M&C 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering, Jeju, Korea, April 16-20, 2017, on USB (2017)

Jacobian-Free Newton-Krylov Coupling Methods for Nuclear Reactors

Erik D. Walker[a], Benjamin Collins[b], and Jess C. Gehin[b]

[a]*University of Tennessee – Knoxville, TN 37996 – ewalk@utk.edu*
[b]*Oak Ridge National Laboratory – Oak Ridge, TN 37831 – collinsbs@ornl.gov – gehinjc@ornl.gov*

Abstract – *The applicability of a Jacobian-Free Newton-Krylov (JFNK) method for use in coupled reactor problems is examined. The ability for JFNK to simultaneously solve coupled systems of equations offers the potential for significant computational speedup over traditional solver methods. Simplified 1D and infinite homogeneous medium problems were devised to test the effectiveness of JFNK on coupled problems. These problems demonstrated that updating the absorption cross section from within the JFNK solver can decrease computational times significantly. Additionally, a JFNK eigenvalue solver was fully implemented into the neutron transport code MPACT. Problems tested using this solver showed that JFNK offers, at worst, comparable performance as long as an appropriate preconditioner is used.*

I. INTRODUCTION

The solution of a coupled multiphysics problem is most often solved using a fixed-point, or Picard, iteration in which each set of physics is solved separately, and the resulting outputs are passed between each solver. Generally, once two different codes are coupled into one, one set of physics is solved first while the other set of physics is solved only after convergence. These separate solvers treat one another as black boxes in that they only use information from the other as an input and do not share information between each other before convergence. In nuclear reactor applications, coupling neutronics to thermal-hydraulics (TH) is no exception. A typical workflow is as follows: first, the neutronics equations are solved to calculate the fluxes in the problem, which in turn are used to calculate power. Then a TH solver takes these powers and uses them to determine the temperatures throughout the problem. These temperatures are then passed back to the neutronics solver where they are used to calculate new cross sections. This cycle continues until both the neutronics and TH solutions are stable. While this fixed-point method is easiest to implement and solve, it suffers from a slow convergence rate. There also is no guarantee that the solution will converge at all. This is due to the fact that only the local convergence within each solver is known and tested. The overall global convergence is not actually known but is assumed from the local convergence of each solver. Therefore, it is possible under certain circumstances that global convergence is never reached, although each set of physics is locally converged.

Alternatives to a Picard iteration scheme are available that may offer improvements. Newton-based iterative methods that utilize a Jacobian to provide more information have a quadratic convergence rate and are globally convergent [1]. Certain Newton-based methods avoid having to form the Jacobian, which is desirable when it is either expensive or impossible to compute. These methods are referred to as Jacobian-Free Newton-Krylov (JFNK) methods [2].

1. JFNK Methods

The ability to accurately predict coupled system behavior in a reasonable amount of time is critical for both steady state and transient calculations. All research to date on coupled JFNK systems have been performed with either diffusion or nodal solvers, or were performed on the fine transport mesh with little performance improvement [1] [3] [4] [5] [6]. The unique contribution of this work is the fact that the nonlinear JFNK solver will be applied on the low-order condensed Coarse Mesh Finite Difference (CMFD) equations to further accelerate the solution. In order to achieve improved performance, methods for updating the cross sections on the coarse mesh will need to be investigated to avoid having to perform the computationally expensive fine mesh calculations used to update the cross sections. This method can also be implemented to solve the critical boron search problem and transient problems, in addition to being a k-eigenvalue solver.

The core of any JFNK solver is Newton's Method. Newton's method is an iterative method for finding the roots of a real-valued function, i.e., $f(\boldsymbol{x}) = 0$. If the current root approximation is given by $\boldsymbol{x}_k$, and the subsequent approximation is given by $\boldsymbol{x}_{k+1}$, then the method can be derived from a Taylor series expansion of $f(\boldsymbol{x}_{k+1})$ about $\boldsymbol{x}_k$:

$$f(\boldsymbol{x}_{k+1}) = f(\boldsymbol{x}_k) + f'(\boldsymbol{x}_k)(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + \cdots . \quad (1)$$

This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC0500OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for the United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

Since it is desired that as k gets large, $f(\boldsymbol{x}_{k+1})$ will approach zero, the right hand side of Equation 1 is set to zero and the higher order terms are ignored, leading to

$$0 = f(\boldsymbol{x}_k) + f'(\boldsymbol{x}_k)(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)\,. \tag{2}$$

Solving for $\boldsymbol{x}_{k+1}$ yields Newton's method:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{f(\boldsymbol{x}_k)}{f'(\boldsymbol{x}_k)}\,. \tag{3}$$

Provided the initial guess is sufficiently close to a root, Newton's method has a quadratic convergence rate which is a desirable feature in numerical linear algebra. Newton's method can also be extended to solve an m-dimensional system of nonlinear equations, $F(\boldsymbol{x}) = 0$, where x is now a vector of length m. This version of Newton's method has the same form as Equation 2, but is usually written as the linear system

$$J(\boldsymbol{x}_k)\delta\boldsymbol{x}_k = -F(\boldsymbol{x}_k)\,, \tag{4}$$

where $\delta\boldsymbol{x}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$ and $J(\boldsymbol{x}_k) \equiv F'(\boldsymbol{x}_k)$ is the Jacobian matrix

$$J(\boldsymbol{x}_k) = \begin{bmatrix} \frac{\partial F_1(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_1} & \cdots & \frac{\partial F_1(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_m(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_1} & \cdots & \frac{\partial F_m(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_m} \end{bmatrix}. \tag{5}$$

Therefore the vector $\delta\boldsymbol{x}_k$ is the solution to Equation 4 and is added to the current root approximation, $\boldsymbol{x}_k$, in order to obtain the approximation at the next step.

A JFNK method solves the nonlinear system of equations in Equation 4 without explicitly forming the Jacobian. Instead, in certain Newton-Krylov methods, the explicit elements of the Jacobian, J, are not needed to be known; only the action of the matrix on a vector, $\boldsymbol{v}$, is required. Therefore, to avoid forming the Jacobian entirely, a finite difference is used to approximate this matrix-vector product using

$$J(\boldsymbol{x})\boldsymbol{v} \approx \frac{F(\boldsymbol{x}+\varepsilon\boldsymbol{v}) - F(\boldsymbol{x})}{\varepsilon} \tag{6}$$

where ε is a small perturbation. This is the basis for JFNK methods, and the error in this approximation is proportional to ε. While JFNK has the obvious advantage of applying the quadratically convergent Newton's method on a nonlinear system of equations without the need to form or store the Jacobian, it does have a drawback: it is only feasible on large scale problems with the use of an effective preconditioner [2]. Therefore the study of preconditioners will be a critical part of this work.

2. Preconditioning

In order to improve the convergence of JFNK preconditioners are often used. Preconditioners help speed up the JFNK method by reducing the number of linear iterations needed to reach convergence. Through appropriate preconditioning, the convergence of JFNK can be greatly enhanced.

While JFNK can use either right or left preconditioning, right preconditioning is most often used because left preconditioning changes the norm of the residual, which is how convergence of the linear solver is measured [2]. Therefore, only right preconditioning was examined in this work. Using right preconditioning, the Jacobian-matrix approximation from Equation 6 becomes

$$J(\boldsymbol{x})\boldsymbol{P}^{-1}\boldsymbol{v} \approx \frac{F(\boldsymbol{x}+\varepsilon\boldsymbol{P}^{-1}\boldsymbol{v}) - F(\boldsymbol{x})}{\varepsilon} \tag{7}$$

where $\boldsymbol{P}^{-1}$ is the inverse of the preconditioning matrix.

While the usefulness of JFNK lies in its ability to not need an explicitly formed Jacobian, effective preconditioners typically require some knowledge of the Jacobian. However, the preconditioner can use a much simpler version of the Jacobian and can use approximations to make its formulation easier.

II. DESCRIPTION OF THE ACTUAL WORK

As a proof of concept, two simplified reactor problems were developed: a one-group, one-dimensional homogeneous slab problem, and a multigroup infinite homogeneous medium problem. The one-dimensional slab problem looks at the potential impact of implementing a JFNK nonlinear solver on a problem with spatial dependencies while the infinite homogeneous problem examines the impact on a problem with energy dependence. These problems were run in serial using a CMFD accelerated Method of Characteristics (MOC) code written by the author. The linear system of equations formed in JFNK was solved using a GMRES solver also written by the author.

In addition, JFNK was implemented as an eigenvalue solver in MPACT [7]. MPACT is a MOC neutron transport code currently under development by the Consortium for Advanced Simulation of Light Water Reactors (CASL). This JFNK eigenvalue solver was tested on three simplified reactor problems: a 2D pincell, a 3D fuel rod, and a 7x7 assembly problem.

1. One-Dimensional, One-Group Homogeneous Slab

The one-dimensional (1D), one-group homogeneous slab problem is a common toy problem in reactor analysis

from which important physics can be gleaned. This simplified problem allows for methods to be developed and tested without the added complexity and computational burden of a multidimensional heterogeneous problem with multiple energy groups. The slab is finite in the x dimension with a width L, but is infinite in the y and z dimensions, thus removing their dependencies. The boundary conditions for this problem are those of a vacuum, meaning that there is no incident neutron flux on the edges of the problem. A depiction of this 1D slab problem is shown in Fig. 1.

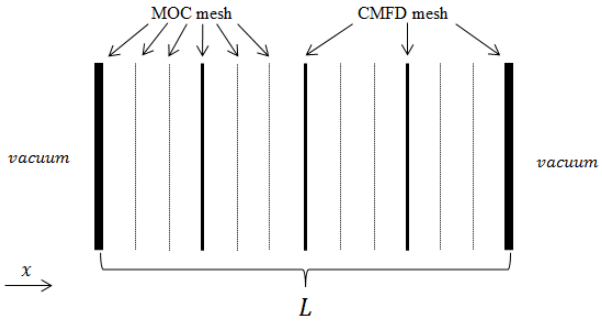

Fig. 1. 1D homogeneous slab problem

For this problem, a simple 1D steady state heat conduction model was used for TH feedback. For this we assume a fuel pin geometry with a fixed surface temperature and constant thermal conductivity. The temperature dependence of the cross section for this problem is:

$$\Sigma(T) = \Sigma_0(T_0)\sqrt{\frac{T_0}{T}}, \tag{8}$$

where T_0 and Σ_0 are the reference temperature and corresponding cross section, respectively.

When performing the temperature updates in JFNK, these calculations are performed using the coarse CMFD mesh. Therefore, the temperatures must first be condensed from the fine MOC mesh to the coarse CMFD mesh. This uses a straightforward average of the fine regions within a coarse mesh. When projecting the coarse temperature back onto the fine mesh, the temperature is assumed the same for every fine region within a coarse mesh. Fig. 2 shows the convergence of the coupled neutronics-TH problem when the temperature and cross section updates are performed in the MOC iteration compared to when they are done in the JFNK iterations.

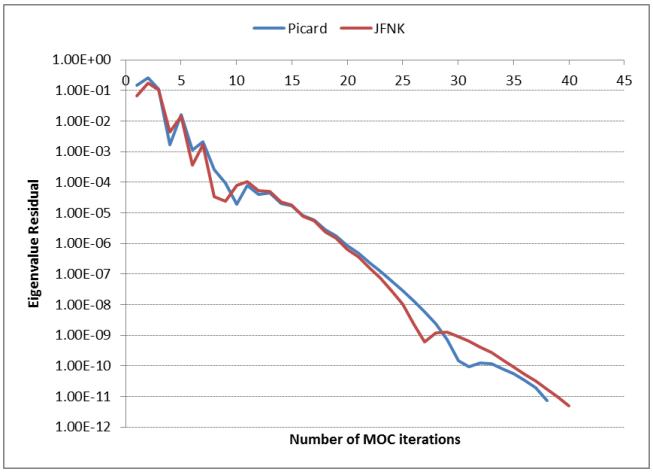

Fig. 2. Eigenvalue convergence between JFNK and standard Picard iteration schemes

As seen in Fig. 2, performing the TH update within the JFNK iterations does not offer a significant speedup compared to the standard Picard iteration. This could be due to the fact that no preconditioner was used, or that this problem is too simple and easy to solve so the costly slowdown of using Picard is not apparent. Whether this is the case or not, the JFNK solver does not hurt the convergence in this problem, and now the method is more robust because the global residual is tested for convergence instead of each local residual.

2. Infinite Homogeneous Medium

While the 1D slab problem in the previous section had a spatial dependence because of its finite size, an infinite homogeneous medium problem has no spatial dependence. Instead, an energy dependence was incorporated through the use of a multigroup framework. A 47 group library was used for this problem. Since the problem is thought of as an infinite material with constant properties, the cross sections are uniform throughout the problem. Because there is no longer a spatial dependence, MOC-CMFD is no longer needed and instead is replaced with a cross section table lookup. The 47 group cross sections were generated from simple 2D pin cell calculations run at varying fuel temperatures from 565 K to 1500 K in five degree increments. Once the pin cell calculations were finished, the converged homogenized cross sections were written to a large file. This was then used as a table lookup to determine the cross sections in the infinite material for a given temperature. A linear interpolation was used to determine the cross section between data points. The temperature feedback used in this problem was chosen using the solution of one of the 2D pin cell problems such that the problem would converge to a predetermined solution. Arbitrarily choosing the 1365 K case and using its eigenvalue leads to the TH feedback used:

$$T = 565K + 688.534 \times k_{eff} \quad (9)$$

where k_{eff} is the dominant eigenvalue. While it is not a robust method, Equation 9 ensures that as the eigenvalue increases or decreases, the temperature follows suit. It also has the added benefit of knowing what the solution should converge to upon completion, allowing for error checking.

It is possible to accelerate the convergence of the temperature and multigroup fluxes if some information about the cross section dependence on temperature is used in the JFNK iterations. Currently, while Newton's method within JFNK is iterating towards a solution, the cross sections are held constant. Even though the temperature is changing, the cross sections aren't updated until after the JFNK solution has converged. In order to give the Newton's method the ability to update the cross sections, the cross section derivative with respect to temperature, $d\Sigma/dT$, is calculated at the current temperature using a forward finite difference approximation. These derivatives are then passed into JFNK and are used to linearly extrapolate the cross sections as the temperature is converging. The comparison of the convergence of these two methods is shown in Fig. 3.

Fig. 3 Convergence of table lookup scheme vs the addition of a linear update

As seen in Fig. 3, the inclusion of a linear update of the cross sections in the Newton iteration is very effective. The linear update curve appears to have quadratic convergence while the table lookup only curve is linear. While this speedup appears to come without much effort, one must think of larger full-scale reactor problems. In this simple problem, there is only one material and therefore one temperature. Using this method on a full-core problem would be prohibitively expensive because one would have to store each cross sections derivative for every flat source region in the core. Therefore, an analysis was performed to determine how to achieve the most acceleration without the large memory requirements.

First, to determine which of the cross sections had the largest impact on speedup, each was linearly updated individually, while the others were only updated outside of the JFNK iteration. The results from this test are given in Fig. 4. The curves from Fig. 3 were included in Fig. 4 because they act as outer limits for this study. It is clear from Fig. 4 that updating χ, $\nu\Sigma_f$, and Σ_s has little, if any, effect on the convergence rate. Updating Σ_a however, has a very significant impact on the problem convergence. Therefore, only the absorption cross section update will be considered for the remainder of this section.

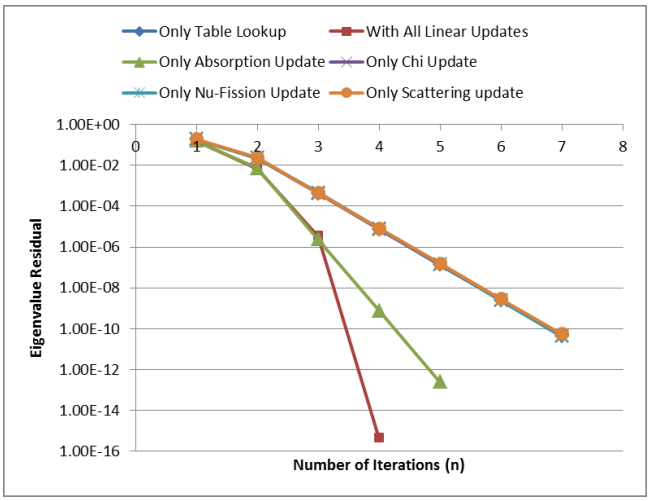

Fig. 4 Convergence plots for each cross section being linearly updated independently

While removing the other linear cross section updates does relieve some of the memory burden, having to determine the absorption cross section derivative for every region in the core might still be too expensive. Therefore, a study was performed to determine how exact the absorption cross section derivatives need to be and what approximations could be made. The first approximation made is to remove the temperature dependence of the derivative and use an average value per group instead. This was done by calculating $d\Sigma_a/dT$ for all temperatures for each group. An average was then calculated by summing all of the derivatives for a given group and then dividing by the number of derivatives summed. This approximation is referred to as the Average Groupwise Derivative approximation. The next approximation tested looked at removing the group dependence and only using a one-group temperature dependent derivative. The first step was to collapse the multigroup cross sections to a one-group cross section. Then $d\Sigma_a/dT$ was calculated using a forward finite difference for a given temperature. This approximation is referred to as the One-Group Derivative approximation. The final simplification examined involves combining the two previous approximations to remove both the temperature and group dependencies on the derivative.

Like the One-Group Derivative approximation, the cross sections are initially collapsed to one-group. Then like the Average Groupwise Derivative approximation, these one-group cross sections are used to calculate the derivatives using a forward finite difference, which are then averaged together. Therefore, $d\Sigma_a/dT$ becomes simply a constant value. This method is called the Average One-Group Derivative approximation. Each of these different methods were implemented independently and their convergence plots are shown in Fig. 5.

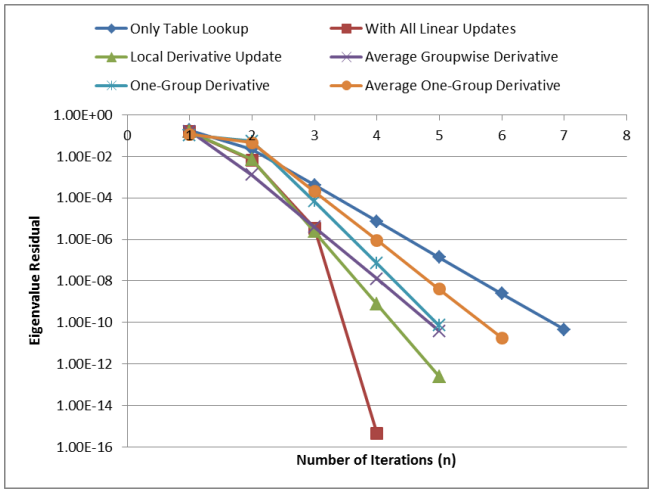

Fig. 5 Convergence plots for different absorption cross section derivative approximations

As seen in Fig. 5, all possible approximations are improvements upon the method that only updates the cross sections after the fluxes have converged. As expected, using a constant value from the Average One-Group Derivative approximation performed the worst, but still performed better than using no approximation and leaving the cross sections constant during the JFNK routine. The Average Groupwise Derivative and the One-Group Derivative approximations both had similar convergence behavior, and performed better than the constant Average One-Group Derivative approximation. Calculating the Local Derivative for every absorption cross section performed the best, but might be limiting when used on a larger problem. Therefore, depending on computational requirements, more than one of these approximations might be feasible on largescale calculations and should be investigated.

Finally it needs to be addressed how the cross section derivatives will be calculated in a real problem where the converged solution isn't known beforehand. One possible method that will be investigated is similar to what was done with this infinite homogeneous problem. A simple 1D pin cell problem could be run on the fly each Newton iteration. The results of which could be used to calculate approximations of $d\Sigma_a/dT$. Another possible option is to formulate a derivative estimator in CMFD during the homogenization process which would then be passed into the JFNK routine. Finally, it will be investigated whether some predetermined constant derivative values are possible, avoiding any additional calculations.

3. Eigenvalue Solver in MPACT

After learning important lessons from the application of JFNK to the simplified problems described above, it was implemented as an eigenvalue solver in the MPACT neutronics code. The implementation of the JFNK and GMRES solvers were provided using the Portable, Extensible Toolkit for Scientific Computation, or PETSc [8].

Like the 1D slab problem, MPACT solves the generalized eigenvalue problem given by

$$\mathbb{M}\boldsymbol{\phi} = \frac{1}{k_{eff}}\mathbb{F}\boldsymbol{\phi}, \tag{10}$$

where $\boldsymbol{\phi}$ is the flux vector containing the scalar fluxes for each coarse mesh region in CMFD, $\mathbb{M}$ is the migration matrix, $\mathbb{F}$ is the fission matrix, and k_{eff} is the dominant eigenvalue.

To test the effectiveness of using JFNK as an eigenvalue solver, three test problems were used: a 2D pincell in a square channel, a 3D fuel pin with 28 axial regions, and a 3D 7x7 assembly, also with 28 axial regions. The specific details of these problems are not necessary; they were simply chosen to test JFNK on systems of varying size. The pincell problem is 32x32, the 3D fuel pin is 896x896, and the 3D assembly is 4480x4480.

As a benchmark, these same problems were run using a default Power Iteration (PI) eigenvalue method performed on the CMFD mesh. At first JFNK was implemented without a preconditioner. These results are given in Table I.

Table I. Unpreconditioned comparison of PI and JFNK eigenvalue solvers

Problem		# MOC Iterations	k-eff	Time
Pincell				
	PI	15	1.1702266	00:01.2
	JFNK	15	1.1702266	00:01.8
3D Fuel Pin				
	PI	15	1.1544792	00:21.3
	JFNK	15	1.1544990	02:38.4
7x7 Assembly				
	PI	10	1.1000484	00:43.5
	JFNK	20	1.1000817	08:36.3

While the JFNK solver converges to the same eigenvalue as the PI method, its computation time is much greater for problems of a larger size. For the largest problem tested, the 7x7 assembly, it took twice as many transport

sweeps to achieve convergence, demonstrating the instability of the unpreconditioned system.

To improve on these results, a preconditioner was used to speed up convergence. To test the ability of JFNK to function without knowing the true Jacobian, an approximate Jacobian of Equation 10 was selected to simply be $\mathbb{M}$. Using $\mathbb{M} = \boldsymbol{P}$, Equation 7 was used to precondition the system. These results are given in Table II.

Table II. Preconditioned comparison of PI and JFNK eigenvalue solvers

Problem		# MOC Iterations	k-eff	Time
Pincell				
	PI	15	1.1702266	00:01.2
	JFNK	15	1.1702266	00:01.6
3D Fuel Pin				
	PI	15	1.1544792	00:21.3
	JFNK	15	1.1544848	00:24.1
7x7 Assembly				
	PI	10	1.1000484	00:43.5
	JFNK	9	1.1000523	00:42.5

While the use of an approximate preconditioner sped up the convergence for all problems, its effect is most noticeable on the larger problems. In fact, the 7x7 assembly problem took one fewer transport sweep than was required by the PI solver.

III. RESULTS

In order to provide background on the proposed approach, a JFNK based nonlinear solver was successfully written and applied to two separate simple problems. It was found that for the 1D one-group homogeneous slab problem, JFNK offered no significant speedup to the convergence of the eigenvalue. This potentially could have been the result of the problem being too simple and easy to solve. Therefore the shortcomings of the standard Picard iteration did not become apparent. However, even though no speedup was achieved, the overall robustness of the convergence behavior was improved because JFNK solves for the global residual, rather than each local residual. However, when JFNK was applied to the infinite homogeneous multigroup problem a significant reduction in the number of transport iterations was observed. When all cross sections were updated linearly within the nonlinear Newton iteration, a speedup of 43% was see compared to leaving the cross sections constant as the temperature changes within the JFNK update. To avoid the computational expense of calculating and storing the cross section derivatives for every region of a large scale problem, alternative methods were tested that wouldn't carry the same memory burden. It was found that updating only the absorption cross section and leaving all other constant carries the most improvement. Simplifications of this absorption update were performed and it was discovered that removing the temperature dependence by averaging the absorption cross section derivative over all temperatures was the second best method. This was followed closely by removing the group dependence and using just a one-group temperature dependent derivative. Finally, both the temperature and energy dependencies were removed and a single constant value was used for all updates. While this method was the worst of all those tested, it was still an improvement on the standard Picard iteration scheme. This shows promise that even if JFNK is used to update the cross sections with a constant value in the nonlinear Newton iterations, it would lead to a speedup over the standard method.

When tested on three simple problems, the JFNK solver implemented in MPACT showed long computational times when used in an unpreconditioned manner. However, when an approximate to the Jacobian was selected to be the migration matrix, $\mathbb{M}$, the JFNK solver performed comparably to the PI solver. In one problem, the 7x7 assembly case, JFNK was slightly faster than PI and required one fewer transport sweep.

IV. CONCLUSIONS

When looking at all of the problems tested in this paper, an overall benefit to using JFNK for coupled problems becomes apparent. Other than unpreconditioned cases, JFNK performed, at worst, comparably to the methods that are currently in use in production transport codes. But when the speedup seen from the infinite homogeneous problem is taken into account, the potential for JFNK to outperform current methods is seen. Since, as an eigenvalue solver, JFNK is comparable to PI, the addition of the nonlinear cross section updates within JFNK should offer significant speedup. Therefore, the overall use of JFNK appears to offer benefits over that of the standard Picard iteration approach.

While these preliminary results are promising, much work is still being done to fully implement JFNK into MPACT for coupled multiphysics problems. Appropriately updating the cross sections on the CMFD mesh from within JFNK is still an active area of research. In addition, selecting an appropriate block Jacobian for the coupled problem is also being investigated. Finally, to fully implement this method, JFNK will include a parallel capability to run large, full-scale reactor core problems on large computer systems.

ACKNOWLEDGMENTS

This research was supported by the Consortium for Advanced Simulation of Light Water Reactors (www.casl.gov), an Energy Innovation Hub (http://www.energy.gov/hubs) for Modeling and Simulation

of Nuclear Reactors under U.S. Department of Energy Contract No. DE-AC05-00OR22725.
This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

REFERENCES

1. S. HAMILTON ET AL, "An assessment of coupling algorithms for nuclear reactor core physics simulations," *Journal of Computational Physics*, vol. 311, pp. 241-257 (2016).
2. D. KNOLL and D. KEYES, "Jacobian-free Newton-Krylov methods: a survey of approaches and applications," *Journal of Computational Physics*, vol. 193, pp. 357-397 (2004).
3. Y. XU, "A Matrix-Free Newton/Krylov Method for Coupling Complex Multiphysics Subsystems," Purdue University, Ph.D. Thesis (2004).
4. A. WARD, "A Newton-Krylov Solution to the Coupled Neutronics-Porous Medium Equations," The University of Michigan, Ph.D. Thesis (2012).
5. D. F. KASTANYA, "Implementation of a Newton-Krylov Iterative Method to Address Strong Non-Linear Feedback Effects in FORMOSA-BWR Core Simulator," North Carolina State University, Ph.D. Thesis (2002).
6. B. HERMAN, "Monte Carlo and Thermal Hydraulic Coupling using Low-Order Nonlinear Diffusion Acceleration," Massachusetts Institute of Technology, Ph.D. Thesis (2009).
7. MPACT Team, MPACT Theory Manual, Version 2.0.0, Tech. Rep. CASL-U-2015-0078-000, Oak Ridge National Laboratory and University of Michigan (2015).
8. S. BALAY et al., "PETSc User Manual," ANL-95/11 – Revision 3.5, Argonne National Laboratory, http://www.mcs.anl.gov/petsc (2014).