A Monte Carlo technique for sensitivity analysis of alpha-eigenvalue with the differential operator sampling method

Toshihiro Yamamoto[a*], Hiroki Sakamoto[b]

[a]*Institute for Integrated Radiation and Nuclear Science, Kyoto University, 2 Asashiro Nishi, Kumatori-cho, Sennan-gun, Osaka, 590-0494, Japan*
[b]*Independent author, Japan*

Abstract

A method for Monte Carlo sensitivity analyses of α-eigenvalue (prompt neutron time decay constant) in a subcritical system is developed using the first-order differential operator sampling (DOS) method. The first-order derivative of α-eigenvalue with respect to nuclear data is calculated using the DOS method that includes the capability of calculating perturbed source effect. This paper is an extension of the author's previous work for development of the sensitivity analysis method for k_{eff}-eigenvalue. Unlike the conventional Monte Carlo method for α-eigenvalue calculation that uses the power iteration of fission sources, this paper introduces a recently developed "time source method". The "time source method" has a weakness for a void-containing subcritical system, which is overcome by assigning a virtual total cross section in the void region. The perturbed source effect, which is caused by the change of nuclear data in a subcritical system, can be calculated by two methods, the source perturbation iteration method and the superhistory method. The source perturbation iteration method is superior in terms of computation efficiency, but a huge computer memory is required. The superhistory method dramatically reduces the memory requirement, although it worsens the variance of the sensitivity coefficients. The method developed in this paper is applied to some numerical tests that use multi-group constants, and it is verified by comparing to the results obtained

* Corresponding author. Tel:+81 72 451 2414; fax:+81 72 451 2658
E-mail address: toshihiro.yamamoto223@gmail.com (T. Yamamoto)

by a deterministic perturbation theory.

Keywords: alpha-eigenvalue; Monte Carlo; sensitivity coefficient; differential operator; superhistory

1. Introduction

Monte Carlo methods for sensitivity and uncertainty (S/U) analysis of k_{eff}-eigenvalue or neutron general responses (e.g., capture to fission ratio) have been developed and installed into production-level Monte Carlo calculation codes such as SCALE (Rearden, 2004; Perfetti, 2012; Perfetti and Rearden, 2016), MCNP (Kiedrowski et al., 2011; Kiedrowski and Brown, 2013), SERPENT (Aufiero et al., 2015), MORET (Jinaphanh et al., 2016), McCARD (Shim and Kim, 2011), and RMC (Qiu et al., 2015; Qiu et al., 2016a; Qiu et al., 2016b). There are several Monte Carlo techniques that are used for the S/U analyses such as "iterated fission probability" (IFP) method (Truchet, et al., 2015; Terranova and Zoia, 2017), "collision-history based method" (Aufiero et al., 2015; Aufiero et al., 2016), "contribution method" (Perfetti and Rearden, 2016), "superhistory" method (Yamamoto, 2018), and so on. This paper newly develops a Monte Carlo technique for calculating sensitivity coefficients of α-eigenvalue with respect to nuclear data, which has not previously been performed. In a subcritical system, k_{eff}-eigenvalue cannot be directly measured. Instead, an α-eigenvalue (prompt neutron time decay constant) can be directly measured using the pulsed neutron method or other reactor noise techniques. The α-eigenvalue is closely related with the subcriticality, and it is an important indicator for verification or validation of computational methods for subcritical system analyses. The sensitivity or uncertainty analyses of α-eigenvalue with respect to nuclear data will contribute to evaluation of nuclear data in a subcritical system. The sensitivity coefficients of α-eigenvalue can be calculated by the deterministic method using the forward and adjoint fluxes in an α-eigenvalue mode calculation. Recently, Endo and A. Yamamoto

(2018) and Favorite (2018) established a calculation method for sensitivity coefficient analysis of α-eigenvalue within the limit of the deterministic method. On the other hand, developing a Monte Carlo method to calculate the sensitivity coefficients of α-eigenvalue has not been attempted as far as the authors of this paper know. The objective of this paper is to develop a Monte Carlo technique of α-eigenvalue sensitivity analyses. For this purpose, an α-eigenvalue needs to be obtained in the Monte Carlo method. Conventionally, a Monte Carlo α-eigenvalue calculation method utilizes the fission source power iteration method for k_{eff}-eigenvalue calculations (Brockway et al., 1985; Yamamoto and Miyoshi, 2003; Yamamoto, 2011; Zoia et al., 2014). The α-eigenvalue is calculated during the course of k_{eff}-eigenvalue calculation. The α-eigenvalue is iteratively updated at the end of each cycle in such a way that the k_{eff}-eigenvalue becomes unity. This method is dubbed as the "fission source method" hereafter. It is well known that this Monte Carlo method for α-eigenvalue suffers from an abnormal termination due to too many neutrons produced in a deep subcritical system that has a large α-eigenvalue. To overcome this drawback, a "time source method" was invented by Shim et al. (2014, 2015). In the "time source method", the α-eigenvalue is truly an eigenvalue of the α-mode eigenvalue equation while the α-eigenvalue in the "fission source method" is an adjustment parameter to make the k_{eff}-eigenvalue unity.

If we choose the "time source" method, the α-eigenvalue can be expressed by a Neumann-series solution in an α-mode eigenvalue equation. Thus, the sensitivity coefficients of the α-eigenvalue with respect to nuclear data can be calculated with the differential operator sampling (DOS) method in the same way that the DOS method was used for the sensitivity coefficients of k_{eff}-eigenvalue (Yamamoto, 2018). This paper proposes a method for calculating sensitivity coefficients of α-eigenvalue by introducing the DOS method in the "time source" method.

It is commonly recognized that the Monte Carlo S/U analysis methods require a huge

amount of memory storage regardless of the method adopted. Several techniques for reducing the memory requirements have been developed. In MCNP, a sparse data handling scheme is employed. It can reduce the memory requirement by a factor of 10 to 100 for many problems (Kiedrowski and Brown, 2013). In McCARD (Shim and Kim, 2011; Choi and Shim, 2016a; Choi and Shim, 2016b), a memory-efficient adjoint estimation method was developed by applying the IFP concept for the Monte Carlo Wielandt method (Yamamoto and Miyoshi, 2004). In RMC (Wang, et al., 2015; Qiu et al., 2015; Qiu et al., 2016a; Qiu et al., 2016b), the superhistory method (Brissenden and Garlick,1986) as well as the Wielandt method was adopted. The DOS method also requires huge amount of memory to calculate perturbed source effect caused by the change of nuclear data. In Yamamoto (2018), the superhistory method was introduced to reduce the memory requirement for calculating the sensitivity coefficients of k_{eff}-eigenvalue. This paper again tries to apply the superhistory method for calculating the sensitivity coefficients of α-eigenvalue with less memory requirement.

In the sections that follow, a new Monte Carlo algorithm for calculating sensitivity coefficients of α-eigenvalue that uses the DOS method is presented. Some numerical tests are performed for verification of the new method. The superhistory method is applied to the numerical tests, and the performance of the superhistory method for sensitivity analyses of α-eigenvalue is examined.

2. Review of Monte Carlo α-eigenvalue calculation methods

2.1 Fission source method

This section reviews the Monte Carlo methods to calculate an α-eigenvalue. First, the conventional "fission source method" is presented. There exist several methods for the "fission source method". A method, published in (Yamamoto and Miyoshi, 2003; Yamamoto, 2011), is presented here. The equation to be solved for an α-eigenvalue

calculation is

$$\boldsymbol{L}\phi(\boldsymbol{r},\boldsymbol{\Omega},E)-\frac{\alpha}{v(E)}\phi(\boldsymbol{r},\boldsymbol{\Omega},E)\ =\frac{1}{k_p}\boldsymbol{F}\phi(\boldsymbol{r},\boldsymbol{\Omega},E), \tag{1}$$

where

$$\begin{aligned}\boldsymbol{L}\phi(\boldsymbol{r},\boldsymbol{\Omega},E)&=\boldsymbol{\Omega}\cdot\nabla\phi(\boldsymbol{r},\boldsymbol{\Omega},E)+\Sigma_t(\boldsymbol{r},E)\phi(\boldsymbol{r},\boldsymbol{\Omega},E)\\&-\int_{4\pi}d\boldsymbol{\Omega}'\int dE'\Sigma_s(\boldsymbol{r},\boldsymbol{\Omega}'\rightarrow\boldsymbol{\Omega},E'\rightarrow E)\phi(\boldsymbol{r},\boldsymbol{\Omega}',E'),\end{aligned} \tag{2}$$

$$\boldsymbol{F}\phi(\boldsymbol{r},\boldsymbol{\Omega},E)=\frac{\chi_p(E)}{4\pi}\int_{4\pi}d\boldsymbol{\Omega}'\int dE'\nu_p\Sigma_f(\boldsymbol{r},E')\phi(\boldsymbol{r},\boldsymbol{\Omega}',E')\ , \tag{3}$$

Σ_t =the macroscopic total cross section, Σ_s =the macroscopic scattering cross section, Σ_f =the macroscopic fission cross section, χ_p =the prompt neutron spectrum, ν_p =the number of prompt neutrons per fission, and $v=$ the neutron velocity. k_p in Eq. (1) is supposed to be unity in the α–mode eigenvalue equation. However, it is explicitly shown because k_p needs to be calculated in the "fission source method". Eq. (1) is very similar to the eigenvalue equation for k_{eff} except that the last term on the left-hand side, $\alpha/v(E)\cdot\phi(\boldsymbol{r},\boldsymbol{\Omega},E)$, is included and that χ and ν are for prompt neutrons. The Monte Carlo algorithm to solve Eq. (1) is almost the same as the one for k_{eff}-eigenvalue calculations. Thus, the "source" of this calculation method is the fission source, and the eigenvalue is the k_p-eigenvalue instead of the α. The difference from the normal k_{eff}-eigenvalue calculation is that we must take into account the second term on the left-hand side of Eq. (1) during the random walk processes of the Monte Carlo calculation. For this purpose, as the particle flies a distance s_j in the jth flight path, the initial weight W_j changes to

$$W_{j+1}=W_j\cdot exp\left(\frac{\alpha}{v_j(E)}s_j\right). \tag{4}$$

Because the α is a positive value in a subcritical system, the weight increases as it flies even through a void region. At the end of each cycle, k_p is calculated in the same manner as in the k_{eff}-eigenvalue calculations. The α used for the next cycle is determined so that k_p approaches unity as

$$\alpha_{m+1} = \alpha_m + c\left(1 - k_{p,m}\right), \tag{5}$$

where m is the cycle number and c is an arbitrary positive value. An appropriate value for the parameter, c, can be easily found by several trial runs. The α-eigenvalue is simply calculated by arithmetic mean of α_m beyond the inactive cycles. As seen in Eq. (5), the α value strongly correlates with the one in the previous cycle. Thus, we have to keep in mind that the standard deviation of the α values does not represent a true standard deviation.

2.2 Time source method

The "time source method" is another approach for α-eigenvalue calculation. The details are presented in Shim et al. (2014, 2015). In this paper, the Monte Carlo algorithm of the "time source method" is briefly explained.

The equation to be solved is

$$\boldsymbol{L}\phi(\boldsymbol{r},\boldsymbol{\Omega},E) - \boldsymbol{F}\phi(\boldsymbol{r},\boldsymbol{\Omega},E) = \alpha\frac{1}{v(E)}\phi(\boldsymbol{r},\boldsymbol{\Omega},E)\,, \tag{6}$$

which is almost the same as Eq. (1) except that k_p does not exist. In the "fission source method", fission neutrons produced in a cycle are not followed within the cycle. They are stored for the sources in the next cycle. On the other hand, in the "time source method", fission neutrons and their all progenies are followed within the cycle until they are all annihilated due to escape or Russian roulette, which means that the computation time of the "time source method" becomes longer than that of the "fission source method". However, the additional computation time for following the fission neutrons strongly depends on the subcriticality. As the subcriticality becomes larger, the increase in the computation time becomes insignificant. The sources for the next cycle m are determined at each collision point as:

$$n = \mathrm{Int}\left[\alpha_{m-1}\cdot\frac{w}{v(E)\Sigma_t(\boldsymbol{r},E)} + \xi\right], \tag{7}$$

where n = the number of sources at the collision point, w = the weight of the colliding particle, ξ = uniform pseudo random number from (0,1). Int [x] denotes the largest integer not exceeding x. α_{m-1} is the α-eigenvalue obtained in the previous cycle. This

term is introduced to keep the number of source particles almost constant throughout the cycles. The random walk process for the "time source method" is exactly the same as the fixed source problem in a subcritical system. In the "fission source method", the energy of a source particle is determined based on the fission neutron spectrum, and the direction of the source particle is isotropic. In the "time source method", the energy and the direction of a source particle are the same as those of the particle that generates the source particle. Thus, the energy and the direction of the colliding particle as well as the position of the collision need to be stored for the use of the next cycle. At the end of each cycle, the α-eigenvalue is calculated using the collision estimator as:

$$\alpha_{m,c} = \frac{S_m}{\sum_j \sum_i w_{ij} v\left(E_{ij}\right)^{-1} \Sigma_t\left(\boldsymbol{r}_{ij}, E_{ij}\right)^{-1}}, \tag{8}$$

where i and j denote the ith collision and jth source particles, respectively, and S_m =the sum of the source particle's weight in the mth cycle. The α-eigenvalue can also be calculated using the track length estimator as:

$$\alpha_{m,tr} = \frac{S_m}{\sum_j \sum_i w_{ij} s_{ij} v\left(E_{ij}\right)^{-1}}, \tag{9}$$

where i denotes the ith track, and $s_{ij} =$ the track length of the ith track from the jth source particle.

The "time source method" has a difficulty in that no source is generated in a void or transparent region. This difficulty does not arise in the "fission source method" because the effect of the term $\alpha/v(E) \cdot \phi(\boldsymbol{r}, \boldsymbol{\Omega}, E)$ is taken into account by Eq. (4) even in a void region. This difficulty can be circumvented by assigning a virtual total cross section in a void region. This method is similar to the Woodcock delta tracking method (Woodcock, 1965). A finite flight distance s in the void region is determined by $s = -\ell n \xi / \Sigma_{tv}$, where $\Sigma_{tv} =$ the virtual total cross section. If the virtual collision occurs within the void region, the number of source particles is determined at the collision point using Eq. (7). After that, the particle keeps flying without changing the weight or the direction. The virtual total

cross section can be determined arbitrarily. If a small virtual total cross section is chosen, the collision rarely occurs and a large number of sources per collision point are generated. If a large virtual total cross section is chosen, many unnecessary collisions occur. In the next section, the "time source method" is demonstrated for a subcritical system containing a void region to study how the virtual total cross section affects the calculation result.

2.3 Numerical tests of the time source method

In this section, a numerical example for α-eigenvalue calculation in a subcritical system including a void region is presented using three-energy group calculations. Fig. 1 shows the geometry for the calculation is a two-dimensional rectangular shape. The inner region is a void region and the outer region consist of a homogenized UO_2 fuel rod array. Table 1 shows the three-energy group constants of the homogenized UO_2 fuel rod array. The group constants are prepared with a standard thermal reactor analysis code SRAC (Okumura et al., 2007). The scatterings are assumed to be isotropic. The reference calculation for the α-eigenvalue is performed with a discrete ordinates transport code DANTSYS (Alcouffe et al., 1995) using the same group constants. A test Monte Carlo program developed by the authors of this paper is used throughout this study. The α-eigenvalues are obtained with the "time source method" and the "fission source method" with 30,000 neutrons per cycle, skipping 20 cycles and running 3,000 active cycles. The α-eigenvalue calculations of the "time source method" are performed with several virtual total cross sections. The results of the α-eigenvalues and the relative CPU time are shown in Table 2. The α-eigenvalues are calculated with the track length estimator (Eq. (9)). The number of source neutrons in the void region are also shown in Table 2. While the number of sources in the void region and the relative CPU time are almost constant regardless of the virtual total cross section, the fluctuation of the number of sources becomes larger for a small virtual total cross section. The α-eigenvalues calculated with the virtual total cross

section that is larger than 1.0 cm^{-1} agree well with the "fission source method" and the deterministic method. Compared with the "fission source method", the "time source method" requires much longer computation time because the fission neutrons have to be followed within a cycle. The virtual total cross section of 0.1 cm^{-1} slightly overestimates the α-eigenvalue, which may be caused by less source points generated in the void region and by improperly distributed source points. The null virtual total cross section generates null time sources in the void region, and the α-eigenvalue is significantly overestimated. Using the collision estimator for the null virtual total cross section, the α-eigenvalue is incorrectly overestimated because the contribution to the time sources are omitted in the void region. The dependence of the computation time on the virtual total cross section is not so large because the number of time sources in the void region is almost constant regardless of the virtual total cross section. However, an excessively large virtual total cross section ($\Sigma_{tv} =$ 20.0 or 50.0 cm^{-1}) worsens the computational efficiency because of many unnecessary virtual collisions.

[Fig. 1][Table 1][Table 2]

3. Sensitivity analysis of α-eigenvalue calculation

3.1 Differential operator sampling method

The differential operator sampling (DOS) method (Rief, 1984; McKinney and Iverson, 1996; Densmore et al., 1997) for the perturbation calculation with the source perturbation being implemented was already established in previous research (Nagaya and Mori, 2005; Nagaya and Mori, 2011; Raskach, 2009; Raskach, 2010; Jinaphanh et al., 2016). The capability of the DOS method was expanded to the second and higher orders (Nagaya and Mori, 2011; Nagaya et al., 2015). The formalism to calculate the first derivative of k_{eff}-eigenvalue with respect to a parameter was presented in detail in many previous publications (e.g., Nagaya and Mori, 2005; Yamamoto, 2018). A recently published paper (Yamamoto, 2018) is one of examples in which the application of the first-order DOS

method to the sensitivity coefficient of k_{eff}-eigenvalue was demonstrated. The sensitivity coefficient of α-eigenvalue with respect to nuclear data is equivalent to the first derivative of the α-eigenvalue with respect to nuclear data. Thus, the first-order DOS method is straightforwardly available for calculating sensitivity coefficients of α-eigenvalue. This section newly presents a Monte Carlo algorithm for applying the first-order DOS method to the sensitivity coefficient of α-eigenvalue.

The DOS method scores an estimate of each differential coefficient with respect to a parameter at each flight path or each collision point within a region where the parameter exists. The fundamentals are the same as the sensitivity coefficient of k_{eff}-eigenvalue. Thus, this section, for the most part, is duplicated from the previous publication (Yamamoto, 2018). The estimates that are scored during the course of the random walk process are shown as follows.

First, a particle starts from a time source position $\boldsymbol{r}$. The energy and the direction of the source particle are inherited from the particle that generates the source particle in the previous cycle. The particle moves from the position $\boldsymbol{r}$ to a collision point $\boldsymbol{r}'$ that is determined by the transport kernel:

$$T(\boldsymbol{r} \to \boldsymbol{r}') = \Sigma_t \exp(-\Sigma_t s). \tag{10}$$

When the particle travels a distance s through the perturbed region and undergoes a collision, the weighting coefficient to be scored is

$$\frac{1}{T}\frac{\partial}{\partial x}T(\boldsymbol{r} \to \boldsymbol{r}') = \frac{1}{\Sigma_t}\frac{\partial \Sigma_t}{\partial x} - s\frac{\partial \Sigma_t}{\partial x}, \tag{11}$$

where x is a parameter to be perturbed. For simplicity, the variables for the energy and the direction are omitted. If x is a macroscopic cross section of reaction x (=s, c, f), $\partial \Sigma_t / \partial x = 1$. Thus, Eq. (11) is

$$\frac{1}{T}\frac{\partial}{\partial x}T(\boldsymbol{r} \to \boldsymbol{r}') = \frac{1}{\Sigma_t} - s, \quad \text{for} \quad x = \Sigma_s, \Sigma_c, \text{and } \Sigma_f. \tag{12}$$

If the particle passes through the perturbed region without undergoing a collision, only the second term on the right-hand side of Eq. (11), $-s\,\partial \Sigma_t / \partial x$, is scored.

When the particle undergoes a collision, the particle's weight is reduced by the non-absorption probability:

$$w' = \frac{\Sigma_s}{\Sigma_t} w. \tag{13}$$

The weighting coefficient for the scattering kernel Σ_s/Σ_t is

$$\frac{\Sigma_t}{\Sigma_s}\frac{\partial}{\partial x}\frac{\Sigma_s}{\Sigma_t} = \frac{1}{\Sigma_s}\frac{\partial \Sigma_s}{\partial x} - \frac{1}{\Sigma_t}\frac{\partial \Sigma_t}{\partial x}. \tag{14}$$

The sum of the weighting coefficients of Eqs. (11) and (14) is accumulated until the *i*th collision:

$$W_{s,i} = -\frac{\delta_{x\Sigma_s}}{\Sigma_{t,i}} + \sum_{l=1}^{i} \frac{1}{\Sigma_{s,l}}\frac{\partial}{\partial x}\Sigma_{s,l} - \sum_k s_k \frac{\partial}{\partial x}\Sigma_{t,k}, \tag{15}$$

where $\delta_{x\Sigma_s}\{= 0 \text{ if } x \neq \Sigma_s, = 1 \text{ if } x = \Sigma_s\}$. The second term on the right-hand side of Eq. (14), $-1/\Sigma_t \cdot \partial\Sigma_t/\partial x$, cancels out the same term in Eq. (11) except at the last collision. The second term on the right-hand side of Eq. (15) means the sum of $1/\Sigma_{s,l} \cdot \partial\Sigma_{s,l}/\partial x$ until the *i*th collision where $\Sigma_{s,l}$ is the macroscopic scattering cross section for the *l*th scattering. The third term on the right-hand side of Eq. (15) means the sum of $s_k \cdot \partial\Sigma_{t,k}/\partial x$ in the *k*th flight distance of the perturbed region until the *i*th collision.

At each collision point, the number of fission neutrons *m* is calculated as:

$$m = \text{Int}\left[\frac{\nu\Sigma_f}{\Sigma_t} w + \xi\right]. \tag{16}$$

If a fission neutron is generated at this point, i.e., $m > 0$, the weighing coefficient for the fission kernel is determined as:

$$\frac{\Sigma_t}{\nu\Sigma_f} \cdot \frac{\partial}{\partial x}\frac{\nu\Sigma_f}{\Sigma_t} = \frac{1}{\nu\Sigma_f} \cdot \frac{\partial \nu\Sigma_f}{\partial x} - \frac{1}{\Sigma_t} \cdot \frac{\partial \Sigma_t}{\partial x}. \tag{17}$$

If a fission occurs at the *i*th collision, the following weighting coefficient is transferred to each fission neutron:

$$W_{f,i} = W_{s,i} + \sum_n \frac{1}{\nu\Sigma_{f,n}} \cdot \frac{\partial \nu\Sigma_{f,n}}{\partial x}, \tag{18}$$

where the summation for *n* is carried out over all fissions in the perturbed region until the *i*th collision. The second term on the right-hand side of Eq. (17), $-1/\Sigma_t \cdot \partial\Sigma_t/\partial x$,

cancels out the same term in Eq. (11).

The scorings of Eqs. (15) and (18) are repeated at each flight, collision and fission until the particle from the time source and all its progenies are discarded. As a result, the first derivative of the inverse of the α-eigenvalue with respect to the perturbation parameter x for the mth particle history is given by

$$\frac{\partial}{\partial x}\left(\frac{1}{\alpha}\right) = \sum_i \frac{1}{v_i \Sigma_{t,i}} w_i W_{f,i}, \tag{19}$$

where w_i =the particle weight of the ith collision. From Eq. (19), the first derivative of the α-eigenvalue is derived as:

$$\frac{\partial}{\partial x}\alpha_m^{NP} = -\alpha^2 \sum_i \frac{1}{v_i \Sigma_{t,i}} w_i W_{f,i}. \tag{20}$$

The superscript *NP* denotes that Eq. (20) does not include the perturbed source effect caused by the change of x. $W_{f,i}$ is represented for some perturbation parameters as follows:

$$W_{f,i} = \begin{cases} -\sum_k s_k, & x = \Sigma_c, \quad (21) \\ -\frac{1}{\Sigma_{t,i}} + \sum_{l=1}^{i} \frac{1}{\Sigma_{s,l}} - \sum_k s_k, & x = \Sigma_s, \quad (22) \\ \sum_{n=1}^{N} \frac{1}{\Sigma_{f,n}} - \sum_k s_k, & x = \Sigma_f, \quad (23) \\ N, & x = \chi_g, \quad (24) \end{cases}$$

where N is the number of fissions in the perturbed region until the ith collision. Eq. (24) represents the sensitivity coefficient with respect to the fission spectrum in the gth group. After all the particles starting from the time source positions for one cycle are exhausted, the sensitivity coefficient in the cycle is calculated:

$$\frac{\partial \alpha_j^{NP}}{\partial x} = \frac{1}{M} \sum_{m=1}^{M} \frac{\partial \alpha_{m,j}^{NP}}{\partial x}, \tag{25}$$

where M = the number of particle histories in one cycle and j = the cycle number.

3.2 The differential operator sampling method with perturbed source effect

The change of a cross section perturbs the time source distribution, thereby affecting the sensitivity coefficients. The first derivative given in Eq. (25) does not include the perturbed source effect. The DOS method for the perturbed source effect is already established for the sensitivity coefficient of k_{eff}-eigenvalue (Nagaya and Mori, 2005; Nagaya and Mori, 2011; Yamamoto, 2018). The perturbed source effect for the sensitivity coefficient of α-eigenvalue can be calculated with the same procedure as for k_{eff}-eigenvalue. Thus, this section, for the most part, is duplicated from the previous publication (Yamamoto, 2018). The perturbed source effect is calculated as follows.

At a point where the *l*th time source neutron in the *j*th cycle is generated at the *i*th collision in the *m*th particle history, the following quantity is scored:

$$w_{l,j,n}^{f} = W_{f,i,j}, \qquad \text{for } n = 1, \tag{26}$$

$$w_{l,j,n}^{f} = W_{f,i,j} + W_{PS,m,j,n-1}, \text{for } N \geq n \geq 2, \tag{27}$$

where $W_{f,i,j}$ is the same one as defined in Eq. (18) except that the cycle index *j* is added. $W_{PS,m,j,n}$, which is defined in Eq. (28), represents the perturbed source effect in the weighting coefficient and it is inherited from the previous $(j-1)$th cycle. The subscript *n* stands for the index of iteration for the source perturbation because the perturbed source effect needs to be calculated by an iteration procedure. *N* is the maximum iteration number, and the iteration is repeated until the perturbed source effect converges. In each cycle, $w_{l,j,n}^{f}$ is stored for $N \geq n \geq 1$ and $L \geq l \geq 1$ where *L* is the total number of time source neutrons generated in each cycle.

At the end of the *j*th cycle, $w_{l,j,n}^{f}$ calculated by Eq. (26) or (27) is normalized so that we obtain $W_{PS,l,j+1,n+1}$ for the next cycle:

$$W_{PS,l,j+1,n+1} = w_{l,j,n}^{f} - \frac{1}{L}\sum_{l=1}^{L} w_{l,j,n}^{f}, \qquad \text{for } L \geq l \geq 1 \text{ and } N-1 \geq n \geq 1. \tag{28}$$

This normalization process is to keep the size of the sampling constant in each cycle. The

perturbed source effect of the first derivative of the α-eigenvalue in the mth history in the jth cycle is scored at each collision point:

$$\frac{\partial}{\partial x}\alpha_{m,j}^{PS} = -\alpha^2 \sum_i \frac{1}{v_i \Sigma_{t,i}} w_i W_{PS,m,j,N}, \tag{29}$$

where the summation is carried out at each collision in the mth history. Eventually, the first derivative of the α-eigenvalue in the jth cycle is

$$\frac{\partial}{\partial x}\alpha_j = \frac{1}{M}\sum_{m=1}^{M}\left(\frac{\partial}{\partial x}\alpha_{m,j}^{NP} + \frac{\partial}{\partial x}\alpha_{m,j}^{PS}\right). \tag{30}$$

The memory requirement for this iteration procedure is $N \times L \times$ (bytes per variable for each cross section). The number of cross sections is (the number of isotopes) × (the number of reactions) × (the number of energy groups). If the sensitivity coefficients of many isotopes, reactions, and fine energy groups are sought at the same time, the iteration procedure in the DOS method (or in the IFP method) results in prohibitively huge memory requirements. This is because $w_{l,j,n}^{f}$ (defined in Eq. (26) or (27)) needs to be stored for $N \geq n \geq 1$ and $L \geq l \geq 1$ until $W_{PS,l,j,N}$ (defined in Eq. (28)) is obtained.

3.3 Memory reduction with the superhistory method

The superhistory method was applied for memory reduction in the sensitivity calculation of k_{eff}-eigenvalue in (Yamamoto, 2018). This paper applies the superhistory method to α-eigenvalue calculations and the sensitivity analyses of α-eigenvalue. In the superhistory method of this paper, the time source neutrons (instead of fission neutrons) in a cycle are tracked over N (>1) generations, each of which is called a "supergeneration". The α-eigenvalue calculated in the Nth supergeneration (i.e., the last supergeneration of the cycle) is adopted as the α-eigenvalue of the cycle, and the time source neutrons generated in the Nth supergeneration are inherited to the next cycle.

When the lth time source neutron is generated at the ith collision at the first supergeneration of the jth cycle, the weighting coefficient, $W_{s,i,1}$, (defined in Eq. (15) for

the *i*th collision in the history) is assigned to the time source neutron as:

$$W^s_{l,j,1} = W_{s,i,1}, \tag{31}$$

where $W^s_{l,j,1}$= the weighting coefficient of the *l*th time source neutron in the first supergeneration of the *j*th cycle. The number of time source neutrons in each supergeneration is determined using Eq. (7). The total number of time source neutrons in each supergeneration is nearly constant due to the multiplication of α in Eq. (7) that is obtained in the previous cycle.

In the second supergeneration, $W^s_{l,j,1}$, which is assigned to the *l*th time source neutron, is further transferred to an l'th time source neutron that is generated by the *l*th source neutron as:

$$W^s_{l',j,2} = W^s_{l,j,1}. \tag{32}$$

This procedure is repeated until the last supergeneration of the *j*th cycle. At the end of the last supergeneration, the sensitivity coefficient of α with respect to a parameter *x* for the *m*th superhistory in the *j*th cycle is given by

$$\frac{\partial}{\partial x}\alpha_{m,j} = -\alpha^2 \sum_i \frac{1}{v_i \Sigma_{t,i}} w_i W^s_{m,j,N}, \tag{33}$$

where the summation is carried out at each collision in the *N*th supergeneration and $W^s_{m,j,N}$ is the weighting coefficient for the *m*th superhistory in the *N*th supergeneration. The final result of the first derivative of α includes the perturbed source effect if the number of the supergeneration *N* is large enough. The weighting coefficient in the last supergeneration $W^s_{m,j,N}$ can be obtained one by one in each superhistory. Thus, it is not necessary to store a large amount of information until all histories in each cycle are terminated. The memory requirement for the superhistory method can be reduced by a factor of (the number of histories per cycle) × (the number of iterations for source perturbation) compared to the method in the previous section.

4. Numerical tests for sensitivity coefficient of α-eigenvalue

4.1 Differential operator sampling method with perturbed source effect

The sensitivity calculation methods for α-eigenvalue using the DOS method are tested in this section. The geometry of the test problem is shown in Fig. 2. The geometry is a two-dimensional rectangular shape. The inner and outer regions consist of a homogenized light-water moderated mixed oxide fuel rod array and a homogenized UO_2 fuel rod array, respectively. Table 1 shows the three-energy group constants for these regions. The sensitivity coefficients are calculated with respect to the macroscopic cross sections (capture, fission, and scattering cross sections) or the fission spectrum in the *inner* region (i.e., the MOX fuel rod array).

[Fig. 2]

The reference calculations for the sensitivity coefficients are performed with DANTSYS using the same group constants. The three-energy group forward and adjoint angular fluxes of α-eigenvalue mode are calculated with the S_n order 8. Using the angular fluxes, the sensitivity coefficients are calculated based on the linear perturbation theory:

$$S_x = \frac{x}{\alpha}\frac{d\alpha}{dx} = \frac{-\langle \boldsymbol{\Phi}^* x \frac{d\boldsymbol{F}}{dx}\boldsymbol{\Phi}\rangle - \langle \boldsymbol{\Phi}^* x \frac{d\boldsymbol{S}}{dx}\boldsymbol{\Phi}\rangle + \langle \boldsymbol{\Phi}^* x \frac{d\Sigma_t}{dx}\boldsymbol{\Phi}\rangle}{\alpha\langle \boldsymbol{\Phi}^* \boldsymbol{V^{-1}} \boldsymbol{\Phi}\rangle}, \tag{34}$$

where x = a cross section or a fission spectrum, $\boldsymbol{\Phi}$ =the forward flux, $\boldsymbol{\Phi}^*$ =the adjoint flux, $\langle\ \rangle =$ the integration over all phase space, $\boldsymbol{F}$ =the production operator, $\boldsymbol{S}$ =the scattering operator, and $\boldsymbol{V^{-1}} =$ the inverse of neutron velocity. The formula in Eq. (34) is almost the same as the one in (Endo and A. Yamamoto, 2018). The difference is that Eq. (34) neglects the dependence of the neutron velocity on the parameter x unlike the formula in (Endo and A. Yamamoto, 2018).

For verification of the test Monte Carlo program developed for the purpose of this paper, k_{eff} and α-eigenvalues are calculated both with the Monte Carlo program and DANTSYS. The results are compared in Table 3. The eigenvalues by the Monte Carlo

method agree well with the results by DANTSYS. As stated in Sec. 2.2, the computational time for an α-eigenvalue calculation with the "time source method" becomes longer with the subcriticality. This is demonstrated in Table 4, where the relative computation times are compared for several levels of subcriticality.

[Table 3][Table 4]

The sensitivity coefficients of α-eigenvalue with respect to the capture, fission, scattering cross sections and the fission spectrum are shown in Tables 5, 6, 7, and 8, respectively. The sensitivity coefficient with respect to the scattering cross section in the third group is insignificant and it is omitted in Table 7. The sensitivity calculations are performed with 30,000 neutrons per cycle, skipping 20 cycles and running 2,000 active cycles. The number of iterations for the perturbed source effect is 12. This number of iterations is large enough to obtain the converged solutions as shown later. The results with DANTSYS agree with those with the Monte Carlo method within three standard deviations except for the fission spectrum. The geometry of this test problem is two-thirds smaller than the test problem in (Yamamoto, 2018) where the sensitivity coefficients of k_{eff}-eigenvalue were calculated. The remaining conditions are unchanged. Comparing the sensitivity coefficients of α-eigenvalue in this paper with those of k_{eff}-eigenvalue in (Yamamoto, 2018), the perturbed source effect of α-eigenvalue is minor and it is approximately one-tenth as small as that of k_{eff}-eigenvalue. The perturbed source effect is the most significant for the scattering cross section in the second group. Fig. 3 shows the perturbed source effect for the scattering cross section in the second group as a function of the number of iterations for source perturbation. The perturbed source effect converges approximately after 3 iterations. The convergence of the perturbed source effect of α-eigenvalue is faster than that of k_{eff}-eigenvalue that requires more than 7 iterations as seen in Fig. 2 in (Yamamoto, 2018). The perturbed source effect depends on the level of subcriticality. It can be anticipated that the perturbed source effect becomes more

significant and its convergence becomes slower as the subcriticality becomes larger. This is corroborated in Table 9, where the fractions of the perturbed source effect in the total sensitivity coefficients are listed for several levels of subcriticality.

[Table 5][Table 6][Table 7][Table 8][Table 9][Fig. 3]

The geometry of another numerical test is shown in Fig. 4. The inner and outer regions consist of a light water and a homogenized UO_2 fuel rod array, respectively. In the same manner as in the previous test, the sensitivity coefficients of α-eigenvalue are calculated with respect to the macroscopic cross sections (capture and scattering cross sections) in the *inner* region (i.e., the light water). The sensitivity coefficients of α-eigenvalue with respect to the capture and scattering cross sections are shown in Tables 10 and 11, respectively. The results with DANTSYS agree with those with the Monte Carlo method within two standard deviations. The statistical uncertainty is relatively large for the sensitivity coefficient to the scattering cross section. Looking back to Eq. (22), we understand that this large uncertainty is caused by the cancelation of the second term $(1/\Sigma_{s,l})$ and the third term (s_k) in Eq. (22).

[Table 10][Table 11] [Fig. 4]

4.2 Superhistory method

The sensitivity coefficients for the test problems in Sec. 4.1 are calculated using the superhistory method. First, it is investigated how many supergenerations are needed to achieve convergence of the sensitivity coefficients. Fig. 5 shows the sensitivity coefficients with respect to the scattering cross section in the second group as a function of the number of supergenerations. The convergence of the sensitivity coefficients seems to be achieved after 8 supergenerations. Thus, all results of the sensitivity coefficients in this paper are obtained with 8 supergenerations. The results of the superhistory method for the test problems in Sec. 4.1 are shown in Tables 12 through 17. In the tables, the relative figure-of-merit (=1/(variance)/(CPU time)) of the superhistory method with respect to the source perturbation iteration method is shown. The superhistory method well reproduces

the results of the source perturbation method except for the scattering cross section of the light water. The difference in the case of the scattering cross section is due to the large statistical uncertainty. The FOM of the superhistory method is reduced to approximately less than 20% of the source perturbation method. However, the memory reduction is an obvious advantage of the superhistory method, albeit with less computational efficiency.

[Table 12][Table 13][Table 14][Table 15][Table 16][Table 17][Fig. 5]

5. Conclusions

Following the procedure for calculating the sensitivity coefficients of k_{eff}-eigenvalue that has been already presented in the author's previous publication, a new Monte Carlo calculation technique for the sensitivity coefficients of α-eigenvalue is proposed in this paper. The conventional Monte Carlo α-eigenvalue calculation method uses the power iteration method for k_{eff}-eigenvalue. This conventional α-eigenvalue calculation that uses the power iteration of fission sources is not available for the sensitivity analyses of α-eigenvalue. Instead, the "time source method" is introduced for the sensitivity analyses of α-eigenvalue. One weakness of the "time source method" is that no time source can be defined in a void region. This weakness can be circumvented by assigning a virtual total cross section in the void region, which is similar to the Woodcock delta tracking method. Although the virtual total cross section can be arbitrarily defined, a smaller virtual total cross section leads to a large statistical uncertainty in time sources in the void region and a biased α-eigenvalue. The virtual total cross section should be large enough to the extent that it does not inadvertently increase the computation time due to unnecessary virtual collisions.

Using the "time source method", the sensitivity coefficients of α-eigenvalue can be calculated in the same manner as the sensitivity coefficients of k_{eff}-eigenvalue. However, the "time source method" requires longer computation time as compared to the "fission source method" because fission neutrons and all their progenies have to be followed within

a cycle. The sensitivity coefficient of α-eigenvalue is the first derivative of α-eigenvalue with respect to nuclear data. Thus, the first-order differential operator sampling method is available for this purpose. The perturbed source effect needs to be considered for the sensitivity analyses of α-eigenvalue, and it can be calculated using the source perturbation iteration method or the superhistory method. It is found that the perturbed source effect in the sensitivity coefficient of α-eigenvalue is not so significant as that of k_{eff}-eigenvalue. The source perturbation iteration method requires a huge memory if the sensitivity coefficients of a large number of isotopes, reactions and energy groups are calculated in one calculation. On the other hand, the superhistory method does not need to store a large amount of information because the superhistory method tracks a single particle history over approximately ten supergenerations that constitute one cycle. Significant memory reduction can be achieved by introducing the superhistory method.

The numerical tests in this paper deal with the sensitivity coefficients with respect to the multi-group macroscopic cross sections. However, the algorithm presented in this paper can be expanded straightforwardly to sensitivity analyses of α-eigenvalue in the continuous energy Monte Carlo.

References

Alcouffe, R.E., Baker, R.S., Brinkley, F.W., Marr, D.R., O'Dell, R.D., Walters, W.F., 1995. DANTSYS: A diffusion accelerated neutral particle transport code system, LA-12969-M.

Aufiero, M., Bidaud, A., Hursin, M., Leppänen, J., Palmiotti, G., Pelloni, S., Rubiolo, P., 2015. A collision history-based approach to sensitivity/perturbation calculations in the continuous energy Monte Carlo code SERPENT, Ann. Nucl. Energy 84, 245–258.

Aufiero, M., Martin, M. Fratoni, M., 2016. XGPT: Extending Monte Carlo Generalized Perturbation Theory capabilities to continuous-energy sensitivity functions, Ann. Nucl. Energy 96, 295–306.

Brissenden, R.J., Garlick, A.R., 1986. Biases in the estimation of k_{eff} and its error by Monte Carlo methods. Ann. Nucl. Energy 113, 63–83.

Brockway, D., Soran, P., Whalen, P., 1985. Monte Carlo α calculation, Tech. Rep. LA-UR-85-1224, Los Alamos National Laboratory.

Choi, S. H., Shim, H. J., 2016a. Memory-efficient calculations of adjoint-weighted tallies by the Monte Carlo Wielandt method, Ann. Nucl. Energy 96, 287–294.

Choi, S. H., Shim, H. J., 2016b. Generalized sensitivity calculation in the Monte Carlo Wielandt method, Trans. Am. Nucl. Soc. 115, 575–577.

Densmore, J. D., McKinney, G.W., Hendricks, J.S., 1997. Correction to the MCNP perturbation feature for cross-section dependent tallies, LA-13374.

Endo, T., Yamamoto, A., 2018. Sensitivity analysis of prompt neutron decay constant using perturbation theory. J. Nucl. Sci. Technol. (in press).

Favorite, J. A., 2018. SENSMG: first-order sensitivities of neutron reaction rates, reaction-rate ratios, leakage, k_{eff}, and α using PARTISN. Nucl. Eng. Sci., 192, 80–114.

Jinaphanh, A., Leclaire, N., Cochet, B., 2016. Continuous-energy sensitivity coefficients in the MORET code, Nucl. Sci. Eng. 184, 53–68.

Kiedrowski, B.C., Brown, F.B., Wilson, P.P.H., 2011. Adjoint-weighted tallies for *k*-eigenvalue calculations with continuous-energy Monte Carlo. Nucl. Sci. Eng. 168(3), 226–241.

Kiedrowski, B.C., Brown, F.B., 2013. Adjoint-based *k*-eigenvalue sensitivity coefficients to nuclear data using continuous-energy Monte Carlo. Nucl. Sci. Eng. 174, 227–244.

McKinney, G. W., Iverson, J. L., 1996. Verification of the Monte Carlo differential operator technique for MCNP, Los Alamos National Laboratory, LA-13098.

Nagaya, Y., Mori, T., 2005. Impact of perturbed fission source on the effective multiplication factor in Monte Carlo perturbation calculations, J. Nucl. Sci. Technol., 42, 428–441.

Nagaya, Y., Mori, T., 2011. Estimation of sample reactivity worth with differential operator sampling method, Prog. Nucl. Sci. Technol., 2, 842–850.

Nagaya, Y., Okumura, K., Mori, T., 2015. Recent developments of JAEA's Monte Carlo code MVP for reactor physics applications, Ann., Nucl. Energy 82, 85–89.

Okumura, K., Kugo, T., Kaneno K, Tsuchihashi, K., 2007. SRAC2006: A comprehensive neutronics calculation code system. JAEA-Data/Code 2007-004.

Perfetti, C. M., 2012. Advanced Monte Carlo methods for eigenvalue sensitivity coefficient calculations, Doctoral dissertation, University of Michigan.

Perfetti, C. M., Rearden, B. T., 2016. Development of a generalized perturbation theory method for sensitivity analysis using continuous-energy Monte Carlo methods, Nucl. Sci. Eng. 182, 354–368.

Qiu, Y., Liang, J., Wang, K., Yu, J., 2015. New strategies of sensitivity analysis capabilities in continuous-energy Monte Carlo code RMC. Ann. Nucl. Energy 81, 50–61.

Qiu, Y., Aufiero, M., Wang, K., Fratoni, M., 2016a. Development of sensitivity analysis capabilities of generalized responses to nuclear data in Monte Carlo code RMC, Ann.

Nucl. Energy 97, 142–152.

Qiu, Y., Shang, X., Tang, X., Liang, J., Wang, K., 2016b. Computing eigenvalue sensitivity coefficients to nuclear data by adjoint superhistory method and adjoint Wielandt method implemented in RMC code, Ann. Nucl. Energy 87, 228–241.

Raskach, K. F., 2009. An improvement of the Monte Carlo generalized differential operator method by taking into account first- and second-order perturbations of fission source. Nucl. Sci. Eng. 162, 158–166.

Raskach, K. F., 2010. Extension of differential operator method to inhomogeneous problems with internal and external neutron sources. Nucl. Sci. Eng. 165, 320–330.

Rearden, B.T., 2004. Perturbation theory eigenvalue sensitivity analysis with Monte Carlo techniques. Nucl. Sci. Eng. 146, 367–382.

Rief, H., 1984. Generalized Monte Carlo perturbation algorithms for correlated sampling and a second-order Taylor series approach. Ann. Nucl. Energy 9, 455–476.

Shim, H. J., Kim, C. H., 2011. Adjoint sensitivity and uncertainty analyses in Monte Carlo forward calculations. J. Nucl. Sci. Technol. 48(12), 1453–1461.

Shim, H. J., Jeong, B. K., Kang, S. M., Kim, C. H., 2014. Monte Carlo alpha iteration algorithm for a prompt neutron decay constant calculation. Trans. Am. Nucl. Soc. 111, 725–726.

Shim, H. J., Jang, S. H., Kang, S. M., 2015. Monte Carlo alpha iteration algorithm for a subcritical system analysis. Sci. Technol. Nucl. Installations 2015, Article ID 859242.

Terranova, N., Zoia, A., 2017. Generalized iterated fission probability for Monte Carlo eigenvalue calculations, Ann. Nucl. Energy 108, 57–66.

Truchet, G., Leconte, P., Santamarina, A., Damian, F., Zoia, A., 2015. Computing adjoint-weighted kinetics parameters in TRIPOLI-4® by the Iterated Fission Probability method, Ann. Nucl. Energy 81, 17–26.

Wang, K., Li, Z., She, D., et al., 2015. RMC – a Monte Carlo code for reactor core analysis.

Ann. Nucl. Energy 82, 121–129.

Woodcock, E. R., 1965. Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry, Argonne National Laboratory, ANL-7050.

Yamamoto, T., Miyoshi, Y., 2003. An algorithm of α- and γ-mode eigenvalue calculations by Monte Carlo method. Proc. 7th Int. Conf. on Nuclear Criticality Safety (ICNC '03), JAERI-Conf 2003-019, Japan Atomic Energy Research Institute, Tokai, Japan, October 2003.

Yamamoto, T., Miyoshi, Y., 2004. Reliable method for fission source convergence of Monte Carlo criticality calculation with Wielandt's method. J. Nucl. Sci. Technol. 41(2), 99–107.

Yamamoto, T., 2011. Higher order α mode eigenvalue calculation by Monte Carlo power iteration. Prog. Nucl. Sci. Technol. 2, 826–835.

Yamamoto, T., 2018. Eigenvalue sensitivity analysis capabilities with the differential operator method in the superhistory Monte Carlo method. Ann. Nucl. Energy 112, 150–157.

Zoia, A., Brun, E., Malvagi, F., 2014. Alpha eigenvalue calculations with TRIPOLI-4. Ann. Nucl. Energy 63, 276–284.

List of figures

Figure

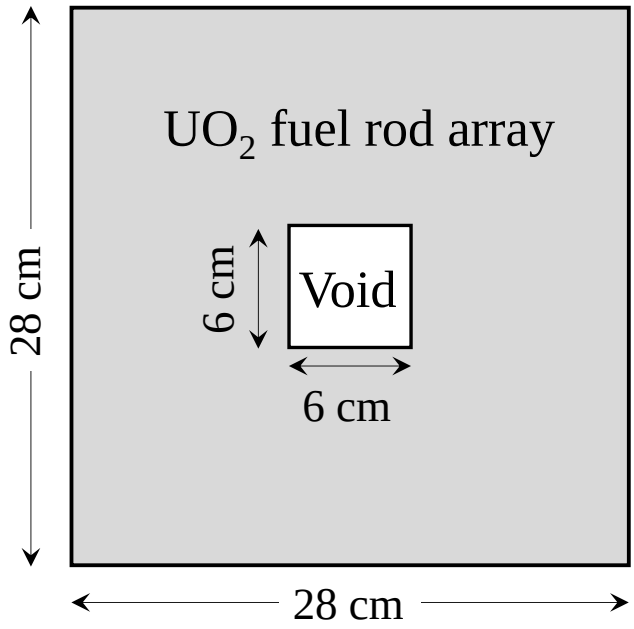

Fig. 1 Geometry of the test problem for α-eigenvalue calculation (void problem).

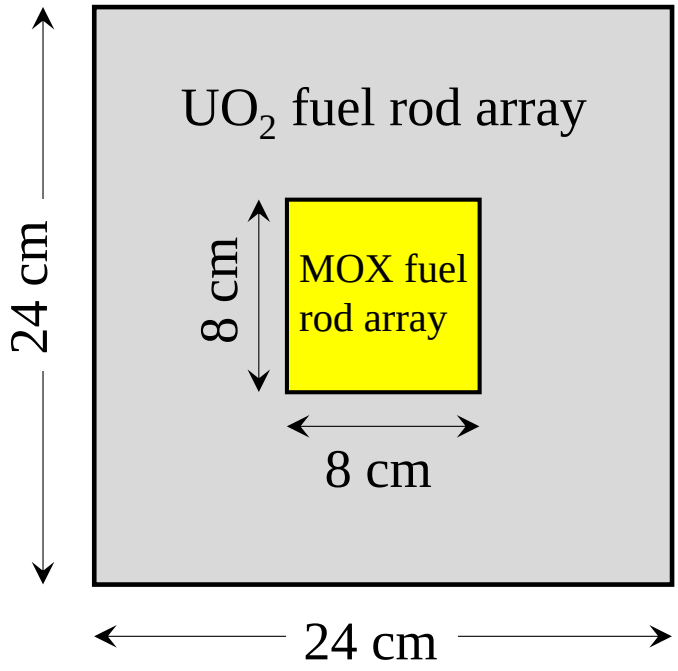

Fig. 2 Geometry of the test problem for sensitivity analysis of α-eigenvalue (MOX problem).

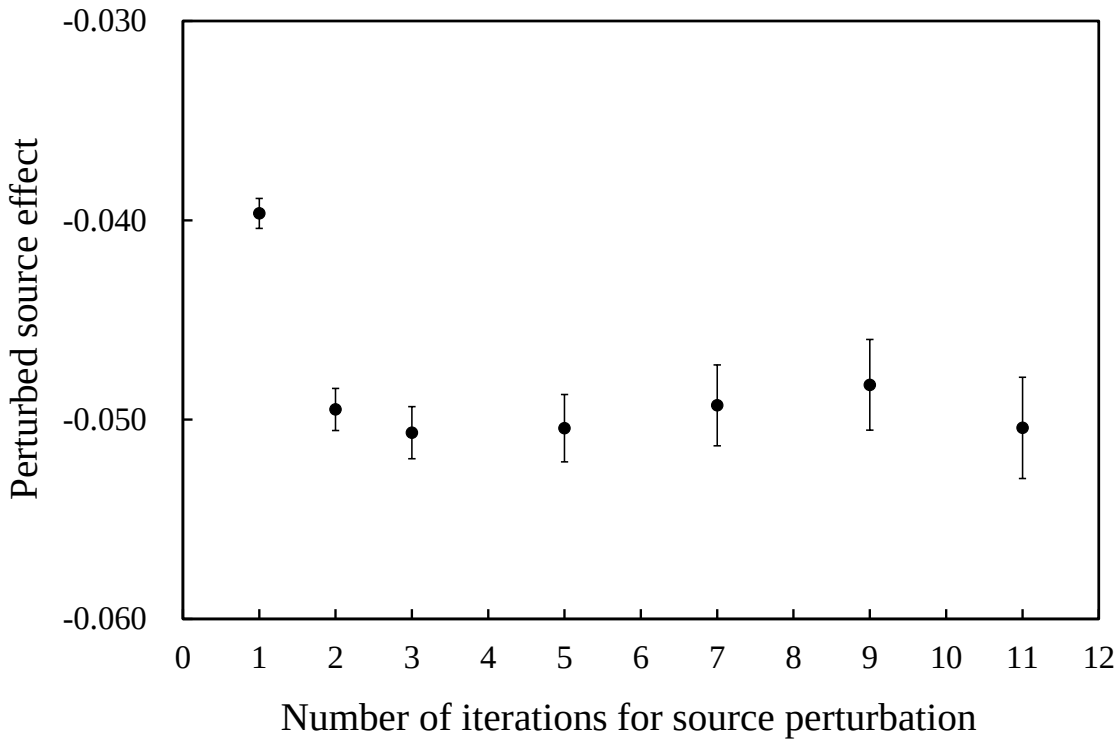

Fig. 3 Perturbed source effect vs. the number of iterations for source perturbation (the scattering cross section in the second group).

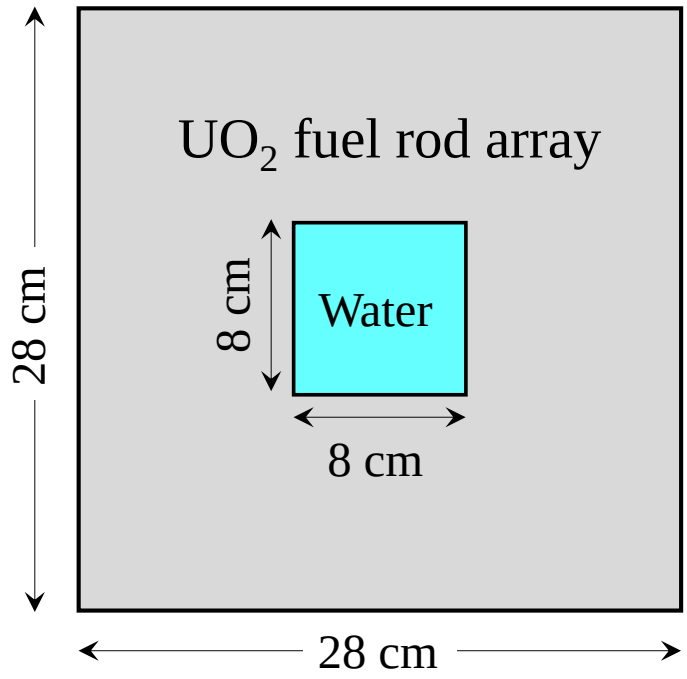

Fig. 4 Geometry of the test problem for sensitivity analysis of α-eigenvalue (water-hole problem).

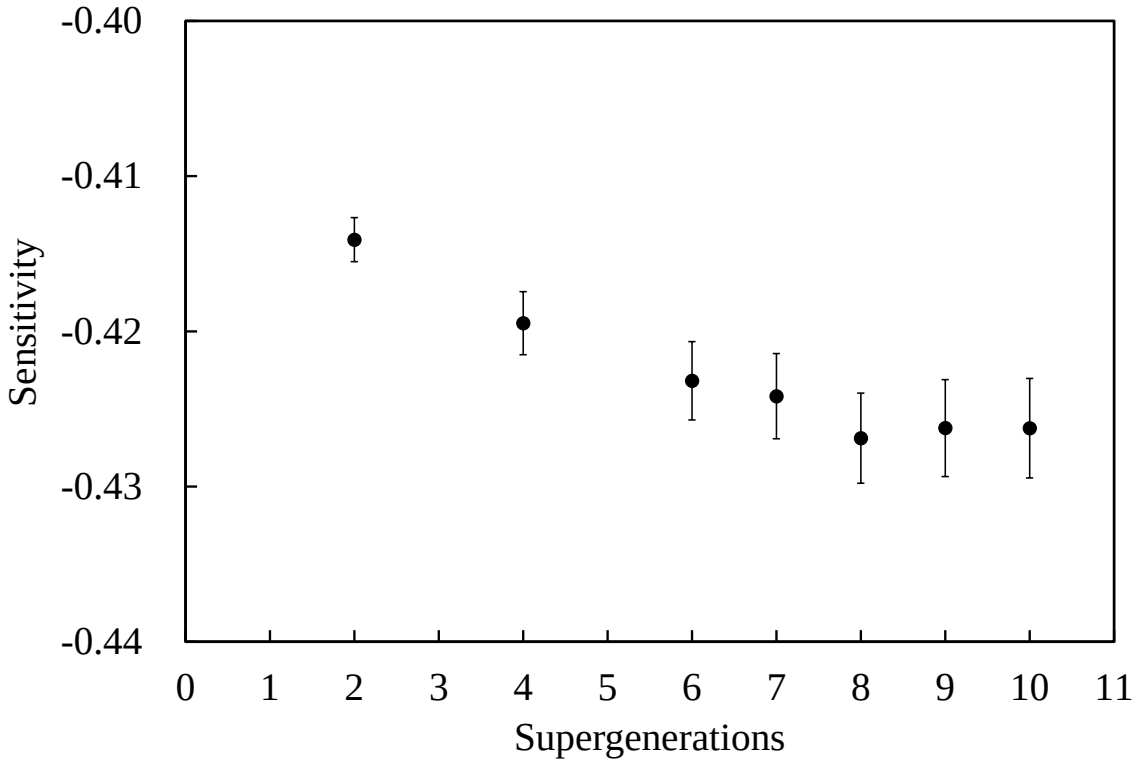

Fig. 5 Sensitivity coefficient vs. the number of supergenerations (the scattering cross section in the second group for “MOX problem”).

Table

Table 1 Three-group constants for UO_2 fuel rod array, MOX fuel rod array, and light water

		UO_2 fuel rod array	MOX fuel rod array	Light water
Total cross section (cm^{-1})	Σ_{t1}	0.29829	0.289397	0.33207
	Σ_{t2}	0.83334	0.825987	1.1265
	Σ_{t3}	1.6389	1.6600	2.7812
Fission cross section (cm^{-1})	Σ_{f1}	0.0030586	0.0025989	–
	Σ_{f2}	0.0021579	0.0019544	–
	Σ_{f3}	0.056928	0.070119	–
Absorption cross section (cm^{-1})	Σ_{a1}	0.003385	0.003265	0.00030500
	Σ_{a2}	0.011895	0.011435	0.00036990
	Σ_{a3}	0.086180	0.12441	0.0182500
Group transfer cross section (cm^{-1})	$\Sigma_s^{1\to2}$	0.073843	0.071620	0.10464
	$\Sigma_s^{1\to3}$	0.0	0.0	0.0
	$\Sigma_s^{2\to3}$	0.043803	0.044045	0.097961
Neutrons per fission	ν	2.4	2.8	–
Fission spectrum	χ_1	0.878198	0.878198	–
	χ_2	0.121802	0.121802	–
	χ_3	0	0	–
Neutron velocity (cm/s)	v_1		1.66743×10^9	
	v_2		1.73734×10^7	
	v_3		3.46850×10^5	

Table 2 Results of α-eigenvalue calculation for the void-containing system.

Virtual total cross section (cm^{-1})	α (s^{-1})*	Number of sources in the void region***	Number of source points in the void region****	Relative CPU time
0.0	3780.8 ± 1.1 (4131.8 ± 2.4)**	0	0	0.93
0.1	3585.6 ± 2.4	2879 ± 20	107.1 ± 0.2	1.00
1.0	3577.6 ± 1.3	2922 ± 7	255.1 ± 0.2	1.04
8.0	3578.7 ± 1.1	2918 ± 3	1338 ± 1	1.03
20.0	3579.1 ± 1.0	2918 ± 2	2300 ± 2	1.13
50.0	3578.0 ± 1.0	2919 ± 2	2918 ± 2	1.31
Fission source	3576.8 ± 0.4	–	–	0.0761
DANTSYS	3577.2	–	–	–

*The track length estimator is used.

**The collision estimator is used.

***The sum of "n" calculated by Eq. (7).

****The sum of the number of points where the time sources are produced.

Table 3 Results of α-eigenvalue and k_{eff} for the test problem.

	α (s^{-1})	k_{eff}
Monte Carlo	6355.3 ± 0.7*	0.84066 ± 0.00009
DANTSYS	6343.7	0.84078

*This α-eigenvalue is obtained by the "fission source method".

Table 4 Comparison of computation times for the "time source method" with different ν values.

	α (s^{-1})	k_{eff}	Relative CPU time
$\nu \times 1.00$	6356.8 ± 1.5	0.84066 ± 0.00009	1.00
$\nu \times 0.98$	7008.7 ± 1.5	0.82368 ± 0.00009	0.91
$\nu \times 0.95$	7978.8 ± 1.5	0.79851 ± 0.00009	0.79

Table 5 Sensitivity coefficients of α to the capture cross section in the MOX fuel rod array (MOX problem).

	1st Gr.	2nd Gr.	3rd Gr.
DANTSYS	8.891×10^{-3}	2.182×10^{-1}	7.653×10^{-1}
Differential operator (MC)			
Without source perturbation	8.658×10^{-3}	2.085×10^{-1}	7.161×10^{-1}
	$(4.1\times10^{-6})^{*}$	(9×10^{-5})	(2.6×10^{-4})
Perturbed source effect	1.935×10^{-4}	8.886×10^{-3}	4.636×10^{-2}
	(1.21×10^{-5})	(3.07×10^{-4})	(1.01×10^{-3})
Total	8.852×10^{-3}	2.174×10^{-1}	7.624×10^{-1}
	(1.3×10^{-5})	(3.2×10^{-4})	(1.0×10^{-3})
MC/DANTSYS	0.996	1.000	1.000

*One standard deviation

Table 6 Sensitivity coefficients of α to the fission cross section in the MOX fuel rod array (MOX problem).

	1st Gr.	2nd Gr.	3rd Gr.
DANTSYS	-6.683×10^{-2}	-5.398×10^{-2}	-5.650×10^{-1}
Differential operator (MC)			
Without source perturbation	-6.485×10^{-2}	-5.287×10^{-2}	-5.850×10^{-1}
	$(1.1\times10^{-4})^{*}$	(1.1×10^{-4})	(4.2×10^{-4})
Perturbed source effect	-2.060×10^{-3}	-3.279×10^{-4}	2.204×10^{-2}
	(3.37×10^{-4})	(3.212×10^{-4})	(1.25×10^{-3})
Total	-6.691×10^{-2}	-5.320×10^{-2}	-5.630×10^{-1}
	(3.5×10^{-4})	(3.4×10^{-4})	(1.3×10^{-3})
MC/DANTSYS	1.001	0.986	0.996

*One standard deviation

Table 7 Sensitivity coefficients of α to the scattering cross section in the MOX fuel rod array (MOX problem).

	1st Gr.	2nd Gr.
DANTSYS	-3.327×10^{-1}	-4.222×10^{-1}
Differential operator (MC)		
Without source perturbation	-3.015×10^{-1}	-3.727×10^{-1}
	$(6.5\times10^{-4})^{*}$	(1.4×10^{-3})
Perturbed source effect	-2.946×10^{-2}	-5.041×10^{-2}
	(1.96×10^{-3})	(4.34×10^{-3})
Total	-3.309×10^{-1}	-4.231×10^{-1}
	(2.1×10^{-3})	(4.6×10^{-3})
MC/DANTSYS	0.995	1.002

*One standard deviation

Table 8 Sensitivity coefficients of α to the fission spectrum in the MOX fuel rod array (MOX problem).

	1st Gr.	2nd Gr.
DANTSYS	1.482×10^{0}	2.721×10^{-1}
Differential operator (MC)		
Without source perturbation	1.443×10^{0}	2.684×10^{-1}
	$(7\times10^{-4})^{*}$	(2.1×10^{-4})
Perturbed source effect	2.882×10^{-2}	1.083×10^{-2}
	(2.00×10^{-3})	(6.1×10^{-4})
Total	1.472×10^{0}	2.792×10^{-1}
	(2.1×10^{-3})	(6.4×10^{-4})
MC/DANTSYS	0.993	1.026

*One standard deviation

Table 9 Fractions of perturbed source effect in the total sensitivity coefficients for different ν values.

	α (s^{-1})	k_{eff}	Fraction of perturbed source effect
$\nu\times1.00$	6356.8 ± 1.5	0.84066 ± 0.00009	0.119
$\nu\times0.90$	9586.2 ± 1.5	0.75645 ± 0.00009	0.163
$\nu\times0.80$	11692 ± 1.4	0.70251 ± 0.00009	0.205

Table 10 Sensitivity coefficients of α to the capture cross section in the light water (water-hole problem).

	1st Gr.	2nd Gr.	3rd Gr.
DANTSYS	1.854×10^{-3}	3.857×10^{-3}	6.349×10^{-1}
Differential operator (MC)			
Without source perturbation	1.838×10^{-3}	3.781×10^{-3}	6.172×10^{-1}
	$(9\times10^{-7})^{*}$	(1.7×10^{-6})	(2.3×10^{-4})
Perturbed source effect	1.830×10^{-5}	7.485×10^{-5}	1.927×10^{-2}
	(2.83×10^{-6})	(5.41×10^{-6})	(7.8×10^{-4})
Total	1.856×10^{-3}	3.856×10^{-3}	6.364×10^{-1}
	(3.0×10^{-6})	(5.7×10^{-6})	(8.1×10^{-4})
MC/DANTSYS	1.001	1.000	1.002

*One standard deviation

Table 11 Sensitivity coefficients of α to the scattering cross section in the light water (water-hole problem).

	1st Gr.	2nd Gr.
DANTSYS	-1.017×10^{-1}	-8.269×10^{-2}
Differential operator (MC)		
Without source perturbation	-9.415×10^{-2}	-7.329×10^{-2}
	$(4.5\times10^{-4})^{*}$	(1.06×10^{-3})
Perturbed source effect	-9.686×10^{-3}	-1.189×10^{-2}
	(1.356×10^{-3})	(3.29×10^{-3})
Total	-1.038×10^{-1}	-8.518×10^{-2}
	(1.4×10^{-3})	(3.46×10^{-3})
MC/DANTSYS	1.020	1.030

*One standard deviation

Table 12 Sensitivity coefficients with the superhistory method for the case of Table 4 (capture cross section, MOX problem).

	1st Gr.	2nd Gr.	3rd Gr.
DANTSYS	8.891×10^{-3}	2.182×10^{-1}	7.653×10^{-1}
Differential operator (MC)			
Source perturbation iteration method	8.852×10^{-3}	2.174×10^{-1}	7.624×10^{-1}
	(1.3×10^{-5}) *	(3.2×10^{-4})	(1.0×10^{-3})
Superhistory method	8.866×10^{-3}	2.167×10^{-1}	7.634×10^{-1}
8 supergenerations	(2.1×10^{-5})	(5.0×10^{-4})	(1.7×10^{-3})
FOM of superhistory method**	0.097	0.145	0.104

*One standard deviation

**Relative FOM of the superhistory method with respect to the source perturbation iteration method

Table 13 Sensitivity coefficients with the superhistory method for the case of Table 5 (fission cross section, MOX problem).

	1st Gr.	2nd Gr.	3rd Gr.
DANTSYS	-6.683×10^{-2}	-5.398×10^{-2}	-5.650×10^{-1}
Differential operator (MC)			
Source perturbation iteration method	-6.691×10^{-2}	-5.320×10^{-2}	-5.630×10^{-1}
	(3.5×10^{-4})*	(3.4×10^{-4})	(1.3×10^{-3})
Superhistory method	-6.662×10^{-2}	-5.321×10^{-2}	-5.644×10^{-1}
8 supergenerations	(4.4×10^{-4})	(4.2×10^{-4})	(2.0×10^{-3})
FOM of superhistory method**	0.171	0.172	0.124

*One standard deviation

**Relative FOM of the superhistory method with respect to the source perturbation iteration method

Table 14 Sensitivity coefficients with the superhistory method for the case of Table 6 (scattering cross section, MOX problem).

	1st Gr.	2nd Gr.
DANTSYS	-3.327×10^{-1}	-4.222×10^{-1}
Differential operator (MC)		
Source perturbation iteration method	-3.309×10^{-1}	-4.285×10^{-1}
	$(2.1\times10^{-3})^{*}$	(4.6×10^{-3})
Superhistory method	-3.311×10^{-1}	-4.231×10^{-1}
8 supergenerations	(1.9×10^{-3})	(4.1×10^{-3})
FOM of superhistory method**	0.161	0.182

*One standard deviation

**Relative FOM of the superhistory method with respect to the source perturbation iteration method

Table 15 Sensitivity coefficients with the superhistory method for the case of Table 7 (fission spectrum, MOX problem).

	1st Gr.	2nd Gr.
DANTSYS	1.482×10^{0}	2.721×10^{-1}
Differential operator (MC)		
Source perturbation iteration method	1.472×10^{0}	2.792×10^{-1}
	$(2.1\times10^{-3})^{*}$	(6.4×10^{-4})
Superhistory method	1.475×10^{0}	2.780×10^{-1}
8 supergenerations	(3.3×10^{-3})	(9.1×10^{-4})
FOM of superhistory method**	0.152	0.190

*One standard deviation

**Relative FOM of the superhistory method with respect to the source perturbation iteration method

Table 16 Sensitivity coefficients with the superhistory method for the case of Table 8 (capture cross section, water-hole problem).

	1st Gr.	2nd Gr.	3rd Gr.
DANTSYS	1.854×10^{-3}	3.857×10^{-3}	6.349×10^{-1}
Differential operator (MC)			
Source perturbation iteration method	1.856×10^{-3}	3.856×10^{-3}	6.364×10^{-1}
	$(3.0\times10^{-6})^{*}$	(5.7×10^{-6})	(8.1×10^{-4})
Superhistory method	1.854×10^{-3}	3.845×10^{-3}	6.351×10^{-1}
8 supergenerations	(4.4×10^{-6})	(8.7×10^{-6})	(1.2×10^{-3})
FOM of superhistory method**	0.127	0.087	0.086

*One standard deviation

**Relative FOM of the superhistory method with respect to the source perturbation iteration method

Table 17 Sensitivity coefficients with the superhistory method for the case of Table 9 (scattering cross section, water-hole problem).

	1st Gr.	2nd Gr.
DANTSYS	-1.017×10^{-1}	-8.269×10^{-2}
Differential operator (MC)		
Source perturbation iteration method	-1.038×10^{-1}	-8.518×10^{-2}
	$(1.4\times10^{-3})^{*}$	(3.46×10^{-3})
Superhistory method	-1.003×10^{-1}	-8.129×10^{-2}
8 supergenerations	(1.2×10^{-3})	(3.03×10^{-3})
FOM of superhistory method**	0.240	0.247

*One standard deviation

**Relative FOM of the superhistory method with respect to the source perturbation iteration method