

USE OF RAY TRACING ALGORITHMS FOR DOSE CALCULATION IN THE VIRTUAL ANALOG OF THE IRT MEPhI REACTOR

Azizov Sh.M.

expert of the State Ecological Expertise, Tashkent

Kiryukhin P.K.

Engineer, Department of "Theoretical and Experimental Physics of Nuclear Reactors", NRNU MEPhI, Moscow

E-mail: cproangel@gmail.com

Abstract

This work focuses on the development of a system for estimating neutron radiation dose loads for the virtual analog of the IRT MEPhI nuclear reactor. This system is implemented based on ray tracing algorithms and is designed to simulate the distribution of dose loads in real-time at any point behind the reactor shielding [3, 4]. The project aims to create a safe and accessible platform for training students in nuclear physics and nuclear reactors, within which the interaction of particles and shielding layers is simulated without the need for access to the experimental facility. This approach not only improves the training level of young specialists but also provides an opportunity to study the physics of radiation processes more deeply by visualizing key aspects of the interaction of neutrons and gamma radiation with the materials of the reactor's protective screens.

Keywords

Virtual analog, nuclear reactor physics, radiation protection, neutrons, gamma quanta, Unreal Engine, IRT MEPhI, Ray Tracing, UDP.

Introduction

The development of virtual models of nuclear reactors is a relevant task aimed at improving safety and accessibility in studying complex physical processes associated with radiation. This paper is devoted to creating a system that calculates neutron and gamma radiation dose loads in a virtual environment based on ray tracing algorithms. The use of the Unreal Engine platform provides the opportunity to integrate powerful graphical and computational tools to create interactive visualization. Thanks to its flexible programming interface and high performance, Unreal Engine allows for effective modeling of particle trajectories, their interaction with materials, as well as the generation and propagation of secondary effects, such as gamma radiation. Additionally, the engine's capabilities support real-time data processing, which is important for the adaptive adjustment of model parameters. The main goal of the work is to develop an adaptive platform that can be used for both educational purposes and scientific research to analyze radiation processes in a safe environment.

Mechanism of the Model

One of the key features of the system is the two-way communication with the virtual reactor control interface via the UDP protocol [5], chosen for its high data transfer speed and



minimal latency. In this model, Unreal Engine interacts with a separate application implemented in C++ via UDP (Fig. 1). This application solves diffusion equations, modeling changes in reactor power and neutron fluxes based on input data. If parameters change in the virtual environment (for example, the position of control rods in Fig. 2), data on these changes are transmitted to the application, where results are calculated using numerical methods and then sent back to Unreal Engine.



Figure 1. Communication via UDP Protocol



Figure 2. IRT MEPHI Control Panel in Unreal Engine

The data exchange process operates at a frequency of 10 packets per second, allowing for the prompt processing of reactor parameter changes and the transmission of results in real-time. Thus, high accuracy, interactivity, and visual clarity of the model are achieved, making it suitable for educational and research tasks. After receiving the data, the program solves the reactor kinetics equations to determine several parameters, including changes in reactor power and their influence on neutron fluxes.

Based on these calculations, the following operations are performed:



1. Determination of neutron flux values.
2. Normalization of the flux to the unit fluence dose array [6] to calculate the specific equivalent dose of neutrons before shielding.

Physical Factors in Calculations

To increase the accuracy and realism of the simulation, the following elements are integrated into the system:

- *Exponential attenuation deviation coefficient.* This coefficient allows for accounting for the specifics of neutron flux intensity changes at small distances from the source.
- *Buildup factor.* This accounts for the specifics of particle scattering and absorption in various reactor materials and the surrounding environment.

Calculation of Dose Loads

The relaxation length method is used to calculate the specific equivalent neutron dose behind protective barriers (1):

$$\dot{H}(d) = \dot{H} B f \exp - \frac{\Delta d_i}{\lambda_i}, \quad (1)$$

where: $\dot{H}(d)$, \dot{H} – specific equivalent doses after and before the shielding; B – buildup factor; f – coefficient of deviation from the exponential law of neutron attenuation; Δd – thickness of the shielding; λ – relaxation length.

All physical parameters of materials are pre-entered into a database, and their identification is carried out using tags [3]. When a trace ray passes through an object, the system retrieves its parameters and uses them for the calculation (1).

The sources of gamma radiation in the model are:

- The reactor core;
- Layers of protective barriers where secondary gamma radiation arises as a result of neutron absorption.

Normalization arrays are used to calculate the gamma radiation dose from the source. Exponential attenuation is also calculated.

To calculate the intensity of secondary gamma radiation from neutrons, the following expression is used:

$$I_\gamma = \frac{1}{2} \varphi_n \frac{\Sigma_{n,\gamma}}{\Sigma_a} \left(1 - \frac{L^2}{\lambda^2} \right)^{-1} \left[-e^{-\frac{d}{\lambda}} E_1 \frac{d}{\lambda} (\mu\lambda - 1) - e^{\frac{d}{\lambda}} E_1 \mu\lambda \frac{d}{\lambda} + \ln \left| \frac{\mu\lambda - 1}{\mu\lambda} \right| + \right. \\ \left. + \frac{L}{\lambda} e^{-\frac{d}{L}} E_1 \frac{d}{L} (\mu\lambda - 1) - e^{\frac{d}{L}} E_1 \mu L \frac{d}{L} + \ln \left| \frac{\mu L - 1}{\mu L} \right| \right]. \quad (2)$$

where: λ – neutron relaxation length in the substance; L – thermal neutron diffusion length; μ – linear attenuation coefficient of the γ -quantum.

$$\frac{\Sigma_{n,\gamma}}{\Sigma_a} \approx 1 \text{ except for } {}^{10}\text{B} \text{ and } {}^6\text{Li}.$$



The exponential integral function E_1 , depending on the thickness size (in centimeters), has two forms [6].

For $d < 5$:

$$E_1(d) = -C + \ln d - \frac{d}{1 \cdot 1!} + \frac{d^2}{2 \cdot 2!} - \frac{d^3}{3 \cdot 3!} + \frac{d^4}{4 \cdot 4!} - \dots, \quad C = 0,5772. \quad (3)$$

For $d > 5$:

$$E_1(d) = F \frac{e^{-d}}{d}, \quad F = 1 - \frac{1!}{d} + \frac{2!}{d^2} - \frac{3!}{d^3} + \frac{4!}{d^4} - \dots \quad (4)$$

Based on the calculated gamma radiation intensity and the specific equivalent dose, the program calculates the contribution of gamma radiation to the total dose.

The total dose is calculated as the sum of dose loads from neutron and gamma radiation (5):

$$\dot{H}_{\text{итоговая}} = \sum_i \dot{H}_n + \sum_k \dot{H}_\gamma. \quad (5)$$

where: \dot{H}_n — dose from neutron radiation; \dot{H}_γ — dose from gamma radiation.

Software Implementation

The model uses ray tracing algorithms to simulate particle trajectories and interactions with materials (Figure 3). Each interaction is processed taking into account the physical parameters of the object obtained from the database. The figures show how the ray tracing changes as a person moves. Through the UDP protocol, the radiation source (reactor) sends 10 rays per second towards the target.

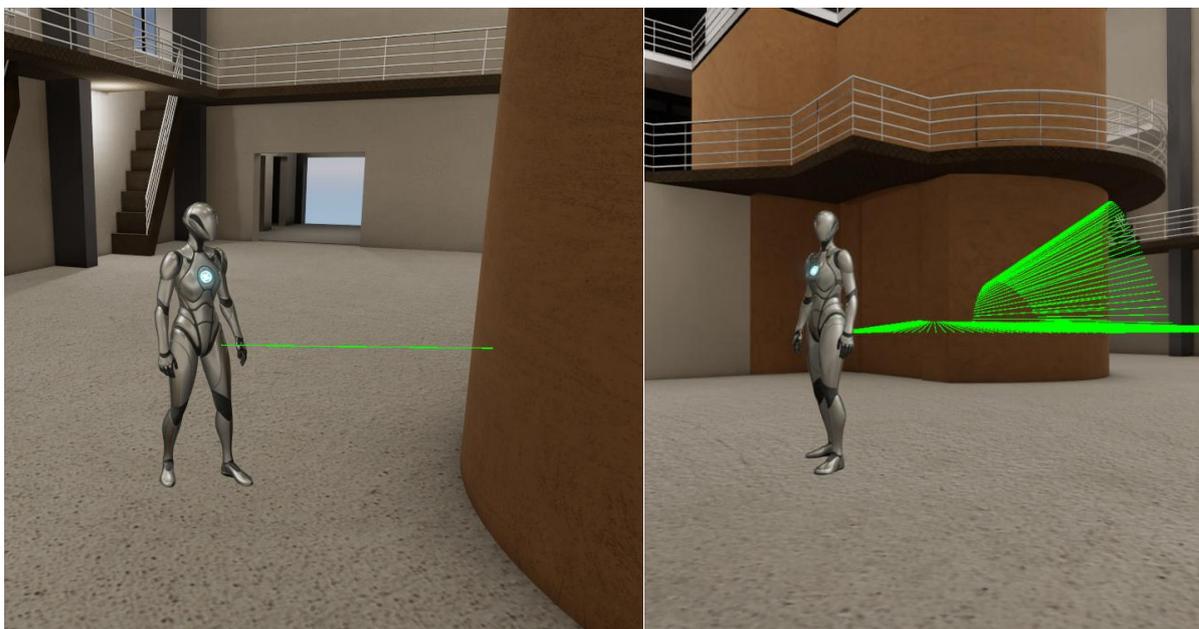


Figure 3. Ray tracing update in real-time mode



The tracing scheme describing the calculation workflow is presented in Fig. 4. A trace ray exits the source; upon each collision with objects, it records coordinates and, using a tag, retrieves parameters of that object, such as relaxation length, diffusion length, density, etc.

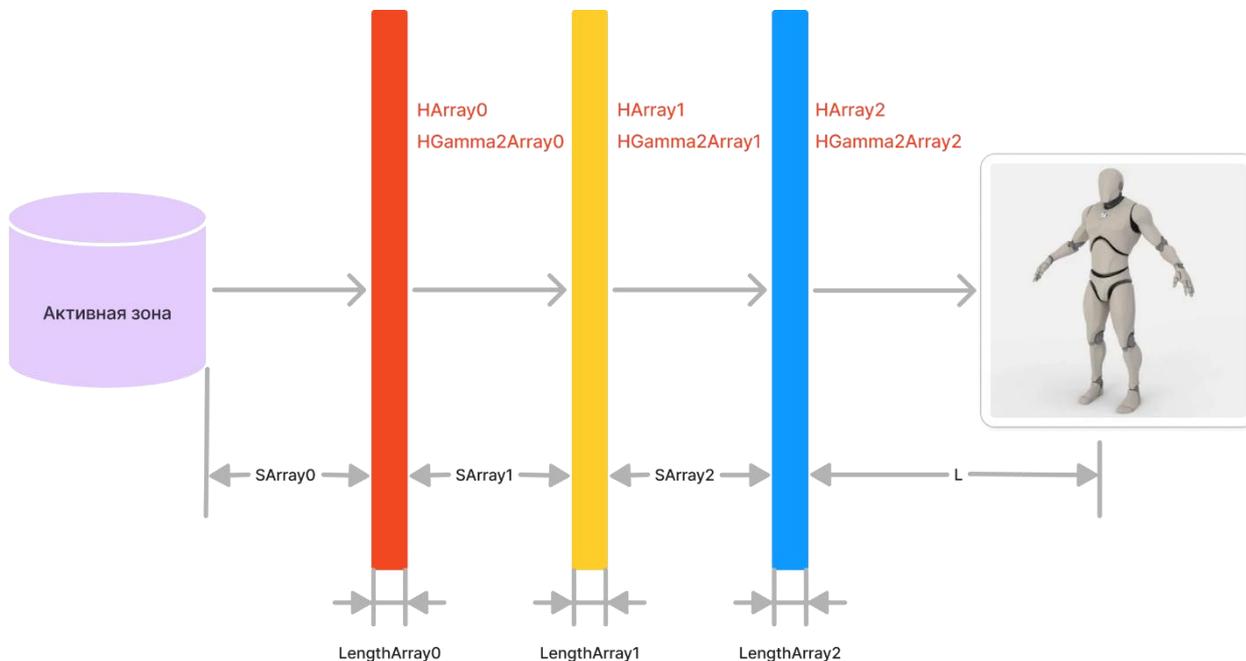


Figure 4. Ray tracing scheme

All this data is collected into arrays and processed in C++ [3]. In the diagram:
 SArray – distance between objects,
 HArray – calculated total dose after this shielding,
 LengthArray – thickness of the object,
 TagArray – various physical quantities, for example, TagArray0 – relaxation length,
 L – distance from the person to the shielding.

To find the final dose, a loop over num (where num is the number of shielding layers) is used when num > 1 (Fig. 5).

```

for (int32 numb = 0; numb < TagArray0.Num(); numb++)
{
    if (numb == 0)
    {
        H1 = TagArray3[0] * 0.0000000000333 * N0 / pow(SArray[0], 2) * exp(-1.0 * LengthArray[0] / TagArray0[0]);
        HArray.Add(H1);
        IGamma2_1 = 1.0/2.0 * N0 / (4.0 * 3.14 * pow(SArray[0], 2)) * (1.0 / (1.0 - pow(TagArray2[0], 2) / pow(TagArray0[0], 2))) * (-1.0 * exp(-1.0 * LengthArray[0] / TagArray0[0]) * TagArray1[0]);
        IGamma2Array.Add(IGamma2_1);
    }
    else
    {
        H = TagArray3[numb] * HArray[numb-1] / pow(SArray[numb], 2) * exp(-1 * LengthArray[numb] / TagArray0[numb]);
        HArray.Add(H);
        IGamma2 = 1.0/2.0 * N0 / (4.0 * 3.14 * pow(SArray[numb], 2)) * exp(-1 * LengthArray[numb] * TagArray1[numb]) * (1.0 / (1.0 - pow(TagArray2[numb], 2) / pow(TagArray0[numb], 2))) * (-1.0 * exp(-1.0 * LengthArray[numb] / TagArray0[numb]) * TagArray1[numb]);
        IGamma2Array.Add(IGamma2);
    }
}

for (int32 m = 0; m < TagArray0.Num(); m++)
{
    HGamma2 = 160.0 * pow(10.0, -13.0) * IGamma2Array[m];
    HGamma2Array.Add(HGamma2);
}
    
```

Figure 5. Part of the code for finding the total dose



Conclusion

The developed virtual model of the nuclear reactor provides unique opportunities for studying radiation processes in a safe environment. The use of ray tracing algorithms implemented on the Unreal Engine platform allows for modeling complex particle interactions to estimate radiation doses. Unreal Engine ensures high visual realism and flexibility in simulation configuration, making the system a powerful tool for both education and scientific research. The integration of modern virtualization technologies and a powerful game engine expands the capabilities of traditional approaches, offering an innovative solution for studying and analyzing nuclear physics.

References

1. García-Hernández R.J., Kranzlmüller D. NOMAD VR: Multiplatform virtual reality viewer for chemistry simulations.
2. Afanasyev V., Kryuchkov E., Petrov V., Saldikov I., Ternovykh M. "Concept of development and use of virtual laboratory works on unique experimental equipment in the field of nuclear power installations". *Vestnik of the National Research Nuclear University MEPhI*. 2012. 1: 111-2.
3. Unreal Engine <https://www.unrealengine.com>.
4. Gusev N.G., Klimanov V.A., Mashkovich V.P., Suvorov A.P. "Physical foundations of radiation protection". Moscow, Energoatomizdat, 1989.
5. UDP Protocol: Main characteristics <https://www.sky.pro>.
6. Golikov I.G. "Lectures on dosimetry and protection". St. Petersburg: Polytechnic University, 2016. 150 p.

