Annals of Nuclear Energy 118 (2018) 99–106

Contents lists available at ScienceDirect

Annals of Nuclear Energy

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

A new Krylov subspace method based on rational approximation to solve stiff burnup equation

Xuezhong Li, Jiejin Cai *

School of Electric Power, South China University of Technology, Guangzhou, Guangdong 510640, China

ARTICLE INFO

Article history:
Received 19 February 2018
Received in revised form 2 April 2018
Accepted 4 April 2018

Keywords:
Burnup equation
Matrix exponential
Krylov Subspace Method
GMRES
Rational approximation

ABSTRACT

A burnup equation can be solved with matrix exponential method and its solution can be written as $n(t) = e^{At}n(0)$. In burnup calculation, general Krylov Subspace Method can solute a matrix–vector efficiently in a subspace but fails to keep a high precision. To solve this problem, a new kind of Krylov Subspace Method, Generalized Minimal Residual Method (GMRES) is implemented, based on a rational approximation method. It shows its great advantage in computation speed, which is more than four times faster than the same kind of rational approximation solved in a whole space while its accuracy is also guaranteed. Some optimizations, such as shift-Invariant technique, precondition technique and restart technique, are also implemented on burnup calculation.

1. Introduction

Burnup calculation is to simulate changes in the material composition of nuclear fuels, which is very important for the reactors operating, cooling and radio protection. Accuracy and speed of calculation are the most important issues.

At present, there are two main ways to solve the burnup equation. One is to solve burnup chains, such as Transmutation Trajectory Analysis (TTA) (Cetnar, 2006), while the other is to use a matrix exponential approximation. In recent years, the matrix exponential method has been widely studied and it seems to be more attractive in both accuracy and speed (Moler et al., 2003), which may help a lot in the burnup calculation. The burnup equation, which represents the decay and transmutation of more than 1000 nuclides, can be written in a matrix form as:

$$\frac{dn(t)}{dt} = An(t); \tag{1}$$

In Eq. (1), $n = [n_1, \ldots, n_i, \ldots, n_n]^T$ and $n_i$are the atomic density of the nuclide i. The n × n matrix A is called burnup matrix whose diagonal elements represent the removal rate of each nuclides and off-diagonal elements represent production rates from other nuclides (Isotalo and Pusa, 2016):

$$A_{i,i} = -\lambda_i - \phi\sum_r \sigma_{i,r}; \tag{2}$$

and

$$A_{i,j\neq i} = b_{i,j}\lambda_j + \sum_r y_{i,j,r}\sigma_{j,r}\phi; \tag{3}$$

where λ_i = decay constant of nuclide i; Φ = neutron flux; $b_{i,j}$ = branching ratio from nuclide j to nuclide i; $\sigma_{i,r}$ = cross section of reaction r of nuclide i; $y_{i,r}$ = yield of nuclides when nuclides j undergoes reaction r;

In a small time-interval, the burnup matrix A is calculated as a constant. Then Eq. (1) is a time-domain ordinary differential equation and its solution is:

$$n(t) = e^{At}n(0); \tag{4}$$

where n(t) is the nuclide concentration vector.

In this case, the burnup calculation can be easily solved if proper measures are taken to deal with the matrix exponential e^{At}. However, the n × n burnup matrix A is not only large in size but also stiff to calculate. Because of the existence of the short-lived nuclides, whose rates of change outclass other nuclides, the norm and the stiffness of the matrix becomes very huge. In decay calculation, the norm of the matrix is up to 10^{15}. Besides, multiplying the time step t, which may be up to 10^5, will cause even more computational problems.

There are many way to solve matrix exponential, including truncated Taylor series expansion, Pade rational approximation

* Corresponding author at: School of Electric Power, South China University of Technology, 381 Wushan Road, Tianhe, Guangzhou 510640, China.
E-mail address: chiven77@hotmail.com (J. Cai).

https://doi.org/10.1016/j.anucene.2018.04.005
0306-4549/

method (PRAM), Chebyshev rational approximation method (CRAM) (Pusa and Leppanen, 2010) and Krylov Subspace Method (KSM) (Yamamoto et al., 2007).

Since Pade rational approximation is expanded near the origin, it can only deal with a matrix of small norm excluding the burnup matrix. Using scaling and squaring technique (Al-Mohy and Higham, 2009; Higham, 2005) may help a lot. Truncated Taylor serious expansion has similar problems and it also has a large round-off error.

Chebyshev rational approximation method, which expands e^x near the negative real axis, seems to work well since the eigenvalues of the burnup matrix converge to the negative real axis (Pusa and Leppanen, 2010). All the methods mentioned above share a flaw that they require a large computational cost and thus take a long time.

Krylov Subspace Method (KSM) is implemented widely on a system of linear equations. It can deal with a large matrix very fast because it projects a large matrix into a small one with the help of Arnoldi algorithm (Hochbruck and Lubich, 1997). An advantage of KSM to solve matrix exponential is that only matrix-vector multiplications are carried out and therefore lower computation cost is required. A lot of meliorations have been studying, such as shift and invert technique (Botchev, 2016) and precondition techniques (Yeung et al., 2017). However, the accuracy of KSM depends on the spectral properties of the matrix but unfortunately the burnup matrix is too stiff to projected accurately with the normal Arnoldi algorithm.

To solve this computational problem, this paper mainly introduced a new Krylov Subspace Method based on rational approximation, using a specially modified Arnoldi algorithm: Generalized Minimal Residual Method (GMRES). The remainder of this paper is organized as follows: In Section 2, rational approximation is briefly introduced. In Section 3, the general KSM is reviewed. Section 4 mainly describes a New Krylov Subspace Method and its rough errors analysis is also discussed. Implemented on the burnup calculation, the numerical results and discussions are presented in Section 5 and conclusion is summarized in Section 6.

2. Rational approximation method

With rational approximation, e^x can be approximated in a certain interval and written in a form as:

$$\mathcal{R}_\nu = \Phi_\nu / \Psi_\nu; \tag{5}$$

where ν means the order of the numerator and denominator;

This can also be written in partial fraction expansion as:

$$\mathcal{R}_\nu(\lambda) = \tau_0 + \sum_{j=1}^{\nu} \frac{\tau_j}{(\lambda - \xi_j)}; \tag{6}$$

where $\xi_1, \ldots, \xi_\nu$ are the poles and $\tau_1, \ldots, \tau_j$ are the corresponding residues. τ_0 is the limit of the function $\mathcal{R}_\nu$ at infinity. Applied to the burnup calculation, the solution can be written as:

$$n(t) = \mathcal{R}_\nu(\lambda) n_0 = \tau_0 n_0 + \sum_{j=1}^{\nu} \tau_j (A - \xi_j I)^{-1} n_0; \tag{7}$$

In Eq. (7), with no need for matrix inversion, only ν linear systems of equation $(A - \xi_j I)x = \mathrm{b}$ need to be solved. In general, Gauss-Seidel iterative method is a general way to get the solution when n is large enough and transforming into a sparse form will even make it more efficiently.

3. General Krylov Subspace Method

Krylov Subspace Method (KSM) is a very efficient way to solve large and stiff ordinary differential equations (ODEs). KSM tries to project a matrix of large size onto a subspace and the matrix becomes a small one, which is easier for calculation. And the solution is searched in a subspace instead of the whole space. The subspace can be written as:

$$\mathrm{K}\{\mathrm{A}, \mathrm{x}\} = \mathrm{span}\{\mathrm{v}, \mathrm{Av}, \ldots, A^{m-1}\mathrm{v}\}; \tag{8}$$

In KSM, matrix exponential is not directly calculated but it relies on a matrix-vector value. With Arnoldi procedure [shown in Algorithm 1], which is similar to Gram-Schmidt procedure, the orthogonal basis $V_m = \{v_1, \ldots, v_m\}$ is built. This means that the linear combination of $\overrightarrow{v_1}, \overrightarrow{v_2}, \overrightarrow{v_3}, \ldots, \overrightarrow{v_{j+1}}$ can fully express the vector $\mathrm{A}\overrightarrow{v_j}$ in subspace $\mathrm{K}\{\mathrm{A}, \mathrm{x}\}$.

Algorithm 1 Arnoldi procedure

Input: a n × n burnup matrix tA, origin composition n_0, Krylov subspace dimension mOutput: a n × m matrix V_m

$\beta = \|n_0\|_2$;
$\overrightarrow{v_1} = \overrightarrow{n_0}/\beta$;
For j = 1 to m
  $\vec{p} = \mathrm{tA}\overrightarrow{v_j}$;
  For i = i to j
    $h_{ij} = \overrightarrow{v_i}^T \vec{p}$;
    $\vec{p} = \vec{p} - h_{ij}\overrightarrow{v_i}$;
  End
    $h_{ij} = \|\vec{p}\|_2$;
    $\overrightarrow{v_{j+1}} = \vec{p}/h_{j+1,j}$;
End

With Algorithm 1, a n × m matrix V_m is calculated and it satisfies the following equation (Yamamoto et al., 2007):

$$AV_m = V_m H_m + v_{m+1} h_{m+1,m} e_m^T; \tag{9}$$

where H_m is an m × m upper-Hessenberg matrix; $\overrightarrow{e_m} = [0, 0, \ldots, 0, 1]^T \in \mathbb{R}^m$. By multiplying V_m^T at the both sides of the Eq. (9), the equation becomes:

$$H_m = V_m^T A V_m; \tag{10}$$

In Eq. (4), we notes that: $n_0 = \beta\overrightarrow{v_1} = \beta V_m \overrightarrow{e_1}$ and t is a given constant. Therefore :

$$\begin{aligned} n_t &= \exp(At) * n_0 \\ &= (V_m V_m^T) * \exp(At) * (V_m V_m^T) * n_0 \\ &= V_m * \exp(V_m^T A t V_m) * V_m^T * n_0 \\ &= \beta V_m * \exp(H_m t) * \overrightarrow{e_1} \\ &= \beta V_m * \mathcal{R}_\nu(H_m t) * \overrightarrow{e_1} \end{aligned} \tag{11}$$

In this case, a matrix exponential of m × m instead of n × n needs to be calculated, which will greatly improve its computational efficiency since m≪n. KSM is not used to solve matrix exponential directly and it needs to couple with a rational approximation, such as PRAM and CRAM, which means to solve $\mathcal{R}_\nu(H_m t)$ with a rational approximation. However, this kind of approximation has serious computational problems due to the stiffness of the burnup matrix. The large error consists of two parts (Lopez and Simoncini, 2006). The first one comes from using rational approximation to approximate $H_m t$ and the other comes