An Initial Study of the Convergence Rate of Griffin's Pebble Bed Reactors Algorithm

T. M. Paganin[1]**, S. Schunert**[2]**, L. R. C. Moraes**[1]**, R. Vasques**[1]

[1]The Ohio State University,
Department of Mechanical and Aerospace Engineering,
201 W. 19th Avenue, Columbus, OH 43210, USA

[2]Idaho National Laboratory,
1955 N. Fremont Ave, Idaho Falls, ID 83415, USA

mallmannpaganin.1@osu.edu, sebastian.schunert@inl.gov,
moraes.14@osu.edu, vasques.4@osu.edu

ABSTRACT

This paper presents an initial study of the convergence properties of an iterative algorithm for computing the burnup distribution in a pebble bed reactor (PBR) in its equilibrium core condition. The algorithm is implemented in the Griffin code. Griffin is a reactor multiphysics analysis application jointly developed by Idaho National Laboratory (INL) and Argonne National Laboratory (ANL). Griffin's PBR algorithm is discussed and simulation data are presented. An alternative matrix formulation of the algorithm is presented that facilitates analysis of the iterative algorithm. The dependence of the spectral radius of the iterative algorithm on operational and discretization parameters is investigated.

KEYWORDS: Pebble Bed Reactors, Depletion, Convergence Rate, Griffin

1. INTRODUCTION

The Pebble Bed Reactor (PBR) is one of the most promising advanced reactors design for near-term deployment [1]. Its basic design features a reactor core composed of a bed of spherical fuel elements (pebbles) consisting of thousands of fuel particles embedded in a graphite matrix. Hundreds of thousands of pebbles are randomly packed in the reactor core, generating a sustained fission chain reaction which is cooled by high-pressure gas forced through the interstitial spaces between the pebbles. The pebbles are in constant motion, getting in and out of the reactor core in a recirculating process.

The pebbles used in PBRs can achieve very high temperatures without damaging their structure. This makes the PBR a very attractive design for high-temperature gas-cooled reactors (HTGRs). The technology readiness level of gas reactor technology is high, making the PB-HTGR a strong candidate for near-term deployment [2]. Moreover, its constant refueling (recirculating) process ensures low operational downtime, increasing the power plant's economy. However, these unique features of PBRs also bring multiple technical challenges, especially concerning safety analyses and optimization of this type of reactor. The PBR core is highly complex and challenging to simulate accurately, due to a large number of pebbles and their constant reshuffling into different burnup zones. This reshuffling significantly increases the number of possible core configurations, power profiles, and depletion history of the pebbles.

The development of new algorithms and computer codes to be used in PBRs simulations is currently under progress by several institutions. Idaho National Laboratory (INL) is implementing PBR capabilities to its reactor multiphysics analysis code (Griffin[3]) within MOOSE (Multiphysics Object-Oriented Simulation Environment [4]) framework. Therefore, there is interest in the optimization and convergence analysis of the computer codes developed by INL.

A summary of the remainder of this paper is given next. In Section 2 we present the Griffin PBR algorithm and its discrete formulation as well as an equivalent matrix formulation. The idea of describing an alternative way (matrix formulation) is that it gives an alternative and clearer way to analyze the convergence rate of Griffin's algorithm. In Section 3 we present numerical results to compare the outcomes generated by Griffin's code and the matrix formulation, as well as to analyze the effect that some design parameters have on the convergence rate of Griffin's codes. Finally, in Section 4, we provide further investigation of the spectral radius observed, illustrating the behavior from a physical and mathematical standpoint. We discuss more tests to expand the range of parameters and also provide an analysis of the properties of the matrix used in the matrix formulation method. We close both the Section and the paper with a short discussion of intended future work.

2. MATHEMATICAL FORMULATION

The main objective of the PBR algorithm developed on Griffin is to calculate the equilibrium state of the PBR core [5,6]. The steps of this algorithm are presented by the schematic in Fig. 1. The code uses a separate solver to calculate the neutron flux and power distribution inside the reactor core based on an estimate for the isotope distribution (outer loop). It then uses these results to calculate the burnup of the equilibrium state of the pebbles (inner loop). The outcome from the inner loop is used to feed the outer loop with new isotope concentration estimations, constituting an iterative process that is executed until it reaches a convergence criterion.

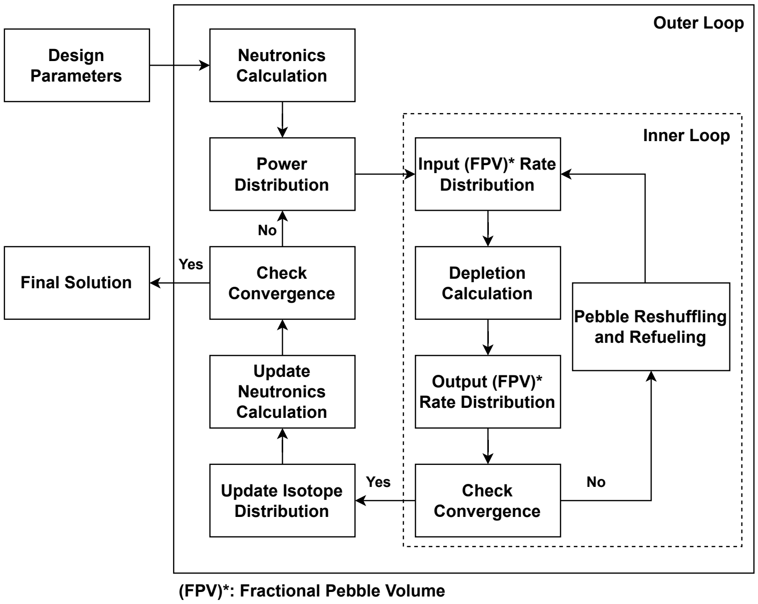
Outer Loop
Design Parameters
Neutronics Calculation
Power Distribution
Inner Loop
Input (FPV)* Rate Distribution
No
Yes
Final Solution
Check Convergence
Depletion Calculation
Pebble Reshuffling and Refueling
Update Neutronics Calculation
Output (FPV)* Rate Distribution
Update Isotope Distribution
Yes
Check Convergence
No
(FPV)*: Fractional Pebble Volume

Figure 1: Griffin's PBR algorithm.

This work focuses on analyzing the inner loop since, its convergence has a higher dependency on design parameters and a major impact on the algorithm performance. There are two main elements inside this loop: the first is the depletion and pebble flow over the reactor, and the second is the inflow and outflow boundary conditions in tandem with the pebble reshuffling and replenishing mechanisms. To present the governing equations for these procedures, we must first introduce two important concepts: the *streamline* and the *fractional pebble volume* (n).

The streamline is the path traced out by pebbles while moving across the reactor [5,6]. It is assumed that once a pebble enters a streamline, it will keep traveling in that same streamline until it exits the reactor. That pebble will not move to adjacent streamlines throughout this pass. Pebbles that enter the reactor close to each other will remain close for the whole pass through the reactor [7]. The reactor can be discretized into several streamlines (k) to account for different paths with different conditions. The advantage of this approximation is the simplification of the spatial problem from multidimensional to one-dimensional.

The second concept is the variable of interest in the inner loop. Pebbles enter the reactor at position ($s = 0$) in streamline (k) and move along (s) through that same streamline. The fractional pebble volume ($n_k(s)$) is the ratio between the volume occupied by pebbles around position (s), and the total volume around that position (s). In other words, it reflects the volume fraction occupied by pebbles of a particular type in a given region. It can also be defined with additional parameters [5,6]:

$$n_{c,k}(s,\tau)d\tau = \frac{\begin{bmatrix}\text{Volume occupied by a pebble of type } c\\ \text{identified by initial composition on streamline } k\\ \text{with burnup between } \tau \text{ and } \tau + d\tau \text{ at position } s\end{bmatrix}}{\text{Total Volume at position } s}, \tag{1}$$

where (c) is the type of pebble and (τ) is the burnup of the pebble. The type of pebble (c) can represent different compositions, such as fuel pebbles, graphite pebbles, or burnable absorbers. The variable ($n_{c,k}(s,\tau)$) is presented in a continuous form; however, it requires a defined finite total volume representation for the denominator.

The ratio of the total pebble volume in the reactor and the total volume of the reactor core is defined as the *packing fraction*. This ratio has also a significant impact on the fractional pebble volume since it determines how much void or coolant volume the reactor has in proportion to pebbles.

Finally, the pebble migration and depletion, presented as the first element of the inner loop, can be described as a balance equation for the fractional pebble volume over a finite differential volume [5], given by

$$\frac{\partial\big(u_k(s)A_k(s)n_{c,k}(s,\tau)\big)}{\partial s} + A_k\frac{\partial\big(p_c(\vec{r_k}(s),\tau)n_{c,k}(s,\tau)\big)}{\partial \tau} = 0. \tag{2}$$

Here, (A_k) and (u_k) are the cross-sectional area and tangential velocity to the streamline (k), respectively. The term p_c is the power density of that pebble at a certain position ($\vec{r_k}(s)$) in the reactor with burnup level between (τ) to ($\tau + d\tau$). For simplicity, only one type of pebble and a single streamline were considered in this work, dropping the (c, k) terms out of the equations.

The second element of Griffin's inner loop consists of the recirculation and replenishment of pebbles. This mechanism is illustrated in Fig. 2. The pebbles enter the reactor through the pebble handling and storage system (PHSS) and are placed in a certain streamline (k). These pebbles migrate through the reactor, depleting a certain amount, and then are collected at the outlet of the reactor. The collected pebbles are tested and the ones that surpassed the burnup limit (τ_{max}) are discarded and replenished by fresh new pebbles at the PHSS. The pebbles that are partially depleted and that were not discarded are mixed with the fresh pebbles and reinserted in the inlet of the reactor in random streamlines. Therefore, the boundary condition at the inlet depends on the boundary condition at the outlet.

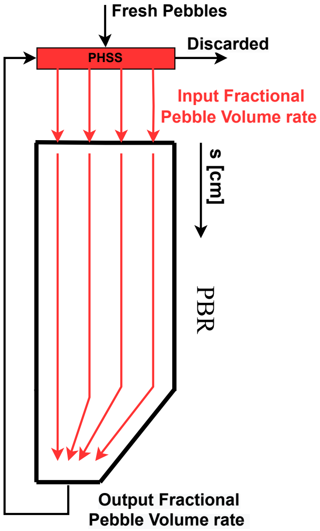
Fresh Pebbles
Discarded
PHSS
Input Fractional Pebble Volume rate
s [cm]
PBR
Output Fractional Pebble Volume rate

Figure 2: Reactor Streamline Schematic.

The inner loop's iterative trait arises from this iteration between the inflow and outflow boundary conditions. The iterative method aims to calculate a converged fractional pebble volume rate distribution at the entry of the reactor ($\dot{n}(0, \tau)$). It calculates the burnup using Eq. (2) and updates the boundary conditions at each iteration step. The boundary conditions used at each iteration step (j) are the pebble volume fraction rate at the entry, ($\dot{n}^j(0, \tau) = \dot{n}^j_{in}(\tau)$), and exit, ($\dot{n}^j(s = S, \tau) = \dot{n}^j_{out}(\tau)$), of the reactor. The reshuffling operates by reinserting those partially depleted pebbles into the reactor in the following iteration ($\dot{n}^{j+1}_{in}$), and also by substituting the depleted pebbles with new fresh pebbles. Convergence is asserted when the difference between the results generated in two consecutive iterations is sufficiently small:

$$\|\dot{n}^{j+1}_{in}(\tau) - \dot{n}^j_{in}(\tau)\| < \epsilon. \tag{3}$$

This problem is homogeneous, and therefore it is determined only up to a scaling factor. A unique solution can be obtained by enforcing a normalization at the end of each step. In the equilibrium state, the reactor is in the steady state condition and the sum of fractional pebble volume rate is preserved to ensure that the packing factor is conserved. Therefore, the fractional pebble volume rate at the end of each iteration is divided by its norm. The normalization also prevents loss or gain caused by machine precision, avoiding under or overflow.

2.1. Discrete Form

Equation (2) can be discretized as

$$n_{g,m} = \frac{u_{m-1}A_{m-1}}{u_m A_m + \frac{\bar{p}_{g,m}V_m}{\Delta\tau_g}} n_{g,m-1} + \frac{\frac{\bar{p}_{g-1,m}V_m}{\Delta\tau_{g-1}}}{u_m A_m + \frac{\bar{p}_{g,m}V_m}{\Delta\tau_g}} n_{g-1,m}, \tag{4}$$

where the pebble burnup is discretized in ($g = 1, 2, \dots, G$) groups, and the streamline is discretized in ($m = 1, 2, \dots, M$) volume cells. Here, (M) is the last cell of the reactor before the exit and (G) is the number of burnup groups. It is required that there is at least one group above the maximum burnup limit (τ_{max}) to account for pebbles that will be discarded and replenished with new fresh pebbles in the next pass. The remaining lower burnup groups will be reinserted into the reactor. Furthermore, ($n_{g,m}$) and ($\bar{p}_{g,m}$) are, respectively, the fractional pebble volume and the average power density for the pebbles in the burnup group (g) and cell (m). (V_m) is the volume of the cell and ($\Delta\tau_g = \tau_g - \tau_{g-1}$) is the size of the burnup group (g). The last burnup group is determined to be large enough for a pebble to never surpass its upper limit ($g = G + 1$) in a single pass. The first derivative of the power density is approximated using the upwind approximation [5].

Griffin solves the system of equations generated in Eq. (4) by calculating progressively through a grid of (M) elements and (G) burnup groups. The calculation of ($n_{g,m}$) depends directly on the previous cells and smaller burnup groups that were generated upstream (linear sweep). The boundary condition at the entry ($\dot{n}_{in}^{j}$), which is calculated based on the previous iteration ($\dot{n}_{out}^{j-1}$), initiates the iterative step (j). Iterations are carried out until the convergence criterion is satisfied (Eq. (3)) for the inlet boundary condition.

2.2. Matrix Form

To study the convergence rate of Griffin's PBR algorithm's inner loop, we propose a different representation of the Eq. (4). In this formulation, we consider the following matrix form:

$$\mathbf{I}\vec{n}_m = \mathbf{A}_m\vec{n}_{m-1} + \mathbf{B}_m\vec{n}_m, \qquad \vec{n}_m = \big\{n_{1,m}, n_{2,m}, \cdots, n_{G,m}\big\}, \tag{5}$$

where ($\mathbf{I}$) is the identity matrix of order (G), ($\mathbf{A}_m$) is a diagonal matrix of order (G), and ($\mathbf{B}_m$) is a subdiagonal matrix of order (G), such that

$$\mathbf{A}_m = \begin{bmatrix} a_1 & & & \\ & a_2 & & \\ & & \ddots & \\ & & & a_G \end{bmatrix}, \qquad \mathbf{B}_m = \begin{bmatrix} 0 & & & & \\ b_{2,1} & 0 & & & \\ 0 & b_{3,2} & 0 & & \\ & \ddots & \ddots & \ddots & \\ & & 0 & b_{G,G-1} & 0 \end{bmatrix}, \tag{6a}$$

with (a_g) and ($b_{g,g-1}$) given by

$$a_g = \frac{u_{m-1}A_{m-1}}{u_m A_m + \frac{\bar{p}_{g,m}V_m}{\Delta\tau_g}} \qquad b_{g,g-1} = \frac{\frac{\bar{p}_{g-1,m}V_m}{\Delta\tau_{g-1}}}{u_m A_m + \frac{\bar{p}_{g,m}V_m}{\Delta\tau_g}}. \tag{6b}$$

The vector ($\vec{n}_m$) represents the pebble volume fraction of each (g) burnup group on cell (m).

The boundary condition at the output of the reactor ($\vec{n}_{out}^{j}$) is the pebble volume fraction distribution at the end of the last cell (M) of the streamline at iteration step (j). It is calculated by applying (M) times a linear operator ($\mathbf{S}_m$), one for each streamline cell, over the initial pebble volume fraction distribution at the entry of the reactor ($\vec{n}_{in}^{j}$):

$$\vec{n}_{out} = \prod_{m=1}^{M} \big((\mathbf{I} - \mathbf{B}_m)^{-1}\mathbf{A}_m\big)\vec{n}_{in} = \prod_{m=1}^{M} \big(\mathbf{S}_m\big)\vec{n}_{in}. \tag{7}$$

The inverse matrix ($(\mathbf{I} - \mathbf{B}_m)^{-1}$) can be represented by a Taylor expansion:

$$(\mathbf{I} - \mathbf{B}_m)^{-1} = \sum_{k=0}^{\infty} \mathbf{B}_m^k = \sum_{k=0}^{G-1} \mathbf{B}_m^k. \tag{8}$$

Since ($\mathbf{B}_m$) is nilpotent, the expansion is exact for the sum of the first (G) terms. This expansion results in a full lower triangular matrix, with unity in the diagonal terms.

In the case of constant conditions (i.e., uniform grid, constant tangential speed, and the same power density for all burnup groups and cells) the linear operator in Eq. (7) can be reduced to ($S_m = S$). That simplifies Eq. (7) to ($\vec{n}_{out} = \mathbf{S}^M \vec{n}_{in}$). It is relevant to note that ($\mathbf{S}$ and $\mathbf{S}^M$) are also lower triangular matrices. Following the reshuffling procedure, the last burnup group of ($\vec{n}_{out}$) at the iteration step (j) is discarded, and the same quantity is added as fresh pebbles to the input of the next iteration ($\vec{n}_{in}^{j+1}$). This can be mathematically represented as an operator ($\mathbf{P}$) that rearranges matrix ($\mathbf{S}^M$). The operator ($\mathbf{P}$) appears as

$$\mathbf{P} = \begin{bmatrix} 1 & & & & 1 \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 0 \end{bmatrix}. \tag{9}$$

Lastly, the norm of the vector is taken before the next step is initiated. That is,

$$\vec{n}_{in}^{j+1} = \frac{\mathbf{P}\mathbf{S}^M \vec{n}_{in}^j}{\|\mathbf{P}\mathbf{S}^M \vec{n}_{in}^j\|_1}. \tag{10}$$

Note that Eq. (10) has the form of a Power Iteration method with respect to ($\vec{n}_{in}$). Therefore, the method's spectral radius (ρ) can be calculated by the ratio of the two largest eigenvalues (λ_i) of ($\mathbf{P}\mathbf{S}^M$), ($\rho = \|\frac{\lambda_2}{\lambda_1}\|$) [8].

The spectral radius of the system represents the convergence rate of the method. Convergence requires ($\rho < 1$), and the closer to zero it is, the faster the method converges to the solution. When the spectral radius comes close to unity, the method becomes slow, and values above unity display instability. The advantage of the matrix formulation is to be able to analyze the linear system through the behavior of the matrix ($\mathbf{P}\mathbf{S}^M$) and the method's similarities to the power iteration method.

3. RESULTS

A computer program was developed in Python [9] to implement the matrix formulation and to compare it to the linear sweep of the discrete formulation in Section 2.1 for the convergence rate. Two tests were simulated. The first test aimed to compare Griffin's algorithm to the convergence rate of the power iteration method as the number of burnup groups increases. In the second test, we evaluate the spectral radius of both Python and Griffin codes as we change the design parameters.

The former test was performed to compare the convergence rate of Griffin's and the ratio of dominant eigenvalues in the Python's code under similar design parameters, only changing the number of burnup groups. We considered 128 streamline cells, 80 days for residence time, and ($\epsilon = 10^{-12}$)

for the convergence criterion. The reactor's parameters are considered to be 1.50 [m] radius, 10 [m] height, 250 [MW] total power, 3 [cm] pebble radius with 7 [g] of uranium, and packing fraction of 0.6. The burnup discharge limit was considered 200 [MJ$/cm^3$].

Table 1: Spectral Radius over Number of Burnup Groups.

Number of Burnup	Spectral Radius	
Groups	Griffin	Power Iteration
4 (+1[a])	0.492606 (36[b])	0.493325
8 (+1)	0.684700 (60)	0.685253
10 (+1)	0.735360 (72)	0.735257
40 (+1)	0.91411 (256)	0.914250
100 (+1)	0.95499 (480)	0.955103

a - G-th group above τ_{max}.
b - Number of iterations.

The results in Table 1 show that Griffin's and Python's spectral radii match closely. The general trend is observed for both methods, with the spectral radius increasing with the number of burnup groups considered, requiring a higher number of iterations for convergence. There is a small discrepancy between Griffin's and Python's. This effect is most likely caused by the digit precision available in Griffin and also the discrepancy of the initial condition of the estimated neutron flux and power distribution in the Griffin algorithm, which is considered to be constant and uniform in the Python code.

The latter test accounted for the effect of varying design parameters on both codes' convergence rates. Eight burnup groups were used in this simulation. One interesting result obtained by Griffin's and Python's algorithms is that their spectral radius increases when it reaches a specific threshold of replenished pebbles per pass, as illustrated in Fig. 3. This behavior was observed in both methods for different parameters, such as residence time and total reactor power. For simplicity, only the residence time is presented in this paper.

For this case, the spectral radius reaches a first minimum and then presents a considerable increase. This is around the range of 46-47% (240 days) of new pebbles inserted in the system after a single pass, reaching a maximum around the range of 53% (290 days) of replenished pebbles per pass. After that, the spectral radius approaches asymptotic behavior.

4. DISCUSSION

We have shown that Griffin's algorithm behaves in essence as a power iteration method, having consistent results between its convergence rate and the ratio of dominant eigenvalues from the matrix formulation (Python code). The increase of spectral radius when considering a higher number of burnup groups is observed, matching the values obtained for the power iteration method estimation derived from the numerical calculation of the eigenvalues. Further analysis needs to be conducted to estimate how the increase in the number of burnup groups impacts the matrix system and increases the eigenvalues, resulting in a higher spectral radius.

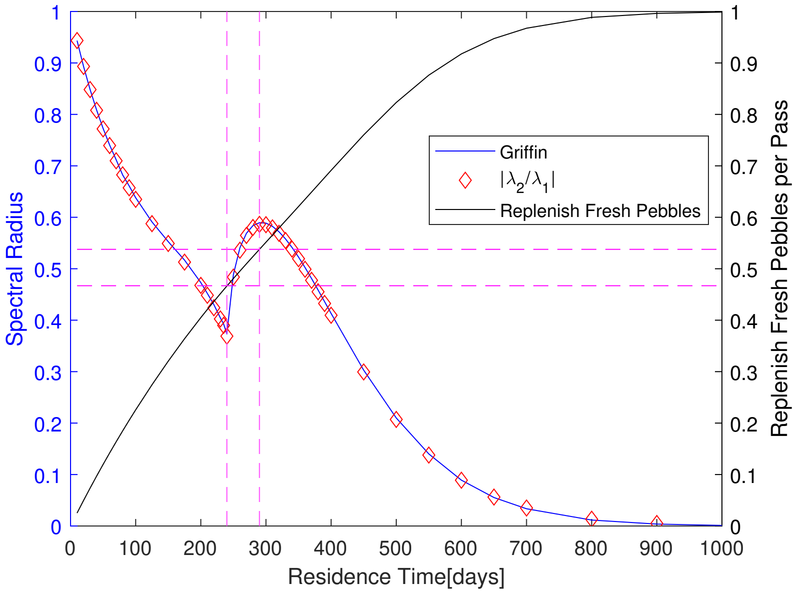

Figure 3: Spectral Radius over Residence Time in the Reactor.

The most surprising result obtained in this work is the observed behavior in Fig. 3. Physically, the algorithm has an upper and lower limit for the spectral radius in agreement with what would be expected. However, the intermediate region presented a nontrivial behavior. It is expected that the algorithm will have a high spectral radius when the pebbles are moving rapidly through the reactor. The lower the residence time, the less the burnup of the pebble per pass and iteration. This leads to a spectral radius closer to one. The opposite happens for large residence time. If the pebbles stay long enough inside the reactor for all of them to be discarded after a single pass, the algorithm will converge in one iteration. Therefore the spectral radius should tend to zero as the residence time increases. Initially, the authors expected a monotonically decreasing spectral radius with residence time. However, the re-circulation causes a significant change in the behavior of the system and therefore the solution.

The factor that apparently dictates this change of behavior is the percentage of pebbles being replaced per pass inside the reactor. Several iterations are required for the reactor to reach an equilibrium state at low residence time because the increments per step are small. The number of iterations required decreases as this parameter increases. However, when approaching the 50% mark for replenishing of fresh pebbles per pass, the refueling factor causes an effect akin to resetting the ($\dot{n}_{in}^{j}$) to similar values per iteration, slowing the convergence rate.

The reason behind the changes in the trend of the spectral radius is not trivial. Further investigation has shown that this behavior happens to any number of burnup groups. In Fig. 4, the spectral radius of the Python computer code is presented as a function of burnup groups and residence time. The $(+1^a)$ stands for the one burnup group above (τ_{max}). As the number of burnup groups increases, there is an earlier change to the behavior towards the second pattern (rapid increase), reaching the first minimum and maximum with lower residence time (210 and 270 days) and refreshed pebbles per pass (42% and 51%). These values seem to converge for a high number of burnup groups. It is also noticeable that for a higher number of burnup groups, there are additional oscillations before

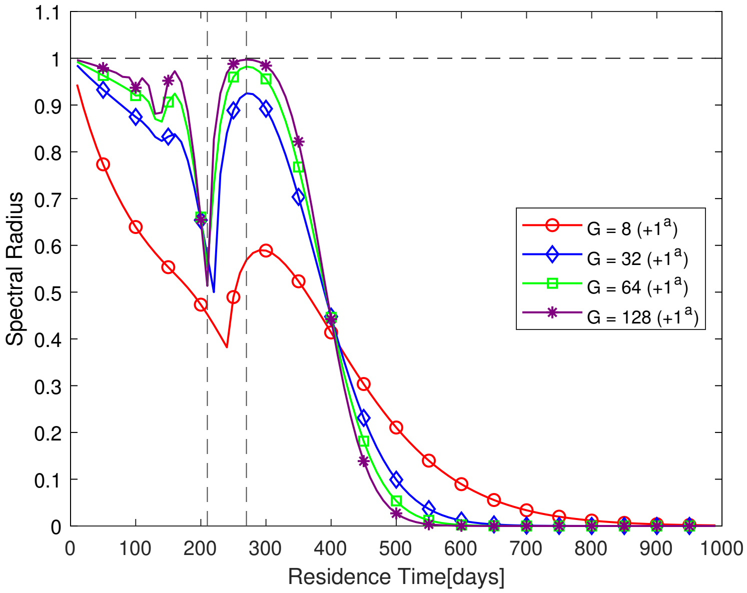

Figure 4: Spectral Radius over residence time and number of burnup groups.

the first minimum and a higher second peak, closer to unity. Understanding the reasons behind these extra oscillations will require further investigation and will be included in future work.

The eigenvalues of the matrix ($\mathbf{PS}^M$) were also evaluated individually. In Fig. 5, the norm of the dominant eigenvalues were calculated numerically over different residence times for the third test presented. Although the first dominant eigenvalue (λ_1) seems to be unity for all cases, the norm of the second (λ_2) and third (λ_3) dominant eigenvalues show an interesting behavior as the residence time increases.

First, there is a region where both the second and third largest eigenvalues are a complex conjugate, having the same norm. This eigenvalue pair decreases as the residence time increases until it reaches the point where the eigenvalues change from a complex conjugate to strictly real values. At this point, the eigenvalue pair bifurcates into two different trends, one reducing significantly and the other presenting a sharp increase to a peak, then reducing monotonically afterwards. The behavior of the latter matches exactly with the trend observed by the convergence rate of Griffin's code.

More studies regarding possible analytical solutions for the eigenvalues could bring some insights into the design parameters that define this threshold from the two different behavior patterns. However, the difficulty to obtain a general analytical equation for the eigenvalues, in this case, arises from the fact that the matrix ($\mathbf{PS}^M$) is no longer lower diagonal and its characteristic polynomial has the order of the number of burnup groups (G).

Investigating this "deceleration" phenomenon is important to understand how certain design parameters could impact the performance of future pebble bed reactor codes and acceleration methods. This may be especially meaningful when simulating more complex systems where the computational cost is an important factor, such as when there are multi-streamlines and multi-physics involved.

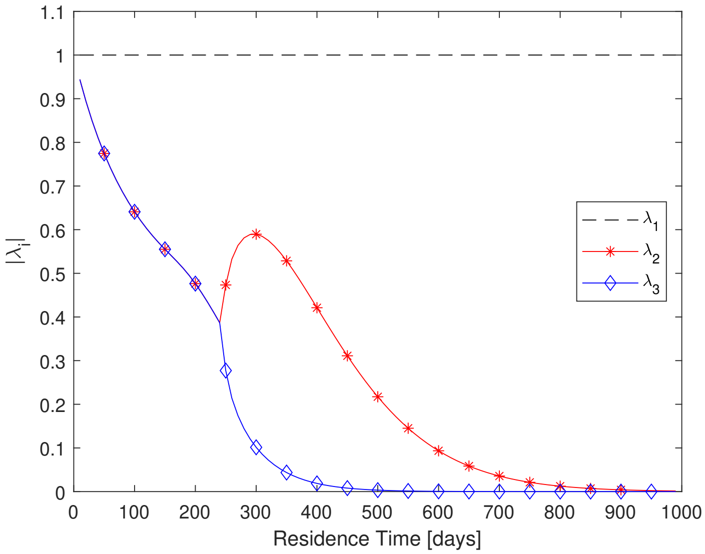

Figure 5: Dominant eigenvalue's norm over residence time.

Immediate future work will be focused on investigating an analytical solution for a low number of streamline cells and burnup groups. Solving the characteristic polynomial of the matrix ($\mathbf{PS}^M$) could provide some analytical equations for the design parameters that lead to the change of eigenvalues from complex to real. The end goal of this research is to develop acceleration techniques to improve the performance of Griffin's PBR algorithm, necessary when simulating realistic scenarios.

REFERENCES

[1] R. Baranwal, *Strategic Vision*, Office of Nuclear Energy, USA, (2021).

[2] S. Şahin and Y. Wu, *Comprehensive Energy Systems*, Elsevier, **Vol. 3**, pp. 590–637, (2018).

[3] C. Lee et al., "Griffin Software Development Plan", Idaho National Laboratory and Argonne National Laboratory, (2021).

[4] A. D. Lindsay et al., "2.0 - MOOSE: Enabling massively parallel multiphysics simulation", *SoftwareX*, **Vol. 20**, (2022).

[5] S. Schunert et al.,"RC Multiphysics Analysis Capability Deployment (FY 2021 - Part 1)", Idaho National Laboratory,pp. 24-46 (2020).

[6] S. Schunert et al.,"An Equilibrium Core Depletion Algorithm for Pebble-Bed Reactors in the Griffin Code", Idaho National Laboratory, USA (2023).

[7] A. Kadak and M. Bazant, "Pebble flow experiments for pebble bed reactors", *2nd International Topical Meeting on HIGH TEMPERATURE REACTOR TECHNOLOGY*, Beijing, China, Sept. 22-24, (2004).

[8] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, Philadelphia & USA (1997).

[9] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*, CreateSpace, Scotts Valley, CA & USA (2009).