ISSN 1063-7788, Physics of Atomic Nuclei, 2023, Vol. 86, No. 11, pp. 2396–2404. © Pleiades Publishing, Ltd., 2023.
Russian Text © The Author(s), 2023, published in Global'naya Yadernaya Bezopasnost', 2023, Vol. 46, No. 1, pp. 79–91.

MATHEMATICAL MODELING IN NUCLEAR TECHNOLOGIES

Selecting Burnup Algorithms in OpenMC Using the Calculated Benchmark of LEU Assembly and MOX Fuel

H. A. Tanash[a,*], D. A. Solovyov[a,], V. G. Zimin[a,***], A. L. Lobarev[a,****], D. A. Plotnikov[a,*****], and N. V. Schukin[a,******]**

[a] *National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Moscow, 115409 Russia*

**e-mail: tanash_hamza@yahoo.com*

***e-mail: vulture@inbox.ru*

****e-mail: vgzimin@mail.ru*

*****e-mail: lobarev.alexey@gmail.com*

******e-mail: pda1995@gmail.com*

*******e-mail: pikenv@gmail.com*

Received December 15, 2022; revised February 16, 2023; accepted February 21, 2023

Abstract—OpenMC is a state-of-the-art Monte Carlo neutron transport simulation code that uses the Python programming language as an API. OpenMC supports eight burnup simulation algorithms. This study presents the results of choosing an integration method for modeling the burnup of fuel assemblies with burnable poisons for VVER-1000 reactors. Burnup simulation results from OpenMC were compared with those reported in the OECD benchmark. Eight different numerical integrators can be used to model burnup in the OpenMC code: PI, CE/CM, LE/QI, CE/LI, CF4, EPC-RK4, SI-CE/LI, SI-LE/QI. The test results showed that the SI-CE/LI and SI-LE/QI integrators require significantly more time to calculate one burnup step than the others with the same accuracy, so they were excluded from further consideration. The PI integrator showed low integration accuracy at the same burnup steps with other integrators. However, PI has a high performance compared to other integrators, and as the integration step decreases, it converges to one solution, which can be chosen as a reference for assessing the quality of other integrators. On the basis of the results obtained using the fine step PI integrator, it was decided to use the CE/LI integrator for further work. The results obtained with CE/LI were compared with those obtained with the VVER-1000 LEU and MOX benchmark for the codes MCU, TVS-M, WIMS8A, HELIOS, and MULTICELL and showed good agreement. Thus, we can conclude the applicability of the CE/LI integrator as part of OpenMC for modeling the burnup of fuel assemblies containing burnable poisons. During the work, the resources of the high-performance computer center of the National Research Nuclear University MEPhI were used.

Keywords: OpenMC, burnup, integration methods, OECD benchmark, VVER-1000 reactors, neutron transfer method, predictor integrator (PI), CE/LI Integrator, ENDF/B-VII.1, neutron multiplication factor

DOI: 10.1134/S1063778823110431

INTRODUCTION

OpenMC is a Monte Carlo neutron transport modeling code with a Python programming language interface [1]. One of the interesting features of OpenMC is the ability to use various numerical integration algorithms (integrators) in solving nuclear fuel burnup problems. OpenMC can employ the following integrators: predictor integrator (PI), constant extrapolation and constant midpoint integrator (CE/CM), linear extrapolation and quadratic interpolation integrator (LE/QI), constant extrapolation and linear interpolation integrator (CE/LI), fourth-order noncommutator integrator (CF4), extended predictor-corrector with fourth-order Runge–Kutta method integrator (EPC-RK4), stochastic implicit CE/LI (SI-CE/LI), and stochastic implicit LE/QI (SI-LE/QI) integrators [2, 3].

The capabilities for simulating nuclear fuel burnup with the OpenMC code were presented by the authors in [4]. In that study, they performed simulations of a pressurized water reactor (PWR) cell and a sodium-cooled fast-neutron reactor (SFR). Additionally, for the validation of the neutron multiplication factor and nuclide concentration modeling during the fuel burnup process, a comparison was made with results obtained from a similar code, Serpent2, for the VERA test [5]. Thus, the capability of accurately simulating fuel burnup using the OpenMC code was demonstrated.

However, the numerical integration algorithms used in OpenMC for burnup have varying levels of accuracy and computational efficiency. Which method should be used for modeling the burnup of fuel assemblies [6] in VVER-1000 reactors? We will conduct numerical tests and select an appropriate integration method on the basis of the results of these tests.

First, let us provide a brief description of the numerical integration algorithms used in OpenMC for fuel burnup modeling.

1. NUMERICAL INTEGRATION ALGORITHMS FOR FUEL BURNUP MODELING

The burnup of fuel materials during the reactor's fuel cycle occurs owing to ongoing nuclear reactions and spontaneous radioactive decay, which alter the material composition of the fuel. These transformations over time are described by the Bateman equations. The simplest form of the Bateman equation can be written as follows:

$$\frac{dx(t)}{dt} = F(x(t),t)x(t), \tag{1}$$

where $F(x(t), t)$ is the matrix of burnup coefficients at time t and $x(t)$ is the vector of nuclide concentrations.

OpenMC includes eight different numerical integrators for solving Eq. (1): PI, CE/CM, LE/QI, CE/LI, CF4, EPC-RK4, SI-CE/LI, and SI-LE/QI. Let us consider them in more detail [7, 8].

1.1. Predictor Integrator (PI)

The predictor integrator is the simplest numerical method used for solving the Bateman equation. This integrator treats $F(x(t), t)$ as a constant matrix. Under this assumption, the solution becomes simpler as shown in Eq. (2), but at the same time, the accuracy of predicting nuclide concentrations over time is reduced:

$$x(t) = X(0)e^{Ft}. \tag{2}$$

A piecewise-constant approximation is used to estimate $F(x(t), t)$ at the beginning of the time step. Integration is performed as shown in the equation

$$x_{i+1} = x_i e^{F_i h}, \tag{3}$$

where h is the time interval of burnup step and i is the step number.

1.2. CE/CM Integrator

CE/CM is the default integration method in the MCNP6 code. This integration method involves constant extrapolation and a constant midpoint at the predictor and corrector steps. In the CE/CM algorithm, the predictor is carried out to the midpoint, after which the decay matrix is evaluated. Only this new matrix is used throughout the rest of the corrector time step. Equations (4) and (5) describe the solution for this integration method, where y is the vector of nuclide concentrations evaluated at the midpoint of the time step:

$$y = e^{F(x_i,t_i)\frac{h}{2}}, \tag{4}$$

$$x_{i+1} = x_i e^{F\left(y,t+\frac{h}{2}\right)h}. \tag{5}$$

1.3. CE/LI Integrator

CE/LI indicates constant extrapolation at the predictor step and linear interpolation at the corrector step. This integrator is the default method used in the Serpent2 code [11]. When using this integration method, the burnup process takes more time than when using the predictor integrator because the predictor calculation is performed until the end of the time step, and an updated decay matrix is then calculated at the end of the time step. Equations (6) and (7) describe the solution for this integration method. The linear corrector step averages the decay matrix at the beginning and end of the time step, as shown in Eq. (7), which improves the accuracy of predicting nuclide concentrations:

$$y = e^{F(x_i,t_i)h}, \tag{6}$$

$$x_{i+1} = x_i e^{\left[\frac{1}{2}F(y,t_i+h)+\frac{1}{2}F(x,t_i,t_i)\right]h}, \tag{7}$$

where y is the nuclide concentration calculated at the end of the predictor time step. The matrix $F(x(t), t)$ is averaged at both the beginning and the end of the time step during integration with the corrector.

1.4. CF4 Integrator

CF4 represents a fourth-order non-commutator integration method. This integration method has a high fourth-order accuracy and requires the calculation of two exponential matrices. Equations (8)–(15) illustrate how the CF4 procedure works to calculate nuclide concentrations:

$$F_1 = hF(x_0), \tag{8}$$

$$x_1 = x_0 e^{\left(\frac{1}{2}F_1\right)}, \tag{9}$$

$$F_2 = hF(x_1), \tag{10}$$

$$x_2 = x_0 e^{\left(\frac{1}{2}F_2\right)}, \tag{11}$$

$$F_3 = hF(x_3), \tag{12}$$

$$x_3 = x_1 e^{\left(F_3-\frac{1}{2}F_1\right)}, \tag{13}$$

$$x_4 = hF(x_3), \tag{14}$$

$$x_4 = x_0 e^{\left(\frac{1}{4}F_1+\frac{1}{6}F_2+\frac{1}{6}F_3-\frac{1}{12}F_4\right)} * e^{\left(\frac{-1}{12}F_1+\frac{1}{6}F_2+\frac{1}{6}F_3+\frac{1}{4}F_4\right)}. \quad (15)$$

1.5. EPC-RK4 Integrator

EPC-RK4 is a combined integration method that includes a predictor-corrector approach using the fourth-order Runge–Kutta method [5]. This integrator offers high accuracy, even though it belongs to second-order methods. The mathematical expression for this method can be described by the following equations (16)–(22) to obtain nuclide concentrations and matrix exponential:

$$x_1 = x_0 e^{\left(\frac{1}{2}F_1\right)}, \quad (16)$$

$$F_2 = hF(x_1), \quad (17)$$

$$x_2 = x_0 e^{\left(\frac{1}{2}F_2\right)}, \quad (18)$$

$$F_3 = hF(x_2), \quad (19)$$

$$x_3 = x_0 e^{\left(\frac{1}{2}F_3\right)}, \quad (20)$$

$$F_4 = hF(x_3), \quad (21)$$

$$x_4 = x_0 e^{\left(\frac{1}{6}F_1+\frac{1}{3}F_2+\frac{1}{3}F_3+\frac{1}{6}F_4\right)}. \quad (22)$$

1.6. LE/QI Integrator

LE/QI stands for linear extrapolation on the predictor step and quadratic interpolation on the corrector step [4]. This integration method belongs to the family of methods of second order in extrapolation and third order in accuracy of interpolation. LE/QI uses nuclide concentrations from previous time steps, but this requires a significant amount of memory for data storage. The following equations (23) and (24) explain how this integration method works mathematically:

$$y = e^{\frac{-h_j}{2h_{j-1}}F(x_{j-1},j-1)+\left(1+\frac{h_j}{2h_{j-1}}\right)F(x_j,j)}, \quad (23)$$

$$x_{j+1} = x_j e^{\left[\frac{-h_j^2}{6h_{j-1}(h_{j-1}+h_j)}F(x_{j-1},j-1)+\left(\frac{1}{2}+\frac{h_j}{6h_{j-1}}\right)F(x_j,j)+\left(\frac{1}{2}-\frac{h_j^2}{6h_{j-1}(h_{j-1}+h_j)}\right)F(x_{j+1},j+1)\right.}. \quad (24)$$

1.7. Stochastic Implicit Method

In the stochastic implicit Euler method (SI-E), to improve the accuracy and stability of the method, the predictor integration step is transformed into an implicit form. A stochastic gradient descent method is used to determine nuclide concentrations and reaction rates at the prediction stage. The core idea of the method can be expressed mathematically through the following equations (25)–(29):

$$x_{i+1}^0 = x_i e^{F(x_i,i)h)}, \quad (25)$$

$$F(i+1)^n = F(i+1, x_{i+1}^n), \quad (26)$$

$$\bar{F}(i+1)^n = \frac{1}{n+1}\sum_{j=0}^{n} F(i+1)^j, \quad (27)$$

$$x_{i+1}^n = x_i e^{\bar{F}(i+1,x_{i+1}^{n-1})}, \quad (28)$$

$$x_{i+1} = x_i^m, \quad (29)$$

where n is the current substep and m is the number of substeps.

The OpenMC code implement two integrators on the basis of the stochastic implicit methods, SI-CE/LI and SI-CE/QI, which extend the stochastic implicit Euler method to the existing CE/LI and LE/QI integrators, respectively. Both of these methods provide increased stability when solving equations for the burnup of very large systems, where xenon oscillations may occur.

1.8. General Characteristics of Integration Methods

The general characteristics of different high-order integrators implemented in OpenMC are provided in Table 1.

Burnup modeling was performed using only six integrators. The SI-CE/LI and SI-CE/QI integrators were not used because of the extensive time of computation (requiring more than 50 computational steps per burnup step with no results obtained). However, as indicated in [12], the accuracy of K_{eff} calculation when burnup is modeled using SI-CE/LI and SI-CE/QI is comparable to the results obtained using other integrators.

2. VVER-1000 LEU AND MOX BENCHMARK

2.1. Benchmark Specification

The VVER-1000 LEU and MOX benchmark was initially published in 2002 [8]. Many researchers have used it as a reference for conducting comparative assessments [12]. This benchmark comprises two different VVER-1000 fuel assemblies: a homogeneous LEU assembly with 12 uranium-gadolinium fuel rods and a profiled MOX assembly with 12 uranium-gadolinium fuel rods. The hexagonal structure of the LEU

Table 1. Overview of high order numerical integrators

Algorithms	Calculation time per step	Predictor type	Corrector type	Accuracy
PI	1	Constant	None	O(1)
CE/LI	2	Constant	Linear	O(2)
CE/CM	2	Constant	Constant	O(2)
LE/QI	3	Linear	Quadratic	O(2)
EPC-RK4	4	Substep	Substep	>O(2)
CF4	4	Substep	Substep	>O(2)
SI-CE/LI	>50	Constant	Linear	>O(2)
SI-LE/QI	>50	Linear	Quadratic	>O(2)

assembly is shown in Fig. 1. This fuel assembly consists of 300 fuel rods with enrichment in ^{235}U of 3.7 wt % and 12 uranium-gadolinium fuel rods with enrichment in ^{235}U of 3.6 wt % and 4 wt % Gd_2O_3, with a central coolant-filled channel and 18 guide tubes for control rods, also filled with coolant. This benchmark aimed to compute the physical parameters of the fuel assemblies in various conditions using five codes: MCU, TVS-M, WIMS8A, HELIOS, and MULTICELL, with a specific power of 108 MW/m^3 for the entire fuel assembly.

2.2. OpenMC Calculations

This study uses version 0.13.0 of the OpenMC code. The following values of calculated parameters are employed for the assembly calculation: 300000 neutron histories per calculation step; 50 inactive and 100 active calculation steps, respectively. The ENDF/B-VII.1 data files containing neutron cross-section data for nuclides are used. Decay chains containing information on all nuclides are utilized for burnup modeling.

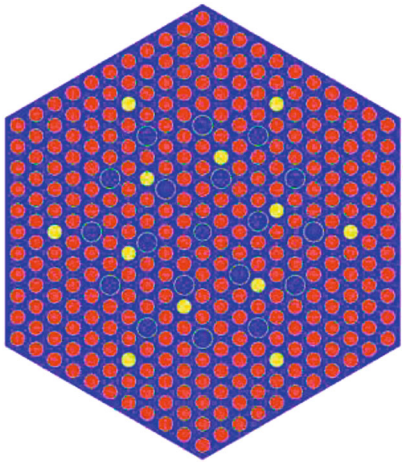

Fig. 1. LEU fuel assembly design.

At the first stage of the study, the results of burnup modeling using various integrators within OpenMC were compared, and the best integrator was selected. The choice of integrator was based on a comparison of accuracy and computational speed. At the second stage, the results obtained using the chosen integrator were compared with the results from the VVER-1000 LEU and MOX benchmark.

Burnup calculations were performed for the S1 state of this benchmark at the fuel temperature of 1027 K and for the coolant and cladding temperatures of 575 K, with equilibrium concentrations of ^{135}Xe and ^{149}Sm. The coolant density was set to 0.7235 g/cm^3 with a boron concentration of 0.6 g/kg. The volumetric power release in the assembly was set to 108 MW/m^3. The burnup calculation was carried out up to a burnup depth of 40 MWd/kg heavy metal (MWd/kgHM). The following burnup steps were chosen: initially, 15 steps of 1 MWd/kgHM were used, followed by 5 steps of 5 MWd/kgHM. To accurately model the burnup of the fuel rods with gadolinium, their fuel pellets were divided into ten equal-area radial zones.

3. RESULTS

3.1. Neutron Multiplication Factor

The variation of K_{eff} with burnup for the LEU fuel assembly using different burnup algorithms in OpenMC for the S1 state is presented in Fig. 2. All the numerical integration algorithms used in OpenMC for modeling nuclear fuel burnup showed approximately the same trends, except for the PI integrator, especially during the initial 10 MWd/kgHM and at the end of the burnup process, as shown in Fig. 2.

The reduced accuracy of the PI integrator is attributed to the larger burnup step, which was 1 MWd/kgHM for the range from 1 to 15 MWd/kgHM and 5 MWd/kgHM for the range from 15 to 40 MWd/kgHM. The low accuracy at the beginning of the burnup modeling is associated with inaccuracies in determining the concentrations of gadolinium isotopes, ^{155}Gd and ^{157}Gd, at such a large burnup step for PI [13]. The results of modeling the concentrations of

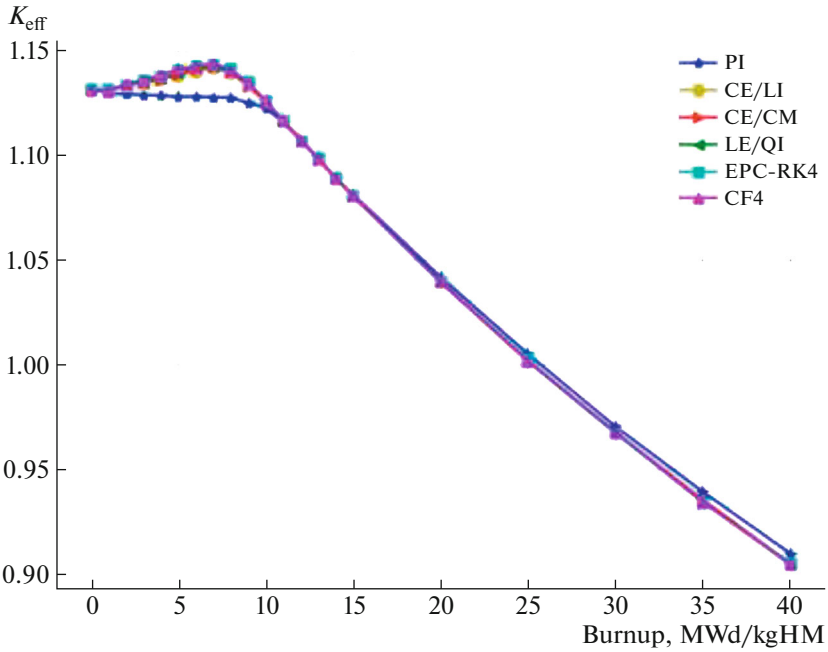

Fig. 2. Variation of K_{eff} as a function of LEU fuel burnup when using different burnup algorithms in OpenMC.

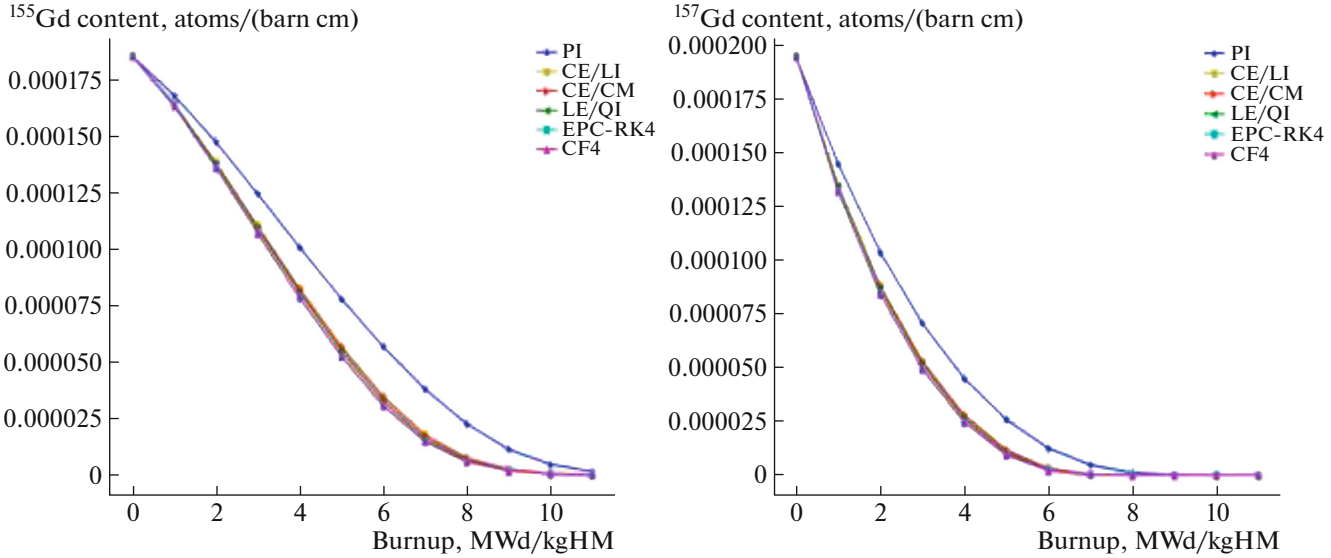

Fig. 3. Concentrations of gadolinium isotopes ^{155}Gd and ^{157}Gd at the start of the burnup modeling.

gadolinium isotopes are shown in Fig. 3. The deviations in K_{eff} modeling using the PI integrator at the end of the burnup process are related to errors in determining the concentrations of uranium isotopes, ^{235}U and ^{238}U, as shown in Fig. 4. However, the PI integrator offers high computational efficiency compared to other integrators. Therefore, we attempted to decrease the PI integrator step to assess whether this would result in significant speed improvements while achieving similar accuracy to other integrators.

To determine the appropriate burnup step for modeling burnup with the PI integrator, calculations with varying smaller burnup intervals were performed. For each new series of calculations, the step was halved. The search for the step was repeated until the results for the current series no longer significantly differed

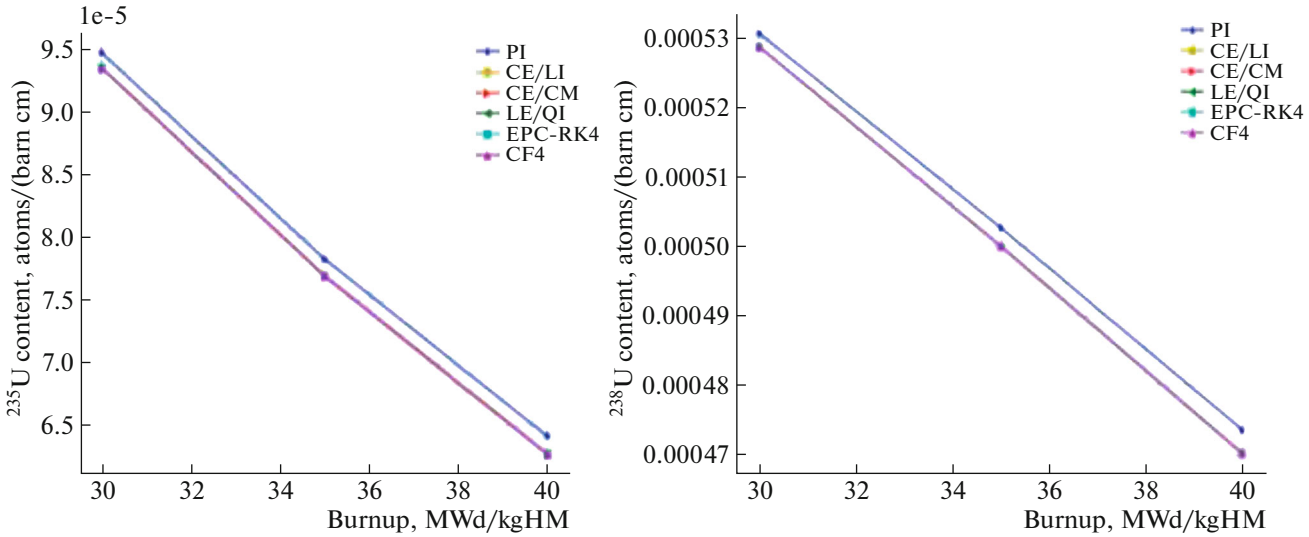

Fig. 4. Concentrations of uranium isotopes ^{235}U and ^{238}U at the end of the simulated burnup process.

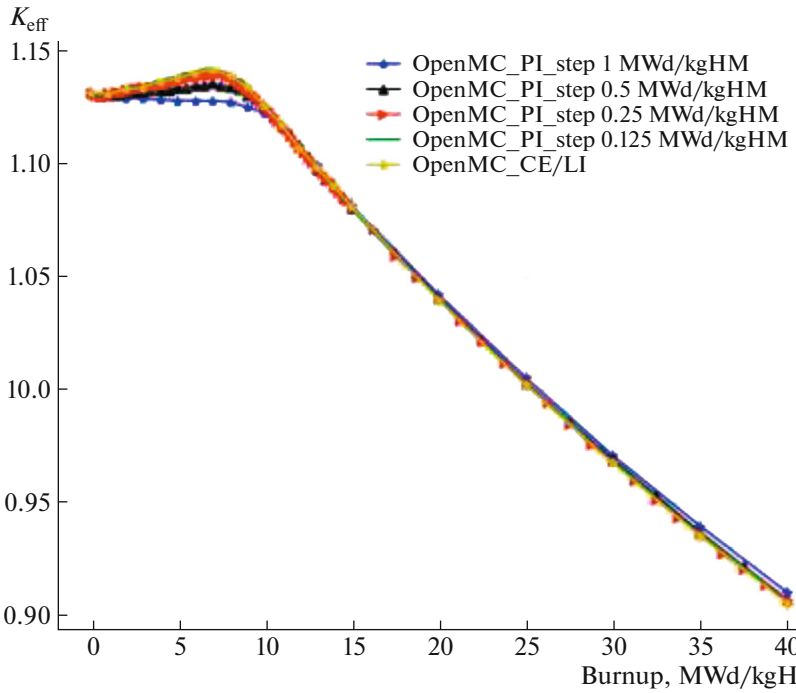

Fig. 5. Modeling of K_{eff} as a function of LEU fuel assembly burnup depth at different values of the calculated integrator step PI.

from the results of the previous series. The results of the step selection are presented in Fig. 5. It can be seen that, as the step decreases, the results obtained using the PI integrator begin to approach the results obtained using other integrators (Figs. 6 and 7).

As the integration step decreases, the results given by the PI integrator converge to the correct solution. This allows for the selection of the most accurate and efficient integrator. The results obtained with the other integration methods in OpenMC were compared to the results obtained with the PI integrator at a smaller step size for further evaluation. For the comparison, the K_{eff} modeling results for the S1 state of the LEU assembly were chosen. The results of comparison are presented in Fig. 8. The relative differences of the results obtained with other integrators compared to PI are evaluated in pcm units $\left(\text{pcm} = \left(\frac{K_{\text{eff}_i} - K_{\text{eff}_{\text{PI}}}}{K_{\text{eff}_{\text{PI}}}} \right) \times 10^5 \right)$. CF4 and EPC-RK4 showed the same values of deviations

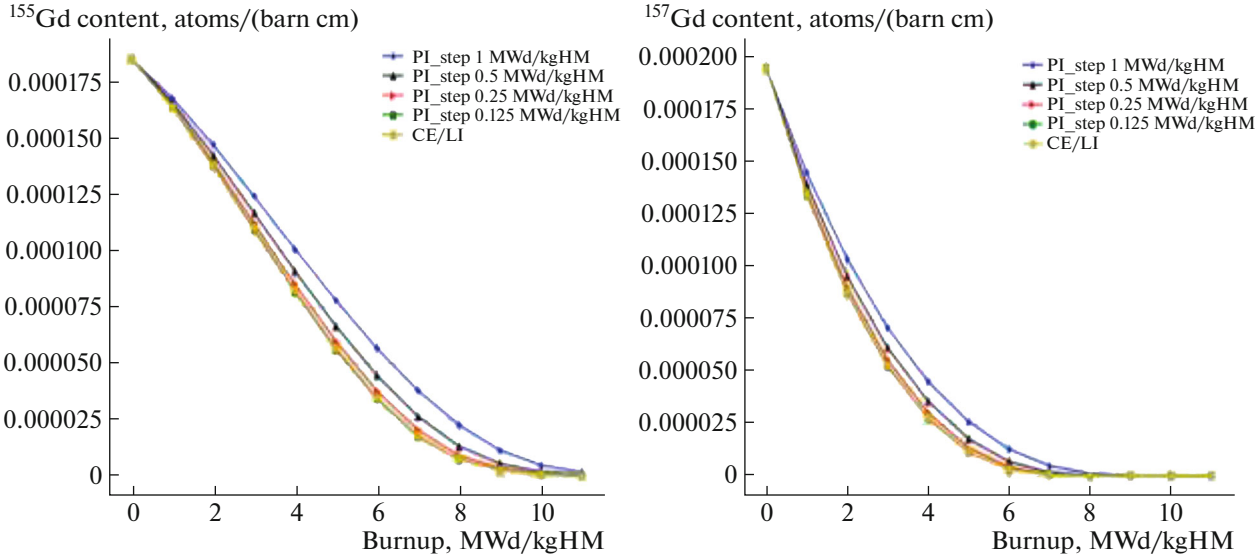

Fig. 6. Concentrations of gadolinium isotopes ^{155}Gd and ^{157}Gd at the beginning of LEU fuel assembly burnup modeling at different values of the calculated integrator step PI.

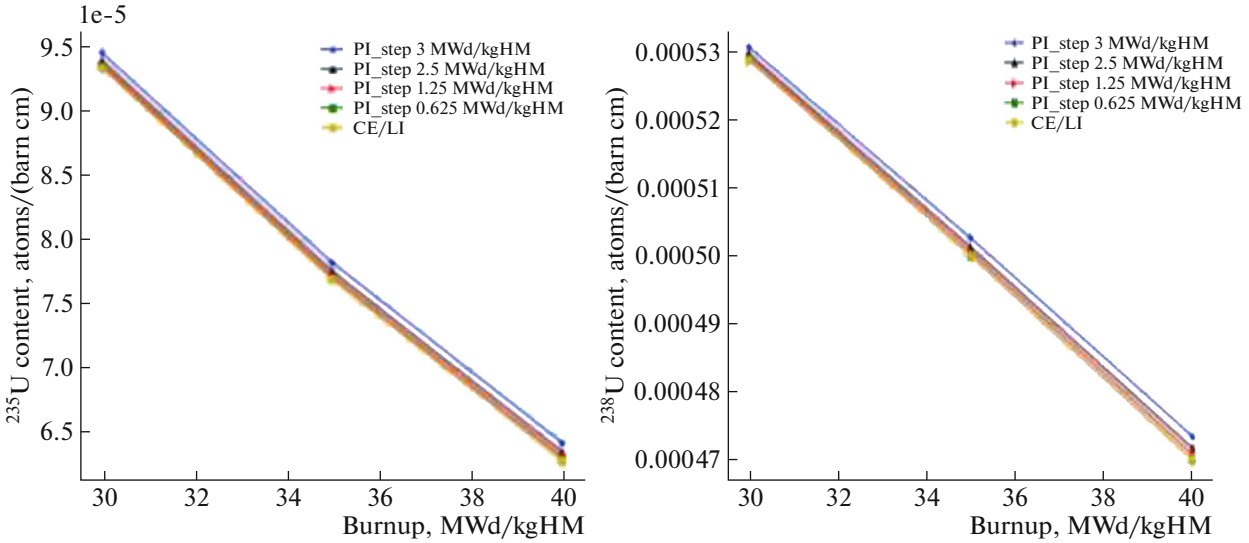

Fig. 7. Concentrations of uranium isotopes ^{235}U and ^{238}U at the end of the LEU fuel assembly burnup modeling at different values of the calculated integrator step PI.

from PI in the range of [–150, 240] pcm, while the deviations from PI for CE/LI and CE/CM were smaller, falling within the range of [–100, 100] pcm. CE/LI was chosen as the preferred integrator because it showed less deviation from PI in the initial phase of the burnup modeling (the region of gadolinium burnup) than CE/CM.

In addition, it is worth noting that using the PI integrator for modeling the burnup of assemblies containing burnable absorbers proves to be inefficient because reducing the step by a factor of 4 does not allow it to achieve comparable accuracy with other integrators. Additionally, it requires more time for calculation, as the CE/LI integrator takes 2 units of time for this calculation instead of 4.

3.2. Comparison with the VVER-1000 LEU and MOX Benchmark

In the VVER-1000 LEU and MOX benchmark, K_{eff} modeling during burnup for the LEU assembly was performed using five different codes: MCU, TVS-M, WIMS8A, HELIOS, and MULTICELL. On the basis of the results obtained from these codes, an average

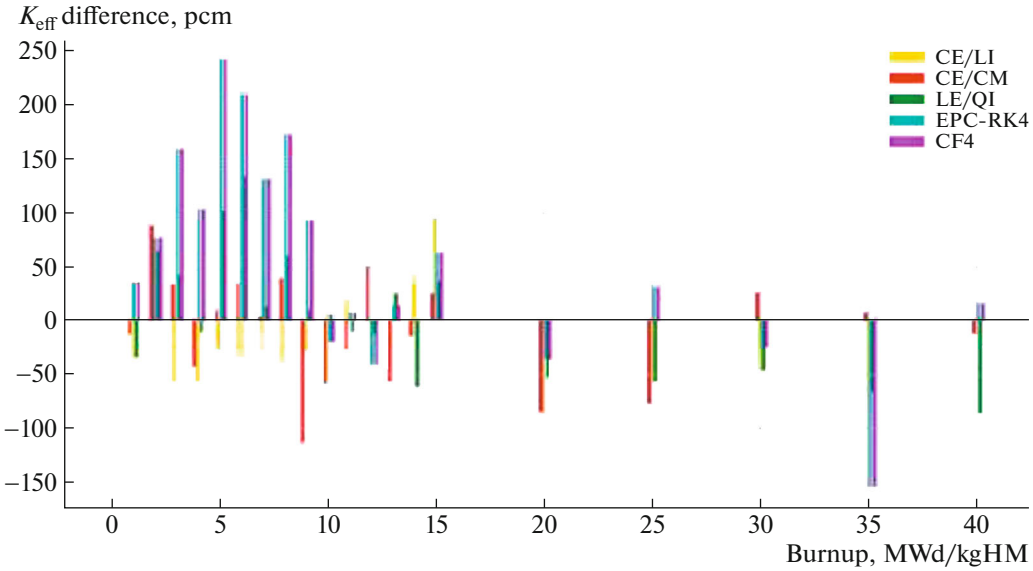

Fig. 8. Deviations of K_{eff} for OpenMC integrators from PI with small burnup step.

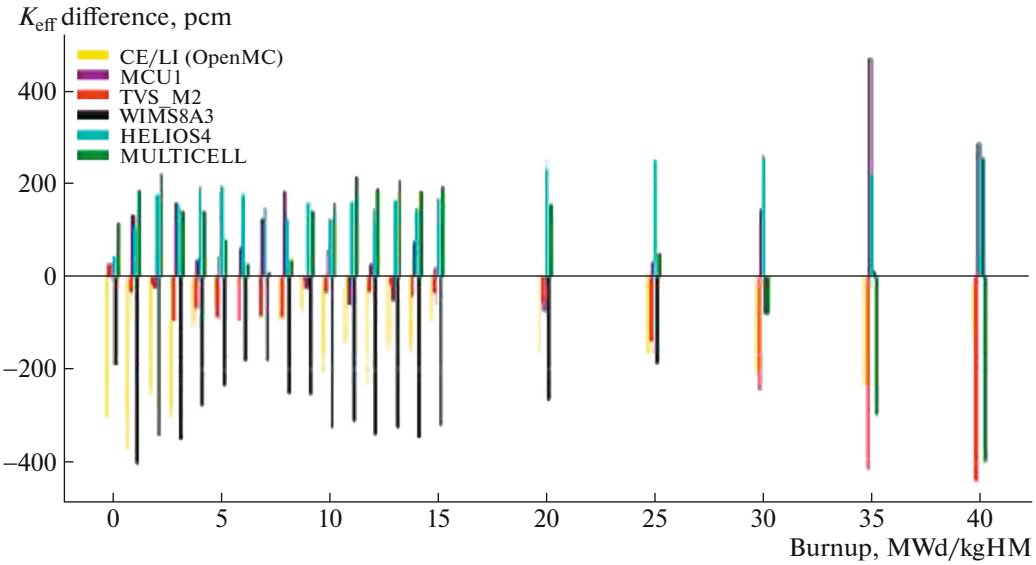

Fig. 9. Deviations of MCU, TVS-M, WIMS8A, HELIOS, MULTICELL and CE/LI OpenMC codes from the average K_{eff} trend for LEU fuel assembly.

burnup-dependent K_{eff} trend was calculated for the LEU assembly. Deviations of K_{eff} trends obtained from each of the five codes from the average K_{eff} trend were then plotted, as shown in Fig. 9. A similar deviation was also plotted for the CE/LI integrator used in OpenMC. The results obtained with OpenMC were not used in calculating the average K_{eff} trend.

The results obtained with the CE/LI integrator in OpenMC showed good agreement with the results obtained by other programs, since the maximum deviation did not exceed 400 pcm, and deviations up to 1000 pcm were considered acceptable.

CONCLUSIONS

The aim of this study was to select a numerical integration algorithm for modeling the burnup of fuel assemblies containing burnable absorbers in the OpenMC code. In OpenMC, there are eight different numerical integrators that can be used for burnup modeling: PI, CE/CM, LE/QI, CE/LI, CF4, EPC-RK4, SI-CE/LI, SI-LE/QI. The test results showed that the SI-CE/LI and SI-LE/QI integrators require significantly more time for calculating a single burnup step compared to the others, while maintaining the same accuracy. Therefore, they were excluded from

further consideration. The PI integrator showed low integration accuracy at similar burnup steps compared to other integrators. However, the PI integrator is computationally efficient, and as the burnup step decreases, it exhibits convergence to a common solution that can be chosen as a reference for assessing the quality of other integrators. On the basis of the results obtained using the PI integrator with a fine step, the decision was made to use the CE/LI integrator for further work. The results obtained with CE/LI were compared to the results obtained in the VVER-1000 LEU and MOX benchmark using the following codes: MCU, TVS-M, WIMS8A, HELIOS, MULTICELL; and they showed good agreement. Therefore, it can be concluded that the CE/LI integrator within OpenMC is applicable for modeling the burnup of fuel assemblies containing burnable absorbers.

Additionally, it is essential to note the importance of dividing burnable absorber regions into zones with independent material compositions to correctly model burnup.

ACKNOWLEDGMENTS

The research was conducted using the resources of the high-performance computing center at the National Research Nuclear University MEPhI.

FUNDING

This work was supported by ongoing institutional funding. No additional grants to carry out or direct this particular research were obtained.

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

REFERENCES

1. P. K. Romano, N. E. Horelik, B. R. Herman, A. G. Nelson, B. Forget, and K. Smith, Ann. Nucl. Energy **82**, 90 (2015).
2. J. Dufek, D. Kotlyar, and E. Shwageraus, Ann. Nucl. Energy **60**, 295 (2013).
3. A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna, Acta Numer. **9**, 215 (2000).
4. A. E. Isotalo and P. Aarnio, Ann. Nucl. Energy **38**, 1987 (2011).
5. C. Josey, Ph. D. Thesis (Massachusetts Inst. Technol., 2017).
6. R. Bahdanovich, A. Gerasimov, and G. Tikhomirov, J. Phys.: Conf. Ser. **1439** (2020).
7. L. Thilagam, C. Sunil Sunny, and V. Jagannathan, Ann. Nucl. Energy **36**, 505 (2009).
8. NEA/NSC/DOC, *A WWER-1000 LEU and MOX Assembly Computational Benchmark* (Nucl. Energy Agency, 2002).
9. J. Park, A. Khassenov, W. Kim, S. Choi, and D. Lee, Ann. Nucl. Energy **124**, 385 (2019).
10. R. E. MacFarlane and D. W. Muir, *The NJOY Nuclear Data Processing System* (1994).
11. J. Leppänen, *Serpent—a Continuous-Energy Monte Carlo Reactor Physics Burnup Calculation Code* (VTT Tech. Res. Centre, Finland).
12. J. Yu, Ann. Nucl. Energy **170**, 108973 (2022).
13. Y. N. Devyatko and V. V. Novikov, Fiz. At. Yadra **81**, 1257 (2018).

Translated by M. Chubarova