

6 Modeling and simulation of phase-transitions in multicomponent aluminum alloy casting

Andreas ten Cate¹ Bernard J. Geurts^{2*} Michael Muskulus³
Daniel Köster² Adrian Muntean⁴ Joost van Opheusden⁵
Alex Peschansky⁶ Bert Vreman⁷ Paul Zegeling⁸

Abstract

The casting process of aluminum products involves the spatial distribution of alloying elements. It is essential that these elements are uniformly distributed in order to guarantee reliable and consistent products. This requires a good understanding of the main physical mechanisms that affect the solidification, in particular the thermodynamic description and its coupling to the transport processes of heat and mass that take place. The continuum modeling is reviewed and methods for handling the thermodynamics component of multi-element alloys are proposed. Savings in data-storage and computing costs on the order of 100 or more appear possible, when a combination of data-reduction and data-representation methods is used. To test the new approach a simplified model was proposed and shown to qualitatively capture the evolving solidification front.

Keywords: aluminum casting, alloys, phase-transition, mushy zone, moving solidification front

¹Molten Aluminum Processing, Corus RD&T, IJmuiden

²University of Twente

³Leiden University

⁴Eindhoven University of Technology

⁵Wageningen University

⁶Delft University of Technology

⁷Vreman Research, Hengelo, bert@vremanresearch.nl

⁸Utrecht University

*corresponding author, b.j.geurts@utwente.nl

6.1 Introduction

In aluminum half products such as direct-chill (DC), cast ingots (aluminum blocks of $0.5 \times 1.5 \times 6 \text{ m}^3$), and billets (aluminum poles of 0.2 – 0.5 m diameter and 6 m length) the spatial distribution of alloying elements is very important. In advanced aluminum products a considerable number of elements (typically elements such as Cu, Cr, Fe, Mg, Mn, Si, Zn) is involved at small to intermediate concentrations. These elements are very important as they determine the specific properties of the final alloy such as strength, fracture toughness, hardness, brittleness, dent resistance, surface quality et cetera. The aluminum research in industries such as Corus aims at developing new products for demanding applications such as the aerospace and automotive markets. It is the objective of this research to optimize the specific properties of the alloy for the particular applications by modifying the alloy composition in a generally narrow composition window. The consistency and homogeneity of the cast product in the solid phase is a prime aspect of casting technology. However, due to the casting process the homogeneity of the cast products may be compromised. Understanding and controlling the mechanisms that contribute to formation of spatial heterogeneity, also called macrosegregation, is therefore crucial.

In the casting process initially all elements in the mixture are in the liquid phase and spatially well-mixed. In semi-continuous casting of aluminum alloys the liquid metal is poured into a cooled mould. The molten metal is chilled by contact with the mould and application of cooling water. As the temperature decreases solidification sets in and a front between the already solidified and the still liquid part develops. It is exactly this transition band between solid and liquid, also known as the ‘mushy zone’, that plays a crucial role in the uniformity and hence the quality of the final cast product. Upon solidification the elements tend to redistribute between the solid and the liquid phases. Each element does this in its own manner, which is controlled by the thermodynamic equilibrium. Consequently, the liquid phase can become enriched and the solid phase can become depleted in elements. Local transport of the liquid phase due to shrinkage induced straining of the solid phase and due to buoyancy driven flow effects in the liquid part of the domain thus will cause redistribution of the elements on the scale of the ingot or billet cross-section. For a comprehensive overview of macrosegregation literature see [1].

This partial segregation is detrimental to the quality of the resulting cast and generally the resulting cast is beyond repair. As a consequence the resulting product is off-spec and has a reduced economic value or becomes rejected, which results in recycling of the entire cast product and obvious economic loss. These additional production costs can potentially be reduced if a more precise understanding of the origin of these cast defects can be obtained. In this paper we describe mathematical models that aim to simulate the details of solidification and transport induced segregation that take into account a large number of different species. We specifically present efficient methods for including in a computationally efficient manner the complex thermodynamics that characterize the solidification of many-species

mixtures used in modern aluminum products.

In casting technology research over the last decennia many numerical models have been developed for the prediction of the many different physical phenomena involved in the casting process (for example [2, 3, 4]). Such computational models generally predict the fluid flow in the liquid part, compute the solidification (the transition from liquid to solid) and calculate how the metal deforms when cooling down. These models often assume a constant composition throughout the domain. This assumption ignores the effect of spatial segregation, which is at the heart of current aluminum casting problems. The crucial step for simulations of industrially relevant alloys is that a large number of elements (about five or more) should be included in the simulations to achieve a proper modeling of the processes and phase-transitions. This leads to a strong increase in the simulation times. The challenge is to propose computational strategies to establish this crucial step in an efficient way.

Currently solidification models are under development that include the variation of composition during solidification (e.g. [5, 6, 7]). This requires that the relation between the local composition and temperature is computed. To a good approximation, this relationship is determined by considerations of thermodynamic equilibrium. A key element is the phase diagram, which gives the relation between phases, composition and temperature. For a binary mixture this already results in a complex parameter-space with widely different transitions in different regions. In case of a realistic multi-element mixture the complexity of the thermodynamic representation rapidly increases. Direct coupling of a thermodynamic database to a solidification simulation may impose limitations to the practical applicability.

In simulations of the casting process that include the effect of composition, the thermodynamic equilibrium needs to be determined each time step and in each grid cell. Commercial software is available to compute the thermodynamic equilibrium via a minimization of the Gibbs free energy (examples are [8], factsage[9], jmatpro[10]), but this is a computationally time consuming step. A direct coupling between the database and the casting simulation will result in infeasible simulation times. The challenge is to propose efficient coupling methods between the solidification simulation and the thermodynamic database. The question is how the solidification path in the computations can be constructed in a computationally efficient manner, considering that thermodynamic equilibrium data contains highly irregular features such as discrete transition points (e.g., an eutectic point) and large variations in the regions in which phase equilibria appear (e.g., some phases appear over a range of 5 Kelvin, others are present over several hundred Kelvin). One approach applied and presented in this work is to adopt local polynomial fits to thermodynamic data. This resulted in a significant reduction of the computational expense with full recovery of the physical properties of the casting process within the required numerical accuracy. The problem posed by CORUS to the 63rd *European Study Group Mathematics With Industry* was twofold: (1) Propose a simple PDE model for the simulation of the aluminum casting process and methods to establish an efficient coupling between the thermodynamic database and the involved PDEs.

(2) Assure that the model can simulate efficiently an industrially relevant number of alloying elements.

In this paper we review the continuum modeling in Section 6.2 and present an efficient method for simplifying the complex and computationally intensive thermodynamics that occur in Section 6.3. A one-dimensional numerical model will be adopted in Section 6.4 to illustrate the basic physical processes arising in the casting process, emphasizing the treatment of the solid-liquid mushy zone front. Finally, concluding remarks will be collected in Section 6.5.

6.2 Modeling transport and phase-transitions in multi-component aluminum casting

In this section, we present a complete model for transport and phase transitions that occur during the aluminum casting process. Our aim here is not to redo more involved mathematical models describing aluminum casting (e.g., [2, 12]), but to find a simple, yet realistic description of fluid flow and solidification of an aluminum alloy which allows to develop and test techniques for handling the multi-element thermodynamics during solidification. The formulations will result in the definition of a one-dimensional model that will be used in Section 6.4 for testing the thermodynamics evolution and to assess whether the main characteristics of the casting process can be recovered.

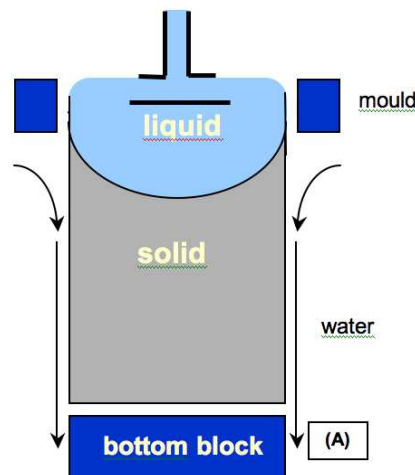


Figure 6.1: Sketch of the basic geometry in the aluminum casting process. The bottom block is continuously lowered as liquid aluminum is added on the top. Throughout water is applied for cooling the boundary of the aluminum block.

We consider a spatial domain split into a solid and a liquid region, see Fig. 6.1. The two regions are separated by a mushy zone, whose exact position has to be cal-

6.2 Modeling transport and phase-transitions in multi-component aluminum casting

culated along with the flow and temperature fields. To describe the fluid mechanics and solidification physics a number of unknowns needs to be introduced. We refer to Table 6.1 for the unknowns of the problem as well as Table 6.2 for the list of the necessary ‘parameters’. We refer to these as parameters, although strictly speaking their values are functions of the primary unknowns (e.g., the latent heat Δh is a function of the molar concentrations of various alloy elements, thermal properties of the hosting material and, of course, of the local temperature). For simplicity, we assume no pouring of liquid material into the solid domain and neither changes nor motion of the physical domain.

Notation	Dimension	Description
ϵ_l	1	local volume fraction occupied by liquid
$\epsilon_s := 1 - \epsilon_l$	1	local volume fraction occupied by solid
c_l^X	mol/m ³	molar concentration of material X in liquid
c_s^X	mol/m ³	molar concentration of material X in solid
\mathbf{v}	m/s	fluid velocity
p	kg/(ms ²)	fluid pressure in liquid and mushy region
T	K	temperature

Table 6.1: Unknowns of the model.

Notation	Dimension	Description
m^X	kg/mol	molar mass of species X
ν, ζ	m ² /s	kinematic standard/bulk viscosity of liquid
\mathbf{g}	m/s ²	gravitational acceleration
K	m ²	permeability tensor in the mushy zone
κ	kg m/(K s ³)	heat conductivity
Δh	kg/(m s ²)	latent heat of phase transition
C_p	kg m ² /(K s ²)	heat capacity at constant pressure

Table 6.2: Parameters of the model.

Our model consists of conservation laws for the liquid and solid mass of all alloy elements X_1, \dots, X_N , the averaged momentum of the fluid flow, and the total internal energy. Since the formation of micro-structure (dendrites, see Fig. 6.2) creates a mushy environment with a definite porous structure of the material, the momentum equation is formally replaced by the conceptually simpler Darcy law; see, e.g., [11]. The unknown ϵ_l serves to distinguish between those parts of the domain that are currently liquid, mushy, or solid. Note that, e.g., the “liquid” region could be defined as that part of the domain with $\epsilon_l \in (0.9, 1]$.

As a first step toward the mathematical model we present the equations describing conservation of mass of each individual element X participating in the solidification process. We express the balance of mass of the liquid and solid species separately.

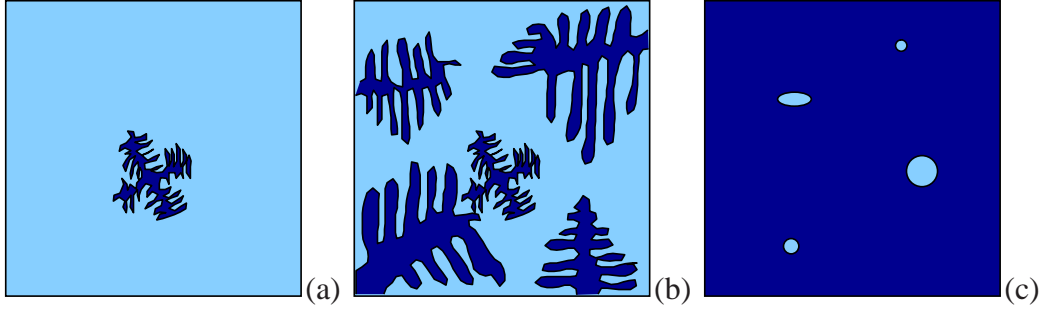


Figure 6.2: Local conditions in liquid (a), mushy (b), and solid regions (c) of the domain.

Assuming that diffusion due to concentration gradients is negligible at the characteristic flow time scale in the solidification process, the conservation of mass of species X in the liquid state is

$$\frac{\partial \epsilon_l c_l^X}{\partial t} + \nabla \cdot (\mathbf{v} \epsilon_l c_l^X) = 0. \quad (6.1)$$

This evolution equation assures that the integral of $\epsilon_l c_l^X$ over any volume Ω in the flow domain can change only due to fluxes through the boundary of Ω . Similarly, the conservation of mass of species X in the solid state reads

$$\frac{\partial \epsilon_s c_s^X}{\partial t} + \nabla \cdot (\mathbf{v} \epsilon_s c_s^X) = 0. \quad (6.2)$$

To characterize the flow in this scenario, we distinguish between liquid, mushy and solid zones. The balance equation for the linear momentum, which applies in the liquid zone, is given by

$$\frac{\partial m_i}{\partial t} + \nabla \cdot (m_i \mathbf{v}) = g_i \rho + \frac{\partial \sigma_{ij}}{\partial x_j}, \quad (6.3)$$

for $i, j = 1, \dots, 3$. The liquid is considered incompressible with ρ is constant. Throughout, we adopt the Einstein convention on summation over repeated indices. Here, we have used the total momentum density m_i in the x_i direction

$$m_i = v_i \rho, \\ \rho = \rho_l + \rho_s = \epsilon_l c_l^{X_k} m^{X_k} + \epsilon_s c_s^{X_k} m^{X_k},$$

with $k = 1, \dots, N$. The two terms on the right-hand side of (6.3) represent gravity and viscous drag, modeled as Newtonian fluid for simplicity:

$$\sigma_{ij} = -p \delta_{ij} + \nu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \frac{2}{3} \eta \frac{\partial v_l}{\partial x_l} \delta_{ij} \right) + \zeta \frac{\partial v_l}{\partial x_l} \delta_{ij}.$$

6.2 Modeling transport and phase-transitions in multi-component aluminum casting

In the mushy zone, solidified alloy dendrites form a dense porous medium. Inspired by [12], we use Darcy's expression to relate velocity and pressure:

$$v_i = -\frac{1}{\epsilon_l \nu \rho_l} K_{ij} \frac{\partial p}{\partial x_j}.$$

This is only an Ansatz. A rigorous derivation via homogenization-type arguments is still needed (see, e.g., [13]). Finally, the solid zone (the aluminum) is described as a state of rest:

$$\mathbf{v} = 0.$$

In the liquid region ($\epsilon_l \approx 1$) we define the velocity through solving the momentum equation, in the solid zone ($\epsilon_s \approx 1$) we use the state of rest and in the remaining mushy zone the Darcy formulation is chosen. Temperature and energy dynamics is sketched next. We express the total internal energy density as

$$e^* = C_p T + \frac{1}{2} v_j^2 + \text{const.}$$

The conservation of total internal energy is given by

$$\frac{\partial}{\partial t}(\rho e^*) + \nabla \cdot (\rho e^* \mathbf{v}) = Q + \nabla \cdot (p \mathbf{v}), \quad (6.4)$$

with the heat source rate expressed as

$$Q = \nabla \cdot (k \nabla T) + \Delta h \frac{\partial \epsilon_l}{\partial t}.$$

Heat is thus added to the system by the liquid-solid phase transitions taking place in the mushy region, expressed by the latent heat Δh , as well as by heat conduction with coefficient k (Fourier's law). Viscous heating due to friction is neglected.

Besides the calculation of the model parameters (which typically depend on the unknowns of the problem), we need to close our model by additional constitutive relations. Here we suggest two such relationships. In principle, (local) thermodynamic properties could be used to determine the pressure as a function of temperature and species concentrations:

$$p = F_1(T, c_l^{X_1}, \dots, c_l^{X_k}). \quad (6.5)$$

Alternatively, we could use information from thermodynamic phase diagrams to calculate the liquid fraction

$$\epsilon_l = F_2(T, c_l^{X_1}, \dots, c_l^{X_k}, p). \quad (6.6)$$

The evaluation of (6.5) (or (6.6)) can be based on information available from thermodynamic databases. Only one of these two expressions needs to be selected -

which particular one is chosen may depend on the application. Furthermore, to solve the pressure a more involved analysis is required in which (6.5,6.6) do not play an important role. The closure poses the problem of efficiently accessing the thermodynamic information, especially in the case when many species are present. Alternatively, this may be obtained via variational principles (by minimizing the corresponding Gibbs functional), which poses the problem of simultaneously solving a PDE system and finding local minimizers to a non-linear non-convex functional. Both these approaches increase the computational effort. Considerable care in the reduction of the mathematical model and algorithm development is needed to achieve realistic costs of simulation. We present an approach based on local polynomial fitting in Section 6.3 and estimate theoretically the computational saving compared to a full gridding of the thermodynamic state-space.

In Section 6.4, some example calculations are given for a simplified one-dimensional model for a slow solidification process of a single species. This model can be readily appreciated as a special case of the general formulation given above. The purpose of this reduced model is to isolate the main characteristics of the solidification process and to test the efficiency of the evaluation with which thermodynamic properties such as Δh are being processed. The 1D model that is proposed can be written as

$$\begin{aligned}\frac{\partial T}{\partial t} - \frac{\partial^2 T}{\partial x^2} - L \frac{\partial \epsilon_l}{\partial t} &= 0, \\ \frac{\partial \epsilon_l}{\partial t} - M \frac{\partial^2 \epsilon_l}{\partial x^2} &= 0,\end{aligned}\tag{6.7}$$

where L is a coefficient related to the latent heat used to produce the phase transitions, while M is a constant effective diffusivity of the liquid. The rationale behind this model is that we neglect all fluid flow, thus $\mathbf{v} = 0$, i.e., both in the liquid and in the solid. Correspondingly, only diffusive transport for ϵ_l remains in this very crude model. In the absence of gravity and at constant pressure, the momentum equations are trivially fulfilled. It remains to discuss the energy conservation equation (6.4). Under the additional assumption that the parameters ρ , k , and C_p are constant, equation (6.4) yields $e^* = C_p T$ in which temperature is governed by

$$C_p \rho \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + \Delta h \frac{\partial \epsilon_l}{\partial t}.$$

If in addition $k = C_p \rho$, then the last equation reduces to

$$\frac{\partial T}{\partial t} - \frac{\partial^2 T}{\partial x^2} - L \frac{\partial \epsilon_l}{\partial t} = 0,$$

where $L = \Delta h / (C_p \rho)$ and we dropped the subscript l . The second equation of the simplified model (6.7) is then obtained by assuming that the liquid fraction is proportional to the temperature T , within some reasonable range of T . In this case,

the second equation of (6.7) is recovered. This model has an effect of latent heat being released, while the solidification front progresses pushed by diffusion. This is a particularly appealing model for numerical analysis and illustration of the main physics of solidification. We return to this in Section 6.4.

6.3 Thermodynamic representations and data reduction

Any CFD simulation of solidification of an alloy requires thermodynamic input in each fluid cell and at each time-step. This input may be the latent heat, the heat capacity, and the local predictions of phase concentrations and compositions that occur for a given temperature under certain thermodynamic assumptions. Minimization of the Gibbs functional ‘on the fly’, i.e., everywhere and anytime, is too time-consuming in this context. One way to circumvent this problem is to employ a thermodynamic database, which is also called a mapping file in the literature [18]. This database can be pre-computed by performing Gibbs minimizations for a large number of specific combinations of temperature and phase concentrations. The database is a discrete numerical representation of the information contained in the physical phase diagram. In general, the local temperature and phase concentrations in a fluid cell in the CFD simulation are not precisely equal to the available discrete values of the entries in the database. Interpolation is thus necessary, which is much less time-consuming than the Gibbs minimization computation itself. In this section we will pursue this method and incorporate polynomial fitting to reduce the storage requirements for the database. Theoretical estimates of the efficiency are also provided.

6.3.1 Polynomial fit

The problem with precomputed databases is that they easily become much larger than the present memory of computers. Consider for example an alloy solidified from the four materials Al, Cu, Fe and Mg. Then a thermodynamic quantity, such as the heat-capacity C_p , is dependent on temperature T and on three independent species concentrations c_1 , c_2 and c_3 , while the remaining one c_4 is given by $c_4 := 1 - c_1 - c_2 - c_3$ in a non-dimensionalized situation. The function C_p then depends on 4 variables. If we would use a uniform grid for each of the four arguments, covered each by 600 points for sake of argument, we would need a memory of $2 \times 4 \times 600^4 = 4\text{TB}$ to store two thermodynamic quantities with single precision. Such a database approach has been considered in [18], where it was noted that calculations of up to four elements can thus be realized, but calculations involving five or more elements seem to be beyond reach at present. The aim of the present section is to investigate whether it is possible to reduce the size of the database strongly, without unduly affecting the accuracy of the thermodynamic input delivered to the CFD-simulation.

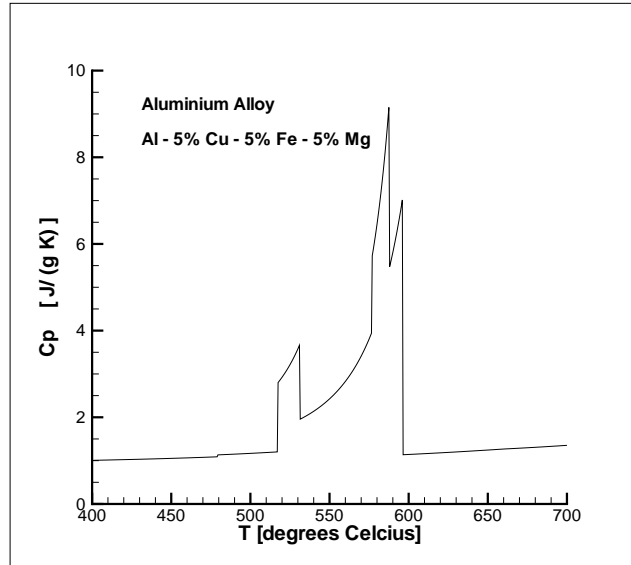


Figure 6.3: Example of the dependency of the heat capacity on temperature for fixed composition at 5% of Cu, Fe and Mg in an Al alloy.

Since the thermodynamic quantities in phase diagrams typically display strong jumps, a large number of grid points is necessary in each direction if a uniform grid is used for the entries of the database. Unstructured nonuniform meshing of the table automatically adapted to the shape of the phase diagram is expected to reduce the size of the grid needed to represent the table. That such a strategy leads to much smaller databases is illustrated in the remainder of this section by considering a simple example of homogeneous solidification.

The temperature in a process of homogeneous solidification of the alloy Al-Cu-Fe-Mg can be described by the following equation:

$$C_P(T, c_1, c_2, c_3) \frac{dT}{dt} = -Q < 0, \quad (6.8)$$

where T is the temperature, assumed to be spatially independent in this case, and Q the heat extracted from the system. The heat-capacity C_P is the so-called effective heat-capacity, in which the latent heat is included. Three concentrations c_1 , c_2 and c_3 are needed to describe the concentration distributions, i.e., the relative amount of molecules of Al, Cu, Fe and Mg. For the present example we assume that the three concentrations of Cu, Fe and Mg are equal, $c_1 = c_2 = c_3 = 5\%$ (mass concentrations). Since the solidification process considered in the present example is homogeneous, the concentrations are constant in space, but also constant in time, because of mass conservation. Therefore, to solve (6.8) the thermodynamic database (the phase diagram) can essentially be reduced to the representation of C_P as a function of temperature.

We computed the temperature dependence of C_P under these concentration conditions for the Al-Cu-Fe-Mg system by minimizing the Gibbs free energy. The

6.3 Thermodynamic representations and data reduction

result is shown in Fig. 6.3, clearly illustrating a central feature of the phase diagrams: strong jumps appear, but in between these ‘discontinuities’, the function is relatively smooth. Fig. 6.3 has been obtained with a uniform temperature grid containing 600 points. Obviously, C_P can be accurately captured with much less points if one would only store the locations at which the ‘discontinuities’ appear, while the smooth parts in between would be approximated by suitable polynomials. This basic observation will be worked out in more detail next, to show the principle.

To reduce the thermodynamic database storage we define a jump location by a threshold of 0.5 J/(gK) , fixed a priori for simplicity. We consider two options for the smooth pieces between jumps: first-order (straight lines) and second-order Lagrange polynomials (parabola). The coefficients of the polynomials can simply be computed from the values at and between two jumps. The two end-points of a smooth region are collocation points for the first-order but also for the second-order polynomial. For the second-order polynomial a third collocation point needs to be added. For this we take the point half way in the interval under consideration. Thus instead of 600 floating point numbers (uniform grid) we need to store much less floating point numbers to represent the behavior in Fig. 6.3 with piecewise continuous polynomials. In particular, we require only 17 numbers in case of linear polynomials, and 23 in case of second-order polynomials.

To assess the quality of the reduced data representations we solve (8) for the three different numerical representations of C_P . We compare (a) the fine-grid representation consisting on 600 uniformly distributed points, (b) a linear polynomial and (c) a second-order polynomial fit. In each case a four-stage Runge-Kutta method with a sufficiently small time-step is used to integrate the equation. The right-hand side is assumed to be constant and equal to $Q = -1 \text{ J/(gs)}$. The results of the computations are shown in Fig. 6.4. The second order polynomial fit provides a very accurate approximation of the fully resolved case – there is no discernible difference between the curves based on method (a) and (b). It is concluded that in this example the size of the database can be reduced by a factor of around 30 without significant loss of accuracy (in this example a reduction from 600 data points to 17 or 23 in case linear or quadratic interpolation is used).

The homogeneous case above is very simple; C_P is reduced to a function of temperature alone because the concentrations remained constant. In practical CFD-calculations the concentrations change. Nevertheless, the above method can in principle also be applied to more practical cases: the temperature dimension can be treated as in the example above, using piecewise discontinuous polynomials, while the concentration dimensions are still treated with linear interpolation on uniform grids. If we would use a structured nonuniform meshing of the concentrations (clustering in the most important regions) for the Al-Cu-Fe-Mg alloy we might be able to obtain a reduction of a factor of 3 in each concentration reduction. Thus the total storage reduction would be a factor $30 \times 3^3 \approx 800$, such that the original database of 4TB would reduce to 5GB and thus fit well into the memory of any modern personal computer.

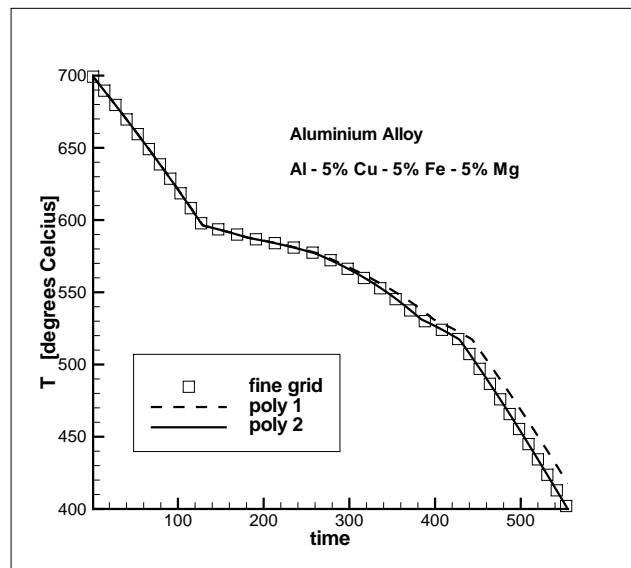


Figure 6.4: Simulation results for 1D solidification with constant composition.

The above basic approach to reducing the storage required for the thermodynamics database can be extended to a more complete computational data-representation scheme for demanding casting problems. In the next sections, we describe the main elements of this methodology.

6.3.2 Alternative approaches

In the following sections we consider an alternative approach based on a non-uniform mesh representation and discuss its merits and disadvantages. The development of this method has been guided by the following principles:

1. The thermodynamic quantities of interest fall into two different categories:
 - a) Quantities that are smooth and change slowly with respect to changes in composition and temperature, for example: *enthalpies* and *phase composition* (what elements are present in a certain phase).
 - b) Quantities that change abruptly and discontinuously, for example: *phase information* (what phases are present and in what relative amounts) and effective *heat capacities*.
2. Some regions of the phase diagram are more important and should be represented with higher accuracy than other regions of less interest. This is partly due to the occurrence of phase changes, but also since some of the elements are only present in rather small concentrations in the system, such that large parts of the phase diagram are (probably) never needed in a simulation.

6.3 Thermodynamic representations and data reduction

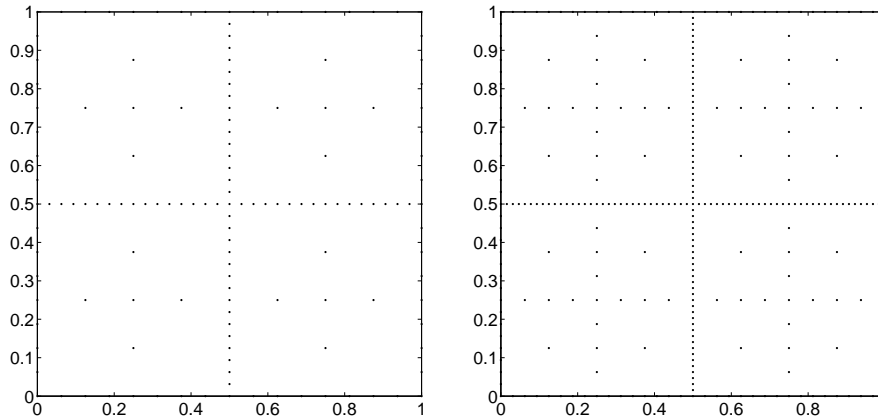


Figure 6.5: Example hierarchical sparse grids (Curtis-Clenshaw type) containing five levels of successive approximations (left panel) and six levels (right panel), respectively.

3. The evaluation of phase diagram data needs to be very efficient, such that complicated interpolation schemes are out of the question.

With regards to the last point, the optimal solution would be a database approach and multi-linear interpolation of the values of query points, which is very fast to implement, running in $O(n)$ time when the database is represented on a regular grid with n grid points per dimension. As has been noted in the previous section, however, such an approach is ultimately infeasible due to the large number of grid cells needed to represent the phase diagram accurately, the space complexity being of order $O(n^d)$, where d is the dimension.

It should be noted, though, that all thermodynamic quantities of interest, for example the heat capacities, can be derived from two ingredients alone: smoothly varying enthalpies and *phase information*. Were this phase information discrete, we could proceed with two different strategies:

1. Model the continuous enthalpies by some simple interpolation scheme.
2. Model the discontinuous phase boundaries separately.

The first point can be realized, for example, by a hierarchical representation on sparse grids [16], for which an efficient implementation in MATLAB is available [21]. The mean of a quantity of interest over the phase diagram is represented as a single number in the first node of the hierarchy, and more localized changes are represented by a number of sparsely distributed points at lower levels of the hierarchy. Fig. 6.5 shows an example of the sparse grids typically used at different levels of detail.

If the thermodynamic data were represented in *thermodynamic variables*⁹ then the phase diagram would consist of a so-called *cell structure* [17], such that each cell represents one unique phase. Unfortunately, it is nontrivial to transform element concentrations into thermodynamic variables and vice versa. But when concentrations are used as variables, then *mixtures* of phases occur, where two or more phases are coexisting in the system in varying amounts. For binary systems, these phase mixtures can be described by analytical formulae, for example the *lever rule* [17], which is simply linear interpolation of two phases with respect to concentration, but already for ternary systems no such simple rule is available. This means that in certain regions of the phase diagram unfortunately not only the phase boundaries have to be represented, but also the complete phase information. On the other hand, this information is usually also smooth inside a given region in the phase diagram.

To conclude: In principle the thermodynamic information needed in actual simulations of solidification processes concerns either (1) smoothly varying data, or (2) discrete information about the phase boundaries. This distinction was already apparent in the example discussed in Subsection 6.3.1.

6.3.3 Tracing the phase boundaries

From the above it is clear that the biggest problem in the efficient calculation of thermodynamic properties is the accurate representation of the boundaries of the phase diagram. These boundaries form a $n - 1$ dimensional hyper-surface if the system is n dimensional, i.e., is described by the relative concentrations of n distinct elements and temperature. Note that concentrations have to sum up to one, so in fact there are only $n - 1$ independent concentration variables to consider. In a binary system, the phase boundaries are one-dimensional, for example.

In general, one can distinguish two basic approaches for the representation of hyper-surfaces such as occur in the thermodynamic closure describing the phase transitions. An *explicit* surface is represented by some parametric surface, given by a multidimensional spline, for example, or a representation as an unstructured grid by simplices. In two dimensions the latter is often realized by a Delaunay triangulation [15]. On the other hand, an *implicit* surface is represented by a number of smooth, local basis functions and the surface is defined as an iso-contour of a scalar function. This method is attractive, since it allows to trace surfaces elegantly and accurately by level set methods [25], but unfortunately the computational costs can be very high.

Since we need phase boundary information for the approach outlined in the following section, we describe here a simple method to trace the boundaries. The information obtained consists of a number of points lying very close to the actual phase boundaries (within a user-specified numerical tolerance) and can be used as input

⁹Thermodynamic variables form a complete set that uniquely describes a thermodynamical system. For the solidification process, these are usually taken to be the temperature, pressure and chemical potentials associated to the involved species

6.3 Thermodynamic representations and data reduction

for the more advanced level set methods mentioned. The approach is demonstrated on a binary system consisting of the two elements Pb and Sn. Thermodynamic data for this system is available through calls to the CHEMAPP library¹⁰ which is the calculational back end of the commercial CHEMSAGE software [19, 20]. The independent variables are temperature T and composition x . The latter measures the relative amount of Pb, such that $0 \leq x \leq 1$. The region of the phase diagram we considered was a temperature range of $320 \leq T \leq 620$, measured in Kelvin.

The boundaries of the phase diagram have been traced by a bracketing method. For simplicity, we have distributed a number of points (320) regularly along the T axis and then bracketed all points where a phase change occurs, varying x , by an iterative bisection method [24]. The algorithm stores two different concentration values $x_1 < x_2$ and evaluates the discrete phase information at both points. If a difference is found, the phase information at the middle point $x_{12} = \frac{x_1+x_2}{2}$ is evaluated. If the phase at x_{12} is the same as the one at x_1 , then x_1 gets updated to x_{12} , otherwise x_2 gets updated. If the phase at the middle point is different from both phases at x_1 and x_2 , respectively, both subintervals are (recursively) bisected. The algorithm continues until $|x_1 - x_2| < \epsilon$; here we used $\epsilon = 10^{-4}$.

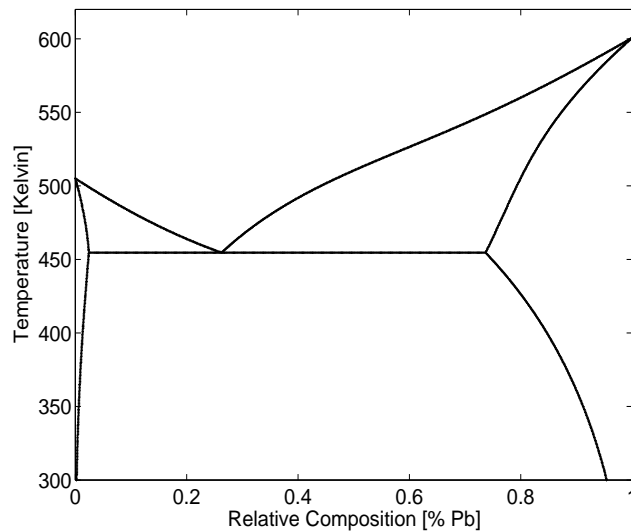


Figure 6.6: Traced phase diagram of Pb-Sn binary example system.

Complementing this “vertical” tracing, we have analogously distributed points along the x axis and bracketed all phase changes, varying T . For this horizontal tracing we have used 500 points. The resulting phase boundaries are shown in Fig. 6.6. In each of the six areas in the figure a physically different equilibrium state is found.

¹⁰A restricted version called CHEMAPP LITE is available for private, non-commercial use.

URL: <http://gттserv.1th.rwth-aachen.de/~cg/Software/ChemApp/>

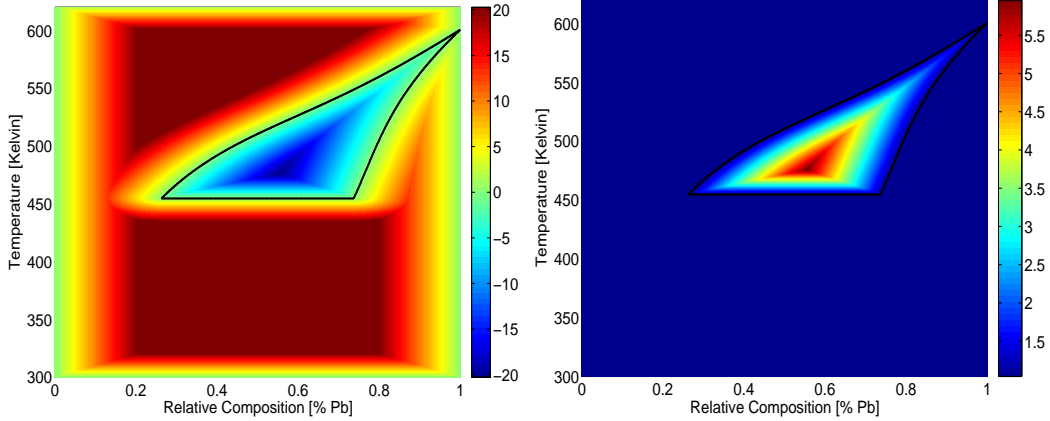


Figure 6.7: Distance (left panel) and size function (right panel) for the liquid-Pb mixture phase in the Pb-Sn binary example system.

6.3.4 Triangulation of phase regions

The next step is the triangulation of the different phase regions. These have been performed with the simple mesh generator developed in [22]. The input needed for this code is a signed distance function $d(T, x)$ that returns the distance to the nearest phase boundary, and a size function $h(T, x)$ that returns the desired edge length of the triangulation at each point, thereby allowing non-uniform adaptive meshing.

Fig. 6.7 shows the distance function for a certain phase region of the liquid-Pb mixture phase. We use the Euclidean distance

$$d(T, x) = \left((T - T^*)^2 + k (x - x^*)^2 \right)^{1/2},$$

where (T^*, x^*) denotes the point on the phase boundary closest to (T, x) and $k = 200$ was used to weigh the contribution of concentration changes with respect to temperature changes. The distance function we used is interpolated on a regular grid, where the distance to the closest phase boundary point has been approximated by the minimum of the distances to the previously traced boundary points.

From this distance function, a size function has been computed. For simplicity, we used

$$h(T, x) = 1 + 10 \exp(|d(T, x)/2d_0|),$$

where $d_0 = \min_{T,x} d(T, x)$ is the characteristic width of the phase region. Results of such an adaptive meshing are shown in Fig. 6.8.

In a practical application of this method, one needs to mesh the phase diagram separately in each region and then join the triangulations at the internal interfaces, i.e., the phase boundaries. A discussion of these issues can be found in [23]. Also, the size function should depend on the local accuracy level that is required. In fact, one can also consider a data-driven approach, where an actual simulation is

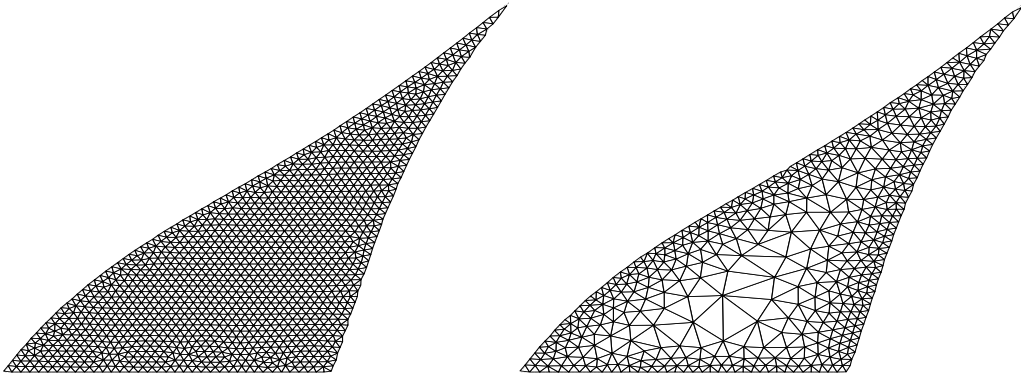


Figure 6.8: Example triangulation of the liquid-Pb mixture phase. Left panel shows results by uniform size function, right panel shows results by distance-dependent size function.

performed in which the locations in the phase diagram that are needed are recorded. From these data one can construct a density function $h(T, x)$, where regions of the phase diagram that are needed often in a simulation would be represented in more detail than regions that are needed rarely. Of course, one can also combine all these considerations into one common size function.

This method generalizes to n dimensions by replacing triangles (2-simplices) by n -simplices. Each simplex is then represented by $n + 1$ points and consists of $\binom{n+1}{2}$ edges. The storage requirements are therefore of order $O(n^2)$ in the number of simplices used. More importantly, when a CFD simulation needs to evaluate phase diagram information, first the corresponding simplex needs to be found, and then the values stored at its edges are linearly interpolated. The location of the simplex containing the query point is an example of a point location problem with a typical time complexity¹¹ of order $O(\log n)$ [15] in the number of stored simplices n , whereas the interpolation is linear.

6.3.5 Localized caching

From the above it should be clear that the problem of efficiently representing phase diagram information is quite difficult, and the familiar tradeoff between storage and time complexity is encountered. Probably the biggest savings in computing time can therefore be expected to be achieved on quite a different level. Recall that thermodynamic data is needed for each grid cell and at each time step, but (1) the local state in each cell (temperature, concentrations) usually changes slowly in between time steps, and (2) in most cases the local state changes slowly between spatially neighboring cells. An efficient implementation should therefore try to also make use of these two properties, recycling already computed thermodynamic data

¹¹In MATLAB this is implemented in the function `tsearch`, which is based on the QHULL code [14] freely available from <http://www.qhull.org/>

as much as possible and only recomputing these data if absolutely necessary.

The basic idea is to store a pointer in each grid cell that points to the thermodynamic data used in the last time step. If the state of the cell does not change in a certain range, the thermodynamic data is reused without computation (in a more advanced implementation, linear changes could be taken into account and interpolated locally at each time step). Of course, the tolerance used would usually depend on the location in the phase diagram: close to a phase boundary the thermodynamic data of each cell should be updated more often than in the middle of a phase region. Furthermore, if the local state of a grid cell changes too much such that re-computation of thermodynamic data is necessary, the local structure of the grid could be used advantageously. Quite often a neighboring grid cell could have used the necessary data in the previous time step¹². Only if no neighbor has the necessary data cached, a re-computation/lookup should be started. Even then, also the representation of the phase diagram could use local structure advantageously. Instead of $O(\log n)$ a constant time complexity (on the average) seems possible.

6.4 Computational modeling of solidification fronts

In this section we consider the PDE system (6.7) to illustrate some basic mechanisms that characterize a progressing solidification front. Emphasis is given in this model to the effects of latent heat release in the absence of flow. The model describes the phenomena in one spatial dimension only, roughly mimicking the behavior along the central axis of the ingot. It will be shown that a simple spatial discretization suffices to capture the physics of the problem and that the qualitative features of the solidification front are well captured. This implies that (6.7) can be used as an efficient vehicle for testing improvements in the thermodynamics treatment without leading to lengthy simulations. This can be beneficial in development stages of reduced thermodynamics representations, while retaining a clear view at the accuracy penalty incurred. In the future, it would be helpful to extend this simple model with a realistic thermodynamic description of the latent heat, to illustrate the computational gain that may be achieved with one of the approaches outlined above. Currently, this model is only used to illustrate the occurrence of solidification fronts in case the latent heat is only roughly parameterized.

We consider the coupled system of equations (6.7) on the unit interval $]0, 1[$. The initial temperature is taken constant and larger than the melting temperature of the mixture, denoted by T_m . Moreover, we consider the initial state to be liquid, implying that at $t = 0$ we have $\epsilon_l = 1$ throughout the system. For convenience, we drop the index l and implicitly assume that $\epsilon := \epsilon_l$ refers to the volume fraction in the liquid phase. Fully solidified material corresponds then to $\epsilon = 0$. To complete the basic description, we impose Neumann conditions at $x = 0$, i.e., put $\partial\epsilon/\partial x(0, t) =$

¹²It even seems possible to use a grid cell's spatial neighbors to interpolate the thermodynamic values at that cell, sufficiently far away from phase boundaries at least

6.4 Computational modeling of solidification fronts

$\partial T/\partial x(0, t) = 0$ and Dirichlet conditions at $x = 1$, i.e., $\epsilon(1, t) = 0$ and $T(1, t) = T_0$ where, with proper non-dimensionalization $T_0 = 1 < T_m$, indicating that at $x = 1$ the solidification front starts:

$$\begin{aligned} \frac{\partial T}{\partial t} - \frac{\partial^2 T}{\partial x^2} - L \frac{\partial \epsilon}{\partial t} &= 0 \\ \frac{\partial \epsilon}{\partial t} - \frac{\partial^2 \epsilon}{\partial x^2} &= 0 \end{aligned} \tag{6.9}$$

$$\begin{aligned} T(x, 0) &= 1 < T_m, \quad \epsilon(x, 0) = 1 \\ \frac{\partial T}{\partial x}(0, t) &= \frac{\partial \epsilon}{\partial x}(0, t) = 0 \\ T(1, t) &= 1, \quad \epsilon(1, t) = 0 \end{aligned}$$

where, for convenience, we use a unit diffusion coefficient $M = 1$.

This problem can be readily discretized using standard finite differences and an explicit time-stepping method. For convenience, we formulate the discrete model on a uniform grid $x_j = jh$ where $h = 1/N$ denotes the mesh spacing. Likewise, we choose a constant time-step Δt and approximate the solution at times $t_n = n\Delta t$. Following the usual steps, we arrive at

$$\begin{aligned} \epsilon_j^{n+1} &= \epsilon_j^n + \nu(\epsilon_{j+1}^n - 2\epsilon_j^n + \epsilon_{j-1}^n) \\ T_j^{n+1} &= T_j^n + \nu(T_{j+1}^n - 2T_j^n + T_{j-1}^n) + \Delta t L_j^n \left(\frac{\partial \epsilon}{\partial t} \right)_j^n \end{aligned} \tag{6.10}$$

for $1 \leq j \leq n-1$. Here, $\epsilon_j^n \approx \epsilon(x_j, t_n)$ and $T_j^n \approx T(x_j, t_n)$. The term $(\frac{\partial \epsilon}{\partial t})_j^n$ is approximated backward in time. At the boundaries we put $T_N^n = 1$ and $\epsilon_N^n = 0$ and use the simple approximation for the Neumann boundary at $x = 0$ as: $T_0^n = T_1^n$ and $\epsilon_0^n = \epsilon_1^n$. In this formulation $\nu = \Delta t/h^2$ which has to be kept sufficiently small in order to maintain stability of the simulation.

The effect of heat released during solidification is represented by the function L . Purely intuitively, one may expect L to be large in case the temperature is close to the melting temperature and considerably smaller at temperatures away from the melting temperature. Suitably normalized, the simplest possible discrete model for L is

$$L_j^n = \begin{cases} \beta & \alpha T_m < T_j^n < T_m \\ 1 & \text{otherwise} \end{cases} \tag{6.11}$$

where for illustration purposes we assume $\beta \gg 1$. More involved models for L can be obtained analogously to that presented in Section 3. However, at this level of detail it is sufficient to indicate the effect of heat release in this crude modeling.

Simulating the solution to the simple model can be done with a straightforward MATLAB implementation. For this purpose we adopted $T_m = 2$, $\beta = 100$ and

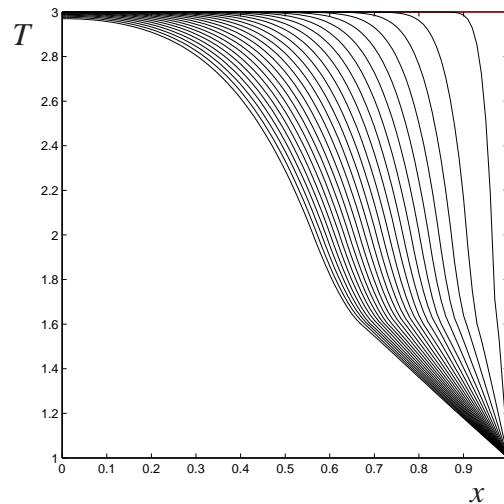


Figure 6.9: Developing temperature profile characterizing the solidification front. The solid develops from the right - subsequent curves correspond to snapshots at different times.

$\alpha = 0.8$. The moving solidification front that is obtained in this way is shown in terms of the temperature profiles in Fig. 6.9. We clearly recognize the progressing solidification. Particular to the adopted model for L is the slight jump in the derivative near the front. In Fig. 6.10 we display the effect of heat release on the location of the mushy zone. We notice that an increased heat release yields a more rapid solidification. This problem was also treated independently with an implicit time-stepping method in combination with an adaptive mesh. This allows to capture the phenomena in more detail at lower computational cost. The final results of the two codes compared very closely, thereby providing an independent check.

6.5 Concluding remarks

In this paper we described the modeling of solidification processes in aluminum casting. We emphasized the central role that the thermodynamics of solidification has. Particularly at realistic numbers of alloying elements the proper description of the thermodynamic components is a strong limiting factor. The obvious brute force approach based on minimization of the Gibbs free energy does not provide a realistic option. Rather, database approaches, not unlike those used in combustion research, need to be developed to bring the computational effort down to a more manageable level. It was argued that simply using a pre-computed database to represent the thermodynamics is insufficient and further data-reduction is mandatory. In Section 3 a simple approach based on piecewise polynomial fitting was described and shown to bring the data-handling down to a realistic level. However, the method cannot be easily extended to spatially dependent situations. For that purpose more

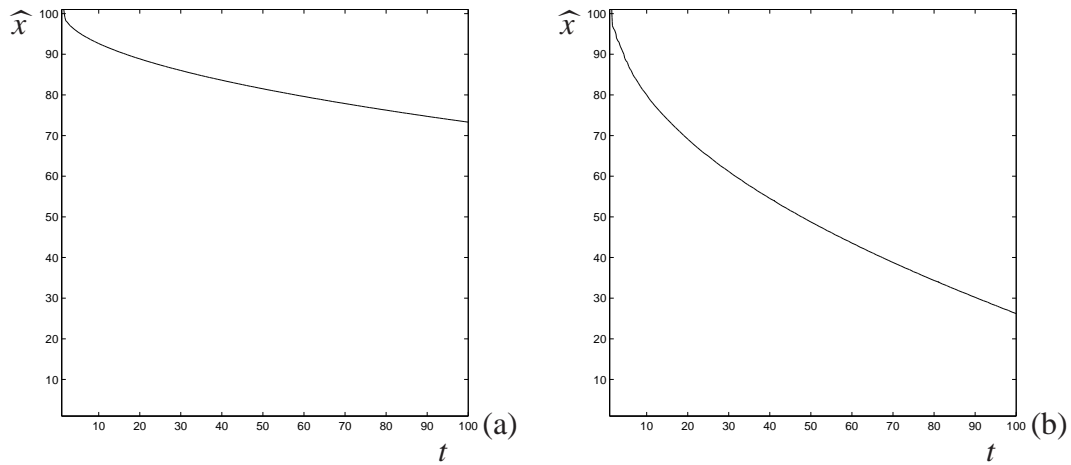


Figure 6.10: Effect of heat release on the solidification front defined at \hat{x} where $T(\hat{x}, t) = T_m$. In (a) we use $\beta = 1$ and in (b) the value $\beta = 100$ is adopted.

involved data representations and methods for efficient processing were suggested as well. The confrontation of these methods with realistic solidification simulations as are adopted in industry is still an open challenge. Based on the experience with the simplified approach, savings on the order of 100 or more appear possible without affecting the accuracy of predictions too much. While developing the improved data-base handling for solidification processes, use could be made of the simplified one-dimensional simulation model that appears to capture the main physics of a progressing solidification front at modest computational costs. This could be a helpful testing ground for the incorporation of several of the proposed data-reduction techniques and measures to speed-up the computations. Research in that direction is much needed and constitutes a challenge for the future.

Bibliography

- [1] Nadella, R. D.G. Eskin, Q. Du, L. Katgerman: 2008. Macrosegregation in direct-chill casting of aluminum alloys. *Prog. Mat. Sci.*, **53** 421–480
- [2] Mortensen, D.: 1999. A Mathematical Model of the Heat and Fluid Flows in Direct-Chill Casting of aluminum Sheet Ingots and Billets. *Metallurgical and Materials Transactions A*, **30B** 119–133
- [3] Fjaer, H.G., and A. Mo: 1990. ALSPEN – A Mathematical Model for Thermal Stresses in Direct Chill Casting of aluminum Billets. *Metallurgical and Materials Transactions A*, **21B** 1049–1061
- [4] Pequet, C., M. Gremaud, M. Rappaz: 2002. Modeling of microporosity, macroporosity, and pipe-shrinkage formation during the solidification of al-

- loys using a mushy-zone refinement method: Applications to aluminum alloys. *Metallurgical and Materials Transactions A*, **33** 2095
- [5] Rerko, R.S., H.c. de Groh, III, C. Beckermann:2003. Effect of melt convection and solid transport on macrosegregation and grain structure in equiaxed Al-Cu alloys. *Mat.Sci. Eng.* **A347** 186–197
- [6] Ahmad, N., H. Combeau, J-L. Desbiolles, T. Jalanti, G. Lesoult, J. Rappaz, M. Rappaz, and C. Stomp: 1998. Numerical Simulation of Macrosegregation: a Comparison between Finite Volume Method and Finite Element Method Predictions and a Confrontation with Experiments. *Metallurgical and Materials Transactions A*, **29A** 617–630
- [7] Du, Q., D.G. Eskin and L. Katgerman: 2007. Modeling Macrosegregation during Direct-Chill Casting of Multicomponent Aluminum Alloys. *Metallurgical and materials transactions A*, **38A** 180–189
- [8] Thermocalc Thermodynamic Database Software, www.thermocalc.com.
- [9] Factsage and Chemapp Integrated Thermodynamic Database System, www.factsage.com.
- [10] JmatPro, Thermo-physical and thermodynamic properties software, www.thermotech.co.uk.
- [11] Bear, J.: 1972. *Dynamics of Fluids in Porous Media*. Dover Publications Inc., New York
- [12] Schneider, M.C., Beckermann, C.: 1995. Formation of Macrosegregation by Multicomponent Thermosolutal Convection during the Solidification of Steel, *Metallurgical and Materials Transactions A*, **26A** 1995–2373
- [13] Goyeau, B. Bousquet-Melou, P., Gobin, D., Quintard, M., Fichot, F.: 2004. Macroscopic modeling of columnar dendritic solidification, *Computational and Applied Mathematics* **23: 2-3** 381-400
- [14] Barber, C.B., Dobkin, D.P., Huhdanpaa, H.: 1996. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software* **22** 469–483
- [15] Berg, M. de, Kreveld, M. van, Overmars, M., Schwarzkopf, O.: 2000. *Computational Geometry. Algorithms and Applications*. Springer-Verlag
- [16] Bungartz, H-J., Griebel, M.: 2004. Sparse grids. *Acta Numerica* **13** 147–269
- [17] DeHoff, R.: 2006. *Thermodynamics in Materials Science*. CRC Press

- [18] Doré, X., Combeau, H., Rappaz, M.: 2000. Modelling of Microsegregation in Ternary Alloys: Application to the Solidification of Al-Mg-Si. *Acta materialia* **48** 3951–3962
- [19] Eriksson, G., Hack, K.: 1990. ChemSage — A Computer Program for the Calculation of Complex Chemical Equilibria. *Metallurgical Transactions B* **21** 1013–1023
- [20] Eriksson, G., Spencer, P.J., Sippola, H.: 1995. A General Thermodynamic Software Interface. In: A. Jokilaakso (ed.), Proceedings of the 2nd Colloquium on Process Simulation, pg. 113, Espoo, Finland, Helsinki University of Technology, Report TKK-V-B104
- [21] Klimke, A., Wohlmuth, B.: 2005. Algorithm 847: spinterp: Piecewise Multilinear Hierarchical Sparse Grid Interpolation in MATLAB. *ACM Transactions on Mathematical Software* **31** 561–579
URL: <http://www.ians.uni-stuttgart.de/spinterp/index.html>
- [22] Persson, P.-O., Strang, G.: 2004. A Simple Mesh Generator in MATLAB. *SIAM Review* **46** 329–345
- [23] Persson, P.-O.: 2005. Mesh Generation for Implicit Geometries. PhD Thesis, Department of Mathematics, MIT
- [24] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: 2002. *Numerical Recipes in C++*. Cambridge University Press
- [25] Zhao, H.-K., Osher, S., Fedkiw, R.: 2001. Fast Surface Reconstruction Using the Level Set Method, In: VLSM Proceedings of the IEEE Workshop on Variational and Level Set Methods (VLSM'01), pg. 194