Scientific Visualization, 2020, volume 12, number 3, pages 100 - 107, DOI: 10.26583/sv.12.3.09

Visualization of neutron characteristics distribution of debris particles

P.A. Pugachev[1,A], E.V. Bogdanova[2,A], I.S. Saldikov[3,A], A.D. Smirnov[4,A], M.Y. Ternovykh[5,A], G.V. Tikhomirov[6,A], H. Takezawa[7,B], T. Muramoto[8,C], J. Nishiyama[9,C], T. Obara[10,C]

[A] Institute of Nuclear Physics and Engineering, National Research Nuclear University MEPhI
[B] Department of Nuclear Safety Engineering, Faculty of Science and Engineering, Tokyo City University
[C] Laboratory for Advanced Nuclear Energy, Institute of Innovative Research, Tokyo Institute of Technology

[1] ORCID: 0000-0002-8175-3957, PAPugachev@mephi.ru
[2] ORCID: 0000-0002-8184-2124, EVBogdanova@mephi.ru
[3] ORCID: 0000-0002-8667-8706, , ISSaldikov@mephi.ru
[4] ORCID: 0000-0003-0401-2298, ADSmirnov@mephi.ru
[5] ORCID: 0000-0002-9830-5501, MYTernovykh@mephi.ru
[6] ORCID: 0000-0002-5332-7272, GVTikhomirov@mephi.ru
[7] ORCID: 0000-0002-1192-0158, takezawa@tcu.ac.jp
[8] ORCID: 0000-0003-2809-8540, muramoto.t.ab@m.titech.ac.jp
[9] ORCID: 0000-0002-0977-127X, jun-nishiyama@lane.iir.titech.ac.jp
[10] ORCID: 0000-0002-7624-4154, tobara@lane.iir.titech.ac.jp

Abstract

Accident at Fukushima Daiichi nuclear power plant led to increase of importance of safety justification for processes at post-accident facilities in nuclear industry. One of such processes is extraction of corium from reactors cavity. Recriticality of this process is defined by potential unacceptable accident. This paper introduces supporting code for neutron fluxes and reaction rates visualization in systems with complex geometry that can be used in modeling of corium removing works. Visualization code is based on Unreal Engine 4 game engine. Code allows observing neutronic functionals distribution in three dimensions. The reseach and provided implementation details help to understand the physical processes that take place as the accidents occur during corium removing works.

Keywords: fuel debris, Monte Carlo, criticality analysis, corium.

1. Introduction

Fukushima Daiichi accident that took place in March 2011 affected entire industry and raised huge amount of new challenges in field of safety justification of post-accident nuclear facilities. One of consequences of the accident is corium – highly radioactive lava-like remains of reactor core. It is essential to avoid any emergency processes during corium handling operations. One of such dangerous operations is corium extraction.

In present times, some of corium is located at the bottom of reactor cavities of Fukushima's damaged units and being cooled with water, but Tokyo Electric Power Company (TEPCO) is planning to extract corium from its current location with the aim of holding it in proper place. First part of extraction process will include crushing corium to small pieces (debris), second is debris extraction itself, during this stage debris particles can fall to the water and can approach to recriticality [1-3]. Safety justification of such situation is not an easy task; it consists

of CFD modeling of particles falling process in order to get their time-dependent positions and following neutronic Monte Carlo simulation with obtained geometry. Since nobody knows real isotopic composition of corium and its structure i.e. possible layers and other spatial heterogeneities, it is essential to find most limiting conditions of distribution of debris particles and their isotopic composition in order to fully justify safety of such operations. Proper data visualization is a good way to do so.
Aim of this work is to create a tool for visualizing neutron fluxes and reaction rates in fuel debris falling to the water in case of accident during corium extraction procedure. Data set for visualization was obtained at the first part of joint Russian-Japanese work related to safety justification of corium handling operations. The following parts of the aforementioned work will be characterized, inter alia, by the complexity of geometry, so visualization tool should have possibility to work with it.

2. Data preparation

2.1 Falling debris model

Model of interest in this work is the system of one thousand debris particles (cubes with edge equals to 1 cm) which fall to the water on concrete bed. Compositions of debris and concrete are based on compositions for post-accident BWR type reactors. Time dependent positions of falling cubes were obtained with help of ParticleWorks CFD code [4]. Entire falling process takes approximately one second, but only five system snapshots were chosen for further analysis with precise neutronic Monte Carlo codes; these snapshots are enough to perform recriticality analysis because they represent the key states of the system, i.e. states with different uranium-water ratios and debris particle densities. Each cube for all five moments of time has its own position and rotation, they may overlap each other and concrete bed, but it does not have much of impact on simulation. Particle distribution for each time moment is presented in figure 1.

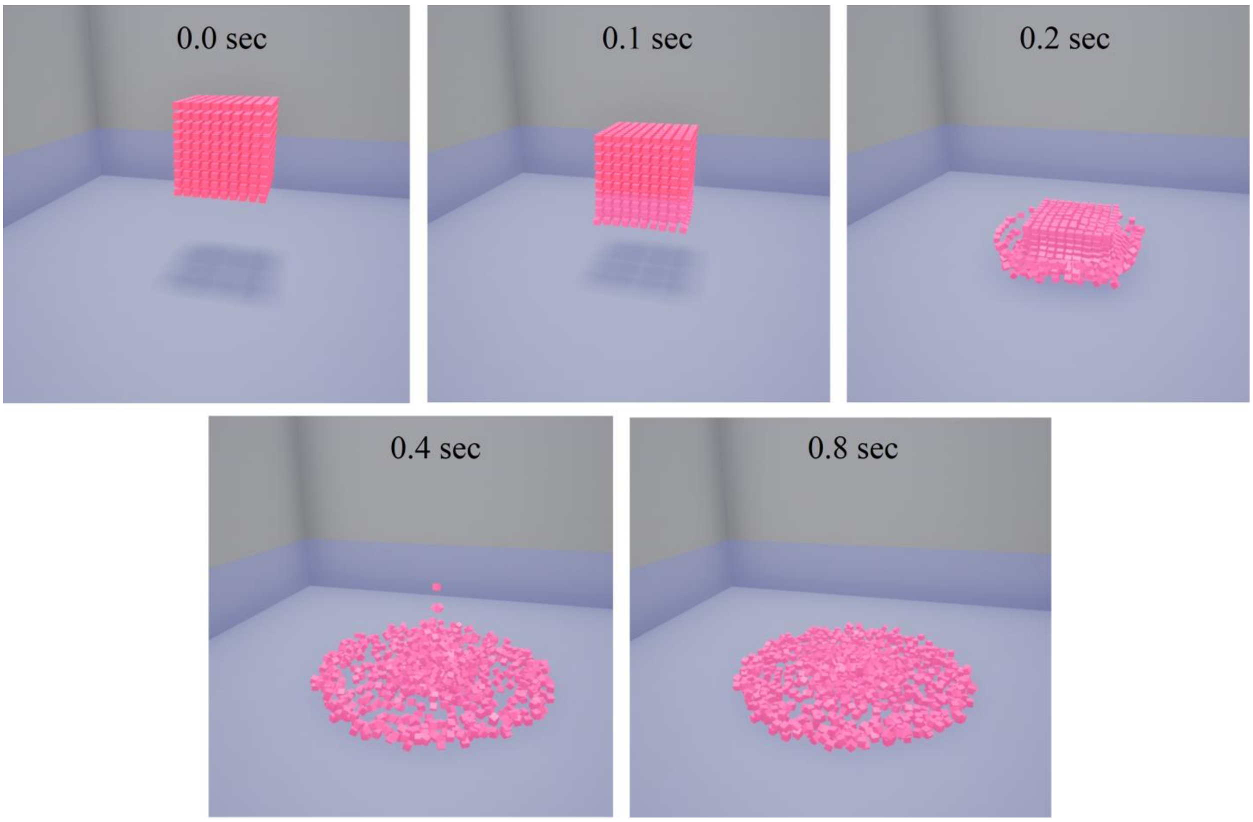

Figure 1. – Particle distribution for each time moment.

As can be seen in figure above, each time snapshot significantly differs from others. From neutronic point of view, it leads to different uranium-water ratio and hence to different multiplication properties of system at each moment of time. Density of particles and heterogeneity of their distribution lead to considerable variation of neutron flux and reaction rates across the system. Visualization of these values can give additional information about the system's state and help to improve current and further analysis of it.

2.2 Neutronic codes and data extraction

Data set for visualization is flux and various reaction rates in debris particles, obtained via precise Monte Carlo simulation, and their standard deviations. Data presented in this work is result of simulation in code Serpent [5] developed at VTT Technical Research Centre of Finland.

Data from Serpent's output files was extracted via simple Python script and saved in format, suitable for further use in visualization software. Such approach (separation of data preparation and visualization process) allows to use results of every Monte Carlo code after proper preparation.

3. Visualization software

In order to visualize data for such a complex system nonstandard approach is required, since it cannot be done with common plotting libraries and environments.

The target function of the visualization software is to render system of cubes giving to each cube color, based on value of interest in corresponding debris particle. Then system can be viewed in 3D or user can make a capture of interesting system's sections. There are many ways to visualize distributions in complex geometry. First is to use graphical engines (e.g. OpenGL) such approach was implemented in [6] to visualize space debris distribution, graphical engines are widely used for visualization of biological structures [7, 8] and for mesh visualization [9], also they may be used as graphical part of complex systems [10] when scientists and developers have no choice but use them. Thus, graphical engines are used when optimization or integration with other software is required. Second approach consists in using 3D modeling software (e.g. Blender) authors of [11] used it to visualize wave function dynamic, also it may be used for visualization of molecular diffusion [12] and dynamic [13]. Third is to use something more high-level, for example, game engines, since using of first two approaches may be not an easy task, because graphical engines are too low-level and require a lot of programming, while 3D modeling software is aimed to solve different kind of problems and, in some cases, not suitable for visualizing purposes.

We have chosen the Unreal Engine 4 for this work due to its scalability and simplicity for render tasks. Unreal Engine 4 is an open source game engine developed by Epic Games [14]. In present time, it is used not only for game development, but also as real-time render engine in filmmaking, architecture etc. Unreal Engine has built in optimization features which are enough for visualization purposes of almost every scale (not for cases mentioned above), a lot of programming work has already been done by engine's developers, thus Unreal Engine is the simplest tool for creating visualization software. In context of scientific visualization, it is usually used when some interactivity is required [15, 16].

User interface of visualization software is presented in figure 2. Via this GUI user can choose moment of time and value to visualize, also it is possible to look at system from every angle and make a picture of desirable section of model. Left bottom part of GUI allows to choose section plane (shown as red grid with an arrow) for further capturing from its position.

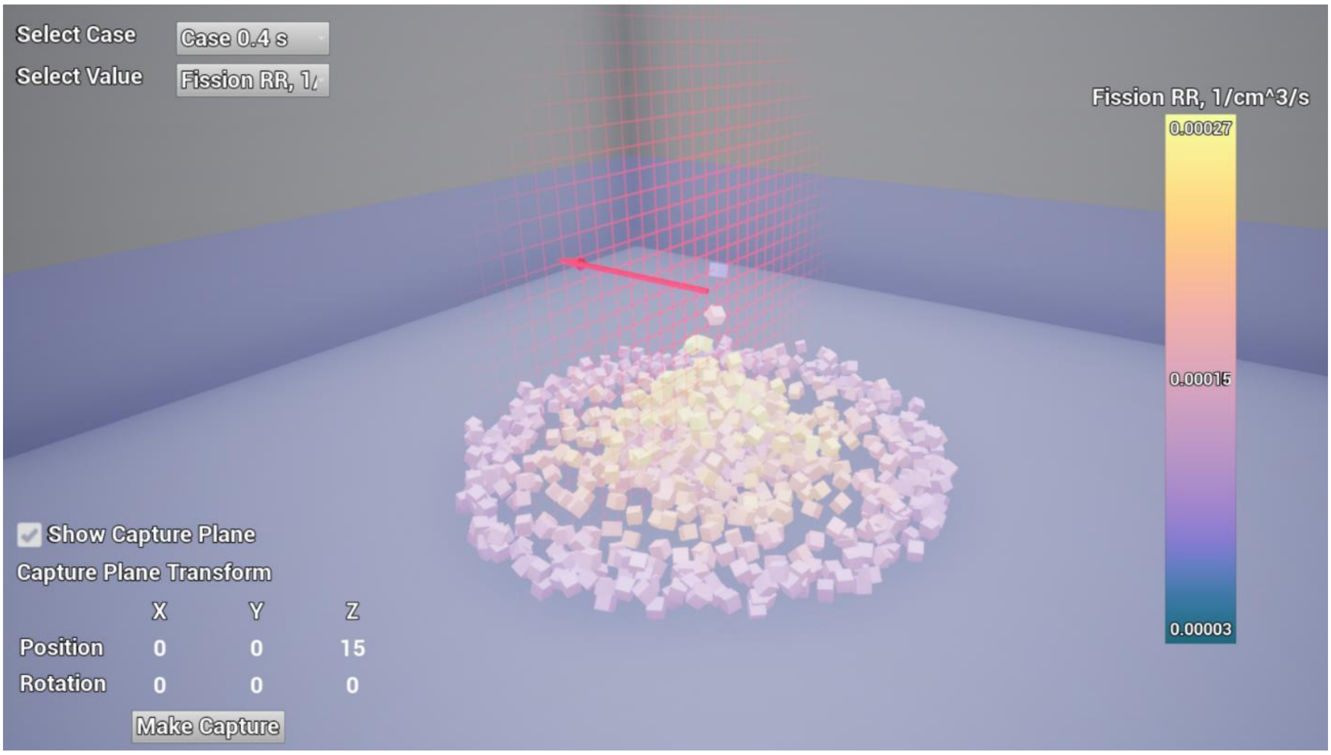

Figure 2. – Graphical user interface.

In the future, on next steps of correlated work, form of debris particles may change to spherical or something more complex; this was taken into account during development process of the software.

4. Results

Integrated neutron flux and fission reaction rate were chosen as most significant and useful data. Their distribution is presented on figures 3 and 4. One can notice debris cubes that overlap concrete bed, this feature of model, which was mentioned above, is the result of direct modeling of falling process.

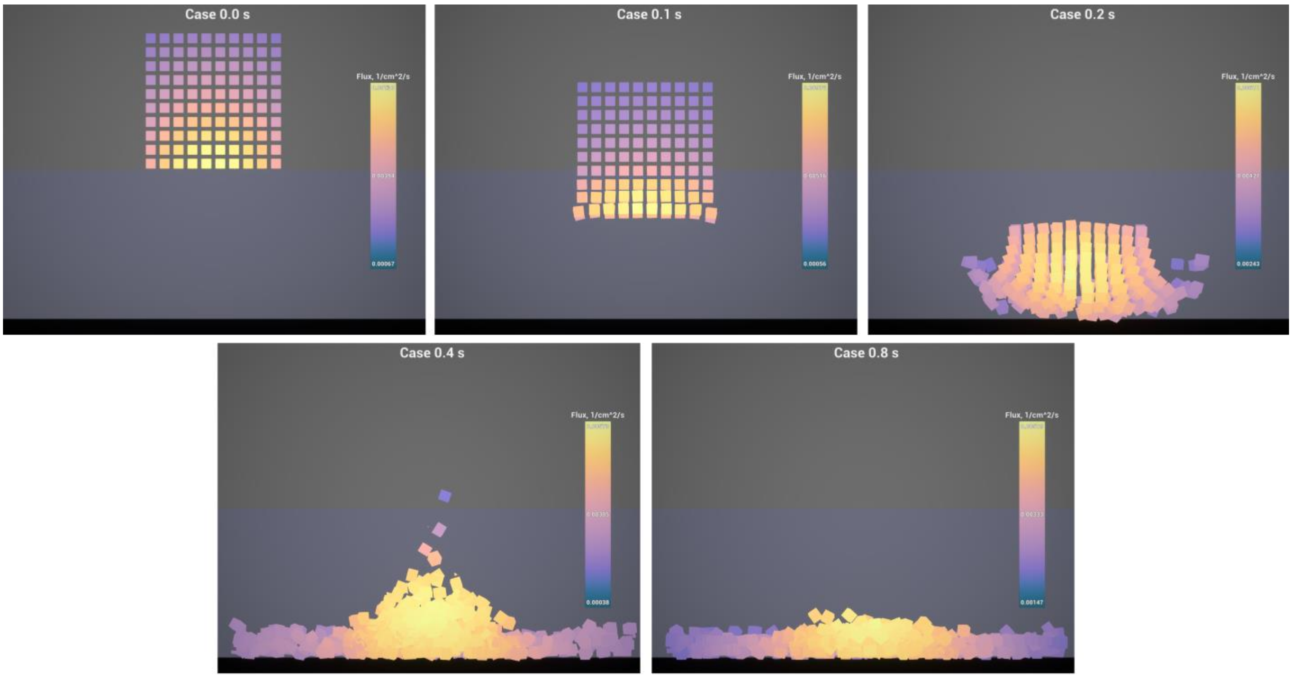

Figure 3. – Integrated neutron flux, YZ plane.

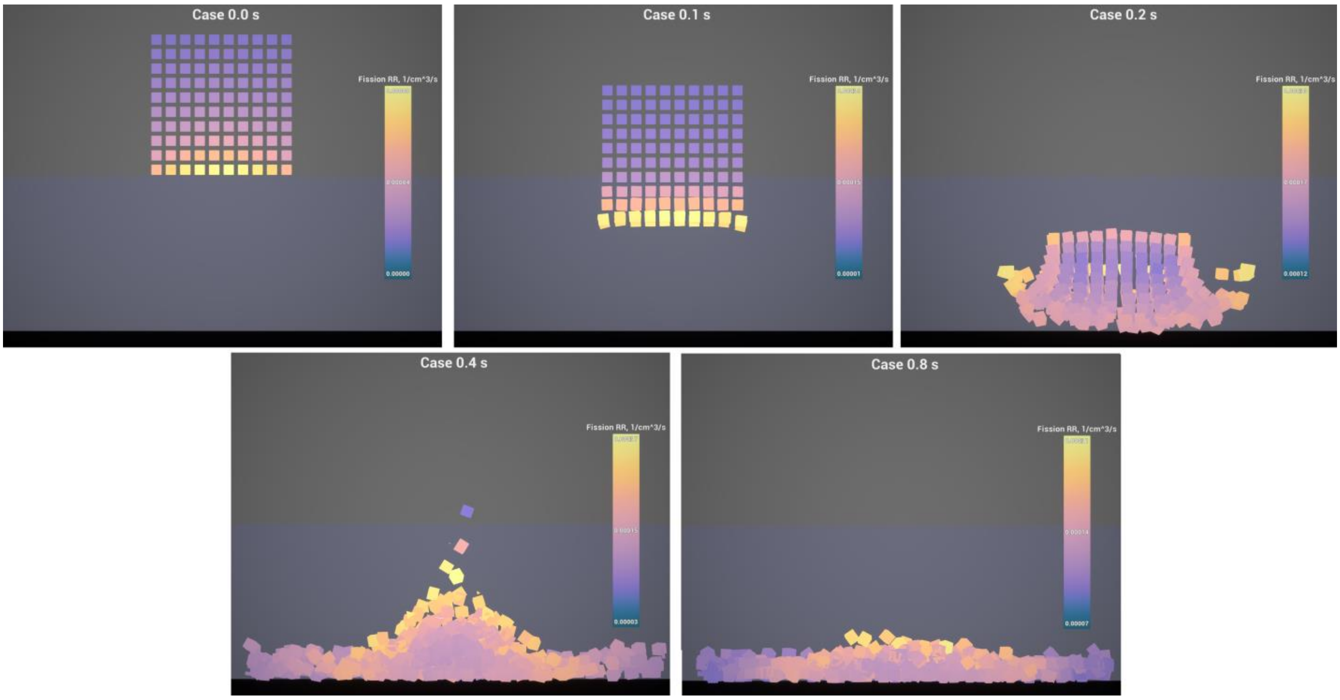

Figure 4. – Integrated fission reaction rate, YZ plane.

As can be seen from figures below, distributions of fluxes and fission reaction rates do not match. This can be explanated by different neutron spectrums across geometry. While integrated flux is higher in the center of debris cluster, fission reaction rate "pike" is shifted towards water, where neutron spectrum is thermal.

Figure 5 shows distribution of standard deviation of integrated neutron flux in each cube. As expected, distribution of this value is reversed comparing with integrated flux distribution.

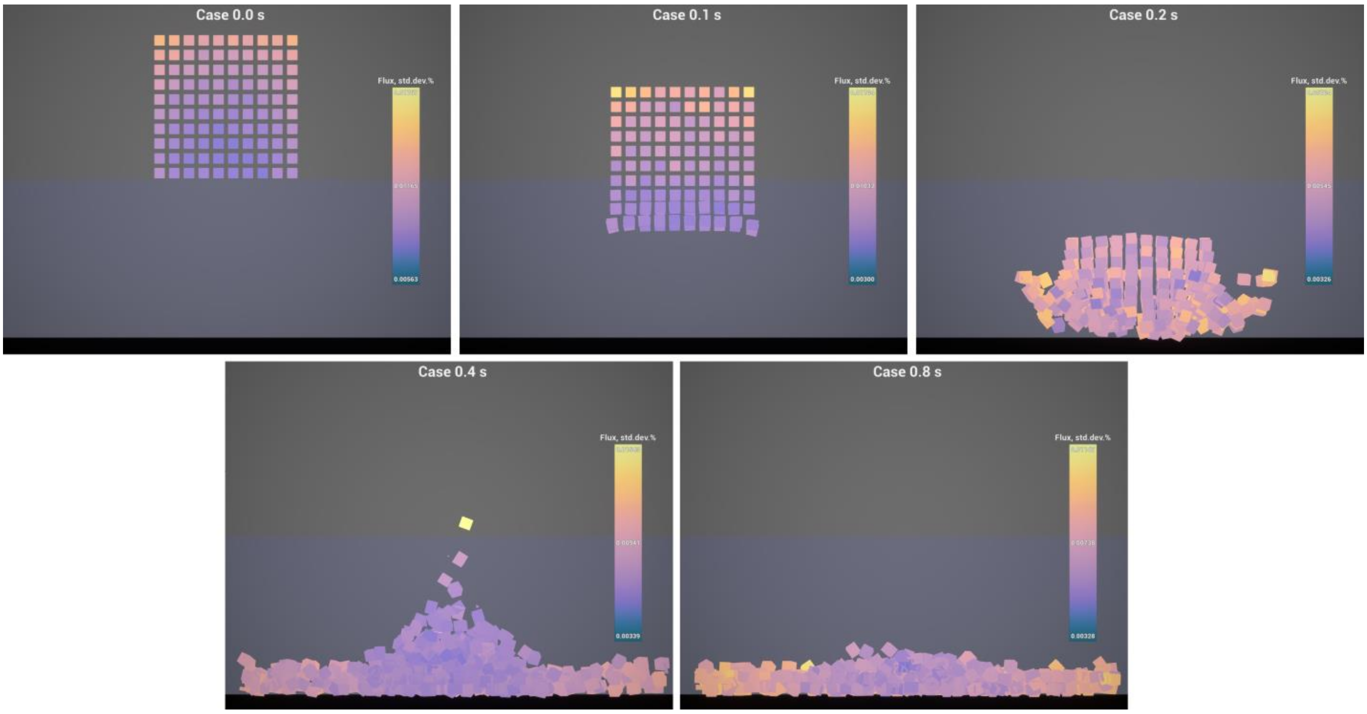

Figure 5. – Standard deviation of integrated neutron flux, percentage, YZ plane.

5. Conclusion

We have developed a new programming tool that targets the visualization of highly distributed neutronic functionals in fuel debris particles. The visualization software allows representing the scientific data of complex systems that is obtained by precise Monte Carlo modeling. Images, obtained via the software, are good illustrative material that helps improving safety justification of corresponding systems. The software was designed with mind of extension for further work.

Acknowledgment

This work was financially supported by the Ministry of Science and Higher Education of the Russian Federation (unique identifier of the project RFMEFI61419X0003).

References

[1] Darnowski P., Potapczyk K., Swirski K. Investigation of the recriticality potential during reflooding phase of Fukushima Daiichi Unit-3 accident. *Ann. Nucl. Energy*, 2017, 99, pp. 495-509.

[2] Freiría López M., Buck M., Starflinger J. A Criticality Evaluation of Fukushima Daiichi Unit 1 Fuel Debris. *Volume 9: Student Paper Competition. American Society of Mechanical Engineers*, 2018.

[3] Tuya D., Obara T. Supercritical transient analysis in hypothetical fuel-debris systems by multi-region approach based on integral kinetic model. *Ann. Nucl. Energy*, 2018, 120, pp. 169-177.

[4] Muramoto T., Nishiyama J., Obara T. Numerical analysis of criticality of fuel debris falling in water. *Ann. Nucl. Energy*, 2019, 131, pp. 112-122.

[5] Leppänen J., Pusa M., Viitanen T., Valtavirta V., Kaltiaisenaho T. The Serpent Monte Carlo code: Status, development and applications in 2013. *Ann. Nucl. Energy*, 2015, 82, pp. 142-150.

[6] Möckel M., Wiedemann C., Flegel S., Gelhaus J., Vörsmann P., Klinkrad H., Krag H. Using parallel computing for the display and simulation of the space debris environment. *Advances in Space Research*, 2011, 48-1, pp 173-183.

[7] Guo D., Nie J., Liang M., Wang Y., Wang Y., Hu Z. View-dependent level-of-detail abstraction for interactive atomistic visualization of biological structures. *Computers & Graphics*, 2015, 52, pp. 62-71.

[8] Palyanov A., Khayrulin S., Larson S. D. Application of smoothed particle hydrodynamics to modeling mechanisms of biological tissue *Advances in Engineering Software*, 2016, 98, pp. 1-11.

[9] Beneš S., Kruis J. Efficient methods to visualize finite element meshes *Advances in Engineering Software,* 2015, 79, pp. 81-90.

[10] Huang J.M., Ong S.K., Nee A.Y.C. Visualization and interaction of finite element analysis in augmented reality. *Computer-Aided Design*, 2017, 84, pp. 1-14.

[11] Figueiras E., Olivieri D., Paredes A., Michinel H. QMBlender: Particle-based visualization of 3D quantum wave function dynamics. *Journal of Computational Science*, 2019, 35, pp. 44-56.

[12] Ghaffar M., Biere N., Jäger D. et al 3D Modelling and Visualisation of Heterogeneous Cell Membranes in Blender. *VINCI '18: Proceedings of the 11th International Symposium on Visual Information Communication and Interaction,* 2018, pp. 64-71.

[13] Rajendiran N., Durrant J.D. Pyrite: A blender plugin for visualizing molecular dynamics simulations using industry-standard rendering techniques. *Journal of Computational Chemistry*, 2017, 39-12, pp. 748-755.

[14] Unreal Engine. https://www.unrealengine.com. Accessed 10 May 2020.

[15] Kiryukhin P., Shcherbakov A., Romanenko V., Pugachev P., Khomyakov D., Tikhomirov G., Zadeba E. Development of a virtual analogue of uranium-graphite subcritical assembly and visualization of the neutron flux distribution in virtual reality. *Procedia Computer Science*, 2020, 169, pp. 192-197.
[16] Horton B.K., Kalia R.K., Moen E. et al Game-Engine-Assisted Research platform for Scientific computing (GEARS) in Virtual Reality. *SoftwareX*, 2019, 9, pp. 112-116.