NUCLEAR SCIENCE AND ENGINEERING: **179**, 42–58 (2015)

Preconditioned Multigroup GMRES Algorithms for the Variational Nodal Method

Yunzhao Li*

Xi'an Jiaotong University, School of Nuclear Science and Technology
28 West Xianning Road, Xi'an, Shaanxi 710049, China

E. E. Lewis

Northwestern University, Department of Mechanical Engineering
2145 Sheridan Road, Evanston, Illinois 60208

Micheal A. Smith

Argonne National Laboratory, Nuclear Engineering Division
9700 South Cass Avenue, Argonne, Illinois 60439

and

Hongchun Wu and Liangzhi Cao

Xi'an Jiaotong University, School of Nuclear Science and Technology
28 West Xianning Road, Xi'an, Shaanxi 710049, China

Received December 4, 2013
Accepted March 24, 2014
http://dx.doi.org/10.13182/NSE13-103

Abstract—*Combinations of three approaches are examined as options to replace the algorithms presently employed in the variational nodal code VARIANT. They are preconditioned Generalized Minimal Residual (GMRES) algorithms, parallelism in energy, and Wielandt acceleration. Together with partitioned matrix and Gauss-Seidel (GS) preconditioners, two GMRES algorithms are formulated to replace the upscattering iteration and facilitate energy parallelism and Wielandt acceleration. The GMRES algorithms are tested on two-dimensional thermal and fast reactor diffusion problems. The two GMRES algorithms yield higher efficiencies in energy group parallelization and Wielandt acceleration than simple parallelization of the existing GS algorithm. With preconditioning the GMRES algorithms reduce the total computing time by a factor of 2 to 4 and in some cases by a factor of* >10.

A multilevel iteration optimization scheme is investigated that automatically adjusts the relative error tolerance of the inner iterations according to the estimated convergence rate of the corresponding outer iterations and updates the Wielandt shift magnitude as the calculations progress. Numerical results based on large two-dimensional thermal and fast reactor diffusion problems demonstrate that automated optimization of the multilevel iterative processes reduces iteration numbers by as much as an order of magnitude.

*E-mail: yunzhao@mail.xjtu.edu.cn

I. INTRODUCTION

The VARIANT code, developed at Argonne National Laboratory[1–3] (ANL), is a production code based upon the multigroup (MG) even-parity transport equation employed for reactor core neutron calculations. VARIANT uses the Variational Nodal Method (VNM), which forms response matrices based upon orthogonal spatial and angular (spherical harmonics) basis functions. Since VARIANT was developed in the mid-1990s, a number of advances in computing technology and code enhancements have allowed VNM to be used more routinely for reactor problems with refined treatments in energy, space, and angle. In this work we investigate further enhancements including energy parallelism, Wielandt acceleration, and preconditioned Krylov methods. In order to examine several combinations of these techniques, we limit the scope of the present study to two-dimensional diffusion problems, assuming that the most effective combination can be extended later to include spherical harmonics transport calculations in three dimensions.

Similar to most other methods for performing reactor criticality calculations, VNM employs three levels of iteration. The outermost is the fission source (FS) iteration (typically termed the "outers" in the literature), which is based upon the Power Method.[1] The convergence of the Power Method is determined by the problem's dominance ratio, defined as the ratio of the second largest eigenvalue to the largest (fundamental) eigenvalue. A dominance ratio close to 1 exhibits very slow convergence in the FS iteration. At each FS iteration, the MG flux system must be solved with both upscattering as well as downscattering included. In VARIANT (Ref. 2), a Gauss-Seidel (GS) algorithm is used to solve the MG flux system as the scattering matrix is downscattering dominated. To handle upscattering (in thermal reactor systems), a fixed iteration scheme is deployed where the user must select the number of iterations to use in each FS iteration. Generally speaking, the stronger the upscattering is, the larger the number of required iterations becomes. Also as a part of the GS iteration, the diagonal must be inverted, which translates to the solution of the within-group (WG) even-parity transport equation with a fixed source. In VNM, the bulk of effort is expended in the solution of the response matrix equations via a Red-Black Gauss-Seidel (RBGS) algorithm,[2] which we term WG iteration (typically termed the "inners" in the literature).

In nuclear reactor simulations, the desire to improve accuracy has generally led to the usage of more energy groups, which involves more computational effort. As an example, reactor analysis in the 1990s was focused primarily on using coarse group cross-section sets: 2 to 4 groups for thermal reactors and 9 to 21 groups for fast reactors. Today, one finds research groups are pushing toward energy group structures with more than 49 groups for thermal reactors and 70 groups for fast reactors. With the existing solution algorithms, using more energy groups increases the main memory requirements and can slow the convergence of the iterative methods. These performance degradations, however, may be mitigated by considering parallelization with respect to energy. In this work, we consider parallelization using message passing interface[4] (MPI) and replace the serial GS algorithm with a block GS (or block Jacobi) iteration over the locally assigned set of energy groups. Considering the fact that the dominance ratios in nuclear reactor core neutronics simulations frequently exceed 0.9 and sometimes are >0.95 (Ref. 5), Wielandt acceleration[6] is selected to accelerate FS iterations because the Chebychev polynomial extrapolation method[1,7] tends to become ineffective for high-dominance-ratio problems. However, Wielandt acceleration introduces strong pseudo upscattering over the entire energy spectrum, slowing the convergence of the classical MG GS algorithm.

Our earlier work[8] demonstrated the value of the Generalized Minimal Residual (GMRES) algorithm[9] when solving the WG VNM response matrix equations. Here, we extend the use of the GMRES algorithm to the MG flux system. This permits a more effective treatment of both the pseudo upscattering introduced by Wielandt acceleration and the energy parallel block GS and Jacobi algorithms than is possible with the legacy GS approach. Two MG GMRES algorithms were investigated: Current-Flux (CF) GMRES, which includes surface currents and volumetric flux moments in the solution vector, and Residual Flux (RF) GMRES, in which the currents are eliminated so that the solution vector contains only the volumetric flux moments. The orthogonality of the spatial basis polynomials in VNM guarantees the preservation of neutron conservation in each expansion order, thus allowing right partitioned matrix (*Rpm*) preconditioning to the two GMRES algorithms. Alternatively, the RF GMRES algorithm allows inverting a lower triangular matrix using only one GS sweep to provide left Gauss-Seidel (*Lgs*) preconditioning. Somewhat analogous to our application of GMRES is the work of Slaybaugh,[10] who applied GMRES to the Denovo discrete ordinates code[11] to handle the parallelization with respect to energy.

Because there are three levels of iteration in the VNM numerical computing process, namely, the FS iteration, the MG iteration, and the WG iteration, unnecessary calculations occur if the inner iterations are converged extremely tightly. Conversely, the convergence of the outer iterations may deteriorate if the inner iterations are converged too loosely. Thus, optimal iteration convergence criteria for the inner iterations are desirable to minimize the total computational effort. However, optimized settings may vary from problem to problem. We present an optimized iteration scheme to replace fixed numbers of iterations[5,12] in VARIANT (Ref. 2). The new approach takes the problem dominance ratio into account and also updates the Wielandt shift automatically as the FS iteration progress.

The contents of the following four sections are as follows. In Sec. II, we reformulate VNM in matrix-vector form, introduce energy group parallelization with a Wielandt acceleration scheme, and present the details of the GMRES algorithms. In Sec. III, the resulting algorithms are applied to thermal and fast spectrum reactor problems. Section IV presents results for iteration optimization and the self-adjusted Wielandt shift techniques and applies the methods to the thermal and fast reactor problems introduced in Sec. III. Conclusions drawn from this work are discussed in Sec. V.

II. FORMULATION

For each energy group the VNM discretization of the neutron diffusion equation yields three equations: WG source, surface current response matrix, and volumetric flux reconstruction. Defining a vector containing all partial current moments from all nodal surfaces $\boldsymbol{j}_g$ ($\mathrm{cm}^{-2}\cdot\mathrm{s}^{-1}$), a vector of volumetric fluxes $\boldsymbol{\varphi}_g$ ($\mathrm{cm}^{-2}\cdot\mathrm{s}^{-1}$), and a vector containing all volumetric sources $\boldsymbol{s}_g$ ($\mathrm{cm}^{-3}\cdot\mathrm{s}^{-1}$), the source equation can be written as

$$\boldsymbol{s}_g = \sum_{g'\neq g} \Sigma_{gg'}\boldsymbol{\varphi}_{g'} + \frac{1}{k}\sum_{g'} F_{gg'}\boldsymbol{\varphi}_{g'} \; ; \tag{1}$$

the WG surface current response matrix equation is written as

$$\left(\boldsymbol{I}_j - \boldsymbol{R}_g\boldsymbol{\Pi}\right)\boldsymbol{j}_g = \boldsymbol{B}_g\boldsymbol{s}_g \; ; \tag{2}$$

the volumetric flux reconstruction equation is given as

$$\boldsymbol{\varphi}_g = \boldsymbol{H}_g\boldsymbol{s}_g - \boldsymbol{C}_g\left(\boldsymbol{I}_j - \boldsymbol{\Pi}\right)\boldsymbol{j}_g \; , \tag{3}$$

where

k = multiplication factor of the system

$\Sigma_{gg'}, F_{gg'}$ = macroscopic scattering and fission cross sections from group g' to group g (cm^{-1}), respectively

$\boldsymbol{B}_g, \boldsymbol{C}_g, \boldsymbol{H}_g, \boldsymbol{R}_g$ = matrices determined by the nodal dimensions and macroscopic cross sections, as detailed elsewhere[2]

$\boldsymbol{I}_j, \boldsymbol{\Pi}$ = identity and spatial connectivity matrices, respectively.

Substituting the source equation into the current and flux equations yields

$$\left(\boldsymbol{I}_j - \boldsymbol{R}_g\boldsymbol{\Pi}\right)\boldsymbol{j}_g - \boldsymbol{B}_g\sum_{g'\neq g}\Sigma_{gg'}\boldsymbol{\varphi}_{g'} = \boldsymbol{B}_g\frac{1}{k}\sum_{g'}F_{gg'}\boldsymbol{\varphi}_{g'} \tag{4}$$

and

$$\boldsymbol{\varphi}_g - \boldsymbol{H}_g\sum_{g'\neq g}\Sigma_{gg'}\boldsymbol{\varphi}_{g'} + \boldsymbol{C}_g\left(\boldsymbol{I}_j - \boldsymbol{\Pi}\right)\boldsymbol{j}_g = \boldsymbol{H}_g\frac{1}{k}\sum_{g'}F_{gg'}\boldsymbol{\varphi}_{g'} \; . \tag{5}$$

Combining all the energy groups into a single system yields the VNM eigenvalue system:

$$\begin{bmatrix} \boldsymbol{I}_j - \boldsymbol{R}\boldsymbol{\Pi} & \boldsymbol{B}\boldsymbol{\Sigma} \\ \boldsymbol{C}\left(\boldsymbol{I}_j - \boldsymbol{\Pi}\right) & \boldsymbol{I}_\varphi - \boldsymbol{H}\boldsymbol{\Sigma} \end{bmatrix}\begin{bmatrix} \boldsymbol{j} \\ \boldsymbol{\varphi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{H} \end{bmatrix}\boldsymbol{f} \; , \tag{6}$$

where

$$\boldsymbol{f} = \frac{1}{k}\boldsymbol{F}\boldsymbol{\varphi} \; . \tag{7}$$

The scattering matrix $\boldsymbol{\Sigma}$ is the assembled form of $\Sigma_{gg'}$ ($g \neq g'$) while $\boldsymbol{F}$ is the assembled form of $F_{gg'}$. The matrices $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{H}$, and $\boldsymbol{R}$ are block diagonal over the energy groups, and $\boldsymbol{I}_\varphi$ is an identity matrix with the same dimension as the assembled flux vector $\boldsymbol{\varphi}$.

Eliminating currents from Eq. (6) yields a second form of the VNM eigenvalue system where only the flux moments are contained in the solution vector:

$$\left(\boldsymbol{I}_\varphi - \boldsymbol{T}\boldsymbol{\Sigma}\right)\boldsymbol{\varphi} = \boldsymbol{T}\boldsymbol{f} \; , \tag{8}$$

where the spatial transport coefficient matrix

$$\boldsymbol{T} = \boldsymbol{H} - \boldsymbol{C}\left(\boldsymbol{I}_j - \boldsymbol{\Pi}\right)\left(\boldsymbol{I}_j - \boldsymbol{R}\boldsymbol{\Pi}\right)^{-1}\boldsymbol{B} \; , \tag{9}$$

is also block diagonal over the energy groups.

Traditionally, the Power Method,[1,7] which we refer to as FS iteration, is employed to solve the eigenvalue problem. Within each FS iteration, the neutron flux must be updated by solving for the MG flux and thus inverting the coefficient matrix on the left side of Eq. (6) or Eq. (8).

After reformulating the VNM into a matrix-vector form, Secs. II.A, II.B, and II.C successively describe the legacy MG GS algorithm, Wielandt acceleration, and energy group parallelization techniques. The two MG GMRES algorithms are formulated in Secs. II.D and II.E, respectively, together with their preconditioners.

II.A. Multigroup GS Algorithm

The GS algorithm separates the scattering matrix into downscatter and upscatter components

$$\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_\mathrm{D} + \boldsymbol{\Sigma}_\mathrm{U} \; , \tag{10}$$

and, using Eq. (8) as the equation to solve, defines the iterative system

$$\boldsymbol{\varphi}^{(m)} = \boldsymbol{T}\left(\boldsymbol{\Sigma}_{\mathrm{D}}\boldsymbol{\varphi}^{(m)} + \boldsymbol{\Sigma}_{\mathrm{U}}\boldsymbol{\varphi}^{(m-1)} + \boldsymbol{f}\right) , \qquad (11)$$

where m indexes the MG iteration number and $\boldsymbol{f}$ is the fixed FS. Solving for $\boldsymbol{\varphi}^{(m)}$ leads to

$$\boldsymbol{\varphi}^{(m)} = \left(\boldsymbol{I}_{\varphi} - \boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{D}}\right)^{-1}\boldsymbol{T}\left(\boldsymbol{\Sigma}_{\mathrm{U}}\boldsymbol{\varphi}^{(m-1)} + \boldsymbol{f}\right) . \qquad (12)$$

This algorithm is very effective since the scattering of most reactors is downscatter dominant; thus, only a few iterations are needed to converge Eq. (12). Given that more than one iteration is needed only when upscattering is present, the GS iterations are often termed upscatter or thermal iterations in the literature.

II.B. Wielandt Acceleration

Wielandt acceleration[6] is used to improve convergence of problems with particularly high dominance ratios. It accomplishes this by splitting the FS and moving part of it to the left side to be treated similarly to the scattering. Thus, instead of solving Eq. (8), Eq. (13) with a Wielandt shift of k' is solved:

$$\left[\boldsymbol{I}_{\varphi} - \boldsymbol{T}\left(\boldsymbol{\Sigma} + \frac{1}{k'}\boldsymbol{F}\right)\right]\boldsymbol{\varphi} = \boldsymbol{T}\left(\frac{1}{k} - \frac{1}{k'}\right)\boldsymbol{F}\boldsymbol{\varphi} . \qquad (13)$$

Defining k_1 and k_2 as the fundamental and the second largest eigenvalue of the system, respectively, the dominance ratio before Wielandt shift is $d = k_2/k_1$. With the Wielandt shift k', the dominance ratio becomes

$$d' = \frac{k_1^{-1} - k'^{-1}}{k_2^{-1} - k'^{-1}} , \qquad (14)$$

which is smaller than the original value, given reasonable values of k' $(> k_1)$.

While Wielandt acceleration can reduce the total number of FS iterations, the MG flux solution becomes more expensive. To explain, we can define a pseudo scattering system as

$$\boldsymbol{\Sigma}' = \boldsymbol{\Sigma} + \frac{1}{k'}\boldsymbol{F} . \qquad (15)$$

Because fission reaction occurs at all energy groups but generates only fast neutrons, fission adds a very strong upscattering component; the conventional GS iteration is degraded considerably as more iterations deal with the pseudo upscattering that has been introduced.

II.C. Energy Group Parallelization

Parallelization in energy using MPI requires parallelizing the construction of the scattering and FSs. For parallelization with N processors, the fission/scattering matrix is divided into $N \times N$ blocks with sequential assignment of energy groups for each block. In this scheme, we let (n, n') represent the contribution for the energy groups on processor n' to the energy groups on processor n, which leads to exactly $N(N - 1)$ MPI communication events to update the contribution on all processors.

The scattering source must be updated during the GS sweep over energy groups, and parallelization introduces latency due to the use of block GS or even block Jacobi iteration. To understand the changes to GS in Eqs. (11) and (12), in a parallel implementation, $\boldsymbol{\Sigma}_{\mathrm{D}}$ consists of the downscattering within each processor while $\boldsymbol{\Sigma}_{\mathrm{U}}$ is altered to contain all of the upscattering and all downscattering between different processors. In the worst case, the block GS algorithm degrades to Jacobi iteration when $N = G$.

II.D. Multigroup CF GMRES Algorithm

GMRES (Ref. 9) is a Krylov subspace iterative method[13] for the numerical solution of a nonsymmetric system of linear equations. It has been widely used in parallel algorithms to overcome the typical performance degradation of classical iterative schemes. For a given FS $\boldsymbol{f}$, the focus of our first GMRES-based approach is to combine the current and flux into a single vector space as seen in Eq. (6). Compared with the GS algorithm, the CF GMRES: (a) solves all the energy groups simultaneously, (b) is scalable in energy due to the straightforward division of work, and (c) is identical in matrix storage requirements to GS.

II.D.1. *Rpm* Preconditioning

A preferred preconditioner for CF GMRES should closely resemble the inverse of the coefficient matrix of the original system. Because of the hierarchical system of orthogonal polynomials in space, a low-order system meets the basic requirement, which we term the *Rpm* preconditioner.

We partition the solution vector as

$$\boldsymbol{x} = \left[\boldsymbol{j}^{\alpha} \quad \boldsymbol{\varphi}^{l} \quad \boldsymbol{j}^{\beta} \quad \boldsymbol{\varphi}^{h}\right]^{\mathrm{T}} , \qquad (16)$$

where α and β denote the low- and high-order current moments and l and h denote the low- and high-order flux moments. The coefficient matrix in Eq. (6) can be partitioned as

$$\boldsymbol{A}=\begin{bmatrix} \boldsymbol{I}_j^{\alpha\alpha}-\boldsymbol{R}^{\alpha\alpha}\boldsymbol{\Pi}^{\alpha\alpha} & -\boldsymbol{B}^{\alpha l}\boldsymbol{\Sigma}^{ll} & -\boldsymbol{R}^{\alpha\beta}\boldsymbol{\Pi}^{\alpha\beta} & -\boldsymbol{B}^{\alpha h}\boldsymbol{\Sigma}^{hh} \\ \boldsymbol{C}^{l\alpha}\left(\boldsymbol{I}_j^{\alpha\alpha}-\boldsymbol{\Pi}^{\alpha\alpha}\right) & \boldsymbol{I}_{\varphi}^{ll}-\boldsymbol{H}^{ll}\boldsymbol{\Sigma}^{ll} & \boldsymbol{C}^{l\beta}\left(\boldsymbol{I}_j^{\beta\beta}-\boldsymbol{\Pi}^{\beta\beta}\right) & -\boldsymbol{H}^{lh}\boldsymbol{\Sigma}^{hh} \\ -\boldsymbol{R}^{\beta\alpha}\boldsymbol{\Pi}^{\beta\alpha} & -\boldsymbol{B}^{\beta l}\boldsymbol{\Sigma}^{ll} & \boldsymbol{I}_j^{\beta\beta}-\boldsymbol{R}^{\beta\beta}\boldsymbol{\Pi}^{\beta\beta} & -\boldsymbol{B}^{\beta h}\boldsymbol{\Sigma}^{hh} \\ \boldsymbol{C}^{h\alpha}\left(\boldsymbol{I}_j^{\alpha\alpha}-\boldsymbol{\Pi}^{\alpha\alpha}\right) & -\boldsymbol{H}^{hl}\boldsymbol{\Sigma}^{ll} & \boldsymbol{C}^{h\beta}\left(\boldsymbol{I}_j^{\beta\beta}-\boldsymbol{\Pi}^{\beta\beta}\right) & \boldsymbol{I}_{\varphi}^{hh}-\boldsymbol{H}^{hh}\boldsymbol{\Sigma}^{hh} \end{bmatrix} . \tag{17}$$

Eliminating the entries relating to higher-order moments yields the following partitioned matrix (*pm*) approximation of $\boldsymbol{A}$:

$$\tilde{\boldsymbol{A}}=\begin{bmatrix} \boldsymbol{I}_j^{\alpha\alpha}-\boldsymbol{R}^{\alpha\alpha}\boldsymbol{\Pi}^{\alpha\alpha} & -\boldsymbol{B}^{\alpha l}\boldsymbol{\Sigma}^{ll} & & \\ \boldsymbol{C}^{l\alpha}\left(\boldsymbol{I}_j^{\alpha\alpha}-\boldsymbol{\Pi}^{\alpha\alpha}\right) & \boldsymbol{I}_{\varphi}^{ll}-\boldsymbol{H}^{ll}\boldsymbol{\Sigma}^{ll} & & \\ & & \boldsymbol{I}_j^{\beta\beta} & \\ & & & \boldsymbol{I}_{\varphi}^{hh} \end{bmatrix} . \tag{18}$$

We have implemented and tested five *pm* preconditioning schemes[14] that include its use as a left or right preconditioner in GMRES. In each case, the matrix $\tilde{\boldsymbol{A}}$ is approximately inverted with its own GMRES algorithm or with the conventional GS algorithm. We note that the convergence criteria for the inversion of the preconditioning system were tested using a fixed iteration number and a relative error tolerance (RET). Based upon the results of our earlier work,[14] the right preconditioning was selected with the flexible GMRES algorithm:

$$\boldsymbol{A}\tilde{\boldsymbol{A}}^{-1}\begin{bmatrix}\boldsymbol{j}_y\\ \boldsymbol{\varphi}_y\end{bmatrix}=\begin{bmatrix}\boldsymbol{B}\\ \boldsymbol{H}\end{bmatrix}\boldsymbol{f} , \tag{19}$$

where the preconditioning system

$$\tilde{\boldsymbol{A}}\begin{bmatrix}\boldsymbol{j}_x\\ \boldsymbol{\varphi}_x\end{bmatrix}=\begin{bmatrix}\boldsymbol{j}_y\\ \boldsymbol{\varphi}_y\end{bmatrix} , \tag{20}$$

is solved with a single GS sweep at the MG level combined with a series of RBGS iterations at the WG level terminated by a pre-fixed RET.

II.E. Multigroup RF GMRES Algorithm

The next GMRES algorithm we tested has only the flux moments in the solution vector (termed RF for residual flux) and thus solves Eq. (8) using a residual form. To explain, we begin with initial guesses $\boldsymbol{\varphi}^{(0)}$ and $\boldsymbol{j}^{(0)}$ whereby the next iterative update may be obtained using

$$\boldsymbol{j}^{update}=(\boldsymbol{I}_j-\boldsymbol{R}\boldsymbol{\Pi})^{-1}\boldsymbol{B}(\boldsymbol{\Sigma}\boldsymbol{\varphi}^{(0)}+\boldsymbol{f}) \tag{21}$$

and

$$\begin{aligned}\boldsymbol{\varphi}^{\text{update}}&=\boldsymbol{H}(\boldsymbol{\Sigma}\boldsymbol{\varphi}^{(0)}+\boldsymbol{f})-\boldsymbol{C}(\boldsymbol{I}_j-\boldsymbol{\Pi})\boldsymbol{j}^{\text{update}}\\ &=\boldsymbol{T}(\boldsymbol{\Sigma}\boldsymbol{\varphi}^{(0)}+\boldsymbol{f}) ,\end{aligned} \tag{22}$$

where $\boldsymbol{\varphi}^{\text{update}}$ is used to obtain the residual

$$\boldsymbol{\varepsilon}=\boldsymbol{\varphi}^{\text{update}}-\boldsymbol{\varphi}^{(0)}=\boldsymbol{T}\boldsymbol{f}-(\boldsymbol{I}_{\varphi}-\boldsymbol{T}\boldsymbol{\Sigma})\boldsymbol{\varphi}^{(0)} . \tag{23}$$

From here, we define the error associated with the initial flux solution:

$$\Delta\boldsymbol{\varphi}=\boldsymbol{\varphi}^{\text{exact}}-\boldsymbol{\varphi}^{(0)} , \tag{24}$$

where $\boldsymbol{\varphi}^{\text{exact}}$ stands for the exact flux solution. We can then solve the alternate form of Eq. (8) given as

$$(\boldsymbol{I}_{\varphi}-\boldsymbol{T}\boldsymbol{\Sigma})\Delta\boldsymbol{\varphi}=\boldsymbol{\varepsilon} . \tag{25}$$

This system has the same form as the original system, but the source term is much simpler. Another advantage of this approach is that the updated current solution is obtained for the next FS iteration during the initial flux solution step. This is accomplished by applying the $\boldsymbol{T}$ matrix only once, instead of twice as would be required if GMRES were applied directly to Eq. (8). Finally, this approach also facilitates the use of a zero initial guess for the current moments during the application of the coefficient matrix within each GMRES iteration.

II.E.1. *Rpm* Preconditioning

Similar to the CF GMRES algorithm, the RF GMRES algorithm can be preconditioned with its *Rpm* preconditioner

$$\boldsymbol{I}_{\varphi}-\tilde{\boldsymbol{T}}\boldsymbol{\Sigma} , \tag{26}$$

in which $\tilde{\boldsymbol{T}}$ is obtained from $\boldsymbol{T}$ in Eq. (9) with the same partitioning that $\tilde{\boldsymbol{A}}$ in Eq. (18) is obtained from $\boldsymbol{A}$:

$$\tilde{\boldsymbol{T}}=\begin{bmatrix}\boldsymbol{H}^{ll}-\boldsymbol{C}^{l\alpha}(\boldsymbol{I}_j-\boldsymbol{\Pi})(\boldsymbol{I}_j-\boldsymbol{R}^{\alpha\alpha}\boldsymbol{\Pi})^{-1}\boldsymbol{B}^{\alpha l} & \\ & \boldsymbol{I}_{\varphi}^{hh}\end{bmatrix} . \tag{27}$$

In this case the *Rpm* preconditioned system is

$$\left(\boldsymbol{I}_{\varphi}-\boldsymbol{T\Sigma}\right)\left(\boldsymbol{I}_{\boldsymbol{\varphi}}-\tilde{\boldsymbol{T}}\boldsymbol{\Sigma}\right)^{-1}\boldsymbol{\varphi}_y=\boldsymbol{Tf}\ \left(\boldsymbol{I}_{\boldsymbol{\varphi}}-\tilde{\boldsymbol{T}}\boldsymbol{\Sigma}\right)\boldsymbol{\varphi}_x=\boldsymbol{\varphi}_y\ , \quad (28)$$

where the *Rpm* preconditioning system

$$(\boldsymbol{I}_{\boldsymbol{\varphi}}-\tilde{\boldsymbol{T}}\boldsymbol{\Sigma})\boldsymbol{\varphi}_x=\boldsymbol{\varphi}_y \quad (29)$$

is solved using a GS in energy and RBGS algorithm on each energy group. In this way, we can reformulate Eq. (29) given an initial guess $\boldsymbol{\varphi}_x^{(0)}=\boldsymbol{\varphi}_y$ as

$$\boldsymbol{\varphi}_x^{(n)}=\tilde{\boldsymbol{T}}(\boldsymbol{\Sigma}_{\mathrm{D}}\boldsymbol{\varphi}_x^{(n)}+\boldsymbol{\Sigma}_{\mathrm{U}}\boldsymbol{\varphi}_x^{(n-1)})+\boldsymbol{\varphi}_y\ . \quad (30)$$

Unlike Eq. (12), the "source term" is located outside of the parentheses, which in this case is not difficult to handle.

II.E.2. *Lgs* Preconditioning

In contrast to the *Rpm* preconditioner, *Lgs* preconditioning to the RF GMRES algorithm can be used to take advantage of the effectiveness of the GS sweep over the downscattering matrix. Applying the inverse of the lower triangular matrix $\boldsymbol{I}_{\boldsymbol{\varphi}}-\boldsymbol{T\Sigma}_{\mathrm{D}}$ to both sides of Eq. (8) yields the preconditioned system

$$(\boldsymbol{I}_{\boldsymbol{\varphi}}-\boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{D}})^{-1}(\boldsymbol{I}_{\boldsymbol{\varphi}}-\boldsymbol{T}\boldsymbol{\Sigma})\boldsymbol{\varphi}=(\boldsymbol{I}_{\boldsymbol{\varphi}}-\boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{D}})^{-1}\boldsymbol{Tf}\ , \quad (31)$$

which can also be written as

$$[\boldsymbol{I}_{\boldsymbol{\varphi}}-(\boldsymbol{I}_{\boldsymbol{\varphi}}-\boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{D}})^{-1}\boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{U}}]\boldsymbol{\varphi}=(\boldsymbol{I}_{\boldsymbol{\varphi}}-\boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{D}})^{-1}\boldsymbol{Tf}\ . \quad (32)$$

Again, a residual form may be adopted where with the initial guess $\boldsymbol{\varphi}^{(0)}$, an updated solution is given as

$$\boldsymbol{\varphi}^{\mathrm{update}}=(\boldsymbol{I}_{\boldsymbol{\varphi}}-\boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{D}})^{-1}\boldsymbol{T}(\boldsymbol{\Sigma}_{\mathrm{U}}\boldsymbol{\varphi}^{(0)}+\boldsymbol{f})\ . \quad (33)$$

Note that $\boldsymbol{\varphi}^{\mathrm{update}}$ is the exact solution $\boldsymbol{\varphi}^{\mathrm{exact}}$ when there is no upscattering ($\boldsymbol{\Sigma}_{\mathrm{U}}=0$). If $\boldsymbol{\varphi}^{\mathrm{update}}\neq\boldsymbol{\varphi}^{\mathrm{exact}}$, the flux $\boldsymbol{\varphi}^{\mathrm{update}}$ is used to obtain the initial residual:

$$\boldsymbol{\varepsilon}=\boldsymbol{\varphi}^{\mathrm{update}}-\boldsymbol{\varphi}^{(0)}=(\boldsymbol{I}_{\boldsymbol{\varphi}}-\boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{D}})^{-1}\boldsymbol{Tf}-[\boldsymbol{I}_{\boldsymbol{\varphi}}-(\boldsymbol{I}_{\boldsymbol{\varphi}}-\boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{D}})^{-1}\times\boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{U}}]\boldsymbol{\varphi}^{(0)}\ . \quad (34)$$

After defining the error associated with the initial flux solution, as in Eq. (24), we obtain

$$[\boldsymbol{I}_{\boldsymbol{\varphi}}-(\boldsymbol{I}_{\boldsymbol{\varphi}}-\boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{D}})^{-1}\boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{U}}]\Delta\boldsymbol{\varphi}=\boldsymbol{\varepsilon}\ . \quad (35)$$

Because $\boldsymbol{\varphi}_g^{\mathrm{update}}=\boldsymbol{\varphi}_g^{\mathrm{exact}}$ for the downscatter-only groups, the residual system can be restricted to just those groups with upscatter, yielding reduced computational effort. However, such an approach is not consistent with Wielandt acceleration and energy group parallelization and thus is not pursued here.

Application of the coefficient matrix $\boldsymbol{I}_{\boldsymbol{\varphi}}-(\boldsymbol{I}_{\boldsymbol{\varphi}}-\boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{D}})^{-1}\boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{U}}$ to the flux vector $\boldsymbol{\varphi}_x$ yields an intermediate vector $\boldsymbol{\varphi}_y$:

$$\boldsymbol{\varphi}_y=[\boldsymbol{I}_{\boldsymbol{\varphi}}-(\boldsymbol{I}_{\boldsymbol{\varphi}}-\boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{D}})^{-1}\boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{U}}]\boldsymbol{\varphi}_x=\boldsymbol{\varphi}_x-\boldsymbol{\varphi}_z\ , \quad (36)$$

where

$$\boldsymbol{\varphi}_z=(\boldsymbol{I}_{\boldsymbol{\varphi}}-\boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{D}})^{-1}\boldsymbol{T}\boldsymbol{\Sigma}_{\mathrm{U}}\boldsymbol{\varphi}_x=\boldsymbol{T}(\boldsymbol{\Sigma}_{\mathrm{D}}\boldsymbol{\varphi}_z+\boldsymbol{\Sigma}_{\mathrm{U}}\boldsymbol{\varphi}_x)\ . \quad (37)$$

Only a single GS sweep over the energy groups is needed to obtain the flux vector $\boldsymbol{\varphi}_z$. Comparing the applications of the coefficient matrices of *Lgs* preconditioned and unpreconditioned RF GMRES indicates that the only difference lies in the GS sweeping in Eq. (37) instead of a Jacobi sweeping. Thus, the computing efforts required applying the coefficient matrices are effectively the same.

III. RESULTS

The evaluations of the above algorithms have been performed at ANL as part of the NODAL code development, which is an option in the UNIC (Ultimate Neutronic Investigation Code) software package.[15] When possible, the production VARIANT code[2] was used as a basis for comparison. All of the calculations were performed using a 64-bit x86 Linux workstation consisting of four 2-GHz Intel Xeon processors with six cores each and 132 Gbytes of aggregate memory deployed in a shared memory architecture.

The two-dimensional test problems shown in Fig. 1 (thermal spectrum) and in Fig. 2 (fast spectrum) were chosen to evaluate the algorithms developed in this work. Table I gives the overview data for the two problems, where the detailed material compositions are available elsewhere.[16] The quarter-core thermal reactor problem has three unique compositions with 117 total assemblies where the Cartesian pitch is 11.34 cm. The 72-group cross-section library was obtained from a pressurized water reactor lattice calculation in the DRAGON code,[17] which has a base 172-group library. The fast reactor problem has five different compositions with 136 total assemblies where the hex pitch is 16 cm and the domain has one-sixth symmetry. A 216-group cross-section library was obtained using several lattice calculations with the MC^2-3 code,[18] which has a base 2082-group library. All of the calculations used fuel assembly–sized nodes where the volumetric flux is represented by a sixth-order polynomial expansion and the leakage on each surface is represented as a third-order polynomial. This leads to 28 degrees of freedom (DOFs) per node, and 4 DOFs for each node surface.

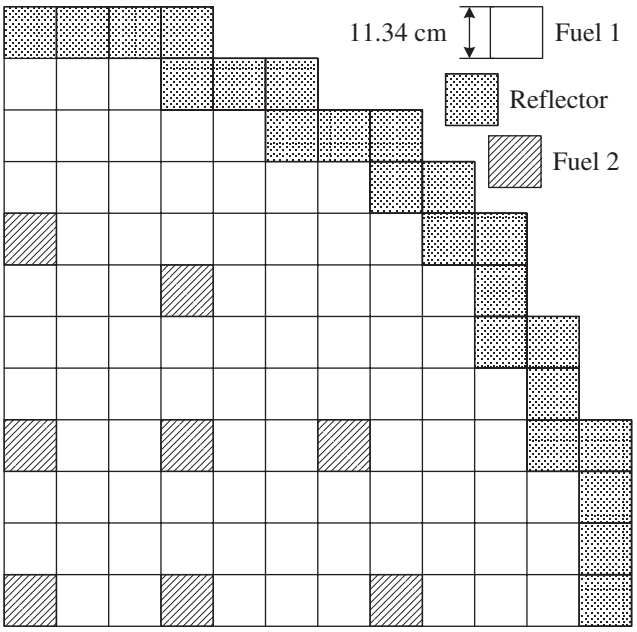

Fig. 1. The configuration for the thermal reactor problem.

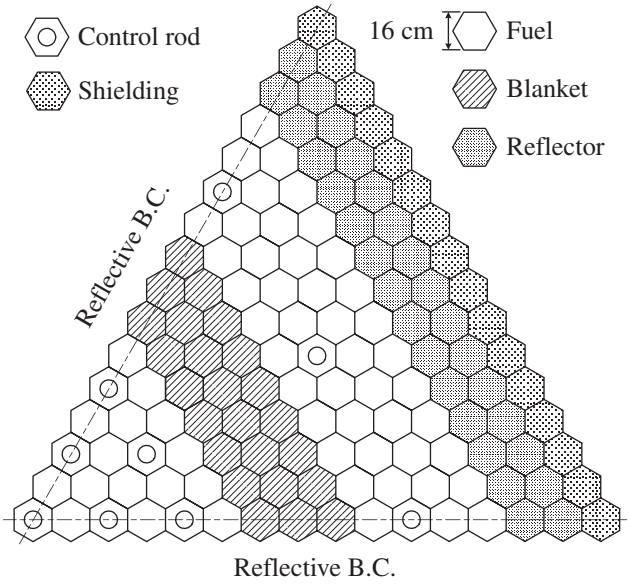

Fig. 2. The configuration for the fast reactor problem.

For both problems, the converge criteria for k_{eff}, FS, and flux were set as $e_k = 1.0 \times 10^{-7}$, $e_f = 1.0 \times 10^{-5}$, and $e_{phi} = 1.0 \times 10^{-5}$, respectively, with the maximum number of FS iterations set to be 200. For the MG iterations within each FS iteration, three convergence criteria were employed: maximum iteration number $M_{\max} = 100$, an absolute error tolerance (AET) for the first iteration $\varepsilon_{\mathrm{MG}} = 1.0 \times 10^{-10}$, and a RET for the remaining iterations $e_{\mathrm{MG}} = 1.0 \times 10^{-3}$. For the WG iteration of each energy group within each MG iteration, different convergence criteria were employed for the two test problems. For the thermal reactor problem, a maximum iteration number $I_g^{\max}$, an AET $\varepsilon_{\mathrm{WG}} = 1.0 \times 10^{-11}$, and a RET $e_{\mathrm{WG}} = 1.0 \times 10^{-3}$ were employed for the partial current update. In contrast, a fixed number of iterations (eight spatial sweeps per group per MG iteration) was employed for the fast reactor problem.

In Sec. III.A, the performance of the conventional GS algorithm is assessed with Wielandt acceleration and energy group parallelization. In Sec. III.B, the various GMRES algorithms outlined above are assessed with Wielandt acceleration and energy group parallelization. In all of the results in Secs. III.A and III.B, the Wielandt acceleration is employed using a constant shift of $k' = k/d$, where k and d were estimated from a preliminary calculation for each of the two test problems. In Sec. IV, a method is detailed for Wielandt acceleration without having preliminary estimates of k and d. With regard to energy group parallelization, only four processors were used unless otherwise stated, and the results were evaluated using the parallel efficiency defined as $\eta_N = T_1/(NT_N)$, where N is the number of processers and T_N is the execution time when N processers are used.

III.A. Multigroup GS Algorithm

The MG GS algorithm was evaluated for four cases on both test problems:

1. serial without Wielandt acceleration
2. serial with Wielandt acceleration
3. parallel without Wielandt acceleration
4. parallel with Wielandt acceleration.

The iteration count and computational time are tabulated for all eight calculations in Table II. In both serial and parallel, the Wielandt acceleration reduces the number of FS iterations for both test problems, but it also increases the number of MG GS iterations within each FS iteration. Energy group parallelization makes use of more computing power, but the block GS iterations degrade the convergence. For the thermal reactor problem, both a Wielandt acceleration and energy group parallelization are found to reduce the total computing time, but for the fast reactor problem, both dramatically increase the computing time.

Figure 3 shows the convergence of the MG GS algorithm at the third FS iteration for the thermal reactor problem. Beginning with the cases without Wielandt acceleration, the serial calculation is observed to converge rapidly at first but then slows to a more gradual convergence trend. The initial rapid convergence is attributable to the dominant downscattering component, which is solved exactly in each sweep, but eventually,

TABLE I

Main Features of the Two Test Problems

Problem	Thermal Reactor Problem	Fast Reactor Problem
Reactor type	Thermal	Fast
Energy groups	72 (23 groups upscattering)	216 groups (no upscattering)
Geometry	Cartesian	Hexagonal
Spatial nodes	117	136
Volume expansion	Sixth order (28 DOFs/node)	Sixth order (28 DOFs/node)
Surface expansion	Third order (4 DOFs/surface)	Third order (4 DOFs/surface)
Volume flux moments	235 872	822 528
Surface current moments	134 784	649 728
Eigenvalue k	1.25306	1.30451
Dominance ratio d	0.935	0.935
Constant Wielandt shift k'	1.34017	1.39520

TABLE II

Base Performance of MG GS Algorithm

Problem	Measurement	Serial	Serial + Wielandt	Parallel	Parallel + Wielandt
Thermal	Fission source iteration	129	17	129	20
	Multigroup iteration	2860	1700	3261	2000
	Time (s)	354	257	168	99
Fast	Fission source iteration	144	18	144	18
	Multigroup iteration	144	1433	720	1800
	Time (s)	208	1961	288	748

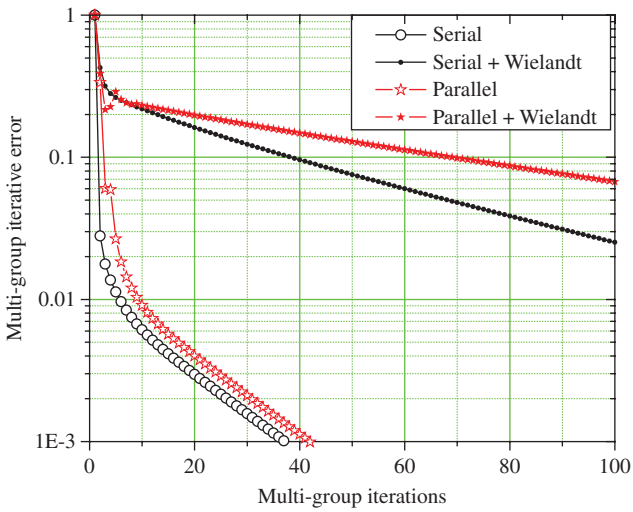

Fig. 3. Typical MG GS iterations of the thermal reactor problem.

the error due to the upscattering begins to limit the accuracy achievable in each iteration. With parallelization, the basic curve shape remains but is offset due to the latency caused by applying block GS. For the Wielandt acceleration cases, the steep drop at the beginning is significantly reduced, and the slope of convergence in the later phase is noticeably slower. Introducing parallelization with the Wielandt acceleration causes additional deterioration in convergence.

Much like Fig. 3, Fig. 4 shows the convergence of the MG GS algorithm at the third FS, but for the fast reactor problem. In this problem, the most distinctive feature is that the serial case without Wielandt acceleration needs only one GS iteration due to the absence of upscattering. Hence, introducing either Wielandt acceleration or block GS can only degrade convergence.

III.B. Multigroup GMRES Algorithms

We next compare the five MG GMRES algorithms listed in Table III with the legacy GS algorithm where we consider three cases:

1. serial without Wielandt acceleration

2. serial with Wielandt acceleration

3. parallel with Wielandt acceleration.

The gains from using parallelization without Wielandt acceleration were insignificant and therefore are not presented.

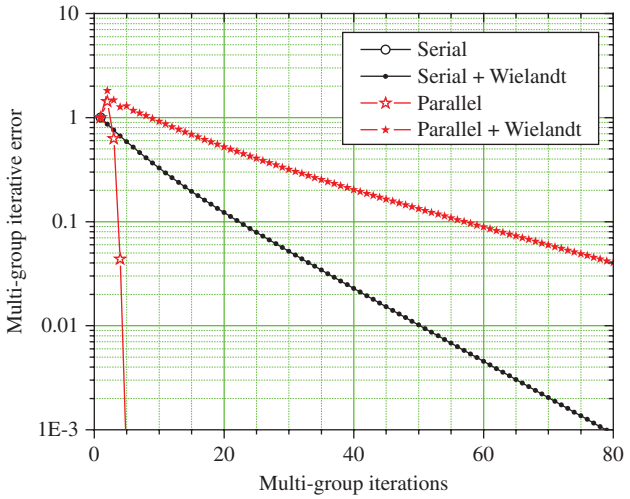

Fig. 4. Typical MG GS iterations of the fast reactor problem.

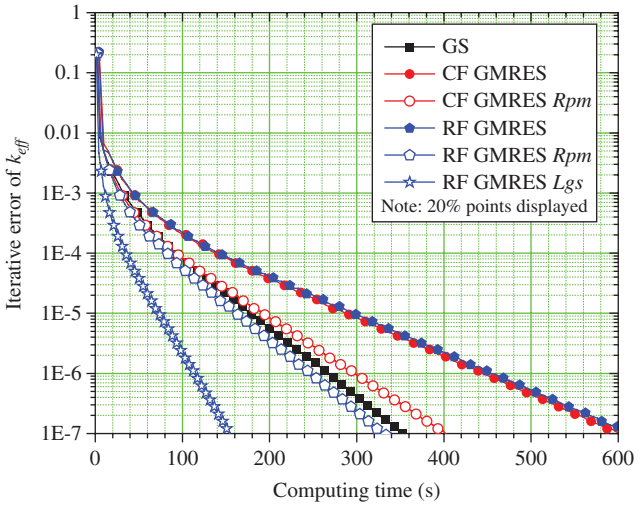

Fig. 5. Fission source convergence for the thermal reactor problem in case 1 (serial without Wielandt acceleration).

TABLE III

The Five MG GMRES Algorithms

Abbreviation	Description
CF GMRES	The unpreconditioned MG CF GMRES algorithm
CF GMRES *Rpm*	The *Rpm* preconditioned MG CF GMRES algorithm
RF GMRES	The unpreconditioned MG RF GMRES algorithm
RF GMRES *Rpm*	The *Rpm* preconditioned MG RF GMRES algorithm
RF GMRES *Lgs*	The *Lgs* preconditioned MG RF GMRES algorithm

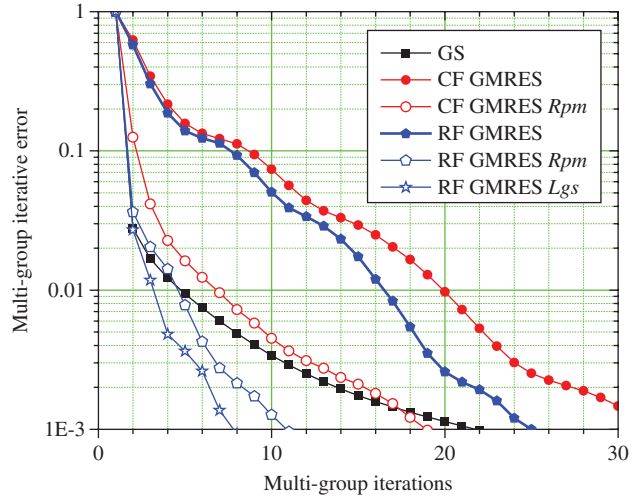

Fig. 6. Typical MG iteration convergence of the thermal reactor problem in case 1 (serial without Wielandt acceleration).

III.B.1. Case 1: Serial Without Wielandt Acceleration

For the thermal reactor problem, all of the serial calculations require ~130 FS iterations (note the dominance ratio). Figure 5 shows the k_{eff} convergence in terms of computing time, while Fig. 6 shows the error of the MG system for a typical fission FS source iteration. The conventional GS algorithm takes 350 s with approximately 22 MG iterations per FS iteration. The two unpreconditioned GMRES algorithms require ~600 s while the *Rpm* preconditioned GMRES algorithms need ~350 s. CF GMRES takes approximately 36 MG iterations per FS iteration while RF GMRES needs approximately 25. The primary reason for this difference is the elimination of the WG solutions in RF GMRES relative to CF GMRES. Similarly, the *Rpm* preconditioned CF and RF GMRES algorithms need 18 and 11 MG iterations per FS iteration, respectively. The *Lgs* preconditioner applied to RF GMRES requires only approximately 8 MG iterations per FS iteration and uses ~150 s.

Moving to the fast reactor problem, all of the serial calculations require 144 FS iterations as seen in Fig. 7. Only one MG iteration is needed per FS iteration; thus, RF GMRES *Lgs* and GS complete the calculation in ~200 s, while the other four options take much more time. CF GMRES and RF GMRES converge fairly slowly because a Jacobi-like scheme is applied to the scattering system. The *Rpm* preconditioning contains a GS sweep that significantly improves the convergence relative to both CF and RF GMRES options, but clearly, it is not as efficient as GS or RF GMRES with *Lgs* preconditioning.

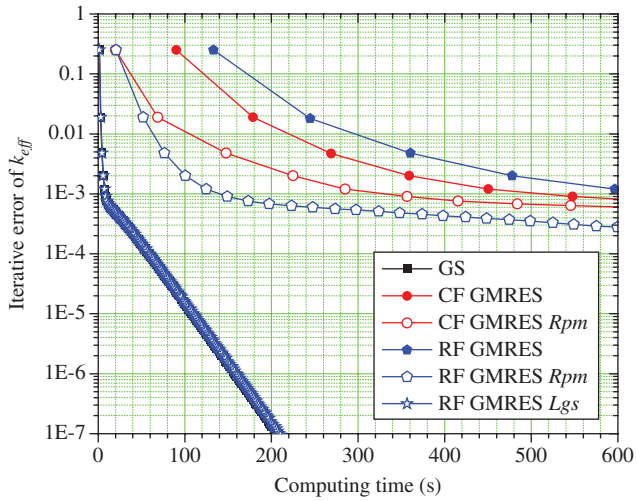

Fig. 7. Fission source iteration convergence of the fast reactor problem in case 1 (serial without Wielandt acceleration).

III.B.2. Case 2: Serial with Wielandt Acceleration

Continuing with the Wielandt accelerated cases of the thermal reactor problem, only 17 FS iterations are needed for convergence. However, the convergence rate of each algorithm is again quite different, due to the effort involved in solving the MG systems, as shown in Fig. 8. In this case, one can clearly see that the RF GMRES approach with *Lgs* precondition is the best while the GS and unpreconditioned GMRES algorithms are the worst. With respect to the MG solve, the GS, unpreconditioned GMRES schemes are terminated by the maximum iteration criteria rather than the targeted RET due to a deterioration in convergence (upscattering problems). Using *Rpm* preconditioning clearly improves the performance of the CF and RF GMRES algorithms, but both are considerably worse than the *Lgs* preconditioned RF GMRES approach. Comparing against the unaccelerated cases, using Wielandt acceleration clearly improves the three preconditioned approaches but is quite ineffective for the others as one would expect.

Continuing with the fast reactor problem, Fig. 9 shows the FS iteration convergence with respect to computational effort. Unlike the thermal reactor problem, the number of FS iterations varied considerably due to convergence breakdown in the MG solve. Neither CF GMRES nor RF GMRES converges to the targeted RET within 100 MG iterations for any of the FS iterations, which is the reason behind the erratic behavior seen in Fig. 9. For CF GMRES *Rpm*, the MG iterations in each FS iteration converge sharply at first but then flatten out, mainly because of the different properties of the spatial matrix $\boldsymbol{I}$-$\boldsymbol{R\Pi}$ and the energy matrix $\boldsymbol{I}$-$\boldsymbol{H\Sigma}$. In contrast to the *Rpm* preconditioned GMRES algorithms, RF GMRES with *Lgs* preconditioning has much less trouble in the MG solve process, using only about seven iterations per call.

Our analysis indicates that the improved behavior of the RF versus CF GMRES schemes is due to the efficient usage of the RBGS solution process on each WG system. In short, the unpreconditioned GMRES scheme is not as efficient as the RBGS approach for inverting the current equations. While this approach is applied in the *Rpm* preconditioner, the expense associated with applying it not justified by improvements in the GMRES convergence.

III.B.3. Case 3: Parallel with Wielandt Acceleration

We next examine the use of Wielandt accelerated in combination with parallelization in energy. Figure 10 shows the FS iterative convergence for the thermal reactor problem. Each of the GMRES algorithms requires 17 FS

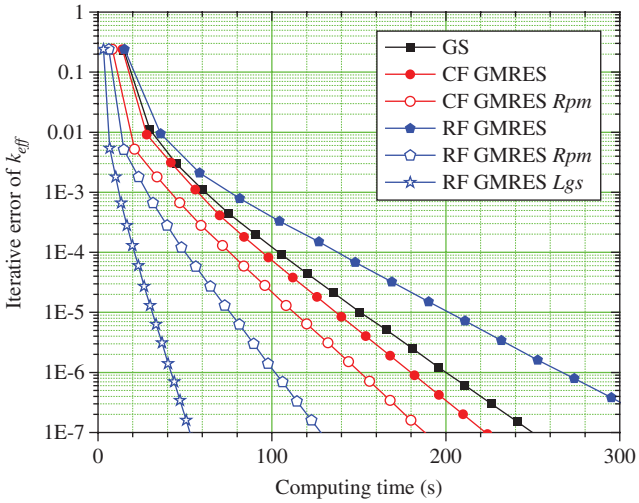

Fig. 8. Fission source iteration convergence of the thermal reactor problem in case 2 (serial with Wielandt acceleration).

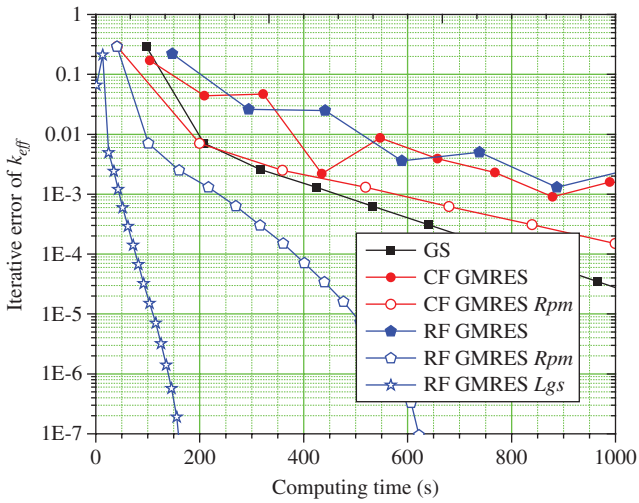

Fig. 9. Fission source iteration convergence of the fast reactor problem in case 2 (serial with Wielandt acceleration).

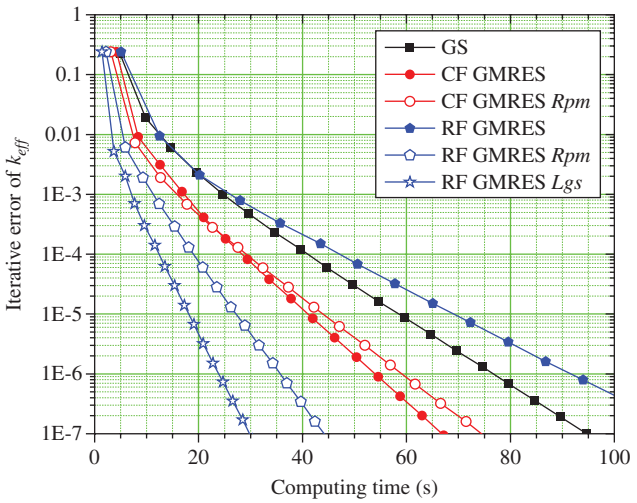

Fig. 10. Fission source iteration convergence of the thermal reactor problem in case 3 (parallel with Wielandt acceleration).

iterations, while more FS iterations (20) are required by the conventional GS algorithm. The GS algorithm is stopped by the maximum MG iteration criterion within each FS iteration, and with parallelization we see 2.5 reduction in computing time or 64% efficiency. The unpreconditioned CF and RF GMRES algorithms scale well with respect to energy, yielding 80% and 71% efficiencies, respectively, but they are both considerably slower than the serial *Lgs* preconditioned RF GMRES algorithm. Only the *Rpm* preconditioned RF GMRES compares well with the RF GMRES *Lgs*, but even it was found to be 50% slower than the relatively poorly performing parallel RF GMRES *Lgs* algorithm. For RF GMRES, there are two factors that affect the efficiency: First, the GS sequential nature of the sweep decreases the parallel efficiency, and second, the reduction of WG iterations per MG iteration improves the load balance. Compared with GS, RF GMRES *Lgs* significantly reduced the computing time in both the accelerated serial case (259 to 54 s) and parallel case (99 to 31 s) as shown in Table IV, but clearly, the parallel efficiency is rather low (43%) due to the GS iteration deterioration during the efficient preconditioning process.

Looking at the fast reactor problem in Fig. 11, the convergence rate was similar to those observed in Fig. 9 for case 2. The parallel efficiencies for the unpreconditioned CF and RF GMRES were found to be 97% and 89%, respectively.

IV. MULTILEVEL ITERATION OPTIMIZATION

In all of the preceding results, the prefixed settings, including convergence criteria for MG and WG iterations and the Wielandt shift, were used to obtain idealized performance results. These settings were determined by performing multiple calculations, which is not a realistic approach for most users to take. In this section, we present a methodology that automatically determines these settings and thus obtains the generic purpose, best achievable performance. The primary goal is to avoid converging the low-level systems too tightly or carrying out any additional high-level iterations. In the adjustment routine, the convergence criteria for the low-level systems

TABLE IV

Computing Time Breakdown Comparison Between GS and *Lgs* Preconditioned RF GMRES Algorithm

Problem	Algorithm		Matrix $\boldsymbol{\Sigma}+\boldsymbol{F}/k'$ (s)			Matrices $\boldsymbol{B}$ $\boldsymbol{C}$ $\boldsymbol{H}$ $\boldsymbol{R}$ (s)	Vector[a] (s)		Other (s)
			App.[b]	Com.[c]	Wait[d]		App.	Wait.	
Thermal	GS	S.[e]	86	—	—	165	—	—	8
		P.[f]	24	9	12	50	—	—	4
	RF GMRES	S.	15	—	—	31	5	—	3
	Lgs	P.	6	2	3	13	2	4	1
Fast	GS	S.	734	—	—	1200	—	—	44
		P.	281	31	50	375	—	—	12
	RF GMRES	S.	67	—	—	105	4	—	5
	Lgs	P.	39	4	7	50	3	—	2

[a]"Vector" = vector orthogonalization.
[b]"App." = actual floating point operation.
[c]"Com." = the communication.
[d]"Wait" = waiting time for processor synchronization.
[e]"S." = serial calculation.
[f]"P." = parallel calculation with four processors.

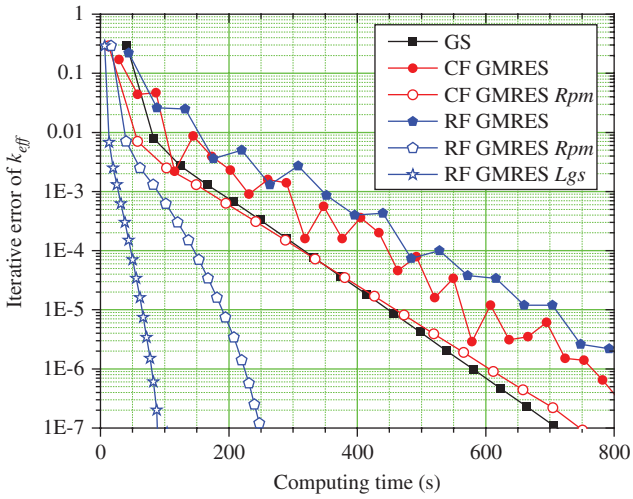

Fig. 11. Fission source iteration convergence of the fast reactor problem in case 3 (parallel with Wielandt acceleration).

are determined according to the convergence rate of the high-level iterations. Because the *Lgs* preconditioned RF GMRES algorithm can be taken as a flexible GMRES without changing a single line of code and the flexible GMRES algorithm can accommodate inexact matrix vector multiplications,[19,20] applying an adaptive iterative convergence criterion within the application of the coefficient matrix does not violate the requirement that left preconditioners must remain fixed at every GMRES iteration.[13]

IV.A. Multigroup Iteration Optimization

To terminate the MG iteration, three criteria are usually used. The first is an AET for the first iteration ($m = 1$):

$$r^{(1)} < \varepsilon_{\text{MG}}$$

and

$$r^{(m)} = \left\| \boldsymbol{\varphi}^{(m)} - \boldsymbol{\varphi}^{(m-1)} \right\|_2 . \tag{39}$$

The second is a RET for the rest of the iterations ($m > 1$):

$$\frac{r^{(m)}}{r^{(1)}} < e_{\text{MG}} . \tag{40}$$

The third is a maximum iteration number $M_{\max}$.

If each MG flux solution is converged sufficiently tightly, each FS iteration can reduce the error by a factor of d ($0 < d < 1$), which is the dominance ratio of the problem. Thus, the RET for the MG flux system within each FS iteration should be related to the dominance ratio. Following a series of VNM tests, it became apparent that the best MG RET lies in the range of $[0.008d, 0.04d]$. RETs larger than $0.04d$ cannot guarantee the shortest computing times as the number of FS iterations can grow considerably.

Excessive loosening of the convergence of the MG iterations slows the convergence of the FS iterations and thus increases the estimate of the dominance ratio and in some cases makes the estimation series of the dominance ratio unstable. The convergence of the FS iteration can be represented by the estimate of the dominance ratio:

$$d_n = \frac{\left\| \boldsymbol{F}\boldsymbol{\varphi}_{(n)} - \boldsymbol{F}\boldsymbol{\varphi}_{(n-1)} \right\|_2}{\left\| \boldsymbol{F}\boldsymbol{\varphi}_{(n-1)} - \boldsymbol{F}\boldsymbol{\varphi}_{(n-2)} \right\|_2} , \tag{41}$$

where n indexes the FS iteration. Theoretically, after several FS iterations these estimates should gradually approach the true dominance ratio and remain stable once the FS iteration has been stabilized.

The estimate of the dominance ratio must be reasonable before any MG RET optimization is applied. We have found that it is impossible to estimate the dominance ratio with sufficient accuracy based upon measurements of just the first four FS iterations. Given that an acceptable range for the MG RET is $(0.008 \text{ to } 0.04)d$, we begin the MG RET at 0.08 for the first FS iteration and tighten it gradually to 0.05, 0.02, and 0.01 for the second, the third, and the fourth FS iterations, respectively. For all FS iterations past the fourth, we check the dominance ratio estimates from the preceding three FS iterations (the first three FS iterations are always excluded). From these estimates, an average dominance ratio estimate $\bar{d}$ and its variance δ_d can be obtained using

$$\bar{d} = \frac{1}{n-n'} \sum_{l=n'}^{n-1} d_l , \quad n' = \max(n-3, 3) \tag{42}$$

and

$$\delta_d = \frac{1}{\bar{d}} \sqrt{\frac{1}{n-n'} \sum_{l=n'}^{n-1} (d_l - \bar{d})^2} , \quad n' = \max(n-3, 3) . \tag{43}$$

From these two parameters, the dominance ratio estimate is evaluated as large (i.e., $\bar{d} > 0.35$) or small, stable (i.e., $\delta_d < 0.2$) or unstable. Then, decisions are carried out appropriately for four different cases, as shown in Fig. 12. The first case is large and stable d, identified as $\bar{d} > 0.95$ and $e_{\text{MG}} > 0.04\bar{d}$ and caused by either a loose MG RET or a problem with a high dominance ratio. The former can be corrected by attempting to tighten the MG RET, while the latter is observed to be unaffected by changes in the MG RET. The second case is a large but unstable d, which indicates that the MG RET is too loose. Thus, we tighten the MG RET by multiplying it by 0.8. The third case is small and stable d, where we then relax

```
If  d̄ > 0.35
   If  δ_d < 0.2                          ! Large and stable d
      If  d̄ > 0.95 & e_MG > 0.04d̄
         e_MG := 0.04d̄                    ! Tighten MG RET
      Else
         e_MG := e_MG / 0.8               ! Relax MG RET
      End if
   Else                                   ! Large but unstable d
      e_MG := 0.8e_MG
   End if
Else if  δ_d < 0.2                        ! Small and stable d
   e_MG := e_MG / 0.8                     ! Relax MG RET
End if
e_MG := min(max(e_MG,0.008d̄),0.1d̄)
```

Fig. 12. Pseudo code for MG iteration RET optimization.

the MG RET immediately since FS iterations with dominance ratio <0.35 will converge rapidly even with a loose MG RET. The fourth case is small but unstable d. In this situation, the only alternative is to wait for an additional FS iteration to be completed. Finally, after all four cases are checked, smaller values of the MG RET are increased to $0.008\bar{d}$, and larger values are decreased to $0.1\bar{d}$.

The effectiveness of Wielandt acceleration depends critically on determining a near optimal eigenvalue shift for the problem under consideration. To understand the problem dependence of the optimal Wielandt shift, a number of VNM problems have been tested using different Wielandt shifts. Figure 13 shows computing time versus k/k' for both serial and parallel implementations of the MG iteration, where k is the true eigenvalue and $k' = k + \delta k$ is the constant shift. The horizontal coordinate $k/(k + \delta k)$ represents the proportion of the FS moved from the right side of the equation to the left. The larger the dominance ratio is, the more dramatic is the Wielandt acceleration. The computing time decreases linearly when $k/k' < d$, while it may become unstable when $k/k' > d$. Consequently, we chose to set the Wielandt shift to k/d.

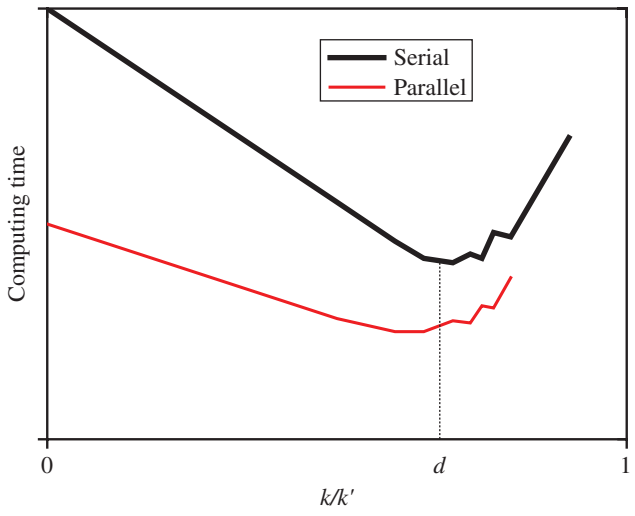

Fig. 13. The influence of Wielandt shifts to total computing efforts.

Similar to the MG RET optimization, the update of the Wielandt shift is only attempted beginning with the fifth FS iteration, when reasonable estimates of the eigenvalue and dominance ratio can be made. Subsequent updates are separated by at least three FS iterations to allow the eigenvalue and dominance ratio estimates to stabilize. For each Wielandt shift revision, the estimate of the dominance ratio is

$$d_{(n-1)} = \frac{1/k_{(n-1)} - 1/k'}{1/k_2 - 1/k'} , \tag{44}$$

where

$k_{(n-1)}$ = estimate of the true eigenvalue

k_2 = second eigenvalue

k' = preceding Wielandt shift.

To complete the shift procedure, we first note that

$$k' := \frac{k}{d} = k_{(n-1)} \frac{k_{(n-1)}}{k_2} , \tag{45}$$

where ":=" stands for the assignment of the value after it to the variable before it. Solving for k_2 from Eq. (44) and substituting it into Eq. (45) then leads to the new shift:

$$k' := k_{(n-1)} \left(\frac{1 - k_{(n-1)}/k'}{d_{(n-1)}} + \frac{k_{(n-1)}}{k'_{(n-1)}} \right) . \tag{46}$$

Wielandt shift optimization can be combined with MG RET optimization, as shown in Fig. 14. There are three primary differences between Fig. 14 and Fig. 12. First, the counter of FS iterations after the last Wielandt shift update is modified with the current Wielandt shift. Second, for large and stable d, the modification of e_{MG} is performed only when there is no Wielandt shift. Third, for a large and stable d, the Wielandt shift is updated only if $\bar{d} > 0.5$ and $N_w > 3$.

From the results presented in Sec. III.B, we have observed that the application of the $\boldsymbol{T}$ matrix, defined by Eqs. (8) and (9), requires nearly all of the computing effort, which means the total computing effort or time is proportional to the total number of MG iterations. Thus, the effects of MG RET optimization can be demonstrated by the number of MG iterations as shown in Table V.

```
N_w=N_w+1
If d̄ > 0.35
   If δ_d < 0.2                                          ! Large d and stable
      If No Wielandt & d̄ > 0.95 & e_MG > 0.04d̄
         e_MG := 0.04d̄                                   ! Tighten MG RET
      Else if NoWielandt & d̄ < 0.95
         e_MG := e_MG/0.8                                ! Relax MG RET
      Else if Wielandt & d̄ > 0.5 & N_w .GT. 3
         k' := k_(n-1)( (1-k_(n-1)/k')/d_(n-1) + k_(n-1)/k'_(n-1) ), N_w:=1   ! Revise Wielandt shift
      End if
   Else if N_w .GE. 3                                    ! Large d but unstable
      e_MG := 0.8e_MG
   End if
Else if δ_d < 0.2                                        ! Small d and stable
   e_MG := e_MG/0.8
End if
e_MG := min(max(e_MG,0.008d̄),0.1d̄)
```

Fig. 14. Pseudo code for MG iteration RET optimization and Wielandt shift revision.

For the thermal reactor problem, the total number of MG iterations is reduced by a factor of ~ 2 in both serial and parallel calculations. For the fast reactor problem, there is little MG RET optimization benefit since only one MG iteration per FS iteration in serial and three for parallel are required by RF GMRES *Lgs* due to the absence of upscattering.

The Wielandt shift optimization results for the thermal and fast reactor problems are listed in Table VI. The constant Wielandt shift together with the MG RET optimization significantly reduces the total number of MG iterations. Replacing the constant Wielandt shift to the self-adjusted Wielandt shift has little effect on the number of MG iterations within each FS iteration but significantly reduces the number of FS iterations. Thus, the self-adjusted Wielandt shift significantly reduces the computing effort required by RF GMRES *Lgs*.

IV.B. Within-Group Iteration Optimization

Within each MG iteration, the $\boldsymbol{T}$ matrix must be applied, which requires the spatial response matrix equation to be inverted for each energy group. The inversion is necessary to update the partial current vector to be consistent with the updated source vector as indicated in Eq. (2). The legacy variational nodal code employs the RBGS algorithm and uses three criteria to terminate the iteration: an AET for the first iteration,

$$\gamma_g^{(1)} \le \varepsilon_{\mathrm{WG},g} \tag{47}$$

and

$$\gamma_g^{(i)} = \left\| \boldsymbol{j}_g^{(i)} - \boldsymbol{j}_g^{(i-1)} \right\|_2 ; \tag{48}$$

a RET for the rest of iterations,

$$\frac{\gamma_g^{(i)}}{\gamma_g^{(1)}} \le e_{\mathrm{WG},g} ; \tag{49}$$

and a maximum iteration number $I_g^{\max}$, where i indexes the WG iteration number.

For each energy group's WG iterations within each MG iteration, both the RET and the maximum iteration number could be optimized. Similar to the relationship between FS and MG iterations, the RET of the WG

TABLE V

Multigroup Iteration Optimization Performance Measured in Total MG Iteration Number

Problem	Serial		Parallel	
	No Optimization	Optimized	No Optimization	Optimized
Thermal reactor problem	1029	532	1411	866
Fast reactor problem	144	144	720	718

TABLE VI

Wielandt Shift Optimization Performance Measured in Total MG Iteration Number

Problem	Serial			Parallel		
	No	Constant	Optimized	No	Constant	Optimized
Thermal reactor problem	532	238	147	866	348	351
Fast reactor problem	144	108	106	718	200	190

iteration is optimized by relating it to the spectral radius of the MG iteration, which can be estimated as follows:

$$\rho_{MG} = \frac{1}{M^{active} - 1} \sum_{m=2}^{M^{active}} \frac{r^{(m)}}{r^{(m-1)}} , \quad (50)$$

where M^{active} is the actual MG iteration number in the last FS iteration. For the first four FS iterations and for the first two MG iterations within each of the remaining FS iterations, the RET of the WG iteration is set as $e_{\text{WG}} = 0.1$. For subsequent iterations, ρ_{MG} is selected as the RET of WG iteration.

To optimize the maximum iteration number of the WG iteration, the averaged optical depth parameter is employed for each energy group g:

$$\bar{\tau}_g = \frac{\sum_{node} \Sigma_{t,g,node} D_{node} V_{node}}{\sum_{node} V_{node}} , \quad (51)$$

where

$\Sigma_{t,g,node}$ = total macroscopic cross section (cm^{-1})

D_{node} = characteristic dimension (cm)

V_{node} = volume of the node (cm^3).

The characteristic dimension for nodes in two-dimensional Cartesian and hexagonal geometries are $\frac{2ab}{a+b}$ and $\frac{2\sqrt{3}}{\pi}p$, respectively, where a and b are the length and width of the node and p is the pitch of the hexagon. We determine that energy groups with large optical depths (such as $\bar{\tau}_g > 1$) need small numbers of spatial sweeps.

In groups where $\bar{\tau}_g > 1$, $I_g^{\max} = 1$ is selected. For groups in which $\bar{\tau}_g < 1$, the maximum WG iteration number is selected as

$$I_g^{\max} = \min\left(\max\left(\text{int}\left(\frac{\log e_{\text{WG}}}{\log \bar{\rho}_{\text{WG}}} \right) + \left| \text{int}\left(\frac{\log e_{md}}{\log \bar{\rho}_{\text{WG}}} \right) \right|, 2 \right), I'_{\max} \right) , \quad (52)$$

where

e_{WG} = RET just obtained

$I'_{\max}$ = maximum WG iteration number provided by the user

e_{md} = energy group averaged error reduction modification factor obtained from

$$e_{md} = \sqrt[G]{\prod_g e_{\text{WG}} \frac{\gamma_g^{(1,1)}}{\gamma_g^{\left(1, I_g^{\text{active}}\right)}}} , \quad (53)$$

where I_g^{active} refers to the WG iteration number from the last FS iteration and MG iteration. The energy group averaged spectral radius is

$$\bar{\rho}_{\text{WG}} = \sqrt[G]{\prod_g \rho_g} , \quad (54)$$

where the spectral radius of the WG system in group g is

$$\rho_g = \begin{cases} \min\left(\max\left(\frac{1}{I_g^{\text{active}} - 1} \sum_{i=2}^{I_g^{\text{active}}} \frac{\gamma_g^{(1,i)}}{\gamma_g^{(1,i-1)}}, 0.5 \right), 0.99 \right) & I_g^{\text{active}} \geq 2 \\ e_{\text{WG}} & I_g^{\text{active}} = 1 \end{cases} . \quad (55)$$

The results of WG iteration optimization for the thermal and fast reactor problems are listed in Table VII for the cases without and with optimized Wielandt shift. Observe that the WG optimization reduces the total number of WG iterations by a factor of 5 for the thermal reactor problem when there is no Wielandt acceleration and by factors of 6 and 9 for the serial and parallel calculations with Wielandt acceleration. For the fast reactor problem, the reduction factors are 8 and 21 for the serial and parallel calculations without Wielandt acceleration and are 3 and 4 with Wielandt acceleration.

V. DISCUSSION

The legacy GS algorithm applied to MG iterations takes advantage of the nearly lower triangular property of the scattering matrix when only weak upscattering is

TABLE VII

Within-Group Iteration Optimization Performance Measured in Total WG Iteration Number

Problem	WG Iteration Optimization	No Wielandt		Optimized Wielandt	
		Serial	Parallel	Serial	Parallel
Thermal reactor problem	Without	199 871	309 904	58 780	167 845
	With	39 084	60 559	9 231	17 086
Fast reactor problem	Without	248 832	2 468 867	462 785	777 296
	With	29 376	115 290	141 912	174 312

present. It suffers, however, from the pseudo upscattering introduced by Wielandt acceleration and from the transformation from GS to block GS or block Jacobi in energy parallelization.

In contrast, the GMRES algorithms readily accommodate upscattering by solving for the energy groups simultaneously. For all cases considered, the *Rpm* preconditioner effectively reduces the total number of MG GMRES iterations and thus also the computing time. For both the thermal and fast reactor problems, *Lgs* preconditioning of RF GMRES reduces the total number of MG iterations and the computing time more significantly than the *Rpm* preconditioning, primarily because it requires no additional computational effort to solve the preconditioning system.

From our investigations, the *Lgs* preconditioned RF GMRES algorithm has been found to be the best, simply because it combines the GMRES algorithm and the legacy GS scheme. Compared to the existing GS algorithm, it can reduce the computing time by a factor of ~ 2 for the thermal reactor problem and performs identically for the fast reactor problem, where there is no upscattering. With Wielandt acceleration and energy group parallelization (with four processors), where the legacy GS severely suffers from slow convergence, RF GMRES *Lgs* reduces the total computing time by a factor of ~ 10 for the thermal reactor problem and by a factor of ~ 2 for the fast reactor problem. Both GS and RF GMRES *Lgs* lose parallel efficiency because they suffer from the energy sweeping degeneration to block GS or even Jacobi. However, the new algorithms encounter much less trouble since they put the GS iteration into a preconditioning process instead of the main iteration.

Relating the RET of the low-level iteration to the convergence rate of the corresponding high-level iteration serves to optimize the convergence criteria of the MG iteration within each FS iteration as well as that of the WG iteration within each MG iteration. The choice of $k' = k/d$ offers an approach to self-adjusting the Wielandt shift as the FS iteration proceeds. However, such optimizations are empirical in nature and lack a rigorous mathematical basis needed to guarantee their performance. Both the multilevel iteration optimization and the Wielandt shift self-adjusting techniques are applicable to the *Lgs* preconditioned RF GMRES and are found to be fairly effective in improving computational efficiency.

This investigation has been carried out within the frame of VNM. It has focused on diffusion criticality calculation in two dimensions. With the experience gained, a fruitful line of research may be to extend methods developed to three-dimensional problems and to higher angular order spherical harmonic approximations to the transport equation.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No. 11305123), the State Scholarship Fund of China (No. 2010628100), and the U.S. Department of Energy (No. DE-AC02-06CH11357).

REFERENCES

1. E. E. LEWIS and W. F. MILLER, JR., *Computational Methods of Neutron Transport*, John Wiley & Sons, New York (1984).

2. G. PALMIOTTI, E. E. LEWIS, and C. B. CARRICO, "VARIANT: Variational Anisotropic Nodal Transport for Multidimensional Cartesian and Hexagonal Geometry Calculation," ANL-95/40, Argonne National Laboratory (1995).

3. C. B. CARRICO, E. E. LEWIS, and G. PALMIOTTI, "Three-Dimensional Variational Nodal Transport Methods for Cartesian, Triangular, and Hexagonal Criticality Calculations," *Nucl. Sci. Eng.*, **111**, 168 (1992); http://dx.doi.org/10.13182/NSE92-1.

4. W. GROPP, E. LUSK, and A. SKJELLUM, *Using MPI: Portable Parallel Programming with the Message Passing Interface*, 2nd ed., MIT Press, Cambridge, Massachusetts (Nov. 1999).

5. D. R. FERGUSON and K. L. DERSTINE, "Optimized Iteration Strategies and Data Management Considerations for Fast Reactor Finite Difference Diffusion Theory Codes," *Nucl. Sci. Eng.*, **64**, 593 (1977); http://dx.doi.org/10.13182/NSE77-5.

6. T. M. SUTTON, "Wielandt Iteration as Applied to the Nodal Expansion Method," *Nucl. Sci. Eng.*, **98**, 169 (1988); http://dx.doi.org/10.13182/NSE88-1.

7. Z. XIE and L. DENG, *Numerical Calculation Method of Neutron Transport Theory*, Northwestern Polytechnical University Press, Xi'an, China (2005).

8. E. E. LEWIS et al., "Preconditioned Krylov Solution of Response Matrix Equations," *Nucl. Sci. Eng.*, **173**, *3*, 222 (2013); http://dx.doi.org/10.13182/NSE11-106.

9. Y. SAAD and M. H. SCHULTZ, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems, *SIAM J. Sci. Stat. Comput.*, **7**, 856 (1986); http://dx.doi.org/10.1137/0907058.

10. R. N. SLAYBAUGH, "Acceleration Methods for Massively Parallel Deterministic Transport," PhD Thesis, University of Wisconsin–Madison (2011).

11. T. M. EVANS et al., "Denovo—New Three-Dimensional Parallel Discrete Ordinates Code in SCALE," *Nucl. Technol.*, **171**, *2*, 171 (2010); http://dx.doi.org/10.13182/NT10-5.

12. A. HEBERT, "Variational Principles and Convergence Acceleration Strategies for the Neutron Diffusion Equation," *Nucl. Sci. Eng.*, **91**, *4*, 414 (1985); http://dx.doi.org/10.13182/NSE85-6.

13. Y. SAAD, *Iterative Methods for Sparse Linear Systems*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania (2003).

14. Y. LI, E. E. LEWIS, and M. A. SMITH, "A *p* Preconditioned GMRES Algorithm for Multi-Group Variational Nodal Eigenvalue Problems," *Trans. Am. Nucl. Soc.*, **105**, 508 (2011).

15. G. PALMIOTTI et al., "UNÌC: Ultimate Neutronic Investigation Code," *Proc. Mathematics and Computations and Supercomputing in Nuclear Applications (M&C+SNA 2007)*, Monterey, California, April 15–19, 2007, American Nuclear Society (2007).

16. Y. LI, "Advanced Reactor Core Neutronics Computational Algorithms Based on the Variational Nodal and Nodal SP_3 Methods," Xi'an Jiaotong University (2013) (in Chinese).

17. G. MARLEAU, A. HÉBERT, and R. ROY, "A User's Guide for DRAGON," École Polytechnique de Montréal (Dec. 1997).

18. C. H. LEE and W. S. YANG, "Development of Multigroup Cross-Section Generation Code MC^2-3 for Fast Reactor Analysis," *Proc. Int. Conf. Fast Reactors (FR 2009)*, Kyoto, Japan, December 7–11, 2009.

19. V. SIMONCINI and D. B. SZYLD, "Theory of Inexact Krylov Subspace Methods and Applications to Scientific Computing," *SIAM J. Sci. Comput.*, **25**, 454 (2003); http://dx.doi.org/10.1137/S1064827502406415.

20. Y. SAAD, "A Flexible Inner-Outer Preconditioned GMRES Algorithm," *SIAM J. Sci. Comput.*, **14**, 461 (1993); http://dx.doi.org/10.1137/0914028.