TU Delft

Delft University of Technology

Document Version
Final published version

Licence
CC BY

Citation (APA)
Liang, X., Bos, C., Hermans, M., & Richardson, I. (2023). An Improved Cellular Automata Solidification Model Considering Kinetic Undercooling. *Metallurgical and Materials Transactions B: Process Metallurgy and Materials Processing Science*, *54*(3), 1088-1098. https://doi.org/10.1007/s11663-023-02742-3

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

ORIGINAL RESEARCH ARTICLE

An Improved Cellular Automata Solidification Model Considering Kinetic Undercooling

XIAOHUI LIANG, CORNELIS BOS, MARCEL HERMANS, and IAN RICHARDSON

A cellular automata (CA) model has been developed for solidification simulation considering the kinetic undercooling at the interface. The state-of-the-art model incorporates a decentered growth algorithm to suppress the grid anisotropy and a generalized height function method to calculate the curvature accurately. To develop a CA model which is independent of the mesh size, a new diffusion term is proposed to handle the diffusion between the interface cells and liquid cells. The developed CA model is employed to simulate the single-dendritic solidification of an Al–3Cu (wt pct) alloy. The simulated tip velocities agree with the prediction of the Kurz–Giovanola–Trivedi (KGT) model. Further studies show that the developed CA model converges to an equilibrium model with increasing kinetic mobility values. Moreover, it is found that the virtual liquid cell assumption which is commonly used in existing CA models may lead to a deviation in the mass balance. The mass balance error has been resolved by redistributing solutes from neighboring liquid cells in each time step. The developed CA model could be potentially used in solidification simulations with a high undercooling, which is common in welding and additive manufacturing.

https://doi.org/10.1007/s11663-023-02742-3
© The Author(s) 2023

I. INTRODUCTION

DENDRITE formation is commonly observed during solidification of welding or casting. Solidification simulation is important for a better understanding of dendrite formation and a better control of welding or casting processes, as it can provide *in situ* information which is inaccessible through an experimental approach.[1] Various models including phase field models,[2–5] cellular automata (CA) models[6–9] and level set models[10–12] have been employed for solidification simulations. Compared to other approaches, CA models require only moderate computational resources and are widely used for solidification simulations.

Most CA models[13–15] assume local equilibrium at the interface and calculate the growth velocity with a diffusion-controlled method. Nastac[13] developed a time-dependent CA model where the interface velocity v is calculated with a flux balance at the interface

$$v = \frac{1}{c^{l,eq}(1-k)}\left(\nabla c^{l} - \nabla c^{s}\right)\cdot\vec{n}, \qquad [1]$$

where $c^{l,eq}$ is the equilibrium liquid concentration at the interface, k the partitioning coefficient, $\vec{n}$ the interface normal unit vector, c^{l} and c^{s} the liquid and solid concentration, respectively. This method is called front velocity method by Reuther and Rettenmayr.[1] Following this approach, Beltran-Sanchez and Stefanescu[16,17] proposed a virtual front tracking method to capture liquid cells within a CA simulation. The stable tip velocity obtained from their CA simulations agreed with prediction from the Lipton–Glicksman–Kurz model.[18] Later, Michelic *et al.*[19] extended the virtual front tracking model to simulate solidification in a multi-component system. Zhang *et al.*[20] employed a CALPHAD approach to calculate the equilibrium concentrations and the growth velocity in 3D solidification simulations of an Al–Cu–Mg melt.

Apart from the approach of Nastac,[13] Zhu and Stefanescu[21] proposed a cellwise mass balance method to calculate the growth velocity in a CA model assuming local equilibrium condition. Assuming the liquid concentration of the interface cells is always equal to the equilibrium concentration, the solid fraction change Δf_s in a time increment is given by a lever rule[21]

XIAOHUI LIANG and MARCEL HERMANS are with the Materials Science and Engineering, TU Delft, Mekelweg 2, 2628 CD Delft, The Netherlands. Contact e-mail: x.liang-2@tudelft.nl CORNELIS BOS is with the Materials Science and Engineering, TU Delft and also with the Research & Development, Tata Steel, 1970 CA IJmuiden, The Netherlands. IAN RICHARDSON is with the Materials Science and Engineering, TU Delft and also with the IR Welding Consultancy, 2651 LV Berkel en Rodenrijs, The Netherlands.
Manuscript submitted August 15, 2022; accepted January 26, 2023.
Article published online February 21, 2023.

$$\Delta f_s = \frac{c^{l,eq} - c^l}{c^{l,eq}(1-k)}. \quad [2]$$

This approach has been employed by Pan and Zhu[22] to simulate the solidification in a 3D case. Yin *et al.*[23] coupled a cellwise mass balance CA model with a lattice Boltzmann model to simulate solidification under a melt flow.

For the rapid solidification encountered in welding and additive manufacturing, consideration of non-equilibrium effects is necessary. According to the interface attachment kinetics, the interface velocity v is related to the kinetic undercooling ΔT_k by[24]

$$v = v_0\left(1 - \exp\left(-\frac{\Delta S \Delta T_k}{RT}\right)\right), \quad [3]$$

where R the gas constant, T the temperature, ΔS the entropy change and v_0 a constant which is of the order of the velocity of sound for pure metals. When $v \ll v_0$, this relationship can be simplified to

$$v = \mu \Delta T_k, \quad [4]$$

where μ is the kinetic mobility.

Several CA models[15,25,26] have been developed which consider non-equilibrium effects during solidification. Zhao *et al.*[27] developed a 3D CA model which calculated the growth velocity based on Eq. [3] and simulated the solidification of a Fe–1.5C (wt pct) alloy in a 3D case. Burbelko *et al.*[25] calculated the growth velocity as a linear function of the kinetic undercooling and studied the formation of the primary austenite and globular eutectic grains during the solidification of a ductile iron in a thin wall casting process. Zhu *et al.*[15] developed a two-dimensional CA model to predict the microstructures and micro-segregation in a solidified ternary alloy. Geng *et al.*[26] used a non-equilibrium CA model to simulate the dendrite growth along the fusion boundary of a laser beam weld.

To improve the quantitativeness of the CA model, a number of works have been published in different areas including curvature calculation,[28,29] grid anisotropy,[30,31] and mesh size dependency.[16,19,21] For a quantitative solidification simulation, curvature at the CA interface cells needs to be calculated accurately. In early CA models, a cell counting method[13,30] or a level set method[16,19,21] was employed to calculate the curvature at the interface cells based on the solid fraction field. Previously, Reuther and Rettenmayr[28] compared the performance of the cell counting method, the level set method and a height function method in two benchmark problems and concluded that the height function method calculates curvature more accurately than the other two methods. Wei *et al.*[29] adopted a modified height function method to calculate curvature in a eutectic solidification simulation.

Grid anisotropy is commonly observed when a simple capture rule is used for dendrite growth.[31] Without any crystal anisotropy, dendrite arms tend to form in the grid directions when a von Neumann capture rule is employed, whereas dendrite arms tend to form in the diagonal directions of the grid when a Moore capture rule is employed. To suppress the grid anisotropy, different growth algorithms have been developed. Zhu and Stefanescu[21] employed a virtual front tracking method to capture liquid cells. The decentered growth algorithm proposed by Gandin and Rappaz[32] has been widely used to simulate formation of dendrites with different orientations in 2D[30,33] and 3D[34] simulations. The decentered growth algorithm exhibited a good performance in suppressing the grid anisotropy.

A quantitative CA model should be independent of the mesh size. In most equilibrium CA models,[16,19,21] convergence analyses have been performed. The growth velocity at the dendrite tip in single-dendritic solidification simulations converges to a stable value with decreasing mesh size. However, no convergence analysis has been found in any non-equilibrium CA model. In this case, existing non-equilibrium CA models must be regarded as qualitative.

Moreover, a quantitative CA model should be mass conserved. In most CA models,[16,19] diffusion within the liquid and solid regions are solved separately. To handle the discontinuity at the interface, interface cells are virtually treated as liquid cells. The virtual liquid assumption neglects the solid fraction of the interface cells and may lead to a mass balance deviation in CA models with a front velocity method, as mentioned by Michelic *et al.*[19] Reuther and Rettenmayr[1] indicated that CA models based on a cellwise mass balance offer a better solutes conservation compared to CA models with a front velocity method. To develop a quantitative CA model, further study on the influence of the virtual liquid cell assumption is necessary.

In this work, we describe a quantitative non-equilibrium CA model. A decentered growth algorithm[32] is employed to suppress the grid anisotropy and a generalized height function method[35] is adopted to calculate the curvature accurately. To make a CA model which is independent of the mesh size, a new diffusion term is proposed to handle the diffusion between interface cells and liquid cells. In addition, the influence of the virtual liquid cell assumption is studied within a multi-dendritic solidification simulation. A correction term is defined and applied to ensure a mass balance within the CA simulation.

II. CA MODEL SETUP

A. *Basics*

In the CA model, the domain to be studied is discretized into many square cells. Each cell has state variables including phase state (to which phase it belongs), grain index (to which grain it belongs), solid fraction f_s and concentration c. For a solidification problem, possible phase states and solid fraction are liquid ($f_s = 0$), interface ($0 < f_s < 1$) and solid ($f_s = 1$) states. The interface cells also have additional state variables including growth velocity v, growth length l and interface normal vector $\vec{n}$.

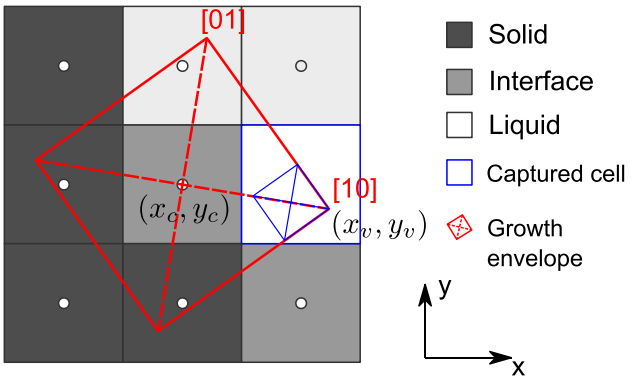

Fig. 1—Illustration of the capturing rule in the decentered growth algorithm.

Solidification within the CA model is simulated by updating the solid fraction and the phase status of the cells near the solidification interface, which is performed based on following transition rules:

- a liquid cell transforms into an interface cell when it is captured by a growing interface cell with a decentered growth envelope capture algorithm;
- an interface cell transforms into a solid cell, when it has no liquid neighbor in its von Neumann neighborhood.

B. *The Decentered Growth Algorithm*

The decentered growth algorithm developed by Gandin and Rappaz[32] is employed in the current model. The growth of each interface cell is described by a growth envelope, which is a quadrangle in the 2D case. Each growth envelope is orientated with its half diagonals parallel with the ⟨10⟩ preferential growth directions of the solid grain. The length of the envelope half diagonal is defined as the growth length l, which records dendrite growth in the preferential growth direction. In each time increment, the growth length of each interface cell is updated according to

$$\Delta l = l^{t+\Delta t} - l^t = v\Delta t, \qquad [5]$$

where v is the growth velocity in the interface cell, Δl the increase in the growth length and Δt the time increment.

The capture rule of the decentered growth algorithm is shown in Figure 1. If the center of a liquid cell falls in the growth envelope of an interface cell, then this liquid cell is captured and transforms into a new interface cell. Within the new interface cell, a new growth envelope is formed with one of its vertices overlapping with the nearest vertex of the parent growth envelope. The new growth envelope inherits the orientation of the parent growth envelope. Its initial growth length is defined as a ratio (α) of the growth length of the parent growth envelope. If we denote the coordinates of the center and the nearest vertex of the parent growth envelope with $(x_c,\ y_c)$ and $(x_v,\ y_v)$, then the center of the new growth envelope is given by $((1-\alpha)x_c + \alpha x_v,\ (1-\alpha)y_c + \alpha y_v)$.

C. *Calculation of the Growth Velocity*

At the interface, the thermodynamic equilibrium is given by[24]

$$\Delta T = T_0 - T = \Delta T_k + \Delta T_c + \Delta T_r, \qquad [6]$$

where ΔT is the local undercooling, T_0 the liquidus temperature, ΔT_k the kinetic undercooling, ΔT_c the constitutional undercooling and ΔT_r the curvature undercooling. The kinetic undercooling ΔT_k,[24] the constitutional undercooling ΔT_c and the curvature undercooling ΔT_r[21] are given by

$$\Delta T_k = \frac{v}{\mu}, \qquad [7]$$

$$\Delta T_c = -m(c^{l,*} - c^0), \qquad [8]$$

$$\Delta T_r = \Gamma\kappa f(\phi,\theta), \qquad [9]$$

where $c^{l,*}$ is the liquid concentration of the interface cell, m the slope of liquidus line in the linearized phase diagram, c^0 the nominal concentration, Γ the Gibbs–Thomson coefficient, κ the curvature and $f(\phi,\theta)$ the anisotropic function for the interfacial energy, which is given by[21]

$$f(\phi,\theta) = 1 - 15\varepsilon\cos(4(\phi-\theta)), \qquad [10]$$

where ε is the anisotropy coefficient of the interfacial energy, θ the angle between the preferential growth direction and the x axis and ϕ the angle between the interface normal $\vec{n}$ and the x axis. The interface normal vector $\vec{n}$ is determined by[19]

$$\vec{n} = \frac{\nabla f_s}{|\nabla f_s|}. \qquad [11]$$

The interface velocity v is then given by

$$v = \mu\big(T_0 - T + m(c^{l,*} - c^0) - \Gamma\kappa f(\phi,\theta)\big). \qquad [12]$$

After reformulation, the interface velocity v is given by

$$v = \mu m(c^{l,*} - c^{l,eq}), \qquad [13]$$

with the equilibrium concentration $c^{l,eq}$ under a Gibbs–Thomson effect given by

$$c^{l,eq} = c^0 - \frac{T_0 - T - \Gamma\kappa f(\phi,\theta)}{m}. \qquad [14]$$

In some cases, the interface concentration $c^{l,*}$ might be larger than the equilibrium concentration $c^{l,eq}$, which leads to a negative interface velocity for an alloy with $m<0$. This is because the partitioned solutes in previous time steps does not have enough time to diffuse out of the interface cell. As we are simulating a solidification problem without considering remelting, the interface velocity is limited with

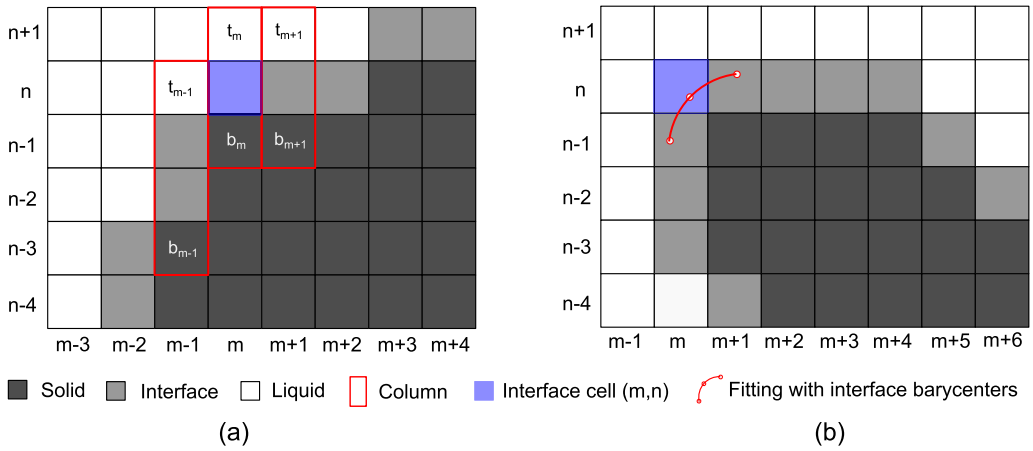

Fig. 2—Illustration of the GHF curvature calculation method: (*a*) when it is possible to establish an adaptive stencil with three consistent columns, generalized height function is employed for curvature calculation, and (*b*) when it is not possible to establish an adaptive stencil with three consistent columns in any principal direction, curvature is evaluated by fitting the barycenter of the interface segments.

$$v = \max(v, 0). \quad [15]$$

After several time steps, the interface concentration drops to a value below the equilibrium concentration due to diffusion, which brings the interface velocity back to a positive value.

The solid fraction of each interface cell is updated in each time increment with[33]

$$\Delta f_s = \frac{v\Delta t}{\Delta x(|\cos\theta| + |\sin\theta|)}, \quad [16]$$

where Δt is the time increment.

The interface solid concentration $c^{s,*}$ is given by

$$c^{s,*} = kc^{l,*}. \quad [17]$$

The solid concentration c^s of each interface cell is calculated by averaging the interface solid concentration $c^{s,*}$ over different time increments[21]

$$c^s = \frac{\sum c_n^{s,*}\Delta f_s}{\sum \Delta f_s}, \quad [18]$$

where $c_n^{s,*}$ is the solid concentration at the interface in increment n.

D. *Calculation of the Curvature*

In this work, the generalized height function method proposed by Popinet[35] is adopted to calculate the curvature of each interface cell.

Consider an interface cell with index (m, n), as shown in Figure 2. To calculate the curvature, an adaptive stencil consisting of three columns is constructed in the direction of the largest component of the interface normal $\vec{n}$. A column is called consistent if it has a base cell ($f_s = 1$) and a top cell ($f_s = 0$) and the solid fraction decreases monotonically from the base cell to the top cell. The indices of the base cell and the top cell within the column are b and t, respectively. If all columns in this stencil are consistent, then the height of each column is calculated by summing up the solid fraction from the base cell to the top cell. If the largest component of the interface normal $\vec{n}$ is in the y direction, then the height of each column is written as[35]

$$h_i = b_i + \Delta x \sum_{j=b_i}^{j=t_i} f_s(i,j), \quad \text{for } i = m-1, m, m+1. \quad [19]$$

A height function $H(x)$ is defined between the heights of the three columns and their distances to the central column. Curvature is calculated based on the derivatives of the height function,[35]

$$\kappa = -\frac{H_{xx}}{\left(1 + H_x^2\right)^{\frac{3}{2}}}, \quad [20]$$

where H_x and H_{xx} are the first and the second derivative of the height function, which are calculated with a central finite difference.

If it is not possible to construct a consistent stencil, then curvature is determined by fitting the interface with a parabola.[35] First, the interface segments in the considered interface cell and neighboring interface cells are determined with a piecewise linear interface calculation.[17] Then, the barycenter of the interface segments in the considered interface cell and neighboring interface cells are fitted with a parabola in a rotated coordinate system which is defined with its y axis parallel with the interface normal $\vec{n}$ of the considered interface cell. The parabola function is[35]

$$y = a_0 + a_1 x + a_2 x^2, \quad [21]$$

where a_0, a_1 and a_2 are fitting parameters. Curvature at the considered interface cell is then given by[35]

$$\kappa = -\frac{2a_2}{\left(1 + a_1^2\right)^{\frac{3}{2}}}. \quad [22]$$

Table I. Material Properties of an Al–3Cu (Wt Pct) Alloy[36]

Symbol	Description	Value	Unit
c^0	nominal concentration	3.0	wt pct
D^l	liquid diffusion coefficient	3.0×10^{-9}	m^2/s
D^s	solid diffusion coefficient	3.0×10^{-13}	m^2/s
k	partitioning coefficient	0.17	1
m	liquidus slope	−2.6	K/(wt pct)
T_0	liquidus temperature	650.6	°C
Γ	Gibbs–Thomson coefficient	2.4×10^{-7}	K/m
ε	anisotropy coefficient	0.0267	1
μ	kinetic mobility (default)	1×10^{-3}	$m\,s^{-1}\,K^{-1}$

E. *Diffusion and Partitioning*

In the current CA model, diffusion is solved for the solid and liquid regions separately. The governing equations for the diffusion between the liquid cells and the diffusion between the solid cells are given by

$$\frac{\partial c^l}{\partial t} = \nabla \cdot \left(D^l \nabla c^l\right), \quad [23]$$

$$\frac{\partial c^s}{\partial t} = \nabla \cdot \left(D^s \nabla c^s\right), \quad [24]$$

where D^l and D^s are the diffusion coefficients in the liquid and solid, respectively.

In traditional CA models,[17,19,20] partitioning at the interface is considered by adding a source term on the right-hand side of Eq. [23],

$$\frac{\partial c^l}{\partial t} = \nabla \cdot \left(D^l \nabla c^l\right) + \frac{\partial f_s}{\partial t}\left(c^{l,*} - c^{s,*}\right), \quad [25]$$

where $c^{l,*}$ and $c^{s,*}$ are the interface concentration in the liquid and solid. However, as will be shown in Section III–A, directly using Eq. [25] in a non-equilibrium CA model leads to a large mesh size dependency. In order to avoid this, the diffusion between interface cells and the liquid cells are handled with a new equation,

$$\frac{\partial c^l}{\partial t} = \nabla \cdot \left(D^l \nabla c^l \cdot \vec{n}\right) + \frac{\partial f_s}{\partial t}\left(c^{l,*} - c^{s,*}\right). \quad [26]$$

With Eq. [26], the sum of the projections of the diffusion fluxes out of the interface cells onto the interface normal direction are calculated.

Equation [25] or [26] is solved with a finite difference method and a Euler forward discretization. Meanwhile, the virtual liquid cell assumption has been employed, in which the interface cells within the CA model are virtually treated as liquid cells.[17,19,20] The virtual liquid cell assumption avoids extremely large concentration changes when the liquid fraction of the interface cell is close to 0. However, it introduces a deviation into the mass balance. Although the mass balance deviation problem may be insignificant in a single time step, the error adds up over numerous time steps as reported by Michelic *et al.*[19]

Consider an interface cell in a CA simulation using a virtual liquid cell assumption. In a time increment, the concentration change of the interface cell is Δc, which is calculated as the product of time step Δt and the right-hand side of the diffusion equation (Eq. [25] or [26]). The value of Δc corresponds to the concentration change in a liquid cell. The solid fraction of the interface cell is neglected, although it has a value which is larger than or equal to 1. In this case, an error of $-f_s \Delta c$ is introduced in the mass balance. Note Δc is equal to the sum of the concentration change due to diffusion Δc^d and the concentration change due to solutes partitioning Δc^p, which are calculated by the first and the second term on the right-hand side of the diffusion equation. In this case, Δc can either be positive or negative, depending on the relative value of the Δc^d and Δc^p. If Δc is positive, then solutes are artificially lost and the average concentration of the system decreases. If Δc is negative, then solutes are artificially added and the average concentration of the system increases. To redress the mass balance error, the concentration of the neighboring liquid cells is modified by $\frac{f_s \Delta c}{n_l}$, where n_l is the number of liquid cells in the Moore neighborhood of the considered interface cell. In this case, the artificially lost or added solutes are added or removed from neighboring liquid cells to keep a mass balance.

F. *Time Increment*

In the CA model, the time increment is determined by

$$\Delta t = \min\left(0.2\frac{\Delta x}{v_{\max}},\ 0.2\frac{\Delta x^2}{D^l}\right), \quad [27]$$

where $v_{\max}$ is the maximum growth velocity within the simulation domain and Δx the mesh size.

G. *Material Parameters*

The presented model can be applied for the solidification simulation of any binary and (with small modification) multi-component alloys. Here, the

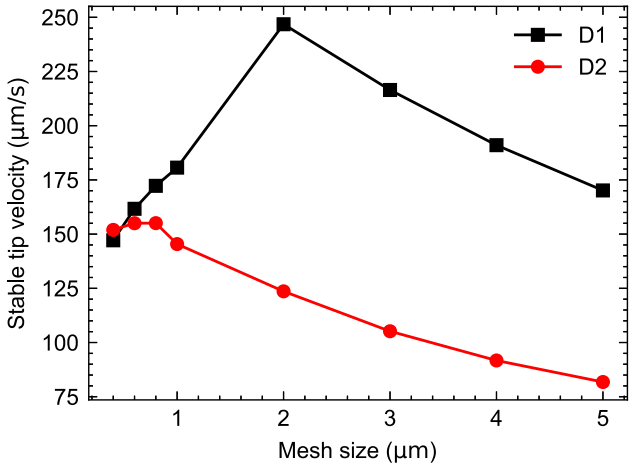

Fig. 3—The stable tip velocity in the mesh convergence tests for the CA model using D1 and D2.

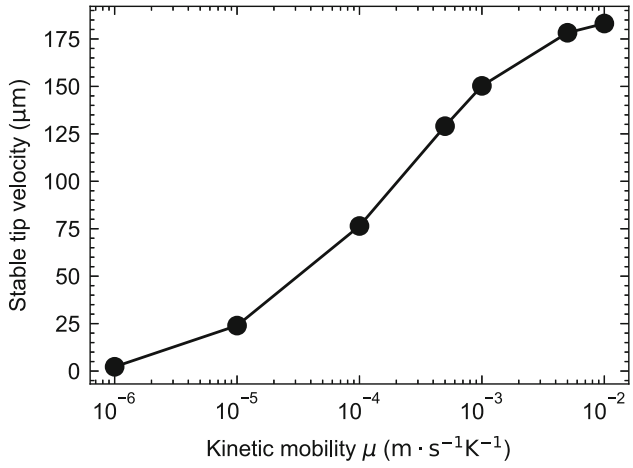

Fig. 4—The stable tip velocity in the simulations with different kinetic mobility μ values.

performance of the model is illustrated by simulation of solidification in a Al–3Cu (wt pct) alloy which has been well studied in literature.[17,36] The parameters of the Al–3Cu (wt pct) alloy are given in Table I.

III. RESULTS AND DISCUSSION

A. *Mesh Convergence Tests*

To test the mesh size dependency of the developed CA model, single dendritic growth of the Al–3Cu (wt pct) alloy has been simulated with different mesh sizes under a constant undercooling of 3 K. Due to the discretization of the CA model, the velocity at the dendrite tip does not change monotonically. To smooth the tip velocity curve, the tip velocity is averaged between two tip cell advancements. The averaged tip velocity decreases with increasing time and eventually converges to a stable value. The stable tip velocity is evaluated at time 0.3 seconds.

The mesh size convergence test has been performed for the CA model with Eq. [25] (D1) and Eq. [26] (D2). In the D1 simulation, the diffusion between interface cells and liquid cells is calculated by summing up the magnitudes of all the diffusion fluxes, whereas in the D2 simulation, the diffusion between interface cells and liquid cells is calculated by summing up the projections of the diffusion fluxes onto the interface normal $\vec{n}$ direction. The choice of diffusion equation has an influence on the mesh size convergence behavior. As shown in Figure 3, with decreasing mesh size, the stable tip velocity in the D1 simulations first increases and then decreases. In the D2 simulations, the stable tip velocity converges to a value around 150 μm/s, indicating that simulations without a mesh size effect can be performed if Eq. [26] is applied.

The different mesh size convergence behaviors can be explained by considering the interface cell at a dendrite tip in an isothermal solidification simulation. Discretizing the diffusion equation gives

$$\Delta c^{\mathrm{l}} = -\frac{J\Delta t}{\Delta x} + \Delta f_{\mathrm{s}}(c^{\mathrm{l},*} - c^{\mathrm{s},*}), \qquad [28]$$

where J is the effective diffusion flux out of the tip cell and is equal to $D^{\mathrm{l}}\nabla c^{\mathrm{l}}$ in a D1 simulation and $D^{\mathrm{l}}\nabla c^{\mathrm{l}} \cdot \vec{n}$ in a D2 simulation. Substituting Eq. [16] into Eq. [28] with $\theta = 0$ gives

$$\Delta c^{\mathrm{l}} = -\frac{J\Delta t}{\Delta x} + \frac{v\Delta t}{\Delta x}(c^{\mathrm{l},*} - c^{\mathrm{s},*}). \qquad [29]$$

In the steady state, the liquid concentration of the tip cell is constant $\Delta c^{\mathrm{l}} = 0$, which indicates a balance between the solutes addition due to partitioning and solutes removal due to diffusion. This means that the dendrite growth is diffusion-limited in the steady state. From Eq. [29], the growth velocity at the steady-state dendrite tip in a D1 simulation can be written as

$$v = \frac{D^{\mathrm{l}}\nabla c^{\mathrm{l}}}{c^{\mathrm{l},*} - c^{\mathrm{s},*}}, \qquad [30]$$

while the growth velocity at a steady-state dendrite tip in a D2 simulation can be written as

$$v = \frac{D^{\mathrm{l}}\nabla c^{\mathrm{l}} \cdot \vec{n}}{c^{\mathrm{l},*} - c^{\mathrm{s},*}}. \qquad [31]$$

It is clear that Eq. [31] is in the same form as the equation proposed by Nastac.[13] In the D1 simulations, the tip velocity v is proportional to the sum of the magnitude of all the diffusion fluxes out the interface cell, while in the D2 simulation, the tip velocities v is proportional to the diffusion flux in the interface normal direction $\vec{n}$. In this case, the tip velocity tends to be overestimated especially when the tip cell is surrounded by three liquid cells in its von Neumann neighborhood, as indicated by Reuther and Rettenmayr.[1] This explains that the stable tip velocities in the D1 simulations are larger than the stable tip velocities in Figure 3. As we decrease the mesh size, the dendrite tip is discretized with more cells. In this case, the overestimation of tip velocity in the D1 simulations becomes weaker. Thus, the stable tip velocity in the D1 simulations decreases with a decreasing mesh size.

$$\Delta T = \Delta T_k + \Delta T_c + \Delta T_r. \quad [32]$$

The kinetic undercooling ΔT_k is given by[24]

$$\Delta T_k = \frac{v}{\mu}. \quad [33]$$

The constitutional undercooling is given by[37]

$$\Delta T_c = mc^0\left(1 - \frac{1}{1-(1-k)I_v(P_e))}\right), \quad [34]$$

where $I_v(P_e)$ is the Ivantsov function and P_e the Peclet number given by[37]

$$P_e = \frac{rv}{2D}, \quad [35]$$

where r is the tip radius and v the tip velocity. The 2D Ivantsov function is given by[37]

$$I_v(P_e) = \sqrt{\Pi P_e}\exp(P_e)\mathrm{erfc}(\sqrt{P_e}). \quad [36]$$

The curvature undercooling ΔT_r is given by[37]

$$\Delta T_r = \frac{\Gamma}{r}. \quad [37]$$

The tip radius is determined with a stability criterion[37]

$$-m\frac{vc^{l,*}(1-k)}{D^l} = \frac{1}{\sigma^*}\frac{\Gamma}{r^2}, \quad [38]$$

where $c^{l,*}$ the liquid concentration at the interface and σ^* the stability factor. The stability factor σ^* here is 0.10654.[38]

The stable tip velocities in the CA simulations with different undercoolings and different kinetic mobility μ are compared with the predictions of the modified KGT model, as shown in Figure 6. As the kinetic mobility μ increases, the stable tip velocities predicted by the CA model and KGT model under different undercoolings increases. When the undercooling is larger than 2 K, the stable tip velocities simulated by the CA model with different kinetic mobility μ agree well with the KGT model. However, at the 2 K undercooling, the simulated tip velocities in the CA simulations with different kinetic mobility μ are larger than the predictions of the modified KGT model. Besides, the stable tip velocity in the CA simulation with a 6 K undercooling and a $5 \times 10^{-3}\,\mathrm{m\,s^{-1}\,K^{-1}}$ kinetic mobility is smaller than the prediction of the modified KGT model. The reason for the mismatch is that a constant stability factor σ^* is used in the KGT model. However, this factor varies with different nominal concentration c^0 and undercooling ΔT, as indicated by Ramirez and Beckermann.[39]

D. *Influence of the Virtual Liquid Cell Assumption*

In the current CA model, interface cells are virtually treated as liquid cells to solve for diffusion between interface cells and liquid cells. As explained in Section II–E, such an assumption neglects the solid amount of the interface cells and may lead to a deviation in the mass balance. To study the influence of the virtual liquid

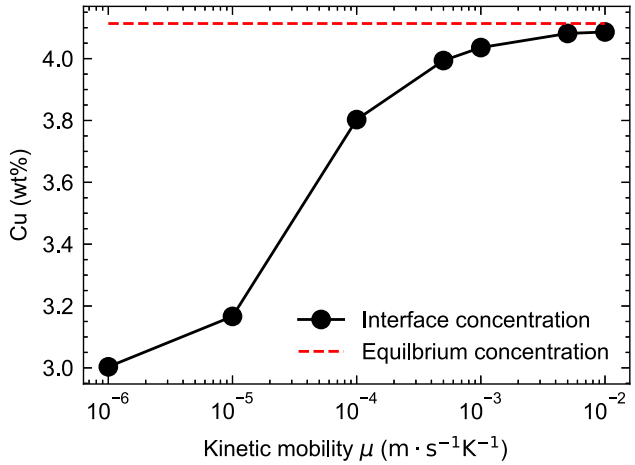

Fig. 5—The Cu concentration at the dendrite tip in the simulations with different kinetic mobility μ values.

B. *Influence of the Kinetic Mobility*

In the current CA model, the growth velocity is calculated as a product of the kinetic mobility μ and the kinetic undercooling ΔT_k. To test the influence of the kinetic mobility, single dendritic growth of the Al–3Cu (wt pct) alloy has been simulated with different mobility values under a constant undercooling of 3 K with a mesh size of 0.5 μm. The stable tip velocities are plotted in Figure 4. With increasing kinetic mobility, the stable tip velocity increases and tends to converge. As the kinetic mobility increases, the kinetic undercooling ΔT_k decreases and the simulation approaches a fully diffusion-controlled solidification simulation. Moreover, the Cu concentration $c^{l,*}$ at the dendrite tip approaches the equilibrium concentration $c^{l,eq}$ with increasing kinetic mobility, as shown in Figure 5. This is reasonable, as the non-equilibrium effect becomes less significant with increasing kinetic mobility μ. When the kinetic mobility is infinite, local equilibrium is achieved at the interface.

Increasing the kinetic mobility makes the simulation more computationally expensive. A large kinetic mobility means that a small deviation of the interface concentration can lead to a large interface velocity. This results in a smaller time step, since the time step is limited by the maximum interface velocity in the system according to Eq. [27]. As the time step is decreased, the number of time steps to finish a simulation increases, which means the computational cost increases.

C. *Comparison with the KGT Model*

The developed CA model is verified by comparing with the KGT model.[18,37] The KGT gives an analytical solution for a dendrite growing into an infinite melt in the steady state. It is modified here to include the kinetic undercooling. At the interface, the total undercooling ΔT is given by[24]

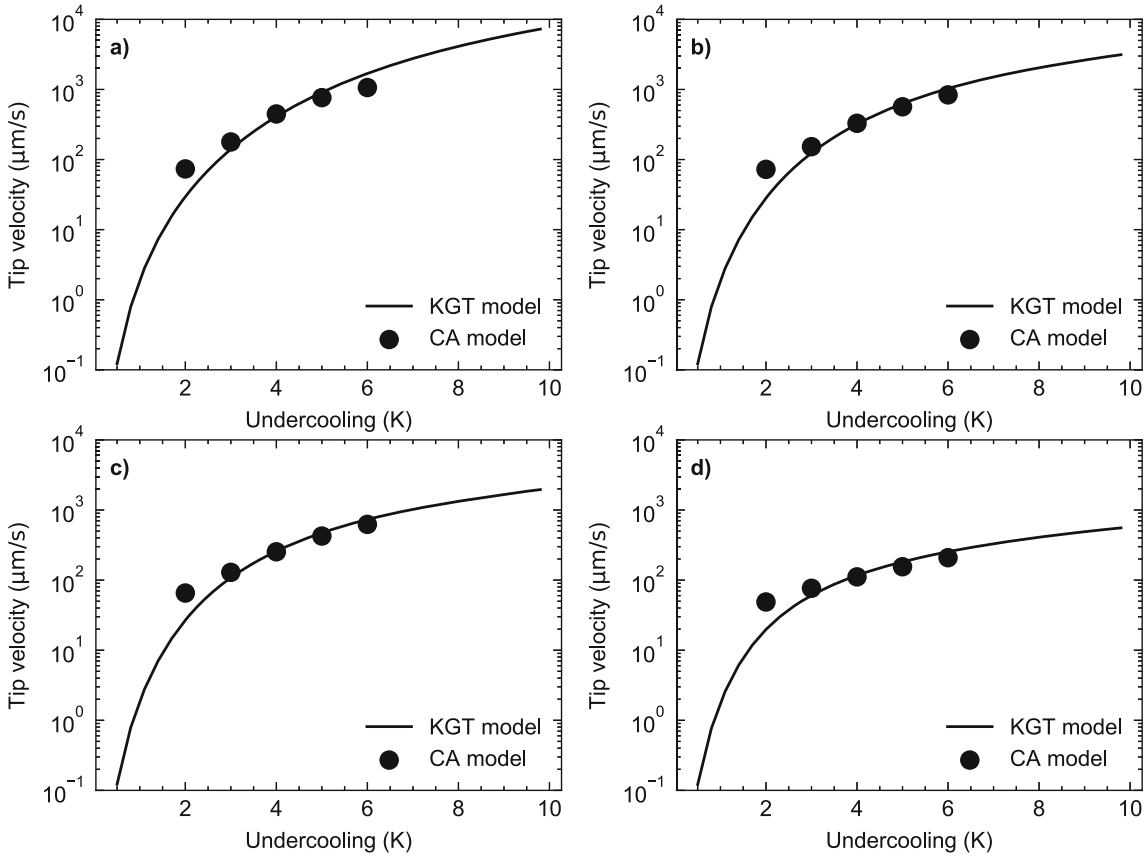

Fig. 6—Comparison between the stable tip velocities predicted by the CA model and KGT model with different undercooling and different kinetic mobility μ: (*a*) $\mu = 5 \times 10^{-3}\,\mathrm{m\,s^{-1}\,K^{-1}}$, (*b*) $\mu = 1 \times 10^{-3}\,\mathrm{m\,s^{-1}\,K^{-1}}$, (*c*) $\mu = 5 \times 10^{-4}\,\mathrm{m\,s^{-1}\,K^{-1}}$, and (*d*) $\mu = 1 \times 10^{-4}\,\mathrm{m\,s^{-1}\,K^{-1}}$.

cell assumption on the mass balance, multi-dendritic solidification of the Al–3Cu (wt pct) alloy has been simulated with and without the mass balance correction. The simulation domain size is $300 \times 300\,\mu\mathrm{m}^2$ and the mesh size is $0.5\,\mu$m. A periodic boundary condition is employed for the simulation. The cooling rate is 50 K/s. The simulation finishes after 62,000 steps and takes 13 minutes with 24 cores of Intel XEON E5-6248R at 3.0 GHz. The concentration profiles at times 0.10, 0.15 and 1.0 seconds in the simulations with and without the mass balance correction are given in Figure 7.

Initially, 12 nuclei with random orientations are placed randomly within the simulation domain. Those nuclei grow and develop dendrite arms in their preferential growth directions, as shown in Figures 7(a) and (d). This shows that dendrites with different orientations can be well simulated with the decentered growth algorithm. As solidification proceeds, the growth of the primary arms is suppressed by nearby dendrites. Secondary dendrite arms form, as shown in Figures 7(b) and (e). Most liquid in the inter-dendritic regions is enriched to the equilibrium concentration. With further solidification, the dendrite arms coarsen and the liquid concentration increases with increasing undercooling, as shown in Figures 7(c) and (f). At this stage, the growth velocity and the kinetic undercooling in the interface cells are very small. In this case, the liquid concentration at the interface cells is close to the equilibrium liquid concentration and the solidification can be approximated by the Scheil–Gulliver solidification condition.[40] Comparing Figures 7(a) through (c) with Figures 7(d) through (f), the grain morphology difference between the simulations with and without the mass balance correction is insignificant. Circles have been employed to highlight the small differences between Figures 7(c) and (f). Compared to Figure 7(c), the liquid channels in Figure 7(f) are smaller and more coalescence is observed, which indicates that the solid fraction in the simulation without the mass balance correction (Figure 7(f)) is larger than the solid fraction in the simulation with the mass balance correction (Figure 7(c)). Besides, the small difference between the two different simulations indicates that the mass balance correction has little influence on the growth kinetics.

Figure 8 shows the evolution of the average concentration in the simulations with and without the mass balance correction. The average concentration in the simulation with the mass balance correction remains constant, while the average concentration in the simulation without the mass balance correction decreases continuously. This indicates that solutes are lost in the simulation without the mass balance correction. In the multi-dendritic solidification simulation, the temperature drops continuously, which leads to an increase in the equilibrium liquid concentration. As shown in Section III–B, the interface concentration is close to

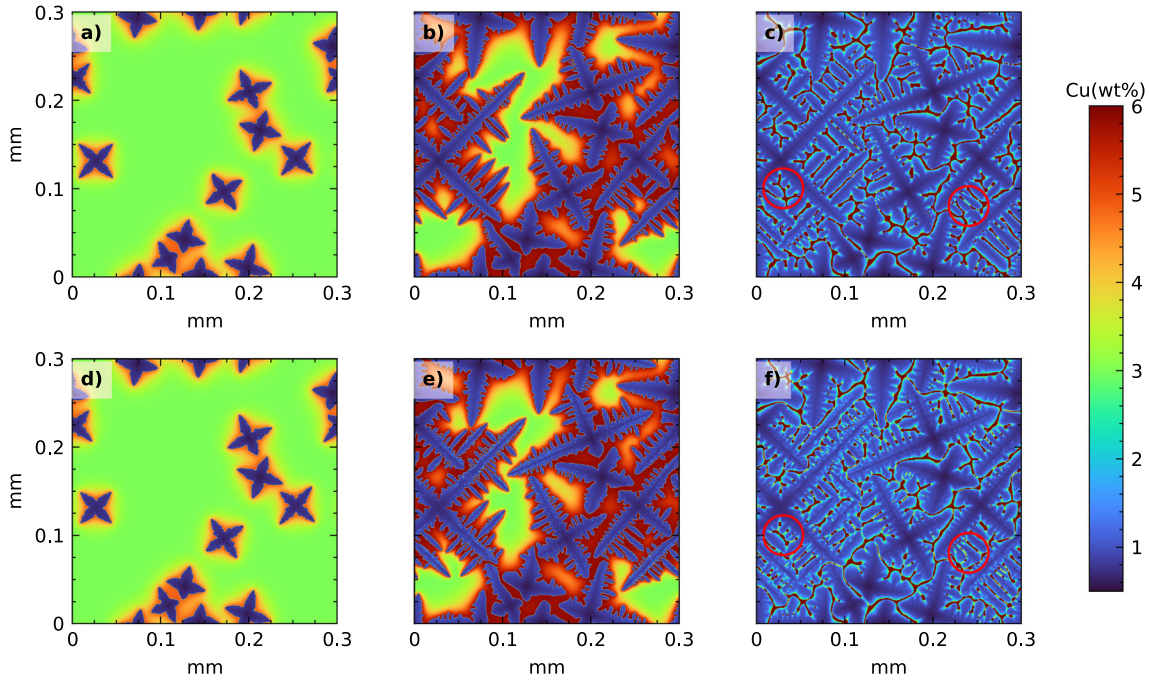

Fig. 7—The concentration profiles at time 0.10 s (*a*, *d*), 0.15 s (*b*, *e*) and 1.0 s (*c*, *f*) in the multi-dendritic solidification simulations with (a through c) and without (d through f) the mass balance correction.

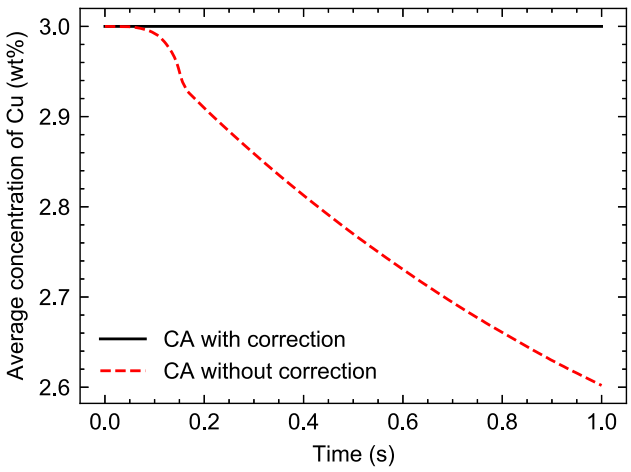

Fig. 8—The evolution of the average concentration in the multi-dendritic solidification simulations with and without the mass balance correction.

the equilibrium liquid concentration. Thus, the interface concentration tends to increase with decreasing temperature over numerous time steps. In this case, the sign of concentration change Δc of the interface cell is positive, which leads to a loss of the solutes. With the solutes artificially lost, the solid fraction in the simulation without the mass balance correction (Figure 7(f)) is thus larger than the solid fraction in the simulation with the mass balance correction (Figure 7(c)).

The results from the CA simulations are then compared with the Scheil–Gulliver equation. The Scheil–Gulliver equation[40] describes the solute redistribution during solidification of an alloy in a continuously cooling condition. It assumes the diffusion within the liquid is infinitely fast and the interface is always at a thermodynamic equilibrium. The liquid concentration c^{l} can be obtained as a function of the solid fraction of the system f_{s}[40]

$$c^{\mathrm{l}} = c^{0}(1 - f_{\mathrm{s}})^{k-1}. \quad [39]$$

The relation between temperature T and the system solid fraction f_{s} is given by

$$T = T_0 + m\left(c^{0}(1 - f_{\mathrm{s}})^{k-1} - c^{0}\right). \quad [40]$$

Note that the Scheil–Gulliver equation assumes a thermodynamic equilibrium at the interface, whereas the current CA model considers a non-equilibrium effect with the kinetic undercooling. However, due the employment of a relatively large kinetic coefficient ($1 \times 10^{-3}\,\mathrm{m\,s^{-1}\,K^{-1}}$), the interface concentration is close to the equilibrium liquid concentration, as shown in Figure 5. Moreover, the solidification condition at the late stage can be approximated by the Scheil–Gulliver solidification condition due to the small growth velocity. In this case, the current CA model is comparable with the Scheil–Gulliver equation.

The relations between temperature T and the solid fraction f_{s} in the CA simulations with and without the mass balance correction are compared with the Scheil–Gulliver equation in Figure 9. In the early stage of the solidification, the solid fraction predicted by the different CA simulations differs from the solid fraction predicted by the Scheil–Gulliver equation. This is because the liquid diffusion coefficient in the CA simulation is finite, which leads to a hump in the concentration profile in front of the interface. At the

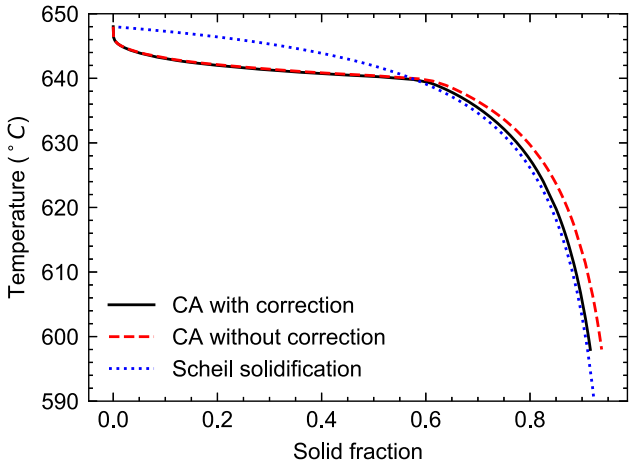

Fig. 9—The relation between temperature and the solid fraction in the multi-dendritic solidification simulations with and without the mass balance correction compared with the Scheil–Gulliver equation.

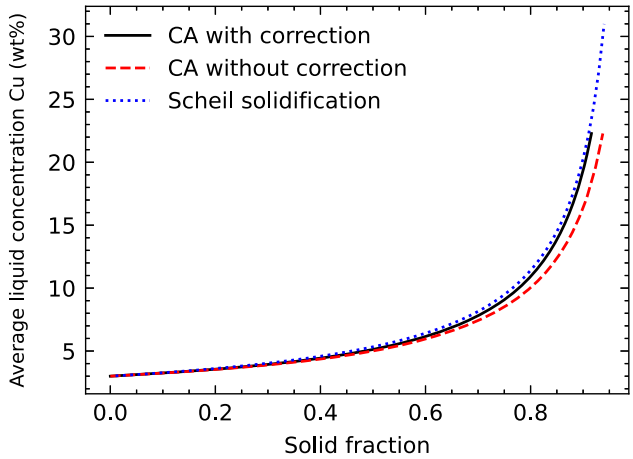

Fig. 10—The relation between the average liquid concentration and the solid fraction in the multi-dendritic solidification simulations with and without the mass balance correction compared with the Scheil–Gulliver equation.

interface, the Cu concentration is smaller than but close to the equilibrium concentration due to the non-equilibrium effect of the kinetic undercooling ΔT_k. In the liquid far from the interface, the Cu concentration is much smaller than the equilibrium concentration. In this case, the average Cu concentration in the liquid is smaller than the equilibrium concentration, which leads to a solid fraction smaller than the equilibrium solid fraction. As solidification proceeds, coarsening and coalescence of dendrites occurs. Liquid remains in the inter-dendritic region and the diffusion distance is much smaller, which decreases the required diffusion time. Moreover, the non-equilibrium effect introduced by the kinetic undercooling ΔT_k is negligible, as the interface velocity is very small in this stage. In this case, the diffusion condition can be approximated by the Scheil–Gulliver condition. The solid fraction in the CA simulation approaches the equilibrium solid fraction predicted by the Scheil–Gulliver equation. At a temperature of 600 °C, the solid fraction in the CA simulation with the mass balance correction is close to the prediction of the Scheil–Gulliver equation, while the solid fraction in the CA simulation without the mass balance correction is around 3 wt pct larger. This agrees with the observations in Figure 7. The reason is that the solutes within the CA simulation without the mass balance correction are artificially lost, which leads to a larger equilibrium solid fraction. Moreover, the relations between the average liquid concentration and the solid fraction f_s in the CA simulations and predicted by the Scheil–Gulliver equation are given in Figure 10. For the same solid fraction, the liquid concentration in the CA simulation without the mass balance correction is smaller than the liquid concentration in the CA simulation with the mass balance correction and the Scheil–Gulliver calculation. This shows that solute segregation may be underestimated by the CA simulation without the mass balance correction.

IV. CONCLUSIONS

In this work, a non-equilibrium CA model has been developed. The growth velocity is calculated as a linear function of the kinetic undercooling. To construct a CA model which is independent of the mesh size, a new diffusion term is proposed to handle the diffusion between the interface cells and the liquid cells, in which the diffusion out of an interface cell is calculated by summing up the projections of diffusion fluxes onto the interface normal direction. With the new diffusion term, the stable tip velocity is proportional to the diffusion flux in the interface normal direction, which agrees with Nastac's equation for growth velocity calculation. Moreover, it minimize the overestimation of tip velocity when the tip cell is surrounded by three liquid cells in its von Neumann neighborhood and improves the behavior in mesh size convergence tests.

The developed CA model has been employed to simulate single dendritic growth under different undercooling conditions. The simulated stable tip velocity agrees well with the prediction of a modified KGT model. With increasing kinetic mobility, the stable tip velocity tends to converge and the interface concentration at the dendrite tip approaches the equilibrium liquid concentration.

Moreover, the influence of the virtual liquid cell assumption has been studied in a multi-dendritic solidification simulation. In a continuously cooling solidification simulation, solutes are artificially lost due to the virtual interface cell assumption, which leads to a decrease in the average concentration. The mass balance error is removed by redistributing the lost solutes to neighboring liquid cells. With the mass balance correction, the average concentration remains constant throughout the simulation. Comparison with the prediction by the Scheil–Gulliver equation shows a good agreement in the final stage of the solidification.

ACKNOWLEDGMENTS

This research was carried out under Project Number T17019g in the framework of the Research Program of the Materials Innovation Institute (M2i) (www.m2i.nl) supported by the Dutch Government.

CONFLICT OF INTEREST

On behalf of all the authors, the corresponding author states that there is no conflict of interest.

OPEN ACCESS

This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

REFERENCES

1. K. Reuther and M. Rettenmayr: *Comput. Mater. Sci.*, 2014, vol. 95, pp. 213–20.
2. R. Kobayashi: Modeling and numerical simulations of dendritic crystal growth. *Physica D*, 1993, vol. 63, pp. 410–23. https://linkinghub.elsevier.com/retrieve/pii/016727899390120P.
3. S. Gurevich, A. Karma, M. Plapp, and R. Trivedi: *Phys. Rev. E*, 2010, vol. 81, p. 011603.
4. J. Liu, H.P. Duarte, and S. Kou: Evidence of back diffusion reducing cracking during solidification. *Acta Mater.*, 2017, vol. 122, pp. 47–59. https://www.sciencedirect.com/science/article/pii/S1359645416307364.
5. M. Ohno: *Phys. Rev. E*, 2012, vol. 86, p. 051603.
6. H. Yin and S. Felicelli: *Acta Mater.*, 2010, vol. 58, pp. 1455–65.
7. M. Rolchigo and R. LeSar: *Comput. Mater. Sci.*, 2019, vol. 163, pp. 148–61.
8. X. Ao, H. Xia, J. Liu, and Q. He: *Mater. Des.*, 2020, vol. 185, p. 108230.
9. Y. Zhang and J. Zhang: *Addit. Manuf.*, 2019, vol. 28, pp. 750–65.
10. F. Gibou, R. Fedkiw, R. Caflisch, and S. Osher: *J. Sci. Comput.*, 2003, vol. 19, pp. 183–99.
11. Y.T. Kim, N. Goldenfeld, and J. Dantzig: *Phys. Rev. E*, 2000, vol. 62, p. 2471.
12. L. Tan and N. Zabaras: *J. Comput. Phys.*, 2007, vol. 221, pp. 9–40.
13. L. Nastac: *Acta Mater.*, 1999, vol. 47, pp. 4253–62.
14. W. Wang, P.D. Lee, and M. Mclean: *Acta Mater.*, 2003, vol. 51, pp. 2971–87.
15. M. Zhu, W. Cao, S. Chen, C. Hong, and Y. Chang: *J. Phase Equilib. Diffus.*, 2007, vol. 28, pp. 130–38.
16. L. Beltran-Sanchez and D.M. Stefanescu: *Metall. Mater. Trans. A*, 2003, vol. 34A, pp. 367–82.
17. L. Beltran-Sanchez and D.M. Stefanescu: *Metall. Mater. Trans. A*,, 2004, vol. 35A, pp. 2471–85.
18. J. Lipton, M. Glicksman, and W. Kurz: *Mater. Sci. Eng.*, 1984, vol. 65, pp. 57–63.
19. S.C. Michelic, J.M. Thuswaldner, and C. Bernhard: *Acta Mater.*, 2010, vol. 58, pp. 2738–51.
20. X. Zhang, J. Zhao, H. Jiang, and M. Zhu: *Acta Mater.*, 2012, vol. 60, pp. 2249–57.
21. M. Zhu and D. Stefanescu: *Acta Mater.*, 2007, vol. 55, pp. 1741–55.
22. S. Pan and M. Zhu: *Acta Mater.*, 2010, vol. 58, pp. 340–52.
23. H. Yin, S.D. Felicelli, and L. Wang: *Acta Mater.*, 2011, vol. 59, pp. 3124–36.
24. R. Trivedi and W. Kurz: *Int. Mater. Rev.*, 1994, vol. 39, pp. 49–74.
25. A. Burbelko, D. Gurgul, W. Kapturkiewicz, J. Początek, and M. Wróbel: *Arch. Foundry Eng.*, 2011, vol. 11, pp. 13–18.
26. S. Geng, P. Jiang, Y. Ai, R. Chen, L. Cao, C. Han, W. Liu, and Y. Liu: *J. Laser Appl.*, 2018, vol. 30, p. 032406.
27. Y. Zhao, R. Qin, and D. Chen: *J. Cryst. Growth*, 2013, vol. 377, pp. 72–77.
28. K. Reuther and M. Rettenmayr: *Comput. Mater. Sci.*, 2019, vol. 166, pp. 143–49.
29. L. Wei, Y. Cao, X. Lin, M. Wang, and W. Huang: *Comput. Mater. Sci.*, 2019, vol. 156, pp. 157–66.
30. S. Luo and M.Y. Zhu: *Comput. Mater. Sci.*, 2013, vol. 71, pp. 10–18.
31. M. Marek: *Physica D*, 2013, vol. 253, pp. 73–84.
32. C. Gandin and M. Rappaz: *Acta Mater.*, 1997, vol. 45, pp. 2187–95,.
33. W. Tan, N.S. Bailey, and Y.C. Shin: A novel integrated model combining Cellular Automata and Phase Field methods for microstructure evolution during solidification of multi-component and multi-phase alloys. *Comput. Mater. Sci.*, 2011, vol. 50, pp. 2573–85. https://linkinghub.elsevier.com/retrieve/pii/S0927025611002059.
34. S. Liu, K. Hong, and Y.C. Shin: *Comput. Mater. Sci.*, 2021, vol. 192, p. 110405.
35. S. Popinet: *J. Comput. Phys.*, 2009, vol. 228, pp. 5838–66.
36. M.A. Zaeem, H. Yin, and S.D. Felicelli: *Appl. Math. Model.*, 2013, vol. 37, pp. 3495–3503.
37. W. Kurz, B. Giovanola, and R. Trivedi: *Acta Metall.*, 1986, vol. 34, pp. 823–30.
38. D. Sun, M. Zhu, S. Pan, and D. Raabe: Lattice Boltzmann modeling of dendritic growth in a forced melt convection. *Acta Mater.*, 2009, vol. 57, pp. 1755–67. https://linkinghub.elsevier.com/retrieve/pii/S1359645408008884.
39. J. Ramirez and C. Beckermann: Examination of binary alloy free dendritic growth theories with a phase-field model. *Acta Mater.*, 2005, vol. 53, pp. 1721–36. https://linkinghub.elsevier.com/retrieve/pii/S135964540400758X.
40. E. Scheil: *Int. J. Mater. Res.*, 1942, vol. 34, pp. 70–72.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.