Triple Hybrid Simulation Method for Tungsten Fuzzy Nanostructure Formation*)

Atsushi M. ITO[1,2], Arimichi TAKAYAMA[1] and Hiroaki NAKAMURA[1,3]

[1]*Department of Helical Plasma Research, National Institute for Fusion Science, National Institutes of Natural Sciences, 322-6 Oroshi-cho, Toki 509-5292, Japan*

[2]*Department of Fusion Science, The Graduate University for Advanced Studies, 322-6 Oroshi-cho, Toki 509-5292, Japan*

[3]*Department of Electrical Engineering, Graduate School of Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan*

(Received 27 December 2017 / Accepted 11 March 2018)

To represent the formation of fuzzy nanostructures produced on a tungsten surface by exposure to a helium plasma, we have developed a hybrid simulation method that combines the binary collision approximation, molecular dynamics, and kinetic Monte Carlo calculations (BCA-MD-KMC). Since the MD code has been parallelized using the domain decomposition method (DDM) for execution in a multi-CPU environment, we developed the BCA code from scratch to mesh it efficiently with the DDM. The BCA-MD-KMC hybrid simulation code achieved a helium irradiation time of 0.1 seconds or longer, in spite of functioning at the level of atomic-scale models. In consequence, we have been able to observe the formation of concave and convex structures on a tungsten surface in the simulation.

Keywords: fuzz, helium plasma, hybrid simulation, binary collision, molecular dynamics, kinetic Monte Carlo, plasma-wall interaction

DOI: 10.1585/pfr.13.3403061

1. Introduction

Irradiation of a tungsten surface by a helium plasma produces fuzzy nanostructures [1, 2] due to the self-agglomeration of helium atoms in the tungsten. The generation of such fuzzy nanostructures has in fact been confirmed in the Large Helical Device (LHD) [3]. There has been some concern that these fuzzy structures may cause problems such as arcing and the generation of microcracks in the material [4], enhancing tritium retention [5] and decreasing the thermal conductivity of the divertor plates [6]. The physical and chemical properties of the fuzzy nanostructures have also been investigated in order to advance plasma applications. In addition, the formation of nanostructures due to helium-plasma irradiation has been confirmed for several metals other than tungsten [7–9].

Molecular dynamics (MD) [10–23], the binary collision approximation (BCA) [21, 24, 25], and several models [26,27] have been employed in simulations aimed at understanding the formation of fuzzy nanostructures. In particular, multi-scale simulations [28] are necessary to treat such complex processes as helium bubble formation and the growth of fuzzy nanostructures. We have previously investigated the hybridization of MD and Monte Carlo (MC) simulations for multi-scale simulations of fuzzy nanostructure formation [29, 30]. In these MD-MC hybrid simulations, we solve the long-timescale diffusion of helium atoms in tungsten using a MC random walk, while we obtain the deformation of the material due to the pressure from the helium bubbles using MD.

However, this MD-MC hybrid simulation approach is insufficient for explaining how the fuzzy structure can become a long fiber because the injection of helium ions is not simulated in real time. Since the MD simulation of the injection process requires longer computational times, the helium atoms were directly inserted into the target material at a depth below the surface corresponding to the penetration depth (range) of the injection process determined beforehand from a BCA simulation. Thus, in this hybrid simulation, the helium atoms start to diffuse from the inserted positions. The same procedure was used to skip the injection process in a full MD simulation of helium bubble formation [31]. However, the change in the injection process between a flat surface and a rough grown surface cannot be represented by this approach because BCA simulations have shown that the penetration depth, reflection ratio, and sputtering yield depend strongly on the surface morphology [24, 25]. In addition, the BCA-MD hybrid simulation has shown that injected helium atoms are easily stopped in helium bubbles [32].

In the present work, we have therefore added the BCA directly into the simulation to deal with the injection process in real time, forming a BCA-MD-KMC hybrid simu-

author's e-mail: ito.atsushi@nifs.ac.jp

*) This article is based on the presentation at the 26th International Toki Conference (ITC26).

lation.

2. Combining the Binary Collision Approximation with the Domain Decomposition Method

In our new BCA-MD-KMC hybrid simulation, the first improvement involves the procedure for inserting helium atoms. The surface structure of the target material and the distribution of the diffusing helium atoms are changed in the MD and the KMC computational phases, while the injection of a helium ion is calculated by using the BCA phase on the fly. In particular, the atomic positions in the target material at any given moment in the MD are employed in the structure of the target material used in the BCA for helium ion injection.

To ensure compatibility of the BCA-MD-KMC hybrid simulation, we have developed the simulation code for the BCA from scratch in the C++ language. The reason for this is due to the following problem encountered in connecting the BCA and the MD. To treat a large sample composed of 10^6 - 10^8 atoms or more on a parallel supercomputer system, a MD simulation is generally parallelized using the domain decomposition method (DDM); in particular, our MD code "GLIPS" uses DDM. In the MD phase of the BCA-MD-KMC hybrid simulation, information about the atoms is distributed among many CPUs. On the other hand, traditional BCA codes [33, 34] only perform serial calculations. Of course, a BCA simulation is sufficiently fast even if it is a serial calculation. However, if a traditional BCA code is to be hybridized with a MD code, the information about atoms distributed among many CPUs for the MD phase must be gathered into a single CPU for the BCA calculation. The communication cost of gathering this information is greater than the calculational cost of the BCA simulation. Moreover, in recent supercomputer systems, the memory size per CPU core/node is not large, because the number of CPU cores is so great. For these reasons, the gathering of information about all the atoms is not appropriate.

In the present work, we have therefore developed a new code, "BCA with DDM on GLIPS (BDoG)." The theoretical models for binary collisions [35] and for the electron stopping power [36] used for the BCA are the same as in the ACAT code [33, 34], while collision detection for projectiles is calculated using a procedure similar to that employed in the MD with DDM in the GLIPS code. Note that "projectile" here means either an injected particle or a recoiled atom.

The method of combining the BCA with the DDM is illustrated in Fig. 1. The decomposition rule for the target material is the same as that used in the MD phase. Simply put, the simulation box is separated into rectangular, parallel, piped domains, and one CPU takes charge of each domain. The separated domains in the BCA phase of the simulation are the same as those in the MD phase. Although the target material is separated in this way, the routine used to track the trajectory of a projectile in principle involves serial processing. Therefore, one CPU follows the projectile within the separated domain to track its trajectory for the binary collision calculation, while other CPUs are resting. When the projectile moves to the next domain, the working CPU begins resting, and the CPU of the next domain starts working to track the projectile.

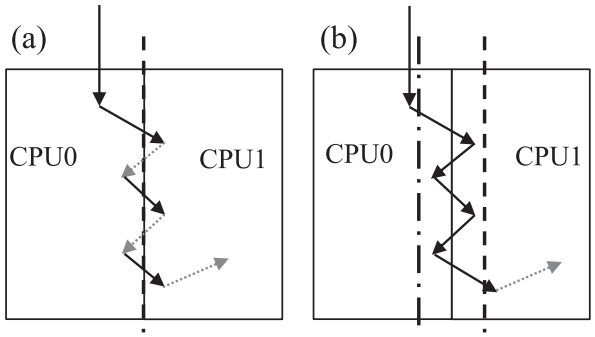

Fig. 1 Combining the BCA with the domain decomposition method. The solid and dotted arrows are the trajectories calculated on CPU 0 and CPU 1, respectively. (a) The boundary for switching between working CPUs (dashed line) is the same as the domain boundary. (b) In the BDoG code, the boundary for switching the calculation from CPU 0 to CPU 1 (dashed line) is set to the right of the domain boundary, while the boundary for switching from CPU 1 to CPU 0 (dot-dashed line) is set to the left of the domain boundary.

A simple approach is to switch the working CPU when the projectile crosses the domain boundary, as shown in Fig. 1 (a). However, this may cause frequent switching of the working CPU because the projectile often follows a small zigzag trajectory along the domain boundary. For instance, channeling along the lattice corresponds to such a small zigzag trajectory. In this approach, the communication cost for switching between working CPUs is high.

In contrast, in the BDoG code, the boundaries for switching between working CPUs are set outside the domain boundary, as shown in Fig. 1 (b). In this way, a small zigzag trajectory can be calculated without the necessity of frequent switching between working CPUs.

This improvement solves both the problem of the cost of the gathering information and the problem of memory resources. The BCA simulation can then be smoothly connected with the MD simulation for the BCA-MD-KMC hybrid simulations.

3. BCA-MD-KMC Hybrid Simulation

The second improvement in the BCA-MD-KMC hybrid simulation from the MD-MC hybrid simulation used in our previous work is that the algorithm for the MC phase of the calculation is replaced by the KMC algorithm.

In the MD-MC hybrid simulation, the MC phase for

the diffusion of helium atoms is simply a random walk of particles on a lattice. The KMC algorithm has two advantages over this MC algorithm:

The first advantage is that the elapsed time required for the simulation can be more reasonably estimated with theoretical adequacy.

The second advantage is that not only can the diffusive migration of helium atoms in tungsten be comfortably treated as an event in the KMC but so also can the injection of a helium ion from the plasma, which is calculated by the BCA. The probability of an injection event in the KMC is defined by the product of the incident flux ϕ and the surface area S. We note that the surface area S used to define the event probability is the cross-sectional area of the simulation box perpendicular to the direction of incidence, rather than the area of the (potentially convoluted) morphological surface. When a diffusion event is chosen by the KMC algorithm, a diffusing helium atom moves to the next site, while when an injection event is chosen, a BCA calculation is performed for the atomic configuration representing the target material at that moment.

These theoretical advantages enable a reasonable representation of the competition between the injection of plasma particles and the desorption of diffusing impurity atoms in the material. These advantages were also confirmed by our previous study, which developed a BCA-KMC hybrid simulation for hydrogen retention in tungsten undergoing plasma irradiation [37].

Execution of the BCA-MD-KMC hybrid simulation follows the flow diagram shown in Fig. 2. First, the simulation system is initialized. In particular, the target tungsten material is positioned at the bottom of the simulation box, and a few impurities which are helium atoms and vacancies are placed within the target material.

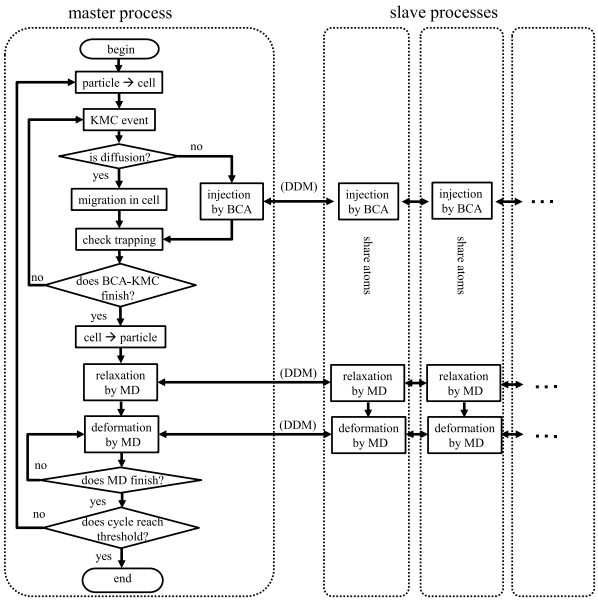

Fig. 2 Flow diagram for the BCA-MD-KMC hybrid simulation.

After initialization, the loop of the BCA-KMC phase begins. In this phase, an event—which may be a migration event of a diffusing helium atom or an injection event of a helium ion—is chosen according to the KMC algorithm. The migration of a helium atom is represented as a migration in the cell system that corresponds to the target material obtained from the atomic particles using the same method as that employed in our previous MD-MC hybrid simulations [29, 30]. If a migration event is chosen, a helium atom moves to the next cell. If an injection event is chosen, a BCA simulation is performed for the injection of a helium atom, where the target material is composed of atomic particles and is not the cell system. If the injected helium ion stops in the target material, it is added to the cell system for migration as a diffusing helium atom. After each event, the program checks to determine whether the helium atom of the event was trapped by a cell corresponding to a helium cluster/bubble. If the helium atom is trapped, the number of trapped helium atoms N_t increases. Once trapped, a helium atom never becomes a candidate for a migration event. If a diffusing helium atom reaches a cell in the vacuum region, it is erased, and the number of desorbed helium atoms increases.

After every event, the code checks to determine whether the condition has been satisfied for terminating the BCA-KMC phase. If N_t becomes smaller than the threshold number of trapped atoms L_t, the BCA-KMC phase continues, and the next event is chosen. When $N_\mathrm{t} \geq L_\mathrm{t}$, the simulation switches into the MD phase to compute the deformation of the target material.

In the MD phase, the helium atoms newly trapped in the BCA-KMC phase are initially inserted into the target material as atomic particles. The displacements of the target atoms following recoil in the BCA phase are also reflected in the new atomic positions of the target material. Before calculating the deformation of the whole target material, only the positions of the inserted helium atoms and the displaced atoms are relaxed, while all the other atoms are held fixed. This relaxation is necessary in order to prevent unnatural repulsion due to the insertion of these atoms into places too close to other atoms.

Next, a MD simulation is performed using a Langevin thermostat at temperature T for an interval of S_MD steps. In the MD phase, the surface morphology is changed due to the pressure from the helium bubbles. Several helium atoms escape into the vacuum region from holes created by "bursting." Therefore, after S_MD steps of the MD simulation, the helium atoms located in the vacuum region are removed from the system. When the number of removed helium atoms N_r is greater than the threshold number of removed helium atoms L_r, a MD simulation is performed again for S_MD steps. When $N_\mathrm{r} \leq L_\mathrm{r}$, the MD phase is finished.

The elapsed time t for the BCA-MD-KMC hybrid simulation is defined as the elapsed time in the BCA-KMC phase, because the timescale for the MD phase is less than

10^{-6} - 10^{-8} times the timescale for the BCA-KMC phase. From a full MD simulation by Kobayashi *et al.* [20], the timescale for "loop punching," which is a typical deformation of the target tungsten material due to the pressure from a helium bubble, is a fraction of a nanosecond or less. Loop punching can therefore be represented by the MD phase in the present hybrid simulation. After each cycle, the elapsed time is compared with the time limit for the simulation $t_{\rm f}$. When $t < t_{\rm f}$, the BCA-KMC phase is started again. The cell information is reconstructed from the current atomic configuration calculated in the MD phase. When $t \geq t_{\rm f}$, the BCA-MD-KMC simulation is finished.

Thus, in the BCA-MD-KMC hybrid simulation, the code cycles between the BCA-KMC phase and the MD phase. The routine described above for conditional branching, and the migration events in the KMC phase, is executed on the master CPU, while the BCA phase and the MD phase are executed on multiple CPUs, according to the DDM. The information about atomic particles is separated into domains that are used in common by the BCA and the MD subprograms.

Finally, we list the parameters necessary to run the BCA-MD-KMC hybrid simulation. The physical parameters are the incident energy $E_{\rm I}$, the incident flux ϕ, the temperature T of the target material, the diffusion coefficient D for the helium atoms in the target material, the atomic parameters for the incident ions and the target material, the size of the target material, and the surface direction. Control parameters are the time step Δt and the number of steps $S_{\rm MD}$ for an interval of the MD phase, the initial ratio He/W of the number of retained helium atoms to the number of tungsten atoms in the target material, the cell size for the migration of helium atoms in the KMC phase, the resistance coefficient γ for the Langevin thermostat in the MD phase, the threshold number of trapped atoms $L_{\rm t}$ used to determine the end of a BCA-KMC phase, and the threshold number of removed helium atoms $L_{\rm r}$ used to determine the end of an MD phase. In general, the time limit $t_{\rm f}$ for the simulation is determined according to the job scheduler of the computer system.

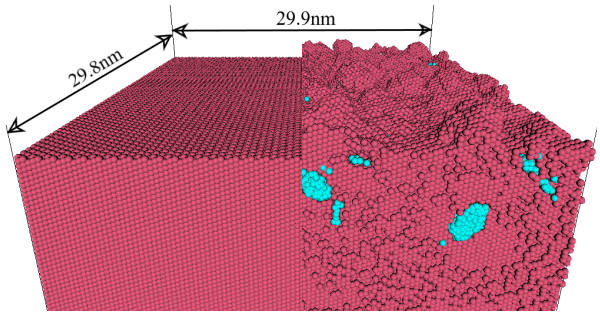

Fig. 3 A snapshot of a BCA-MD-KMC hybrid simulation. The left side shows the initial surface, and the right side shows the surface at 0.13 s. The red and blue spheres are tungsten and helium atoms, respectively.

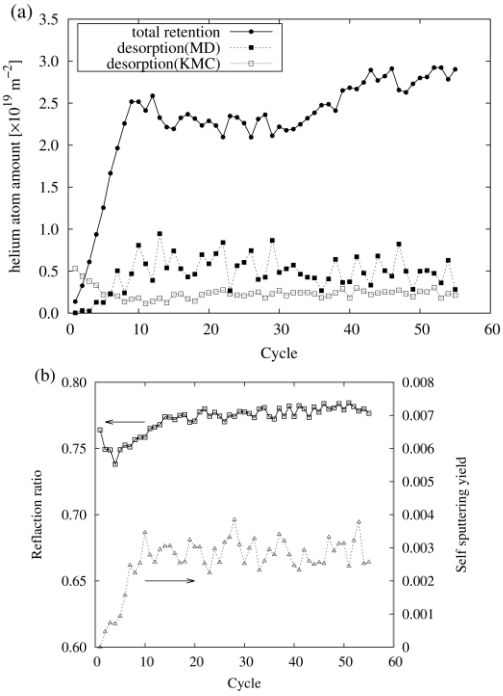

Fig. 4 (a) Total amount of helium retained and the number of desorbed helium atoms in the MD phase and the KMC phase as functions of the number of cycles. (b) The reflection ratio and the self-sputtering yield of helium atoms in the BCA phase.

4. Example

In this section, we demonstrate the BCA-MD-KMC hybrid simulation. The parameters for the present example are $E_{\rm I} = 50\,{\rm eV}$, $\phi = 1.4 \times 10^{22}\,{\rm m^{-2}\ s^{-1}}$, $T = 2000\,{\rm K}$, $D = 1.0 \times 10^{-9}\,{\rm m^2\ s^{-1}}$, the incident particles are helium ions, the target material is tungsten, which has dimensions of 29.9 nm × 29.8 nm × 33.3 nm, with a (111) surface. In addition, $\Delta t = 1.05 \times 10^{-15}$ s, $S_{\rm MD} = 500$, the initial He/W ratio is 10^{-4}, $\gamma = 0.952 \times 10^{11}\,{\rm s^{-1}}$, $L_{\rm t}/S = 10.0\,{\rm nm^{-2}}$, and $L_{\rm r}/S = 0.0\,{\rm nm^{-2}}$. The size of the simulation box in the x and y directions is the same as that of the target material, while in the z direction it is double the size of the target material. Initially, the surface of the target is parallel to the x-y plane and is located at the bottom of the simulation box. We assume periodic boundary conditions in the x and y directions. The tungsten atoms in the bottommost layer are fixed throughout the simulation. In the MD phase, we use an embedded-atom-type potential [15]. We performed the present simulation using 80 CPU cores.

Figure 3 shows the initial surface and the surface at an elapsed time of 0.13 s. The simulation clearly confirms the formation of roughness on the surface, which was produced by the bursting of helium bubbles. We consider this process of roughness formation to be the initial phase of the fuzzy-nanoscale-formation process. Figure 4 (a) shows that the total retention of helium atoms increases rapidly

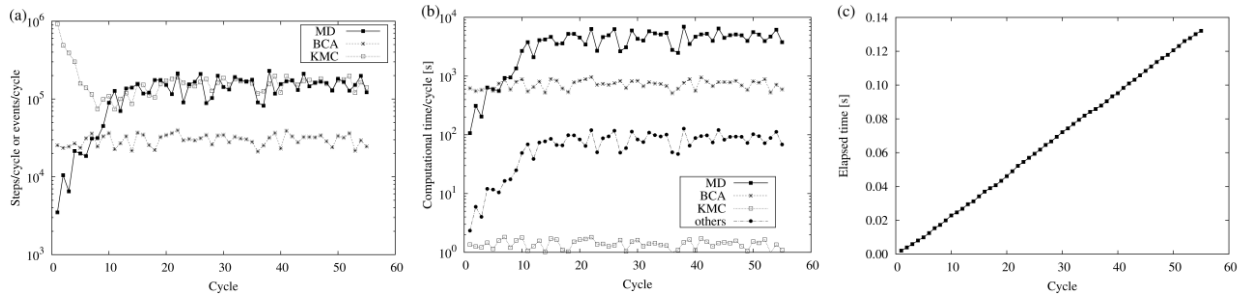

Fig. 5 (a) The number of simulation steps in the MD phase, the number of injection events in the BCA phase, and the number of migration events in the KMC phase as functions of the number of cycles. (b) The computational time for each phase as a function of the number of cycles. (c) The elapsed time as a function of the number of cycles.

up to the 10th cycle, and then the increase slows down. In this connection, the desorption of helium atoms up to the 10th cycle is dominated by desorption in the KMC phase, which is the desorption of single helium atoms diffusing to the surface. Thereafter, the process is dominated by desorption in the MD phase, which is due to the bursting of helium bubbles near the surface.

One experimental measurement [38] found that, before the fuzzy nanostructure was generated, the number of helium atoms retained was $7.5 \times 10^{19}\,\mathrm{m}^{-2}$, while after the growth of the fuzzy nanostructure, the helium retention saturated at $7.0 \times 10^{20}\,\mathrm{m}^{-2}$. In other experimental measurements [39, 40], the helium retention reached $10^{19}\,\mathrm{m}^{-2}$ when the fluence was $10^{20}\,\mathrm{m}^{-2}$, and the helium retention saturated at $5.0 \times 10^{22}\,\mathrm{m}^{-2}$ when the fluence was $10^{22}\,\mathrm{m}^{-2}$ or more. In the present simulation shown in Fig. 4 (a), the helium retention reached $3.0 \times 10^{19}\,\mathrm{m}^{-2}$. Considering that the elapsed time for the present simulation was 0.13 s, at which point the fluence corresponded to $1.8 \times 10^{21}\,\mathrm{m}^{-2}$, we conclude that the present simulation agrees with the experimental results concerning helium retention.

Physical information about the injection process can also be obtained because the injection process is represented by the BCA phase, as shown in Fig. 4 (b). The reflection ratio is 0.73 to 0.78 at the incident energy of 50 eV, and variations in the reflection ratio are smaller than 5 percent. The present reflection ratio, which is derived from the BCA part of the BDoG code, agrees with a BCA simulation using the MARLOWE code [41], with experiments [41], and with a MD simulation [42]. Sputtering, in which a tungsten atom is expelled from the surface of the target material, did not occur in the present simulation. However, the self-sputtering yield, in which a helium atom retained in the target material is sputtered out by an incident helium ion, increases as the number of cycles increases. By comparing Figs. 4 (a) and (b), it seems that self-sputtering yield increases with the total retention amount. Because the energy transfer is a maximum when the projectile and target atoms have the same masses, self-sputtering occurs more easily than the sputtering of tungsten atoms. This also agrees with the simulation result that the incident helium ions almost stop in the helium bubbles, as demonstrated by Saito et al. using the BCA-MD hybrid simulation [32].

We next consider the relationship between computational time and elapsed time from the viewpoint of computational efficiency. In particular, in the hybrid simulation, the load balance between the simulation methods used is important. Figures 5 (a) and (b) show that as the number of cycles increases, the number of simulation steps and the computational time in the MD phase both increase up to about the 20th cycle. These increases are caused by increases in the number of desorbed helium atoms, as shown in Fig. 4 (a). After that, the number of simulation steps in the MD phase becomes almost constant. Although the number of migration events in the KMC phase is comparable to or greater than the number of simulation steps in the MD phase, the computational time in the KMC phase is very small. The second calculation load is generated by the BCA phase. The number of injection events and the computational time in the BCA phase are almost constant. The computational time per injection event in the BCA phase was 0.02 - 0.03 s/shot. Parallelization of the BDoG code evidently was efficiently performed. Figure 5 (c) shows that the increase in the elapsed time is proportional to the number of cycles. From this fact, we can conclude that the BCA-MD-KMC hybrid simulation has good computational performance for long-time simulations.

5. Conclusion

We have developed a BCA-MD-KMC hybrid simulation code with the goal of reproducing fuzzy nanostructure formation. The key point of the present work is that the injection of helium ions is represented by the BCA. This process is hard to calculate by MD simulations because the calculational load is too high, even if recent supercomputer systems are used. In contrast, by using the BCA, many injection events can be calculated in a short time. To connect the BCA into an MD code parallelized for a multi-CPU system, we have proposed a method for adapting the BCA for the DDM, and we have developed it as the BDoG code.

As a result, the elapsed time in a BCA-MD-KMC hy-

brid simulation reached 0.13 s, even though the simulation treated atomic-scale processes. This simulation has confirmed the generation of concave and convex structures on a tungsten surface due to helium irradiation. The present simulation demonstrated the new BCA-MD-KMC hybrid code using a computer with only 80 CPU cores. Because BCA-MD-KMC hybrid simulations will be accelerated using supercomputer systems, we expect that longer elapsed times will enable such simulations to reproduce the formation of fuzzy nanostructures.

Acknowledgments

We thank Dr. Shuichi Kato for helpful comments in terms of the BCA. The present development was performed on "Plasma Simulator" (FUJITSU FX100) of NIFS with the support and under the auspices of the NIFS Collaboration Research program (NIFS16KNSS068, NIFS16KNSS074, and NIFS16KNXN329). This work was supported by JSPS KAKENHI Grant Number JP15H05563.

[1] S. Takamura, N. Ohno, D. Nishijima and S. Kajita, Plasma Fusion Res. **1**, 051 (2006).
[2] S. Kajita, W. Sakaguchi, N. Ohno, N. Yoshida and T. Saeki, Nucl. Fusion **49**, 095005 (2009).
[3] M. Tokitani, S. Masuzaki, H. Kasahara, Y. Yoshimura, R. Sakamoto, N. Yoshida, Y. Ueda, T. Mutoh, LHD Experiment Group and S. Nagata, Nucl. Mater. Energy **12**, 1358 (2017).
[4] S. Kajita, S. Takamura and N. Ohno, Nucl. Fusion **49**, 032002 (2009).
[5] H. Iwakiri, K. Morishita and N. Yoshida, J. Nucl. Mater. **307-311**, 135 (2002).
[6] S. Kajita, S. Takamura, N. Ohno, D. Nishijima, H. Iwakiri and N. Yoshida, Nucl. Fusion **47**, 1358 (2007).
[7] S. Takamura and Y. Uesugi, Appl. Surf. Sci. **356**, 888 (2015).
[8] K. Omori, A.M. Ito, K. Shiga, N. Yamashita, K. Ibano, H.T. Lee and Y. Ueda, J. Appl. Phys. **121**, 155301 (2017).
[9] S. Kajita, T. Nojima, Y. Tomita, N. Ohno, H. Tanaka, N. Yoshida, M. Yajima, T. Akiyama, M. Tokitani and T. Yagi, Surf. Coat. Technol. **340**, 86 (2018).
[10] K.O.E. Henriksson, K. Nordlund and J. Keinonen, Nucl. Instrum. Methods Phys. Res. **B 244**, 377 (2006).
[11] K.O.E. Henriksson, K. Nordlund, A. Krasheninnikov and J. Keinonen, Fusion Sci. Technol. **50**, 18 (2006).
[12] F. Sefta, K.D. Hammond, N. Juslin and B.D. Wirth, Nucl. Fusion **53**, 073015 (2013).
[13] A. Lasa, K.O.E. Henriksson and K. Nordlund, Nucl. Instrum. Methods Phys. Res. **B 303**, 156 (2013).
[14] R.D. Smirnov and S.I. Krasheninnikov, Nucl. Fusion **53** 082002 (2013).
[15] A.M. Ito, Y. Yoshimoto, S. Saito, A. Takayama and H. Nakamura, Phys. Scr. **T159**, 014062 (2014).
[16] L. Hu, K.D. Hammond, B.D. Wirth and D. Maroudas, Surf. Sci. **626**, L21 (2014).
[17] X.-C. Li, X. Shu, P. Tao, Y. Yu, G.-J. Niu, Y. Xu, F. Gao and G.-N. Luo, J. Nucl. Mater. **445**, 544 (2014).
[18] Y.L. Zhou, J. Wang, Q. Hou and A.H. Deng, J. Nucl. Mater. **446**, 43 (2014).
[19] F. Ferroni, K.D. Hammond and B.D. Wirth, J. Nucl. Mater. **458**, 419 (2015).
[20] R. Kobayashi, T. Hattori, T. Tamura and S. Ogata, J. Nucl. Mater. **463**, 1071 (2015).
[21] T.P.C. Klaver, S. Zhang and K. Nordlund, J. Nucl. Mater. **492**, 113 (2017).
[22] G. Wei, F. Ren, W. Qin, W. Hu, H. Deng and C. Jiang, Comput. Mater. Sci. **148**, 242 (2018).
[23] L. Pentecoste, A.-L. Thomann, P. Brault, T. Lecas, P. Desgardin, T. Sauvage and M.-F. Barthe, Acta Mater. **141**, 47 (2017).
[24] H. Nakamura, S. Saito, A.M. Ito and A. Takayama, Plasma Fusion Res. **11**, 2401080 (2016).
[25] J. Drobny, A. Hayes, D. Curreli and D.N. Ruzic, J. Nucl. Mater. **494**, 278 (2017).
[26] S.I. Krasheninnikov, Phys. Scr. **T145**, 014040 (2011).
[27] D. Trufanov, E. Marenkov and S. Krasheninnikov, Physics Procedia **71**, 20 (2015).
[28] B.D. Wirth, K.D. Hammond, S.I. Krasheninnikov and D. Maroudas, J. Nucl. Mater. **463**, 30 (2015).
[29] A.M. Ito, A. Takayama, Y. Oda, T. Tamura, R. Kobayashi, T. Hattori, S. Ogata, N. Ohno, S. Kajita, M. Yajima, Y. Noiri, Y. Yoshimoto, S. Saito, S. Takamura, T. Murashima, M. Miyamoto and H. Nakamura, J. Nucl. Mater. **463**, 109 (2015).
[30] A.M. Ito, A. Takayama, Y. Oda, T. Tamura, R. Kobayashi, T. Hattori, S. Ogata, N. Ohno, S. Kajita, M. Yajima, Y. Noiri, Y. Yoshimoto, S. Saito, S. Takamura, T. Murashima, M. Miyamoto and H. Nakamura, Nucl. Fusion **55**, 073013 (2015).
[31] K.D. Hammond, S. Blondel, L. Hu, D. Maroudas and B.D. Wirth, Acta Mater. **144**, 561 (2018).
[32] S. Saito, H. Nakamura, M. Tokitani, R. Sakaue and K. Yoshida, Jpn. J. Appl. Phys. **55**, 01AH07 (2015).
[33] Y. Yamamura and Y. Mizuno, Inst. Plasma Phys. Nagoya University, IPPJ-AM-40 (1985).
[34] Y. Yamaura and W. Takeuchi, Nucl. Instrum. Methods **B29**, 461 (1987).
[35] W. Eckstein, *Computer Simulation of Ion-Solid Interactions* (Springer-Verlag, Berlin, Heidelberg, 1991).
[36] J.F. Ziegler, J. Biersack and U. Littmark, *The Stopping and Range of Ions in Matter* (Pergamon Press, New York, 1985).
[37] A.M. Ito, S. Kato and H. Nakamura, IAEA Fusion Energy Conference 2016, Oct. 17-22, 2016, Kyoto, MPT/P5-23, proceedings.
[38] R.P. Doerner, M.J. Baldwin, M. Simmonds, J.H. Yu, L. Buzi and T. Schwarz-Selinger, Nucl. Mater. Energy **12**, 372 (2017).
[39] H.T. Lee, A.A. Haasz, J.W. Davis, R.G. Macaulay-Newcombe, D.G. Whyte and G.M. Wright, J. Nucl. Mater. **363-365**, 898 (2007).
[40] P.E. Lhuillier, T. Belhabib, P. Desgardin, B. Courtois, T. Sauvage, M.F. Barthe, A.L. Thomann, P. Brault and Y. Tessier, J. Nucl. Mater. **433**, 305 (2013).
[41] M.T. Robinson, J. Nucl. Mater. **103**, 525 (1981).
[42] V. Borovikov, A.F. Voter and X-Z. Tang, J. Nucl. Mater. **447**, 254 (2014).