A High-efficiency Virtual Submesh Cellular Automata Method for Solidification Simulation with Low Mesh Anisotropy

Ling SHI,1) Songzhe XU,1) Heyu LU,1) Chaoyue CHEN,1) Sansan SHUAI,1) Tao HU,1) Andrew KAO,2) Jiang WANG1)* and Zhongming REN1)

1) State Key Laboratory of Advanced Special Steels, School of Materials Science and Engineering, Shanghai University, Shanghai, China.
2) School of Computing and Mathematical Sciences, University of Greenwich, London, United Kingdom.

(Received on March 8, 2022; accepted on April 9, 2022)

The cellular automata (CA) method has been widely used in microstructure simulation during solidification, but it is time-consuming when used for multiple orientations and large-scale computation. In this work, a novel high-efficiency virtual submesh cellular automata (VISCA) method is proposed to reduce the mesh-induced artificial anisotropy. This method conceptually divides the mesh into a series of submeshes by giving a unique index to each CA cell. The proposed VISCA method results in a good accuracy in all orientations using uniform mesh, while an aggressive time stepping can be used. Compared with the previous CA models, the VISCA method exhibits a significant reduction in computational cost and improvement in computational efficiency. The nucleation is simulated based on a continuous nucleation model, and the Kurz-Giovanola-Trivedi (KGT) model is used to simulate the growth of the grains nucleated in the bulk material. The proposed VISCA method is comprehensively validated using an analytical solution, a casting simulation, and a directional solidification experiment. A quantitative comparison with the experiment in the directional solidification example shows that the difference in position of Columnar-to-Equiaxed Transition (CET) between the simulation and the experiment is within 1 mm while the length of directional solidification is 60 mm.

KEY WORDS: grain growth; virtual submesh cellular automata method; artificial anisotropy; solidification simulation; computational efficiency.

1. Introduction

The microstructure of an alloy has a significant influence on its mechanical and functional properties. The microstructure varies according to different manufacturing processes such as welding, casting, and additive manufacturing. A review of the solidification microstructure by Boettinger et al.1) describes that the material microstructure is a strategic link among its processing, properties, and performance. Thus, the microstructure simulation becomes the critical point in the prediction of mechanical and functional properties of the produced parts. In addition to experiments, the dynamic microstructural evolution can be obtained through numerical simulation with the development of computing technology.

As summarized by Zinovieva et al.,2) the most popular methods for the simulation of microstructure evolution during solidification are phase-field (PF) and cellular automata (CA). The PF method3) is an approach that can solve different kinds of problems about phase interface evolution. For the solidification simulation, the PF method performs well with the reproduction and prediction of the microstructure evolution at the microscale.4) Based on the work of kobayashi,5) Karma proposed a quantitative Phase field model for pure materials6,7) and binary alloys8,9) by thin interface analysis. The computational efficiency extremely increased compared to the previous models due to the extended interface thickness. However, due to the enormous demands of computational effort within a mesoscale domain (several dendrites), it is difficult to simulate the microstructural evolution at a larger scale, such as a weld pool or an ingot.

The CA method is known as another well-developed mathematical method for simulating metal solidification. It is commonly used in macroscale and mesoscale solidification simulations. In mesoscale, Beltran-Sanchez and Stefanescu10,11) proposed a CA model without using the KGT (Kurz et al.12) in 1986) model. They introduced the sharp interface model to control the phase boundary instead of integrating the tip velocity to compute the length of the primary dendrite. Then, Zhu and Stefanescu13) compared
the Lipton-Glicksman-Kurz (LGK) model (Lipton et al. 14) in 1984) with the virtual front tracking method, which also introduced the sharp interface model in the CA method. The virtual front tracking method shows a good agreement with the LGK model on the dendritic morphology in 2D and 3D simulations. Zhang and Sun 20 coupled the same CA method with the Lattice Boltzmann method to simulate the microporosity formation and dendrite growth during solidification. Wei 16 introduced a height function method to improve the calculation accuracy of interface curvature. Jelinek and Felicelli 27 coupled the CA method with the Lattice Boltzmann method to simulate dendritic solidification under forced convection and the computer program was parallelized using the Message Passing Interface (MPI) technique. These CA models mainly focus on mesoscale simulations, and they are challenging to be extended to larger scales. On a larger scale, an original CA method was first proposed in 1993 to reproduce the different preferred growth orientations during nucleation in an isothermal environment by Rappaz and Gandin. 18 It was realized that the squared mesh would misdirect the primary dendrite orientation resulting in artificial anisotropy, and it is necessary to correct this misdirection after a few computation steps. The inherent initiator of anisotropy is induced by the neighboring cell capture method coupled with the computation mesh. The spreading speed is anisotropic in Von Neumann and Moore neighboring cells, and they will twist the primary dendrite orientation to 0° and 45°, respectively. More details of this anisotropy can be found in the article of Marek. 19 In addition, to address the situation of a non-isothermal environment, Rappaz et al. 20 used a finite element (FE) method to compute the temperature field in the domain. The FE method splits the domain into elements, and there are multiple CA cells located in every single element. Because of the coarser mesh density of the heat transfer simulation, larger scales can be reproduced. Their CA–FE model deeply influenced the solidification simulation in the following decades. Ishida et al. 21 used Rappaz’s model investigate and optimum pouring temperature of the molten for obtaining the fine grain structure during high Mn steel casting. Wang et al. 22 investigated columnar to equiaxed transition in free-cutting steel based on the CA–FE model. Zhu et al. 25 investigated the effect of solidification pressure on the compactness degree of high nitrogen steel with the CA–FE model. Zhu et al. 24 proposed a modified CA method in 2001, in which the temperature field and the concentration field were computed using a finite-difference (FD) method. Zhu considered the curvature undercooling by computing the local solid fraction at the solid-liquid interface. The curvature undercooling is considered in the computation of tip dendrite growth velocity using the KGT model. However, in their work, the correction of the mesh anisotropy was still not considered, and it twisted all the dendrites to 0° or 45°.

In recent years, various researchers have contributed to developing the CA method proposed by Rappaz to reduce anisotropy. For instance, Luo et al. 25 developed the CA method by proposed a modified decentered square growth algorithm to reduce the mesh anisotropy. Zhan et al. 26 reported a multilayer mesh method to simulate the dendritic growth with different orientations. This approach is highly time-consuming with complicated operations due to the usage of multilayer mesh. To improve the computational efficiency, Zinovieva et al. 27 designed a modified technique to realize the correction in a uniform mesh of squared cells. They applied the modified method for predicting the grain structure evolution of 316L austenitic stainless steel under selective laser melting process. 27 The simulation results are in good qualitative agreement with experimental data. However, the distance to trigger the next cell is not constant in their algorithm, and some cells in specific orientations will be triggered at a longer distance. The computational accuracy of the CA method is relative to the maximum value of trigger distance among all different orientations, while the computational efficiency is relative to its minimum value. As a result, the balance between accuracy and efficiency is a bottleneck to the simulations in reducing anisotropy, while scarce attention has been paid to this aspect.

Based on the CA model proposed by Rappaz and Gandin, 18,20 a high-efficiency virtual submesh cellular automata (VISCA) method is developed in this work. The uniform mesh of squared cells is used with the same computational accuracy for each growth direction. The CA mesh is conceptually divided into a series of submeshes by giving a unique index to each CA cell. The continuous nucleation model and the KGT model are employed for the simulation of nucleation and crystal growth.

The paper is outlined as follows. Section 2 introduces the heat transfer, nucleation and grain growth formulations used in the VISCA method. The method of generating submeshes is explained in this section as well. Section 3 describes the numerical implementation of the VISCA method and illustrates the neighboring cell capture method during grain growth in detail. Section 4 validates this model using analytical, numerical and experimental results from different solidification cases, and the substantial efficiency improvement is also illustrated in this section. Section 5 draws the conclusion and points out future research directions.

2. Model Description

The solidification process consists of nucleation and grain growth. According to the previous work by Rappaz and Gandin, 18,20 there are three determining factors for the final solidified microstructure: (i) the heterogeneous nucleation on the surface and within the molten metal, (ii) the crystallographic orientation generated randomly during nucleation, and (iii) the dendrite tip growth velocity, which is determined by local undercooling. The two-dimensional solidification process is schematically illustrated in Fig. 1, where θ is the angle between the preferred crystallographic direction and the global x-axis. The dotted line represents the solid/liquid interface.

2.1. Virtual Submesh

Because of the mesh anisotropy caused by squared cells, the simulation of grains growth at different directions is unrealizable in a single uniform mesh. In this work, as is shown in Fig. 2(a), the global mesh is conceptually divided into a series of virtual submeshes with different orientations, and the grain is located within the submesh with an orientation parallel to it. For the simulation of a crystal that always grows along with (100) orientation (such as nickel-base
superalloy and aluminum-silicon alloy), the growth anisotropy (not the mesh anisotropy) is fourfold symmetrical in the two-dimensional case, and the nucleation orientation can be expressed within \((0, \pi/2)\). As shown in Fig. 2(a) (i)–(vi), six classes of virtual submeshes are able to realize 24 classes of nucleation orientation within \((0, 2\pi)\). After the nucleation calculation, a random number \(\theta\) indicating the crystal orientation degree continuously that ranges from 0 to \(\pi/2\) is given to a series of cells (nucleation points) in the liquid. A random number is sorted into six types of virtual submeshes by a classification rule shown in Fig. 2(b). The submesh is then generated simultaneously around the nucleation point, and more details are provided in Section 3.

The virtual submesh step length \(\Delta X\) is larger than the global mesh step length \(\Delta x\), and the ratio between them is defined as \(\epsilon\). The \(\epsilon\) of each virtual submesh is independent, and the maximum and minimum values are defined as \(\epsilon_{\text{max}}\) and \(\epsilon_{\text{min}}\), respectively. The difference between \(\epsilon_{\text{max}}\) and \(\epsilon_{\text{min}}\) is \(\epsilon_{\text{delta}}\), and the average value of \(\epsilon\) is \(\epsilon_{\text{mean}}\). On one hand, for a constant \(\Delta x\), a larger \(\epsilon_{\text{max}}\) will obviously decrease the computational accuracy due to fewer cells in the same space. On the other hand, a smaller \(\epsilon_{\text{min}}\) will intensively increase the lower limit of the time step, which can decrease the computational efficiency. When the \(\epsilon_{\text{delta}}\) equals zero (the computational accuracy is the same at different orientations), the resultant computational efficiency will be higher. In the work of Zinovieva\(^2\) and Zhan,\(^2\) the trigger distance between cells in a direction can be several times greater than that in the other directions. In other words, the computational accuracy in their algorithm is uneven for every grain orientation. Consequently, the lower limit of the time step has to be set to a lower level, which is already unnecessary in some directions.

The CA method can simulate with a limited number of orientations \(n_{2\pi}\). A larger value of \(n_{2\pi}\) will increase the magnitude of \(\epsilon_{\text{mean}}\). Furthermore, a smaller \(\epsilon_{\text{delta}}\) will also increase the magnitude of \(\epsilon_{\text{mean}}\). Obviously, the size of the global matrix is positively correlated with \(\epsilon_{\text{mean}}\), and a more prominent global matrix will cost more CPU and memory resources. To maintain a balance of reality and computational efficiency, the number of orientations \(n_{2\pi}\) should be more significant, and the values of \(\epsilon_{\text{delta}}\) and \(\epsilon_{\text{mean}}\) should be smaller. Because the 24 classes of orientation are sufficient to reproduce the real solidification process, the \(n_{2\pi}\) is set to 24. The value of \(\epsilon_{\text{mean}}\) is set to approximately 6 to ensure that the \(\epsilon_{\text{delta}}\) is always smaller than one under the current \(n_{2\pi}\) in the VISCA method. The precise value of \(\epsilon\) at every orientation is shown in Fig. 2(a).

### 2.2. Heat Transfer Formulation

In this work, the temperature field is solved using a finite difference method. The governing equation for the heat transfer process can be expressed as follows:

\[
\rho C_p \frac{\partial T}{\partial t} = \nabla \cdot (\lambda \nabla T) + \rho L f_s + \phi, \quad \text{................ (1)}
\]

where \(T\) is the temperature, \(t\) is the time, \(\lambda\) is the thermal

---

![Fig. 1. Grains growth with different orientations. (Online version in color.)](image1)

![Fig. 2. (a) Virtual submeshes for different orientations and (b) its classification rule. The ratio between the virtual submesh step length of \(\Delta X\) and the global mesh step length of \(\Delta x\) is shown under the submesh in Fig. 2(a). The continuous orientation degree \(\theta\) is sorted into six classes of submesh by a classification rule shown in Fig. 2(b).](image2)
Material properties are interpolated using \( f_s \) activated for nucleation. The nucleation density increment between \( t_n \) be considered as a function of local undercooling \( \Delta T_b \) and liquid phases. The boundary condition is specified as 

where \( \Delta T \) is expressed as 

To improve the computation stability of explicit schemes, the finite difference form in the enthalpy method by Voeller\(^1\) can be employed for solving Eq. (1). A coarser mesh can be used to solve heat equation for reducing computation cost, and intergrid transfer for interface computed from CA (by averaging) and temperature field computed from heat equation (by interpolation) is required. In this work, the grid size for solving heat equation is set to be ten times as large as that of the virtual submesh.

### 2.3. Nucleation Formulation

Following the pragmatic approach used by Rappaz et al.\(^{18}\) the continuous nucleation distribution can be expressed as follows:

\[
n(\Delta T) = \frac{\eta_{\text{max}}}{\Delta T_\sigma \sqrt{2\pi}} e^{-\left(\frac{(\Delta T - \Delta T_\sigma)^2}{2\Delta T_\sigma^2}\right)} d(\Delta T), \quad \cdots(2)
\]

where \( n \) is the nucleation density, \( \Delta T \) is the undercooling, \( \eta_{\text{max}} \) is the maximum nucleation density, \( \Delta T_\sigma \) is the mean undercooling, and \( \Delta T_{\sigma} \) is the standard deviation, and the maximum nucleation density in 2D can be derived from the work by Zinovieva\(^2\) as

\[
\eta_{\text{max}} = \left(\frac{\pi}{6}\right)^{\frac{3}{2}} (\eta_{\text{max2D}})^{\frac{3}{2}}.
\]

The nucleation density increment between \( t^n \) and \( t^{n+1} \) time step can be expressed as

\[
\Delta n = n(\Delta T^{n+1}) - n(\Delta T^n).
\]

For a nonisothermal situation, the nucleation distribution is not uniform in space. By solving the Eq. (2), the nucleation probability function (2D version) is constructed for every global mesh cell, which is given as follows:

\[
P_n = \Delta n \cdot S_{\text{CA}} = \frac{n_{\text{max2D}} \cdot \Delta x^2}{2} \left[ \text{erf}\left(\frac{\Delta T_n - \Delta T^n}{\sqrt{2} \Delta T_\sigma}\right) - \text{erf}\left(\frac{\Delta T_n - \Delta T^{n+1}}{\sqrt{2} \Delta T_\sigma}\right) \right],
\]

where \( S_{\text{CA}} \) is the area of a global mesh equal to \( \Delta x^2 \), and \( P_n \) is the nucleation probability for correlating cells at the current time step. To reduce the number of random numbers, only correlating cells (\( 0 \leq \Delta T_n < \Delta T^{n+1} \leq \Delta T_n + \Delta T_\sigma \)) are activated for nucleation.

### 2.4. Grain Growth

The growth velocity of the primary dendrite tip \( V_{\text{tip}} \) can be considered as a function of local undercooling \( \Delta T \). The equations from the LGK model\(^{19}\) and KGT model\(^{12}\) are used to gain the relationship between \( \Delta T \) and \( V_{\text{tip}} \). The following equations\(^{12}\) link the \( V_{\text{tip}} \) and the Peclet number \( P_c \):

\[
V_{\text{tip}} = \frac{2k}{P_c} + \frac{m_l C_0 (1 - k) \xi}{D_l [1 - (1 - k) \cdot Iv(P_c)]} + G = 0 \quad \cdots(4)
\]

\[
\xi = 1 - \frac{2k}{\sqrt{1 + \left(\frac{2\pi}{P_c}\right)^2} - 1 + 2k}, \quad \cdots(5)
\]

where \( \Gamma \) is the Gibbs–Thomson coefficient, \( D_l \) is the liquid diffusion coefficient, \( k \) is the partition coefficient, \( m_l \) is the liquidus slope, \( C_0 \) is the initial concentration, \( P_c \) is the temperature gradient, and \( Iv(P_c) \) is the Ivanov function of concentration Peclet number expressed as follows:

\[
Iv(P_c) = \frac{V_{\text{tip}}}{2D_l}, \quad \cdots(6)
\]

By solving Eq. (2) with a given value of \( V_{\text{tip}} \), the concentration Peclet number \( P_c \) can be calculated. The following equations\(^{13}\) link the \( P_c \) and the undercooling \( \Delta T \):

\[
\Delta T = m_l C_0 \left(1 - \frac{1}{1 - (1 - k) \cdot Iv(P_c)} \right) \frac{2\Gamma}{R} \quad \cdots(7)
\]

where \( R \) is the dendrite tip radius and Eq. (7) is satisfied only when the liquidus slope and partition coefficient are constant. The undercooling \( \Delta T \) is solved by substituting \( P_c \) (which is calculated above) into Eqs. (7) and (8). By solving these equations with parameters in Table 2, the relationship between \( V_{\text{tip}} \) and \( \Delta T \) is fitted in Eq. (9) and

### Table 1. Offset address for virtual submesh expansion.

<table>
<thead>
<tr>
<th>Submesh index</th>
<th>Orientation degree</th>
<th>Offset address</th>
</tr>
</thead>
<tbody>
<tr>
<td>( i )</td>
<td>( (0, \pi/24) \cup (11 \pi/24, \pi/2) )</td>
<td>( (6,0) \cdot (0,6) \cdot (0,-6) )</td>
</tr>
<tr>
<td>( ii )</td>
<td>( (\pi/24, \pi/8) )</td>
<td>( (6,-2) \cdot (-6,-2) \cdot (-6,2) )</td>
</tr>
<tr>
<td>( iii )</td>
<td>( (\pi/8, 5\pi/24) )</td>
<td>( (5,3) \cdot (-5,3) \cdot (-5,-3) \cdot (3,-5) )</td>
</tr>
<tr>
<td>( iv )</td>
<td>( (5\pi/24, 7\pi/24) )</td>
<td>( (4,4) \cdot (-4,4) \cdot (-4,-4) \cdot (4,-4) )</td>
</tr>
<tr>
<td>( v )</td>
<td>( (7\pi/24, 3\pi/8) )</td>
<td>( (3,5) \cdot (-3,5) \cdot (-3,-3) \cdot (3,3) )</td>
</tr>
<tr>
<td>( vi )</td>
<td>( (3\pi/8, 11\pi/24) )</td>
<td>( (2,6) \cdot (-2,-6) \cdot (2,-6) \cdot (6,-6) )</td>
</tr>
</tbody>
</table>

### Table 2. Material properties of Al−7mass%Si used in the simulation.

<table>
<thead>
<tr>
<th>Property</th>
<th>Variable</th>
<th>Unit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>( \rho )</td>
<td>kg/m(^3)</td>
<td>2 600</td>
</tr>
<tr>
<td>Heat capacity</td>
<td>( C_p )</td>
<td>J/(kg · K)</td>
<td>998</td>
</tr>
<tr>
<td>Solid thermal conductivity</td>
<td>( \lambda_s )</td>
<td>W/(m · K)</td>
<td>170</td>
</tr>
<tr>
<td>Liquid thermal conductivity</td>
<td>( \lambda_l )</td>
<td>W/(m · K)</td>
<td>66</td>
</tr>
<tr>
<td>Liquidus temperature</td>
<td>( T_l )</td>
<td>K</td>
<td>930</td>
</tr>
<tr>
<td>Mean undercooling</td>
<td>( \Delta T_v )</td>
<td>K</td>
<td>8</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>( \Delta T_\sigma )</td>
<td>K</td>
<td>4</td>
</tr>
<tr>
<td>Maximum nucleation density</td>
<td>( n_{\text{max}} )</td>
<td>m(^3)/m(^2)</td>
<td>9 \times 10(^{20})</td>
</tr>
<tr>
<td>Liquidus slope</td>
<td>( m_l )</td>
<td>K/kg wt%</td>
<td>-6</td>
</tr>
<tr>
<td>Initial concentration</td>
<td>( C_0 )</td>
<td>wt%</td>
<td>7</td>
</tr>
<tr>
<td>Partition coefficient</td>
<td>( k )</td>
<td>( 1 )</td>
<td>0.117</td>
</tr>
<tr>
<td>Gibbs-Thomson coefficient</td>
<td>( \Gamma )</td>
<td>K · m</td>
<td>9 \times 10(^{-8})</td>
</tr>
<tr>
<td>Diffusion coefficient</td>
<td>( D_l )</td>
<td>m(^2)/s</td>
<td>1 \times 10(^{-9})</td>
</tr>
</tbody>
</table>

\(^1\) To improve the computation stability of explicit schemes, the finite difference form in the enthalpy method by Voeller\(^{20}\) can be used for computing derivatives in Eq. (1).
plotted in Fig. 3.

\[ V_{\text{tip}} = 2.499 \times 10^{-5} \Delta T + 1.551 \times 10^{-6} \Delta T^2 + 5.606 \times 10^{-7} \Delta T^3 \]

\[ \text{..........................(9)} \]

2.5. Time Increment

There are two kinds of time increment used during the entire calculation. The CA time increment \( \Delta t \) is as follows:

\[ \Delta t = 0.25 \min \frac{\Delta X}{\max(V_{\text{tip}}) D_x} \]

\[ \text{..........................(10)} \]

where \( \max(V_{\text{tip}}) \) is the maximum value of the dendrite tip velocity of the entire computational domain. The global time increment \( \Delta t \) is the duration of one complete calculation cycle, including one cycle of temperature, nucleation, and growth simulations. The time increment for heat transfer is denoted as \( \Delta T \), and for explicit time stepping \( \Delta T \) can be expressed as follows:

\[ \Delta T = \frac{\Delta x_T^2}{4 \frac{\lambda}{\rho C_p}} \]

\[ \text{..........................(11)} \]

where \( \Delta x_T \) is the mesh step length for heat equation. Since a smaller time step has to be employed for explicit heat equation, the temperature field is solved \( r \) times during \( \Delta t \) to ensure synchronization in time. For the implicit time stepping, Eq. (10) is used, and the \( \Delta r_T \) is equal to \( \Delta t \).

3. Procedure of the Virtual Submesh Method

The computation procedure is illustrated in Fig. 4.

The first part is the simulation of the temperature field using an updated interface (i.e., values of \( f \)) computed from the previous time step. The second part of the simulation procedure is the nucleation calculation. The nucleation probability of each cell in the global mesh can be derived from Eq. (3) through the local undercooling \( \Delta T \). After rolling a random number, the new nucleation cell in the liquid at the current time step is selected with a random orientation \( \theta \). Then, a virtual submesh with the same orientation is constructed around the nucleated cell. The third part of the

Fig. 3. The relationship between \( V_{\text{tip}} \) and \( \Delta T \) is fitted by 1000 couples of discrete solutions. The value of \( V_{\text{tip}} \) ranges from 1 \( \text{um/s} \) to 0.1 \( \text{m/s} \) and densified around 1 \( \text{um/s} \). (Online version in color.)

Fig. 4. Calculation procedure of solidification simulation of the VISCA method. The calculation procedure consists of three parts: heat diffusion calculation, nucleation calculation, and grain growth calculation.
simulation procedure is the grain growth calculation. The virtual submesh around the nucleation cell expands along with the grain growth, as shown in Fig. 5, where $L_d$ is the primary dendrite length of the current cell (not the real one), and $\Delta X$ is the virtual submesh step length. After the nucleation, the growth state $L_d$ is stored in a specific matrix. At the beginning of the third part, the nucleation points on the global mesh will be led to three different ways according to the specific conditions:

(a) If the $L_d/\Delta X$ of the current cell is within the range of $(0, 1)$, the $V_{tip}$ can be obtained from Eq. (9). The growth of dendrite length at the current time step will be added to $L_d$ (as shown in Fig. 5(a)), which is the product between $V_{tip}$ and $\Delta t$.

(b) For $L_d/\Delta X$ larger than 1, the $L_d$ of the current cell will be set to $\Delta X$, and the adjacent cells of the inactivated state in virtual submesh ($a_1–a_4$ in Fig. 5(a)) will be activated, and the surplus of $L_d - \Delta X$ will be stored in the growth state $L_d$ matrix. Afterwards, a series of cells on the global matrix (the offset address of them are shown in Table 1) surrounded by four adjacent ones on the submesh will be given the same value as the central cell (the value is orientation degree $\theta$, which can be named as the cover skin (because the submesh is not discontinuous on the global mesh, the grain should be performed on global mesh). Then, the virtual submesh expands around the activated cells. The offset address for submesh expansion is shown in Table 1, Fig. 5(b).

(c) Once the submesh expanding into skin cells, the submesh stops expanding, and the cell will not trigger any adjacent cells. As shown in Fig. 6(a), the red grains are surrounded by the submesh cells $a_1–a_8$, and $a_1$ is found to expand into the skin cells of another grain. The submesh cells $b_1–b_7$ in Fig. 6(b) are generated from $a_1–a_8$, and the submesh cell $a_1$ will not trigger other cells. As shown in Fig. 6(c), the cell $a_1$ still can cover the red skin cells except for the blue cells.

After three parts of the calculation, the solid fraction $f_s$ will

---

**Fig. 5.** Schematic diagrams of VISCA growth and the generation of virtual submesh: (a) When $L_d$ of the central cell is smaller than $\Delta X$, $a_1–a_4$ are still not activated (b) When $L_d$ of the central cell is more significant than $\Delta X$, $a_1–a_4$ are activated, and the $L_d$ of them starts to grow. The white cells are global meshes, and the black ones are virtual submeshes. The gray ones are skin cells of the growing grain.

**Fig. 6.** Schematic diagrams of grain growth simulation between grains: (a)–(c) showing the submesh expansion process when the growing grain (in red) touches other grains (in blue or green). (Online version in color.)
be updated, which then determines the material properties and latent heat for the next time step.

4. Results and Discussion

4.1. Validation Examples

A numerical simulation of microstructure evolution during solidification (shown in Fig. 7) is produced by the VISCA method. The values next to the colorbar are the index of phase and orientation (-1 is for liquid and (0–π/2) is for solid). Because of the abundant successful simulation experiences and the <100> growth direction, the Al−7mass%Si alloy is used in this work for the simulation of grain growth.18,29) The material parameters used in the current simulation are shown in Table 2. The parameters of nucleation, diffusion and interfacial energy refer to the work by Zhu et al.29) The thermophysical parameters refer to the work by Liu et al.30,31)

The metal melt of Al−7mass%Si freeze in an atmosphere of 300 K, and the initial temperature of the melt is 1000 K. The computational domain is two-dimensional with a size of 10 mm × 10 mm. The global CA mesh size is 1 000 × 1 000 and the temperature mesh size is 100 × 100. The implicit algorithm is chosen for this case. With the rapid decrease in melt temperature, the nucleation and grain growth process originate almost simultaneously. The topography of crystal grains is visually similar to the cast experiment.32)

To further validate the VISCA algorithm, a simulation of directional solidification is implemented for comparison with the experiment and simulation published by Liu et al.30) The experiment is implemented in a low gravity environment where natural convection does not affect the nucleation and temperature field, and it is more suitable for validating the solidification simulation algorithm. The ‘FM1’ experiment in their work is simulated to verify the VISCA algorithm. The directional solidification experiment includes three cooling stages with different pulling velocity: 0.01 mm/s in stage I, 0.2 mm/s in stage II and 3 mm/s in stage III. The thermophysical and nucleation parameters used in the simulation are the same as the simulation work of Liu. The computational domain is two-dimensional with a size of 4 mm × 60 mm, and the global mesh size is 200 × 3 000. The vertical temperature field used in simulation is generated from the cooling curves measured at several evenly distributed locations (positions of TC6–TC12 are plotted in Fig. 8(a)) during experiment, and the horizontal
temperature field is considered to be uniform. Some points on the cooling curves (endpoints of stage I–stage III) are taken to generate simplified cooling curves by linear interpolation (plotted in Fig. 8(b)).

The simulation result is shown in Fig. 8(a) with a length scale and a colorbar for grain orientation (range from 0 to $\pi/2$). The position of Columnar-to-Equiaxed Transition (CET) is at 127 mm, which is the same as the experiment. The position of CET in simulation fits well with the experiment, and the grain growth velocity spans from 3.5 um/s to 200 um/s in this case, proving that the algorithm is able to reproduce the solidification process at both high and low growth velocities.

### 4.2. Reduction of Artificial Anisotropy

To assess the performance of reducing artificial anisotropy using the VISCA algorithm, the grains growth in an isothermal environment is simulated. The heterogeneous nuclei of six orientation angles $\theta_1$–$\theta_6(0, \pi/12, \pi/6, \pi/4, \pi/3, 5\pi/12)$ are uniformly located in the domain. The same case also implemented by ‘correction 2’ model of Zinovieva et al.2) The dimensionless computational parameters are shown in Table 3. To avoid the loss of the primary dendrite length, The time increment $\Delta t$ of VISCA is set to maximum allowed value of $0.25\Delta x / \max(V_{tip}) = 0.25$. The primary dendrite lengths can be measured by calculating the area of grains before any possible collision. The analytical solution of primary dendrite length is $y = x$ since all the grains are grown with a constant velocity 1. For VISCA algorithm with $\Delta t = 1.5$ in Figs. 9(a), 9(b), it is obvious that the dendrite lengths of different grains remain the same and all the curves converge to the analytical solution. The results show that the velocity loss of growth is avoided. For the previous method with $\Delta t = 0.25$ in Fig. 9(c), the simulation

| Table 3. Offset address for virtual submesh expansion. |
|---------------------------------|--------|-------|
| Computational parameter       | Variable | Value | Unit |
| Global mesh step length        | $\Delta x$ | 1     | 1    |
| Virtual submesh step length    | $\Delta X$ | $f(\theta) \approx 6\Delta x$ | 1    |
| Global mesh size m × n         | 900 × 1200 | 1     |
| Dendrite tip velocity $V_{tip}$| 1      | 1     |
| Time increment (VISCA) $\Delta t$ | $0.25\Delta x / \max(V_{tip}) = 1.5$ | 1    |
| Time increment (Previous) $\Delta t$ | $0.25\Delta x / \max(V_{tip}) = 0.25$ | 1    |

**Fig. 9.** Comparison of grain growth velocity difference among grains of different orientations between VISCA and previous algorithm: (a)–(b) The primary dendrite length of growing grains by VISCA algorithm with $\Delta t = 1.5$. (c)–(d) The primary dendrite length of growing grains by previous algorithm with $\Delta t = 0.25$. The heterogeneous nuclei of six orientation angles $\theta_1$–$\theta_6(0, \pi/12, \pi/6, \pi/4, \pi/3, 5\pi/12)$ are uniformly located in the domain. (Online version in color.)
results are similar to VISCA. The primary dendrite lengths in Fig. 9(d) also converge to the analytical solution. But the vibration frequencies of these curves are different, and it’s the reason why previous method must work with a smaller time increment. These numerical experiments illustrate that both two methods overcome artificial anisotropy, and the VISCA algorithm is able to work in a much bigger time increment $\Delta t$.

### 4.3. Computational Efficiency

In this section, the comparison of computational efficiency between the VISCA algorithm and the ‘correction 2’ model of Zinovieva\(^2\) is performed. The case used for comparison is the same as Section 4.2. The computational efficiency comparison is plotted in Fig. 10. The relationship between physical time and computation time is shown in Fig. 10(a). It is evident that the VISCA method can calculate longer physical time during the same computation time. In other words, the VISCA method has higher computational efficiency. The relative computational efficiency plotted in Fig. 10(b) is the ratio between the physical time of the VISCA algorithm and that of the previous algorithm under the same CPU time. The computational efficiency is dramatically increased by over 3.5 times using the VISCA algorithm in this work, and the computational efficiency can be simultaneously improved along with the grain growth. The efficiency advantage of the VISCA algorithm is more remarkable for the initially nucleated grains, which is approximately four times higher than that reported in previous work.\(^2\) The efficiency advantage will be continually improved with grain growth.

As stated above, the simulation requires correction to eliminate the velocity loss. The practical methods are increasing the mesh step length or reducing the time step. Although the chosen time step has a significant effect on the computation time, reducing the time step is still the only option to maintain the computational accuracy in the current scale. The final time step $\Delta t$ is 1.5 (dimensionless) and 0.25 (dimensionless), respectively, to eliminate the grain size inhomogeneity at different orientations. In the previous algorithm, the computation accuracy at each direction is different since the distinct trigger distance between cells. For example, the trigger distance for $\theta$ equals to $\arctan(3/5)$ is $\sqrt{34}$ times higher than that for $\theta$ of 0 in the previous algorithm. A smaller trigger distance can lead to more iterations. As a result, the direction with lower accuracy always waits for the direction with higher accuracy. It will waste lots of computational effort because the total accuracy equals the lowest.

The VISCA algorithm divides the global mesh into a series of submesh with approximately equal submesh step lengths, and thus, the upper limits of the theoretical time step for grains of different orientations are similar in value. Compared to the previous algorithm using irregular accuracy for different growth directions, the submesh method will save much computational effort for those grains of excessive computational accuracy when other grains are growing in low accuracy.

### 5. Conclusions

Based on the cellular automata method proposed by Rappaz,\(^18\) the current work provides a novel algorithm that can compute grain growth at different orientations without any direction correction. Using a virtually divided global mesh, a more aggressive time stepping can be achieved while the computational accuracy for different directions is retained the same. In traditional approaches, the computational efforts substantially increase as grains grow, eventually resulting in extremely slow computation. However, the proposed VISCA method solves this issue by reducing massive redundant time steps in specific directions and tremendously improves the overall efficiency. The simulation results from the VISCA method present good agreements with other analytical, simulation and experiment results. We anticipate deploying this framework to solidification simulations in steel casting and additive manufacturing in the future.
Acknowledgments

This work was jointly funded by the National Science and Technology Major Project “Aeroengine and Gas Turbine” of China (2017-VII-0008-0102), the “Sailing Program” of Shanghai Science and Technology Committee of China (19YF1415900), the National Natural Science Foundation of China (Nos. 52001191), the Shanghai Municipal Science and Technology Commission Grant of China (No. 20511107700, No. 21511103600), and Shanghai Rising-Star Program (20QA1403800).

REFERENCES