Annals of Nuclear Energy 60 (2013) 295–300

Contents lists available at SciVerse ScienceDirect

Annals of Nuclear Energy

journal homepage: www.elsevier.com/locate/anucene

The stochastic implicit Euler method – A stable coupling scheme for Monte Carlo burnup calculations

Jan Dufek [a,*], Dan Kotlyar [b], Eugene Shwageraus [b]

[a] Royal Institute of Technology, Nuclear Reactor Technology, AlbaNova University Center, 10691 Stockholm, Sweden
[b] Department of Nuclear Engineering, Ben-Gurion University, POB 653, Beer Sheva 84105, Israel

ARTICLE INFO

Article history:
Received 11 February 2013
Received in revised form 14 May 2013
Accepted 16 May 2013
Available online 10 June 2013

Keywords:
Monte Carlo burnup calculations
Coupling schemes
Numerical stability

ABSTRACT

Existing Monte Carlo burnup codes use various schemes to solve the coupled criticality and burnup equations. Previous studies have shown that the coupling schemes of the existing Monte Carlo burnup codes can be numerically unstable. Here we develop the Stochastic Implicit Euler method – a stable and efficient new coupling scheme. The implicit solution is obtained by the stochastic approximation at each time step. Our test calculations demonstrate that the Stochastic Implicit Euler method can provide an accurate solution to problems where the methods in the existing Monte Carlo burnup codes fail.

1. Introduction

A large number of Monte Carlo burnup codes have been developed over the last decade, linking various Monte Carlo neutron transport codes to various depletion codes or inbuilt procedures, e.g. MCB 2 (Cetnar et al., 2000), MOCUP (Moore et al., 1995), MONTEBURNS 2 (Poston and Trellue, 1999), MCNPX 2.6 (Fensin et al., 2010), SERPENT (OECD/NEA Data Bank Computer Program Services, 2010), ALEPH (Haeck and Verboomen, 2006), and many others. While adding the depletion capability to Monte Carlo neutron transport codes vastly broadens the possible applicability of Monte Carlo calculations, this has not been fully appreciated in practice yet. The major reason for this has been the great computational demands required by the Monte Carlo codes to calculate accurate neutron flux distributions and group cross sections that are needed for depletion calculations. Thus, the use of Monte Carlo burnup codes has been limited to research and to simple-geometries. Consequently, the knowledge collected from using the Monte Carlo burnup codes has been very limited; yet it is generally believed that Monte Carlo burnup codes give results superior to deterministic core simulators when given the required computing power. Nevertheless, the continuously decreasing computing power cost starts to allow the Monte Carlo burnup codes to perform the full-core fuel cycle calculations, and the weak points of existing Monte Carlo burnup codes are becoming apparent (Dufek and Hoogenboom, 2009; Dufek et al., 2013).

One of the major problems of the existing Monte Carlo codes that became apparent recently is their numerical instability, manifesting itself by spatial oscillations of the neutron flux or by diverging the neutron flux to vastly incorrect spatial distribution over the subsequent time steps, see Dufek and Hoogenboom (2009) and Dufek et al. (2013). This instability is not driven by statistical errors in the computed neutron flux; it is driven by the feedback between the neutron flux and the nuclide field – nuclides with large neutron reaction rates, such as the fissile nuclides, fission products with large neutron cross sections, and burnable absorbers. In thermal reactors, the strongest feedback can be attributed to ^{135}Xe; this feedback is natural, and it can lead to well-known xenon oscillations in thermal nuclear reactors. These oscillations are, however, normally suppressed actively by the control system; steady-state conditions are ensured during the normal operation. The fuel cycle calculations must therefore ensure steady-state conditions as well; any departure of the neutron flux from the steady-state will introduce errors in the fuel depletion. This requirement translates into a constraint imposed on the coupled system of criticality and burnup equations.

Methods in deterministic nodal simulators have been tuned to perfection; their stability is achieved by enforcing the steady-state xenon distribution or by using stable implicit methods for solving the coupled criticality and burnup equation. Nevertheless, the gained knowledge has not been reflected into the methods in the Monte Carlo burnup codes yet. The existing Monte Carlo burnup codes adopt either the explicit Euler method, or the mid-point method or other predictor–corrector methods; however, all these methods are known to be only conditionally stable (Hoffman, 2001). This means the methods are stable only when the target

[*] Corresponding author. Tel.: +46 76 337 4280.
E-mail address: jandufek@kth.se (J. Dufek).

0306-4549/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.anucene.2013.05.015

time period is discretised into sufficiently small time steps. The existing Monte Carlo burnup codes may thus become numerically unstable (Dufek and Hoogenboom, 2009; Dufek et al., 2013) as the calculations are commonly performed with large time steps. Enforcing the steady-state xenon distribution could indeed stabilise the Monte Carlo burnup calculations (Isotalo et al., 2013); however, ^{135}Xe does not represent the only feedback. Other, especially the fissile nuclides can drive the numerical instability in burnup calculation with time steps of several weeks or months (Dufek and Gudowski, 2005). Therefore, in this paper, we choose the stable implicit Euler method for Monte Carlo burnup calculations. We attempt to design not only a stable scheme, but also a scheme that is efficient when applied to Monte Carlo burnup calculations.

The paper is organised as follows. Section 2 states the governing equations. Section 3 briefly describes the coupling schemes in the existing Monte Carlo burnup schemes. In Section 4 we derive the Stochastic Implicit Euler (SIE) method – the new Monte Carlo burnup coupling scheme, and suggest the possible ways of implementing the method in Monte Carlo burnup codes. Section 5 shows results of numerical test calculations demonstrating the numerical stability of the SIE method. Section 6 summarises our conclusions.

2. Governing equations

The geometry and material properties of the whole reactor can be described by the nuclide field $\mathbf{N}(\mathbf{r})$; the elements of this vector denote concentrations of various nuclides at the position $\mathbf{r}$. The fundamental-mode neutron flux $\phi(\mathbf{r},\mathbf{\Omega},E)$ is then determined by $\mathbf{N}(\mathbf{r})$ and the boundary conditions.

The nuclide field $\mathbf{N}(\mathbf{r})$ changes in nuclear reactors during the operation due to the depletion process driven by the neutron flux $\phi(\mathbf{r},\mathbf{\Omega},E)$. Since the neutron flux is determined by the nuclide field, the fundamental-mode neutron flux changes during the reactor operation as well. The purpose of fuel cycle calculations is to determine the changes in $\mathbf{N}(\mathbf{r},t)$ and $\phi(\mathbf{s}) \equiv \phi(\mathbf{r},\mathbf{\Omega},E,t)$ during the whole fuel cycle.

The above problem can be described by two coupled equations: the burnup equation that describes the time change of the nuclide field, and the criticality (eigenvalue) neutron transport equation that gives the fundamental-mode neutron flux in the core. The burnup equation (Bell and Glasstone, 1970),

$$\frac{\mathrm{d}\mathbf{N}(\mathbf{r},t)}{\mathrm{d}t} = \mathbb{M}(\phi,T)\mathbf{N}(\mathbf{r},t), \tag{1}$$

is an ordinary differential equation where

$$\mathbb{M}(\phi) = \int_0^\infty \phi(\mathbf{r},E,t)\mathbb{X}(T)\ \mathrm{d}E + \mathbb{D},$$

where $\mathbb{X}$ is a cross-section and fission yield matrix, $\mathbb{D}$ is a decay matrix, and $T(\mathbf{r},t)$ is the temperature at $\mathbf{r}$ in time t. Eq. (1) has a formal solution (Bell and Glasstone, 1970)

$$\mathbf{N}(\mathbf{r},t) = \mathbf{N}_0(\mathbf{r})\exp[\mathbb{M}(\phi,T)(t-t_0)], \tag{2}$$

where $\mathbf{N}_0(\mathbf{r})$ is the nuclide field at time t_0. The neutron flux $\phi(\mathbf{r},E,t)$ is approximated at time t by the fundamental-mode eigenfunction of the criticality equation

$$B(\mathbf{N})\phi(\mathbf{s}) \equiv [L(\mathbf{N}) - \frac{1}{k}F(\mathbf{N})]\phi(\mathbf{s}) = 0, \tag{3}$$

where $L(\mathbf{N})\phi(\mathbf{s})$ represents the migration and loss of neutrons from $\mathbf{s}$, and $F(\mathbf{N})\phi(\mathbf{s})$ accounts for neutron production in $\mathbf{s}$ due to fission.

As explained in Section 1, the core conditions are required to be steady-state at all time steps of the fuel cycle calculations. Even the natural xenon oscillations that could develop in the cycle calculations with short time steps of few hours are to be prevented. The development of natural xenon oscillations in burnup calculations with short time steps can be prevented by forcing the concentration of ^{135}Xe to its saturated level; i.e., a level that is naturally established for $t \to \infty$ (with the neutron flux and fission rate fixed). Nevertheless, the saturated concentration of ^{135}Xe is reached practically after several days; thus, this additional constrain to the above system of equations is not necessary when the time steps are larger than several days. We wish to point out the difference between the saturated and the steady-state (equilibrium) xenon concentration. While the equilibrium xenon concentration that is in steady-state with the neutron flux can be computed only in an iterative manner (Dufek and Gudowski, 2006), the saturated xenon concentration can be derived directly from the burnup equation. Indeed, enforcing the saturated xenon concentration cannot ensure the steady-state conditions or the numerical stability; the saturated xenon distribution may undergo spatial oscillations over the subsequent time steps in cycle calculations (of thermal reactors) unless the whole system of equations is solved by a stable numerical method.

Note that in the following text, the fundamental-mode flux $\phi(\mathbf{s})$ that satisfies Eq. (3) with the operator $B(\mathbf{N})$ is denoted as $\phi_{B(\mathbf{N})}$. In Section 4.2, $\phi_{B(\mathbf{N})}$ specifically denotes the fundamental-mode neutron flux that is computed by a Monte Carlo criticality code in a reactor with the nuclide field $\mathbf{N}(\mathbf{r})$.

3. Coupling schemes in existing codes

The existing Monte Carlo burnup codes use various schemes to couple the burnup and criticality equations. The detail description of the schemes was given by Dufek and Hoogenboom (2009) and Dufek et al. (2013), therefore we describe the schemes only briefly in this section.

The simplest scheme employs the explicit (forward) Euler method. The codes that use this method (MCB 2, MOCUP, ALEPH 1.1.2, and many others) divide the target time interval into a number of time steps, and assume the neutron flux at the beginning of the step to remain constant over the step. The flux is computed at the beginning of each step, and the fuel is depleted over the whole step using the beginning-of-step flux.

Other Monte Carlo burnup codes use various predictor–corrector methods. For instance, a large group of codes (e.g. MONTEBURNS 2 and MCNPX 2.6) uses the modified midpoint method (Hoffman, 2001). These codes deplete the fuel over the time step with the middle-of-step flux. This flux can be obtained after a trial depletion of the fuel using the beginning-of-step flux. Some codes can iterate the middle-of-step flux several times by re-depleting the fuel with the middle-of-step flux obtained in previous iteration steps.

Some deterministic and Monte Carlo burnup codes (e.g. SERPENT 1.1.18) use the linear flux approximation (Yang and Downar, 1990) – another predictor–corrector method. This method depletes the fuel over the time step with the neutron flux that is taken as an average of the beginning-of-step and end-of-step fluxes. This is, in fact, the modified Euler method (Hoffman, 2001). The fuel is first roughly depleted over the whole step with the beginning-of-step flux, and an estimate of the end-of-step flux is computed. The average of the both fluxes is then used to re-deplete the fuel over the time step. Some codes can recalculate the end-of-step flux, and iterate the average flux several times.

Few of the existing Monte Carlo burnup codes use other predictor–corrector methods. For instance, MCODE uses a simple predictor–corrector method applied to the nuclide field; this method is also used in deterministic burnup calculations (Stamm'ler and Abbate, 1983). The method first depletes the fuel with the beginning-of-step flux over the whole step (this gives the predicted end-of-step nuclide field), and calculates the estimate of the end-of-step flux. Then the fuel is re-depleted with the end-of-step flux over the whole step (this gives the corrected end-of-step nuclide field). The final end-of-step nuclide field is taken as the average of the predicted and corrected end-of-step nuclide fields.

All the above methods are only conditionally stable (Hoffman, 2001); i.e., they are stable when the time steps are sufficiently short. The strong feedbacks make these methods prone to numerical instabilities (with respect to neutron flux distribution and thus the fuel depletion) in fuel cycle calculations of critical nuclear reactors, as shown by Dufek and Hoogenboom (2009) and Dufek et al. (2013).

4. The Stochastic implicit Euler method

4.1. Derivation of the method

We demand the following basic properties from a Monte Carlo burnup coupling scheme:

- the method must be stable (unconditionally, i.e. with arbitrarily large time steps),
- the method must provide a simple way to improve the solution within the time step when this is requested,
- the method must be efficient when implemented in Monte Carlo burnup codes.

The simplest method that can satisfy the unconditional stability property is the implicit Euler method (Hoffman, 2001). Similarly to all implicit methods, the implicit Euler method uses the future data that can be obtained only by solving an equation at each time step. Translated into the context of burnup problems, the implicit Euler method depletes the fuel over the whole time step with the end-of-step neutron flux and microscopic cross sections.

Note that ϕ in the following text still represents the neutron flux $\phi(\mathbf{r},\mathbf{\Omega},E)$ with the dependences on the position, direction and energy; therefore, the one-group neutron microscopic cross section can be derived from ϕ for the purpose of solving the burnup equation with ϕ. All references to Eq. (2) in this paper thus assume the one-group neutron microscopic cross section were determined by ϕ.

Now we address the problem of obtaining the end-of-step flux. Let $\mathbf{N}_i$ and ϕ_i denote the nuclide field and neutron flux at the end of ith time step, respectively. When $\mathbf{N}_{i-1}$ is depleted with the end-of-step flux ϕ_i over the ith time step then $\mathbf{N}_i$ equals

$$\mathbf{N}_i = \mathbf{N}_{i-1}\exp[\mathbb{M}(\phi_i, T)(t_i - t_{i-1})], \tag{4}$$

while ϕ_i is given by

$$\phi_i = \phi_{B(\mathbf{N}_i)}. \tag{5}$$

Substituting N_i from Eq. (4) into Eq. (5) forms a non-linear equation for ϕ_i,

$$\phi_i = \phi_{B(\mathbf{N}_{i-1}\exp[\mathbb{M}(\phi_i,T)(t_i-t_{i-1})])} \tag{6}$$

For sake of simplicity, let G denote the right-hand side of Eq. (6) as a function of ϕ_i. Then Eq. (6) reduces into the simple form

$$\phi_i = G(\phi_i). \tag{7}$$

In case of Monte Carlo calculation, function G is approximated by a stochastic function $\hat{G}$ that contains an additional noise term ε,

$$\widehat{G} = G + \varepsilon.$$

This changes Eq. (7) into

$$\phi_i = \widehat{G}(\phi_i), \tag{8}$$

which is the non-linear stochastic root-finding problem that has been solved by Dufek and Gudowski (2006).

A similar non-linear equation can be formed for N_i when ϕ_i from Eq. (5) is substituted into Eq. (4), which gives

$$\mathbf{N}_i = \mathbf{N}_{i-1}\exp[\mathbb{M}(\phi_{B(\mathbf{N}_i)}, T)(t_i - t_{i-1})], \tag{9}$$

which can be again reduced into a simple form

$$\mathbf{N}_i = H(\mathbf{N}_i), \tag{10}$$

where H denotes the right-hand side of Eq. (9) as a function of $\mathbf{N}_i$.

In the following text we complete the derivation of the method that can solve Eq. (8); however, the same method can be equally applied to the stochastic version of Eq. (10),

$$\mathbf{N}_i = \widehat{H}(\mathbf{N}_i), \tag{11}$$

as it is demonstrated in Section 4.2.

Non-linear equations may be solved by various numerical methods; however, not all methods can guarantee convergence. For instance, the common fixed-point iteration

$$\phi_i^{(n+1)} = \widehat{G}(\phi_i^{(n)}) \tag{12}$$

cannot ensure convergence unless the equation is transformed into a stable form. Nevertheless, Eq. (8) is too complex for transforming into a stable form; therefore, a stable method is needed.

Non-linear stochastic root-finding problems are often transformed into a stochastic optimisation problem that aims at finding a minimum of a stochastic objective function $\hat{f}$. To find the solution ϕ^* to Eq. (8), the stochastic objective function $\hat{f}$ can be designed, e.g. as

$$\hat{f}(\phi) = |\phi - \widehat{G}(\phi)|. \tag{13}$$

The stochastic optimisation problem can be solved most efficiently by the stochastic approximation method; the stochastic approximation refers to a class of methods that compute or use an approximation of the gradient of the objective function. The method solves Eq. (8) iteratively: given $\phi^{(1)}$, until convergence repeat

$$\phi_i^{(n+1)} = \phi_i^{(n)} - a_n\hat{g}_i^{(n)} \tag{14}$$

where $a_n > 0$ is the step-size, and $\hat{g}_i^{(n)}$ is an estimate of the gradient of the objective function,

$$g_i^{(n)} = \nabla f\left(\phi_i^{(n)}\right). \tag{15}$$

Robbins and Monro (1951) proved that $\phi^{(n)} \to \phi^*$ if $a_n > 0$, $\sum_{n=1}^{\infty} a_n = \infty$, and $\sum_{n=1}^{\infty} a_n^2 < \infty$; these properties are satisfied by the Robbins–Monro algorithm that generates the step-size as

$$a_n = \frac{a}{n} \tag{16}$$

where a is a positive scalar. Often, algorithms use the reduced step-size α_n that is defined as

$$\alpha_n = \frac{a_n}{a}; \tag{17}$$

then the Robbins–Monro algorithm reduces into

$$\alpha_n = \frac{1}{n}. \tag{18}$$

Dufek and Gudowski (2006) showed that the gradient of the objective function can be approximated by a single Monte Carlo criticality calculation, and that the neutron flux can be then iterated as

$$\phi_i^{(n+1)} = \phi_i^{(n)} - \alpha_n d(\phi_i^{(n)} - \widehat{G}(\phi_i^{(n)})) \tag{19}$$

where d is positive and smaller than 1. Unless the initial guess is known to a certain accuracy, we advice to set $d = 1$ to achieve the best efficiency; then Eq. (19) can be re-written into a common relaxation form

$$\phi_i^{(n+1)} = (1-\alpha_n)\phi_i^{(n)} + \alpha_n \widehat{G}\left(\phi_i^{(n)}\right). \tag{20}$$

We would like to point out that Eq. (20) is equivalent to

$$\phi_i^{(n+1)} = \frac{1}{n}\sum_{j=1}^{n}\widehat{G}\left(\phi_i^{(j)}\right) \tag{21}$$

when the Robbins–Monro algorithm is used to generate the reduced step-size. This can be proven by the mathematical induction: for $n = 1$, both Eqs. (20) and (21) give $\phi_i^{(2)} = \widehat{G}(\phi_i^{(1)})$, and for step $n + 1$ Eq. (20) can be written as

$$\begin{aligned}\phi_i^{(n+1)} &= \left(1-\frac{1}{n}\right)\frac{1}{n-1}\sum_{j=1}^{n-1}\widehat{G}\left(\phi_i^{(j)}\right) + \frac{1}{n}\widehat{G}\left(\phi_i^{(n)}\right)\\ &= \frac{1}{n}\sum_{j=1}^{n}\widehat{G}\left(\phi_i^{(j)}\right),\end{aligned}$$

which proves Eq. (21).

Next, we would like to note that Eq. (21) shows that all Monte Carlo criticality calculations are reflected in the final solution with equal statistical weight $1/n$, which ensures a good efficiency of this iteration. The statistical errors in the iterated flux are reduced with each new iteration step.

Finally, we would like to remark that the Robbins–Monro algorithm is suitable for generating the step-size when the sample-size for estimating $G\left(\phi_i^{(n)}\right)$ (number of simulated neutron histories) is fixed over all iteration steps. Dufek and Gudowski (2006) showed that it may be beneficial to vary the sample-size over the iteration steps; for that case Dufek and Gudowski (2006) derived new algorithms that could be considered here. In this paper, however, we choose to use a fixed sample-size over all iteration steps, and thus to employ the Robbins-Monro algorithm.

4.2. Implementation of the method

Here, we describe the possible implementation of the Stochastic Implicit Euler method that was derived in Section 4.1. As explained above, an inner iteration is performed at each time step; either the neutron flux or the nuclide field is relaxed during this iteration. Therefore two basic implementations are possible; Algorithm 1 describes the Stochastic Implicit Euler method with relaxation of the neutron flux, while Algorithm 2 describes the Stochastic Implicit Euler method with relaxation of the nuclide field.

Note that Algorithms 1 and 2 are purely schematic. The algorithms make use of Eq. (21); however, the relaxed neutron flux or the nuclide density could equally be described by Eqs. (19) and (18). Indeed, Monte Carlo burnup codes do not need to remember results of all inner iteration steps; instead the sum of the nuclide fields or neutron fluxes can be simply updated at each inner iteration step. Especially Algorithm 1 can be implemented easily as the calculation of the relaxed neutron flux and group cross sections can simply combine samples over the inner iteration steps.

Algorithm 2 may be less suitable for implementing in Monte Carlo burnup codes since the fuel is always depleted with a neutron flux that may have possibly large statistical errors (as the flux is computed in a single inner iteration step), and it is hard to predict the effect of very large errors in the flux on the solution of the depletion equation. The errors in the relaxed nuclide field decrease only thanks to averaging the nuclide field over the inner iteration steps. It may also be seen as a disadvantage that the inner iteration in Algorithm 2 does not naturally combine the neutron fluxes, since the neutron flux is often a required result from cycle calculations. Therefore we have included an additional line #12 in Algorithm 2 where the neutron fluxes from the inner steps are combined at the end of each time step; the combined flux is also used as the beginning-of-step flux in the next time step.

Algorithm 1. The Stochastic Implicit Euler method with relaxation of the neutron flux.

```
1:  input: $\mathbf{N}_0$
2:  $\phi_0 \leftarrow \phi_{B(\mathbf{N}_0)}$
3:  for $i \leftarrow 0, 1, \ldots,$ do
4:    $\mathbf{N}_{i+1}^{(0)} \leftarrow \mathbf{N}_i \exp[\mathbb{M}(\phi_i)\Delta t]$
5:    for $n \leftarrow 1, 2, \ldots, c$ do
6:      $\phi_{i+1}^{(n)} \leftarrow \phi_{B(\mathbf{N}_{i+1}^{(n-1)})}$
7:      $\bar{\phi}_{i+1}^{(n)} \leftarrow \sum_{j=1}^{n}\phi_{i+1}^{(j)}/n$
8:      $\mathbf{N}_{i+1}^{(n)} \leftarrow \mathbf{N}_i \exp[\mathbb{M}(\bar{\phi}_{i+1}^{(n)})\Delta t]$
9:    end for
10:   $\mathbf{N}_{i+1} \leftarrow \mathbf{N}_{i+1}^{(c)}$
11:   $\phi_{i+1} \leftarrow \bar{\phi}_{i+1}^{(c)}$
12: end for
```

Algorithm 2. The Stochastic Implicit Euler method with relaxation of the nuclide field.

```
1:  input: $\mathbf{N}_0$
2:  $\phi_0 \leftarrow \phi_{B(\mathbf{N}_0)}$
3:  for $i \leftarrow 0, 1, \ldots,$ do
4:    $\bar{\mathbf{N}}_{i+1}^{(0)} \leftarrow \mathbf{N}_i \exp[\mathbb{M}(\phi_i)\Delta t]$
5:    for $n \leftarrow 1, 2, \ldots, c$ do
6:      $\phi_{i+1}^{(n)} \leftarrow \phi_{B(\bar{\mathbf{N}}_{i+1}^{(n-1)})}$
7:      $\mathbf{N}_{i+1}^{(n)} \leftarrow \mathbf{N}_i \exp[\mathbb{M}\left(\phi_{i+1}^{(n)}\right)\Delta t]$
8:      $\bar{\mathbf{N}}_{i+1}^{(n)} \leftarrow \sum_{j=1}^{n}\mathbf{N}_{i+1}^{(j)}/n$
9:    end for
10:   $\mathbf{N}_{i+1} \leftarrow \bar{\mathbf{N}}_{i+1}^{(c)}$
11:   $\phi_{i+1} \leftarrow \sum_{j=1}^{c}\phi_{i+1}^{(j)}/c$
12: end for
```

5. Numerical test calculations

5.1. Numerical test model

Numerical tests in this section demonstrate the stability of Algorithms 1 and 2. For this purpose we use the same fuel cell model as defined by Dufek et al. (2013). The model represents a square fuel cell with the following properties:

Fuel	UO_2
Cladding material	Zr
Moderator	light water
Radius of fuel pellets	0.41 cm
Outer radius of cladding	0.475 cm
Rod pitch	1.26 cm
Length of the fuel rod	300 cm
U enrichment in ^{235}U	3.1 wt.%
Fuel density	10 g/cm^3
Water density	0.7 g/cm^3
Linear power rating	40 kW/m

Reflective boundary conditions are imposed on all six sides of the cell; therefore, the cell has no neutron leakage. To allow for the fuel depletion according to the local neutron flux, the cell is divided into eight equidistant space zones along the fuel rod, and the fuel material in each zone is defined independently. The codes were set up to deplete the fuel in each zone independently according to the actual neutron flux in the zone.

This model is suitable for studying the numerical stability of coupling schemes for several reasons. First, the model has a large dominance ratio, which is a condition at which the numerical instabilities may develop (when an unstable coupling scheme is used). Then, the system has no neutron leakage; thus, the correct steady-state flux (and the fuel burnup) must be uniform along the fuel rod at any time, which makes this model convenient to monitor errors in the computed flux. Indeed, no instability could possibly develop without the definition of independent burnable materials.

In this model, the relative error in flux ϕ computed by the Monte Carlo burnup code can be found at any fuel burnup as

$$\varepsilon(\phi) = \sum_{z=1}^{8} \left| \frac{\langle \phi \rangle_z}{\langle \phi \rangle} - \frac{1}{8} \right|, \tag{22}$$

where $\langle \phi \rangle_z$ is the flux integrated over energy, angle, and zone z, and

$$\langle \phi \rangle = \sum_{z=1}^{8} \langle \phi \rangle_z.$$

5.2. Results

Algorithm 1 was tested by a modified version of the SERPENT 2 code using the ENDFB 7 library, and Algorithm 2 was tested by a modified version of the BGCore code (Fridman et al., 2008) using the JEFF3.1 library.

The numerical stability of each algorithm is demonstrated on four independent burnup calculations with the time step size of 7, 14, 30 and 60 days. Each burnup calculation consisted of 10 time steps. Each time step performed 11 inner iterations. Each criticality calculation simulated a batch of 5000 neutrons in 1000 inactive and 5000 active cycles. Note that about the same number of neutron histories was simulated at each time step in the published test calculations of the predictor–corrector coupling schemes in existing Monte Carlo burnup codes (Dufek et al., 2013); the results thus can be directly compared. The initial fission source was uniformly distributed in the fuel.

We would like to note that the choice of the optimal number of the inner iteration steps is not addressed in this paper. In this section we only demonstrate the feasibility of stable Monte Carlo burnup calculations, and the rather large number of inner iteration steps serves this purpose.

Test results of Algorithm 1 are summarised in Table 1. The table contains the relative errors of the neutron flux ϕ_i (line #11 in Algorithm 1) at all burnup calculation. As this flux is directly used in the fuel depletion its errors thus represent also errors in the fuel depletion.

Test results of Algorithm 2 are summarised in Table 2. Algorithm 2 applies the relaxation on the nuclide fields; nevertheless, the neutron fluxes obtained in the inner iteration steps are used in the depletion process equally. Therefore, for the purpose of evaluating the stability we find it reasonable to combine the fluxes over the inner steps (line #12 in Algorithm 2) and demonstrate the errors in the average flux here.

Table 1
Relative error $\varepsilon(\phi_i)$ in the neutron flux in the test of Algorithm 1 (%), computed by SERPENT.

i	Δt (days)			
	7	14	30	60
1	0.27	0.13	0.36	0.27
2	0.07	0.63	0.28	0.21
3	0.15	0.63	0.47	0.28
4	0.19	0.51	0.55	0.30
5	0.18	0.56	0.28	0.20
6	0.23	1.25	0.28	0.19
7	0.59	0.94	0.12	0.37
8	0.60	0.47	0.56	1.06
9	0.35	0.44	0.22	0.38
10	0.18	0.23	0.21	1.19

Table 2
Relative error $\varepsilon(\phi_i)$ in the neutron flux in the test of Algorithm 2 (%), computed by BGCore.

i	Δt (days)			
	7	14	30	60
1	0.52	0.76	0.31	3.50
2	0.46	0.39	0.76	1.45
3	0.34	0.35	0.42	0.41
4	0.39	0.21	0.18	0.37
5	0.23	0.14	0.12	0.24
6	0.78	0.13	0.50	0.39
7	0.73	0.26	0.24	0.26
8	0.48	0.71	0.24	0.27
9	0.89	0.28	0.47	0.38
10	0.62	0.38	0.48	0.34

Tables 3 and 4 contain corresponding results from published tests of the predictor–corrector schemes common in the existing Monte Carlo burnup codes (Dufek et al., 2013). Table 3 provides results of the linear flux approximation - the default predictor–corrector method in the SERPENT code. Table 4 gives results of the standard predictor–corrector method applied on the nuclide field.

For readers' convenience we also include a graphical demonstration of the numerical instability of the standard predictor–corrector coupling scheme here; the previously published Fig. 1 depicts spatial oscillations of the neutron flux in the considered system during the first four time steps of a depletion calculation with time steps Δt = 30 day.

The above results demonstrate the good stability of the Stochastic Implicit Euler method. While the existing predictor–corrector schemes suffer from the poor numerical stability, the small errors in the neutron flux in the test calculations of the Stochastic Implicit Euler method are mainly of the statistical nature. The two basic implementations of the Stochastic Implicit Euler method (Algorithms 1 and 2) show very similar results in our test calculations.

Table 3
Relative error $\varepsilon(\phi_i)$ in the neutron flux in the test of the linear flux approximation (%), computed by SERPENT.

i	Δt (days)			
	7	14	30	60
1	0.1	0.6	0.7	4.2
2	0.7	0.5	9.8	28.2
3	0.4	0.4	19.3	14.3
4	0.9	1.0	13.3	10.5
5	1.0	2.5	7.9	2.1
6	1.4	3.9	4.2	3.2
7	0.1	6.1	3.1	2.0
8	0.2	9.3	2.1	2.3
9	2.0	8.8	1.3	1.6
10	1.9	6.2	1.0	2.1

Table 4
Relative error $\varepsilon(\phi_i)$ in the neutron flux in the test of the standard predictor–corrector scheme (%), computed by BGCore.

i	Δt (days)			
	7	14	30	60
1	0.6	0.6	1.0	1.5
2	0.1	0.5	7.9	24.5
3	0.3	1.3	19.8	24.2
4	1.1	1.5	15.9	10.0
5	0.2	2.9	7.8	8.0
6	1.1	5.9	6.0	5.1
7	0.2	10.0	4.5	4.2
8	0.7	8.7	2.8	4.2
9	0.8	7.1	3.3	4.2
10	0.3	5.1	3.3	4.3

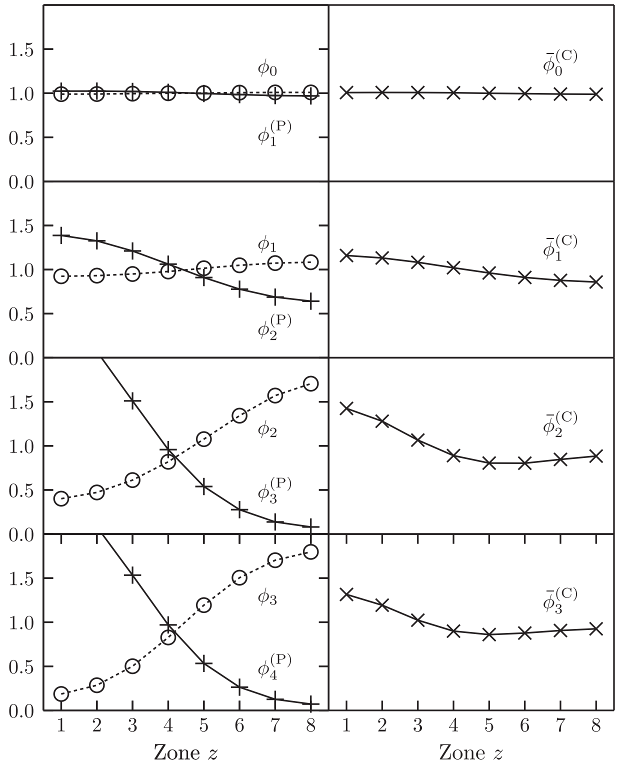

Fig. 1. Neutron fluxes in the test of the predictor–corrector method in the SERPENT code ($\Delta t = 30$ day); ϕ_i is the neutron flux computed at the beginning of ith time step, $\phi_{i+1}^{(P)}$ is the neutron flux predicted at the end of ith time step, and $\bar{\phi}_i^{(C)}$ is the average of ϕ_i and $\phi_{i+1}^{(P)}$ according to which the fuel is depleted over the ith time step. Published by Dufek et al. (2013).

6. Conclusions

We have derived the Stochastic Implicit Euler method - a new coupling scheme for Monte Carlo burnup calculations, as a solution to the poor numerical stability of the coupling schemes in existing Monte Carlo burnup codes.

We have suggested two possible ways of implementing the Stochastic Implicit Euler method, by relaxing either the neutron flux or the nuclide fields at each time step. We wish to point out that by relaxing the neutron flux the relaxation is automatically applied on the nuclide field through the burnup equation; similarly, by relaxing the nuclide field the relaxation is automatically applied on the neutron flux through the criticality equation. Simultaneous explicit relaxation of both the neutron flux and the nuclide field would slow down the convergence of the inner iteration.

The included numerical tests demonstrate the good stability of the method; stable burnup calculations are possible even with large time steps. We still advise to set the time steps as short as possible in order to simulate accurately the spatial and spectral changes in the neutron flux in time. The error of the Stochastic Implicit Euler method is proportional to the time step length.

We wish to point out that the purpose of this paper was to derive and demonstrate a stable coupling scheme for Monte Carlo burnup calculations. Specific optimisations of this scheme, such as the choice of the time step length, the number of inner iterations, the number of neutron histories simulated at each Monte Carlo criticality calculation, and the number of inactive and active source cycles, were not addressed in this paper. Therefore, the number of inner iteration steps was fixed in the test calculations; however, the inner iteration can in principle continue until a specific convergence criterion is satisfied.

The future studies may consider higher-order stable methods combined with the stochastic approximation for obtaining the implicit solution at each time step. Possible improvement may also be achieved by altering the Robbins–Monro algorithm for algorithms that control both step-size and sample-size.

Acknowledgements

This work was partially funded by the European Commission under the High Performance Monte Carlo Reactor Core Analysis (HPMC) Project, within the 7th EU Framework Program, Project Number 295971. We are much obliged to A.E. Isotalo (Aalto University, Finland) for implementing the method in SERPENT.

References

Bell, G.I., Glasstone, S., 1970. Nuclear Reactor Theory. Van Nostrand Reinhold Company, New York, NY (10001).

Cetnar, J., Gudowski, W., Wallenius, J., 2000. User Manual for Monte-Carlo Continuous Energy Burnup (MCB) Code-Version 1C. Royal Institute of Technology, Stockholm.

Dufek, J., Gudowski, W., 2005. Managing xenon oscillations in Monte-Carlo burnup simulations of thermal reactors. In: XII Meeting on Reactor Physics Calculations in the Nordic Countries, Halden.

Dufek, J., Gudowski, W., 2006. Stochastic approximation for Monte Carlo calculation of steady-state conditions in thermal reactors. Nucl. Sci. Eng. 152 (3), 274–283.

Dufek, J., Hoogenboom, J.E., 2009. Numerical stability of existing monte carlo burnup codes in cycle calculations of critical reactors. Nucl. Sci. Eng. 162 (3), 307–311.

Dufek, J., Kotlyar, D., Shwageraus, E., Leppänen, J., 2013. Numerical stability of the predictor-corrector method in Monte Carlo burnup calculations of critical reactors. Ann. Nucl. Energy 56, 34–38.

Fensin, M.L., Hendricks, J.S., Anghaie, S., 2010. The enhancements and testing for the MCNPX 2.6.0 depletion capability. Nucl. Technol. 170 (1), 68–79.

Fridman, E., Shwageraus, E., Galperin, A., 2008. Efficient generation of one-group cross sections for coupled monte carlo depletion calculations. Nucl. Sci. Eng. 159 (1), 37–47.

Haeck, W., Verboomen, B., January 2006. ALEPH 1.1.2 – Aa Monte Carlo Burn-up Code. Tech. Rep. ISSN 1379-2407, Belgian Nuclear Research Centre, Mol, Belgium.

Hoffman, J.D., 2001. Numerical Methods for Engineers and Scientists, 2nd ed. Marcel Dekker, Inc., New York.

Isotalo, A.E., Leppänen, J., Dufek, J., 2013. Preventing xenon oscillations in Monte Carlo burnup calculations by enforcing equilibrium xenon distribution. Ann. Nucl. Energy, 60, 78–85, ISSN 0306-4549.

Moore, R.L., Schnitzler, B.G., Wemple, C., Babcock, R., Wessol, D., September 1995. MOCUP: MCNP-ORIGEN 2 Coupled Utility Program. Tech. Rep. INEL-95/0523, Idaho National Engineering Laboratory.

OECD/NEA Data Bank Computer Program Services, 2010. SERPENT – a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code, NEA-1840. <http://www.oecd-nea.org/tools/abstract/detail/nea-1840>.

Poston, D.I., Trellue, H.R., September 1 1999. Users Manual, Version 2.0 for Monteburns, Version 1.0. 2nd ed.

Robbins, H., Monro, S., 1951. A stochastic approximation method. Ann. Math. Stat. 22 (3), 400–407.

Stamm'ler, R.J., Abbate, M.J., 1983. Methods of Steady-State Reactor Physics in Nuclear Design. Academic Press, London – New York, pp. 388.

Yang, W.S., Downar, T.J., 1990. A linear time-dependent flux approximation for nuclide depletion based on a perturbation method. Ann. Nucl. Energy 17 (1), 37–42.