Available online at www.sciencedirect.com

ScienceDirect

Ciência & Tecnologia dos Materiais 29 (2017) 8–13

http://ees.elsevier.com/ctmat

Parallel kinetic Monte Carlo simulation of Al_3Sc precipitation

Alfredo Moura[a], António Esteves[b*]

[a] *Institute of Polymers and Composites, University of Minho, Guimarães, Portugal*
[b] *Centro ALGORITMI, University of Minho, Braga, Portugal*

Abstract

The present paper reports the precipitation process of Al_3Sc structures in an aluminum scandium alloy, which has been simulated with a synchronous parallel kinetic Monte Carlo (spkMC) algorithm. The spkMC implementation is based on the vacancy diffusion mechanism. To filter the raw data generated by the spkMC simulations, the density-based clustering with noise (DBSCAN) method has been employed. spkMC and DBSCAN algorithms were implemented in the C language and using MPI library. The simulations were conducted in the SeARCH cluster located at the University of Minho. The Al_3Sc precipitation was successfully simulated at the atomistic scale with spkMC. DBSCAN proved to be a valuable aid to identify the precipitates by performing a cluster analysis of the simulation results. The achieved simulations results are in good agreement with those reported in the literature under sequential kinetic Monte Carlo simulations (kMC). The parallel implementation of kMC has provided a 4x speedup over the sequential version.

Keywords: Al_3Sc precipitation; vacancy diffusion; kinetic Monte Carlo; spkMC; MPI.

1. Introduction

In material sciences, precipitated structures acquire a fundamental role due to the ability of imposing enormous obstacles for dislocation movements within the material structure. The knowledge and application of statistical mechanics, namely the kinetic Monte Carlo method (kMC) [1], will be the prime focus of this paper. kMC will be applied in studying the precipitation phenomenon on an aluminum scandium alloy [2]. The documented work, tackles kMC focusing on parallelism, synchronization between processes, and communication with message passing interface (MPI).

The outcome is a set of software applications that allows us (i) to run kMC simulations on a single or multiple processors, (ii) to apply Density Based Spatial Clustering of Applications with Noise (DBSCAN) technique for data mining outputs [3], and (iii) to perform a comparison between the results obtained with the MC simulations and the values predicted by classical nucleation theory (CNT). The practical results of these applications are (i) simulation reports, (ii) DBSCAN reports about clusters and precipitates, (iii) files for 3D visualization of the simulation (at various simulation stages - snapshots), and (iv) files for 3D visualization of the precipitates.

Therefore the main contributions of the present work to the reviewed literature are, besides the mentioned applications, an accelerated simulation algorithm through the parallelization of kMC with MPI.

2. Background and motivation

The constant increase in available computational resources has been accompanied by a growing interest in modeling and simulation of precipitation at different scales, each one with its own advantages. The number of publications and studies related to the topic of precipitation kinetics at the atomic level has increased too [4]. Examples of materials subjected to such studies are alloy materials, such as Fe-Cu [5], Fe-P-C [6], Fe-Cu-Ni-Si [7], Al-Cu [8]. Aluminum alloys

* Corresponding author.
E-mail address: esteves@di.uminho.pt (A. Esteves)

http://dx.doi.org/10.1016/j.ctmat.2016.01.003

have also their share of published studies and our contribution will focus on the study of an Al-Sc alloy.
The literature has documented the enormous potential of the addition of scandium to aluminum alloys for the use mainly in aerospace and automotive applications as it benefits the alloy in recrystallization inhibition, strength increase, grain size refinement as well as the reduction and elimination of hot cracking in welding [9].
Monte Carlo (MC) simulations have also been used on the study of other phenomena [10], such as grain growth [11], abnormal grain growth [12], thin film deposition and growth [13][14], sintering for nuclear fuel aging [15], and bubble formation in nuclear fuels [16].
There are also several published works on parallel kinetic MC simulations, namely the works reported in [17], [18], [19], [20], [21], [22], and [23].

3. kMC and spkMC theoretical background

The simulation of Al_3Sc precipitation with sequential kMC is fully documented in [2], therefore, this section summarizes the theory as a set of equations, behind the simulation. The transition rate for an *XY* atom to jump into a vacancy, with $XY \in \{Al,Sc\}$, is calculated by Eq. 1.

$$\Gamma_{XY\_V} = \nu_{XY}\, exp\left(-\frac{\Delta E_{XY\_V}}{k_B T}\right) \quad (1)$$

The Al and Sc energy increase, due to the motion from its stable site to the saddle point position, is designated activation energy and is obtained by Eq. 2.

$$\Delta E_{XY\_V} = e_{XY}^{sp} - \sum_{i,n} \varepsilon_{XY\_i}^{(n)} - \sum_{j,n} \varepsilon_{j\_V}^{(n)} \quad (2)$$

As a vacancy site is surrounded by twelve nearest neighbors, twelve jump rates are calculated. They are the jump frequencies Γ_1, Γ_2, until Γ_{12}. One of the twelve frequencies is selected, based on their values and on a random number as expressed by Eq. 3.

$$\sum_{i=1}^{n} \Gamma_i \leq random\ number \leq \sum_{i=1}^{n+1} \Gamma_i \quad (3)$$

Eq. 4 describes the computation of the simulation real time. C_V^{sim} and C_V^{real} are the simulation vacancy concentration and the real vacancy concentration in the Al-Sc alloy, respectively. The ratio of the concentrations is multiplied by the inverse of the accumulated jump frequencies.

$$t_{MC}^{real} = \left(\frac{C_V^{sim}}{C_V^{real}}\right) \times \left(\sum_{i=1}^{12} \Gamma_i\right)^{-1} = \left(\frac{C_V^{sim}}{C_V^{real}}\right) \times t_{MC}^{sim} \quad (4)$$

As it can be observed in eq. 4, the simulation time is rescaled by a factor C_V^{sim}/C_V^{real} to take into account the difference between the simulation and the real vacancy concentrations.

A parallel version of kMC enables multiple events to occur simultaneously. To parallelize kMC we adopted the synchronous parallel generalization of the rejection-free n-fold kMC method documented in [21]. The main steps of spkMC are the following:

1. The accumulated frequency of each sub-domain Ω_k is obtained by Eq. 5 as the sum of n_k individual rates r_{ik}.

$$R_k = \sum_i^{n_k} r_{ik} \quad (5)$$

2. The maximum rate is defined by Eq. 6 and it is also established the relation with the sum of individual rates and the rates of null events (r_{0k}).

$$R_{max} \geq \max_{k=1,\ldots,K} \{R_k\}\ ;\ r_{0k} = R_{max} - R_k \quad (6)$$

The purpose of null events is to eliminate time conflicts between processes in such a way that null events prevent the stochastic trajectory of the simulation to be deeply changed.
Figure 1 shows the frequency rate buildup for several processes. Processes with an accumulated frequency inferior to R_{max} will include null events with rates r_{0k}.

3. The spkMC time increment is defined in Eq. 7, where ξ is a random number between 0 and 1.

$$\delta t_p = -\frac{ln\,\xi}{R_{max}} \quad (7)$$

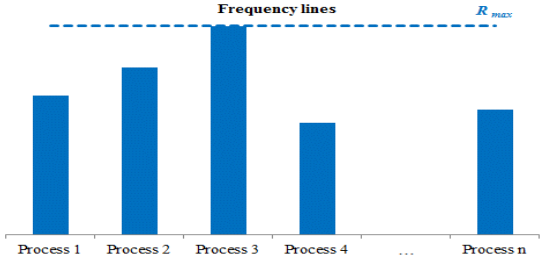
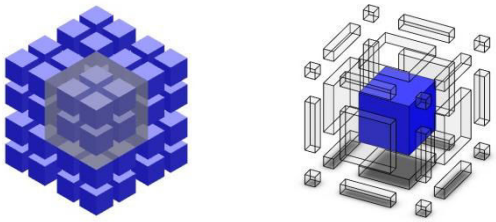


Fig. 1. Sum of frequencies for *n* processes.

4. Implementation of spkMC algorithm

Figure 2 illustrates how the lattice was partitioned into subdomains in spkMC. Each subdomain is further divided into 8 sectors (octants). At the simulation beginning, each sector has a single vacancy. Border conflicts between processes are avoided by a checkerboard scheme. Thus, each sector has a distinct color. The sectors are processed sequentially going through the 8 colors, and always keeping all processes in the sector with the same color. Figure 2 b) illustrates what is designated as the boundary region of a subdomain. The part of a subdomain that is used by other subdomains boundary region is called ghost

region. The depth of the boundary and ghost regions was defined as being half the depth of the neighbor sectors.



a) b)

Fig. 2. 3D lattice partitioning.

For calculating the simulation time, as explained in the previous section, synchronization points between processes were introduced, which are called *end of sprint*. The sprints are a sequence of MC steps without inter-process communication. At the end of a sprint, the altered sites of a boundary and ghost regions are communicated to the correspondent neighbor processes by calling MPI *send*, *receive*, or *put* routines.

The spkMC algorithm was implemented in C language and using the OpenMPI library. During the process of improving the algorithm performance three prototypes were developed: *lock-unlock*, *send-recv*, and *opt-send-recv*. The first prototype uses MPI one-sided communication with lock-unlock synchronization. The second and third prototypes use MPI point-to-point communication. We present next a short description of the optimized *opt-send-recv* prototype.

At the beginning of simulation, the master process sends an extended subdomain to all processes. After that, all processes compute the first and second nearest neighbors for its extended subdomain. Next, the algorithm iterates over (i) the number of simulation sprints, (ii) the number of sectors in a subdomain, (iii) the number of MCSs in a sprint, and (iv), the list of sector vacancies. In each iteration, the following actions take place: (i) the activation energies, the vacancy exchange frequencies, and the real time are calculated; (ii) a first neighbor is selected randomly for new position of the vacancy, (iii) the vacancy is swapped with the selected neighbor, and (iv) the vacancy move is stored. At the end of each sector sprint (i) the list of moves is trimmed, to eliminate those moves that begin and finish with the same type of particle (Al-Al, Sc-Sc, and V-V moves), and (ii) the moves are sent and received to/from the neighbor process in X direction, then in Y direction, and finally in Z direction. At the specified snapshot points, the master process gathers the subdomains from all processes and writes its information to a file.

The material parameters that support the calculation of the activation energies and the vacancy exchange frequencies are the first and second nearest-neighbor pair effective energies, the saddle point energies and the attempt frequencies. The values of these parameters, which were obtained mostly from [24], are: $\varepsilon_{AlAl}^{(1)}$ =-0.56 eV, $\varepsilon_{ScSc}^{(1)}$ =-0.65 eV, $\varepsilon_{AlSc}^{(1)}$ =-0.759+21.0×10^{-6}×T eV, $\varepsilon_{AlSc}^{(2)}$ =0.113-33.4×10^{-6}×T eV, $\varepsilon_{AlV}^{(1)}$=-0.222 eV, $\varepsilon_{ScV}^{(1)}$=-0.757 eV, $\varepsilon_{AlAl}^{(2)}$=0, $\varepsilon_{ScSc}^{(2)}$=0, e_{Al}^{sp}=-8.219 eV, e_{Sc}^{sp}= -9.434 eV, ν_{Al}=1.36×10^{14} Hz, and ν_{Sc}=4×10^{15} Hz.

Clouet et al. [24] compute the vacancy-vacancy pair effective energy through Eq. 8.

$$\varepsilon_{VV}^{(1)} = 2\varepsilon_{AlV}^{(1)} - \varepsilon_{AlAl}^{(1)} - E_{2V}^{bin} \quad (8)$$

where the divacancy binding energy was considered to be E_{2V}^{bin}=0.2 eV. This means that the vacancy-vacancy interaction is attractive. These authors claim it is necessary to perform a deeper work to carry out simulations with several vacancies. Takai and coworkers present repulsive vacancy-vacancy binding energies up to the third nearest-neighbor within aluminum alloys [25]. In the present work, we assumed the value $\varepsilon_{VV}^{(1)}$= -0.084 eV for the vacancy-vacancy pair effective energy, which was computed by Eq. 8.

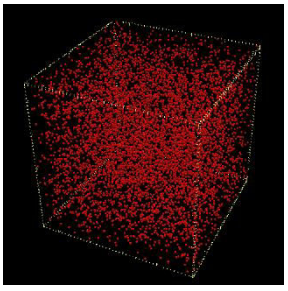
5. Results

The results presented in this section were obtained from simulations run on the SeARCH cluster, located at the University of Minho. Simulation configurations include lattice sizes of 56×56×56 and 104×104×104 FCC cells, 5x10^{11} MCSs, 1% of scandium, temperatures of 673.15K and 873.15K, and 2x2x2 (with a total of 8 vacancies), 4x2x2 (with a total of 16 vacancies), 3x3x3 (with a total of 27 vacancies), and 4x4x4 (with a total of 64 vacancies) process grids.

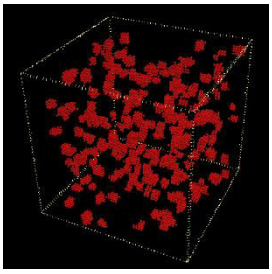
Figure 3 shows that spkMC really simulates Al_3Sc precipitation. Figures 3 b) and c) demonstrate how DBSCAN clustering analysis facilitates the 3D visualization of precipitates. DBSCAN post-processes the simulation raw data and obtains the precipitation metrics reported in figure 4 [2].

The sequence of graphics included in Figure 4 was generated by DBSCAN from the simulation snapshots. The graphs summarize the main metrics concerning the precipitates. Figures 4 a) and b) show the evolution of precipitates size in terms of the average radius, for a simulation temperature of 673.15 and 873.15 K, respectively. Figures 4 c) and d) illustrate the

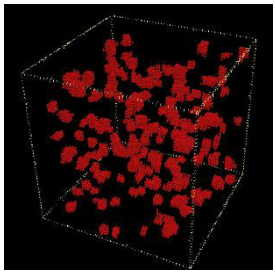
evolution of the presence of scandium in the aluminum solid solution and in precipitates, respectively. Figures 4 e) and f) present the most valuable metric for assessing the precipitation results and, as a consequence, to compare simulation alternatives to each other (spkMC with kMC) and to the CNT. This metric quantifies the number of stable precipitates normalized by the number of lattice sites (N_{sp}/N_s). We can confirm that spkMC produces precipitation results statistically equivalent to those of sequential kMC. At the temperature of 873.15K, which is almost in the limit of the phase diagram region of interest, we can observe some deviations between sequential and parallel simulations results.



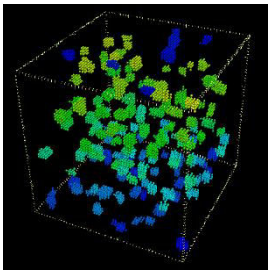
a)



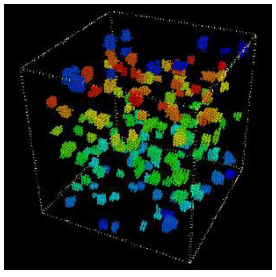
b)



c)

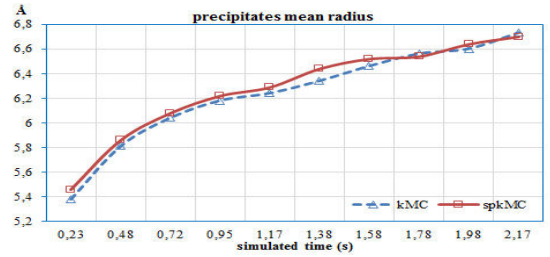


d)

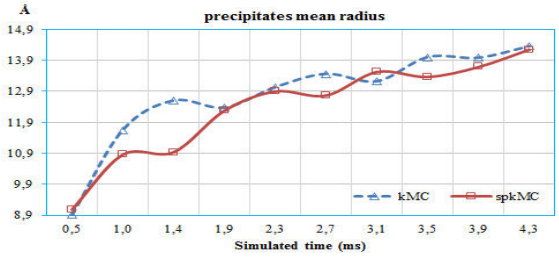


e)

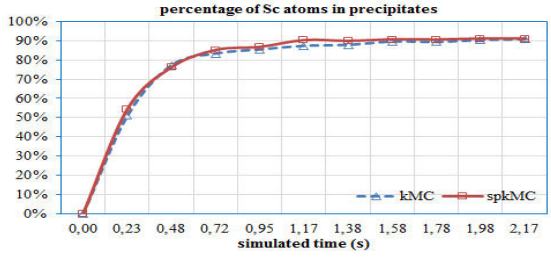
Fig. 3. Simulation evolution with spkMC: a) t = 0s; b) t = 1.17s; c) t = 2.17s. Simulation state comparison at t = 2.17s between d) kMC and e) spkMC.



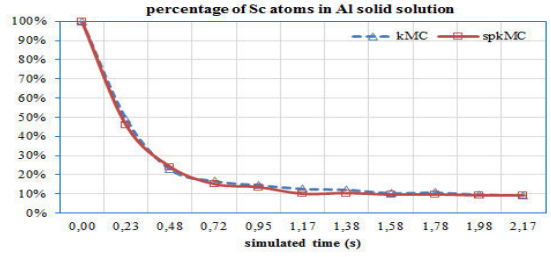
a) Temperature = 673.15 K.



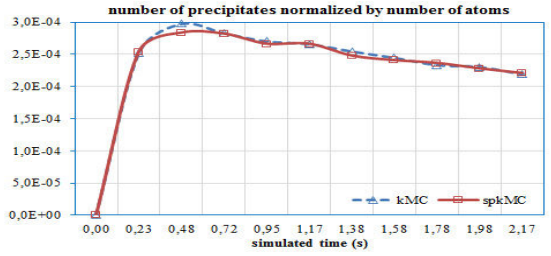
b) Temperature = 873.15 K.



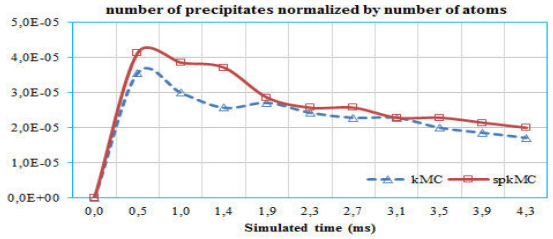
c) Temperature = 673.15 K.



d) Temperature = 673.15 K.



e) Temperature = 673.15 K.



f) Temperature = 873.15 K.

Fig. 4. Comparison between sequential and parallel kMC results.

The work documented in [2] demonstrates a good agreement between kMC and CNT, based on the values of steady-state nucleation rate (J^{st}) and cluster size distribution (C_{nSc}). If we combine this finding with the fact that there is a close proximity between the results obtained with kMC and spkMC simulations, we can infer that the comparison between spkMC and CNT would follow the same trend.

Figure 5 allows to compare the best performance achieved with each kMC implementation: the sequential version and the three different spkMC implementations (*lock-unlock*, *send-recv,* and *opt-send-recv*). The graph also shows that the evolution from the first parallel implementation (*lock-unlock*) to the third one (*opt-send-recv*) attained a speedup of about 7×. In the two non-optimized versions, *lock-unlock* and *send-recv*, increasing the lattice size improves performance. Performance improves because by increasing the length of sprints increases the ratio between computation and communication. Optimized *opt-send-recv* implementation allows a speedup of 4× in relation to sequential kMC. I.e. a simulation that takes 6 days with kMC reduces to 1 day and 10 hours.

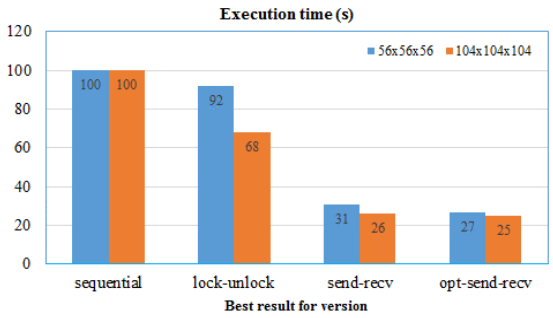


Fig. 5. The minimum execution time necessary to run 10^8 MCSs in the four kMC implementations.

Figure 6 shows the execution time used by spkMC to perform 10^8 MCS. The execution time is displayed according to the number of processors used in the simulation and it was obtained with the *opt-send-recv* implementation of spkMC. As we can see, spkMC performance scales reasonably well up to 16 processors, and almost stabilizes after that.

The parallel efficiency and scalability of the spkMC implementations are low. Defining parallel efficiency as $\eta(N,P) = 100*1/P*(T_{seq}(N)/T_{par}(N,P))$, where N is the problem size, P is the number of processors, $T_{seq}(N)$ is the sequential execution time, and $T_{par}(N,P)$ is the parallel execution time in P processors, the maximum efficiency for the *opt-send-recv* implementation with 8 cores is $\eta = 100*1/8*100/50 = 25\%$. Figure 7 presents the parallel efficiency of spkMC according to the number of processors used in the simulation.

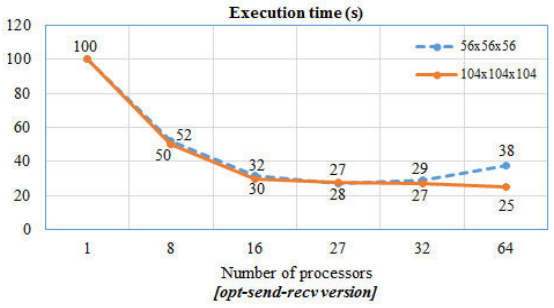


Fig. 6. Execution time necessary for spkMC running 10^8 MCSs.

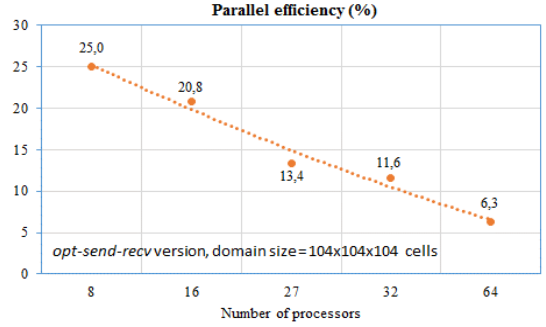


Fig. 7. Parallel efficiency of the spkMC algorithm.

6. Conclusions and future work

Precipitation results obtained with spkMC are in good agreement with sequential kMC. The Al_3Sc precipitates average size increases with the simulation

time. The number of stable precipitates increases in the initial phase and generally stabilizes after this phase. The scandium concentration in the aluminum solid solution, reduces over the simulation time and therefor after some time remains constant at a residual value. Furthermore it is demonstrated that despite the precipitation problem not being embarrassingly parallel, a 4× reduction in execution time with spkMC, when compared to kMC, is an interesting achievement.

Future improvements of spkMC consist in introducing simplifications and approximations in the algorithm, namely to avoid the migration of vacancies between sectors. This simplification eliminates the complexity associated with multiple vacancies management and increases the overall performance by improving the load balancing between processes. As the results proved that inter-process communication is a major performance constraint, it will be developed shared memory (OpenMP) and hybrid (MPI-OpenMP) implementations. It is also planned to adapt the spkMC simulation to other binary and ternary alloys.

Acknowledgements

This work has been supported by FCT - Fundação para a Ciência e Tecnologia within the Project Scope UID/CEC/00319/2013.

References

[1] A.F. Voter, in: K.E. Sickafus, E.A. Kotomin, B.P. Uberuaga (eds), Radiation Effects in Solids, NATO Science Series, vol 235. Springer, Dordrecht, 2007, 1-23.
[2] A. Moura, A. Esteves, Simulation of the Nucleation of the Precipitate Al3Sc in an Aluminum Scandium Alloy using the Kinetic Monte Carlo Method, 13th IEEE Int. Conf. on Nanotechnology, Beijing, China, 5-8th August, 2013.
[3] M. Ester, H.-P. Kriegel, J. Sanders, X. Xu, Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, 2nd Int. Conf. on Knowledge Discovery and Data Mining, 1996.
[4] J. Röyset, Scandium in aluminum alloys overview: physical metallurgy, properties and applications, Metallurgical Science and Technology, Hydro Aluminum R&D Sunndal, N-6600 Sunndalsöra, Norway.
[5] S. Schmauder, P. Binkele, Comp. Mat. Science 24 (2002) 42.
[6] P. Binkele, P. Kizler, S. Schmauder, Atomistic Monte Carlo simulations of the diffusion of P and C near grain boundaries in bcc iron, 30th MPA-Seminar in conjunction with the 9th German-Japanese Seminar, 2004.
[7] K. Betsuyaku, T. Ohnuma, N. Soneda, Progress in Nuclear Sci. and Tech. 2 (2011) 538.
[8] S. Hirosawa, T. Sato, A. Kamio, H. Flower, Acta Mater. 48 (2000) 1797.
[9] Z. Ahmad, J.O.M. 55 (2003) 35.
[10] S. Plimpton, C. Battoile, M. Chandross, L. Holm, A. Thompson, V. Tikare, G. Wagner, E. Webb, X. Zhou, Crossing the Mesoscale No-Man´s Land via Parallel Kinetic Monte Carlo, Sandia Report 2009-6226, 2009.
[11] Aaron Hay, Applying Massively Parallel Kinetic Monte Carlo Methods to Simulate Grain Growth and Sintering in Powdered Metals, MSc Thesis, Naval Postgraduate School, Monterey, CA, 2011.
[12] G. Grest, M. Anderson, D. Srolovitz, A. Rollett, Scripta Met. et Mat. 24 (1990) 661.
[13] Martha Gallivan, Optimization, Estimation, and Control for Kinetic Monte Carlo Simulations of Thin Film Deposition, IEEE Conf. on Decision and Control, 4 (2003) 3437.
[14] P. Zhang, X. Zheng, D. He, Sci. China Ser. G: Phy. & Ast. 46(6) (2003) 610.
[15] C. Cardona, V. Tikare, S. Plimpton, Int. J. Comp. Mat. Sci. Surf. Eng. 4(1) (2011) 37.
[16] Benjamin Holtzman, A New Age of Fuel Performance Code Criteria Studied Through Advanced Atomistic Simulation Techniques, MSc Thesis, University of Illinois at Urbana-Champaign, 2010.
[17] T. Fukuta, K. Ozaki, Trans. Mat. Res. Soc. Japn. 35(2) (2010) 201.
[18] B. Sadigh, P. Erhart, A. Stukowski, A. Caro, E. Matínez, L. Zepeda-Ruiz, Phys. Rev. B 85 (2012) 184203.
[19] G. Arampatzis, M. Katsoulakis, P. Plecháč, M. Taufer, L. Xu, J. Comp. Phys. 231 (2011) 7795.
[20] S. Plimpton, J. Comp. Phys. 117 (1995) 1.
[21] E. Martínez, P.R. Monasterio, J. Marian, J. Comp. Phys. 230 (2011) 1359.
[22] J. Nielson, M. d'Avezac, J. Hetherington, M. Stamatakis, J. Chem. Phys. 139 (2013) 224706.
[23] F. Shi, Y. Shim, J.G. Amar, Phys. Rev. E 76 (2007).
[24] E. Clouet, M. Naster, C. Sigli, Phys. Rev. B 69(6) 064109 (2004) 1.
[25] O. Takai, R.Yamamoto, M. Doyama, Y. Hisamatsu, Phys. Rev. B, 10(8) (1974) 3113.