

Constructing three-dimensional (3D) polycrystalline models of FCC crystal for numerical modeling and simulation with multibody potential

Rustam Akhunov

May 12, 2018

Contents

1	Introduction	2
2	Overview of existing methods and results	2
3	Building the Voronoi tessellation	2
3.1	Building the Voronoi tessellation fitting the given grain size distribution	2
3.2	Building the Voronoi tessellation fitting the given grain orientation distribution	4
4	Filling tessellation with particles (atoms)	5
5	Relaxation of atomic structure	5
6	Results and discussion	5
7	Methods	5
8	Additional Information	5
9	Conclusion	5
10	Acknowledgments	6

Abstract

The three-dimensional (3D) polycrystalline models of FCC crystal with the log-normal grain size distribution are constructed by constrained Voronoi tessellation. For achieving needed distribution of grain sizes and grain orientation we used Genetic Algorithm (GA) with Least Square

(LS). We used molecular-dynamics method in LAMMPS software to obtain the relaxed polycrystal. For the relaxation process and for simulation we used multibody potential.

1 Introduction

There are known a lot of modern application of polycrystalline materials. Polycrystallines are used for tritium generation [1], automotive, healthcare and non-destructive testing [2], solar cells [3], semiconducting elements [4] etc. With increasing demand of new properties of new materials increases also need for the modern methods of polycrystalline simulation [7]. And this work is one more attempt to deal with the simulation of polycrystallines in molecular dynamics environment such as LAMMPS [5, 6]

There are numbers of experiments and theoretical research for the intrinsic structure, effective thermal conductivity, lattice dynamical, optical and thermodynamic properties of polycrystalline [7]. But still many characteristics have not been researched sufficiently.

We know few methods for generating digital nanostructured materials for use in the simulation: PoissonVoronoi tessellation (PVT) [9], Monodisperse grain size (MGS) [8], LaguerreVoronoi tessellation (LVT) [10], JohnsonMehl (JM) model [11] etc. In this work we used Voronoi tessellation model for the reason of its good solution for the complex structural nanocrystals and also convenient interface for the genetic algorithm, that we used during our polycrystalline generation. It is also generally accepted that many single-phase fully dense nanocrystals are described by a log-normal grain size distribution. And we took this distribution as a reference for polycrystalline generation.

2 Overview of existing methods and results

3 Building the Voronoi tessellation

As it was mentioned above we use Voronoi tessellation for building our polycrystalline structure. But for obtaining real structure we have to fit some properties of natural crystals during generation of tessellation. In our work we are especially interested in fitting tessellation to the two crucial distributions: grain size distribution and grain orientation distribution. All generation done in periodic condition of the boundary.

3.1 Building the Voronoi tessellation fitting the given grain size distribution

In our generation process we tested several distributions for grain size but only one of them was chosen as a distribution for real polycrystal, it is lognormal distribution [7, 15].

As far as we have a set of 3D points in the space as a individual we are considering all operators acting on these points because we do not have any other objects to deal with. So, crossover operator just exchange points between two individuals according to the some rule defined in the implementation. And mutation operator just shifts the given points of individual according to the defined probability.

For fitting grain size distribution we used genetic algorithm. In order to find best conditions of algorithm convergence we implemented several variants for crossover and mutation operator. Especially for crossover we have one-point, two-point and uniform crossover. First and second variants just straightforward classical implementation of the operator. While uniform crossover is kind of non-standard variant where point-to-point exchange rule follow random distribution. In other words, we take one point from first individual and randomly exchange with random point from the second individual. While in one- and two-point crossover we exchange points according to the virtual distance between them. Particularly, we exchange point if they are close to each other. You can see this mapping in figure

referencetothe figure

For the mutation operator we implemented variant with different distribution for point shifting value. Especially we implemented and tested two kind of distribution: uniform and normal. The maximum amplitude for the shifting is adopting to the system under consideration. We compute average volume for the one particle by dividing the whole simulation box by the amount of particles. After we take to cube root of this volume and obtain the adopted shifting maximum amplitude for the mutation operator.

Another variant for the mutation operator it is just reinitialization of the given point.

If we generalize the whole algorithm we implemented we can highlight 4 parameters that impact on the convergence and convergence speed of the genetic algorithm. They are crossover operator type, mutation distribution, mutation probability and population size.

By changing and varying the set of these parameters we obtain particular configuration of the algorithm. Our goal is to find the optimal configuration when the algorithm will converge and will do it quickly enough.

In order to obtain the optimal configuration we chose some discrete values for population size and mutation probability and assemble all possible configurations from these values. For mutation probability we considered values 0.8, 0.2, 0.05, 0.01. For population size we considered 10, 40, 100. For operators variants are already discrete. Namely, for crossover operator: one-point, two-point and uniform and for mutation: uniform, normal, reinitialization. All these configurations are accepted by the bash script as a parameter that makes testing of all possible configurations easy enough.

For each particular configuration we run 48 instances due to our idea to find the mean value and standard deviation for each considered point.

For aggregation data from each configuration runs and finding mean value and standard deviation we used python pandas module.

The result for best, worst and tree medium configurations are presented in the graphs below.

figurewithbestconfigetc

In these figures there are presented 3 lines: the best individual's penalty, the worst individual's penalty and the average penalty for the number of evaluating of penalty function.

3.2 Building the Voronoi tessellation fitting the given grain orientation distribution

For reaching selected grain orientation distribution we also used genetic algorithm. But it is supposed to use this optimization only after reaching grain size distribution because in case of grain orientation we do not change tessellation but rather we change only orientation of the grains.

Each orientation is represented by Euler angles. Although we have orientation for each grain we are interested in relative orientation between grains. We introduce this orientation as a difference between orientations of two selected grains. It can be imagined as angles needed to rotate first orientation in order to obtain the second orientation and vise versa.

In particular we use the same crossover operators for choose which points we have to exchange but instead of exchanging points themselves we exchange their orientations and leave points positions intact. Therefore we need only one individual of tessellation and many configuration of orientation.

For mutation operator we change slightly the Euler angles of the grain. We consider only one octant with angles from 0 to $\pi/2$ because we are dealing only with cubic lattice in this work. Latter on outside of this work we will work with more complex lattices.

We can think about two algorithms for grain sizes and grain orientation as a similar problems with the similar properties. For example, when we change position of one point in grain size algorithm we change sizes of current cell and all neighboring cells. The same for grain orientation algorithm. When we change orientation for one cell we change all relative orientation around particular grain.

For grain orientation algorithm we also have the set of possible parameters that influence on the convergence. They are the same as for grain size algorithm. We also have to find optimal configuration for the algorithm and this is why we performed the same procedure of calling script that variates all possible configuration. After running all configuration several times (48 times) we in the same way calculate mean value and standard deviation. Here are results of worst, best and tree medium configurations:

figuresrepresentinggrainorientationalgorithmresults

4 Filling tessellation with particles (atoms)

After obtaining fitted Voronoi tessellation we have to fill it with the atoms. The filling should correspond to the reached grain orientation distribution. In order to do this we generate lattice with given lattice constant around the grain point with size exceeding grain size twice to cover all possible artifacts during rotation. After that we rotate generated set of atoms by the euler angles that we got during grain orientation optimization. Now we cut all redundant atoms that lie outside of the considered grain and so we have filled grain of the polycrystalline.

The only trick that we need during this generation is following the periodic condition of the system. This is why if we have grain touching the boundary of the simulation box we need to transfer atoms to the corresponding part of grain lying in the opposite side of the box.

5 Relaxation of atomic structure

6 Results and discussion

7 Methods

8 Additional Information

9 Conclusion

References

- [1] Johnson C. E. J. *Tritium behavior in lithium ceramics* Nucl. Mater 1999
- [2] Pardo, Lorena, Ricote, Jess (Eds.) *Multifunctional Polycrystalline Ferroelectric Materials* Springer 2011
- [3] Schropp, Ruud E. I. Zeman, Miro *Amorphous and microcrystalline silicon solar cells : modeling, materials, and device technology* Boston (Mass.) : Kluwer academic 1998.
- [4] Harbeke, G. (Ed.) *Polycrystalline Semiconductors* Springer-Verlag Berlin Heidelberg 1985
- [5] S. Plimpton *Fast Parallel Algorithms for Short-Range Molecular Dynamics* J Comp Phys, 117, 1-19 1995
- [6] S. Plimpton *LAMMPS WWW Site* <http://lammps.sandia.gov>

- [7] Shen, Y. et al. *Constructing three-dimensional (3D) nanocrystalline models of Li_4SiO_4 for numerical modeling and simulation*. Sci. Rep. 5, 10698; doi: 10.1038/srep10698 2015
- [8] Wang J. et al. *Computer simulation of the structure and thermo-elastic properties of a model nanocrystalline material* Philos. Mag. A. 73, 517555 1996
- [9] Weaire D. et al. *On the distribution of cell areas in a Voronoi network* Philos. Mag. B 53, 101105 1986
- [10] Fan Z. G. et al. *Simulation of polycrystalline structure with Voronoi diagram in Laguerre geometry based on random closed packing of spheres* Comput. Mater. Sci. 29, 301308 2004
- [11] Okabe A. et al. *Spatial Tessellations-Concepts and Applications of Voronoi Diagrams* Wiley, New York, 2000
- [12] S. N. Sivanandam , S. N. Deepa *Introduction to Genetic Algorithms*, Springer Publishing Company, Incorporated 2007
- [13] Melanie Mitchell *An Introduction to Genetic Algorithms* MIT Press, Cambridge, MA 1998
- [14] S. Suwas and R. K. Ray *Crystallographic Texture of Materials* Springer-Verlag London 2014
- [15] Xuan Liu, Andrew P. Warren, *Comparison of crystal orientation mapping-based and image-based measurement of grain size and grain size distribution in a thin aluminum film* Acta Materialia 79, 138145 2014

10 Acknowledgments