*Nuclear Mathematical and Computational Sciences: A Century in Review, A Century Anew*
Gatlinburg, Tennessee, April 6-11, 2003, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2003)

# IMPLEMENTATION OF A NEWTON-BICGSTAB SOLVER TO TREAT THE STRONG NON-LINEARITY IN THE FORMOSA-B BOILING WATER REACTOR CORE SIMULATOR CODE

**Doddy Kastanya and Paul J. Turinsky**
Electric Power Research Center
Department of Nuclear Engineering
North Carolina State University
Box 7909
Raleigh, NC 27695-7909
dfkastan@eos.ncsu.edu; turinsky@eos.ncsu.edu

## ABSTRACT

A Newton-BICGSTAB solver has been developed to reduce the CPU execution time of BWR core simulators. The new solver treats the strong non-linearities in the problem explicitly using Newton's method, replacing the traditionally used nested iterative approach. Taking advantage of the higher convergence rate provided by Newton's method and utilizing an efficient preconditioned BICGSTAB solver to solve the linearized system of equations within each Newton iteration, we have developed a computationally efficient Newton-BICGSTAB solver to evaluate the three-dimensional, two-group neutron diffusion equations coupled with a two-phase flow model within a BWR core simulator. The robustness of the solver has been tested against numerous BWR core configurations and consistent results have been observed each time. The best exact Newton-BICGSTAB solver provides a speedup of 2.1 to the core simulator with reference to the traditional approach. On average, the exact Newton-BICGSTAB solver provides a speedup of around 1.7. We also examined several inexact Newton-BICGSTAB solvers. The results from this study indicate that the inexact Newton-BICGSTAB solver can provide an additional speedup up to 20% with reference to the exact Newton-BICGSTAB solver.

*Key Words*: Newton-Krylov, BICGSTAB, BWR core simulator, non-linear neutronics/thermal hydraulics equations, solution of neutron diffusion equations.

## 1. INTRODUCTION

In the course of performing a boiling water reactor (BWR) reload core analysis, a core simulator is extensively used in the preliminary process of choosing a pair of loading pattern-control rod pattern (LP-CRP) by verifying the attractiveness of the pair for various normal operation conditions as well as different accident scenarios to guarantee a safe and economical operation of the reactor. Therefore, having a fast core simulator is an important factor in expediting the reload core design and analysis.

Automated mathematical optimization tools developed for determining an optimum LP-CRP pairing in a BWR core are becoming available and can be used by a core designer to ease the process of selecting a good LP-CRP pair for a particular reload core design. Independent of the optimization method employed by the tools, these tools have to evaluate thousands of LP-CRP pairings to assess their attractiveness before arriving at the optimum or near-optimum pair. Each evaluation involves solving the three-dimensional, two-group neutron diffusion equation coupled with a two-phase fluid flow model, *e.g.* the three-equation mixture drift flux model. Given that these tightly coupled non-linear equations must also be solved as a function of cycle burnup, the associated computational burden could become excessive. For example, a FORMOSA-B LP-CRP optimization during which approximately 4000 combinations of LP-CRP pairings are sampled typically takes about 27 CPU hours on a 2 GHz DELL 530 workstation platform.

Although the speedup gained from this study might not seem significant for an evaluation of an LP-CRP pair which usually needs only a couple of minutes to complete, reducing the time associated with the evaluation of a single LP-CRP pair is still a key factor for reducing the overall running time of an optimization run. In this project, this target is pursued by assessing the effectiveness of a Newton-Krylov solver in replacing the nested iterative approach or similar methods currently utilized by most BWR core simulators for treating the non-linear feedback properties. Addressing the non-linearity of the problem explicitly using the Newton's method, which is capable of performing simultaneous updates of all unknowns, aided by an appropriately preconditioned Krylov method for solving the linearized system of equations within the Newton iteration, should reduce the CPU execution time of an optimization run.

This paper summarizes the development of a robust and computationally efficient Newton-Krylov algorithm to solve the three-dimensional, two-group neutron diffusion equation coupled with a two-phase flow model. To assess the performance of this new algorithm, we have implemented the Newton-Krylov algorithm in the core simulator part of the in-core fuel management optimization code FORMOSA-B [1, 2] replacing the currently utilized traditional nested iterative algorithm. Although results reported in this paper are based upon the implementation in the FORMOSA-B code, they should be reproducible by other BWR core simulators.

## 2. METHODOLOGY

### 2.1. The Strengths of Non-Linear Feedback Mechanisms

Prior to implementing the Newton-Krylov algorithm, we need to determine which unknowns, including the non-linear feedbacks unknowns, should be updated simultaneously by Newton's method. To answer this question, we have performed a comprehensive study to determine which feedback mechanisms can be categorized as strong feedbacks. The number of outer (fission-source) iterations in a traditional nested iterative approach is used as a measure of the strength of a particular feedback mechanism. If the total number of outer iterations increases noticeably when a non-linear feedback effect is treated, then the feedback is categorized as a strong feedback. The non-linear feedback associated with the thermal hydraulics, coolant inlet flow redistribution, transient fission products, and the non-linear nodal iterative method spatial coupling correction coefficients have been examined. The study was performed on four core types, namely a 368 assembly GE BWR/4 core, 800 assembly GE BWR/6 core, 724 assembly GE BWR/3 core, and 560 assembly GE BWR/4 core, to verify the consistency of the results.

Based on consistent results observed over all four core types examined, we concluded that the thermal hydraulics feedbacks are considered strong feedbacks. Therefore, their associated unknowns will be treated using Newton's method. The remaining weak feedbacks will be treated in a nested iterative manner.

## 2.2. Mathematical Background

Consider a non-linear equation given by

$$\overline{A}(\overline{\Phi}) = \overline{Q} \tag{1}$$

where the non-linear operator $\overline{A}$ operates on the unknown vector $\overline{\Phi}$ whose vector dependence indicates different reactor core properties. Eq. (1) includes appropriate boundary conditions and, if an eigenvalue problem is involved, normalization constraint. For the problem at hand, the unknown vector $\overline{\Phi}$ consist of all unknowns and can be subdivided into the eigenvalue ($\lambda$), where $\lambda = 1/k_{eff}$ , the two-group neutron flux ($\overline{\phi}$), the strong non-linear feedback properties ($\overline{\xi}$), and the weak non-linear feedback properties ($\overline{\Psi}$).

The subdivision of the non-linear feedback properties is performed in order to allow different iterative update frequencies to be applied for the strong and the weak feedback properties. Expressing Eq. (1) explicitly in terms of the eigenvalue, neutron flux, and feedback properties, we obtain

$$\overline{A}(\overline{\Phi}) = \begin{bmatrix} (\overline{\overline{M}}(\overline{\xi},\overline{\Psi}) - \lambda\overline{\overline{F}}(\overline{\xi},\overline{\Psi}))\overline{\phi} \\ \hat{\overline{T}}(\overline{\phi},\overline{\xi},\overline{\Psi}) \\ \overline{W}(\overline{\phi},\overline{\xi},\overline{\Psi}) \end{bmatrix} = \begin{bmatrix} \overline{0} \\ \hat{\overline{Q}}_{\xi} \\ \overline{0} \end{bmatrix} \tag{2}$$

The first row of Eq. (2) indicates the balance equation for the neutron flux, *i.e.* the spatially discretized form of the three-dimensional, two-group neutron diffusion equation. $\overline{\overline{M}}$ and $\overline{\overline{F}}$ denote the loss and production operators, respectively. The second row of Eq. (2) represents some form of two-phase fluid mass, energy, and momentum conservation equations, along with the constitutive equations which are used to determine the coolant density distribution in the core, plus the core power level normalization constraint. The third row states the balance equations for the weak non-linear feedback properties. Those who are interested in the actual equations involved in this system of equations are referred to one of the author's doctorate dissertation [3] which is available electronically through the NCSU Library website (www.lib.ncsu.edu).

To understand the non-linearity embedded in Eq. (2), one needs to realize that the cross sections used in forming the operators are non-linearly dependent directly on the neutron flux and indirectly on the power density. The direct dependencies of cross sections on flux are due to transient fission products, *e.g.* $Pm^{149}$ and $Xe^{135}$. The indirect dependencies of cross section on power density are revealed through their dependencies on the moderator density and fuel temperature.

From this overall non-linear problem, we can separate out the non-linear system of equations involving the strong non-linear feedbacks, which will be solved utilizing Newton's method, and we can mathematically express it as follow:

$$\bar{B}(\bar{\Theta}) \equiv \begin{bmatrix} (\bar{\bar{M}}(\bar{\xi},\bar{\Psi}) - \lambda \bar{\bar{F}}(\bar{\xi},\bar{\Psi}))\bar{\phi} \\ \bar{T}(\bar{\phi},\bar{\xi},\bar{\Psi}) \\ \left\langle \overline{\kappa\Sigma_f}, \bar{\phi} \right\rangle \end{bmatrix} = \bar{Q} \equiv \begin{bmatrix} \bar{0} \\ \bar{Q}_\xi \\ Q_P \end{bmatrix} \tag{3}$$

where $\bar{\Theta} \equiv \left[\bar{\phi}^T, \bar{\xi}^T, \lambda\right]^T$.

The core power level constraint equation has been separated out of the $\bar{\hat{T}}$ operator in Eq. (3), with the $\bar{T}$ operator denoting the two-phase flow thermal hydraulics conservation and constitutive equations. As noted earlier, since an eigenvalue equation is being solved, a constraint equation is required, in this case to normalize the flux. The constraint equation allows the eigenvalue to be introduced as an additional unknown.

The exact Newton equation for the non-linear equation, Eq. (3), suppressing the dependence upon the weak non-linear feedbacks, $\bar{\Psi}$, which are updated in an outer iterative loop, can be expressed as

$$\bar{\bar{J}}(\bar{\Theta}_m)\bar{\Theta}_{m+1} = \bar{R}(\bar{\Theta}_m)\,, \tag{4}$$

where the Jacobian matrix has been defined as

$$\bar{\bar{J}}(\bar{\Theta}_m) \equiv \left.\frac{\partial \bar{B}(\bar{\Theta})}{\partial \bar{\Theta}}\right|_{\bar{\Theta}_m} \tag{5}$$

and

$$\bar{R}(\bar{\Theta}_m) \equiv \bar{\bar{J}}(\bar{\Theta}_m)\bar{\Theta}_m - (\bar{B}(\bar{\Theta}_m) - \bar{Q})\,, \tag{6}$$

with index $m$ now denoting the Newton iteration count. For our problem, the Jacobian matrix can be written as

$$\bar{\bar{J}}(\bar{\Theta}_m) = \begin{bmatrix} \left[\bar{\bar{M}}(\bar{\xi}) - \lambda \bar{\bar{F}}(\bar{\xi})\right] & [(\frac{\partial \bar{\bar{M}}(\bar{\xi})}{\partial \bar{\xi}} - \lambda \frac{\partial \bar{\bar{F}}(\bar{\xi})}{\partial \bar{\xi}})\bar{\phi}] & -\bar{\bar{F}}(\bar{\xi})\bar{\phi} \\ \frac{\partial \bar{T}(\bar{\phi},\bar{\xi})}{\partial \bar{\phi}} & \frac{\partial \bar{T}(\bar{\phi},\bar{\xi})}{\partial \bar{\xi}} & \bar{0} \\ \frac{\partial \left\langle \overline{\kappa\Sigma_f}, \bar{\phi} \right\rangle}{\partial \bar{\phi}} & \frac{\partial \left\langle \overline{\kappa\Sigma_f}, \bar{\phi} \right\rangle}{\partial \bar{\xi}} & 0 \end{bmatrix}_{\bar{\Theta}_m} \tag{7}$$

Eq. (7) denotes a 3x3 block matrix whose block matrix components are denoted by $\bar{\bar{J}}(\bar{\Theta}_m)_{i,j}$ for $i, j = 1,2,3$. Using Eqs. (4) and (7), one obtains the following iterative algorithm for the Newton's method:

$$\bar{\bar{J}}(\bar{\Theta}_m)_{1,1}\bar{\phi}_{m+1} + \bar{\bar{J}}(\bar{\Theta}_m)_{1,2}\bar{\xi}_{m+1} + \bar{\bar{J}}(\bar{\Theta}_m)_{1,3}\lambda_{m+1} = \bar{\bar{J}}(\bar{\Theta}_m)_{1,1}\bar{\phi}_m + \bar{\bar{J}}(\bar{\Theta}_m)_{1,2}\bar{\xi}_m + \bar{\bar{J}}(\bar{\Theta}_m)_{1,3}\lambda_m - [\bar{\bar{M}}(\bar{\xi}_m) - \lambda_m \bar{\bar{F}}(\bar{\xi}_m)]\bar{\phi}_m, \tag{8}$$

$$\bar{\bar{J}}(\bar{\Theta}_m)_{2,1}\bar{\phi}_{m+1} + \bar{\bar{J}}(\bar{\Theta}_m)_{2,2}\bar{\xi}_{m+1} = \bar{\bar{J}}(\bar{\Theta}_m)_{2,1}\bar{\phi}_m + \bar{\bar{J}}(\bar{\Theta}_m)_{2,2}\bar{\xi}_m - [\bar{T}(\bar{\phi}_m, \bar{\xi}_m) - \bar{Q}_{\xi}], \tag{9}$$

and

$$\bar{\bar{J}}(\bar{\Theta}_m)_{3,1}\bar{\phi}_{m+1} + \bar{\bar{J}}(\bar{\Theta}_m)_{3,2}\bar{\xi}_{m+1} = \bar{\bar{J}}(\bar{\Theta}_m)_{3,1}\bar{\phi}_m + \bar{\bar{J}}(\bar{\Theta}_m)_{3,2}\bar{\xi}_m - [\left\langle \left(\overline{\kappa\Sigma_f}\right)_m \bar{\phi}_m \right\rangle - Q_P]. \tag{10}$$

Reintroducing the weak non-linear feedback values, they are determined by solving for $\bar{\Psi}$ using

$$\bar{W}(\bar{\Theta}_{\hat{m}_j}, \bar{\Psi}_{o+1}) = \bar{0}, \text{ for j=1,2,\ldots} \tag{11}$$

where $\hat{m}_j$ denotes the specific values of $m$ that updates to $\bar{\Psi}$ occur at, *i.e.* update frequency.

This set of equations is clearly more difficult to solve than the original matrix equations associated with the traditional nested iterative approach due to the simultaneous couplings of more unknowns and denser matrix. However, the Newton's method can still be computationally attractive since it converges to the solution at a-higher-than-linear rate provided that a good initial guess of the unknowns is utilized.

### 2.3. Linearization Error Study

In order to assess the feasibility of employing Newton's method for treating the strong non-linearities in the BWR core simulator model, we would like to first understand how accurate the first-order approximation is for our problem since Newton's method is based upon linearizing the non-linear equations. A linearization error study has been performed to answer this question. In addition, this study was aimed to check the singularity property of the Jacobian matrix system at convergence and to serve as a tool to validate the derivation and solution of the linearized system of equations.

The study was performed on a 368 assembly GE BWR/4 core. Since this study is focused on a single Newton step, the core simulator options for treating non-linear feedbacks not explicitly treated by the Newton's method, *e.g.* inlet coolant flow redistribution, critical flow search, and fission products, are turned off. The study begins by obtaining the solution at the 100% power level. Then, we performed linearization around this solution, *i.e.* evaluate the Jacobian matrix at 100% power level, and subsequently solve for the changes in the eigenvalue, two-group neutron flux, and moderator pressure, density, void fraction, and internal energy distribution at a perturbed core power level, 100±x%, *via* one Newton iteration. We also solved for these core variables at the perturbed power level using a traditional iterative solver. The differences between the exact changes, *i.e.* traditional solver, and the approximate changes, *i.e.* one Newton iteration, in these variables are defined as the linearization errors.

Results from the linearization error study, after perturbing the core to ±10% power level from the reference 100% power level, indicate that the errors introduced by the first-order approximation are insignificant. The other thing learned from this study is that the Jacobian matrix is not singular. The singularity of the Jacobian matrix becomes a concern since the $\overline{\overline{J_{11}}}$ block, defined as $\left(\overline{\overline{M}}(\bar{\xi},\overline{\Psi})-\lambda\overline{\overline{F}}(\bar{\xi},\overline{\Psi})\right)$, is singular when the value of $\lambda$ is the true eigenvalue associated with the generalized eigenvalue problem. This is indeed the case for our linearization error study since we utilize the eigenvalue problem solution (at 100% power level) as our reference. However, in reality additional terms are added to the $\overline{\overline{J_{11}}}$ block to account for cross-section changes due to fuel temperature changes. This follows since the cross-section is a function of fuel temperature, which in turns is a function of power density, which in turns is a function of flux, and we reduce the cross-section dependence to flux. These additional terms when added to the $\overline{\overline{J_{11}}}$ block prevent this block from being singular.

### 2.4. Krylov Methods

The effectiveness of Krylov methods for solving the linearized system of equations within each Newton iteration has been analyzed. For this project, we require Krylov methods which can solve a system of equations with a non-symmetric coefficient matrix, since our Jacobian matrix is non-symmetric. We decided to examine three such Krylov methods, namely the bi-conjugate gradient stabilized (BICGSTAB), conjugate gradient squared (CGS), and restart generalized minimal residual (GMRES). The most computationally efficient solver will be employed in the final implementation of the Newton-Krylov solver. The algorithms for these methods are available in the literature ([4] through [6]) and will not be discussed here. Various preconditioner matrices were examined as explained in Section 2.6. in the course of this assessment of Krylov methods, which resulted in selection of the BICGSTAB method.

### 2.5. The Newton-Krylov Algorithm

The algorithm for the Newton-Krylov solver implemented in FORMOSA-B is depicted in Fig. 1. The Newton iteration performs simultaneous updates of the eigenvalue, two-group neutron flux, and thermal hydraulics properties. Note that the fission source iterations have been completely eliminated. This is possible since the singular matrix system that appears in the eigenvalue problem does not appear in Newton's method, *i.e.* the Jacobian matrix is non-singular. Although this figure is fairly self-explanatory, there are two items which should be pointed out. First, the weak non-linear feedback properties are being updated at a certain frequency of Newton (outer) iterations. Second, the box labeled "Setup the Jacobian matrix system and perform the LU factorization" is thicker than the other boxes to indicate that these processes are done at a certain frequency depending upon whether an exact or inexact Newton's method is utilized.

Given that the Newton-Krylov algorithm is a nested iterative algorithm, a specific stopping criteria is needed for each level of the nested iterative loops. As the stopping criteria for the Krylov solver, we require the L$_2$-norm of the reduction in the residual to be less than a specified value

$$\frac{\left\|\bar{r}_{(m,n)}\right\|_2}{\left\|\bar{r}_{(m,0)}\right\|_2} \le \varepsilon_{Krylov}\,, \tag{12}$$

where

$$\bar{r}_{(m,n)} = \begin{bmatrix} \bar{R}_{(m)_1} \\ \bar{R}_{(m)_2} \\ R_{(m)_3} \end{bmatrix} - \begin{bmatrix} \bar{\bar{J}}_{11} & \bar{\bar{J}}_{12} & \bar{\bar{J}}_{13} \\ \bar{\bar{J}}_{21} & \bar{\bar{J}}_{22} & \bar{0} \\ \bar{\bar{J}}_{31} & \bar{\bar{J}}_{32} & 0 \end{bmatrix} \begin{bmatrix} \delta\bar{\phi}_{(m,n)} \\ \delta\bar{\xi}_{(m,n)} \\ \delta\lambda_{(m,n)} \end{bmatrix}, \tag{13}$$

and the pair $(m,n)$ denotes the (Newton, Krylov) iteration step counts. Note that when this Krylov iteration stopping criteria is satisfied, say at Krylov iteration $\hat{n}$, we denote $(m) = (m,\hat{n})$. For terminating the Newton iteration, the following stopping criteria is used:

$$\frac{\left\|\bar{\bar{M}}(\bar{\xi}_m,\bar{\Psi}_o)\bar{\phi}_m - \lambda_m \bar{\bar{F}}(\bar{\xi}_m,\bar{\Psi}_o)\bar{\phi}_m\right\|_2}{\left\|\lambda_m \bar{\bar{F}}(\bar{\xi}_m,\bar{\Psi}_o)\bar{\phi}_m\right\|_2} \le \varepsilon_{Newton}\,. \tag{14}$$

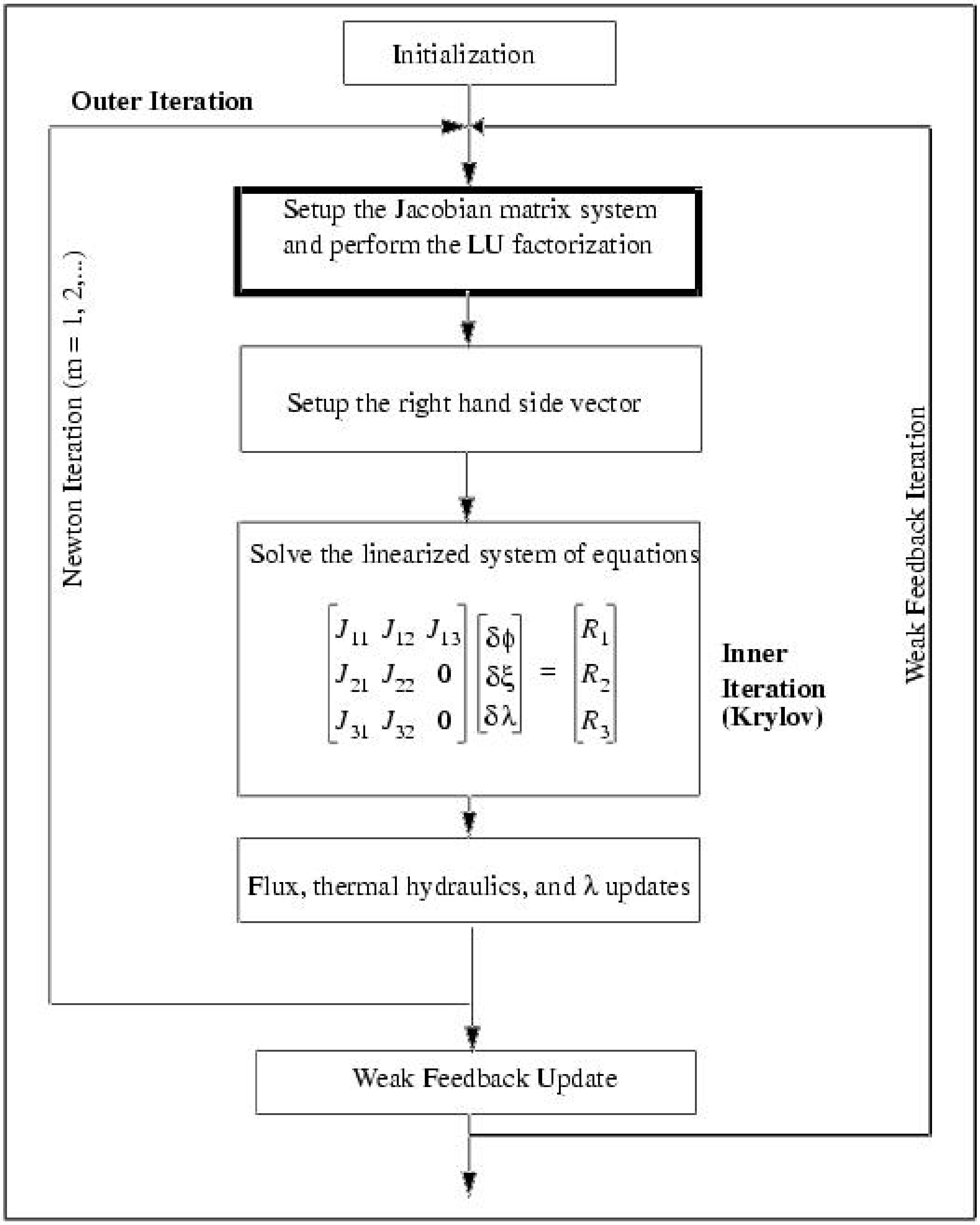
Initialization
Outer Iteration
Setup the Jacobian matrix system and perform the LU factorization
Setup the right hand side vector
Solve the linearized system of equations
$\begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & 0 \\ J_{31} & J_{32} & 0 \end{bmatrix} \begin{bmatrix} \delta\phi \\ \delta\xi \\ \delta\lambda \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix}$
Inner Iteration (Krylov)
Flux, thermal hydraulics, and λ updates
Weak Feedback Update
Newton Iteration (m = 1, 2,...)
Weak Feedback Iteration

**Figure 1. The Newton-Krylov algorithm.**

We realize that this stopping criteria only checks the neutron diffusion equation solution, while the Jacobian matrix system also includes thermal hydraulics properties. Hence, the stopping criteria was formulated by assuming that when the two-group fluxes and eigenvalue converge to sufficient accuracy, the thermal hydraulics properties have already converged to the accuracy desired. We have performed numerical tests to show that this is indeed a valid assumption. Another advantage of utilizing this stopping criteria is related to the fact that the term whose norm is taken and used as the numerator in Eq. (14) is readily available since it is a part of the right hand side vector for the subsequent Newton iteration. Therefore, evaluation of the Newton iteration's stopping criteria does not add much to the computational cost.

Similar to the traditional solver, to terminate the overall solver, we checked the convergence of the eigenvalue, fission neutron source density, and moderator density, in the outer most loop of the nested loops, *i.e.* the weak non-linear feedback loop. Mathematically, these stopping criteria can be written as

$$\frac{\left|k_{eff}^{(o)}-k_{eff}^{(o-1)}\right|}{k_{eff}^{(o)}}\leq\varepsilon_k \,, \tag{15}$$

$$\frac{\left\|[\overline{\overline{F}}\overline{\phi}]^{(o)}-[\overline{\overline{F}}\overline{\phi}]^{(o-1)}\right\|_2}{\left\|[\overline{\overline{F}}\overline{\phi}]^{(o)}\right\|_2}\leq\varepsilon_F \,, \tag{16}$$

and

$$\frac{\left\|\overline{\rho}^{(o)}-\overline{\rho}^{(o-1)}\right\|_2}{\left\|\overline{\rho}^{(o)}\right\|_2}\leq\varepsilon_\rho \,, \tag{17}$$

where recall that the superscript $o$ denotes the weak non-linear feedback iterative count.

### 2.6. Development of Preconditioners

In general, the rate of convergence of iterative methods depends on the spectral properties of the coefficient matrix. For Krylov methods, the condition number of the coefficient matrix plays an important role in determining the rate of convergence. One may attempt to transform the original linear system into an equivalent system in the sense that it still has the same solution, but it has a more favorable condition number. Such a transformation can be attained by utilizing a preconditioner matrix.

One needs to realize that utilizing a preconditioner in a Krylov method adds some extra computational time both in the initial setup and during the iterations for applying it. However, there is a trade-off between the increasing computational time associated with constructing and applying the preconditioner and the advantage of increased convergence rate. We have formulated and examined several preconditioners and selected the one presented in this section for its superior performance.

The preferred preconditioner employs a core-wise grouping of the unknown types. This grouping allows us to associate parts of the Jacobian to the underlying balance equations since they are distinctly separated from one another. However, the original Jacobian matrix, implied in Eq. (7), is slightly rearranged in this preconditioner. The preconditioner can be symbolically written as

$$\overline{\overline{P}} = \begin{bmatrix} \overline{\overline{\tilde{J}}}_{11} & \overline{\overline{0}} \\ \overline{\overline{\tilde{J}}}_{21} & \overline{\overline{\tilde{J}}}_{22} \end{bmatrix}, \tag{18}$$

where

$$\overline{\overline{\tilde{J}}}_{11} = \begin{bmatrix} \overline{\overline{J}}_{11} & \overline{\overline{J}}_{13} \\ \overline{\overline{J}}_{31} & 0 \end{bmatrix}, \tag{19}$$

$$\overline{\overline{\tilde{J}}}_{21} = \begin{bmatrix} \overline{\overline{J}}_{21} & 0 \end{bmatrix}, \tag{20}$$

and

$$\overline{\overline{\tilde{J}}}_{22} = \begin{bmatrix} \overline{\overline{J}}_{22} \end{bmatrix}. \tag{21}$$

When solving the preconditioned matrix system, due to the structure of the matrix, changes in the flux distribution and eigenvalue can now be evaluated first and subsequently used for determining the changes in the thermal hydraulics properties. This is similar to the fission source iterations and thermal hydraulics update in the traditional iterative solver. Porsching's algorithm [7] is used when determining the action of $\overline{\overline{\tilde{J}}}_{11}^{-1}$. The block incomplete LU approximation [8] is used to determine the action of $\overline{\overline{J}}_{11}^{-1}$ within Porsching's algorithm.

## 3. RESULTS

Presented in this section are the results from testing the Newton-BICGSTAB solver utilizing a single loading pattern (LP) as well as multiple LPs. The result from a control rod optimization run is also included to show the performance of the solver for an optimization run. We use this approach to test the robustness of the solver against an increasing number of problems starting from analyzing a single configuration, *e.g.* when performing the preconditioner study to select the final preconditioner to be implemented, to evaluating numerous core configurations during an optimization run.

### 3.1. Evaluation of a Single Loading Pattern

After deciding on utilizing the preconditioned BICGSTAB as the solver for the linearized system of equations, we tested the solver on a single LP to observe the average behavior of the solver over several burnup steps. This exercise is also intended to find a set of stopping criteria which provides the highest speedup while still maintaining the robustness of the solver.

The model used for this study is an 800 assembly GE BWR/6 core. The cycle modeled has 17 burnup steps. The nodal method is employed to solve the three-dimensional, two-group neutron diffusion equation. The three-equation mixture drift flux model is used to treat the two-phase flow in the core. An inlet flow redistribution calculation is also performed to obtain a uniform inlet and exit pressure across the core. In addition, the critical flow search is also executed.

The test is actually divided into two parts. In the first part, the true errors in the neutron flux distribution and the moderator density distributions are used to terminate the overall Newton-BICGSTAB solver. The true error measures the RMS of the relative difference between the iterative values of the flux and moderator density distributions and the reference values. The reference quantities are obtained using the traditional iterative algorithm with a set of extremely tight stopping criteria. The results from this part of the test are presented in Table I. Speedup in Table I denotes the ratio of the CPU times for the traditional solver to Newton-BICGSTAB solver, both utilizing the same true error stopping criteria. We have chosen the true stopping criteria to be $1\times10^{-2}$ for both the neutron flux and moderator density distributions. This implies that the results from these solvers are, on average, within 1% of the true solutions. There are two things to point out from this table. Firstly, the number of outer iterations is reduced by a factor of 12 as a result of the Newton-BICGSTAB implementation. This result numerically confirms the assertion that the Newton's method will provide a higher convergence rate compared to the traditional approach. Secondly, this test shows that the Newton-BICGSTAB solver can provide a speedup of about 1.7 with respect to the traditional solver. In practice the true solutions are not available and, hence, the measure of true error cannot be utilized as the stopping criteria. Therefore, we still need to evaluate the performance of the solver when the real stopping criteria is used.

The second part of the test is aimed to answer this question. The stopping criteria utilized for the overall solver are 1e-4, 5e-4, and 5e-2 for $\varepsilon_k$, $\varepsilon_F$, and $\varepsilon_\rho$ (as defined in Eqs. (15) through (17)), respectively. The results from this part of the test are summarized in Table II. The purpose of examining multiple pairs of Newton and Krylov stopping criteria is two-fold. First, we would like to understand the trade-off between how much we gain in terms of improved agreement in the results against how much we lose in terms of potential speedup. Results in Table II indicate that while there is no apparent improvement in the quality of the results as tighter stopping criteria are employed, significant decreases in the speedup factor are observed. These results suggest that a set of relatively relaxed stopping criteria should be employed by the solver. Second, we would like to examine the possibility of employing a set of dynamic stopping criteria, *i.e.* changing stopping criteria as the iteration progresses. This is what Case 7 in Table II represents. The philosophy used in dynamically changing the stopping criteria is that a more relaxed stopping criteria can be employed early in the iteration, while a tighter stopping criteria is needed when we are close to a converged solution. For example, employing a relaxed BICGSTAB stopping criteria in the first couple of Newton iterations will surely introduce error contaminations to the Newton iteration. However, at this stage since we are still far away from the solution, we believe that these contaminations are much less significant than the error introduced by the first-order approximation of the Newton's method. Later on when we are close to the solution, since the Jacobian matrix is now "pointing" in the right direction and since we would like to reach the solution as fast as possible, ideally in one iteration, we have to solve for the changes in the variables as accurately as we can; therefore, a tighter BICGSTAB stopping criteria is employed. The result shown in the last column of Table II indicates that this approach can produce a higher speedup than the statically determined set of stopping criteria.

**Table I. The Newton-BICGSTAB performance on a single LP using true error stopping criteria.**

| | Case 1 | Case 2 |
|---|---|---|
| **Numerical Information** | | |
| Newton S.C. | 1.00e-2 | 5.00e-3 |
| Krylov S.C. | 1.00e-2 | 5.00e-3 |
| $\varepsilon_{true}^{\phi}$ | 1.00e-2 | 1.00e-2 |
| $\varepsilon_{true}^{\rho}$ | 1.00e-2 | 1.00e-2 |
| Newton/BU Step | 11.53 | 12.94 |
| Newton Iterations | 196 | 220 |
| Traditional Outer Iterations | 2417 | |
| Speedup | 1.67 | 1.55 |
| **Results Comparison (average differences)** | | |
| $k_{eff}$ (in pcm) | 14.84 | 12.77 |
| Flow Fraction | 0.0041 | 0.0032 |
| MFLPD | 5.8235e-4 | 5.7059e-4 |
| MAPRAT | 6.8824e-4 | 6.2355e-4 |
| MFLCPR | 3.9882e-3 | 3.1765e-3 |

**Table II. The Newton-BICGSTAB performance on a single LP.**

| | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 | Case 6 | Case 7 |
|---|---|---|---|---|---|---|---|
| **Numerical Information** | | | | | | | |
| Newton S.C. | 1.00e-2 | 1.00e-3 | 5.00e-4 | 1.00e-2 | 1.00e-3 | 5.00e-4 | Var(1) |
| Krylov S.C. | 1.00e-2 | 1.00e-2 | 1.00e-2 | 5.00e-3 | 5.00e-3 | 5.00e-3 | Var(2) |
| Newton/BU Step | 10.88 | 14.24 | 16.18 | 10.82 | 14.18 | 16.18 | 9.71 |
| Newton Iterations | 185 | 242 | 275 | 184 | 241 | 275 | 165 |
| Traditional Outer Iterations | 2377 | | | | | | |
| Reduction in Outer Iterations(3) | 12.85 | 9.82 | 8.64 | 12.92 | 9.86 | 8.64 | 14.41 |
| Speedup(4) | 1.59 | 1.29 | 1.17 | 1.59 | 1.31 | 1.18 | 1.76 |
| **Results Comparison (average differences)** | | | | | | | |
| $k_{eff}$ (in pcm) | 15.91 | 16.39 | 17.31 | 15.49 | 18.14 | 17.17 | 15.46 |
| Flow Fraction | 0.0044 | 0.0042 | 0.0044 | 0.0041 | 0.0045 | 0.0043 | 0.0042 |
| MFLPD | 6.00e-4 | 6.29e-4 | 6.65e-4 | 6.29e-4 | 7.06e-4 | 6.64e-4 | 5.71e-4 |
| MAPRAT | 7.82e-4 | 8.71e-4 | 8.94e-4 | 8.35e-4 | 9.29e-4 | 8.94e-4 | 7.94e-4 |
| MFLCPR | 4.25e-3 | 4.17e-3 | 4.35e-3 | 4.08e-3 | 4.52e-3 | 4.33e-3 | 4.01e-3 |

(1): Variable stopping criteria: 1E-1 for Newton iteration count less than 6
1E-2 for Newton iteration count between 6 and 10
1E-3 for Newton iteration count above 10
(2): Variable stopping criteria: 1E-1 for Newton iteration count less than 4
1E-2 for Newton iteration count greater than or equal to 4
(3): Reduction in Outer iteration is defined as [# traditional outer iteration]/[# Newton iteration]
(4): Speedup=[Traditional CPU time]/[Newton-BICGSTAB CPU time]

### 3.2. Evaluation of Multiple Loading Patterns

The robustness of the solver utilizing the chosen set of stopping criteria from the single LP study, *i.e.* the dynamically determined stopping criteria, is examined further by testing the solver using multiple LPs. The core model used is an 800 assembly GE BWR/6 core. The cycle modeled has 17 burnup steps. The two-phase thermal hydraulics model with inlet flow redistribution is employed. In addition, the critical flow search is also performed. The nodal method is applied for solving the three-dimensional, two-group neutron diffusion equation. 200 LPs are examined during this study.

The exact Newton-BICGSTAB solver shows an overall speedup of 2.1 to the core simulator with reference to the traditional approach where the red/black line SOR method is utilized as the inner iteration solver. The exact Newton-BICGSTAB solver provides a 12% increase in the overall speedup when compared against the traditional solver utilizing a more computationally efficient preconditioned BICGSTAB method as the inner iteration solver [traditional (BICGSTAB)]. We have also examined the quality of the results and the differences observed are comparable to those shown in Table II.

### 3.3. Results from a Control Rod Pattern Optimization Run

Since the principal objective of this project is to reduce the CPU execution time of an optimization run, it is appropriate to use an optimization run as the ultimate robustness test for the Newton-BICGSTAB solver. The core model used for the optimization is an 800 assembly GE BWR/6 core. For the thermal hydraulics model, in addition to the model used in the previous studies, the spacer-void model which corrects the void fraction to account for the effects of spacer grids is also incorporated. The cycle analyzed has 10 depletion steps.

The results from this study are summarized in Table III. Both optimization runs satisfy the thermal margin constraints imposed and produce similar quality results. With regard to the overall execution time, the Newton-BICGSTAB solver provides a speedup of 1.9. However, after taking into account that a different number of CRPs are evaluated in each run, the average speedup gained by the new solver is around 1.6, which is still consistent with what was observed in the previous studies.

**Table III. CRP Optimization Results.**

| Variable | Constraint | Reference Pattern | Optimized Pattern | |
|---|---|---|---|---|
| | | | Traditional | Newton-BICGSTAB |
| MFLPD | <0.940 | 0.9540 | 0.9417 | 0.9396 |
| MAPRAT | <0.940 | 0.8598 | 0.8848 | 0.8733 |
| MFLCPR | <0.940 | 0.9456 | 0.9298 | 0.9235 |
| Number of CRPs evaluated | | | 209 | 178 |
| Time (sec[1]) | | | 1261 | 665 |

(1): Timing was performed on a 2 GHz DELL 530 workstation.

### 3.4. Results from the Inexact Newton's Method Studies

Some studies on the feasibility of employing an inexact Newton's method to provide an additional speedup to the Newton-BICGSTAB solver have been initiated. The objective of this study is two-fold. Firstly, we would like to show that a proper utilization of the inexact Newton's method can provide an additional speedup to the exact Newton's method. Secondly, we would like to show some promising results from this study to motivate further research in this area.

There are three types of inexact Newton-BICGSTAB solvers examined in this study. The first type is a straight-forward implementation of Chord and Shamanskii's methods [4]. Results from this test show that the most efficient Newton-Shamanskii solver can provide an additional 6%

increase in speedup over the exact Newton-BICGSTAB solver. The second type is called a hybrid Chord-Shamanskii method, where instead of setting up the Jacobian matrix only once at the beginning of the Newton iteration (as done in the Chord method), an additional Jacobian matrix update is performed after *m* Newton iterations. This approach is formulated to reduce the total number of Newton iterations of the Chord method while avoiding too many Jacobian matrix updates of the Shamanskii's method. A parametric study is performed to determine the optimum point for performing the second Jacobian matrix update. The optimum performance of this family of inexact Newton-BICGSTAB solvers provides an additional 11% increase in speedup over the exact Newton-BICGSTAB solver. The third type of inexact Newton-BICGSTAB solver is obtained simply by setting to zero value the elements of the Jacobian matrix below a certain cut-off value. Assuming that this action does not alter the direction of the search, this approach reduces the cost for matrix-vector multiplication within the BICGSTAB algorithm and also minimizes the cost for setting up the Jacobian matrix itself. Testing this approach in conjunction with the optimum hybrid Chord-Shamanskii method, we observe increase in the overall speedup of 20% (a single LP test) and 14% (multiple LPs test) over the exact Newton-BICGSTAB solver.

## 4. CONCLUSIONS

The implementation of a robust and computationally efficient Newton-BICGSTAB solver successfully reduces the CPU execution time for the FORMOSA-B BWR core simulator. Both exact and inexact variants of the Newton-BICSTAB solver have been examined. The average speedup for the exact Newton-BICGSTAB solver is around 1.7. The inexact Newton-BICGSTAB solver can provide a speedup up to 2.1. The robustness of the solver has been tested against numerous core configurations and consistent results have been observed each time.

## ACKNOWLEDGMENTS

This work is sponsored by the Department of Energy Nuclear Engineering Education Research Grant No. 99ID13773.

## REFERENCES

1. Brian R. Moore, Paul J. Turinsky, and Atul A. Karve, "FORMOSA-B: A Boiling Water Reactor Incore Fuel Management Optimization Package," *Nucl. Tech.*, **126**, pp. 153-169 (1999).
2. Atul Karve and Paul J. Turinsky, "FORMOSA-B: A Boiling Water Reactor Incore Fuel Management Optimization Package II," *Nucl. Tech.*, **131**, pp. 48-68 (2000).
3. D. F. Kastanya, *Implementation of a Newton-Krylov Iterative Method to Address Strong Non-Linear Feedback Effects in FORMOSA-B BWR Core Simulator*, Doctorate Dissertation, N.C. State University (2002).
4. C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia (1995).

5. Y. Saad and M. H. Schultz, “GMRES: A Generalized Minimal Residual Algorithm for Solving Non-Symmetric Linear Systems,” *SIAM J. Sci. Stat. Comput.*, Vol. 7, No. 3, pp. 856-869 (1986).
6. H. A. Van der Vorst, “Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Non-Symmetric Linear Systems,” *SIAM J. Sci. Stat. Comput.*, Vol. 13, No. 2, pp. 631-644 (1992).
7. T. A. Porsching, J. H. Murphy, and J. A. Redfield, “Stable Numerical Integration of Conservation Equations for Hydraulic Networks,” *Nucl. Sci. Eng.*, **43**, pp. 218-225 (1971).
8. H. G. Joo, *Domain Decomposition Methods for Nonlinear Nodal Spatial Kinetics*, Doctorate Dissertation, Purdue University (1996).