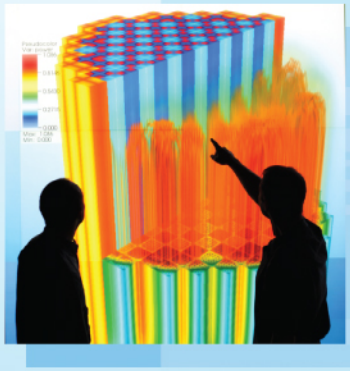
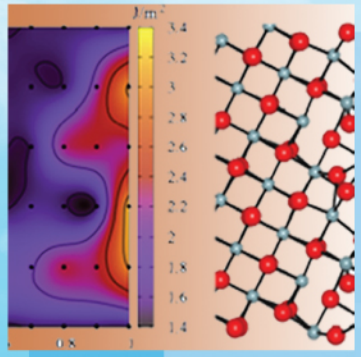
CASL-U-2014-0081-000

Power uprates
and plant life extension



Engineering design
and analysis

Science-enabling
high performance
computing



Fundamental science

Plant operational data

Consortium for Advanced Simulation of LWRs

L3:PHI.CMD.P8.01 Evaluation of Coupling Approaches

M. Berrill, K. Clarno, S. Hamilton
Oak Ridge National Laboratory

Roger Pawlowski
Sandia National Laboratory

May 1, 2014

U.S. DEPARTMENT OF ENERGY | Nuclear Energy

CASL-U-2014-0081-000

CASL Milestone Report L3:PHI.CMD.P8.01 Evaluation of Coupling Approaches

M. Berrill, K. Clarno, S. Hamilton, and R. Pawlowski

May 1, 2014

Abstract

This paper evaluates the performance of multiphysics coupling algorithms on a light water nuclear reactor core simulation. The simulation couples a k-eigenvalue neutron transport code and a finite element/control volume thermal hydraulics code. We compare Picard iteration (block Gauss-Seidel), Anderson Accelerated Picard iteration, and multiple variants of preconditioned Jacobian-free Newton-Krylov (JFNK). The performance of the methods are evaluated over a range of energy group sizes, boron concentrations and core power levels. A novel physics-based Jacobian-vector product approximation has been developed to demonstrate the effectiveness of JFNK as a solution method for core calculations.

1 Introduction

This report summarizes the research performed in Milestone L3:PHI.CMD.P8.01 "Evaluation of Coupling Approaches". This is a follow on study to milestone L3:VRI.PSS.P7.06 "Numerical Studies of Subchannel-Neutronics Coupling". The goal of this work was to investigate the use of alternate coupling algorithms in terms of efficiency, accuracy and robustness. The results are meant to provide CASL with guidance on coupling strategies.

The current VERA core simulator capabilities all rely on Picard iteration. Examples include the CTF/Insilico coupling, the CTF/Insilico/Peregrine coupling and the CTF/MPACT coupling. While this choice tends to be the fastest code development path to solving coupled physics, the Picard algorithm suffers from drawbacks including a lack of global convergence and, at best, a q-linear convergence rate [6]. This typically requires user defined relaxation methods to achieve convergence. Newton-based methods, however are shown to be globally convergent and with q-quadratic convergence rate. Newton-based methods typically require more invasive access to application codes including residual and sensitivity information. To avoid having to implement a Jacobian, the Jacobian-free Newton-Krylov (JFNK) method [7] is employed to solve the coupled system.

CASL-U-2014-0081-000

While JFNK has been successfully applied in many areas, it is an open research topic as to whether JFNK can be competitive with Picard iteration on realistic core simulator applications.

In the previous milestone, L3:VRI.PSS.P7.06, the basic capabilities were implemented to compare JFNK, Picard and Anderson Accelerated Picard. The initial results were mixed with no method standing out. This capability was "Exploratory" under the TriBITS lifecycle model [2]. The simulations were limited to single pin and the JFNK algorithm was not optimized for the physics. To perform a fair comparison against JFNK, this milestone included the development of novel Jacobian-vector approximations that yielded significant performance gains. This approximation is critical as the number of energy groups in the neutronics calculation is increased.

JFNK-based couplings require residuals and preconditioners. This is not a trivial requirement, and therefore the codes for this analysis were chosen based on the capability to supply residuals. Significant work would be required to obtain residuals from the current core simulator pieces. The Advanced Multi-Physics (AMP) framework was chosen to provide thermal modeling of the fuel pins and a subchannel model. The Insilico code was chosen for the radiation transport model. The codes were coupled using AMP and Trilinos/Thyra vector abstractions using the nonlinear solvers in Trilinos/NOX.

Section 2 describes the physics models and application codes. Section 3 describes the coupling algorithms. Section 4 contains the results and Section 5 presents conclusions and future work.

2 Physics Models

In this document we consider solution of multiphysics problems involving coupling between neutron transport and heat transfer. For nuclear reactor problems, the standard formulation of the neutron transport equation is the k-eigenvalue problem

$$\begin{aligned}\hat{\Omega}\cdot\nabla\psi(\vec{r},E,\hat{\Omega})+\sigma(\vec{r},E,T)\psi(\vec{r},E,\hat{\Omega}) = \\ \int_0^\infty dE'\int_{4\pi} d\hat{\Omega}'\,\sigma_s(\vec{r},E'\to E,\hat{\Omega}'\to\hat{\Omega},T)\psi(\vec{r},E',\hat{\Omega}')+ \\ \frac{1}{k}\chi(\vec{r},E)\int_0^\infty dE'\int_{4\pi} d\hat{\Omega}'\,\nu\sigma_f(\vec{r},E',T)\psi(\vec{r},E',\hat{\Omega}')\,, \quad (1)\end{aligned}$$

where $\hat{\Omega}$ is the direction of particle travel, E is the particle energy, T is the temperature of the background material, σ is the total cross section, σ_s is the scattering cross section, $\nu\sigma_f$ is the neutron production cross section, and χ is the fission spectrum. The goal is to find the largest value of the eigenvalue k and the corresponding eigenvector ψ. Because Eq.(1) represents an eigenvalue problem, the vector ψ has no explicit magnitude. We choose a natural normalization by setting the global heat generation rate (due to nuclear fission occurring in the

fuel) to a pre-defined value, i.e.

$$\int dV \int_0^\infty dE \int_{4\pi} d\hat{\Omega}\, \kappa\sigma_f \psi = P^* \,. \tag{2}$$

As noted in Eq. (1), the cross sections are dependent on the temperature of the media, T. Thus, for a reactor not operating at a constant temperature it is also necessary to solve a heat transfer equation with fission providing the thermal source, i.e.

$$-\nabla \cdot K(T)\nabla T = \int_0^\infty dE \int_{4\pi} d\hat{\Omega}\, \kappa\sigma_f(E)\psi(E,\hat{\Omega}) \,, \tag{3}$$

where K is the material thermal conductivity and κ is the heat generated per fission event. It should be noted that the concise notation on the left hand side of (3) hides significant complexity in that it involves not only conduction in the solid fuel and clad regions, but additionally effective heat transfer between the fuel pellets and the clad (generally modeled using a gap conductance model) as well as convective heat transfer between the outer surface of the cladding and the surrounding coolant and potentially models describing the fluid flow itself.

More specifically, the LHS operator includes the solution of multi-domain thermal diffusion coupled to a subchannel physics model. Each fuel pellet and the clad for each rod can be treated as an individual domain over which we solve the thermal diffusion equation:

$$-\nabla \cdot K(T)\nabla T = q(\mathbf{x}) \,, \tag{4}$$

where K is the thermal conductivity and $q(\mathbf{x})$ is the thermal source due to heat generated from nuclear fission in each pellet. The individual domains are coupled through thermal conduction between the pellets and across the gap between the pellets and the clad. The exterior of the clad is then coupled to the coolant through the subchannel model that solves the conservation of mass, momentum, and energy equations:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \tag{5}$$

$$\frac{\partial \rho v_i}{\partial t} = -\nabla \cdot (\rho v_i \vec{v}) + (-\nabla p + \nabla \cdot \vec{\tau}) - \vec{g} \tag{6}$$

$$\frac{\partial U}{\partial t} + \nabla \cdot (U\vec{v}) = -p\nabla \cdot \vec{v} + \Phi + \nabla K(T)\nabla T + \dot{q} \tag{7}$$

where ρ is the mass density, $\vec{v}$ is the velocity, p is the pressure, $\vec{g}$ is the force exerted by gravity, $\vec{\tau}$ is the viscosity tensor, U is the internal energy density, Φ is the dissipation function, and $\dot{q}$ is the thermal source. Note that the internal energy density is related to the enthalpy density through $U = h - p$. We solve these equations using a two-equation approximation in which we assume that the coolant flow is only in the z-direction and neglect thermal diffusion between

the channels. Assuming steady-state this reduces to:

$$\frac{\partial \rho v^2}{\partial z} + \frac{\partial p}{\partial z} - \frac{\partial \tau}{\partial z} = -g\left(-\nabla p + \nabla \cdot \tau\right) - g\,, \tag{8}$$

$$\frac{\partial h v}{\partial z} = -v\frac{\partial p}{\partial z} - \Phi + \frac{\partial}{\partial z}\nabla K(T)\frac{\partial T}{\partial z} + \dot{q}\,, \tag{9}$$

In our subchannel model we therefore require two independent variables per grid point and choose enthalpy and pressure. The thermal source consists primarily of convective heat transfer from the clad and is the primary coupling mechanism with the temperature of the clad/pellets. We solve these equations tightly coupled to the thermal diffusion equations for the clad and pellet domains. For notational simplicity, in the remainder of this document we will allow T to refer to both the solution of thermal diffusion in the pellets/clad and the enthalpy/pressure solution in the subchannel. Furthermore, we will allow the operator $L(T)$ to refer to the equations for thermal diffusion (4) as well as the equations for the subchannel solution (8)-(9).

Our goal is therefore to find distributions ψ, T and a value k such that Eqs. (1)–(3) are simultaneously satisfied. For ease of notation in the remainder of this document, we introduce an operator notation for discretized forms of the preceding equations:

$$\mathbf{A}(T)\phi = \lambda \mathbf{B}(T)\phi \tag{10}$$

$$\mathbf{R}_S\mathbf{R}_E\mathbf{B}(T)\phi = P^* \tag{11}$$

$$\mathbf{L}(T)T = \mathbf{R}_E\mathbf{B}(T)\phi\,, \tag{12}$$

where $\mathbf{R}_S$ and $\mathbf{R}_E$ are restriction operators in space and energy, respectively. Here Eq. (10) has taken advantage of the fact that only the angle-integrated variable ϕ must generally be stored rather than the corresponding angle-dependent ψ. In addition we have written the eigenvalue as $\lambda \equiv \frac{1}{k}$.

Because the focus of this exploratory investigation is on the assessment of possible coupling algorithms rather than on delivering a production capability, physics components and codes have been selected based on flexibility and ease of implementation rather than consistence with CASL preferred capabilities. For the radiation transport component, the SP_N discretization within the Denovo code has been selected. The simple nature of the SP_N equations offers significant advantages in the current study, the operators $\mathbf{A}$ and $\mathbf{B}$ from Eqs. (10)–(12) can be explicitly constructed as sparse matrices and therefore algebraic preconditioners (algebraic multigrid, for instance) can be easily applied. Through the Insilico reactor geometry/materials interface, cross sections generated by the XSProc module of the SCALE package are available. This combined transport capability is the same as that used in the completion of CASL AMA progression benchmark problem 7 [12]. For the heat transfer and subchannel flow portions of the problem, the AMP software package has been employed. AMP offers a very flexible multiphysics infrastructure that allows interoperability between different linear algebra components. In the current study several Trilinos solvers are used, with AMP providing an interface to nonlinear solvers in NOX, linear solvers in Belos, and algebraic multigrid preconditioning in ML.

3 Solution Approaches

The radiation transport and heat transfer systems described in Eqs. (10)–(12) can be written as a single system of coupled equations as

$$f\begin{pmatrix}\phi\\ \lambda\\ T\end{pmatrix} = \begin{bmatrix} f_\phi(\phi,\lambda,T)\\ f_\lambda(\phi,T)\\ f_T(\phi,T)\end{bmatrix} = \begin{bmatrix}\mathbf{A}(T)\phi - \lambda\mathbf{B}(T)\phi\\ \mathbf{R}_S\mathbf{R}_E\mathbf{B}(T)\phi - P^*\\ \mathbf{L}(T)T - \mathbf{R}_E\mathbf{B}(T)\phi\end{bmatrix} = \mathbf{0}\,. \tag{13}$$

One straightforward approach to solving this system of equations is to alternate between solves of the individual physics in a block Gauss-Seidel approach. This Picard iteration can be written as

$$\mathbf{A}(T^m)\phi^{m+1/2} = \lambda^m\mathbf{B}(T^m)\phi^{m+1/2} \tag{14}$$

$$\phi^{m+1} = \frac{P^*}{\mathbf{R}_S\mathbf{R}_E\mathbf{B}(T^m)\phi^{m+1/2}}\phi^{m+1/2} \tag{15}$$

$$\mathbf{L}(T^{m+1})T^{m+1} = \mathbf{R}_E\mathbf{B}(T^m)\phi^{m+1}\,. \tag{16}$$

It has been observed in several studies with various physics approximations and/or discretizations that this simple iteration scheme applied to light water reactor problems (such as those of interest to CASL) is prone to poor convergence and possibly divergence due to oscillations induced by certain error modes. The standard remedy for this issue is to introduce a damping parameter, ω, such that (16) is replaced by the two step approach

$$\mathbf{L}(T^{m+1/2})T^{m+1/2} = \mathbf{R}_E\mathbf{B}(T^m)\phi^{m+1} \tag{17}$$

$$T^{m+1} = \omega T^{m+1/2} + (1-\omega)T^m\,. \tag{18}$$

Optimal values for the damping parameter are typically between 0.3 and 0.6 [8, 10, 14]. Note that it is also possible to perform damping on the scalar flux (or power) instead of temperature.

3.1 Anderson Acceleration

The Picard iteration described above is an example of a fixed-point problem,

$$x = \mathbf{G}(x). \tag{19}$$

Fixed-point problems are amenable to acceleration techniques. Recently, an accelerator called Anderson acceleration (AA) [1] has garnered much attention and has been shown to be effective on certain classes of problems [4, 9, 3]. AA builds on Picard iteration by storing solutions from previous iterations to accelerate convergence. Fang and Saad [4] have shown that AA is related to quasi-Newton methods (known as a type II Broyden or "bad" Broyden) while Walker and Ni [13] have shown that for linear systems of equations, AA is "essentially equivalent" to GMRES. The stored solution vectors are used to form an unconstrained minimization problem to evaluate the next solution. Lott

et. al. [9] have shown, for instance, that AA not only improved the speed of convergence, but also significantly improved the robustness.

Although it is possible to apply Anderson acceleration directly to the monolithic set of Eq. (13), preliminary studies indicate that this approach exhibits very poor convergence behavior for the problem of interest and is unlikely to be competitive without significant work to develop appropriate preconditioning techniques. Note that an approach similar to this was used for the stand-alone neutronics k-eigenvalue problem in Ref. [3] with some success, though the absence of multiphysics feedback and the differing solution algorithms used for comparison make it difficult to draw substantive comparisons to our current problem. A more appealing approach is to use Anderson acceleration to accelerate and/or stabilize Picard iteration. From Eqs. (14)-(16) we can define two mapping operations. The first mapping, $\mathbf{g}(T)$, computes a power distribution from a known temperature distribution, i.e.

$$\mathbf{g}(T) = \frac{\mathbf{R}_E\mathbf{B}(T)\phi}{\mathbf{R}_S\mathbf{R}_E\mathbf{B}(T)\phi}, \tag{20}$$

where ϕ is the solution to the eigenvalue problem

$$\mathbf{A}(T)\phi = \lambda\mathbf{B}(T)\phi. \tag{21}$$

The second mapping operation, $\mathbf{h}(P)$ is given by solving the nonlinear equation

$$\mathbf{L}(T)T = P. \tag{22}$$

With this notation, the Picard iteration scheme described by Eqs. (14)-(16) can be written as the fixed-point mapping

$$T^{j+1} = \mathbf{h}\left(\mathbf{g}(T^j)\right), \tag{23}$$

which can be solved using Anderson acceleration. Alternative mappings were also explored. A reverse the mapping that accelerates the power unknowns was implemented

$$P^{j+1} = \mathbf{g}\left(\mathbf{h}(P^j)\right), \tag{24}$$

and all unknowns, x, were accelerated using the mapping

$$x^{j+1} = \begin{pmatrix} T^{j+1} \\ P^{j+1} \end{pmatrix} = \begin{pmatrix} \mathbf{g}(T^j) \\ \mathbf{h}(P^j) \end{pmatrix}. \tag{25}$$

3.2 Jacobian-Free Newton-Krylov

Ideally, we would like to use Newton's method to solve (13), however computation of the Jacobian is problematic because the temperature dependence of nuclear data is not available in closed form, but it is rather the result of complex cross section processing routines which extract pre-tabulated data from files and perform numerous small-scale transport calculations to produce collapsed cross

sections suitable for use in a radiation transport code. The availability of a nonlinear function but difficulty in formulating the corresponding Jacobian matrix suggests that the use of Jacobian-free Newton Krylov (JFNK) methods might be appropriate [7]. JFNK methods are based on two primary ideas. First, if the Newton correction equation given by

$$\mathbf{J}(u^m)\delta^m = -f(u^m) \tag{26}$$

is solved using a Krylov subspace method then access to the full Jacobian matrix is not necessary, only the action of the Jacobian applied to a vector is required. The second idea is that the product of the Jacobian and a given vector can be approximated using a finite difference approach, e.g.

$$\mathbf{J}(u^m)v \approx \frac{f(u^m + \epsilon v) - f(u^m)}{\epsilon}. \tag{27}$$

Note that second (or higher) order approximations are possible, but the form in (27) is far more common because it requires only a single function evaluation per Jacobian-vector product (the value of $f(u^m)$ can be computed and stored once per Newton iteration so that each subsequent Jacobian-vector product requires only the evaluation of $f(u^m + \epsilon v)$). The JFNK method therefore allows an approximate Newton method to be performed using only evaluations of the nonlinear function.

3.2.1 JFNK Preconditioning

Even though the use of JFNK eliminates the need to explicitly form or store the Jacobian matrix, some knowledge about the Jacobian is still beneficial in order to construct a preconditioner so that (26) can be solved effectively. The true Jacobian corresponding to the function defined by (13) can be written as

$$\mathbf{J}\begin{pmatrix}\phi\\ \lambda\\ T\end{pmatrix} = \begin{bmatrix} \mathbf{A}(T) - \lambda\mathbf{B}(T) & -\mathbf{B}(T)\phi & \dfrac{\partial(\mathbf{A}(T)\phi - \lambda\mathbf{B}(T)\phi)}{\partial T} \\ \mathbf{R}_S\mathbf{R}_E\mathbf{B}(T) & 0 & \dfrac{\partial(\mathbf{R}_S\mathbf{R}_E\mathbf{B}(T)\phi)}{\partial T} \\ -\mathbf{R}_E\mathbf{B}(T) & 0 & \dfrac{\partial(\mathbf{L}(T)T - \mathbf{R}_E\mathbf{B}(T)\phi)}{\partial T} \end{bmatrix}, \tag{28}$$

where the entries in the last column of this block matrix are left as simply partial derivatives with respect to temperature to indicate that these terms are generally not available in closed form due to the dependence on material properties. A simple approach to preconditioning a linear system involving (28) is a block diagonal approach in which preconditioners for each physics are applied independently, i.e.

$$\mathbf{P}_1\begin{pmatrix}\phi\\ \lambda\\ T\end{pmatrix} = \begin{bmatrix} \widehat{\mathbf{A}}(T) - \lambda\widehat{\mathbf{B}}(T) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \widehat{\mathbf{L}}(T) \end{bmatrix}, \tag{29}$$

where $\widehat{(\cdot)}$ indicates that some approximation of the operator is used in the preconditioner. Possible choices for this approximation include incomplete factorizations, algebraic multigrid methods, or physics-based approximations. This preconditioner selection has the advantages of being relatively simple to construct and inexpensive to apply (assuming the approximations to the individual physics operators are inexpensive). However, neglecting terms in the Jacobian that represent coupling between different physics components in the preconditioner is likely to result in a large number of iterations being required for convergence. Development of preconditioning strategies that account for coupling behavior between physics components has the potential to significantly improve the convergence behavior of the JFNK approach and the development of such techniques is an attractive area of research for future investigations.

3.2.2 Approximate Jacobian-Vector Products

Improving the quality of the preconditioner used to solve linear systems involving the Jacobian matrix is one route to improving the efficiency of the overall solution strategy, but as noted previously even a very strong preconditioner may not be sufficient to reduce the number of cross section processing steps to a very small number. Another approach to achieve this goal is to use an approximation to the Jacobian when computing the Newton direction in the spirit of quasi-Newton methods, i.e.

$$\widehat{\mathbf{J}}(u^m)\delta^m = -f(u^m)\,. \tag{30}$$

In the interest of maintaining the attractive matrix-free nature of JFNK, we can determine a corresponding approximate function evaluation, $\widehat{f}$, such that a finite difference operation approximates a product with $\widehat{J}$ rather than the full Jacobian, i.e.

$$\widehat{\mathbf{J}}(u^m)v \approx \frac{\widehat{f}(u^m + \epsilon v) - f(u^m)}{\epsilon}. \tag{31}$$

It should be noted that the base function evaluation, $f(u^m)$, is used to evaluate convergence of the nonlinear iterations and must be the full nonlinear operator, including computation of all cross sections at the specified temperature.

One possibility for approximating the Jacobian is to simply neglect the temperature variation of cross sections during an approximate Jacobian-vector product, corresponding to an approximate Jacobian of

$$\widehat{\mathbf{J}}\begin{pmatrix}\phi\\ \lambda\\ T\end{pmatrix} = \begin{bmatrix}\mathbf{A}(T) - \lambda\mathbf{B}(T) & -\mathbf{B}(T)\phi & 0\\ \mathbf{R}_S\mathbf{R}_E\mathbf{B}(T) & 0 & 0\\ -\mathbf{R}_E\mathbf{B}(T) & 0 & \dfrac{\partial(\mathbf{L}(T)T)}{\partial T}\end{bmatrix}, \tag{32}$$

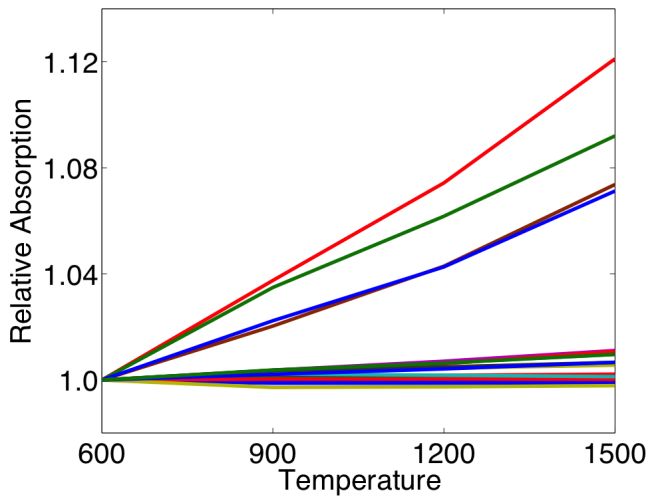


Figure 1. Temperature dependence of 23 group absorption cross sections. Each curve represents the absorption cross section in a different energy group.

or equivalently an approximate function evaluation of

$$\widehat{f}\begin{pmatrix}\phi+\Delta\phi\\ \lambda+\Delta\lambda\\ T+\Delta T\end{pmatrix}=\begin{bmatrix}\mathbf{A}(T)(\phi+\Delta\phi)-(\lambda+\Delta\lambda)\mathbf{B}(T)(\phi+\Delta\phi)\\ \mathbf{R}_S\mathbf{R}_E\mathbf{B}(T)(\phi+\Delta\phi)-P^*\\ \mathbf{L}(T+\Delta T)(T+\Delta T)-\mathbf{R}_E\mathbf{B}(T)(\phi+\Delta\phi)\end{bmatrix}. \tag{33}$$

This selection, however, results in no updated temperature information being communicated from the heat transfer solver to the neutronics domain during a given nonlinear iteration. This lack of information is expected to have a detrimental effect on the convergence behavior of the nonlinear solver.

It may be possible to include some information about the temperature feedback effect on cross sections without performing a full cross section processing step at each function evaluation. This can be accomplished by noting that the strongest temperature feedback effect is due to an increase in absorption with increasing temperature (largely due to Doppler broading in ^{238}U). As shown in Fig. 1, the temperature dependence of absorption cross sections is approximately linear over a wide range of temperatures. This suggests a modification to the Jacobian approximation of Eq. (32) which uses a linear approximation to the temperature dependence of the absorption cross section and neglects the

temperature dependence of all other cross sections:

$$\widehat{\mathbf{J}}\begin{pmatrix}\phi\\ \lambda\\ T\end{pmatrix} = \begin{bmatrix}\mathbf{A}(T)-\lambda\mathbf{B}(T) & -\mathbf{B}(T)\phi & \dfrac{\partial\sigma_a}{\partial T}\phi\\ \mathbf{R}_S\mathbf{R}_E\mathbf{B}(T) & 0 & 0\\ -\mathbf{R}_E\mathbf{B}(T) & 0 & \dfrac{\partial(\mathbf{L}(T)T)}{\partial T}\end{bmatrix}. \tag{34}$$

The function evaluation corresponding to Eq. (34) is

$$\widehat{f}\begin{pmatrix}\phi+\Delta\phi\\ \lambda+\Delta\lambda\\ T+\Delta T\end{pmatrix} = \begin{bmatrix}\mathbf{A}(T)(\phi+\Delta\phi)-(\lambda+\Delta\lambda)\mathbf{B}(T)(\phi+\Delta\phi)+\dfrac{\partial\sigma_a}{\partial T}\phi\Delta T\\ \mathbf{R}_S\mathbf{R}_E\mathbf{B}(T)(\phi+\Delta\phi)-P^*\\ \mathbf{L}(T+\Delta T)(T+\Delta T)-\mathbf{R}_E\mathbf{B}(T)(\phi+\Delta\phi)\end{bmatrix}. \tag{35}$$

4 Results

To test the behavior of different nonlinear solvers, we consider the solution of AMA Problem 6 [11]. This problem consists of a single 17×17 PWR fuel assembly with 264 fuel pins containing 3.1% enriched UO_2, 24 guide tubes and a single central instrumentation tube. Eight spacer grids are located along the axial length of the assembly, as well as upper and lower assembly nozzles. A full description of the problem, including detailed material and geometric details, is contained in Ref. [11].

For the base configuration, we model the assembly at 17.67 MW and 1300 ppm dissolved boron; the effect of power level and boron concentration on solver convergence will be studied later in this section. The base configuration uses a 56 energy group cross section library and uses the XSProc module of the SCALE package to collapse these cross sections to 23 groups for Denovo SP_N calculations. Distinct cross sections are used for each fuel pin and for each of 49 axial levels. An SP_3 angular order (containing 2 angular moments) is used for all calculations, along with P_1 scattering. A 2×2 spatial mesh per pin cell is used in the x–y plane, with a maximum axial mesh size of 2 cm, resulting in 290,156 mesh cells for the full assembly. The AMP heat transfer problem contains 15,504 mesh cells per fuel pin (over both the fuel and clad meshes), resulting in approximately 4.1 million total cells. A linear continuous finite element discretization of the heat transfer problem is used. All problems in this study are executed in parallel on 289 processing cores on the OLCF EOS cluster, resulting in 1 fuel pin per core.

Five different solver approaches are considered. First is a damped Picard iteration with the damping applied to the temperature component of the solution as described in Section 3. Except where otherwise noted, all calculations use a damping factor of $\omega = 0.45$ which appears to produce nearly optimal

convergence behavior for a wide range of problems. Within Picard iteration the k-eigenvalue problem is solved with a generalized Davidson eigensolver and the heat transfer and subchannel flow problem is solved using a JFNK approach (due to the temperature dependence of the thermal conductivity, the heat transfer problem by itself is nonlinear). The second solver is Anderson acceleration of Picard iteration, as described in Section 3.1. The next solver consists of a full JFNK solver using the nonlinear function evaluation described by Eq. (13). Recall that this selection requires performing full cross section processing at every *linear* iteration. The final two solvers are modified JFNK approaches based on the function evaluations in Section 3.2.2. The approach denoted by MJFNK1 corresponds to the function evaluation of Eq. (33) in which the temperature dependence of the cross sections is entirely neglected during a nonlinear iteration. The approach denoted by MJFNK2 corresponds to Eq. (35) in which a linear approximation to the temperature dependence of the absorption cross section is used during each nonlinear iteration. This linear approximation is determined by performing stand-alone XSProc calculations at 600K and 1500K to compute pin-homogenized absorption cross sections. In both modified JFNK approaches, only a single cross section processing step is performed per nonlinear iteration.

A brief mention should be made of the memory usage of the various solution approaches. Picard iteration itself uses virtually no memory on its own (only a single extra copy of the solution vector to determine convergence), and therefore the memory usage is determined by the individual solvers. In this study, the JFNK solver used by the heat transfer solver internally uses a GMRES linear solver which required a subspace containing up to 20 vectors containing the thermal solution. The generalized Davidson solver used in the SP_N calculations requires multiple subspaces, which combined required the storage of up to 75 vectors of the length of the SP_N solution vector. Memory requirements for Anderson acceleration are equivalent to Picard except that a small number of addition vectors (in this study, 2) are used by the solver. For the JFNK-based methods, the primary memory requirement is through the GMRES linear solver used. In this study, no restarting was employed resulting in a subspace size of as many as 80 vectors, each having the combined length of the thermal and SP_N solutions. Note that all of the GMRES linear solvers as well as the generalized Davidson offer the possibility of using more aggressive restarting capabilities to limit the size of the subspaces that are used, typically at a cost of performing a small number of additional iterations to reach the same convergence criteria. In addition to memory associated with solver subspaces, there is an additional memory cost associated with forming the problem operators and corresponding preconditioners. These costs, however, are consistent across every solver option because consistent parameters and preconditioning options were used.

Figure 2 shows the convergence behavior for Picard iteration and each of the JFNK variants. Picard iteration demonstrates the expected linear convergence rate and convergence for the full JFNK is quadratic. The first modified JFNK method follows the convergence behavior of the full JFNK for the first few iterations before separating and ultimately converging linearly with a rate similar to that of Picard iteration. The second modified JFNK follows the con-

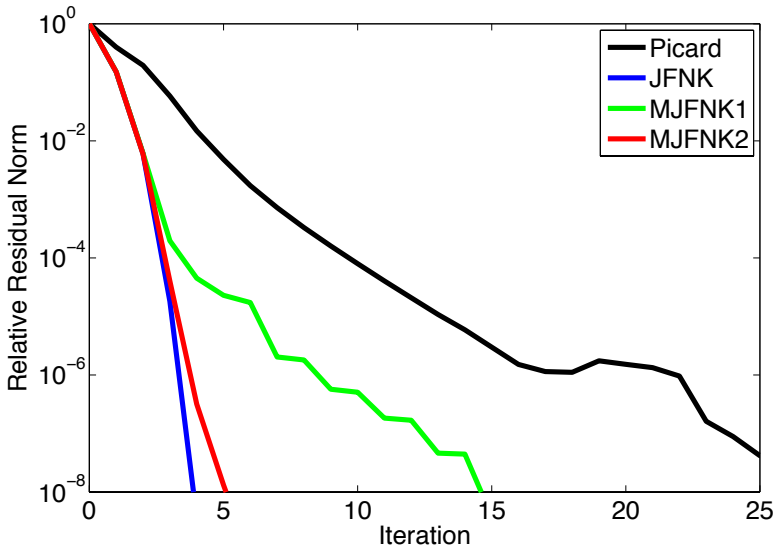


Figure 2. Nonlinear convergence behavior for Picard and JFNK methods.

vergence rate of full JFNK even more closely and although it eventually deviates from quadratic convergence it does result in linear convergence at a very rapid rate. It should be noted that the convergence criteria used to generate this plot were several orders of magnitude tighter than what is typically applied to such multiphysics problems. This was done to emphasize the pertinent convergence features of each method.

The convergence behavior of Picard iteration as a function of damping parameter at several different power levels is shown in Figure 3. These curves have the same general shape as the corresponding plot from Ref. [5] and consistent with the behavior observed in Refs. [8, 10, 14], all of which reported using damping factors between 0.3 and 0.6. An interesting feature of this curve is that the convergence behavior depends on the power level, with high power levels requiring smaller damping factors for optimal convergence.

Table 1 provides the time required to achieve a relative convergence tolerance of 10^{-4} for each solution strategy at four different power levels. The time required for Picard iteration to converge is approximately constant for most power levels, showing a slight upward trend at higher power levels indicative of the stronger coupling present at high power. Anderson acceleration converges in nearly the same time as Picard but fails to converge at higher power levels. The Anderson failures were predominantly the result of violating bounds on material property evaluations. Although this failure to converge appears to be a downside for Anderson, its ability to reproduce the timing of an optimally damped Picard iteration without any such tuning parameter of its own actually indicates that there is significant potential for Anderson acceleration or

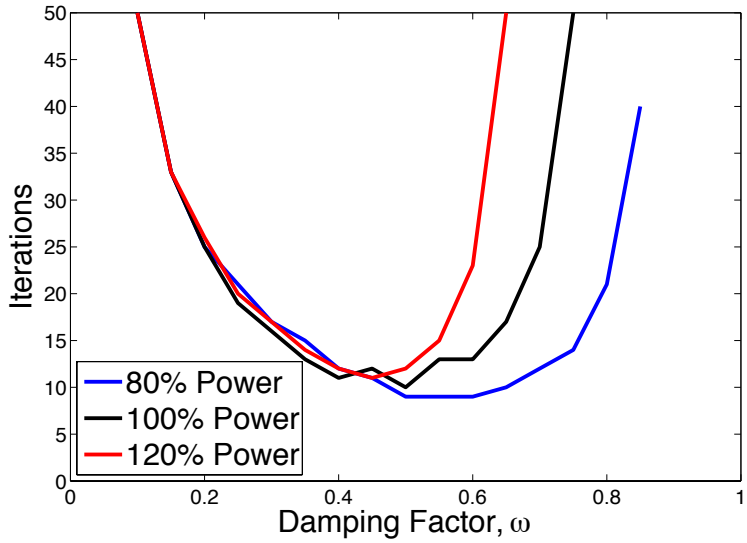


Figure 3. Picard convergence vs. damping parameter for different power levels.

similar methods. Due to lack of convergence at normal power levels, however, Anderson acceleration will not be considered for the remaining studies in this paper. We note that all three of the mappings described in Section 3.1 were implemented, but had a negligible impact on performance. Despite the fact that JFNK exhibits quadratic convergence behavior, the large cost per iteration due to performing cross section processing at every linear iteration results in large run times that are not competitive with other approaches. The first modified JFNK method performs very well at low power levels, significantly reducing runtimes relative to Picard iteration. At high power levels, however, the time required for convergence greatly increases, reaching over twice the runtime of Picard at 120% power. This behavior is easily understood by noting that the low heat generation rate at low power results in relatively small changes in temperature and therefore the effect of neglecting the temperature dependence of cross sections is not too large, but high power levels produce large temperature variations and a corresponding degradation due to not capturing the effects of changes in cross sections. The second modified JFNK approach performs very well across all power levels, resulting in the fastest time to solution for all cases and displaying very little variation with power level from 80–100%.

Table 2 shows the time to reach convergence for each solver approach at several different boron concentrations. The convergence behavior does not appear to be very strongly dependent on the boron concentration, the first three solvers produced the longest runtime at 600 ppm boron while MFJNK2 had the longest runtime at 0 ppm. For MJFNK1, there is over a factor of two difference in the time to solution at 600 ppm versus 2000 ppm; the other solvers do not

Method	Power Level			
	60%	80%	100%	120%
Picard	1296	1150	1353	1523
Anderson	1305	-	-	-
JFNK	6168	5958	8908	8844
MJFNK1	734	1035	1693	3164
MJFNK2	695	985	971	961

Table 1. Solver timing in seconds as function of power level.

Method	Boron Concentration			
	0 ppm	600 ppm	1300 ppm	2000 ppm
Picard	1370	1604	1353	1445
JFNK	8079	9200	8908	8391
MJFNK1	2743	3858	1693	1704
MJFNK2	1325	1008	971	1002

Table 2. Solver timing in seconds as function of boron concentration.

experience such strong dependence. It should be noted that for MJFNK2 a single linear approximation to the absorption cross section is used for all boron concentrations, indicating a high level of robustness of the approach to this parameter.

Table 3 shows the overall timing results for three different combinations of cross section processing and transport energy group structures, effectively modifying the cost of cross section processing from relatively cheap to very expensive. When eight group cross section processing is performed, the computational time for most methods is dominated by the time spent in the thermal (and subchannel) portions of the calculation. In this case the full JFNK approach is still slower than Picard iteration, though it is easy to envision that improvements in preconditioning could lead to such an approach offering an advantage relative to Picard. However, this coarse energy group structure does not produce results with appropriate accuracy and therefore finer energy group stuctures are needed. As the number of energy groups is increased, the time spent in cross section processing quickly dominates the overall runtime. This is especially dramatic in the case of 252 group cross section processing in which the time required for the full JFNK approach is more than an order of magnitude greater than for Picard iteration. Thus, for the large numbers of energy groups typically required for accurate reactor physics calculations it is apparent that naïve application of JFNK will result in far worse behavior than Picard iteration. By appealing to modified JFNK approaches that avoid frequent cross section processing, however, significant reduction in computational time relative to Picard is possible, with around a factor of two reduction evident for MJFNK2 in the

Method	XSProc Groups/SP_N Groups		
	8/8	56/23	252/23
Picard	980	1353	3868
JFNK	1685	8908	51727
MJFNK1	1070	1693	3793
MJFNK2	636	971	2013

Table 3. Solver timing in seconds for different energy group structures.

Method	Total	SP_N	Thermal	XSProc
Picard	980 (11)	32 (161)	851 (302)	65 (11)
JFNK	1685 (5)	34 (203)	444 (203)	1075 (203)
MJFNK1	1070 (8)	61 (371)	763 (371)	40 (8)
MJFNK2	636 (5)	34 (208)	427 (208)	30 (5)

Table 4. Timing in seconds (operator applies) by component for 8 group XSProc, 8 group SP_N.

252 group XSProc case.

In order to more thoroughly assess the merits of the different solvers, we now examine the amount of time spent in each physics component as well as the number of times each physics component was applied. Tables 4, 5, and 6 provide this detailed information for the same three different energy group structures; Note that solver setup, mesh transfer, and related operations are not included in any component and therefore the sum of the time spent in the various physics components will sum to less than the total time.

Comparing the individual columns of these tables, we can discern exactly where the computational gains of the modified JFNK approach are realized. In all cases, the amount of time spent in the thermal operator and cross section processing are greatly reduced relative to Picard but the time spent in the SP_N operator is reduced only very little or possibly increased. The reason for this is that Picard iteration is able to utilize a generalized Davidson eigensolver that has been demonstrated to be highly efficient for stand-alone k-eigenvalue calculations. Multiphysics coupling involving radiation transport solvers that do not have optimized eigensolvers or using discretizations that do not easily lend themselves to a generalized Davidson approach would be expected to experience more significant timing reduction in the transport portion of the calculation.

As a final note on the timing behavior of Picard iteration, we compare the Picard solver used in this study (AMP) to the VERA core simulator multiphysics coupling involving Denovo SP_N, XSProc, and CobraTF (VERA-CS) for the same reference configuration. The timing breakdown for the AMP and CobraTF solutions of this problem are shown in Table 7. The discrepancy in the SP_N runtimes is due to the fact that, while both couplings used the same gen-

Method	Total	SP_N	Thermal	XSProc
Picard	1353 (11)	178 (273)	832 (298)	312 (11)
JFNK	8908 (5)	160 (258)	649 (258)	7912 (258)
MJFNK1	1693 (8)	259 (434)	898 (434)	257 (7)
MJFNK2	971 (5)	137 (228)	470 (228)	171 (5)

Table 5. Timing in seconds (operator applies) by component for 56 group XSProc, 23 group SP_N.

Method	Total	SP_N	Thermal	XSProc
Picard	3868 (13)	157 (244)	939 (332)	2739 (13)
JFNK	51727 (5)	139 (228)	591 (228)	50817 (228)
MJFNK1	3793 (10)	313 (523)	1103 (523)	2049 (10)
MJFNK2	2013 (5)	133 (223)	466 (223)	1221 (5)

Table 6. Timing in seconds (operator applies) by component for 252 group XSProc, 23 group SP_N.

eralized Davidson eigensolver, different preconditioning options were selected which results in slightly different convergence behavior. The reason that the thermal component of the calculation is so similar even though AMP is performing a 3D within-pin calculation and CobraTF is using a low order model is that AMP is able to perform its calculation in parallel on all 289 computational cores available for this problem whereas for this single assembly problem CobraTF is only able to use a single core. The similarity in convergence behavior for Picard iteration between these different code couplings seems to suggest that improvements resulting from JFNK-based solver in the AMP code would likely carry over to the CobraTF-based coupling if such solvers were implemented in that framework.

From this study, we find that MJFNK2 can have a significant impact on runtime efficiency. A logical question is whether these assumptions can be easily applied to the Picard coupling. It is entirely possible to do this, but can be *very dangerous*. Newton-based methods can check consistent residuals of the monolithic system (Eq. (13)) to determine convergence of the global coupled system. In all of the Newton-based methods above, the exact cross sections were

Method	Total	SP_N	Thermal	XSProc
AMP	1353 (11)	178 (273)	832 (298)	312 (11)
VERA-CS	1352 (10)	257 (267)	768 (1663)	294 (10)

Table 7. Timing in seconds (operator applies) by component for AMP and CobraTF coupling.

always computed (no approximations) for the residual norms that determine convergence. In the case of typical black-box Picard coupling, there is no access to a residual (consistent or lagged) and the only way to determine convergence is to look for stagnation of response quantities (e.g. leading eignevalue) or data transfer values (e.g. temperature and power) between iterations. If an approximation is used in the black-box iterations, stagnation won't necessarily be due to convergence but could also be due to these missing terms. The result will be a code that claims convergence but is very far from a consistent solution. Users unfamiliar with this approximation could easily try to run VERA on problems where the approximation is not valid and get false convergence, leading to very inaccurate results. This could be very embarrassing to CASL. The only way to use the approximation effectively would be to apply the approximation far from the solution and then once the Picard iteration thinks it is converged, stop using the approximation and perform at least two more Picard iterations with full cross section processing to see the true change. Most likely the code would take more iterations, but a good percentage might be done with the cheaper approximations. This procedure could also have significant impact on robusness of the Picard method as well. We have not tried this to date, but it would be a good follow on experiment in the next PoR.

5 Conclusions

In this study we have provided an assessment of several different nonlinear solvers for use in problems involving coupled neutronics and thermal hydraulics. In particular, comparisons of damped Picard iteration, Anderson acceleration, and Jacobian-free Newton-Krylov have been performed. Because a naïve implementation of JFNK results in performing a very large number of cross section processing steps, two modified variants of JFNK have been introduced which only require processing cross sections at each nonlinear iteration. Numerical results on CASL AMA Problem 6 indicate that:

- if on-the-fly generation of cross sections represents a large portion of the runtime of a calculation then a direct JFNK implementation results in a prohibitively large number of cross section processing steps,
- approximations to the nonlinear operator that avoid cross section processing can largely preserve the fast convergence rate of JFNK without the overhead,
- computational savings of 1.4-2 relative to Picard iteration with Newton-like methods are possible, and
- for the core simulator problems in CASL the Newton-like methods offer robust convergence behavior that does not depend on the selection and optimization of a damping parameter.

6 Future Work

Several areas for future work are suggested by this study. First, the preconditioners used in this study are exclusively block diagonal, utilizing preconditioners targeted at different physics components separately. The development of preconditioners that incorporated additional coupling terms could offer significant reduction in the number of linear iterations required at each nonlinear iteration and therefore reduce the overall computational time.

Another area of interest is in improvements to the Anderson acceleration approach. It has been observed that at low power levels Anderson acceleration converges about as well as Picard iteration when Picard uses an optimum damping parameter. At higher power levels, Anderson acceleration failed to converge primarily due to violating material property bounds. Further work on relating the mixing parameter in Anderson acceleration to the Picard damping parameter may produce improved convergence at these higher powers. Additionally, adding a Newton-based globalization (line search) to the fixed-point mapping in the Anderson solver could achieve a robust solution, although the current theory doesn't necessarily guarantee the mapping is a descent direction.

The application of the JFNK2 approximation to the Picard solver should also be explored. As discussed in the previous section, significant attention should be paid to stopping criteria.

Additional areas for extension of this research lie in development of critical boron search capabilities as well as application of methods to alternate reactor designs. Whereas performing a boron search with Picard iteration typically requires iterating to an eigenvalue of 1, JFNK-based approaches allow the problem to be recast such that the boron concentration replaces the eigenvalue in the solution vector making it possible to converge directly to the critical boron concentration without ever computing an eigenvalue. The only complication to this approach is that modified JFNK approaches (i.e. MFJNK2 in this study) would likely need to approximate derivatives of cross sections with respect to boron concentration in addition to the current fuel temperature derivatives. Similarly, application of these methods to reactor designs other than PWR's may require addressing additional derivative terms. For instance, with the large moderator density variations seen in a BWR it is likely that the MFJNK2 approach would need to account for cross section derivatives with respect to moderator density.

References

[1] D. G. Anderson. Iterative procedures for nonlinear integral equations. *J. Assoc. Comput. Machinery*, 12:547–560, 1965.

[2] R. A. Bartlett, M. A. Heroux, and J. M. Willenbring. Tribits lifecycle model, version 1.0: A lean/agile software lifecycle model for research-based computational science and engineering and applied mathematical software. Sandia Technical Report SAND2012-0561, Sandia National Laboratories, February 2012.

[3] M. T. Calef, E. D. Fichtl, J. S. Warsa, M. Berndt, and N. N. Carlson. Nonlinear krylov acceleration applied to a discrete ordinates formulation of the k-eigenvalue problem. *Journal of Computational Physics*, 238:188–209, 2013.

[4] H. Fang and Y. Saad. Two classes of multisecant methods for nonlinear acceleration. *Numerical Linear Algebra with Applications*, 16:197–221, 2009.

[5] S. Hamilton, K. Clarno, M. Berrill, T. Evans, G. Davidson, R. Lefebvre, and R. Sampath. Multiphysics simulations for LWR analysis. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, Sun Valley, ID, 2013.

[6] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*, volume 16 of *Frontiers in Applied Mathematics*. SIAM, Philadelphia, PA, 1995.

[7] D.A. Knoll and D.E. Keyes. Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193:357–397, 2004.

[8] X. Liu and X. Cheng. Thermal-hydraulic and neutron-physical characteristics of a new SCWR fuel assembly. *Annals of Nuclear Energy*, 36:28–36, 2009.

[9] P. A. Lott, H. F. Walker, C. S. Woodward, and U. M. Yang. An accelerated picard method for nonlinear systems related to variably saturated flow. *Advances in Water Resources*, 38:92–101, 2012.

[10] L. Monti and T. Schulenberg. Coupled ERANOS/TRACE system for HP-WLR 3 pass core analyses. In *International Conference on Mathematics, Computational Methods & Reactor Physics*, Saratoga Springs, NY, 2009.

[11] S. Palmtag. Coupled single assembly solution with VERA (Problem 6). Technical Report CASL-U-2013-0150-000, Consortium for Advanced Simulation of LWRs, 2013.

[12] S. Palmtag. Demonstration of neutronics coupled to thermal-hydraulics for a full-core problem using VERA. Technical Report CASL-U-2013-0196-000, Consortium for Advanced Simulation of LWRs, 2013.

[13] Homer F. Walker and Peng Ni. Anderson acceleration for fixed-point iterations. *SIAM J. Numer. Anal.*, 49(4):1715–1735, August 2011.

[14] J. Yan et al. Coupled computational fluid dynamics and MOC neutronic simulations of westinghouse PWR fuel assemblies with grid spacers. In *14th International Topical Meeting on Nuclear Reactor Thermal Hydraulics*, Toronto, ON, 2011.