A MODIFIED TIGHTLY COUPLED METHOD FOR REACTOR TRANSIENT SIMULATIONS

Jaron P. Senecal and Wei Ji*
Department of Mechanical, Aerospace, and Nuclear Engineering
Rensselaer Polytechnic Institute, Troy, NY
senecj@rpi.edu, jiw2@rpi.edu

ABSTRACT

Strongly coupled multi-disciplinary problems are usually handled quite efficiently with fully coupled computer codes. However, this monolithic approach to multiphysics may not be ideal if the characteristics of the constituent single-disciplinary problems differ greatly. And a more common impetus for using more loosely coupled methods is the relative ease of implementation. The loosely coupled operator splitting method is commonly used due to its simplicity, but it limits feedback effects to first order temporal accuracy. Picard Iteration is a variation on operator splitting that fully resolves multiphysics feedbacks, which puts it in the category of tightly coupled methods. The drawback of Picard Iteration is that each single-disciplinary problem is solved repeatedly at each timestep. The unnecessary work of solving each problem to high accuracy in the initial iterations is referred to as over-solving. This work presents a straightforward modification to the Picard Iteration algorithm which reduces solution time by preventing over-solving. The magnitude of computational savings is problem dependent and tends to increase for more strongly coupled problems. As an example, a transient homogeneous slab reactor problem with thermal feedback was created. The new method results in up to a 40% savings in solution time when compared with Picard Iteration. This method is not difficult to implement because it does not require substantive changes to the Picard Iteration framework. Furthermore, the differences between the two methods' solutions are entirely negligible.

Key Words: **Multiphysics coupling, Picard Iteration, Reactor transient**

1. INTRODUCTION

Nuclear power reactor transients are intrinsically multiphysics problems. Thermal feedback determines the course of the transient, and high-fidelity analysis requires sophisticated neutron and heat transport capabilities. Many multiphysics programs [1] link these two disciplines, but usually with relatively simple operator splitting methods. In this work, a new method [2] of coupling two single-disciplinary solvers is applied to a reactor transient problem in order to solve the multiphysics problem more efficiently. The new coupling method enables a tighter coupling between the individual single-disciplinary

*Corresponding author. Research group website: neams.rpi.edu

solvers. The coupling is realized in the midst of each solver's iterative solution process, which reduces over-solving. An application was built in the MOOSE framework [3] to solve an example problem and demonstrate the gain in computational performance compared to other methods.

2. COUPLING METHODS

There are two primary methods of coupling neutron kinetics and heat transfer. Fully coupled methods [4] provide good computational performance, but they usually require significant modifications to the existing codes. Operator splitting methods [4] are less efficient but flexible enough to incorporate mature codes without rewriting them.

2.1. Full Coupling

Fully coupled methods are efficient because all of the feedback effects are properly included. This allows the simulation to be solved in just one solution process, rather than iteratively with multiple coupled solvers. This monolithic approach allows the use of Newton-based methods, which are desirable for their quadratic convergence. The Jacobian-Free Newton Krylov (JFNK) method is one of the most popular fully coupled methods. The details of JFNK can be found elsewhere [5].

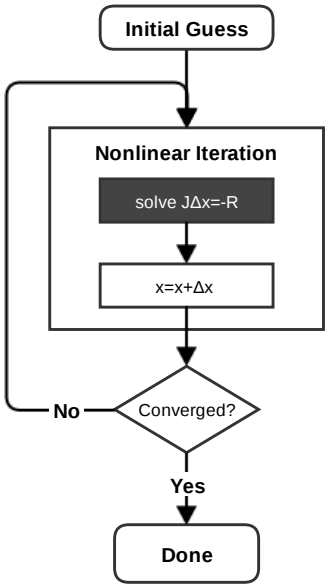

Figure 1. Depiction of the Jacobian-Free Newton Krylov method. The solution vector is x, the residual is R, and J is the Jacobian. Note that there is only one (large) system of equations to solve.

Fully coupled methods are not always feasible, however. Most commonly, two legacy codes are coupled and looser coupling methods allow this to be done with much less code rewriting. Another motivation for other coupling methods is that they allow specialized solution methods and meshes (both spatial and temporal) to be applied to each discipline.

2.2. Picard Iteration

The operator splitting method breaks the problem into pieces with each numerical solver dealing with a fraction of the problem. Picard Iteration is a tight coupling method where the operator splitting process is iterated for each timestep, see Figure 2. These global iterations resolve nonlinearities and allow for larger timesteps.

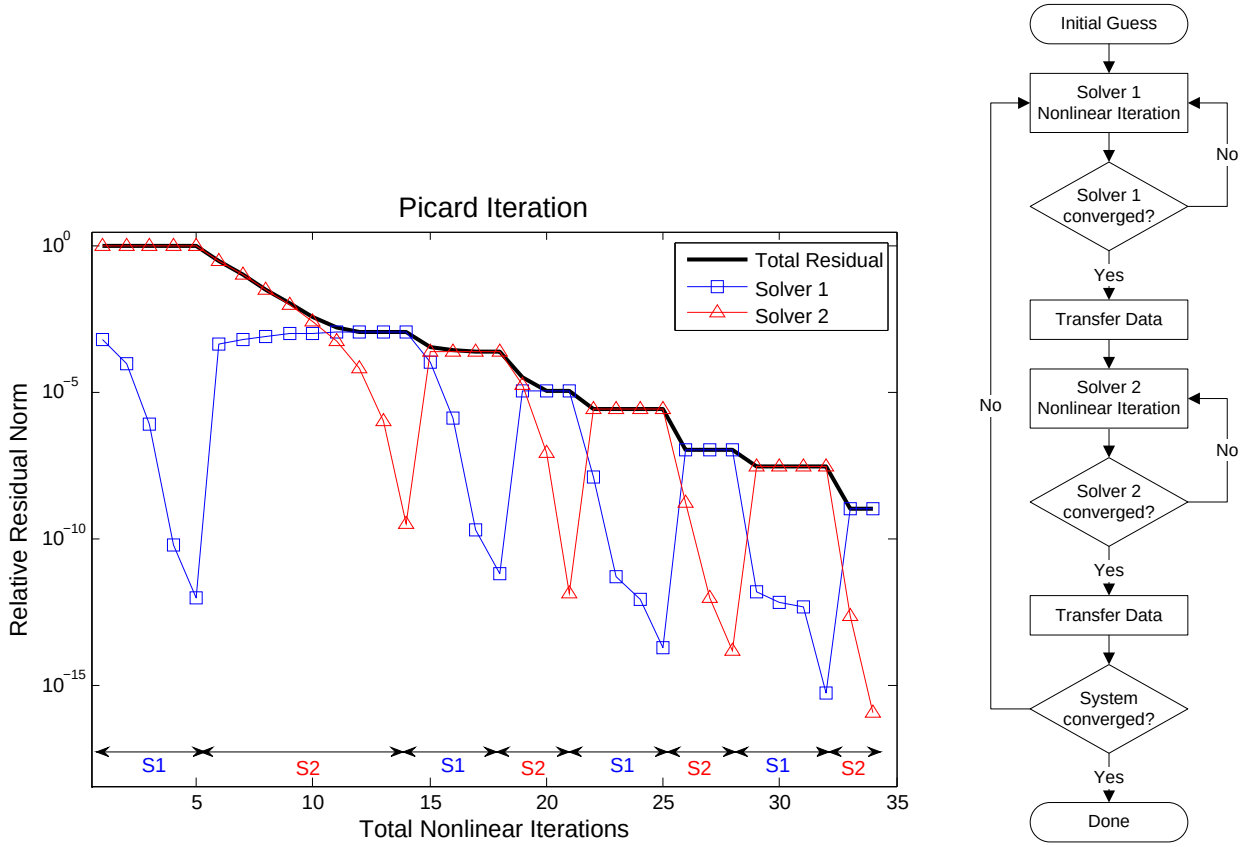

Figure 2. *Left:* Example of the development of the residuals in a two-solver problem coupled with Picard Iteration. The bottom edge denotes which solver is active. *Right:* Flow chart outlining the process of Picard Iteration.

Coupling via Picard Iteration or operator splitting accounts for the majority of multiphysics applications. This is because these schemes are easy to implement and allow developers to build a multiphysics

code from mature preexisting single-discipline codes [1]. The weakness of Picard Iteration is that each component application is solved to convergence multiple times at each timestep. Figure 2 shows that much computational effort is wasted in driving the residual unnecessarily close to zero during the initial iterations. If this over-solving can be avoided, significant time savings will result. The results from Figure 2 are taken from a conjugate heat transfer problem which is discussed in more detail in a previous publication [2].

2.3. Residual Balance

The Residual Balance method [2] is an adaptation to Picard Iteration that improves computational performance while retaining its flexibility. Rather than converging each single-discipline code at every iteration, the constituent problems are only partially solved, see Figure 3. This may result in more global iterations, but usually less total work is required because fewer solver iterations are needed. The Residual Balance method prevents over-solving by switching between codes shortly after the residual of one falls below the other's. Thus the residuals only approach zero once, instead of repeatedly.

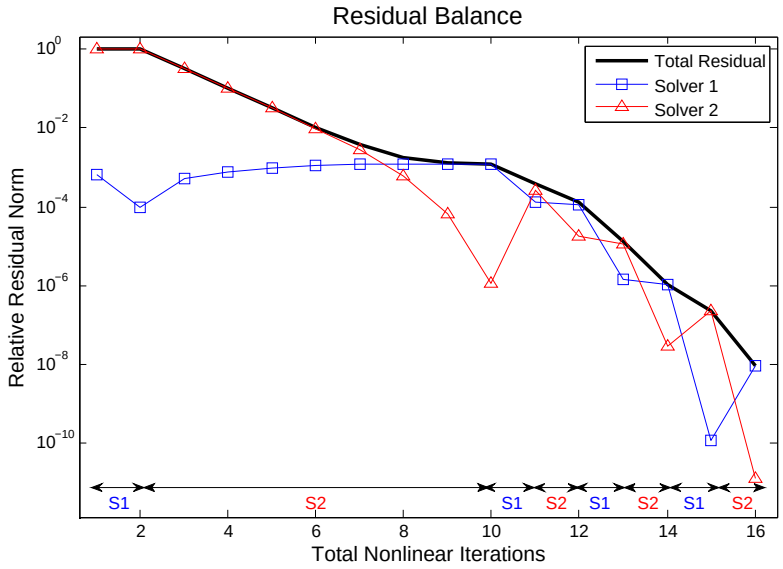

Figure 3. Example of the development of the residuals in a two-solver problem coupled with the Residual Balance method.

The Residual Balance method is easily implemented with two changes to the Picard Iteration logic. First, when a control passes to a solver it takes at least one step. Second, the absolute tolerance is adjusted in order to prevent over-solving.

$$abs\_tol = \max\left(A\left\|R^0_{S2}\right\|, min\_abs\_tol\right) . \quad (1)$$

Here A is a user-supplied parameter and $\|R^0_{S2}\|$ is the L_2 norm of the initial residual from the other

solver's most recent solve. In order to retain the original absolute tolerance functionality, the min_abs_tol parameter is included.

The parameter A allows the user to control the behavior of the Residual Balance method. Generally a value of $A = 0.1$ is satisfactory for the Residual Balance method and results in alternating between single iterations of the constituent codes. Or, for example if A approaches zero, the solution will be stopped by either min_abs_tol or the relative tolerance, and thus it will behave like Picard Iteration. Alternately, if A is smaller for one of the solvers, that one will do more work, which could potentially ease the burden on the other solver.

Birken [6] showed that only partially solving the constituent parts of a coupled system is a robust approach that avoids over-solving. Thus, for sufficiently well behaved systems, there is no danger in relaxed tolerances during the initial iterations of a timestep, because the solution gradually converges to the correct value as ensured by the outer tolerance. In a previous work [2] the Residual Balance method was used to solve a partial differential equation boundary value problem. In this work the Residual Balance method is applied to a reactor dynamics initial value problem.

3. EXAMPLE PROBLEM

This example problem is a homogeneous, one dimensional reactor with temperature feedback. The system initially has a positive reactivity which is then countered by a negative temperature feedback. A variety of timescales are involved, ranging from the prompt neutron jump to the diffusion of heat through the reactor. This relatively simple problem highlights the value of the Residual Balance method in the context of reactor dynamics. Higher fidelity simulations could be used, but the same algorithmic features can be demonstrated by low-order models.

The neutron transport system is described in simple terms by the diffusion equation and a single group of delayed neutrons.

$$\frac{1}{v}\frac{\partial \phi(x,t)}{\partial t} - \nabla \cdot D(T)\nabla\phi(x,t) + \Sigma_a(T)\phi(x,t) - (1-\beta)\nu\Sigma_f\phi(x,t) - \lambda C(x,t) = 0 \tag{2}$$

$$\frac{\partial C(x,t)}{\partial t} - \beta\nu\Sigma_f\phi(x,t) + \lambda C(x,t) = 0 \tag{3}$$

The governing equation for the thermal system is

$$\rho(T)c_p(T)\frac{\partial T(x,t)}{\partial t} - \nabla \cdot k(T)\nabla T(x,t) - E_f\Sigma_f\phi(x,t) = 0\,. \tag{4}$$

The length of the problem domain is $7.5\ \mathrm{cm}$ with reflected boundary conditions on one end. At the other end there is no incoming neutron current and the temperature is held fixed at $100\ ^\circ\mathrm{C}$. The initial

conditions for the flux and delayed neutron precursor concentration are provided by a steady-state (eigenvalue) problem. Initially, the temperature is $100\ ^{\circ}\mathrm{C}$.

The variables and problem parameters are listed in Table I. The energy per fission is adjusted so that the initial power is $14.4\ \mathrm{W}$ for the whole $15\ \mathrm{cm}$ domain. In order create a delayed supercritical (k=1.00040) system, $\nu = 1.765$ was selected. The following relations were developed to approximate Doppler broadening in the one group diffusion equation

$$D(T) = \frac{1}{3}\left(0.01\left(\sqrt{\frac{100+273.15}{T+273.15}}-1\right)+1\right), \tag{5}$$

and

$$\Sigma_a(T) = 0.03\left(0.01\left(\sqrt{\frac{T+273.15}{100+273.15}}-1\right)+1\right). \tag{6}$$

Table I. Problem parameters.

Variable	Value	Units	Description
ϕ	*	$\mathrm{cm}^{-2}\,\mathrm{sec}^{-1}$	Normalized neutron flux
C	*	cm^{-3}	Delayed neutron precursor concentration
T	*	$^{\circ}\mathrm{C}$	Temperature
v	220000	$\mathrm{m/sec}$	Neutron velocity
D	$1/3$*	cm	Diffusion coefficient at $T = 100\ ^{\circ}\mathrm{C}$
Σ_a	0.03*	cm^{-1}	Macroscopic absorption cross section at $T = 100\ ^{\circ}\mathrm{C}$
β	0.007	-	Delayed neutron fraction
ν	1.765	-	Neutrons per fission
Σ_f	0.024	cm^{-1}	Macroscopic fission cross section
λ	0.08	sec^{-1}	Delayed neutron precursor decay constant
k	0.1*	$\mathrm{W/cmK}$	Thermal conductivity
ρ	5*	$\mathrm{g/cm}^3$	Density
c_p	1*	$\mathrm{J/gK}$	Specific heat
E_f	150	J	Energy per normalized fission

* Value varies

3.1. Implementations Notes

The Backwards Euler method was selected for time integration. A mesh size of 75×5 was used for the example transient problem. Although the problem is one-dimensional, a two-dimensional mesh was used in order to simplify visualization and integration over the boundaries. The integrals over the domain are not affected because the second dimension is of unit depth. Convergence studies in time and space show that the peak flux and total energy deposition are within 3% of their converged values.

The problem has 1368 degrees of freedom and was run on a six-core workstation. Future work will naturally involve larger demonstration problems, but many characteristics of the new coupling method can be examined without wasting cluster resources.

The Residual Balance parameter A was set to 0.1, which usually results in alternating iterations of the constituent solvers. The solvers used a relative tolerance of 1×10^{-8} and a minimum absolute tolerance of 1×10^{-12}. The relative tolerance of the underlying linear solver was 1×10^{-5}, so the update step is essentially exact. A timestep was considered converged if the total residual (that is, the sum of the residuals from the two component codes) was reduced by a factor of 1×10^{-8} or if it fell below 1×10^{-12}.

4. RESULTS

MOOSE supports the operator splitting, Picard Iteration, and monolithic approaches to multiphysics. For this example problem it is easy enough to implement and compare several approaches. The Residual Balance method is an improvement on Picard Iteration, but it is not intended to replace full coupling. The fully coupled results are included as the reference case and as a reminder that investing in a complex solver will often earn substantial performance dividends later.

Initially the system is supercritical so there is a 6.06% prompt jump in the flux, followed by a delayed critical rise. As the flux increases, the fission heating rate increases and Doppler broadening occurs. After 165 seconds, the flux peaks and the temperature feedback shuts the reactor down. Figure 4 depicts this transient for a point in the middle of the symmetric domain.

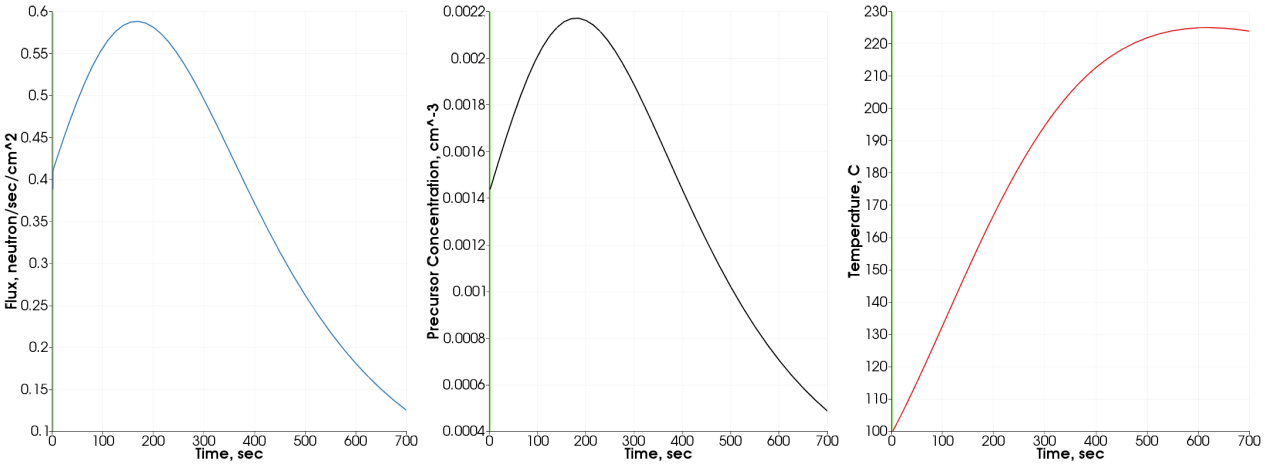

Figure 4. Normalized flux, delayed neutron precursor concentration, and temperature at the midpoint displayed as a function of time.

It is important to note that each solution technique arrives at the same answer. While an operator

splitting method would accumulate errors from unresolved nonlinearities, the Picard (global) iterations at each time step will converge to the same solution as the fully coupled method. The only discrepancies stem from the tolerance parameters, thus the solutions agree to the seventh digit. The peak flux is 0.588 (51.6% above the initial value) and the total energy generated in the 700 sec transient is 10.1 kJ. The center temperature peaks at about 225 °C after 625 sec.

The results listed in Table II show that the Residual Balance method achieves a time savings of up to 24% compared to Picard Iteration in the reactor transient problem. This is accomplished by substantially reducing the number of constituent solver iterations, despite increasing the number of global iterations. However, a fully coupled implementation attains a 73% reduction in solution time compared to Picard Iteration. Thus, fully coupled methods should be used whenever it is feasible to do so.

The Residual Balance method is superior to Picard Iteration in strongly coupled problems. Strongly coupled problems require multiple global iterations to resolve the multiphysics feedbacks. Because the Residual Balance method does fewer constituent iterations per global iteration it is a more efficient way to solve these problems. Transient problems often favor the Residual Balance method, because the solution is similar between timesteps and thus the problem is dominated by global iterations rather than constituent iterations.

Table II. Performance Comparison.

Method	Integral Power, kJ	Iterations Global	Iterations Constituent	Computational Time
JFNK	10.11142	127	n/a	0.266
Picard Iteration	10.11144	124	492	1
Residual Balance	10.11147	132	335	0.760

Weakly coupled problems may favor Picard Iteration. For problems requiring more constituent iterations than global iterations, the optimal method depends on the relative cost of global and constituent iterations. The Residual Balance method will minimize the number of constituent iterations, but at the cost of some additional global iterations. As the timestep size decreases—or the coupling becomes weaker—the advantage of the Residual Balance method diminishes. Table III shows the results from the same example problem but with weaker coupling because the timestep has been cut in half. From a different perspective, the optimal value of the parameter A tends towards zero for weakly coupled problems.

Table III. Performance comparison for a more weakly coupled problem.

Method	Integral Power, kJ	Iterations Global	Iterations Constituent	Computational Time
JFNK	10.1459	184	n/a	0.294
Picard Iteration	10.1451	174	694	1
Residual Balance	10.1442	255	585	1.03

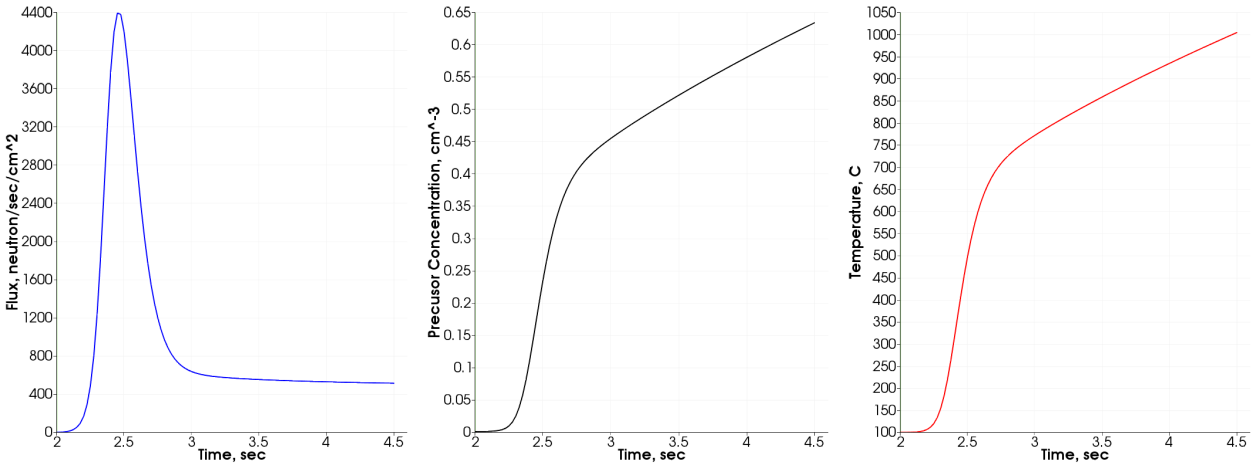

Figure 5. Normalized flux, delayed neutron precursor concentration, and temperature at the midpoint displayed as a function of time for a super-prompt critical transient. The problem starts at $t = 2$ and lasts 2.5 sec.

For a final variation on this example problem, 706 pcm of reactivity was inserted by setting $\nu = 1.78$. Figure 4 depicts the prompt evolution of the transient. This super-prompt critical system is quite strongly coupled, thus lending itself to the Residual Balance method. After 2.5 sec the power begins to level off, so only 100 timesteps were required. The operator splitting method without Picard Iterations was also used to solve this problem in order to demonstrate its unsatisfactory behavior. Running with the same timestep size results in grossly miscalculating the flux peak magnitude. So the operator splitting method was run again, but with ten times more timesteps, which still results in poor accuracy. The results are summarized in Table IV. In this strongly coupled problem the Residual Balance method does very well, running 40% faster than Picard Iteration.

Table IV. Performance comparison for a super-prompt critical example. Operator Splitting 2 was run with the timestep size reduced by a factor of ten.

Method	Peak Flux	Iterations		Computational Time
		Global	Constituent	
Operator Splitting 1	5922.6	100	503	0.549
Operator Splitting 2	4967.8	1000	3313	2.50
Picard Iteration	4393.7	268	1385	1
Residual Balance	4394.0	330	691	0.597

5. CONCLUSION

A modification to Picard Iteration has been developed which enables significant savings in terms of simulation time. Reactor transients have a strong coupling between neutron and thermal transport

which favors the Residual Balance method. It decreases the number of constituent steps required, possibly at the expense of additional data transfers, and results in a considerable reduction in compute time.

All tests conducted so far indicate that the robustness and accuracy properties of Picard Iteration have not been altered in the Residual Balance method (of course there will be some pathological cases where this is not true). Combined with the fact that the method is not difficult to implement, it is almost always worthwhile to do so.

ACKNOWLEDGMENTS

This material is based upon work supported under a Department of Energy, Office of Nuclear Energy, Integrated University Program Graduate Fellowship.

REFERENCES

[1] A. G. Mylonakis *et al.* "Multi-physics and multi-scale methods used in nuclear reactor analysis." *Annals of Nuclear Energy*, **72**: pp. 104–119 (2014).

[2] J. P. Senecal and W. Ji. "Comparison of novel multiphysics coupling methods in moose." In: *Trans. Am. Nucl. Soc.*, volume 113, (pp. 707–710) (2015).

[3] D. R. Gaston *et al.* "MOOSE: a parallel computational framework for coupled systems of nonlinear equations." *Nuclear Engineering and Design*, **239**: pp. 1768–1778 (2009).

[4] D. Keyes *et al.* "Multiphysics simulations: Challenges and opportunities." *International Journal of High Performance Computing Applications*, **27(1)**: pp. 4–83 (2012).

[5] D. A. Knoll and D. E. Keyes. "Jacobian-free newton-krylov methods: a survey of approaches and applications." *Journal of Computational Physics*, **193**: pp. 357–397 (2004).

[6] P. Birken. "Termination criteria for inexact fixed-point schemes." *Numerical Linear Algebra with Applications*, **22(4)**: pp. 702–716 (2015).