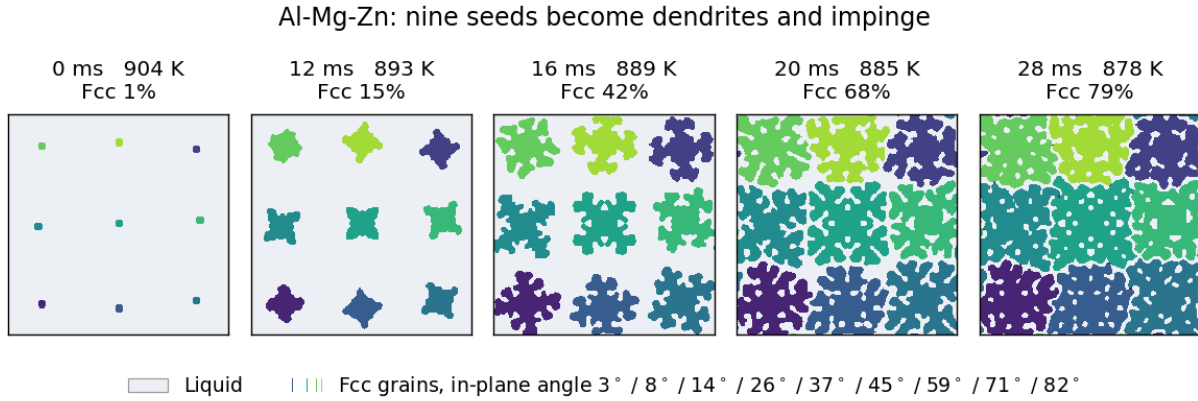


Figure 10: Nine seeds become four-fold dendrites and then impinge. Each grain is shaded by its in-plane orientation, spread over 0 to 90° , and by the second frame every grain is presenting its own arm directions. The grain boundaries are frozen where growth left them, with the last liquid held in the junctions.

Expected result. 49 100 steps and about 17 minutes, finishing at 78.7% solid. Partway through the run, at 885 K, the solid fraction is 65.9% against a Scheil value of 66%.

Compare against Scheil at the end of the run rather than partway through. Growth from discrete seeds trails the equilibrium fraction while the grains are still small, and converges on it once they have filled the domain.

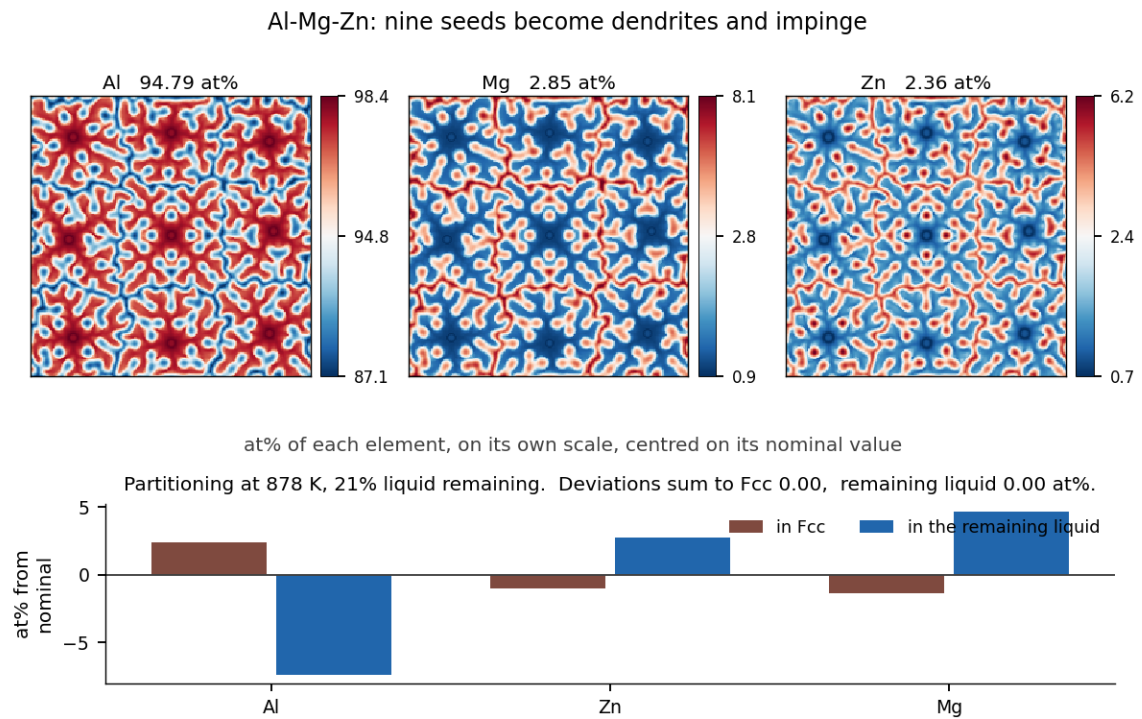


Figure 11: All three elements at the final frame. Each panel is in at%, on its own colour scale spanning that element's own measured range and centred on its nominal value, so blue is depleted and red enriched while the tick labels stay in real mole fraction. A 51 at% solvent and a 1 at% addition have nothing to say to each other on a shared axis. Mg and Zn are both rejected into the interdendritic channels and the grain-boundary junctions while Al is depleted. The bars give the absolute deviation in at%, which sums to zero over the elements.

11 Example 6: one grain growing

Examples/06_SingleNucleus. A single FCC seed in a seven-component nickel superalloy on a 200×200 grid. This is the simplest case in the set and the one to run first when verifying an installation.

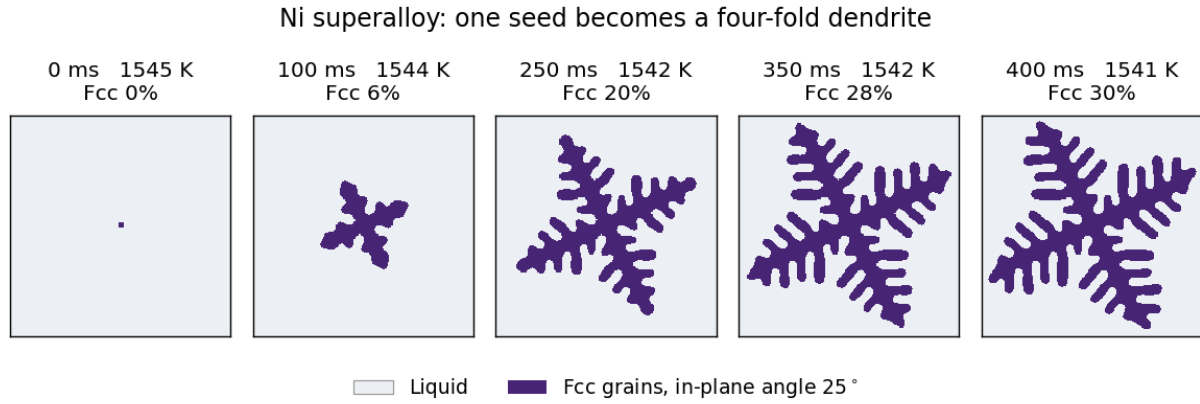


Figure 12: The seed develops into a four-fold dendrite with secondary arms. The arms follow the grain's own 25° in-plane orientation rather than the grid axes, which is what writing the angle first achieves (§4.1).

Expected result. 317 400 steps and about 166 minutes on eight threads, finishing at 30.0% solid, with a driving-force ratio whose median is 0.41.

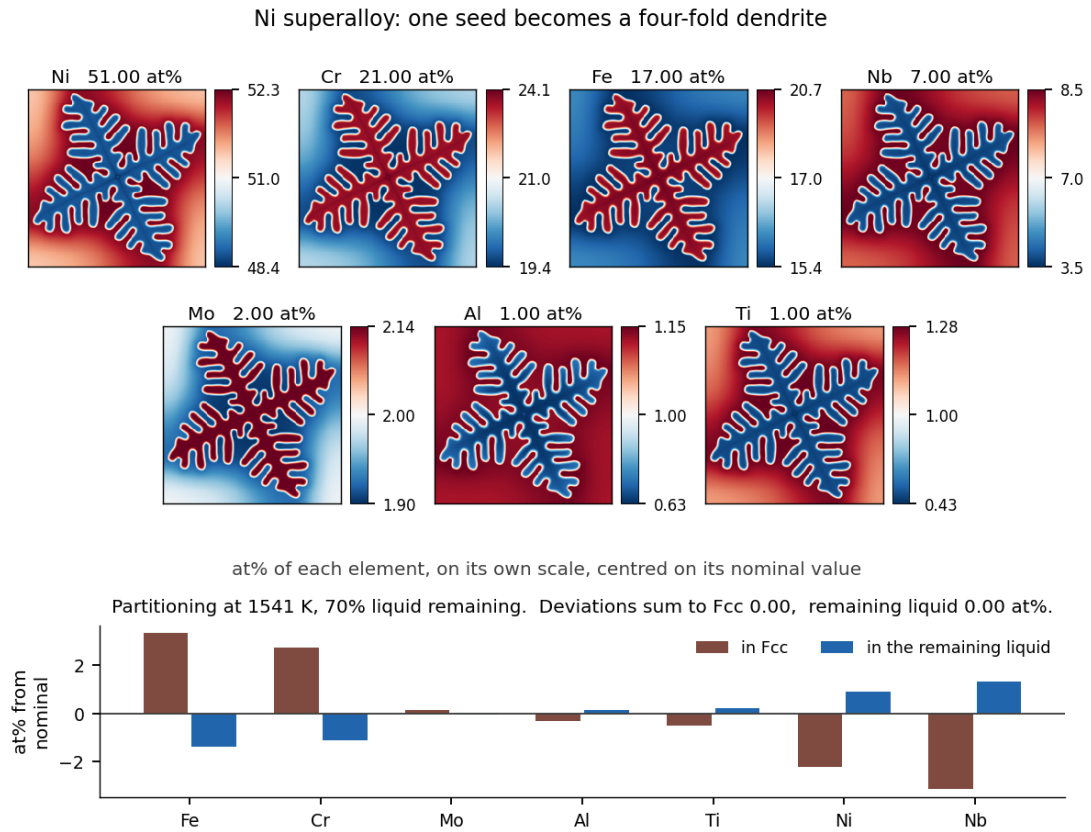


Figure 13: All seven elements at the final frame. Each panel is in at%, on its own colour scale spanning that element's own measured range and centred on its nominal value, so blue is depleted and red enriched while the tick labels stay in real mole fraction. A 51 at% solvent and a 1 at% addition have nothing to say to each other on a shared axis. Ti, Nb and Al are rejected from the dendrite into the liquid; Fe, Cr and Mo do the opposite; Ni moves from 48.4 to 52.3 at%, which is a small fraction of itself but more absolute mass than Ti or Al shift. The deviations sum to zero over the elements, as they must. Compare the panel count with Figure 9: seven components are no more work to set up than three, and each partitions in whichever direction the database dictates, with no partition coefficient supplied anywhere in the input.

12 Example 7: columnar growth with a second phase

Examples/07_AdditiveTwoPhase. The additive-manufacturing configuration and the principal application of the plugin: six FCC seeds at the cold base of a 100×300 domain, a 2.5×10^5 K/m gradient, and the seven-component superalloy with Laves_C14 entered alongside FCC. The base is cooled from 1535 to 1400 K at 800 K/s.

Each seed carries its own in-plane orientation, spread over 0 to 90° , so the seeds compete for the columnar front and the better aligned ones overgrow their neighbours. Primary FCC rejects Nb into the liquid ahead of it, and once the interdendritic liquid is both enriched enough and cold enough, Laves_C14 nucleates in it. This is the terminal solidification sequence responsible for Laves formation in additively manufactured 718.

For this composition a Scheil calculation puts the FCC liquidus at 1537.9 K and the first Laves_C14 at 1504.7 K. That 33 K gap is what lets interdendritic liquid survive down to the Laves onset, and NUC_Tmax for the Laves/liquid pair is set at 1510 K accordingly.

Laves_C14 is declared a line compound (§3), so that it holds its stoichiometric composition once it appears:

```
<Parameter name="CX_Stoichiometric_LAVES_C14" value="1" />
```

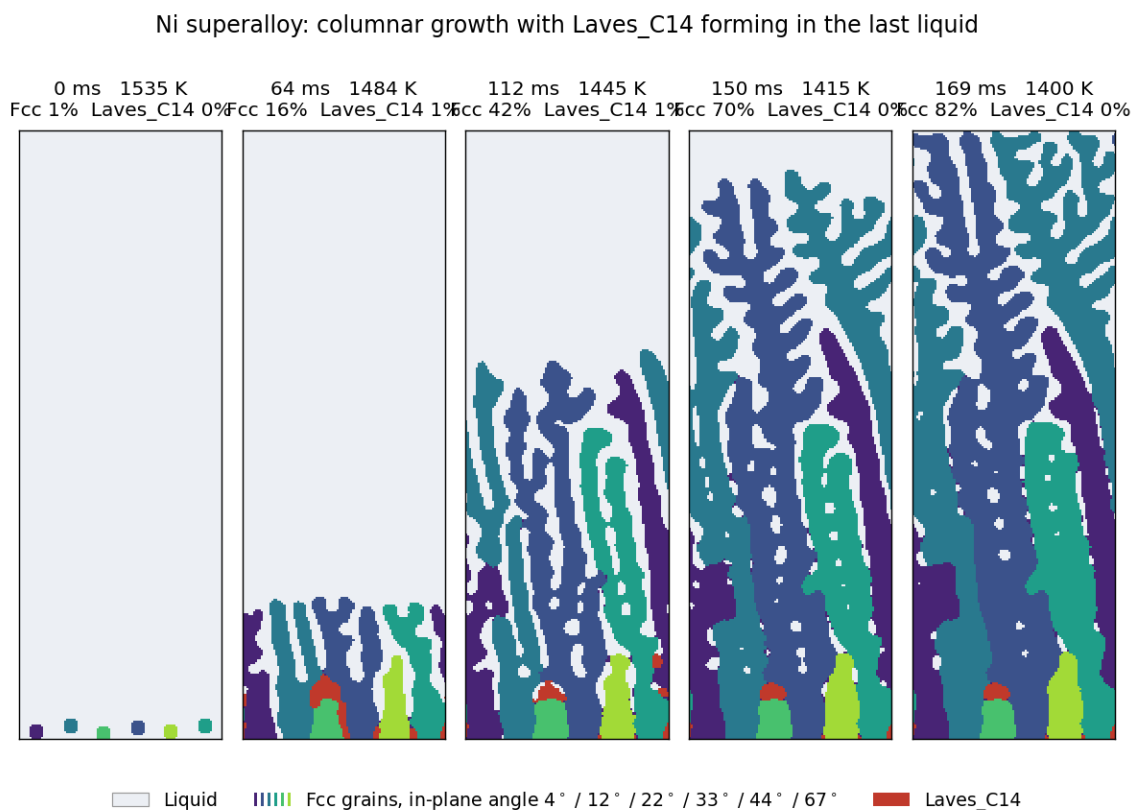


Figure 14: Each seed is shaded by its in-plane orientation, so the columnar competition is visible directly: the grains whose arms lie closest to the gradient widen at the expense of their neighbours, and the boundaries between them are frozen where growth left them. Laves_C14 appears in red, in the interdendritic liquid near the cold base.

Expected result. 501 600 steps and about 292 minutes on eight threads, finishing at FCC 81.8%, Laves_C14 0.43% and liquid 17.8%. Laves nucleation begins at 1494 K, about 30% of the way through the run, once the interdendritic liquid has enriched sufficiently. The phase pair receives one attempt at its site density, so the Laves fraction rises to 2.7% in a burst at onset and then

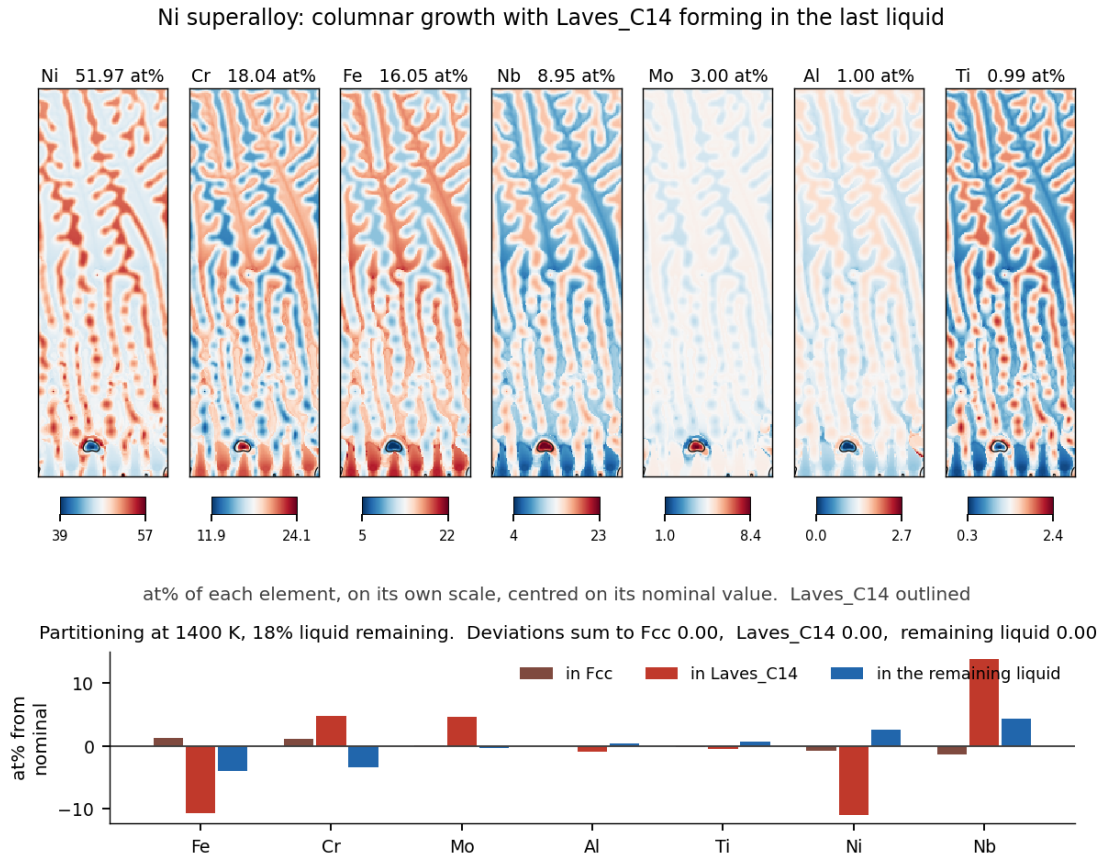


Figure 15: All seven elements at the final frame, with the Laves_C14 regions outlined in black. Each panel is in at%, on its own colour scale spanning that element's own measured range and centred on its nominal value, so blue is depleted and red enriched while the tick labels stay in real mole fraction. A 51 at% solvent and a 1 at% addition have nothing to say to each other on a shared axis. Every element shows the interdendritic segregation channels. Nb and Ti are strongly rejected into them, which is where Laves forms; Fe and Cr go the other way. The bars give the absolute deviation inside each phase: Laves is a Nb and Mo rich intermetallic correspondingly poor in Fe and Ni, and the deviations sum to zero in all three phases. No partition coefficient is supplied anywhere in the input.

settles as the surviving pockets equilibrate with the liquid around them; read the fraction at the end of the schedule rather than at the peak.

Declare an intermetallic second phase a line compound with `CX_Stoichiometric_<PHASE>` (§3). Measured on this case it holds the Laves composition to within a few tenths of an at% on every element, against roughly three times that spread when the composition is solved for. What it fixes is the composition; the volume fraction still follows the liquid the phase is growing from.

Part III: Designing your own case

13 A working recipe

1. **Run a Scheil calculation first.** It takes seconds and gives the liquidus, the freezing range, and which phases appear in what order. Everything below depends on those numbers.
2. **Start from the closest example folder** and change one thing at a time.
3. **Set the start temperature just below the liquidus.** Too far below and the domain freezes before structure develops; above it and nothing grows.
4. **Seed with SPHERE**, radius 3 to 4 cells, and write in-plane orientations explicitly (§4.1).
5. **Run briefly and read the console** before committing to a long run. A climbing solid fraction within a few minutes indicates a sound setup.
6. **Compare against Scheil** at the end. Phase fractions and segregation levels should be in the same range.

14 Sizing the run

Cost scales as grid points multiplied by time steps, and the step is set for you by stability.

- Halving the cell size in 2D quadruples the cell count and also shortens the stable step. Do not refine without reason.
- Cost is dominated by CALPHAD calls rather than by the phase-field arithmetic, which is why the extrapolation mode matters (§15).
- Measured runtimes for the shipped examples are in Table 16.

Set the duration from a growth velocity

Set the duration from the growth velocity, so that the schedule covers the microstructure you want to see. A dendrite tip moves at of order 0.1 to 1 mm/s in these conditions, so crossing a 200 to 300 μm domain takes of order 0.1 to 0.2 s of *simulated* time. Estimate that first, then choose the cooling rate that fits it.

The N/M denominator in the log, as in `Time step : 5100/280000`, is `duration/RTC_dt`. The adaptive stepper runs well above `RTC_dt`, so the run completes in `duration` divided by the effective step, several times fewer steps than that denominator suggests. Use the step rates below to estimate wall clock.

Effective step rates on eight threads, for scaling your own case:

Case	Cells	dx [μm]	Comp.	steps/s	effective dt [s]
01 PeritecticFeC	151×151	2	2	69	1.2e-05
02 DirectionalAlCu	60×200	1	2	293	4.4e-07
03 EutecticAlCu	160×120	1	2	19	4.2e-07
04 TernaryFeCrNi	301×301	2	3	26	5.0e-06
05 EquiaxedAlMgZn	200×200	1	3	48	5.7e-07
06 SingleNucleus	200×200	1	7	32	1.3e-06
07 AdditiveTwoPhase	100×300	1	7	29	3.4e-07

The step follows the cell size. The two cases at $2\text{ }\mu\text{m}$ run near $5 \times 10^{-6}\text{ s}$ where every $1\text{ }\mu\text{m}$ case sits an order of magnitude below that, which is why refining the grid costs more than the cell count alone suggests. Within one cell size the step still varies by about a factor of three, falling as a case carries more interface: Example 6 grows a single dendrite and takes the largest step of the $1\text{ }\mu\text{m}$ set, while Example 3 ends at 97% solid and takes one of the smallest.

Interrupting and resuming a run

A run of several hours is worth protecting against a reboot or a full disk. The plugin can write a checkpoint at intervals and pick the run up again from the last one.

<code>OP_CheckpointEvery</code>	0	Write a checkpoint every N output profiles. 0 disables checkpointing.
<code>OP_CheckpointKeep</code>	2	How many checkpoints to keep on disk. Older ones and their field data are deleted.
<code>OP_Restart</code>	0	Set to 1 to resume from a checkpoint instead of seeding from the <code>.pfin</code> .
<code>OP_RestartStep</code>	-1	Which checkpoint to resume from. -1 takes the newest.

A checkpoint holds the phase fields, the grain list, the composition and the nucleation state. The temperature field is not among them: it is rebuilt from the thermal history at every step, so the checkpoint only has to record the time. A checkpoint is still not small, about 4 MB for a 151×151 two-component case, growing with cell count and with the number of components. Keeping two is usually enough, and one is enough if the disk is tight.

To use it, set `OP_CheckpointEvery` before the first run. To resume, set `OP_Restart` to 1 and run the same batch file again. The resumed run restores the simulated time, the adaptive time step and the list of output profiles still to be written, so it follows the trajectory the run would have taken had it never stopped, and profile numbering continues without a gap.

A resumed run writes into a new output directory, because Pandat appends a suffix when the previous one exists: the second run of `FeC_Peritectic.pbf` writes to `output/FeC_Peritectic_1/`. The plugin finds the earlier checkpoints on its own, so nothing needs to be moved, but the output profiles of one continued run end up split across two directories. They are numbered continuously, so listing both directories in order gives the complete sequence.

A checkpoint records the grid, the cell size, the phase-field resolution and the phase and component lists, and a resume is refused if any of them has changed. Those are the properties the stored field arrays are shaped by: editing the domain and then resuming would read the old fields into a differently shaped grid and run on to a plausible but wrong answer. The refusal names what differs:

```
Error: checkpoint does not match this input: grid size
(checkpoint 1x151x151, input 1x121x151). Refusing to restart.
```

Read that line rather than the banner at the end of the log: a refused resume still finishes with **Simulation is done with Success**.

The thermal history is deliberately *not* on that list, so a resume may change it. This is what lets a checkpoint be used as a warm start: cool a microstructure to some temperature, then continue it under a longer schedule, a different cooling rate, or a hold, without paying for the first part again. Nothing in the stored fields depends on the schedule. The pending output profiles are rebuilt from the new input rather than restored from the checkpoint, so the profile times belong to the schedule you are now running, and the run says so:

```
Restart: continuing under CHANGED thermal-history duration
(checkpoint 0.0075 s, input 0.024 s). The output profile schedule
is rebuilt from this input.
```

The interface properties, nucleation settings, diffusion model and numerics are not recorded either, so those may be changed on a resume as well. Note that the phase-field resolution may *not*: Dual stores a second set of fields that Single does not, so a Dual checkpoint cannot be read into a Single run or the reverse.

15 Extrapolation mode

Both TP_ExtrapolationMode and DP_ExtrapolationMode should normally be 2 (GLOBAL).

GLOBAL keeps one linearised tie line per phase pair for the whole domain and recomputes it when the state drifts. LOCAL keeps one per grid point. LOCAL sounds more accurate but is roughly four times more expensive, because it issues a CALPHAD query per grid point rather than one per phase pair per refresh. In a thermal gradient it can also produce a per-cell speckle in the composition field, since each cell re-anchors its tie line independently as the front passes and solid-state diffusion is far too slow to smooth the result afterwards.

A domain-wide tie line must be evaluated at some temperature, and it has to be the temperature at which the transformation is occurring, that is at the interface, rather than the domain average. In a gradient the domain average can lie entirely outside the two-phase field, in which case the tie line does not exist. The plugin anchors on the interface-average temperature for each phase pair for this reason. In an isothermal case that equals the domain average.

GLOBAL is the setting used by all seven examples and is the right starting point for a new case. LOCAL becomes worth evaluating for a strongly segregating alloy where the composition varies widely across the domain, since it anchors each cell's tie line on that cell's own composition.

16 What to expect

All seven rows below were measured in one sequential batch on the same eight-core desktop, using the shipped settings. 01_PeritecticFeC_Large is not among them: at 501×501 it costs roughly nine times Example 1 and is meant as a resolution check rather than a routine run.

Example	Grid	Comp.	Wall clock	Steps
01 PeritecticFeC	151×151	2	101 min	417 900
02 DirectionalAlCu	60×200	2	64 min	1 125 100
03 EutecticAlCu	160×120	2	16 min	17 800
04 TernaryFeCrNi	301×301	3	253 min	402 100
05 EquiaxedAlMgZn	200×200	3	17 min	49 100
06 SingleNucleus	200×200	7	166 min	317 400
07 AdditiveTwoPhase	100×300	7	292 min	501 600

Cost scales with grid points, with the number of components, since each component adds to the CALPHAD work at every interface cell, and with the phase-field resolution. Component count alone does not rank the cases: Example 4 carries only three components but a 301×301 grid, and costs more than the seven-component Example 6 on its 200×200 grid. Only Examples 3 and 5 finish in under twenty minutes.