Development of parallel implementation for the dendritic crystallograms modeling algorithm

Paringer R.A., Kupriyanov A.V.

Samara State Aerospace University
Image Processing Systems Institute, Russian Academy of Sciences

Abstract. The paper considers a simulation algorithm for dendritic crystallogram images and offers its parallel implementation using MPI technology. As a basis we took an algorithm using an impurity-and-material-substance diffusion equation. The algorithm used as a guide was upgraded. An impurity redistribution method was changed, and the order of crystallization was updated that allowed to maintain the impurity volume during the crystal growth. A separation technique for algorithm stages was proposed on compute cores. An acceleration value of the proposed MPI-implementation has proven to be 20% more than the OpenMP analogue. The resulting implementation may be used to simulate large crystallograms in shared-memory systems.

Keywords: dendrite crystallograms·dendrite simulation·MPI·OpenMP·

Citation: Paringer R.A., Kupriyanov A.V. Development of parallel implementation for the dendritic crystallograms modeling algorithm. Proceedings of Information Technology and Nanotechnology (ITNT-2015), CEUR Workshop Proceedings, 2015; 1490: 285-289. DOI: 10.18287/1613-0073-2015-1490-285-289

1. Introduction

The analysis of medical crystallogram images is an important part of medical diagnostics. Medical crystallograms are the structures formed at crystallization of salts as a result of drying biological liquids (tears, blood, saliva, etc.). Automated processing of crystallogram images will enable to improve the quality of diagnostics and will reduce the time required for a diagnostic procedure [1-4].

It is necessary to simulate the image of the entire drop, therefore in order to speed up the computing it is proposed to use parallel computations, that widely used lately [5, 6]. There are several possible techniques to implement a parallel algorithm. Let us consider two of them: OpenMP and MPI [7]. The OpenMP technique is used for shared-memory systems, and the number of compute cores in such systems rarely exceeds 16, which imposes certain restrictions on the system scaling. The need for MPI implementation arises if you want to simulate a large crystallogram with a fine partition grid.

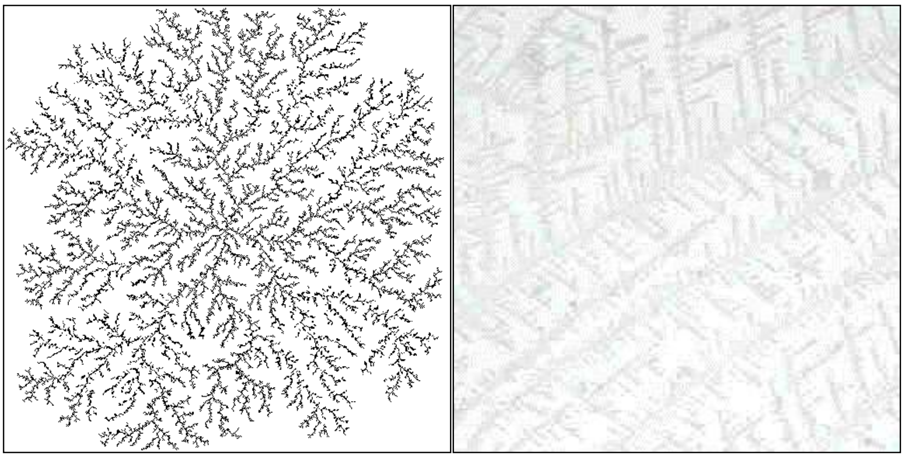

Fig. 1. – Crystallogram image. Simulated (left). Full-scale (right).

2. MPI implementation

The previously described model has been taken as a basis [8]. The OpenMP description of implementation of the simulation algorithm is presented in references below [9]. The model implementation includes consistent application of the following algorithms: diffusion, calculation of crystallization probability, check of material substance quantity, material substance redistribution, impurity redistribution, and crystal dissolution.

One of the model disadvantages is the fact that under certain simulation parameters the impurity conservation law was not executed. The impurity quantity was reduced. Therefore, we used the impurity wave redistribution: the entire impurity in a crystallized cell was redistributed to neighboring cells of different orders. To evaluate the possibility of performing such distribution we had to add the check of impurity quantity; it determines whether the remaining volume will be enough to store the impurity displaced during crystallization.

The second disadvantage is the fact that the crystal dissolution is performed after crystallization and, accordingly, after the check of the material substance quantity available for crystallization that lead to skipping some crystallization stages and to reducing the crystal growth. Therefore, we have changed the order of these actions. First, the crystals are dissolved, and then crystallization is performed, thus the amount of the material substance available for crystallization increases. This change did not lead to significant differences in a crystal shape, but it allowed to reduce the time of the crystal growth up to 2% of the total time. The obtained algorithm is schematically shown in Figure 2.

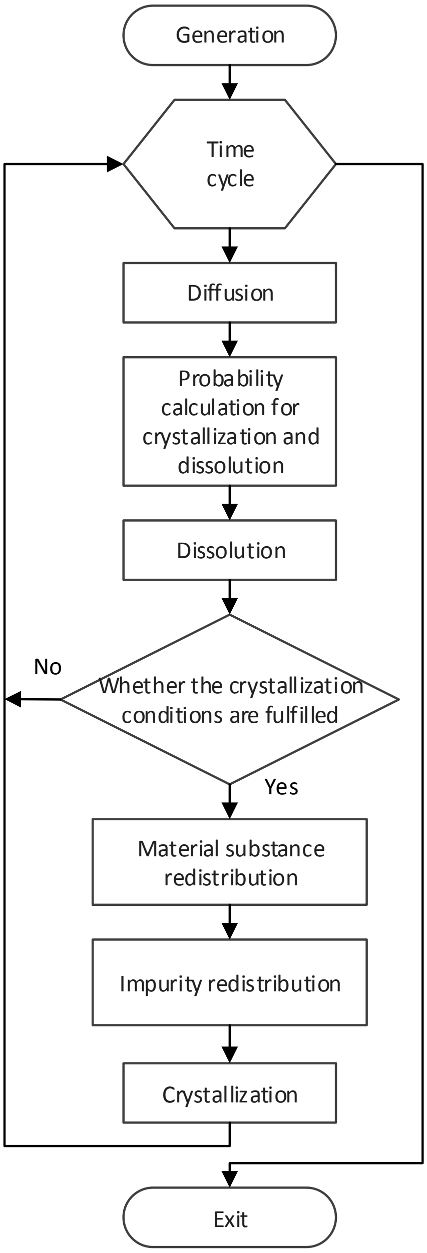
Generation
Time cycle
Diffusion
Probability calculation for crystallization and dissolution
Dissolution
Whether the crystallization conditions are fulfilled
No
Yes
Material substance redistribution
Impurity redistribution
Crystallization
Exit

Fig. 2. – Crystallogram simulation algorithm

Table 1 shows the dependence of the relative operation time of algorithm stages on the image size. It should be noted that the longest time is taken by diffusion of the material substance, however since an explicit scheme is used, the diffusion algorithm may be split up at each stage.

Table 1. Dependence of a relative operation time of algorithm stages on the image size

Algorithm stage	Image size			
	512× 512	1024 ×1024	2048 ×2048	4096 ×4096
Diffusion	0.27	0.32	0.30	0.33
Probability calculation	0.16	0.18	0.17	0.19
Dissolution	0.04	0.04	0.04	0.04
Crystallization condition test	0.09	0.08	0.08	0.08
Impurity redistribution	0.23	0.18	0.18	0.17
Material substance redistribution	0.15	0.14	0.15	0.14
Crystallization	0.06	0.06	0.06	0.06

The parallel implementation in MPI is carried out by the exchange of messages between compute cores. The acceleration takes place due to the fact that independent operations are also performed in parallel, but a part time is spent to transfer data between calculators. Therefore, the less operations of the message exchange, the greater the efficiency of the parallel program.

In the parallel implementation of the algorithm in each time cycle iteration (except the first one) the data are to be casted after the impurity redistribution, and the computing results are to be collected before the operation of material substance redistribution, but only if the crystallization conditions are satisfied. The “Material substance redistribution” and the “Impurity redistribution” methods, due to implementation features (i.e. use of wave redistribution), may not be efficiently parallelized using MPI, since it will require a large number of transmissions. Therefore, the “Material substance redistribution” and the “Impurity redistribution” methods are executed sequentially, but in different flows, since they use independent data.

Acceleration results of the obtained program implementation are shown in Table 2. The resulting acceleration values were 20% higher than in a similar paper using the OpenMP technology [8].

Table 2. Dependence of the acceleration on the number of cores and the image size

Image size	Number of compute cores						
	2	3	4	5	6	7	8
512×512	1.67	2.06	2.34	2.54	2.70	2.82	2.93
1024×1024	1.74	2.22	2.57	2.85	3.07	3.25	3.40
2048×2048	1.75	2.23	2.59	2.86	3.08	3.26	3.40
4096×4096	1.76	2.26	2.65	2.94	3.18	3.38	3.54

3. Conclusion

A parallel algorithm for the simulation of dendritic crystallograms was presented. It was described a modification of the existing sequential algorithm taking into account the chosen technology of parallelization. The resulting implementation turned out to be 20% faster than the OpenMP implementation. The developed algorithm can be used to simulate the high-resolution images crystallograms.

Acknowledgements

This work was partially supported by the Ministry of education and science of the Russian Federation in the framework of the implementation of the Program of increasing the competitiveness of SSAU among the world’s leading scientific and educational centers for 2013-2020 years; by the Russian Foundation for Basic Research grants (# 14-01-00369-a, # 14-07-97040-p_ povolzh'e_a, # 15-29-03823, # 15-29-07077); by the ONIT RAS program # 6 “Bioinformatics, modern information technologies and mathematical methods in medicine” 2015.

References

1. **Denisov AB.** Crystal structures of the oral fluid. Message 1. The estimation method of crystal figures obtained on drying mixed saliva. Dental forum, 2011; 37(1): 50-54.
2. **Shabalin VN, Shatokhina SN.** The morphology of human biological fluids. Chrysostom, 2001; 304 p.
3. **Volchetsky AL, Ruvinova LG, Spasennikov BA, Zenovsky VP.** Crystallization and crystallography: Biomedical aspects. Arkhangelsk: Pomorsky State University Publishing House, 1999; 190 p.
4. **Martusevitch AK.** The crystallographic analysis: general characteristics. The Vyatsk Medical Bulletin, 2002; 3: 59-61.
5. **Vorotnikova DG, Golovashkin DL.** Long vectors algorithms for solving grid equations of explicit difference schemes. Computer Optics, 2015; 39(1): 87-93. [in Russian]
6. **Zhdanov AI, Sidorov YV.** Parallel implementation of a randomized regularized Kaczmarz's algorithm. Computer Optics, 2015; 39(4): 536-541. [in Russian]
7. **Antonov AS.** MPI and OpenMP Multiprogramming technologies. Moscow State University Publishing House, 2012; 344 p.
8. **Baranov VG, Khramov AG.** Simulation of the growth process of dendritic crystal structures. Computer Optics, 2001; 21: 193-197. [in Russian]
9. **Khalirakhmanov DI, Mayakova SA.** Parallel algorithm of the growth simulation of dendritic crystal structures. Proceedings of the International Conference “Parallel Computing Technologies 2012”, 2012; 704–710.