# A Gauss-Seidel-Based Acceleration Method for Solving CRAM Burnup Equations

**Authors:** Sun Yuqing, Zhang Binhang, Yuan Xianbao, Tang Haibo

**Date:** 2025-07-22T16:35:29+00:00

## Abstract

Reactor burnup calculations are crucial to the safe operation and fuel management of nuclear power plants. In recent years, the Chebyshev Rational Approximation Method (CRAM) has emerged as one of the primary methods for solving burnup equation systems. When employing CRAM, complex matrix computations are typically performed using Sparse Gaussian Elimination (SGE), encompassing both symbolic and numeric elimination, which offers limited improvements in computational efficiency. Building upon the self-developed burnup calculation program AMAC, this study develops an accelerated solution method for CRAM burnup equation systems based on the Gauss-Seidel (GS) method. Utilizing three burnup databases of varying scales (containing 71, 221, and 1487 nuclides), comprehensive computational analysis was performed on a light water reactor benchmark problem in terms of both computational accuracy and efficiency. Regarding computational accuracy, with the linear subchain analytical method serving as the reference solution, the GS-based PFD and IPF forms exhibit comparable accuracy. For the calculation of short-lived nuclides, the IPF form achieves superior accuracy compared to the PFD form. In terms of computational efficiency, the GS method significantly outperforms the SGE method, achieving a maximum efficiency improvement of 80.17% across the three burnup libraries, which facilitates enhanced efficiency in high-fidelity burnup calculations. This study recommends the adoption of the GS-based IPF form of the burnup equation system for practical burnup calculations, as it provides a balanced consideration of both computational accuracy and efficiency.

## Full Text

## Study on the Accelerated Method for Solving the CRAM Burnup Equations Based on Gauss-Seidel

**SUN Yuqing[1], ZHANG Binhang[2,3,4,5]*, YUAN Xianbao[2,3], TANG Haibo[2,3]**

[1]College of Mechanical and Power Engineering, China Three Gorges University, Yichang 443002, China
[2]College of Mathematics and Physics, China Three Gorges University, Yichang 443002, China
[3]College of Nuclear Energy Science and Engineering, China Three Gorges University, Yichang 443002, China
[4]Applied Nuclear Technology in Geosciences Key Laboratory of Sichuan Province (Chengdu University of Technology), Chengdu 610000, China
[5]Three Gorges Mathematical Research Center, China Three Gorges University, Yichang 443000, China

**Abstract**

[Background]: Reactor burnup calculation is crucial for the safe operation and fuel management of nuclear power plants. In recent years, the Chebyshev Rational Approximation Method (CRAM) has become one of the primary approaches for solving burnup equations. When employing CRAM, complex matrix calculations typically rely on Sparse Gaussian Elimination (SGE), which involves both symbolic and numerical elimination, offering limited improvement in computational efficiency. [Purpose]: This study aims to develop a Gauss-Seidel (GS)-based acceleration method for solving CRAM burnup equations to enhance the computational efficiency of the burnup equation solver. [Methods]: Based on the self-developed burnup calculation code AMAC, we developed an acceleration method for solving CRAM burnup equations using the GS method. We conducted computational analyses of a light-water reactor benchmark from both accuracy and efficiency perspectives using three burnup databases of different scales (containing 71, 221, and 1487 nuclides). For accuracy evaluation, the linear sub-chain analytical method served as the reference solution. The GS-based Partial Fraction Decomposition (PFD) and Incomplete Partial Fractions (IPF) formulations demonstrated comparable computational precision. For short-lived nuclide calculations, the IPF formulation exhibited superior accuracy compared to the PFD formulation. [Results]: In terms of computational efficiency, the GS method significantly outperformed the SGE method, achieving a maximum efficiency improvement of 80.17% across the three burnup databases, which is beneficial for enhancing high-fidelity burnup calculation efficiency. [Conclusions]: This study recommends adopting the GS-based IPF formulation for burnup equations in practical calculations, as it effectively balances computational accuracy and efficiency.

**Keywords**: Burnup equations; Chebyshev Rational Approximation Method; Gauss-Seidel iteration; Sparse Gaussian Elimination; Acceleration method

## Introduction

Reactor burnup calculation is essential for the safe and stable operation of reactors, as it optimizes core design and controls reactivity to ensure reactor reliability throughout its entire lifecycle. In high-fidelity full-core calculations,

burnup calculations must simultaneously consider both computational accuracy and efficiency to achieve detailed characterization of various physical phenomena in the core and accurately predict key safety parameters during reactor service [1]. The Chebyshev Rational Approximation Method (CRAM) represents one of the primary methods for solving burnup equations. This method requires no special treatment for short-lived nuclides and offers high computational accuracy, making it widely adopted in various physics codes such as SERPENT, NECP-X, and Kylin-2 [2-4]. Most of these programs employ the Partial Fraction Decomposition (PFD) formulation of CRAM burnup equations for calculations. Since its introduction, numerous methodological studies and improvements have been conducted by domestic and international scholars to enhance its computational accuracy and efficiency [5-6]. Pusa et al. proposed the Incomplete Partial Fractions (IPF) rational approximation function expansion based on the PFD formulation, which significantly improved numerical stability and accuracy [7]. However, solving IPF-formulated CRAM burnup equations involves multiple complex matrix inversion and multiplication operations, resulting in relatively low computational efficiency. Although burnup calculations are primarily dominated by neutron transport equation solving, efficiency improvements in burnup equation solving can save hundreds of core-hours in computational time under conditions of large-scale burnup zones and detailed burnup chains, which is beneficial for enhancing high-fidelity burnup calculation efficiency [8].

When solving burnup equations using CRAM, computational time is mainly consumed by burnup matrix inversion, typically addressed using the Sparse Gaussian Elimination (SGE) method [6], which offers limited improvement in burnup calculation efficiency. Therefore, this study develops a Gauss-Seidel (GS)-based acceleration method for solving CRAM burnup equations within our self-developed burnup calculation code AMAC. We further employ a light-water reactor test case and analyze computational accuracy and efficiency using three burnup databases of different scales. By comparing the PFD and IPF formulations under the GS method, this research provides valuable insights for optimizing burnup calculation efficiency.

## 1 Burnup Equations

A burnup system typically contains hundreds to thousands of nuclides. For a burnup system containing different nuclides, it is necessary to solve the burnup equation to obtain the variation of nuclide densities, expressed in matrix form as:

$$\frac{d\mathbf{N}(t)}{dt} = \mathbf{A}\mathbf{N}(t)$$

where $\mathbf{N}(t)$ represents the nuclide density vector at time $t$, and $\mathbf{A}$ denotes the burnup matrix. The burnup matrix $\mathbf{A}$ exhibits strong sparsity and stiffness due to significant orders-of-magnitude differences in half-lives among radioactive

nuclides [9]. To visually demonstrate its distribution characteristics, Figure 1 [Figure 1: see original paper] illustrates the transmutation relationships among nuclides based on the ORIGEN burnup database. Only non-zero elements in the burnup matrix are color-coded in the figure. Row and column indices are arranged according to ZAI order (ZAI = 10000×Z + 10×A + I, where Z is the proton number, A is the mass number, and I is the excitation state of the nuclide) to facilitate rapid indexing and positioning of different nuclides [10-11]. As shown in Figure 1, matrix elements are primarily concentrated near the diagonal and on the right side, with a few elements at the top. Elements near the diagonal result from decay or transmutation reactions, right-side elements represent fission products, and top elements are mainly byproducts of certain reactions, such as H and He.

When all eigenvalues of the burnup matrix are distributed near the negative real axis, CRAM can achieve optimal rational approximation for the matrix exponential in the interval $(-\infty, 0]$ [6]. The rational approximation function $r(z)$ can be expressed as:

$$r(z) = \frac{p(z)}{q(z)} \approx e^z$$

where $p(z)$ and $q(z)$ are rational polynomials of order less than or equal to $k$; $k$ is the rational expansion order. When the order reaches 16, relatively accurate computational results can be obtained:

$$r_{16}(z) = \alpha_0 + \sum_{i=1}^{8} \frac{\alpha_i}{z - \theta_i}$$

where $\mathbf{I}$ is the identity matrix of the same order as the burnup matrix; $\mathbf{N}_0$ represents the initial nuclide density vector; $\alpha_0$ is the limit value of the function as time approaches infinity; and scalar $\alpha_i$ and $\theta_i$ are the residues and poles, respectively. To enhance CRAM computational accuracy, one must consider increasing the expansion order or employing detailed burnup chains.

Further research revealed that when nuclide densities of certain isotopes decrease sharply within a burnup time step, the computational accuracy of the PFD formulation decreases or directly leads to erroneous results [7]. Consequently, Pusa proposed the Incomplete Partial Fractions rational approximation function expansion, which offers higher computational accuracy and numerical stability compared to the PFD formulation. The IPF formulation is expressed as:

$$\mathbf{N}(t) \approx \alpha_0 \mathbf{N}_0 + 2\mathrm{Re}\left\{ \sum_{i=1}^{8} \alpha_i \left[ \prod_{j=1}^{i} (\mathbf{A} - \theta_j \mathbf{I})^{-1} \right] \mathbf{N}_0 \right\}$$

where $\alpha_i$ are IPF coefficients. Comparing equations (4) and (5) reveals that the IPF formulation of CRAM burnup equations requires $k/2$ complex matrix inversion and multiplication operations as the expansion order $k$ increases, resulting in relatively low computational efficiency.

## 2 GS-Based CRAM Burnup Equation Solving Method

As shown in equations (4) and (5), the key to solving different formulations of CRAM burnup equations lies in efficiently and accurately computing the coefficient matrix inverse $(\mathbf{A}-\theta_i\mathbf{I})^{-1}$. Different solving methods are typically selected based on the scale and sparsity of the coefficient matrix. For matrices with low order and sparsity, direct matrix inversion methods are commonly employed. However, for large sparse matrices, this approach suffers from low computational efficiency, prompting most current codes to adopt the SGE method for solving CRAM burnup equations.

The SGE-based solving process comprises symbolic factorization and numerical elimination [12]. Symbolic factorization analyzes potential non-zero elements (fill-in) generated during matrix elimination, aiming to reduce storage overhead during numerical elimination. Numerical elimination performs actual Gaussian elimination on the coefficient matrix based on symbolic factorization results, including forward elimination and back substitution. In the forward elimination stage, a non-zero element is selected as the pivot from unprocessed rows, which is then used to eliminate elements in the same column of other rows, making them zero. Repeating this process yields an upper triangular matrix. The back substitution stage solves unknowns sequentially starting from the last equation of the upper triangular matrix, ultimately obtaining the nuclide density vector. Notably, due to nuclide production and depletion during burnup, the positions of non-zero elements in the coefficient matrix change. Consequently, symbolic factorization must be repeated for the coefficient matrix at each burnup step before numerical elimination can be completed.

In contrast, the GS method is an iterative approach for solving linear systems that exhibits rapid convergence for diagonally dominant matrices. The core idea involves continuously updating each component of the solution vector to gradually approach the exact solution without requiring matrix preprocessing. Therefore, the GS method eliminates the need for repeated symbolic factorization across different burnup steps, yielding higher computational efficiency. Additionally, the GS method offers storage advantages, requiring only the coefficient matrix, vectors, and the iterative solution vector without storing numerous intermediate results, resulting in lower computational overhead. The method's computational process is relatively straightforward, facilitating easier code implementation and maintenance. Considering the sparse and stiff characteristics of burnup matrices, and combining them with the magnitude of coefficients $\alpha_i$ and $\theta_i$ in PFD and IPF formulations, the coefficient matrix exhibits diagonal dominance. Therefore, this study develops a GS-based acceleration method for solving CRAM burnup equations to enhance computational efficiency.

The inverse calculation of the coefficient matrix can be expressed as:

$$\mathbf{x} = (\mathbf{A} - \theta_i \mathbf{I})^{-1} \mathbf{N}_0$$

where coefficient $\theta_i$ and vector $\mathbf{N}_0$ correspond to $\theta_i$ and $\mathbf{N}_0$ in PFD and IPF formulations, respectively. By defining the iteration matrix $\mathbf{M} = (\mathbf{A} - \theta_i \mathbf{I})$ and substituting the matrix expression, equation (6) can be transformed into iterative form:

$$\mathbf{M}\mathbf{x}^{(l+1)} = \mathbf{N}_0$$

where $\mathbf{x}^{(0)} = \mathbf{N}_0$ serves as the initial boundary condition; $\mathbf{x}^{(l+1)}$ represents the solution vector at the $(l+1)$-th iteration; and $l$ is the iteration number.

When solved using the GS iteration, the component-wise iterative format is as follows:

$$x_i^{(l+1)} = \frac{1}{m_{ii}} \left( b_i - \sum_{j=1}^{i-1} m_{ij} x_j^{(l+1)} - \sum_{j=i+1}^{n} m_{ij} x_j^{(l)} \right)$$

where $m_{ii}$ denotes the diagonal element of coefficient matrix $\mathbf{M}$; $m_{ij}$ represents the element in the $i$-th row and $j$-th column of the coefficient matrix; $x_i^{(l+1)}$ is the $i$-th component of the solution vector at the $(l+1)$-th iteration; $x_j^{(l)}$ is the $j$-th component of the solution vector at the $l$-th iteration; and $l$ indicates the iteration count.

Substituting equation (8) into equation (7) and iterating until convergence, and combining with equations (4) and (5), we obtain the final iterative forms for PFD and IPF formulations, respectively:

$$\mathbf{N}(t) \approx \alpha_0 \mathbf{N}_0 + 2\mathrm{Re} \left\{ \sum_{i=1}^{8} \alpha_i \mathbf{x}_i^{(l)} \right\}$$

$$\mathbf{N}(t) \approx \alpha_0 \mathbf{N}_0 + 2\mathrm{Re} \left\{ \sum_{i=1}^{8} \alpha_i \left[ \prod_{j=1}^{i} \mathbf{x}_j^{(l)} \right] \right\}$$

As shown in equations (9) and (10), each iteration result represents the nuclide density at the current iteration count. This study employs a convergence criterion based on the nuclide density norm:

$$\frac{\|\mathbf{x}^{(l+1)} - \mathbf{x}^{(l)}\|_2}{\|\mathbf{x}^{(l+1)}\|_2} < \epsilon$$

where $\epsilon$ is a convergence value set according to actual computational accuracy requirements.

We implemented the GS-based CRAM burnup equation solving acceleration method in the burnup calculation code AMAC. AMAC is a self-developed code for burnup and radioactive decay calculations, designed to compute material nuclide compositions and radiation characteristics. The program integrates multiple burnup solving methods and can be flexibly coupled with transport calculation codes, demonstrating good computational accuracy and efficiency in validation analyses against internationally recognized codes such as ORIGEN and FISPACT [13-14]. In transport-burnup coupling calculations, transport computations account for over 90% of the total computational cost. To improve efficiency and reduce computational expense, AMAC has developed a model-order-reduction-based transport-burnup coupling framework, primarily including: (1) a dynamic mode decomposition-based transport-burnup coupling method to significantly reduce transport calculation frequency and improve efficiency [15]; and (2) a Grassmann manifold-based burnup reduction model to substantially reduce burnup matrix construction costs during coupling calculations [16].

The AMAC program architecture, shown in Figure 2 [Figure 2: see original paper], consists of three main components: pre-processing, solver, and post-processing. Pre-processing reads input cards to obtain material compositions, time steps, flux levels, and other relevant parameters, while simultaneously reading and storing decay constants, microscopic cross-sections, fission yields, and other parameters from the burnup database to construct the burnup matrix for the solver. Considering the sparse nature of burnup matrices, AMAC employs sparse matrix storage formats to reduce storage overhead and improve computational efficiency. The solver includes four different solving methods: TTA, MMPA, CRAM, and QRAM. The CRAM method is further divided into PFD and IPF formulations based on solving form. The post-processing component formats output results, covering nuclide densities, radioactivity, decay heat, decay photon spectra, etc. Based on the aforementioned computational methods and procedures, the method implementation was completed in the AMAC program, with the computational flow shown in Figure 3 [Figure 3: see original paper].

## 3 Numerical Verification and Analysis

This study selected three burnup databases of different scales to analyze a light-water reactor test case, with an irradiation flux of $1.5\times10^{14}$ $\mathrm{cm^{-2}s^{-1}}$, 200 days of irradiation followed by 10 years of cooling, and material compositions as shown in Table 1 [7]. The three selected burnup databases contain 71, 221, and 1487 nuclides (hereinafter referred to as Database-71, Database-221, and Database-1487). The verification strategy for the GS-based CRAM burnup equation solving method comprises two aspects: first, computational accuracy, where GS and SGE methods are employed under PFD and IPF formulations to solve burnup

matrices of different scales, comparing and analyzing their numerical precision; second, computational efficiency, where efficiency analyses of CRAM equations under both solving formulations are conducted across the three databases, considering the impact of different expansion orders on computational efficiency to validate the effectiveness of the GS method. The Transmutation Trajectory Analysis (TTA) method, based on analytical solutions, can precisely calculate every nuclide in burnup chains and is commonly used as a reference solution for various numerical methods [17]. Therefore, this study selected TTA results as the reference solution to verify the correctness of the GS method, with a convergence value $\epsilon$ of $1.0 \times 10^{-20}$. The computational hardware used was a computer with an AMD Ryzen 7 7745HX with Radeon Graphics (3.60 GHz) processor.

### 3.1 Computational Accuracy

We first calculated the test case using GS and SGE methods under CRAM's two different solving formulations, PFD and IPF, across the three burnup databases. Database-71 contains 71 nuclides, including 21 fissile nuclides, 49 fission products, and 1 pseudo-nuclide. Database-221 contains 221 nuclides, including 28 fissile nuclides and 193 fission products. Both databases include 8 reaction types: $(n,\gamma)$, $(n,2n)$, $(n,3n)$, $(n,p)$, $(n,\alpha)$, $(n,f)$, decay, isomeric transition, and electron capture reactions [18], which have been extensively validated and applied in software such as MVP [19] and MOSRA [20]. Database-1487 integrates burnup databases from ORIGEN-S and ORIGEN-2, containing 1487 nuclides, 23 neutron reaction cross-sections, 11 decay reactions, and 30 fission yields for heavy nuclides.

Table 2 presents the number of short-lived nuclides, minimum nuclide lifetime, maximum nuclide lifetime, and number of non-zero elements in the constructed burnup matrix for the three databases. The selection criterion for short-lived nuclides follows the ORIGEN program [10,17], as shown in equation (12):

$$T_{1/2} \leq 0.1\Delta t$$

where $T_{1/2}$ is the nuclide half-life (s) and $\Delta t$ is the time step (s).

To evaluate the numerical precision performance of GS and SGE, we compared the Maximum Nuclide Absolute Relative Difference (MNARD) and Average Nuclide Absolute Relative Difference (ANARD). MNARD refers to the maximum absolute relative difference among all calculated nuclide densities, while ANARD represents the average absolute relative difference across all calculated nuclide densities.

Table 3 shows the MNARD and ANARD for nuclide densities in Database-1487, using SGE calculation results as the reference solution. The results indicate that PFD-16 achieves MNARD and ANARD on the order of $1.0 \times 10^{-10}$ and $1.0 \times 10^{-12}$, respectively, with GS having minimal impact on PFD formulation

calculation accuracy. Meanwhile, the three IPF formulations (IPF-16, IPF-32, and IPF-48) achieve MNARD and ANARD on the order of $1.0\times10^{-13}$ and $1.0\times10^{-14}$, respectively, demonstrating that GS possesses excellent computational accuracy for solving IPF-formulated burnup equations.

Tables 4 and 5 present the MNARD and ANARD for Databases-221 and 71, respectively. In Database-221 calculations, PFD-16 achieves MNARD and ANARD on the order of $1.0 \times 10^{-10}$ and $1.0 \times 10^{-12}$, while the three IPF formulations reach $1.0 \times 10^{-13}$ and $1.0 \times 10^{-14}$. Further reducing the database scale to Database-71, IPF-16 achieves MNARD and ANARD on the order of $1.0 \times 10^{-10}$ and $1.0 \times 10^{-11}$, while IPF formulations reach $1.0 \times 10^{-14}$, approaching the effective digit limit of double-precision floating-point numbers and indicating excellent numerical precision of the GS method. Tables 3 through 5 show that MNARD and ANARD under IPF formulation using the GS method across the three databases all reach above $1.0 \times 10^{-13}$, while PFD formulations reach above $1.0 \times 10^{-10}$. The order-of-magnitude difference between the two formulations primarily stems from the PFD formulation's sensitivity to rounding errors in the rational approximation function, whereas the IPF formulation employs an incomplete partial fraction decomposition rational expansion that is less sensitive to rounding errors, offering better numerical stability and precision. These results verify that the GS method exhibits good numerical precision for solving both IPF and PFD burnup equations across different database scales, meeting the accuracy requirements for practical burnup calculations.

**3.1.1 Burnup Database-1487** Among the three databases used in this study, Database-1487 contains the most comprehensive nuclide types and reaction types, with the largest burnup matrix scale. When stored in sparse format in AMAC, the number of non-zero elements reaches 37,847, as shown in Table 2. Additionally, the magnitude difference between the longest-lived and shortest-lived nuclides reaches $10^{31}$, making the burnup matrix the most stiff. Using the TTA method as the reference solution with a truncation value of $1.0 \times 10^{-20}$, Figure 4 [Figure 4: see original paper] shows the relative deviation of nuclide densities calculated using the GS method based on Database-1487. The results indicate that relative deviations for both GS-based PFD and IPF formulations are less than $1.0 \times 10^{-7}$, with PFD-16 achieving an ANARD of $1.347 \times 10^{-9}$, while IPF demonstrates overall better accuracy than PFD. As the IPF expansion order increases, computational accuracy gradually improves, with IPF-48 reaching an accuracy of $1.0 \times 10^{-15}$ for nuclide calculations.

Considering that Database-1487 contains 484 short-lived nuclides that directly affect the calculation accuracy of end-of-chain nuclides, Table 6 further presents the relative deviations of short-lived nuclides at the end of irradiation under PFD and IPF formulations. The results show that compared to PFD, the GS-based IPF formulation significantly improves calculation accuracy for short-lived nuclides, with most achieving relative deviations on the order of $1.0 \times 10^{-15}$.

Nuclide $^{126}$Sb exhibits the maximum relative deviation of $3.963 \times 10^{-8}$, corresponding to a nuclide density of $1.930 \times 10^{-15}$ atoms/cm$^3$, which has negligible impact on the burnup system. These results verify the correctness of the GS-based CRAM burnup equation solving method for detailed burnup chain calculations.

**3.1.2 Burnup Database-221** In Database-221 calculations, the number of short-lived nuclides decreases to 51 compared to Database-1487, and the magnitude difference between longest-lived and shortest-lived nuclides decreases from $10^{31}$ to $10^6$, significantly reducing burnup matrix stiffness. The number of non-zero elements in the burnup matrix is 5,703. Figure 5 [Figure 5: see original paper] shows the relative deviation of nuclide densities calculated using the GS method. As the expansion order increases, IPF computational accuracy gradually improves, with IPF-16, IPF-32, and IPF-48 all achieving relative deviations below $1.0 \times 10^{-10}$. For PFD results, except for nuclide $^{237}$U with a relative deviation of $4.055 \times 10^{-10}$, all other nuclides show relative deviations below $1.0 \times 10^{-10}$. Table 7 further presents the relative deviations of short-lived nuclides at the end of irradiation under both formulations. The IPF formulation demonstrates enhanced accuracy for short-lived nuclides compared to PFD, with relative deviations below $1.0 \times 10^{-15}$. These results demonstrate that IPF offers higher computational accuracy for short-lived nuclides and verify the applicability and effectiveness of the GS method for medium-scale burnup database calculations.

**3.1.3 Burnup Database-71** Database-71 represents a simplified burnup chain containing only 19 short-lived nuclides, with a magnitude difference of $1.0 \times 10^3$ between longest-lived and shortest-lived nuclides, the lowest burnup matrix stiffness, and 1,149 non-zero elements. By tracking this minimal-scale burnup system, we evaluated the correctness of the GS method for simplified burnup chain calculations. Figure 6 [Figure 6: see original paper] shows the relative deviation of nuclide densities calculated using the GS method, with PFD and IPF results showing good agreement with the reference solution. PFD achieves an ANARD of $1.578 \times 10^{-11}$, while IPF reaches $1.0 \times 10^{-13}$ across three expansion orders, demonstrating higher accuracy. For short-lived nuclides, Table 8 presents the relative deviations of $^{237}$U and $^{242}$Cm at the end of irradiation, confirming that IPF formulation again provides superior accuracy compared to PFD.

### 3.2 Computational Efficiency

As described in Section 2, the SGE method requires symbolic factorization preprocessing before numerical elimination when solving CRAM burnup equations. The GS method requires no preprocessing and directly performs iterative solving on diagonally dominant coefficient matrices. Table 9 presents the computational time for PFD-16 formulation based on SGE and GS methods across the three

databases. Runtime statistics were collected with actual execution times presented at a 95% confidence level. For Database-71, the GS method required only 0.857 ms, achieving a 69.27% efficiency improvement over the SGE method's 2.789 ms. When the database scale increased to Database-221, SGE computation time increased from 2.789 ms to 19.335 ms, while GS time only increased from 0.857 ms to 2.977 ms. As the burnup database scale further increased, the number of non-zero elements in the burnup matrix grew significantly, with SGE computation time for Database-1487 reaching 65.948 ms compared to GS's 35.305 ms, representing a 46.92% efficiency improvement. For the SGE method, symbolic factorization accounts for approximately 30-40% of total computation time, with the primary time consumption in numerical elimination growing with the number of non-zero elements. For the GS method, the average number of iterations is 3 for Databases-71 and 221. For Database-1487, the introduction of numerous short-lived nuclides increases the average iteration count to 5, leading to increased computation time.

Table 10 further presents computational times for IPF-16 formulation based on SGE and GS methods. The results demonstrate significant efficiency improvements with the GS method. Comparing Tables 9 and 10 reveals that under identical database scales and expansion orders, IPF formulation computation time is slightly longer than PFD formulation. This difference primarily stems from their mathematical expressions: as shown in equations (4) and (5), PFD involves summation operations while IPF involves successive multiplication operations, which are computationally more complex and thus marginally increase computation time. As demonstrated in Section 3.1, IPF offers better computational accuracy and numerical stability. This study recommends using IPF formulation for CRAM burnup equation solving in practical applications, as the increased computation time compared to PFD is negligible.

Further increasing the IPF expansion order, Tables 11 and 12 present computational times for IPF-32 and IPF-48 formulations using SGE and GS methods. As the IPF expansion order increases, the number of successive matrix multiplications in equation (5) increases accordingly, leading to increased computation time. Additionally, for Database-1487 calculations, the GS iteration count is 5 compared to an average of 3 for Databases-71 and 221. The introduction of numerous short-lived nuclides increases iteration count, and with 37,847 non-zero elements, each GS iteration consumes more time, increasing overall computation time. Nevertheless, compared to the SGE method, the GS method achieves a 34.30% efficiency improvement for Database-1487 calculations. In summary, the GS-based CRAM burnup equation solving method developed in this study demonstrates computational efficiency advantages over the SGE method across three different database scales.

## Conclusion

Based on the self-developed burnup calculation code AMAC, this study developed a GS-based acceleration method for solving CRAM burnup equations.

Using three burnup databases of different scales, we completed verification and analysis of CRAM burnup equations based on PFD and IPF formulations. In terms of computational accuracy, we first verified that GS and SGE methods achieve comparable numerical precision. Using TTA algorithm results as the reference solution, we completed calculations for GS-based PFD-16, IPF-16, IPF-32, and IPF-48. The results demonstrate that the maximum relative deviation for all nuclides using the GS method is less than $1.0 \times 10^{-7}$, meeting the accuracy requirements for practical burnup calculations in both PFD and IPF formulations. Regarding computational efficiency, the GS method offers advantages over the SGE method, with maximum efficiency improvements reaching 80.17%. Considering that high-fidelity full-core burnup calculations may require solving burnup equations billions of times, an 80% reduction in burnup equation solving time translates to savings of hours or even days of computation time. Future work will integrate this method into a model-order-reduction-based transport-burnup coupling framework for large-scale burnup zone calculations to further enhance overall burnup calculation efficiency and reduce computational costs. This study recommends employing the GS-based IPF formulation for burnup equation solving in practical calculations, as it effectively balances computational accuracy and efficiency, proving beneficial for improving high-fidelity burnup calculation efficiency.

**Author Contributions**: SUN Yuqing established the model, researched and analyzed data, and drafted the manuscript; ZHANG Binhang designed the specific research content and direction, proposed reasonable research plans and theoretical support, guided program usage and data collection, reviewed the intellectual content of the paper, and was responsible for manuscript revision; YUAN Xianbao provided process supervision and results acceptance; TANG Haibo provided technical support.

## References

1. 吴宏春, 刘宙宇, 周欣宇, 等. 数值反应堆的研究现状与发展建议 [J]. 原子能科学技术, 2022, 56(02): 193-212. DOI: 10.7538/yzk.2021.youxian.0982. WU Hongchun, LIU Zhouyu, ZHOU Xinyu, et al. Research status and development suggestion of numerical reactor[J]. Atomic Energy Science and Technology, 2022, 56(2): 193–212. DOI: 10.7538/yzk.2021.youxian.0982.

2. Lopez-Solis R C, François J L, Bastida-Ortiz G E, et al. Fuel depletion analysis of a small sodium fast reactor with KANEXT and SERPENT[J]. Annals of Nuclear Energy, 2016, 98: 26–35. DOI: 10.1016/j.anucene.2016.07.024.

3. Hu Y Y, Liao H K, Luo Q, et al. Study on an improved burnup algorithm in Kylin-2 code[J]. Annals of Nuclear Energy, 2021, 153: 108034. DOI: 10.1016/j.anucene.2020.108034.

4. Wen X J, Liu Z Y, Chen J, et al. Development and validation of the depletion capability of the high-fidelity neutronics code NECP-X[J]. Annals of

Nuclear Energy, 2020, 138: 107096. DOI: 10.1016/j.anucene.2019.107096.

5. Pusa M. Rational approximations to the matrix exponential in burnup calculations[J]. Nuclear Science and Engineering, 2011, 169(2): 155–167. DOI: 10.13182/nse10-81.
6. Pusa M, Leppänen J. Solving linear systems with sparse Gaussian elimination in the Chebyshev rational approximation method[J]. Nuclear Science and Engineering, 2013, 175(3): 250–258. DOI: 10.13182/nse12-52.
7. Maria P. Higher-order Chebyshev rational approximation method and application to burnup equations[J]. Nuclear Science and Engineering, 2016, 182(3): 297–318. DOI: 10.13182/nse15-26.
8. Calvin, Olin William. Depletion chain simplification with pseudo-nuclides to model decay effects[D]. University of Idaho, 2023.
9. De Lanversin, J. de Troullioud, Kuett. ONIX: An open-source burnup code for nuclear archaeology[D]. Princeton University.
10. Croff A G. User's manual for ORIGEN2 computer code[R]. Oak Ridge National Laboratory, 1980.
11. Forrest R A. FISPACT-2007: User manual[M]. EURATOM/UKAEA fusion association, 2007.
12. Golub G H. Matrix computations[M]. 4th ed. America: Johns Hopkins University Press, 1983: 611-616.
13. Zhang B H, Yuan X B, Zhang Y H, et al. Development of a versatile depletion code AMAC[J]. Annals of Nuclear Energy, 2020, 143: 107446. DOI: 10.1016/j.anucene.2020.107446.
14. Zhang, B H, Zhang, C, Yuan, X B, et al. Error analyses and evaluation for solving burnup equations with MMPA method. Annals of Nuclear Energy, 2023, 193(1) DOI:10.1016/j.anucene.2023.110032.
15. Zhang B, Gong H, et al. A novel physics-constrained dynamic mode decomposition based neutronics-depletion coupling method for efficient burnup calculation nuclear reactors[J]. Nuclear Science and Engineering. DOI: https://doi.org/10.1080/00295639.2025.2490312.
16. Zhang B, Bi Y, et al. Adaptive POD-ROM based on Grassmann manifold: A robust and efficient approach for neutronics-depletion coupling calculations in nuclear reactors[J]. Progress in Nuclear Energy, 2025, 189: 105932. https://doi.org/10.1016/j.pnucene.2025.105932.
17. Isotalo A E, Aarnio P A. Comparison of depletion algorithms for large systems of nuclides[J]. Annals of Nuclear Energy, 2011, 38(2–3): 261–268. DOI: 10.1016/j.anucene.2010.10.019.

18. Tasaka, K, Katakura, J, Ihara, H, et al. JNDC nuclear data library of fission products. Second version[R]. Japanese Nuclear Data Committee, Tokai. 1990. https://doi.org/10.11484/jaeri-1320.

19. Nagaya Y, Okumura K, Mori T. Recent developments of JAEA's Monte Carlo code MVP for reactor physics applications[J]. Annals of Nuclear Energy, 2015, 82: 85–89. DOI: 10.1016/j.anucene.2014.09.037.

20. Nakano Y, Okubo T. Plutonium isotopic composition of high burnup spent fuel discharged from light water reactors[J]. Annals of Nuclear Energy, 2011, 38(12): 2689–2697. DOI: 10.1016/j.anucene.2011.08.009.

*Note: Figure translations are in progress. See original paper for figures.*