

## EFFICIENCY OF A HYBRID PARALLEL ALGORITHM FOR PHASE-FIELD SIMULATION OF POLYCRYSTALLINE SOLIDIFICATION IN 3D

P. STRACHOTA, A. WODECKI, AND M. BENEŠ\*

**Abstract.** We revisit our previously developed algorithm for phase-field simulation of solidification of an arbitrary number of crystals with random crystallographic orientations and a fully resolved 3D dendritic geometry. In this contribution, its hybrid parallel implementation based on the combination of MPI and OpenMP standards undergoes parallel efficiency tests. The results reveal the settings for optimal performance and their dependence on the number of CPUs used. Next, the performance benefits of using the algorithm for single crystal growth are explained. Finally, a very high resolution simulation is demonstrated together with its computational costs.

**Key words.** crystal growth, finite volume method, hybrid parallelization, MPI, OpenMP, orientation field, phase field

**AMS subject classifications.** 65Y05, 65M08, 80A22, 74N05

**1. Introduction.** In simulations of dendritic crystal growth, phase field methods [7, 11] are a common instrument. For each point in space and time, an order parameter  $p(t, \mathbf{x})$  distinguishes between liquid and solid state of matter. To simulate the anisotropic evolution of  $m$  crystals with different crystallographic orientations, either multiple order parameters  $p_1, p_2, \dots, p_n$  must be involved (the so-called multi-phase-field method [21, 20]), or the orientation information must be handled separately, leading to the orientation-field approach [9].

In our previous work [19], we introduced a unique algorithm of domain partitioning with respect to crystallographic orientation. This partitioning evolves as the crystals grow. The C/C++ numerical algorithm used for solving the phase field model has its origin in [17] and was later implemented as a hybrid OpenMP/MPI parallel code, as shown in [18] together with the parallel scalability tests. The algorithm that realizes the orientation information handling has been incorporated into this hybrid parallel framework. In this paper, we provide an analysis of the efficiency of the resulting hybrid parallel implementation and explore its capabilities in terms of very high resolution simulations.

**2. Problem Formulation.** The dimensionless phase field model of solidification of a single component melt with anisotropic surface tension [4, 3, 16] is posed in a domain  $\Omega \subset \mathbb{R}^3$  and time interval  $\mathcal{J} = (0, T)$  as

$$\frac{\partial u}{\partial t} = \Delta u + L \frac{\partial p}{\partial t} \quad \text{in } \mathcal{J} \times \Omega, \quad (2.1)$$

$$\alpha \xi^2 \frac{\partial p}{\partial t} = \xi^2 \nabla \cdot T^0(\nabla p, \mathbf{B}) + f(u + \delta \hat{u}, p, \nabla p, \mathbf{B}; \xi) \quad \text{in } \mathcal{J} \times \Omega, \quad (2.2)$$

$$b_c(u) = 0 \quad \text{on } \mathcal{J} \times \partial\Omega, \quad (2.3)$$

$$T^0(\nabla p, \mathbf{B}) \cdot \mathbf{n} = 0 \quad \text{on } \mathcal{J} \times \partial\Omega, \quad (2.4)$$

$$u|_{t=0} = u_{\text{ini}}, \quad p|_{t=0} = p_{\text{ini}} \quad \text{in } \Omega, \quad (2.5)$$

---

\*Dept. of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, (pavel.strachota@jfifi.cvut.cz).

with the boundary condition operator  $b_c$  in (2.3) given by either  $b_c(u) = u - u_{\partial\Omega}$  or  $b_c(u) = \nabla u \cdot \mathbf{n}$ .  $u$  represents the temperature field and  $p$  the phase field. The value  $p = 0$  corresponds to the liquid phase and  $p = 1$  to the solid phase, with a smooth transition in between. The phase interface is defined by the relation  $\Gamma(t) = \{\mathbf{x} \in \Omega \mid p(t, \mathbf{x}) = \frac{1}{2}\}$ . The reaction term is currently used in the form [4, 2]

$$f(u, p, \nabla p, \mathbf{B}; \xi) = ap(1-p) \left( p - \frac{1}{2} \right) + \xi^2 b \beta \phi^0(\nabla p, \mathbf{B})(u^* - u).$$

The model parameters involve the dimensionless melting point of the material  $u^* = 1$ , the latent heat  $L$ , the attachment kinetics coefficient  $\alpha$ , positive constants  $a, b, \beta$  [4] and the parameter  $\xi$  controlling the thickness of the diffuse interface [5].

The crystallographic orientation given locally by the transformation matrix  $\mathbf{B}$  determines the crystal growth anisotropy via the anisotropic dual Finsler metric  $\phi^0$  and the associated operator  $T^0$ , as explained below.

To obtain more realistic simulation results, random time-independent temperature noise [12]  $\hat{u} : \Omega \rightarrow [-0.5, +0.5]$  is incorporated in the reaction term  $f$ . The perturbation amplitude is controlled by the parameter  $\delta \geq 0$ .

The problem (2.1)–(2.5) describes the evolution of a two-phase system where the solid region grows from one or more initial nucleation sites (where  $p_{\text{ini}} = 1$ ), starting from a uniform (dimensionless) initial supercooling given by  $u_{\text{ini}} = 0$ .

**Anisotropic Surface Energy and Crystallographic Orientation.** The operator  $T^0$  is derived from the dual Finsler metric  $\phi^0(\boldsymbol{\eta}^*)$ ,  $\boldsymbol{\eta}^* \in \mathbb{R}^3$  [1, 3, 6] as

$$T^0(\boldsymbol{\eta}^*, \mathbf{B}) = \phi^0(\boldsymbol{\eta}^*, \mathbf{B}) \phi_{\boldsymbol{\eta}^*}^0(\boldsymbol{\eta}^*, \mathbf{B}) \quad \text{where } \phi_{\boldsymbol{\eta}^*}^0 = (\partial_{\eta_1^*} \phi^0, \partial_{\eta_2^*} \phi^0, \partial_{\eta_3^*} \phi^0)^T. \quad (2.6)$$

The metric  $\phi^0$  assumes the form [13, 10]

$$\phi^0(\boldsymbol{\eta}^*, \mathbf{B}) = |\boldsymbol{\eta}^*| \psi(\mathbf{B}\mathbf{n}) \quad (2.7)$$

where  $\psi : \mathbb{R}^3 \mapsto (0, +\infty)$  represents the anisotropic surface energy depending on the normal direction  $\mathbf{n}$  to the surface  $\Gamma$  and the crystallographic orientation given by the matrix  $\mathbf{B} \in \mathbb{R}^{3 \times 3}$ . The columns of  $\mathbf{B}^T$  form an orthonormal basis  $\mathcal{B} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ . Putting  $\boldsymbol{\eta}^* = \nabla p$  and

$$\mathbf{n} = -\frac{\boldsymbol{\eta}^*}{|\boldsymbol{\eta}^*|} \quad (2.8)$$

ensures that  $\mathbf{n}$  is the outer normal to  $\Gamma$  for  $\mathbf{x} \in \Gamma$ .

$\psi$  introduces anisotropy depending on the coordinates of  $\mathbf{n}$  in the basis  $\mathcal{B}$  given by  $\mathbf{m} = \mathbf{B}\mathbf{n}$ . For example, the formula for 4-fold anisotropy reads [14]

$$\psi(\mathbf{m}) = 1 + A_1 [m_1^4 + m_2^4 + m_3^4 - 6(m_1^2 m_2^2 + m_2^2 m_3^2 + m_3^2 m_1^2)] \quad (2.9)$$

where the coefficient  $A_1$  specifies the anisotropy strength.

Given  $n$  initial nucleation sites, a different basis  $\mathcal{B}_k$  is intrinsic to each crystal growing from the  $k$ -th nucleus for  $k \in \{1, 2, \dots, n\}$ . The method that determines the subdomains  $\Omega_k \subset \Omega$  subject to crystallographic orientation given by  $\mathcal{B}_k$  has been elaborated in [19] and will be explained briefly in the next section.

**3. Numerical Algorithms.** By summarizing the results of [17], [18], and [19], we describe the key aspects of the numerical algorithm with crystallographic orientation handling.

The solution of the equation system (2.1)–(2.5) is based on the following techniques:

- For spatial discretization of the differential equations, a multipoint cell-centered finite volume scheme (FV-MPFA) on uniform quadrilateral meshes is employed [17]. For its evaluation in each cell, the scheme uses 2 layers of neighboring cells in all directions. The higher order of the discretizations is chosen to reduce the numerical anisotropy caused by the mesh geometry.
- The fourth order Runge-Kutta-Merson (RKM) solver [8] with adaptive time stepping carries out the time integration of the resulting system of ordinary differential equations. Note that keeping the random noise field  $\hat{u}$  in (2.2) constant in time guarantees that the time step adjustment algorithm is not affected. The RKM solver can make both successful and unsuccessful time steps. In the latter case, the time integration is repeated with a smaller time step. Each time step involves multiple evaluations of the FV-MPFA scheme.
- MPI parallelization relies on 1-dimensional mesh decomposition into slices along the  $z$  axis and uses 2 layers of halo cells in order to communicate the results to the neighbors before each evaluation of the FV-MPFA scheme. In addition, the RKM solver uses the MPI reduction operations to decide on the time step adjustment.
- Within each MPI process (aka *rank*), OpenMP parallelization is employed in all loops traversing through the mesh (i.e. RKM solver computations and FV-MPFA scheme evaluation). High efficiency of OpenMP parallelization is achieved by
  - creating the threads only once (per snapshot) at entry to the RKM solver,
  - parallelization of all `for` loops in such a way that individual threads process contiguous blocks of memory and can benefit from cache coherence.

The algorithm of domain decomposition according to different crystallographic orientations makes use of a table of pre-generated bases  $\mathcal{B}_1, \dots, \mathcal{B}_n$  corresponding to  $n$  initial nuclei. In each cell, the index  $k \in \{0, 1, \dots, n\}$  to this *orientation table* is stored. The special value  $k = 0$  means no orientation and results in using  $\psi(\mathbf{m}) = 1$ . The values of  $k$  in all cells form the *orientation index field* (OIF). The OIF is initialized to  $k$  in the  $k$ -th initial nucleus and to 0 elsewhere. Afterward, after each successful time step of the RKM solver, the OIF is updated as follows:

1. OIF is reset to 0 except in cells where  $p > \frac{1}{2}$
2. From each cell with  $k = 0$ , a path is constructed toward the nearest crystal by following the gradient of  $p$ . The path ends in a cell where  $k > 0$  or when a local maximum of  $p$  is reached.
3. If a positive value of  $k$  is found, it is propagated to all cells along the path (the path is *resolved*).

From the above steps, it follows that each newly constructed path ends at the cell where it connects to a previously resolved path. As a result, the total length of all paths is bounded by the number of cells in the mesh.

It is important to note that in contrast to the FV-MPFA scheme, the OIF calculation algorithm in each cell is non-local. Its hybrid parallelization consists in the following:

- The mesh decomposition into slices means that paths can cross one or (even) more slice boundaries and OIF data must be communicated among processes.
  - For unresolved paths ending at a boundary, a request for OIF data from the respective neighbor is created.
  - Likewise, for unresolved paths beginning at the boundary, a request for sending data to the respective neighbor is created.
  - Each MPI rank knows if such paths truly continue beyond the boundary from the values of  $p$  in the halo cells.
  - After performing the requested communications, an attempt to resolve the paths is made. If there are paths crossing multiple slice boundaries, several communication/path resolution iterations are necessary.
- OpenMP parallelization is employed for two tasks:
  1. Parallelization of `for` loops iterating over cells in the mesh slice is performed. For the given starting point, path-construction and resolution is a serial process. However, paths constructed by individual concurrent threads can overlap, which introduces a slight processing overhead. However, the possible concurrent memory writes during path resolution do not represent a race condition as the result is uniquely determined by the values of the phase field  $p$ .
  2. Correct determination of necessary MPI communications is achieved by using OpenMP reduction operations on the send/receive requests.

**4. Results.** In the following paragraphs, the parallel performance and capabilities of the described simulation software will be presented. All performed simulation cases have the following common parameter settings:  $a = 2$ ,  $b = 1$ ,  $\alpha = 3$ ,  $\beta = 300$ ,  $L = 2$ . 4-fold anisotropy given by (2.9) is used with  $A_1 = 0.02$ . The domain  $\Omega$  is a cube in the form  $\Omega = (0, \ell)^3$  discretized by a uniform mesh of  $N \times N \times N$  cells. The values of  $\ell$ ,  $N$ ,  $\xi$ , and the final time  $T$  are reported for each of the cases. Zero Neumann boundary conditions are chosen for both  $u$  and  $p$ .

**Runtime Environment.** The simulations were performed on the HELIOS cluster at the Department of Mathematics, FNSPE CTU Prague, using two types of compute nodes:

- many identical AMD **EPYC** nodes:  $2 \times 16$ -core AMD EPYC 7281@2.1GHz CPU (SMT mode disabled), 128 GB DDR4 RAM,
- a single Intel **XEON** node:  $2 \times 16$ -core Intel XEON Gold 6130@2.1GHz CPU (hyper-threading disabled), 384 GB DDR4 RAM

All nodes are connected by the Intel Omni-Path 100Gbit/s interconnect, which is used for MPI communication as well as parallel storage connection. Note however that I/O operations were excluded from wall time measurement.

The relevant software environment involves CentOS 7.6 Linux, Intel C++ Compiler 13.0 and OpenMPI 2.1.5 with PSM module for communication over Omni-Path. Neither OpenMPI process binding nor OpenMP thread binding were used.

**Parallel Efficiency.** To assess how efficient the hybrid parallel algorithm described in Section 3 is, let A and B be two computations of the identical case, performed on  $n_A$  and  $n_B$  CPU cores, respectively. Let the wall times of the computations be  $t_A$  and  $t_B$ , respectively. We define the *efficiency* of computation A with respect to the reference computation B as

$$E_{AB} = \frac{t_B \cdot n_B}{t_A \cdot n_A} \cdot 100\%. \quad (4.1)$$

For example, a 4-core computation A is 100% efficient with respect to the reference 2-core computation B if A takes half the wall time of B. When  $n_A = n_B$ , we name  $E_{AB}$  the *relative* efficiency.

Figure 4.1 presents the comparison of hybrid parallel computations on a single compute node. For the total number of threads (i.e. used CPU cores) equal to 16 and 32, respectively, all the possible ratios between the number of MPI ranks and the number of OpenMP threads per rank were tested. Both EPYC and XEON nodes were used. As the used mesh size ( $N = 224$ ) and hence the wall times were both relatively small, all the computations were repeated and the average wall times were calculated to control for time measurement inaccuracies. On XEON, 2 runs were performed that gave almost identical results (with relative differences  $< 0.1\%$ ), so further runs were not launched. On EPYC nodes, however, all computations were repeated 8 times, resulting in time measurements with the coefficient of variation between 0.8% and 3.5%.

The relative efficiencies between the different combinations of the numbers of threads and ranks were plotted. In this situation, “MPI wins”, i.e. using hybrid parallelization slightly reduces the performance compared to MPI-only parallelization. This may be caused by both hardware reasons (memory affinity on NUMA architecture [15]) and the overhead in the path finding algorithm. The algorithm without OIF behaved a little more in favor of OpenMP [18], but on a different hardware.

In addition, the fastest of the computations on 16 and 32 cores were compared with the serial (single-core) computation. There is a remarkable efficiency drop on the XEON node, which can be attributed to the “turbo” CPU feature able to increase the base frequency from 2.1 GHz up to 3.7 GHz when some of the CPU cores are idle. In contrast to that, the used EPYC CPUs feature the “all core boost” technology able to raise the frequency of all cores up to 2.7 GHz (compared to 2.1 GHz baseline) as long as the thermal conditions permit. Our tests indicate that in the HELIOS cluster, the “boost” frequency can be maintained under maximum load for prolonged periods of time.

In Figure 4.2, efficiency of computations on a finer mesh ( $N = 416$ ) with a higher number of crystals (50 instead of 10) is demonstrated. Such a mesh resolution allows the parallelization on more than one compute node. Computations on 1, 2, and 4 nodes were performed, using all 32-cores of each node in different combinations of MPI ranks and OpenMP threads. The results show that while MPI alone is still best performing on a single node, its efficiency decreases sharply when 2 and 4 nodes are used. The mesh slices become thinner, requiring more MPI communication to happen, possibly including multiple iterations of the path resolution algorithm. The more total CPU cores are used, the less MPI ranks and more OpenMP threads per rank are required to achieve the best performance. In the optimal setting, multi-node computations exhibit a high efficiency with respect to the single-node computation.

**The Effect and Cost of OIF Computation.** The model without OIF (in the form described in [17]) can be used to simulate the growth of a single crystal and a single crystallographic orientation (aligned with the coordinate axes) governs the whole domain  $\Omega$ . By introducing the OIF even for a single crystal, the following consequences apply:

1. The OIF is set to  $k = 1$  (the single orientation present) in the neighborhood of the crystal. Unlike in the simpler algorithms [21, 20], this neighborhood reaches as far as the discretization of  $\nabla p$  on the mesh is nonzero. This is almost exactly sufficient for the right hand side of (2.2) to yield identical

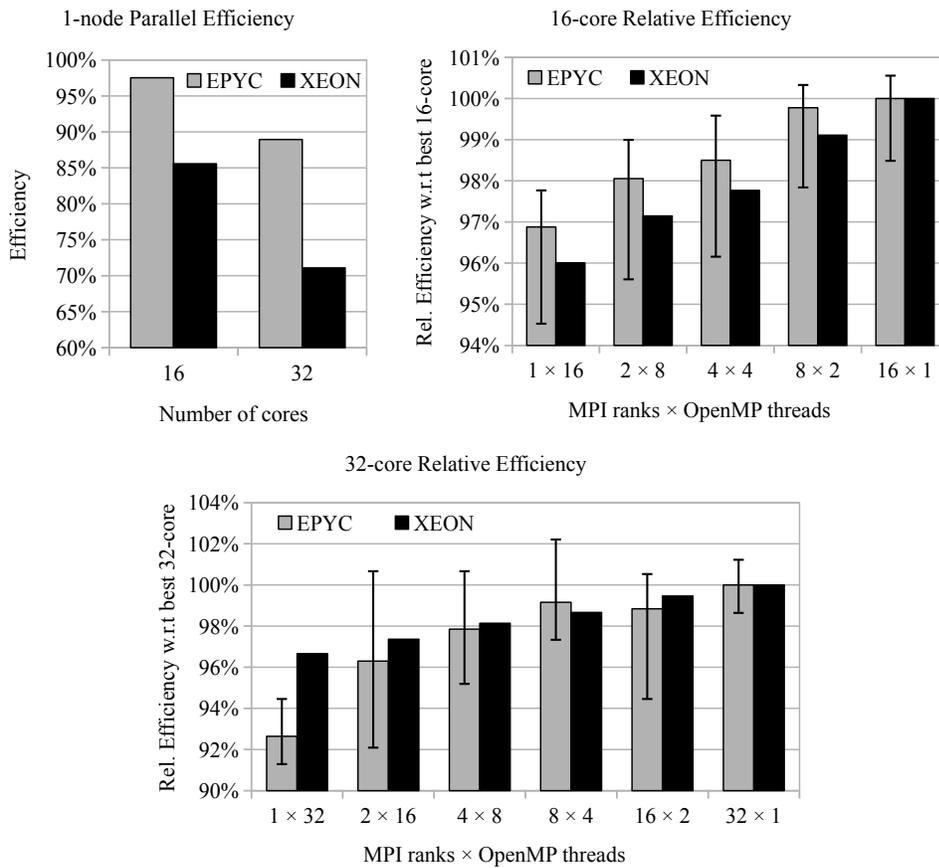


FIGURE 4.1. Efficiency and relative efficiency of hybrid parallelization on a single compute node with different numbers of MPI ranks and OpenMP threads. The (on average) best performing variants on both 16 and 32 cores were the pure MPI ( $16 \times 1$  and  $32 \times 1$ ) computations. Their average wall times were used to evaluate the efficiencies w.r.t. single-core computation and also as the reference values of  $t_B$  in (4.1) for evaluating the relative efficiencies. For EPYC, the solid bars correspond to the average wall times and the error bars indicate the minimum and maximum measured relative efficiencies out of all repeated runs.  $\xi = 0.02$ ,  $\delta = 0$ , domain dimension  $\ell = 8$ , mesh resolution  $N = 224$ , final time  $T = 0.2$ ,  $n = 10$  initial nuclei.

(nonzero) values as if the OIF were 1 everywhere, which is equivalent to not using OIF at all. Currently, the use of the higher-order FV-MPFA scheme and OIF updates after each successful time step only (in order not to interfere with the RKM time step adjustment) can technically cause nuances between the results obtained with and without the use of the OIF.

2. The computation of the OIF represents extra work that can slow down the simulations.
3. The use of a simpler numerical scheme wherever  $k = 0$  (“no orientation” far from the crystal) can possibly *speed up* the simulations.

The real effect of introducing OIF on computational costs for a sample single crystal growth simulation is shown in Figure 4.3. The influence of different settings of hybrid parallelization is also demonstrated. The algorithm with OIF is clearly faster. During the computation, 20 snapshots of the result were saved at regular intervals. The wall

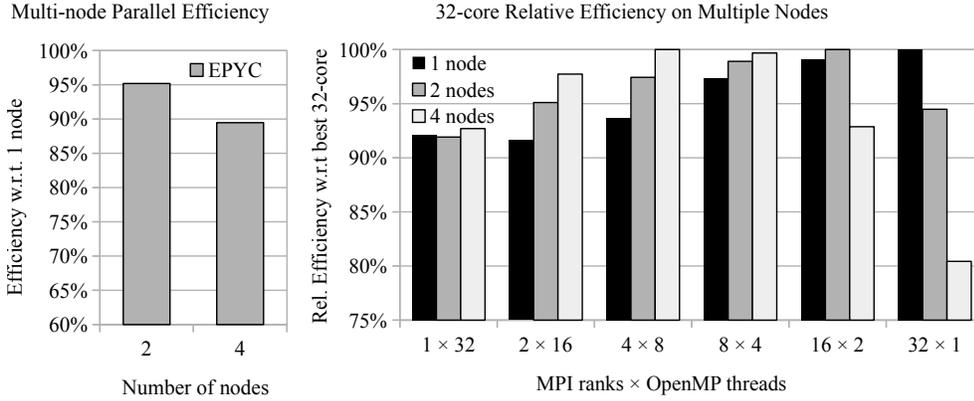


FIGURE 4.2. Efficiency and relative efficiency of hybrid parallelization on multiple compute nodes with different numbers between of MPI ranks and OpenMP threads on each 32-core node. The efficiencies (w.r.t single-node computation) are taken for the best performing variant on 2 and 4 nodes, respectively.  $\xi = 0.011$ ,  $\delta = 0$ , domain dimension  $\ell = 8$ , mesh resolution  $N = 416$ , final time  $T = 0.2$ ,  $n = 50$  initial nuclei.

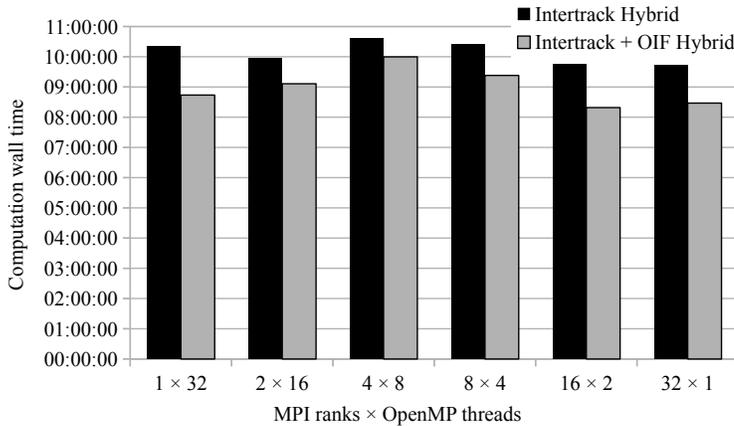


FIGURE 4.3. Comparison of the total wall times of the simulation of single crystal growth by the hybrid parallel version of Intertrack with [19] and without [18] OIF.  $\xi = 0.011$ ,  $\delta = 0$ , domain dimension  $\ell = 8$ , mesh resolution  $N = 512$ , final time  $T = 0.2$ . One EPYC node was used.

times per snapshot in Figure 4.4 indicate how the algorithm with OIF benefits from consequence no. 3. The speed increase is naturally more significant when the crystal is small. Interestingly, even the computation without OIF gradually slows down as the simulation progresses.

Last but not least, the obtained crystal shape (not shown here) is *identical* in both simulations.

**Very High Resolution Simulation.** As the last result, we demonstrate the ability to simulate crystal growth with a large number of crystals on a very high resolution mesh. 800 nucleation sites were randomly placed into a domain  $\Omega = (0, 16)^3$  discretized by a mesh of  $1024^3$  cells. Random noise was used to support the complex dendritic growth of the crystals. The simulation up to (dimensionless) time  $t = 0.3$  took 6896 time steps and 23 hours and 52 minutes of wall time on 8 EPYC nodes,

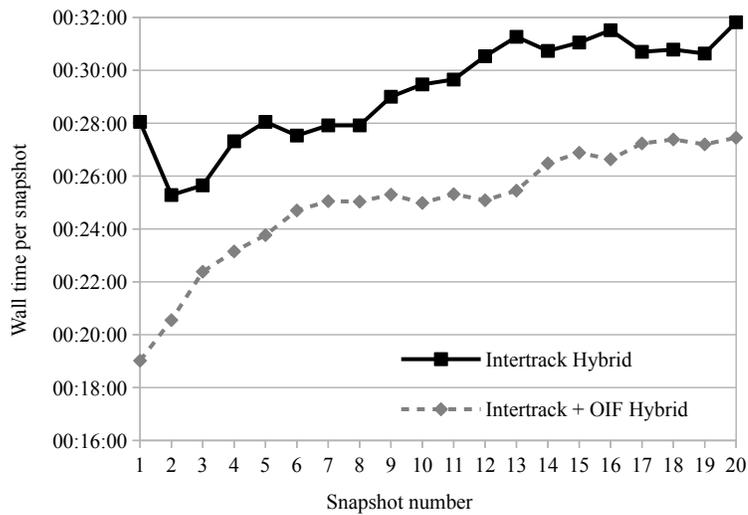


FIGURE 4.4. Comparison of the wall times per snapshot of the simulation of single crystal growth by the hybrid parallel version of Intertrack with [19] and without [18] OIF.  $\xi = 0.011$ ,  $\delta = 0$ , domain dimension  $\ell = 8$ , mesh resolution  $N = 512$ , final time  $T = 0.2$ . The computation with 16 MPI ranks  $\times$  2 OpenMP threads on one EPYC node (see Figure 4.3) was investigated.

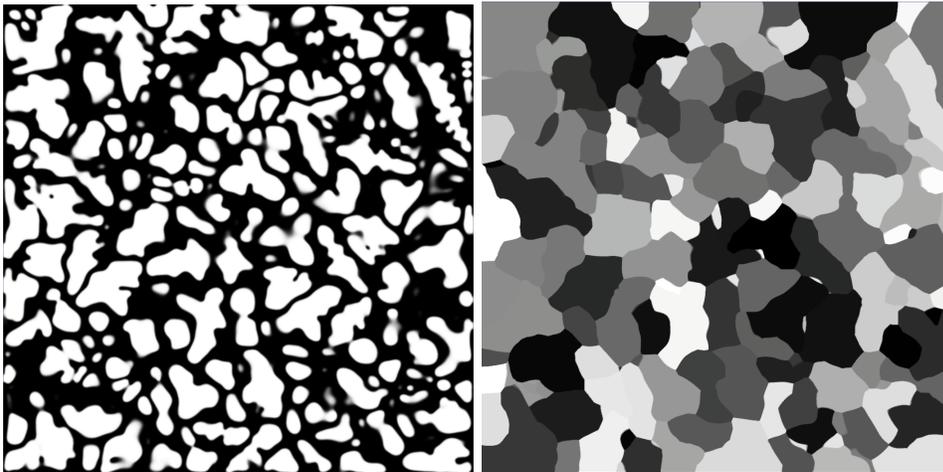


FIGURE 4.5. The phase field  $p$  on a gray scale (left) and subdomains with different orientations (right) depicted in different shades of gray. A slice through the plane  $z = 8$  in a very high resolution simulation with  $\xi = 0.011$ ,  $\delta = 0.05$ , domain dimension  $\ell = 16$ , mesh resolution  $N = 1024$ , time  $t = 0.3$ , and  $n = 800$  initial nuclei.

each hosting 4 MPI ranks with 8 OpenMP threads. Saving 320 GB of data in 16 snapshots of the evolving solution took another 12 minutes. The slice through the resulting phase field  $p$  and the partitioning of the domain  $\Omega$  based on different values of the OIF are shown in Figure 4.5. The visualization of the final shape of the crystals is in Figure 4.6.

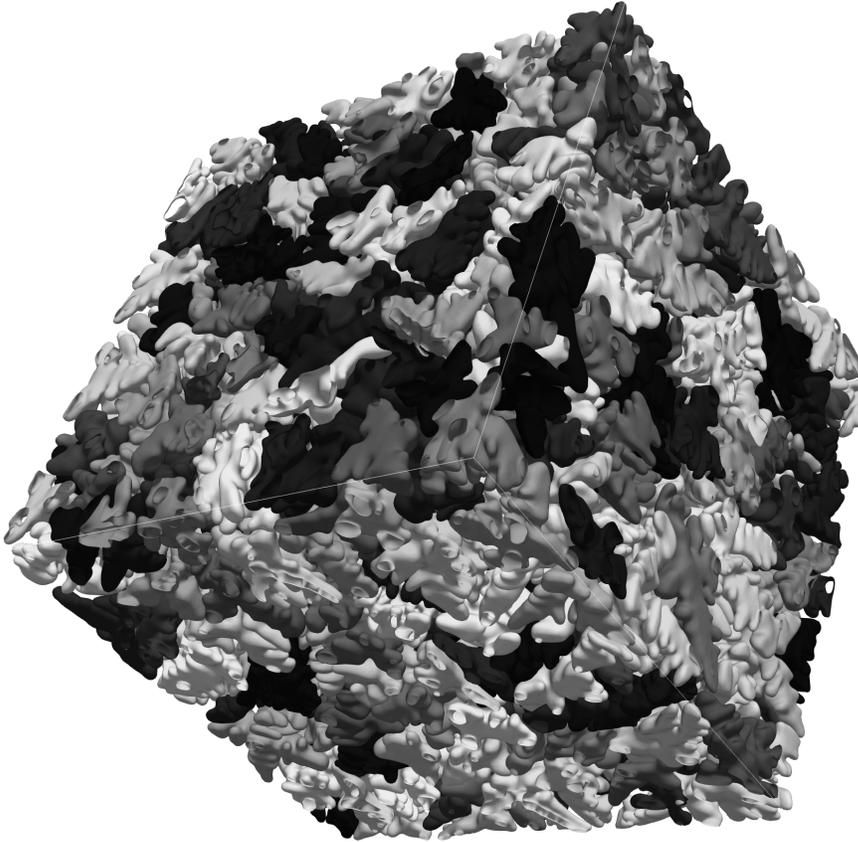


FIGURE 4.6. 3D visualization of the crystal shapes in a very high resolution simulation with  $\xi = 0.011$ ,  $\delta = 0.05$ , domain dimension  $\ell = 16$ , mesh resolution  $N = 1024$ , time  $t = 0.3$ , and  $n = 800$  initial nuclei. Each crystal has a unique random crystallographic orientation and is colored in a different shade of gray.

**5. Conclusion.** The performed tests revealed that hybrid MPI/OpenMP parallelization of the presented algorithm is a viable technique. While its performance is comparable to MPI on a smaller number of CPU cores, it becomes unequivocally superior as the number of cores increases. The evaluation of the results also revealed interesting differences in the behavior of Intel and AMD server processors used in the test. In addition, using OIF for simulations of the growth of a single crystal proved to have rather unexpected performance benefits

**Acknowledgments.** This work is part of the project *Centre of Advanced Applied Sciences* (Reg. No. CZ.02.1.01/0.0/0.0/16-019/0000778), co-funded by the European Union.

## REFERENCES

- [1] G. Bellettini and M. Paolini. Anisotropic motion by mean curvature in the context of Finsler geometry. *Hokkaido Math. J.*, 25(3):537–566, 1996.
- [2] M. Beneš. *Phase Field Model or Microstructure Growth in Solidification of Pure Substances*. PhD thesis, Czech Technical University in Prague, 1997.
- [3] M. Beneš. Anisotropic phase-field model with focused latent-heat release. In *FREE BOUNDARY PROBLEMS: Theory and Applications II*, volume 14 of *GAKUTO International Series in Mathematical Sciences and Applications*, pages 18–30, 2000.
- [4] M. Beneš. Mathematical and computational aspects of solidification of pure substances. *Acta Math. Univ. Comenianae*, 70(1):123–151, 2001.
- [5] M. Beneš. Diffuse-interface treatment of the anisotropic mean-curvature flow. *Appl. Math.-Czech.*, 48(6):437–453, 2003.
- [6] M. Beneš. Computational studies of anisotropic diffuse interface model of microstructure formation in solidification. *Acta Math. Univ. Comenianae*, 76:39–59, 2007.
- [7] W. J. Boettinger, S. Coriell, A. L. Greer, A. Karma, W. Kurz, M. Rappaz, and R. Trivedi. Solidification microstructures: Recent developments, future directions. *Acta Mater.*, 48:43–70, 2000.
- [8] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley, Chichester, 2nd edition, 2008.
- [9] Z. Guo and S. M. Xiong. Study of dendritic growth and coarsening using a 3-D phase field model: Implementation of the Para-AMR algorithm. *IOP Conf. Ser.: Mater. Sci. Eng.*, 84:012067, 2015.
- [10] M. E. Gurtin. *Thermomechanics of Evolving Phase Boundaries in the Plane*. Oxford Mathematical Monographs. Oxford University Press, 1993.
- [11] A. Karma and W.-J. Rappel. Quantitative phase-field modeling of dendritic growth in two and three dimensions. *Phys. Rev. E*, 57(4):4, 1998.
- [12] R. Kobayashi. Modeling and numerical simulations of dendritic crystal growth. *Physica D*, 63:410–423, 1993.
- [13] R. E. Napolitano and S. Liu. Three-dimensional crystal-melt Wulff-shape and interfacial stiffness in the Al-Sn binary system. *Phys. Rev. B*, 70(21):214103, 2004.
- [14] M. PunKay. Modeling of anisotropic surface energies for quantum dot formation and morphological evolution. In *NNIN REU Research Accomplishments*, pages 116–117. University of Michigan, 2005.
- [15] P. Stenström, T. Joe, and A. Gupta. Comparative performance evaluation of cache-coherent NUMA and COMA architectures. In *ISCA '92 Proceedings of the 19th annual international symposium on Computer architecture*, volume 20, pages 80–91. ACM New York, NY, USA, May 1992.
- [16] P. Strachota. *Analysis and Application of Numerical Methods for Solving Nonlinear Reaction-Diffusion Equations*. PhD thesis, Czech Technical University in Prague, 2012.
- [17] P. Strachota and M. Beneš. Design and verification of the MPFA scheme for three-dimensional phase field model of dendritic crystal growth. In A. Cangiani, R. L. Davidchack, E. Georgoulis, A. N. Gorban, J. Levesley, and M. V. Tretyakov, editors, *Numerical Mathematics and Advanced Applications 2011: Proceedings of ENUMATH 2011, the 9th European Conference on Numerical Mathematics and Advanced Applications, Leicester, September 2011*, pages 459–467. Springer Berlin Heidelberg, 2013.
- [18] P. Strachota and M. Beneš. A hybrid parallel numerical algorithm for three-dimensional phase field modeling of crystal growth. In A. Handlovičová and D. Ševčovič, editors, *ALGORITMY 2016, 20th Conference on Scientific Computing, Vysoké Tatry - Podbanské, Slovakia, March 14 - 18, 2016. Proceedings of contributed papers and posters*, pages 23–32. Comenius University, Bratislava, 2016.
- [19] P. Strachota and A. Wodecki. High resolution 3D phase field simulations of single crystal and polycrystalline solidification. *Acta Phys. Pol. A*, 134(3):653–657, September 2018.
- [20] Y. Suwa. Phase-field simulation of grain growth. Technical Report 102, Nippon Steel, January 2013.
- [21] Y. Suwa and Y. Saito. Computer simulation of grain growth by the phase field model. effect of interfacial energy on kinetics of grain growth. *Mater. Trans.*, 44:2245–2251, 2003.