A SIMPLIFIED BURNUP CALCULATION STRATEGY WITH REFUELING IN STATIC MOLTEN SALT REACTOR

A. K. SRIVASTAVA, ANURAG GUPTA, P. D. KRISHNANI

Reactor Physics Design Division,
Bhabha Atomic Research Centre, Trombay, Mumbai–400085, India
Email: ashishks@barc.gov.in

ABSTRACT

Molten Salt Reactors, by nature can be refueled and reprocessed online. Thus, a simulation methodology has to be developed which can consider online refueling and reprocessing aspect of the reactor. To cater such needs a simplified burnup calculation strategy to account for refueling and removal of molten salt fuel at any desired burnup has been identified in static molten salt reactor in batch mode as a first step of way forward. The features of in-house code ITRAN has been explored for such calculations. The effect of refueling fresh fuel and removal of burned fuel has been studied in batch mode with in-house code ITRAN. Also, the effect of refueling and burnup on change in reactivity per day has been analyzed. Important results are presented.

INTRODUCTION

Molten Salt Reactors (MSRs) are among the nuclear power systems, which suits the need of India’s 3rd Stage nuclear power program for effective utilization of thorium [1, 2]. Due to inherent online refueling and reprocessing features in molten salt reactor, the modeling of such reactors becomes complicated. Thus, there is need to develop tools and methodology, which can take into account such features on online basis. In a way forward, an effort has been made to explore available in-house code ITRAN [3] for simplified problems of such nature in static molten salt reactor with an aim to automatize the code at lattice level to perform such calculations and explore the possibility of extending such calculations at core level in future.

Detailed investigations worldwide have shown that fluoride based molten salts fuel are promising fuel candidate in molten salt reactor. Therefore, molten salt fuel, i.e. LiF-ThF4-UF4, has been analyzed for it’s burnup behavior with refueling a fixed amount of fresh fuel and removing same amount of burned fuel at any desired burnup in batch mode. Since, the code ITRAN does not have the feature of online refueling at present, it was decided to keep $K_{eff}$ slightly greater than 1 (1.005 mk) and burnup calculation was performed. In the present analysis, refueling is done by removing certain amount of burned fuel and adding same amount of fresh fuel at the moment reactor is about to become subcritical. Total number of nuclides considered in the study is 96. Total amount of fuel in core and external circuit is 12600 liters. The basic property of molten reactor is shown in table 1.

<table>
<thead>
<tr>
<th>Brief Description of core</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Thermal / Electric power</td>
<td>1900 MW_th/ 850 MW_e</td>
</tr>
<tr>
<td>Fuel molten salt - initial composition (mol%)</td>
<td>LiF_4 (77.6%) – $^{232}$ThF_4 (19.7%) – $^{233}$UF_4 (2.7%)</td>
</tr>
<tr>
<td>Fuel salt density at 750 °C (g/cm³)</td>
<td>4.3</td>
</tr>
<tr>
<td>Core dimensions (cm)</td>
<td>Radius: 1.0, Height: 2.0</td>
</tr>
<tr>
<td>Fuel Salt Volume (core + external circuit)</td>
<td>12600 liters</td>
</tr>
</tbody>
</table>

Table 1: Basic property of fuel and core under investigation
METHOD AND RESULTS

The simulations have been carried out using in-house code ITRAN. ITRAN performs lattice analysis using first flight collision probability / interface current methods as function of burnup, whereas burnup calculation is done by solving Bateman equation. The effect of refueling of fresh fuel and removal of burned fuel has been taken into account by appropriately adjusting each nuclide number density in the input file using the following relation.

\[ N_{mix,i} = \frac{N_{fresh,i} \cdot V_{fresh} + N_{burned,i} \cdot (V_{total} - V_{burned})}{V_{total}} \]

where, \( N_{mix,i} ; N_{fresh,i} ; N_{burned,i} \) are number density of mixed, fresh & burned fuel of \( i \)th nuclide. \( V_{fresh} ; V_{burned} ; V_{total} \) are volume of fresh, removed and total fuel. In the analysis, the volume of fresh fuel added to the core is equal to volume of removed burned fuel. The results show that, \( K_{eff} \) falls sharply with burnup initially, which slows down later at higher FPD due to buildup of U\(^{233}\) in situ. The change in reactivity per day decreases from 0.82 mk/day to 0.45 mk/day within a period from 0 FPD to 39 FPD (figure 3). However, the difference of change in reactivity also decreases with burnup. Thus, it is apparent that refueling rate is initially higher for fresh core which decreases as core burns due to production of U\(^{233}\). It is also found that, change in reactivity per day is higher for a fuel comprising higher amount of fresh fuel at any given burnup (figure 2). In figure 1, during the first refueling at 5 FPD, 5000 liters of fuel is replaced with fresh fuel, while in 2\(^{nd} \) refueling at 7 FPD the effect of replacement of different amount of fuel has been shown. It is easily seen from figure 1, that larger the amount of burned fuel replaced with fresh fuel, higher is \( K_{eff} \). The \( K_{eff} \) value has been obtained by supplying geometric buckling in ITRAN input for the analysis.

CONCLUSIONS

A methodology has been identified to take into account the refueling and removal of molten salt fuel in batch mode from in-house code ITRAN. However, in order to perform analysis on online basis, there is need to automatize the code. A higher refueling rate is required in fresh core due to small buildup of U\(^{233}\) initially. The effects of refueling different volumes of fuel for corresponding removal of burned fuel has been studied and results are presented in figure 1. The change in reactivity per day has been found to decrease with burnup due to in situ formation of U\(^{233}\). Also, for fuel replaced with larger volume, the change in reactivity per day has been found to be higher.

REFERENCES

1) DAE report, “Shaping the 3\(^{rd}\) stage of indian nuclear power program”, http://dae.nic.in/writereaddata/pdf_32.
3) P.D.Krishnani, “Input preparation of the computer code ITRAN”, RPDD: Note No.: RPDD/RE/11, June 23, 2006