NUCL SCI TECH (2016) 27:65
DOI 10.1007/s41365-016-0070-1

# Development of a MCNP5 and ORIGEN2 based burnup code for molten salt reactor

**Guo-Min Sun**[1,2] · **Mao-Song Cheng**[1]

Received: 29 April 2014 / Revised: 14 June 2015 / Accepted: 14 June 2015 / Published online: 11 May 2016


**Abstract** The Molten Salt Reactor (MSR) is one of the six advanced reactor nuclear energy systems for further research and development selected by Generation IV International Forum (GIF), which is distinguished by its core in which the fuel is dissolved in molten fluoride salt. Because fuel flow in the primary loop, the depletion of MSR is different from that of solid-fuel reactors. In this paper, an MCNP5 and ORIGEN2 Coupled Burnup (MOCBurn) code for MSR is developed under the MATLAB platform. Some new methods and novel arrangements are used to make it suitable for fuel flow in the MSR. To consider the fuel convection and diffusion in the primary loop of MSR, fuel mixing calculation is carried out after each burnup time step. Modeling function for geometry with repeat structures is implicated for reactor analysis with complex structures. Calculation for a high-burnup reactor pin cell benchmark is performed using the MOCBurn code. Results of depletion study show that the MOCBurn code is suitable for the traditional solid-fuel reactors. A preliminary study of the fuel mixture effect in MSR is also carried out.

**Keywords** Molten salt reactor · Burnup · MOCBurn · MCNP · ORIGEN2

This work was supported by the "Strategic Priority Research Program" of the Chinese Academy of Science (No. XDA02010000).

✉ Guo-Min Sun
sunguomin@sinap.ac.cn

[1] Shanghai Institute of Applied Physics, Chinese Academy of Sciences, Shanghai 201800, China

[2] University of Chinese Academy of Sciences, Beijing 100049, China

## 1 Introduction

As a cooperative international organization, the GIF aims to carry out research and development on technologies for nuclear energy systems of the fourth generation, which will have been available for industrial deployment by 2030. In 2002, the GIF selected six nuclear reactor technologies being clean, safe and cost-effective, so as to meet increasing energy demands on a sustainable basis, while being resistant to diversion of materials for weapons proliferation and secure from terrorist attacks. Molten salt reactor [1–3] is among the six G4 reactor designs.

A molten salt reactor (MSR), in which the nuclear fuel is dissolved in the coolant of molten fluoride salt, is operated at near atmospheric pressures, hence the reduced mechanical stress is endured by the system. This simplifies aspects of reactor design with improved safety [4]. The MSR technology was developed first in 1950s in the USA [5]. The molten salt reactor experiment (MSRE) was designed and operated [6], and the molten salt breeder reactor (MSBR) [7] was finished at Oak Ridge in 1960s. There is currently a regained interest in molten salt reactor technology, due to its advantages of inherent safety, decreased radioactive waste, effective utilization of nuclear resources, proliferation resistance and the attractive concept of thorium fuel cycle.

The burnup analysis in a nuclear reactor is very important in reactor analysis and design. Several methods and codes can be used to analyze the burnup of solid-fuel reactors. Most of the codes are based on MCNP [8] and ORIGEN [9], such as Monteburns [10], MOCUP [11], MCBurn [12], COUPLER [13], MCCOOR [14] and MCORGS [15]. But the nuclear fuel circulation in an MSR makes the fuel depletion be so different from those of

solid-fuel reactors that the existing methods and codes are not suitable for burnup analysis of a molten salt reactor.

The thorium-based molten salt reactor (TMSR) nuclear energy system is one of the "Strategic Priority Research Program" of the Chinese Academy of Sciences (CAS) [16]. The burnup analysis is an important part in the R&D of TMSR [17, 18]. To simulate the depletion in MSR, a new code MOCBurn based on MCNP5 and ORIGEN2 is developed under the MATLAB [19] platform. The MCNP, a Monte Carlo code of neutron transportation calculation developed by LANL, is used for evaluation of the neutron flux and reaction rates in all the cells in a nuclear reactor. The ORIGEN2, an isotope generation and depletion code developed by ORNL, calculates the burnup in a given neutron flux. The MOCBurn simulates the flow and mixing of fuel salt flow in an MSR.

## 2 MOCBurn code system

### 2.1 Overview of MOCBurn

The MOCBurn code was developed using the Object Oriented Programming (OOP) [20, 21] technology under the MATLAB platform based on MCNP5 and ORIGEN2. In running the MOCBurn, the one-group cross section of selected nuclides and neutron flux of burnup cells are transferred from MCNP5 to ORIGEN2, and the resulting material inventory after irradiation and/or decay is transferred from ORIGEN2 back to MCNP5.

The flow diagram of MOCBurn is showed in Fig. 1. The burnup predictor–corrector algorithm [10, 12] is implemented in this code system to improve the precision, as it is implemented in Monteburns, MCBurn and some other codes. In an MSR, the fuel mixture is carried out after every burnup time step or decay time step. First, initialize the code system with the user input file. Next, for a first run case, the MCNP5 is run to get the initial $K_{\text{eff}}$, neutron flux distribution and the problem-dependent neutron-spectrum-weighted cross section. For a continue-run case, the code system is initialized with the output file from last run. Then, the code system gets into the burnup steps run until all the steps are finished. In each burnup step, the predictor–corrector algorithm is implemented first to obtain $K_{\text{eff}}$, neutron flux and the problem-dependent neutron-spectrum-weighted cross section for the current time step, and the point burnup calculation of every burnup cell is then carried out using ORIGEN2 code. After that, the MCNP5 calculation is performed to obtain parameters for the next time step calculation. The detailed calculation procedure for molten salt reactor is described in the Fig. 2.

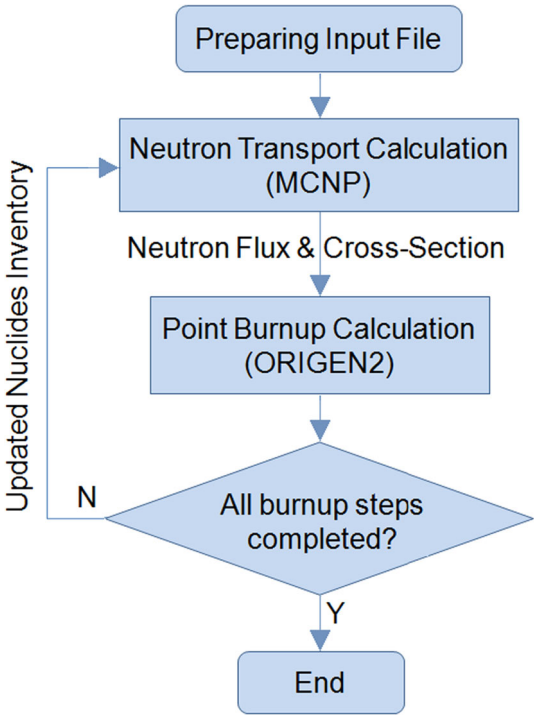


**Fig. 1** Flowchart of the MOCBurn code system

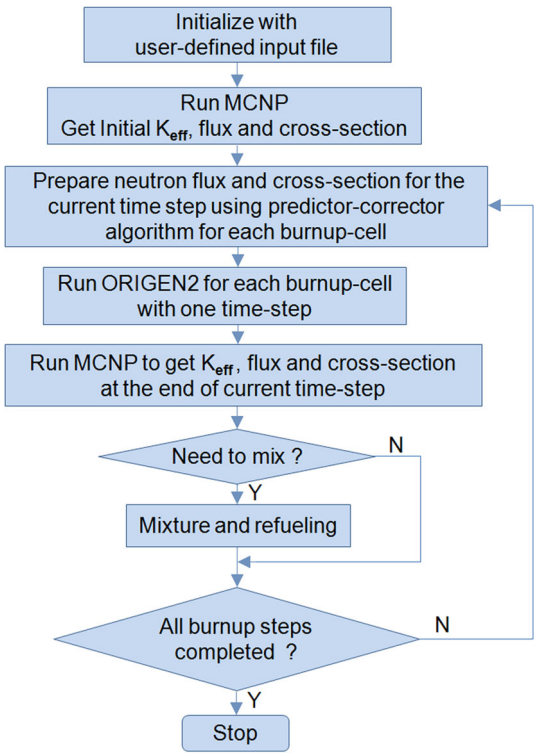


**Fig. 2** Calculation procedure for a molten salt reactor case

### 2.2 Characteristics of MOCBurn

The MOCBurn code is designed with new characteristics and arrangements for MSR analysis, specifically.

(1) Treatment of the fuel mixture due to the molten salt motion

In an MSR, the fuel salt moves all the time through the primary loop, and fuel mixing occurs in the convection and diffusion of molten salt. This is the most critical difference between an MSR and a solid-fuel reactor. The fuel mixing needs additional calculation and consideration in a burnup analysis code. The MOCBurn code calculates the fuel mixing after each burnup time step, if necessary, during the MSR analysis.

(2) Treatment of repeat structures

Repeat structures are widely used in modern nuclear reactors. The MOCBurn code can model repeat structures of complex geometry.

(3) Simulation of refueling and processing online

In an MSR, the fuel can be added online, and processed online or offline. The MOCBurn code can simulate the online and offline fuel adding and processing.

(4) Easy to modify and update

The MOCBurn code under OOP technology is easy to modify or update. In the MOCBurn, two abstract classes, *NeutronTransport* and *PointBurnup*, are defined. The class *Mcnp* is inherited from class *NeutronTransport*, packaging the Monte Carlo neutron transport code MCNP. The class *Origen* is inherited from class *PointBurnup*, packaging the ORIGEN2 code. The code system can be modified or updated by just replacing the class *Mcnp* and class *Origen*. One can alternate the class *Mcnp* by other class inherited from class *NeutronTransport* packaged other neutron transport or neutron diffusion code. Also, one can alternate the class *Origen* by other class inherited from class *PointBurnup* packaged other burnup code, such as multi-group burnup code.

(5) Easy to be cross platform

The MOCBurn code can run on either Windows or Linux operating systems.

## 3 Validation of MOCBurn

In a solid-fuel reactor, a fuel assembly is loaded into or removed from the core as a whole unit. So, the burnup analysis for a solid-fuel assembly ends after a whole refueling cycle when it reached the designed burnup level. In this regards, in online or batch reprocessing procedure of a MSR in operation, part of the molten salt is removed from the primary loop, and most nuclides, enduring long time neutron exposure, evolve in similar ways to those in a solid-fuel reactor of high burnup. So, a high-burnup PWR $UO_2$ lattice benchmark is used to verify the MOCBurn.

### 3.1 Description of the high-burnup PWR $UO_2$ lattice benchmark

This benchmark is a high-burnup $UO_2$ lattice problem [22], which employed a single pin cell model of a standard Westinghouse $17 \times 17$ PWR assembly. The geometry model is shown in Fig. 3, the detailed pin cell model parameters are listed in Table 1 and the initial material compositions are given in Table 2. Reflecting boundary is used for the unit cell. This model represents cross section of the fuel pin while axial dependence is ignored, so the top and bottom boundary conditions are set to reflecting boundaries. The ENDF/B-VII-evaluated nuclear data libraries are used to generate ACE-formatted libraries for MCNP. The temperature is 300 K, and the fuel is uranium dioxide with 9.75 w/o $^{235}U$ enrichment which can burn to a high burnup (about 100 MWd/kg).

### 3.2 Results and analysis

The CASMO-4 [23] results from the benchmark are used as references. The CASMO-4 uses neutron cross

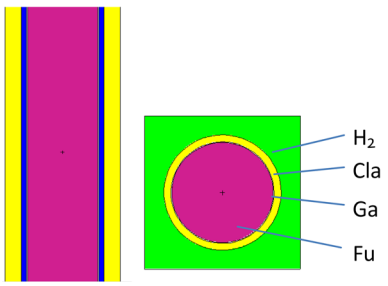


**Fig. 3** Pin cell model

**Table 1** Pin cell model parameters (at 300 K)

| Parameters | Values |
|---|---|
| Fuel pellet radius (cm) | 0.4096 |
| Cladding inner radius (cm) | 0.4178 |
| Cladding outer radius (cm) | 0.4750 |
| Pin pitch (cm) | 1.26 |
| Fuel density (g/cm$^3$) | 10.3 |
| Cladding density (g/cm$^3$) | 6.550 |
| Coolant density (g/cm$^3$) | 0.997 |
| Power density (kW/liter core) | 104.5 |
| Specific power (W/gU) | 34.6679 |

**Table 2** Initial compositions (at 300 K)

| Components | Nuclides | Weight percent (w/o) | Number density (1/cm$^3$) |
|---|---|---|---|
| Fuel | $^{234}$U | 0.0688 | 1.82239E+19 |
| (9.75 w/o $UO_2$) | $^{235}$U | 8.5946 | 2.26826E+21 |
| | $^{238}$U | 79.4866 | 2.07128E+22 |
| | $^{16}$O | 11.8500 | 4.59686E+22 |
| Cladding | O | 0.125 | 3.08281E+20 |
| (Zircaloy-4) | Cr | 0.10 | 7.58663E+19 |
| | Fe | 0.21 | 1.48338E+20 |
| | Zr | 98.115 | 4.24275E+22 |
| | Sn | 1.45 | 4.81835E+20 |
| Coolant | $^{1}$H | 11.19 | 6.66295E+22 |
| ($H_2O$) | $^{16}$O | 88.81 | 3.33339E+22 |

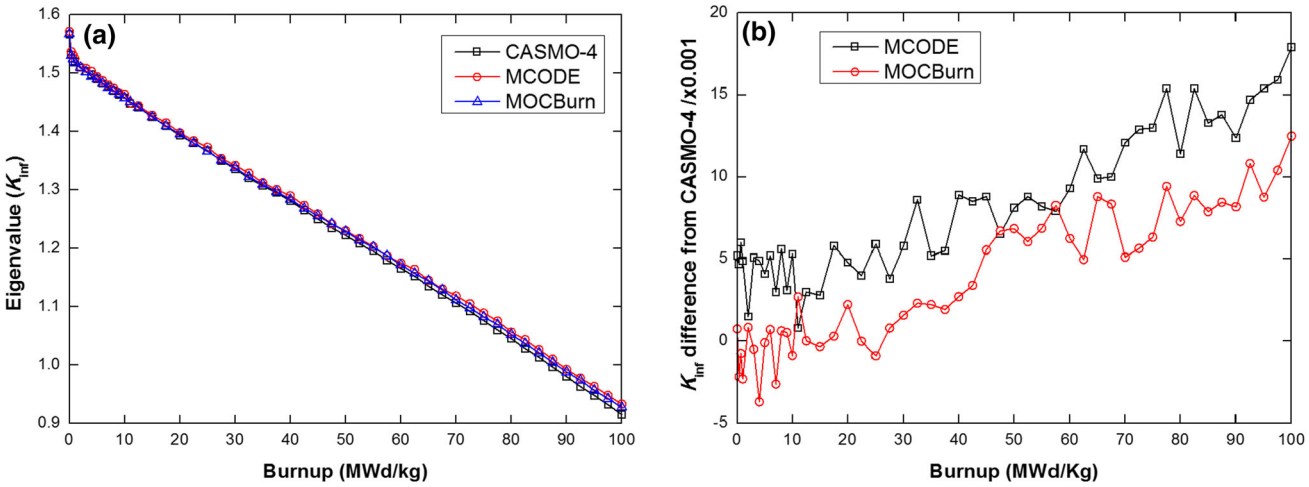


**Fig. 4** Eigenvalues by CASMO-4, MCODE and MOCBurn (**a**), and the differences from CASMO-4 (**b**), as a function of burnup

section libraries processed from JEF-2.2 and ENDF/B-VI, the MCODE uses the libraries based on ENDF/B-VI library data and our MOCBurn code system uses the cross section libraries processed from the ENDF/B-VII library data.

In Fig. 4a, the eigenvalue histories are compared among MOCBurn, CASMO-4 and MCODE, as a function of burnup, and the eigenvalue differences of MCODE and MOCBurn from CASMO-4 are shown in Fig. 4b. It can be seen that MOCBurn–CASMO-4 difference is less than the MCODE–CASMO-4 difference. Particularly, at the high burnup of >60 MWd/kg, the MOCBurn is apparently better than MCODE in agreeing with CASMO-4. This means the MOCBurn code is more suitable for the high-burnup problems than the MCODE.

In Table 3, the nuclide compositions at 100 MWd/kg obtained by CASMO-4 are given, and the MCODE and MOCBurn results are compared in fractional differences of nuclide composition from those by CASMO-4. It can be seen that, the fractional difference of all actinides are less than 10 % except $^{243}$Am. For the fission products, all the fractional differences are less than 30 %, and most of them are less than 10 % except the Sm isotopes, $^{109}$Ag and $^{153}$Eu. It can be found that there is more $^{235}$U consumed and more $^{239}$Pu produced. At the last steps of burnup calculation, there is less $^{235}$U, which means less fission product of Sm, and more $^{239}$Pu, which means more Am. Considering the high-burnup condition, the differences are mostly acceptable. It shows that the MOCBurn's material composition predictions are acceptable in high-burnup problems.

The benchmarking results of this high-burnup $UO_2$ lattice problem indicate that MOCBurn is suitable for the burnup analysis, even for high-burnup cases.

**Table 3** Fractional difference from CASMO-4 in nuclide concentration at 100 MWd/kg

| Isotopes | CASMO-4 (at/cm$^3$) | MCODE (%) | MOCBurn (%) |
|---|---|---|---|
| $^{95}$Mo | 1.22281E+20 | −0.17 | −0.51 |
| $^{99}$Tc | 1.16862E+20 | 4.54 | 1.60 |
| $^{101}$Ru | 1.19274E+20 | −0.30 | −1.79 |
| $^{103}$Rh | 4.60151E+19 | 3.30 | 2.64 |
| $^{109}$Ag | 6.99101E+18 | 14.62 | 14.10 |
| $^{133}$Cs | 1.14516E+20 | 8.15 | 6.02 |
| $^{135}$Cs | 6.98202E+19 | 0.35 | 0.64 |
| $^{143}$Nd | 7.42463E+19 | 0.34 | −0.86 |
| $^{145}$Nd | 7.10908E+19 | −0.08 | −2.49 |
| $^{147}$Sm | 9.57151E+18 | 14.09 | 11.82 |
| $^{149}$Sm | 1.24554E+17 | −5.83 | −29.40 |
| $^{150}$Sm | 2.67571E+19 | 8.08 | −12.96 |
| $^{151}$Sm | 7.68167E+17 | −10.27 | −25.70 |
| $^{152}$Sm | 9.39450E+18 | −16.19 | −21.13 |
| $^{153}$Eu | 1.18378E+19 | −11.36 | −12.09 |
| $^{234}$U | 6.71252E+18 | 1.20 | 4.97 |
| $^{235}$U- | 2.59522E+20 | 2.10 | −2.21 |
| $^{238}$U | 1.96718E+22 | −0.17 | 0.24 |
| $^{237}$Np | 3.42341E+19 | −8.90 | −2.52 |
| $^{238}$Pu | 1.96654E+19 | −8.26 | −3.27 |
| $^{239}$Pu | 1.47667E+20 | 5.61 | 4.41 |
| $^{240}$Pu | 6.31059E+19 | 8.77 | 7.56 |
| $^{241}$Pu | 4.28013E+19 | 5.13 | 3.84 |
| $^{242}$Pu | 2.62275E+19 | −3.05 | 1.98 |
| $^{241}$Am | 2.35052E+18 | 9.96 | 7.61 |
| $^{243}$Am | 6.23202E+18 | 23.22 | 28.45 |

## 4 Preliminary use in MSR calculation

In operation of an MSR, the fuel flows and diffuses through the primary loop. But the online refueling and the online or batch reprocessing are not carried out at any case. So, the MSR burnup analysis shall be done on the fuel mixing only. As the preliminary utilization for molten salt reactor calculation of MOCBurn, a hexagon molten salt channel cell was used for case study of the fuel mixing.

### 4.1 Molten salt channel cell case description

This case employs a molten salt channel cell model moderated by hexagon graphite. The geometry model is shown in Fig. 5, the model parameters are listed in Table 4 and initial compositions of molten salt are given in Table 5. Reflecting boundary condition is used for the cell. The ENDF/B-VII-evaluated nuclear data libraries are used to generate ACE-formatted libraries for MCNP. The temperature is 900 K, the fuel is LiF–$BeF_2$–$ZrF_4$–$UF_4$ and the given radiation power is 1 MW.

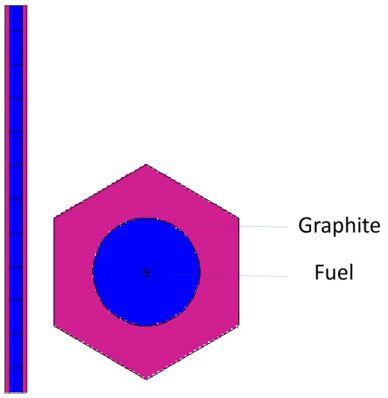


**Fig. 5** Molten salt channel cell model

**Table 4** Molten salt channel cell model parameters (at 900 K)

| Parameters<br>Values | |
|---|---|
| Fuel channel radius (cm) | 7.5 |
| Pin pitch (cm) | 15 |
| Fuel density (g/cm3) | 3.3 |
| Graphite density (g/cm3) | 2.3 |
| Power (MW) | 1 |

This study compared the eigenvalue history and the heavy metal mass separately for the fuel mixing case, which resembles the process in an MSR, and for the non-fuel mixing case, i.e., the case of a solid-fuel reactor.

### 4.2 Results and analysis

Figure 6 shows comparisons of eigenvalue history and heavy metal mass history, between considering and non-considering the fuel mixing. It can be seen that fuel mixing increases the fuel efficiency, leading to the growth of eigenvalue at the first several steps. At the same time, the fuel mixing increases the depletion rate of heavy metals, which decreases the rate of eigenvalue.

**Table 5** Initial molten salt compositions (at 900 K)

| Nuclides | $^{6}$Li | $^{7}$Li | $^{9}$Be | $^{19}$F | $^{90}$Zr | $^{91}$Zr | $^{92}$Zr |
|---|---|---|---|---|---|---|---|
| Atom percent (w/o) | 0.00668 | 66.79332 | 29 | 141.6 | 2.058 | 0.4488 | 0.6860 |

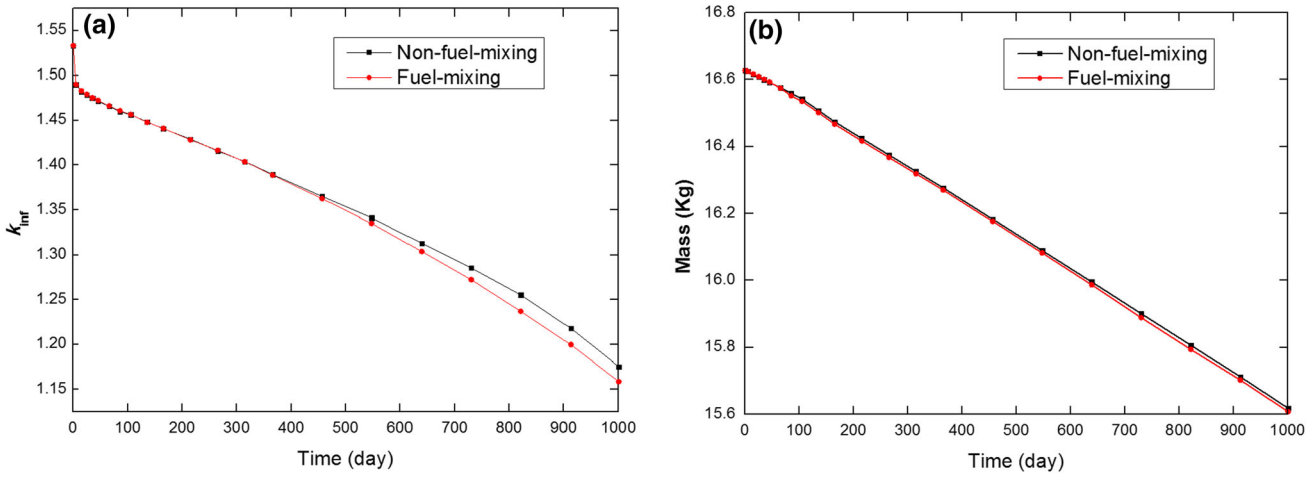


**Fig. 6** Comparisons of the eigenvalue (**a**) and heavy metal mass (**b**), between fuel mixing and non-fuel mixing

## 5 Conclusion

An MCNP- and ORIGEN2-based burnup analysis code system, MOCBurn, has been developed for molten salt reactors. It considers the flow and mixing of molten salt in the primary loop. Based on the benchmark results, it can be concluded that MOCBurn is suitable for the burnup analysis for high-burnup solid-fueled nuclear reactors. The preliminary study of MSR burnup analysis shows the effect of fuel mixing on the eigenvalue and heavy metal mass. More burnup studies on MSR will be carried out using the MOCBurn code system.

## References


1. K. Furukawa, K. Arakawa, L.B. Erbay et al., A road map for the realization of global-scale thorium breeding fuel cycle by single molten-fluoride flow. Energy Convers. Manag. **49**, 1832–1848 (2008). doi:10.1016/j.enconman.2007.09.027
2. D. Heuer, E. Merle-Lucotte, M. Allibert et al., Towards the thorium fuel cycle with molten salt fast reactors. Ann. Nucl. Energy **64**, 421–429 (2014). doi:10.1016/j.anucene.2013.08.002
3. J. Serp, M. Allibert, O. Benes et al., The molten salt reactor (MSR) in generation IV: Overview and perspectives. Prog. Nucl. Energy **77**, 308–319 (2014). doi:10.1016/j.pnucene.2014.02.014
4. B.M. Elsheikh, Safety assessment of molten salt reactors in comparison with light water reactors. J. Radiat. Res. Appl. Sci. **6**, 63–70 (2013). doi:10.1016/j.jrras.2013.10.008
5. W.B. Cottrell (ed.), *Reactor Program of the Aircraft Nuclear Propulsion Project* (Oak Ridge National Laboratory, Oak Ridge, 1952 June). Report No.: ORNL-1234
6. R.C. Robertson, *MSRE Design and Operation Report I: Description of Reactor Design* (Oak Ridge National Laboratory, Oak Ridge, 1965 Jan). Report No.: ORNL-TM-728
7. R.B. Briggs, *Summary of the Objectives, the Design, and a Program of Development of Molten Salt Breeder Reactors* (Oak Ridge National Laboratory, Oak Ridge, 1967 June). Report No.: ORNL-TM-1851
8. Los Alamos National Laboratory, *MCNP-A General Monte Carlo N-Particle Transport Code, Version 5*. LA-UR-03-1987 (2003)
9. Oak Ridge National Laboratory, *A User's Manual for ORIGEN2 Computer Code*. ORNL/TM-7175 (1980)
10. Los Alamos National Laboratory, *Development of a fully-automated Monte Carlo burnup code MONTEBURNS*. LA-UR-99-42 (1999)
11. Idaho National Engineering Laboratory, *MOCUP: MCNP-ORIGEN2 Coupled Utility Program*. INEL-95/0523 (1995)
12. G.L. Yu, K. Wang, Y.H. Wang, MCBurn-A coupling package of MCNP and ORIGEN codes. At. Energy Sci. Technol. **37**, 240–254 (2003). **(in Chinese)**
13. H.Q. Li, Y.W. Yang, Development of code system burnup calculation for reactor, in *The 10th Conference on Reactor Numerical Calculation and Particle Transportation* (Haiyan, Zhejiang, 1 Oct 2004) **(in Chinese)**
14. J.H. Li, S.B. Zhang, E.F. Kryuchkov et al., MCCOOR code system for burnup calculation. Nucl. Power Eng. **27**, 16–19 (2006). **(in Chinese)**
15. X.M. Shi, B.A. Zhang, Development of transport-burnup code MCORGS. Nucl. Power Eng. **31**, 1–4 (2010). **(in Chinese)**
16. M.H. Jiang, H.J. Xu, Z.M. Dai, Advanced fission energy program-TMSR nuclear energy system. Bull. Chin. Acad. Sci. **27**, 366–374 (2012). **(in Chinese)**
17. W.X. Li, Q.N. Li, An simplified scheme for the TMSR fuel cycle, in *The 11th National Workshop on Nuclear Chemistry and Radiochemistry* (Chengdu, China, 2012)
18. Q.N. Li, L. Zhang, W. Zhou, et al., Progress of the fuel cycle technology in the Thorium-based Molten-Salt Reactor nuclear

energy system, in *The 13th National Workshop on Nuclear Chemistry and Radiochemistry* (Dali, China, 2014)
19. S. Attaway, *MATLAB: A Practical Introduction to Programming and Problem Solving* (Butterworth-Heinemann, Oxford, 2013), pp. 4–28
20. C.S. Horstmann, *Object-Oriented Design and Patterns* (Wiley, MA, 2006)
21. E. Balagurusamy, *Object Oriented Programming With C++*, 4th edn. (McGraw-Hill, New York, 2008)
22. Z.W. Xu, P. Hejzlar, M.J. Driscoll, et al., An improved MCNP-ORIGEN depletion program (MCODE) and its verification for high-burnup applications, in *PHYSOR 2002* (Seoul, Korea, 2002)
23. J. Rhodes, M. Edenius, CASMO-4. *A Fuel Assembly Burnup Program. User's Manual*. Studsvik of Scandpower, ssp-09/443-u rev 0 (2009)