Annals of Nuclear Energy 56 (2013) 34–38

Contents lists available at SciVerse ScienceDirect

Annals of Nuclear Energy

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

Numerical stability of the predictor–corrector method in Monte Carlo burnup calculations of critical reactors

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

[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
[c] VTT Technical Research Centre of Finland, P.O. Box 1000, FI-02044 VTT, Finland

ARTICLE INFO

Article history:
Received 12 November 2012
Received in revised form 8 January 2013
Accepted 9 January 2013
Available online 10 February 2013

Keywords:
Monte Carlo burnup calculations
Predictor–corrector method
Numerical stability

ABSTRACT

Monte Carlo burnup codes use various schemes to solve the coupled criticality and burnup equations. Previous studies have shown that the simplest methods, such as the beginning-of-step and middle-of-step constant flux approximations, are numerically unstable in fuel cycle calculations of critical reactors. Here we show that even the predictor–corrector methods that are implemented in established Monte Carlo burnup codes can be numerically unstable in cycle calculations of large systems.

1. Introduction

The geometry and material properties of a nuclear reactor can be described by the nuclide field $\mathbf{N}(\mathbf{r},t)$; elements of this vector denote concentrations of various nuclides at $\mathbf{r}$ and time t. Knowing $\mathbf{N}(\mathbf{r},t)$ and the boundary conditions allows then to determine the neutron flux $\phi(\mathbf{s}) \equiv \phi(\mathbf{r},\mathbf{\Omega},E,t)$. While $\phi(\mathbf{s})$ is directly dependent on $\mathbf{N}(\mathbf{r},t)$, the nuclide field is affected by the depletion process over time; this process is driven by the neutron flux. Thus, the neutron flux $\phi(\mathbf{s})$ as well as the nuclide field $\mathbf{N}(\mathbf{r},t)$ change in time. The purpose of the fuel cycle calculations is to determine as to how these fields change during the whole fuel cycle.

The above problem can be described by two coupled equations: the burnup (ODE) equation that determines the nuclide field changes, and the criticality (eigenvalue) equation that gives the fundamental neutron flux in the core (described below). The existing Monte Carlo burnup codes use various schemes to couple the burnup and criticality equations. Here we continue our previous study (Dufek and Hoogenboom, 2009) of the numerical stability of the coupling schemes used in the existing Monte Carlo burnup codes.

Naturally, not all methods for solving ODE systems are stable. Yet, good results may be obtained with any method when the specific ODE system allows it. Unfortunately, the system of equations that describes the fuel cycle of critical reactors needs to be solved with stable methods. The actual reason of the problematic numerical stability is the presence of very strong feedbacks between the neutron flux and the nuclide field. In thermal reactors, ^{135}Xe accounts for the strongest feedback. Indeed, this represents a problem not only for Monte Carlo burnup codes, but for deterministic codes as well. Deterministic methods in nodal core simulators either use implicit depletion methods or enforce the steady-state xenon distribution to ensure the numerical stability.

As it was shown in the previous work (Dufek and Hoogenboom, 2009), existing Monte Carlo burnup codes use conditionally stable schemes, i.e., schemes that are stable only for sufficiently short time steps. Nevertheless, very short steps often cannot be used due to the high computational cost of the Monte Carlo method. It has been shown that the simplest methods, such as the beginning-of-step and middle-of-step constant flux approximations, are numerically unstable in fuel cycle calculations of critical reactors even with relatively short time steps (Dufek and Hoogenboom, 2009).

In this paper we study the numerical stability of the predictor–corrector methods that are adopted in a number of established Monte Carlo burnup codes. The predictor–corrector methods typically perform two iterations at each time step, solving the burnup and criticality equation at each iteration. The burnup equation describes the time change of the nuclide field $\mathbf{N}$ (Bell and Glasstone, 1970),

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

where

* Corresponding author.
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.2013.01.018

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

where $\mathbb{X}$ is a cross-section and 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 $\mathbf{N}_0(\mathbf{r})$ describes 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

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

where $\mathbf{N}(\mathbf{r},t)$ describes the nuclide field in fuel and static construction materials (i.e., $N_i(\mathbf{r},t)$ is the concentration of nuclide i at $\mathbf{r}$ at time t), $\mathbf{N}_c(\mathbf{r},t)$ describes the control nuclides (mainly the control rods and coolant), $L(\mathbf{N},\mathbf{N}_c)\phi(\mathbf{s})$ represents the migration and loss of neutrons from $\mathbf{s}$, and $F(\mathbf{N},\mathbf{N}_c)\phi(\mathbf{s})$ accounts for neutron production in $\mathbf{s}$ due to fission. In the following text, the fundamental mode flux that satisfies Eq. (3) with the operator B is denoted as the "fundamental mode of B".

The predictor–corrector schemes may be of various forms. For instance, the SERPENT 1.1.18 code (OECD/NEA Data Bank Computer Program Services) applies the corrector step on the neutron flux, as described by Algorithm 1. The BGCore code (Fridman et al., 2008) and the MCODE code (Xu, 2003) use a predictor–corrector scheme for the nuclide field, see Algorithm 2; this method is also used by some deterministic burnup calculations (Stamm'ler and Abbate, 1983).

Algorithm 1. PC scheme in SERPENT

1: **input:** $\mathbf{N}_0$
2: **for** $n \leftarrow 0, 1, \ldots$ **do**
3: $\quad\phi_n \leftarrow$ **fundamental mode of** $B(\mathbf{N}_n)$
4: $\quad\mathbf{N}^{(\mathrm{P})}_{n+1} \leftarrow \mathbf{N}_n \exp[\mathbb{M}(\phi_n)\Delta t_n]$
5: $\quad\phi^{(\mathrm{P})}_{n+1} \leftarrow$ **fundamental mode of** $B\left(\mathbf{N}^{(\mathrm{P})}_{n+1}\right)$
6: $\quad\bar{\phi}^{(\mathrm{C})}_n \leftarrow \left(\phi_n + \phi^{(\mathrm{P})}_{n+1}\right)/2$
7: $\quad\mathbf{N}_{n+1} \leftarrow \mathbf{N}_n \exp\left[\mathbb{M}\left(\bar{\phi}^{(\mathrm{C})}_n\right)\Delta t_n\right]$
8: **end for**

Algorithm 2. PC scheme in BGCore

1: **input:** $\mathbf{N}_0$
2: **for** $n \leftarrow 0, 1, \ldots$ **do**
3: $\quad\phi_n \leftarrow$ **fundamental mode of** $B(\mathbf{N}_n)$
4: $\quad\mathbf{N}^{(\mathrm{P})}_{n+1} \leftarrow \mathbf{N}_n \exp[\mathbb{M}(\phi_n)\Delta t_n]$
5: $\quad\phi^{(\mathrm{P})}_{n+1} \leftarrow$ **fundamental mode of** $B\left(\mathbf{N}^{(\mathrm{P})}_{n+1}\right)$
6: $\quad\mathbf{N}^{(\mathrm{C})}_{n+1} \leftarrow \mathbf{N}_n \exp\left[\mathbb{M}\left(\phi^{(\mathrm{P})}_{n+1}\right)\Delta t_n\right]$
7: $\quad\mathbf{N}_{n+1} \leftarrow \left(\mathbf{N}^{(\mathrm{P})}_{n+1} + \mathbf{N}^{(\mathrm{C})}_{n+1}\right)/2$
8: **end for**

Numerical stability of Algorithms 1 and 2 is tested in Section 2.1 on a fuel rod model with reflective conditions imposed on all boundaries of the model. Algorithm 2 is also tested on a more complex model of a fuel assembly in Section 2.2. Our conclusions are summarised in Section 3.

2. Numerical test calculations

2.1. Fuel cell model

This section analyses the numerical stability of Algorithms 1 and 2. Algorithm 1 is tested by the SERPENT 1.1.18 code (using the ENDFB 7 library), and Algorithm 2 is tested by the BGCore code (using the JEFF3.1 library).

In this test we use a modified version of the fuel cell model defined by Dufek and Hoogenboom (2009). Our 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. The cell is divided into eight equidistant space zones along the fuel rod; the fuel in each zone contains an independently defined material. The codes were instructed to deplete the fuel in each zone independently according to the actual neutron flux in the zone.

Due to no neutron leakage, 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. Also, this model is suitable for studying the numerical stability of coupling schemes since it has a large dominance ratio and a number 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

$$\varepsilon(\phi) = \sum_{z=1}^{8}\left|\frac{\langle\phi\rangle_z}{\langle\phi\rangle} - \frac{1}{8}\right|, \tag{4}$$

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

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

The numerical stability of each scheme was studied 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 criticality calculation simulated a batch of 5000 neutrons in 1000 inactive and 30,000 active cycles. The initial fission source was uniformly distributed in the fuel.

Results of the test of Algorithm 1 are summarised in Table 1 that describes the errors (calculated by Eq. (4)) of the corrected flux $\varepsilon\left(\bar{\phi}^{(\mathrm{C})}_n\right)$ (see line 6 of Algorithm 1) in all time steps of all burnup calculations. The flux $\varepsilon\left(\bar{\phi}^{(\mathrm{C})}_n\right)$ is used in the fuel depletion; thus, its errors represent also errors in the fuel depletion. The results show that only the burnup calculation with the time step of 7 days gave satisfactory results. With larger time steps large errors appeared in the flux (and the fuel depletion). The numerical instabilities are demonstrated also in Fig. 1 that depicts the shapes of the beginning-of-step, predicted and corrected neutron fluxes in the calculation with the time step of 30 days.

Table 1
Relative error $\varepsilon\left(\bar{\phi}_n^{(C)}\right)$ in the test of Algorithm 1 (%).

n	Δt (days)			
	7	14	30	60
0	0.1	0.6	0.7	4.2
1	0.7	0.5	9.8	28.2
2	0.4	0.4	19.3	14.3
3	0.9	1.0	13.3	10.5
4	1.0	2.5	7.9	2.1
5	1.4	3.9	4.2	3.2
6	0.1	6.1	3.1	2.0
7	0.2	9.3	2.1	2.3
8	2.0	8.8	1.3	1.6
9	1.9	6.2	1.0	2.1

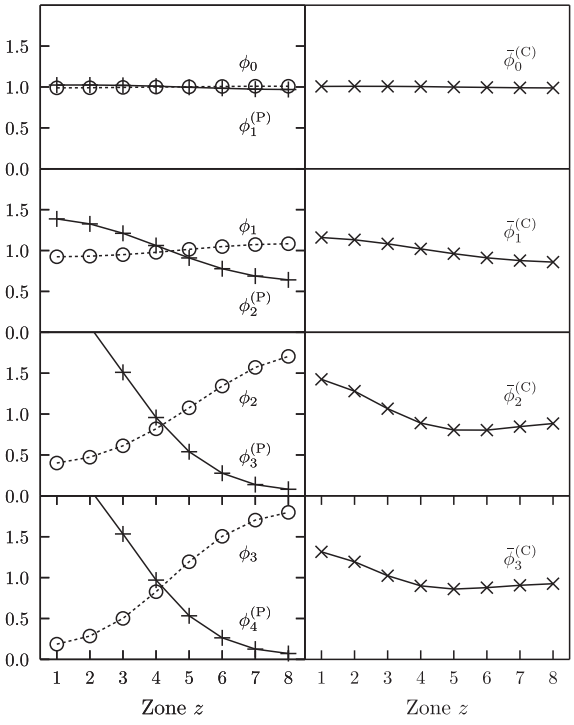

Fig. 1. Neutron fluxes in the test of Algorithm 1 (Δt = 30 day).

Table 2
Relative error $\varepsilon(\bar{\phi}_n)$ in the test of Algorithm 2 (%).

n	Δt (days)			
	7	14	30	60
0	0.6	0.6	1.0	1.5
1	0.1	0.5	7.9	24.5
2	0.3	1.3	19.8	24.2
3	1.1	1.5	15.9	10.0
4	0.2	2.9	7.8	8.0
5	1.1	5.9	6.0	5.1
6	0.2	10.0	4.5	4.2
7	0.7	8.7	2.8	4.2
8	0.8	7.1	3.3	4.2
9	0.3	5.1	3.3	4.3

Results of the test of Algorithm 2 are summarised in Table 2 and depicted also in Fig. 2. Algorithm 2 applies the correction on the nuclide fields, not on the flux; thus, the corrected flux is not used for fuel depletion directly. Nevertheless, both the begging-of-step and predicted fluxes are used in the depletion process equally; therefore, we find it reasonable to summarise the errors of the

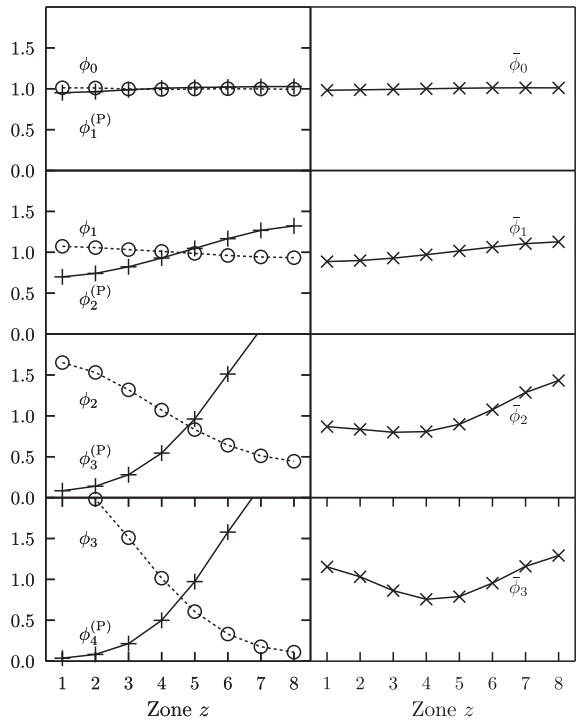

Fig. 2. Neutron fluxes in the test of Algorithm 2 (Δt = 30 day).

average of these fluxes in Table 2 ($\bar{\phi}_n$ in Table 2 and Fig. 2 is computed the same way as $\bar{\phi}_n^{(C)}$ in Algorithm 1). The numerical stability of Algorithms 1 and 2 appears to be similar. Satisfactory results were obtained only with the shortest time step of 7 days here as well. Calculations with larger time steps developed numerical instabilities.

2.2. Fuel assembly model

The analysis was performed with a detailed 3D pin-wise model of a standard Westinghouse PWR 17 × 17 UO2 fuel assembly (Progressive Media Markets Ltd., 2009). The structures of bottom and top reflectors were homogenised. The assembly was axially divided into 18 regions (16 equidistant burnable zones and two reflector zones). The axial distribution of water density was modelled realistically, as depicted on Fig. 3. Reflective radial boundary conditions and void axial boundary conditions were imposed on the model. Due to the axially asymmetrical design and the realistic boundary conditions the dominance ratio of this model must be

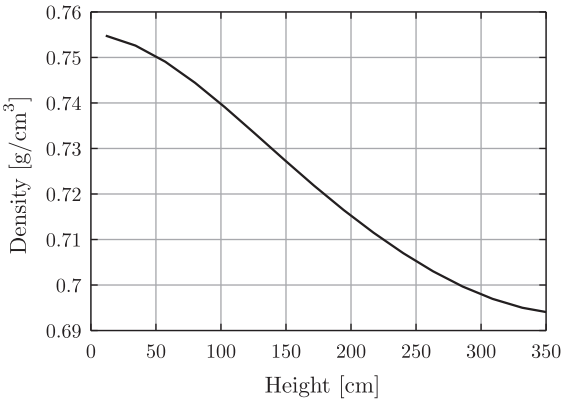

Fig. 3. Water density distribution in the assembly model.

considerably smaller than that of the fuel rod model used in Section 2.1, which must lead to better source convergence during the criticality calculations and overall better numerical stability of the burnup coupling schemes. Yet, to assure the neutron flux was computed with small statistical errors, each criticality calculation simulated 500 inactive and 400 active cycles with 300,000 neutron histories per cycle. The calculations were performed by the BGCore system with the JEFF3.1 cross section library. The assembly was modelled with the following design parameters and operating conditions:

Initial fresh fuel enrichment	3.5 wt.%
Average power density	104 W/cm^3
Assembly active height	366 cm
Top/bottom reflector height	20 cm
No. of fuel rods per assembly	264
No. of guide thimbles per assembly	25
Fuel assembly pitch	21.5 cm
Clad material	Zircalloy-4
Fuel pellet diameter	0.819 cm
Fuel rod diameter	0.950 cm
Pitch to diameter ratio	1.326
Guide tube diameter	1.224 cm
Gap thickness	0.0 cm

The fuel assembly is axially asymmetrical and has realistic boundary conditions; thus, the exact solution is not known at any time. Therefore, we cannot present the errors in the neutron fluxes. Here we only present plots depicting the neutron flux distribution computed at several consequent time steps. In this more realistic case, the neutron flux distribution should indeed change in time as the fuel depletes according to the local flux that is not flat in this case.

When the numerical instability develops in a burnup calculation, it manifests itself by errors in the neutron flux computed at the beginning of time steps. This errors originate in erroneous fuel depletion in previous time steps. While we cannot estimate the errors in the fluxes here we can compare the fluxes computed in subsequent time steps, and search for clear signs of numerical instabilities. Such a sign can be a somewhat chaotic behaviour of the flux over the time steps, or better, existence of the spatial oscillations of the flux, such oscillations that could not be interpreted as standard statistical errors.

Results from Section 2.1 show that the flux does not necessarily has to "oscillate" in numerically unstable calculations. The flux may posses similar and yet wrong distributions over a number of time steps, which makes it very difficult in burnup calculations to notice the instability, especially since the integral quantities, like the k-eigenvalue, may not oscillate. However, the neutron flux does not keep the same wrong distribution over many time steps in unstable calculation; there are certain time steps where it does oscillate. We were searching for such time steps in these test calculations.

First we considered a burnup calculation with the time steps of 25 days; Fig. 4 shows the flux that was calculated at the beginning of several consequent time steps (225, 250, and 275 days). It can be clearly seen that the flux gets through a spatial oscillation over the depicted time steps, revealing a certain level of numerical instability. This numerical instability is indeed less serious than the instability demonstrated in Section 2.1 due to the smaller dominance ratio; nevertheless, the instability is substantial enough to question the credibility of such a burnup calculation.

In the next burnup calculation we decreased the time step to only 5 days; Fig. 5 shows the flux that was calculated at the beginning of several consequent time steps (205, 210, and 215 days). A very similar oscillation of the neutron flux as in the previous burnup calculation can be observed here as well. The amplitude of this oscillation can again exclude the possibility that the errors are only statistical. We would like to stress that the xenon concentration is practically saturated after five days, so the observed oscillation cannot be simply a description of a real xenon oscillation.

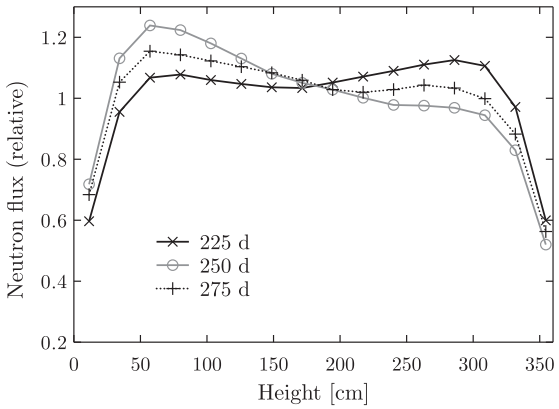

Fig. 4. Beginning-of-step neutron flux distribution in burnup calculation with Δt = 25 day.

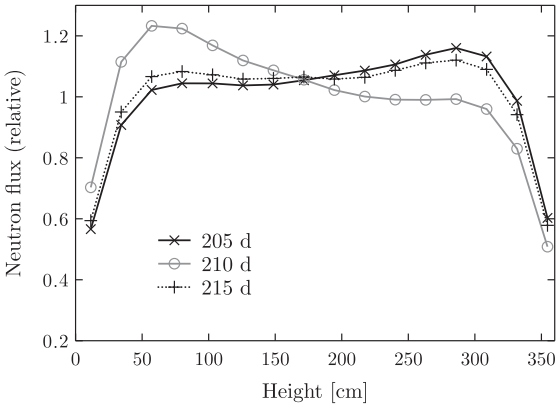

Fig. 5. Beginning-of-step neutron flux distribution in burnup calculation with Δt = 5 day.

We wish to add that Monte Carlo burnup calculations may produce various results for the same input file when the initial seed in the random number generator is not fixed. The errors presented in this section thus do not represent the expected errors as these were sampled by one set of calculations.

3. Discussion

This paper completes the previous study of the numerical stability of simple coupling schemes (the beginning-of-step and middle-of-step flux approximation) used in Monte Carlo burnup codes (Dufek and Hoogenboom, 2009). Here we have studied numerical stability of predictor–corrector coupling schemes that are also implemented in a number of established Monte Carlo burnup codes. For the predictor–corrector methods are known to be conditionally stable, i.e., stable for sufficiently small time steps, it was a question whether the time steps commonly used in Monte Carlo burnup calculations are short enough to consider the calculations stable.

Due to the large computational cost, Monte Carlo burnup calculations are often run with time steps of many weeks or months. No

obvious numerical instabilities are usually noticed since code users rarely inspect the neutron flux distribution over the time steps, and integral quantities, like the k-eigenvalue, may appear stable in unstable calculations.

Results of our test calculation show that the predictor–corrector methods are not suitable for Monte Carlo burnup calculation in general either, and a new stable method is needed. In our test burnup calculations of a fuel rod model (with reflecting boundary conditions) serious numerical instabilities could be observed for the time steps of two or more weeks. Our burnup calculations of a realistic fuel assembly model developed the numerical instability with time step of five days. These time steps are much shorter than those commonly specified in Monte Carlo burnup calculations. Therefore we would like the code users to pay attention to this problem, and inspect the neutron flux (or power) distributions in all steps of their burnup calculation (when a conditionally stable coupling scheme was used in the burnup calculation).

We would like to stress that the numerical stability of the conditionally stable methods that are implemented in established Monte Carlo burnup codes depends generally on many factors, such as the dominance ratio of the system, the number of materials being depleted in the system, and the neutron energy spectrum. Indeed the predictor–corrector method may give good results for systems with a small dominance ratio or systems with only one burnable material or for fast systems.

Acknowledgements

This work was funded by the Swedish Radiation Safety Authority (SSM) and the European Commission under the High Performance Monte Carlo Reactor Core Analysis (HPMC) Project, within the 7th EU Framework Program, Project Number 295971.

References

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

Dufek, J., Hoogenboom, J.E., 2009. Numerical stability of existing Monte Carlo burnup codes in cycle calculations of critical reactors. Nucl. Sci. Eng. 162, 307–311.

Fridman, E., Shwageraus, E., Galperin, A., 2008. Efficient generation of one-group cross sections for coupled Monte Carlo depletion calculations. Nucl. Sci. Eng. 159, 37–47.

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

Progressive Media Markets Ltd., 2009. Fuel Design Data. Nuclear Engineering International, pp. 26–35.

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

Xu, Z., 2003. Design Strategies for Optimizing High Burnup Fuel in Pressurized Water Reactors. Ph.D. Thesis. MIT.