Microstructure Simulation of Aluminum Alloy Using Parallel Computing Technique

Weiming FENG, Qingyan XU and Baicheng LIU

Department of Mechanical Engineering, Tsinghua University, Beijing 100084, China.

(Received on December 20, 2001; accepted in final form on April 8, 2002)

The formation and evolution of the microstructure of casting are important research areas in the field of material science and engineering. The solidified microstructure of aluminum alloy was simulated by combining the CA (Cellular Automaton) model with macro heat transfer. A modified CA (MCA) model, which uses a more similar shape to the actual dendrite to describe the growth grain, was proposed and studied. Because of the huge computational capacity to simulate the microstructure of casting, a relevant parallel computing technique based on the serial arithmetic was developed, which can greatly improve the computing scale and efficiency and can also ensure the computing accuracy as well. The simulation results are compared with the experimental results and agreed quite well.

KEY WORDS: parallel computing technique; aluminum alloy; microstructure simulation; cellular automaton method.

1. Introduction

The final quality of castings depends on the microstructure formed during solidification. The microstructure simulation on the grain scale can predict the microstructure and hence the mechanical properties of casting as well. Therefore it can provide a reliable foundation for the process control and the improvement of the casting quality and has both significant theoretical importance and practical application potential.1–5)

In 1966, Oldfield6) pointed out that the solidification structure of the casting could be simulated and he made an attempt to do so. In the late 1980s, the Monte Carlo method was used for modeling the nucleation and growth of the grains.7–9) In the early 1990s, the Cellular Automaton (CA) model was presented, in which the physical mechanism of heterogeneous nucleation and grain growth was taken into account and a way similar to the deterministic method was introduced to deal with the distribution of nuclei. The CA method has certain physical bases and can reflect the influence of supercooling and solute concentration in a quantitative way.10–13) In recent years, the phase-field model, which based on the Ginzburg-Landau theory and can directly simulate the formation of microstructures, was presented and developed rapidly.14,15)

Because of the complication of solidification and the micron scale of grain size, the number of cells needed to simulate the microstructure of a casting is tremendous. For a common personal computer, the efficiency of serial computation is very low, and this kind of calculation even cannot finish due to the large computational capacity. Then, the parallel computing technique becomes an effective way to do large-scale calculation and hence it is necessary to study the parallel arithmetic for microstructure simulation. In this paper, an aluminum alloy sample is divided into many equal regions, different CPUs are allocated for different regions and then calculated together. After certain computing steps, the CPUs will send messages to each other and correct the computing errors. In this way, the computation and the microstructure simulation of a whole casting could be achieved.

2. Mathematical Model and Computing Method for Microstructure Simulation

The Cellular Automaton (CA) model was presented in the 1990s.11,13) The physical mechanism of heterogeneous nucleation and grain growth was taken into account in this method. In this paper the CA model was used with several modifications in order to improve the simulation microstructure.

2.1. Nucleation Model

2.1.1. Mathematical Model

The moving atoms in liquid will form a cluster as the temperature decreases below the liquidus. When the cluster grows to a certain size, it becomes a crystalline nucleus. This process is called nucleation. There are two basic nucleation mechanisms: homogeneous nucleation and heterogeneous nucleation. In practical production, homogeneous nucleation rarely or never takes place and heterogeneous nucleation occurs either on mold walls or on insoluble impurity particles (these particles can originally exist or be added into the molten metal later). There are two methods, which are instantaneous nucleation16) and continuous nucleation,6) to simulate the nucleation process. In this paper, the
continuous nucleation model is used. Based on Gauss Distribution, a continuous relation is assumed to be kept between the number of nuclei and the undercooling in this model. At a certain undercooling \( \Delta T \), the generated grain density \( n(\Delta T) \) can be obtained by integrating the Gauss Distribution as\(^{17}\):

\[
 n(\Delta T)=\int_0^{\Delta T} [1-f_s(\Delta T')] \frac{dn}{d(\Delta T')} d(\Delta T') \quad \text{............(1)}
\]

\[
 \frac{dn}{d(\Delta T')} = \frac{n_{\text{max}}}{\sqrt{2\pi \Delta T_s}} \exp \left[ -\frac{1}{2} \left( \frac{\Delta T' - \Delta T_N}{\Delta T_s} \right)^2 \right] \quad \text{............(2)}
\]

where \( f_s \) is the solid fraction, \( \Delta T_N \) is the mean nucleation undercooling, \( \Delta T_s \) is the standard deviation, \( \Delta T \) is the undercooling, \( n_{\text{max}} \) is the maximum grain density.

One of the assumptions of this model is that the appearance of nuclei does not take any time. The grain density increases in a slow-fast-slow way as the undercooling increases. When the undercooling is big enough, the grain density can reach the maximum \( n_{\text{max}} \).

2.1.2. Computing Model

Firstly, relatively bigger lattices are used to calculate the temperature field. Secondly, the bigger lattices are divided into numerous smaller and homogeneous CA cells. Finally, based on the CA model, the nucleation and growth computing is operated among these CA cells. All the CA cells are based on the CA model, the nucleation and growth computing result are in very good agreement.

2.2. Growth Model

2.2.1. Mathematical Model

After nucleation in the molten metal, the liquid atoms keep moving to the surface of nuclei as the temperature goes down. The crystals become bigger and bigger and the solid-liquid interfaces move toward the liquid. The growth of crystals mainly depends on the configuration in front of the advancing interface, temperature, concentration, and the nature of the alloy.

According to dendrite tip growth dynamics, the radius of dendrite tip \( R \) and growth speed \( v \) are controlled by these two equations\(^{18,19}\):

\[
 \Omega = \frac{C' - C_0}{C(1-K)} = Iv(P_e) \quad \text{............(5)}
\]

\[
 R = 2\pi \sqrt{\frac{\Gamma}{mG_\xi G}} \quad \text{............(6)}
\]

where \( Iv(P_e) \) is the Ivantsov function of \( P_e \), \( G_\xi \) is the solute gradient at dendrite tip, \( G \) is the temperature gradient of liquid in front of the dendrite tip, \( \xi \) is the function of Pécelt number. Pécelt (\( P_e \)) number is given by:

\[
 P_e = Rv/2D \quad \text{............(7)}
\]

where \( D \) is the diffusion coefficient.

2.2.2. Computing Model

The undercooling in front of dendrite consists of four parts during solidification:

| Table 1. Effect of the cell size on the number of new nuclei decided by \( r \leq p \). |
|-----------------|-----------------|-----------------|
| \( \delta n \) (m\(^2\)) | 1.21 \times 10^9 | 1.21 \times 10^9 | 1.21 \times 10^9 |
| \( V_c \) (m\(^3\)) | 25 \times 10^{-12} | 1 \times 10^{-12} | 25 \times 10^{-12} |
| \( \rho \) (g/cm\(^3\)) | 2.33 | 2.33 | 2.33 |
| Nuclear number (calculated by \( \delta n, V_c \)) | 121 | 121 | 121 |
| Nuclear number (decided by \( r \leq p \)) | About 120 | About 100 | About 90 |

(Al–7.0wt%Si; pouring temperature: 750°C; cooling rate: 10K/sec; calculation region: 1 mm \times 1 mm)
\[ \Delta T = \Delta T_c + \Delta T_t + \Delta T_k + \Delta T_r \] ..................(8)

where \( \Delta T_c \) is the constitutional undercooling, \( \Delta T_t \) is the thermal undercooling, \( \Delta T_k \) is the kinetic undercooling, and \( \Delta T_r \) is the curvature undercooling. For most of the metal alloys, \( \Delta T_t \) and \( \Delta T_k \) can be ignored under normal casting conditions, and therefore:

\[ \Delta T = T_{L}^{eq} + (c^*_s - c_0)m - \Gamma K - T^* \] ..................(9)

where \( T_{L}^{eq} \) is the equilibrium liquidus temperature of the alloy; \( m \) is the slope of the liquidus in the phase diagram, \( \Gamma \) is Gibbs–Thomson coefficient, \( c^*_s \) is the solute concentration at solid–liquid interface, \( c_0 \) is the initial concentration of the alloy liquid, and \( T^* \) is the interface temperature. The concentration field of dendrite tips is obtained from numerical calculation or given by:

\[ c^*_s = c_0(1-f_s)^{k-1} \] ..................(10)

where \( k \) is the solute partition coefficient. The curvature undercooling in Eq. (8) can be obtained by determining the interface curvature \( K \) as:

\[ K = [1 - 2\left( f_s + \frac{N}{l} \right) (N+1)/l] \] ..................(11)

where \( l \) is the lattice parameter of CA cells, \( N \) is the neighbor number of the CA cell.

The relation between solute supersaturation \( \Omega \) and undercooling is given by:

\[ \Delta T = mc_0 [1 - (1 - \Omega (1-k))] \] ..................(12)

Combining equations from Eqs. (5) to (12), we obtain the growth speed of the dendrite tip as:

\[ (1-\Omega)^2 \pi^2 \pi^2 V^2 + \Omega^2 Dw_{c} (1-k) V^2 + \Omega^2 G D^2 = 0 \] ..................(13)

The dendrite tip growth dynamics and preferential growth orientation \( \langle 100 \rangle \) are considered in the computing. The grain orientation is assumed to be random and the probability of new grain orientation falling into the area \( \{ \theta, \theta + d\theta \} \) is given by:

\[ dp(\theta) = 2d\theta / \pi \] ..................(14)

2.3. The Modified CA Model

The grain actually grows in a dendritic way as shown in Fig. 1. Because of the complication, in ordinary CA model, the grain is assumed to grow in square shape (under ideal conditions that the temperature field is homogeneous and equiaxed grains are growing freely) as shown in Fig. 1. Of course, this assumption is not accurate. The grain, in this study, is assumed to grow in some way like shape 2 as shown in Fig. 1. The shape 2 can be described as:

\[ L(\theta) = L_0 [1 + (A-1) \cos 4\theta] \] ..................(15)

When \( A=1.25 \), the profile obtained from Eq. (15) is shown in Fig. 2, which is similar to the outline of the dendrite growth. The comparison of the traditional and modified CA methods is shown in Fig. 3. The nucleation parameters used for the simulation are listed in Table 2.

| \( n_{00}(m^3) \) | \( \Delta T_c (K) \) | \( \Delta T_k (K) \) |
|-----------------|-----------------|-----------------|
| \( 7.2 \times 10^3 \) | 0.5 | 0.1 |
3. The Parallel Computing Algorithm for Microstructure Simulation

The practice in recent years shows that only the large-scale parallel processors can deal with the calculation of the thousand-billionth level. The traditional vector multiprocessor system is not able to achieve this task. The reason is that the computing speed of a single CPU is always restricted by its physical limit, even if the multi-processor structure is adopted. The close coupling among them confines the number of processors and thus makes the calculation impossible. Parallel processors, therefore, are the only choice capable of leading a revolution in computer science.20)

Parallel arithmetic is a kind of programming method used to settle problems on parallel processors. It is composed with some independent computing modules, which can send messages to each other. The characteristic of parallel arithmetic is described by the control method, module scale, and communication configuration. Simply speaking, the solution to a problem is mapped to a group of processes which can be run at the same time, and each process is handled by a processor. In designing a parallel arithmetic for a certain problem on a given parallel computing environment, the most important question is how to decompose a large project into many independent and proper subprojects and make these subprojects operate correctly and keep communication on the parallel processors.

In this paper, the developed parallel arithmetic is a parallel region arithmetic which is based on MPI published in 1994 and runs on the Linux platform.21) Compared with other parallel algorithm, this arithmetic has advantages of the high adaptability, ease to be debugged, and the high property. The computing platform used for parallel calculation is TH-NPSC2 system developed by the Department of Computer Science and Technology in Tsinghua University. Every node in this system is constructed with SMP configuration of two CPUs (PIII 733) and there are 16 nodes (32 CPUs) in the system. Figure 4 shows the parallel arithmetic which is based on a serial one. With this arithmetic, the casting is divided into many parallel regions and one CPU is assigned one such region. After certain time steps, all these regions start to send messages to each other and amend the computing results. These messages are calculation results of gray cells shown in Figure 5. In the process of message exchange, cell parameters, such as the temperature, and status symbol $P_i$ and grain orientation of adjacent micro-cells, will be updated after comparison. The nucleation of micro-cells, which is located on the boundary of one region, will affect the nucleation of some micro-cells of another region and the effect is shown in Figure 6. Because communication parameters are limited in this arithmetic, the communication time can be ignored compared to the time for nucleation and growth of nuclei, and thus, theoretically, there could be a linear relation between number of system nodes and computing efficiency. However, the existence of communication time and other disturbances might decrease the theoretical computational efficiency.
4. Discussion on Experimental and Simulation Results

4.1. Experiments

In order to validate the simulation model, experiments were carried out to obtain the real solidification structure. A number of aluminum alloy sample castings, which have a diameter of 12 mm and a length of 200 mm, were used for the purpose. The raw material was commercial pure aluminum and copper. A 250 kg induction furnace is used to make up the Al–20wt%Cu master alloy. Then the composition of the alloy was adjusted to 4 wt% Cu by using a conventional electric resistance furnace. The molten alloy was poured at 750°C in two types of moulds, sand and metal moulds. The observed specimens were taken from the middle part of the sample castings and observed with an optical microscope after polishing and etching.

4.2. Simulation Results

The simulation was carried out with the developed software to calculate the size and number of grain under the same pouring condition. The metallographs of the sand and metal moulds and the simulated results using modified CA method are shown in Figs. 7 and 8. The thermal and physical properties used in the calculation are listed in Table 3. It is indicated that the simulated results and the experimental results agreed each other quite well. Because of the slow cooling speed of the sand mould, the grains are fairly big and the size is about 300–500 μm, while the cooling speed of the metal mould is faster and the grains are relatively small and the size is about 150–250 μm.

4.3. Computing Efficiency

According to the calculation results, the parallel computing platform increases the calculation efficiency by 20–30 times. The number of macro-lattices of a rod sample casting expounded above is about 40 000 and the number of CA cells in each macro-lattice is about 250 000. It takes 30 h to accomplish the microstructure calculation of 180 macro-lattices using a personal computer while only 1.5 h using the parallel computing environment. The other examples are shown in Table 4.

5. Conclusions

A modified CA method was presented and a proper parallel arithmetic for this model was developed under a paral-
lel computing environment. More accurate shape was proposed to describe the grain growth of aluminum alloy in the modified CA model and thus the accuracy of the simulation result is improved. By dividing the casting into many small regions and taking the advantage of multi-nodes calculation simultaneously, this parallel arithmetic optimizes the serial arithmetic and greatly increases the computing scale and efficiency while the computing accuracy is still ensured. Finally, the simulation results were compared with the experimental results and they agreed each other quite well.

Acknowledgements

The research was sponsored by the Significant Project of National Natural Science Foundation of China (No. 59990470-3), the National Significant Fundamental Research Project of the Ministry of Science and Technology of China (No.G2000067208-3), and the Fundamental Research Project of Tsinghua University.

REFERENCES

1) M. K. Walther: *Mod. Cast.*, 74 (1984), No. 3, 18.
2) M. Kikuchi: *Trans. Jpn. Foundrymen's Soc.*, (1990), 63.
3) C. F. Corbett and M. Picard: *AFS Trans.*, 99 (1991), 369.
4) T. Bex: *Mod. Cast.*, 81 (1991), 23.
5) B. C. Liu: *J Cast Met. Res.*, 11 (1999), 259.
6) W. Oldfield: *ASM Trans.*, 59 (1966), 945.
7) Panping Zhu and R.W. Smith: *Acta Metall.*, 40 (1992), No. 4, 683.
8) J. A. Spittle and S.G.R. Brown: *Acta Metall.*, 37 (1989), No. 7, 1803.
9) P Zhu and R. W. Smith: *Acta Metall.*, 40 (1992), No. 12, 3369.
10) S. Y. Lee, S. M. Lee and C. P. Hong: Modeling of Casting, Welding and Advanced Solidification Processes VIII, ed. by B. G. Thomas and C. Berkmann, TMS, Warrendale, PA, (1998), 383.
11) M. Rappaz and Ch.-A. Gandin: *Acta Metall.*, 41 (1993), No. 2, 345.
12) Q. Y. Xu and B. C. Liu: *China Mechanical Engineering*, 12 (2001), 328.
13) M. Rappaz, Ch. Charbon and R. Sasikumar: *Acta Metall.*, 42 (1994), 2365.
14) Q. Y. Xu and B. C. Liu: The First Int. Conf. on Mechanical Engineering ICME '2000, China Machinery Press, Beijing, (2000).
15) N. Provatas, N. Goldenfeld and J. A. Danziger: Modeling of Casting, Welding and Advanced Solidification Processes VIII, ed. by B. G. Thomas and C. Berkmann, TMS, Warrendale, PA, (1998), 533.
16) J. D. Hunt: *Mater. Sci. Eng.*, 65 (1984), No. 1, 75.
17) Ph. Thévoz, J.-L. Desbiolles and M. Rappaz: *Metall. Trans.*, 20A (1989), 311.
18) W. Kurz, B. Giovanola and R. Trivedi: *Acta Metall.*, 34 (1986), 823.
19) W. W. Mullins and R. F. Sekerka: *J. Appl. Phys.*, 35 (1964), 444.
20) Y. Hu and R. Qiao: *Computer Transaction*, 22 (1999), No. 2, 134 (in Chinese).
21) W. Gropp, E. Lusk and A. Skjellum: Using MPI: Portable Parallel Programming with The Message-passing Interface, MIT Press, Cambridge, Mass., (1999).