INL/CON-16-37879
PREPRINT

Implementation of the Improved Quasi-static Method in RATTLESNAKE/MOOSE for Time-dependent Radiation Transport Modelling

Physics of Reactors Conference (PHYSOR 2016)

Zachary M. Prince and Jean C. Ragusa
(Texas A&M University)

Yaqi Wang
(Idaho National Laboratory)

February 2016

The INL is a
U.S. Department of Energy
National Laboratory
operated by
Battelle Energy Alliance

IMPLEMENTATION OF THE IMPROVED QUASI-STATIC METHOD IN RATTLESNAKE/MOOSE FOR TIME-DEPENDENT RADIATION TRANSPORT MODELLING

Zachary M. Prince and Jean C. Ragusa
Department of Nuclear Engineering
Texas A&M University, College Station, TX, USA
zachmprince@tamu.edu
jean.ragusa@tamu.edu

Yaqi Wang
Idaho National Laboratory
yaqi.wang@inl.gov

ABSTRACT

Because of the recent interest in reactor transient modeling and the restart of the Transient Reactor (TREAT) Facility, there has been a need for more efficient, robust methods in computation frameworks. This is the impetus of implementing the Improved Quasi-Static method (IQS) in the RATTLESNAKE/MOOSE framework. IQS has implemented with CFEM diffusion by factorizing flux into time-dependent amplitude and spacial- and weakly time-dependent shape. The shape evaluation is very similar to a flux diffusion solve and is computed at large (macro) time steps. While the amplitude evaluation is a PRKE solve where the parameters are dependent on the shape and is computed at small (micro) time steps. IQS has been tested with a custom one-dimensional example and the TWIGL ramp benchmark. These examples prove it to be a viable and effective method for highly transient cases. More complex cases are intended to be applied to further test the method and its implementation.

Key Words: **RATTLESNAKE, MOOSE, transient, factorization**

1. INTRODUCTION

The anticipated restart of Transient Reactor Testing (TREAT) Facility at Idaho National Laboratory (INL) has brought significant attention and opportunity to transient modeling. TREAT, which was operational from 1954 to 1994, was designed to test nuclear fuels by subjecting them to various degrees of neutron pulses, from minor transients to accident cases. Neutron transient modeling has always been computationally expensive due to implicit time-stepping caused by the neutron velocity values. Even with the vast improvements in computing technology, straightforward discretization of neutron conservation equations remain computationally challenging for real-world cases. Therefore, methods that

improve on computational speed significantly, at minimal detriment to accuracy, are highly desired. The Department of Energy (DOE) and INL have invested a substantial effort in modeling and simulation for TREAT. This paper presents an implementation of the improved quasi-static (IQS) method for time-dependent neutron transport and diffusion equations with the multiphysics framework MOOSE [1], notably its radiation transport application, RATTLESNAKE.

The improved quasi-static (IQS) method is a spatial kinetics method that involves factorizing the flux solution into space- and time-dependent components [2, 3]. These components are the flux amplitude and its shape. Amplitude is only time-dependent, while the shape is both space- and time-dependent. However, the impetus of the method is the assumption that the shape is only weakly dependent on time; therefore, the shape may not require an update at the same frequency of the amplitude function, but only on macro-time steps. As opposed to other forms of quasi-static approximations, the shape is updated consistently with its own diffusion equation after the amplitude is evaluated. The results of IQS may only differ from straightforward, temporal discretization because the time discretization truncation error in the shape increases with a larger time-step size.

Implementing IQS to RATTLESNAKE in INL's MOOSE framework is an obvious endeavor to enable high-fidelity modeling of the TREAT facility. The rest of this paper will briefly describe the derivation of IQS (in the diffusion setting for brevity), its current application to RATTLESNAKE using the the MultiApp Picard iteration capabilities of MOOSE, and results from examples testing the method's viability and effectiveness.

2. BACKGROUND

In this Section, we recall the equations for the IQS method, starting from the standard multigroup diffusion equations written below:

$$\begin{aligned}\frac{1}{v^g}\frac{\partial \phi^g}{\partial t} =& \frac{\chi_p^g}{k_{\mathit{eff}}}\sum_{g'=1}^{G}(1-\beta)\nu^{g'}\Sigma_f^{g'}\phi^{g'} - \left(-\nabla\cdot D^g\nabla + \Sigma_r^g\right)\phi^g \\ &+ \sum_{g'\neq g}^{G}\Sigma_s^{g'\to g}\phi^{g'} + \sum_{i=1}^{I}\chi_{d,i}^g\lambda_i C_i\,, \quad 1\leq g\leq G \end{aligned} \tag{1}$$

$$\frac{dC_i}{dt} = \frac{\beta_i}{k_{eff}}\sum_{g=1}^{G}\nu^g\Sigma_f^g\phi^g - \lambda_i C_i\,, \quad 1\leq i\leq I \tag{2}$$

with

$$\beta = \sum_{i=1}^{I}\beta_i \tag{3}$$

Factorization is an important step in the derivation of the IQS method. The factorization approach leads to a decomposition of the multigroup flux into the product of a time-dependent amplitude (p) and

a space-/time-dependent multigroup shape (φ):

$$\phi^g(\vec{r},t) = p(t)\varphi^g(\vec{r},t) \tag{4}$$

To obtain the amplitude equations, the multigroup equations are multiplied by a weighting function, typically the initial adjoint flux (ϕ^*), and then integrated over phase-space. For brevity, the inner product over space will be represented with parenthetical notation:

$$\int_D \phi^{*g}(\vec{r}) f^g(\vec{r}) d^3r = (\phi^{*g}, f^g) \tag{5}$$

In order to impose uniqueness of the factorization, one requires the following:

$$\sum_{g=1}^{G} \left(\phi^{*g}, \frac{1}{v^g}\varphi^g \right) = \textit{constant} \tag{6}$$

After some manipulation, the standard point reactor kinetics equations (PRKE) for the amplitude solution are obtained:

$$\frac{dp}{dt} = \left[\frac{\rho - \bar{\beta}}{\Lambda} \right] p + \sum_{i=1}^{I} \bar{\lambda}_i \xi_i \tag{7}$$

$$\frac{d\xi_i}{dt} = \frac{\bar{\beta}_i}{\Lambda} - \bar{\lambda}_i \xi_i \quad 1 \le i \le I \tag{8}$$

Where the functional coefficients are calculated using the space-/time-dependent shape function as follows:

$$\frac{\rho - \bar{\beta}}{\Lambda} = \frac{\sum_{g=1}^{G} \left(\phi^{*g}, \frac{\chi_p^g}{k_{eff}} (1-\beta) \sum_{g'=1}^{G} \nu^{g'} \Sigma_f^{g'} \varphi^{g'} + \sum_{g' \neq g}^{G} \Sigma_s^{g' \to g} \varphi^{g'} - \left(-\nabla \cdot D^g \nabla + \Sigma_r^g \right) \varphi^g \right)}{\sum_{g=1}^{G} \left(\phi^{*g}, \frac{1}{v^g}\varphi^g \right)} \tag{9}$$

$$\frac{\bar{\beta}}{\Lambda} = \sum_{i=1}^{I} \frac{\bar{\beta}_i}{\Lambda} = \sum_{i=1}^{I} \frac{1}{k_{eff}} \frac{\sum_{g=1}^{G} (\phi^{*g}, \chi_{d,i}^g \beta_i \sum_{g'=1}^{G} \nu^{g'} \Sigma_f^{g'} \varphi^{g'})}{\sum_{g=1}^{G} \left(\phi^{*g}, \frac{1}{v^g}\varphi^g \right)} \tag{10}$$

$$\bar{\lambda}_i = \frac{\sum_{g=1}^{G} (\phi^{*g}, \chi_{d,i}^g \lambda_i C_i)}{\sum_{g=1}^{G} (\phi^{*g}, \chi_{d,i}^g C_i)} \tag{11}$$

Finally, the shape equations are solved for the shape. The shape equations are similar to the orignal diffusion equations:

$$\begin{aligned} \frac{1}{v^g} \frac{\partial \varphi^g}{\partial t} =& \frac{\chi_p^g}{k_{\mathit{eff}}} \sum_{g'=1}^{G} (1-\beta) \nu^{g'} \Sigma_f^{g'} \varphi^{g'} - \left(-\nabla \cdot D^g \nabla + \Sigma_r^g + \frac{1}{v^g} \frac{1}{p} \frac{dp}{dt} \right) \varphi^g \\ &+ \sum_{g' \neq g}^{G} \Sigma_s^{g' \to g} \varphi^{g'} + \frac{1}{p} \sum_{i=1}^{I} \chi_{d,i}^g \lambda_i C_i \,, \quad 1 \le g \le G \end{aligned} \tag{12}$$

However, the amplitude and shape equations form a system of coupled equations: the coefficients appearing in the PRKEs depend upon the shape solution while the shape equation has a kernel dependent on amplitude and its derivative. Because solving for the shape can be expensive, especially in two or three dimensions, it is attractive to make the assumption that the shape is weakly time-dependent so the shape can be computed after a multitude of PRKE calculations which is the root of IQS. This is depicted schematically in Fig. 1:

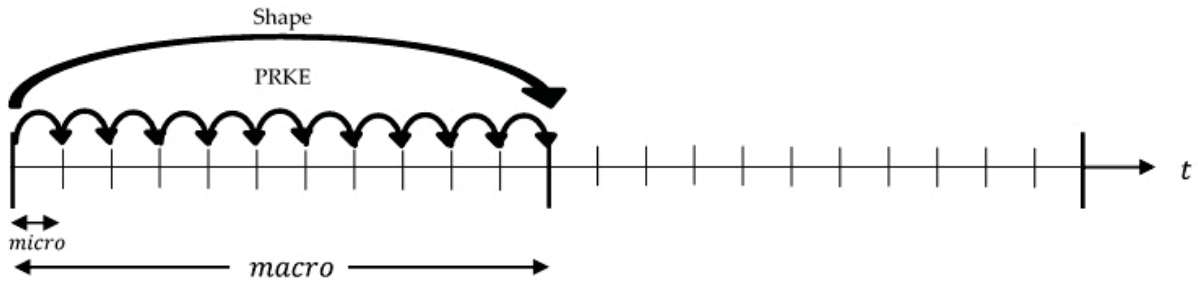


Figure 1: IQS method solution process

Additionally, to improve consistency and accuracy, each macro time step can be iterated so the best shape is used to compute power at the micro time steps. Within the MOOSE framework, nonlinear systems can be tackled in two manners: with Newton's method (usually, a preconditioned Jacobian-free version) and with Picard's iterations (fixed-point method). The latter is employed in the work. This iteration process must converge the shape such that the uniqueness condition $(\frac{d}{dt}\sum_{g=1}^{G}\left(\phi^{*g},\frac{1}{v^g}\varphi^g\right)=0)$ is preserved.

3. IMPLEMENTATION IN RATTLESNAKE

MOOSE, or Multiphysics Object-Oriented Simulation Environment, is a finite-element-based framework developed by INL and is equipped with advanced nonlinear solvers. Rattlesnake is a module of MOOSE meant for neutronics and radiation transport problems. RATTLESNAKE is a radiation transport application within MOOSE and can be coupled to other physics via a Newton or a Picard approach. Implementing the IQS in RATTLESNAKE is meant to enhance its transient modeling capability. RATTLESNAKE utilizes an action system which initiates kernels, user objects, and postprocessors; these typically need to be added manually to the input file, but due to the large phase-space of neutron transport approximations, an automated action system is invoked to add the required MOOSE objects. When implementing the IQS, the action system and its associated MOOSE objects need to be updated. For brevity, we describe the implementation in the case of the CFEM Diffusion action system; similar developments are carried out for the DFEM Diffusion action system, the S_n Transport action system, ... We discuss the CFEM Diffusion action system in detail:

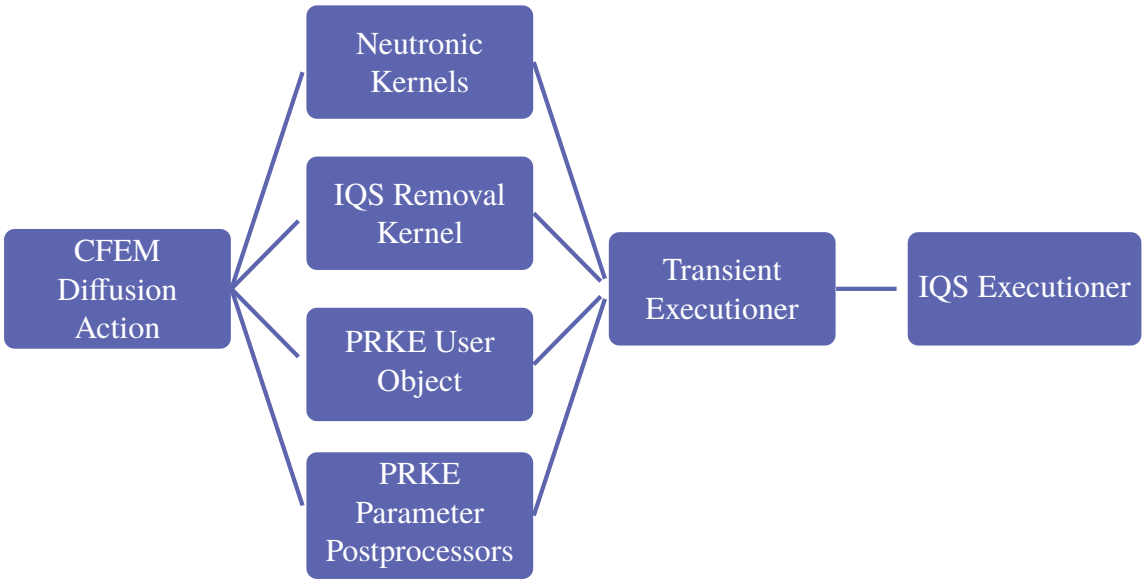


Figure 2: CFEM Diffusion Action System Diagram

3.1. Action System

IQS derives its uniqueness from the executioner type; however, some additional changes needed to be carried out in the RATTLESNAKE/YAK action system in order to support IQS execution. First, changes needed to be made in order to evaluate the shape equation. The shape equation, after some manipulation, is very similar to the time-dependent flux equation, as seen in Eq. (12). To enable RATTLESNAKE to solve this shape equation in lieu of the standard diffusion equation, an additional removal kernel has to be instantiated to evaluate the quantity $\frac{1}{vp}\frac{dp}{dt}\varphi$ and added to FEM weak form when the IQS executioner is selected. Second, four postprocessors are created in order to calculate the PRKE parameters. The parameter calculations were split into the following item: $\frac{\bar{\beta}_i}{\Lambda}$ numerator, $\bar{\lambda}_i$ numerator/denominator, $\frac{\rho}{\Lambda}/\frac{\bar{\beta}}{\Lambda}$ denominator, and $\frac{\rho-\bar{\beta}}{\Lambda}$ numerator. The first three are relatively simple, only relying on material properties and solution quantities. The $\frac{\rho-\bar{\beta}}{\Lambda}$ numerator requires the use of MOOSE's residual `save_in` feature, which saves the residual from a calculated kernel or boundary contribution in the shape evaluation to an auxiliary variable. Finally, a user object was created to pull together all the postprocessor values and carryout the numerator/denominator divisions that were then passed to the executioner.

3.2. Precursor Integration

This section presents two different time-integration methods to solve coupled IQS shape + precursor equations, recalled below using, for simplicity, a single neutron group and a single precursor group.

$$\frac{1}{v}\frac{\partial \varphi}{\partial t} = \nu\Sigma_f(1-\beta)\varphi - \left(-\nabla\cdot D\nabla + \Sigma_a + \frac{1}{v}\frac{1}{p}\frac{dp}{dt}\right)\varphi + \frac{1}{p}\lambda C \tag{13}$$

$$\frac{dC}{dt} = \beta\nu\Sigma_f\varphi p - \lambda C \tag{14}$$

First, we note that we could keep this system of two time-dependent equations and solve it as a coupled system. However, this is unnecessary and a memory expensive endeavor because the precursor equation is only an ODE and not a PDE. Instead, one may discretize in time the shape equation, which typically requires the knowledge of the precursor concentrations at the end of the time step. This precursor value is taken from the solution, numerical or analytical, of the precursors ODE. This document will discuss two techniques for solving the precursor equation. First is a time discretization method that is currently being implemented in RATTLESNAKE. The second is a analytical integration of the precursors, the latter method has proven to be more beneficial for IQS convergence.

3.2.1. Time Discretization using the Theta Method

A fairly simple way to evaluate the precursor equation is to employ the θ-scheme ($0 \leq \theta \leq 1$), explicit when $\theta = 0$, implicit when $\theta = 1$, and Crank-Nicholson when $\theta = 1/2$). Generally, if there is a function u whose governing equation is $\frac{du}{dt} = f(u,t)$, then the θ-discretization is

$$\frac{u^{n+1}-u^n}{\Delta t} = (1-\theta)f(u^n,t) + \theta f(u^{n+1},t)\,. \tag{15}$$

Applying this to the precursor equation:

$$\frac{C^{n+1}-C^n}{\Delta t} = (1-\theta)\beta S_f^n p^n - (1-\theta)\lambda C^n + \theta\beta S_f^{n+1}p^{n+1} - \theta\lambda C^{n+1} \tag{16}$$

Where S_f is the fission source equivalent for shape:

$$S_f^n = (\nu\Sigma_f)^n\varphi^n \tag{17}$$

Rearranging to solve for the precursor at the end of the time step yields

$$C^{n+1} = \frac{1-(1-\theta)\Delta t\lambda}{1+\theta\Delta t\lambda}C^n + \frac{(1-\theta)\Delta t\beta}{1+\theta\Delta t\lambda}S_f^n p^n + \frac{\theta\Delta t\beta}{1+\theta\Delta t\lambda}S_f^{n+1}p^{n+1} \tag{18}$$

Reporting this value of C^{n+1}, one can solve for the shape φ^{n+1} as a function of φ^n and C^n (and p^n, p^{n+1}, $dp/dt|_n$ and $dp/dt|_{n+1}$). Once φ^{n+1} has been determined, C^{n+1} is updated. YAK currently implements both implicit and Crank-Nicholson as options for precursor evaluation.

3.2.2. Analytical Integration

Through prototyping, it has been found that neither implicit nor Crank-Nicholson time discretization of precursors are preferable methods for solving the shape equation in IQS. It has been found that these discretizations result in a lack of convergence of the shape over the IQS iteration process. In order to remedy the error, a analytical representation of the precursors was implemented in the prototype and the shape solution was able to converge (the normalization constant of the IQS method can be preserved to 10^{-10} while the theta-scheme only allowed convergence in the normalization factor to about 10^{-3}). The following section shows how this method was implemented in the prototype and RATTLESNAKE.

Using an exponential operator, the precursor equation can be analytically solved for:

$$\int_{t_n}^{t_{n+1}} C(t')e^{\lambda t'}dt' = \int_{t_n}^{t_{n+1}} \beta(t')S_f(t')p(t')e^{\lambda t'}dt' \tag{19}$$

yielding

$$C^{n+1} = C^n e^{-\lambda(t_{n+1}-t_n)} + \int_{t_n}^{t_{n+1}} \beta(t')S_f(t')p(t')e^{-\lambda(t_{n+1}-t')}dt' \tag{20}$$

Because β and S_f being integrated are not known continuously over the time step, they can be interpolated linearly over the macro step. Such that:

$$h(t) = \frac{t_{n+1}-t}{t_{n+1}-t_n}h^n + \frac{t-t_n}{t_{n+1}-t_n}h^{n+1} \quad t_n \le t \le t_{n+1} \tag{21}$$

However, for the PRKE solve, we do have a very accurate representation of $p(t')$ over the time interval $[t_n, t_{n+1}]$.

Finally, we have the final expression for the analytical value for C^{n+1}:

$$C^{n+1} = C^n e^{-\lambda \Delta t} + \left(a_3 \beta^{n+1} + a_2 \beta^n\right) S_f^{n+1} + \left(a_2 \beta^{n+1} + a_1 \beta^n\right) S_f^n \tag{22}$$

Where the integration coefficients are defined as:

$$a_1 = \int_{t_n}^{t_{n+1}} \left(\frac{t_{n+1} - t'}{\Delta t}\right)^2 p(t') e^{-\lambda(t_{n+1} - t')} dt' \tag{23}$$

$$a_2 = \int_{t_n}^{t_{n+1}} \frac{(t' - t_n)(t_{n+1} - t')}{(\Delta t)^2} p(t') e^{-\lambda(t_{n+1} - t')} dt' \tag{24}$$

$$a_3 = \int_{t_n}^{t_{n+1}} \left(\frac{t' - t_n}{\Delta t}\right)^2 p(t') e^{-\lambda(t_{n+1} - t')} dt' \tag{25}$$

The amplitude (p) is included in the integration coefficient because it has been highly accurately calculated in the micro step scheme, so a piecewise interpolation between those points can be done to maximize accuracy.

The prototype code uses Matlab software to interpolate the amplitude between micro steps and a quadrature integration for the coefficients. So the challenge for RATTLESNAKE was to replicate this procedure: passing the amplitude vector to the DNP auxkernel, interpolating it, and integrating the coefficients.

3.3. Executioner

The IQS executioner derives from the Transient executioner in MOOSE. The IQS executioner contains a loop over micro time steps that solves the PRKEs and then passes the values for p and dp/dt at times corresponding to the macro-time steps into the Transient executioner in order to solve for the shape equation at each macro step. The PRKEs are solved with two options, backward Euler and third order implicit Runge-Kutta method, within the Executioner. The IQS executioner also supplements Transient's Picard iteration process by adding its own error criteria for the IQS method:

$$\text{Error}_{\text{IQS}} = \left| \frac{\left(\phi_g^*, \frac{1}{v_g}\varphi_g^n\right)}{\left(\phi_g^*, \frac{1}{v_g}\varphi_g^0\right)} - 1 \right| \tag{26}$$

The use of the Picard iteration capability of MOOSE's executioner will enable solving the nonlinear IQS equations along with other nonlinearly coupled multiphysics (e.g., thermal-hydraulics) using different time step sizes for neutronics and the other coupled physics.

4. RESULTS

This section describes results of an examples that tests the IQS implementation and shows its effectiveness on computation speed and accuracy. Two examples were selected for this purpose. The first is a homogeneous one-group problem, subjected to a heterogenous material change (absorption cross-section change as a ramp in time for a subset of the geometry). The second is the two-dimensional TWIGL ramp transient benchmark, described further.

4.1. One-Dimensional Custom Example

The example is very simple and computes quickly, it entails a one dimensional, heterogeneous 400 cm slab with a varying absorption cross section. Figure 3 how the regions of the slab are divided and Table I shows the initial material properties. Table II shows the ramp of the absorption cross-section of each region.

1	1	1	1	2	3	1	1	1	1	1	1	1	1	4	4	1	1	1	1

Figure 3: 1-D heterogeneous slab region identification

Region	$D(cm)$	$\Sigma_a(cm^{-1})$	$\nu\Sigma_f(cm^{-1})$	$v(cm/s)$	β	$\lambda(s^{-1})$
1	1.0	1.1	1.1	1,000	0.006	0.1
2	1.0	1.1	1.1	1,000	0.006	0.1
3	1.0	1.1	1.1	1,000	0.006	0.1
4	1.0	1.1	1.1	1,000	0.006	0.1

Table I: 1-D heterogeneous slab material properties and problem parameters

Figure 8 shows the power at each macro time step as compared to the traditional brute force (full flux time discretization) method. The strong correlation between the two curves shows that IQS is consistent with a proven method for a highly transient example. Figure **??** shows that IQS is not only consistent for this example, but also has a better error constant in the convergence study. Figures 5 - 7 plots shape changes in the IQS method, showing where the shape solution is necessary and a simple PRKE evaluation is inadequate.

Material Property	0.0 s	0.1 s	0.6 s	1.0 s	1.7 s
$\Sigma_{a,2}(cm^{-1})$	1.1	1.1	1.095	1.095	1.095
$\Sigma_{a,3}(cm^{-1})$	1.1	1.1	1.09	1.09	1.1
$\Sigma_{a,4}(cm^{-1})$	1.1	1.1	1.105	1.105	1.105

Table II: 1-D heterogeneous slab absorption cross-section slope perturbation

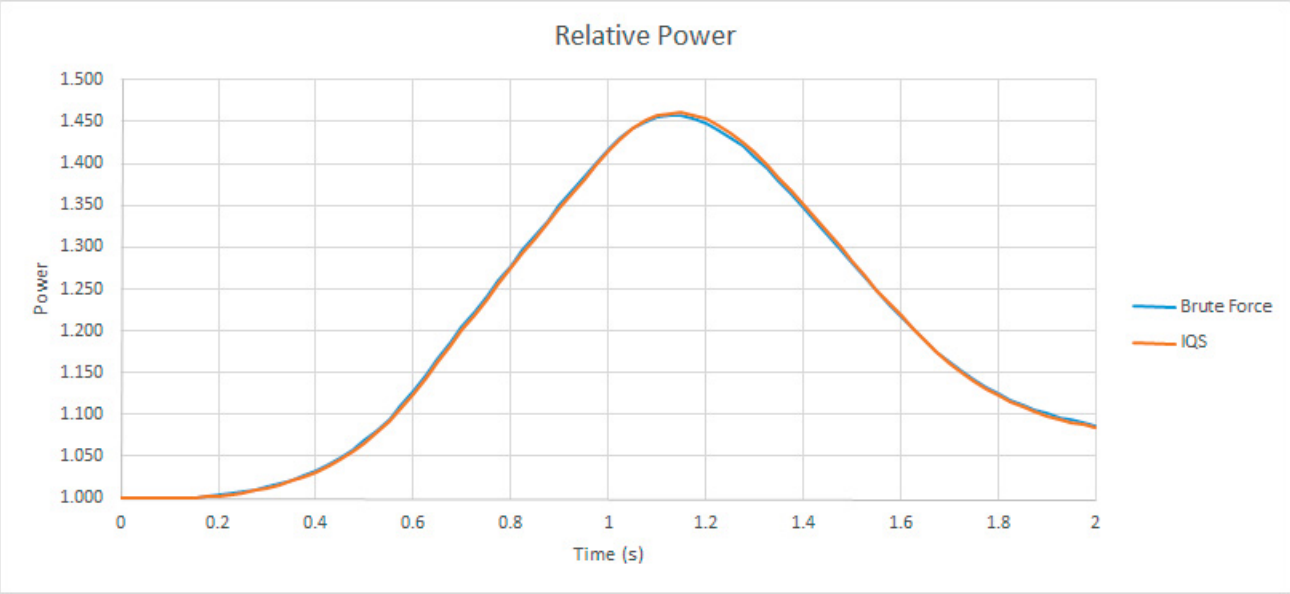


Figure 4: Power level comparison of 1D heterogeneous example between IQS and Brute Force using $\Delta t = 0.025$

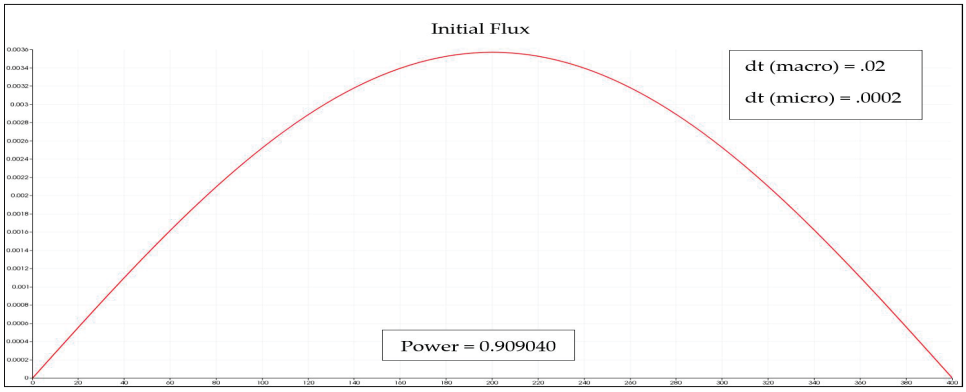


Figure 5: Initial Flux Plot

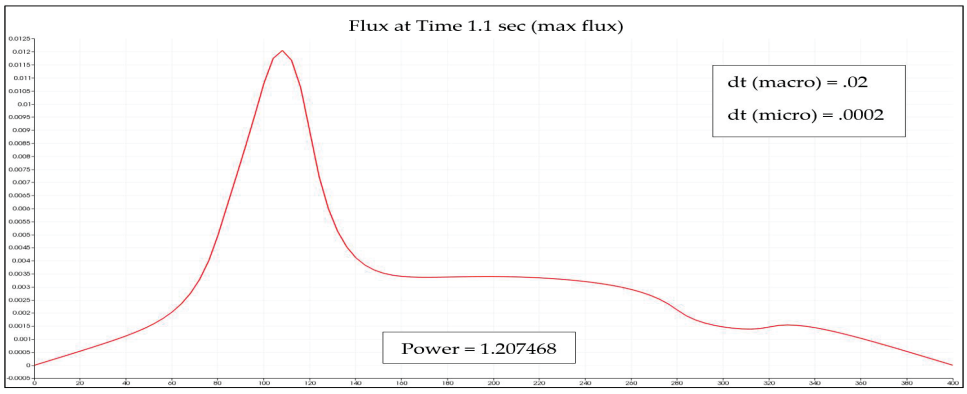


Figure 6: Flux Plot when Absorption Cross Section is at Minimum

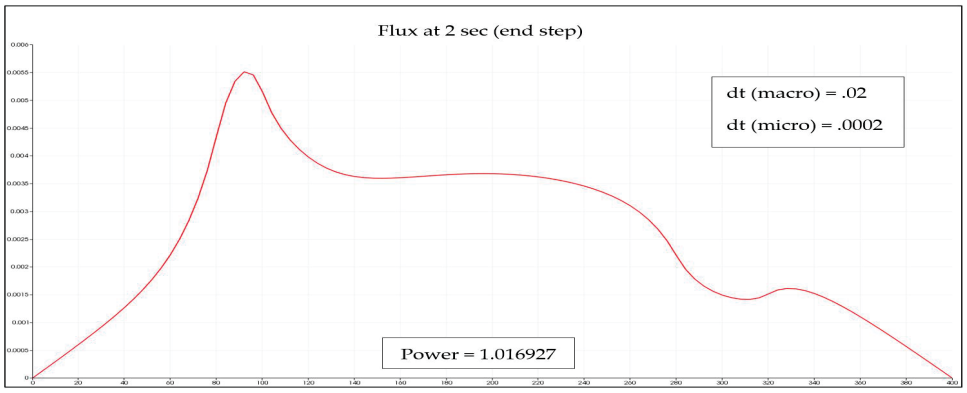


Figure 7: Final Flux Computation (not steady-state)

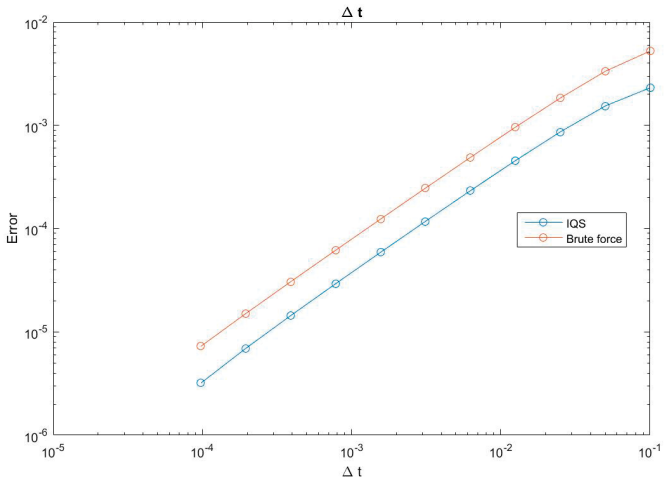


Figure 8: Error convergence comparison of 1D hetergenous example

4.2. TWIGL Benchmark

This benchmark problem originates from the Argonne National Lab Benchmark Problem Book. It is a 2D, 2-group reactor core model with no reflector region shown in Figure 9. Table III shows the material properties of each fuel region and the ramp perturbation of Material 1.

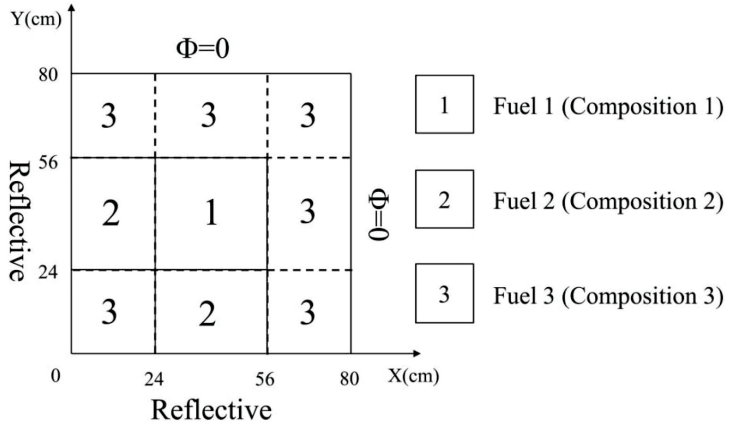


Figure 9: TWIGL benchmark problem description

Material	Group	$D(cm)$	$\Sigma_a(cm^{-1})$	$\nu\Sigma_f(cm^{-1})$	χ	$\Sigma_s(cm^{-1})$ $g \to 1$	$\Sigma_s(cm^{-1})$ $g \to 2$
1	1	1.4	0.010	0.007	1.0	0.0	0.01
	2	0.4	0.150	0.200	0.0	0.0	0.00
2	1	1.4	0.010	0.007	1.0	0.0	0.01
	2	0.4	0.150	0.200	0.0	0.0	0.00
3	1	1.3	0.008	0.003	1.0	0.0	0.01
	2	0.5	0.050	0.060	0.0	0.0	0.00
	ν	$v_1(cm/s)$	$v_2(cm/s)$	β	$\lambda(1/s)$		
	2.43	1.0E7	2.0E5	0.0075	0.08		

Material 1 ramp perturbation:

$\Sigma_{a,2}(t) = \Sigma_{a,2}(0) \times (1 - 0.11667t) \quad t \leq 0.2s$

$\Sigma_{a,2}(t) = \Sigma_{a,2}(0) \times (0.97666t) \quad t > 0.2s$

Table III: 1-D heterogeneous slab absorption cross-section slope perturbation

Figures 10 and 11 show the IQS solution as compared with the Brute Force solution. It is important to note the IQS shape plot is scaled differently than the Brute Force flux plot because the amplitude term is not included, but the gradients of colors is comparable. These plots show that IQS is consistent

in more complex, higher dimensional problems in RATTLESNAKE. Finally, Figure 12 plots the error convergence of IQS and the Brute Force methods. The curves show the impressive convergence of IQS for the highly transience TWIGL example.

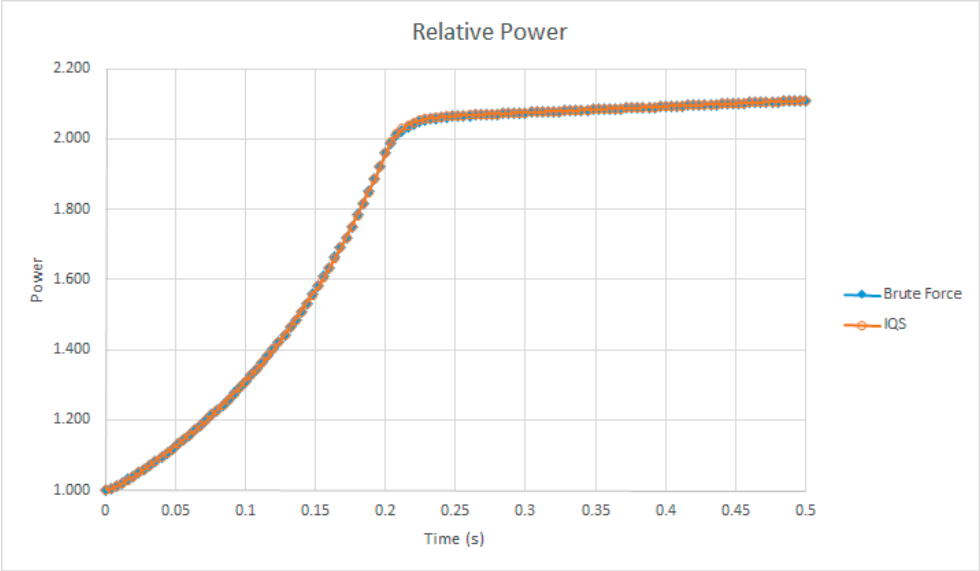
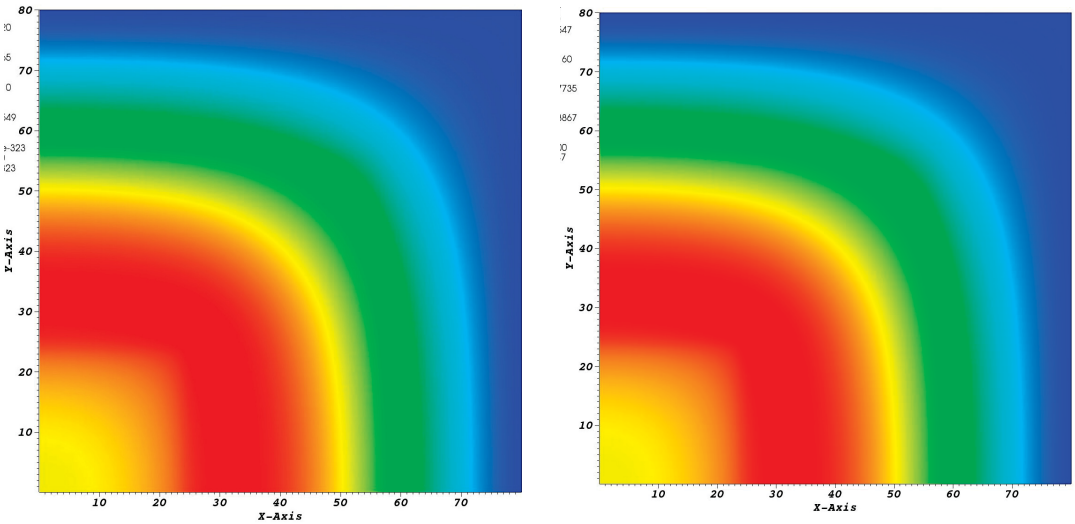


Figure 10: Power level comarison of 1D heterogeneous example between IQS and Brute Force using $\Delta t = 0.004$



(a) Brute force flux

(b) IQS Shape

Figure 11: TWIGL Benchmark flux/shape comparison at $t = 0.2$

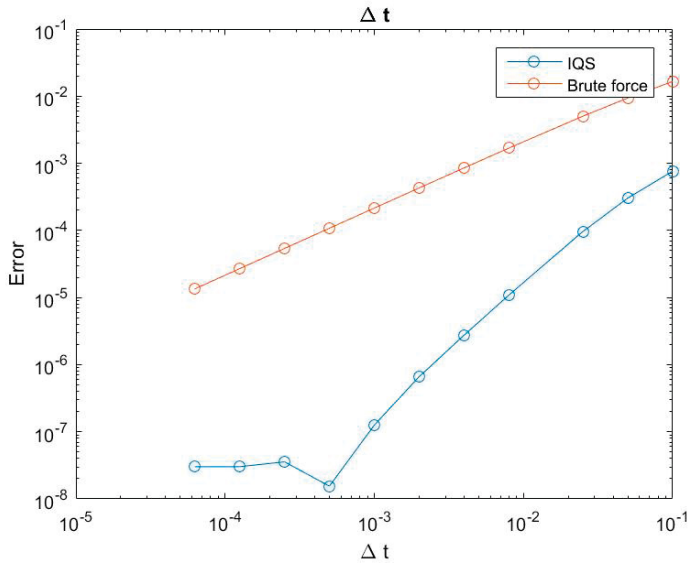


Figure 12: Error convergence comparison of TWIGL Benchmark

5. Conclusions

We have implemented the IQS method within RATTLESNAKE, part of the MOOSE framework. The implementation is complete for the CFEM Diffusion neutron equation and under testing for DFEM Diffusion and S_n transport equations. The application of IQS in MOOSE's Picard nonlinear solver was a relatively simple task using the object-oriented features of the framework. Once the implementation was completed for one action system, extension to other neutron discretizations is straightforward and elegant.

The examples presented in this paper prove that IQS was not only a properly implemented in the MOOSE framework, but a incredibly effective method for highly transient cases. More complex benchmarks (LRA and LMW benchmarks) and realistic cases (TREAT) are intended to be carried out using IQS (e.g., [4, 6]), including a multiphysics neutronics+heat conduction reactor dynamic problem [7].

REFERENCES

[1] D. Gaston *et al.* "Moose: A parallel computational framework for coupled systems of nonlinear equations." *Nucl. Engrg. Design*, **239**: pp. 1768 – 1778 (2009).

[2] K. Ott. "Quasi-static treatment of spatial phenomena in reactor dynamics." *Nuclear Science and Engineering*, **26**: p. 563 (1966).

[3] S. Dulla, E. H. Mund, and P. Ravetto. "The quasi-static method revisited." *Progress in Nuclear Energy*, **50(8)**: pp. 908 – 920 (2008).

[4] J. B. Yasinsky. "Some numerical experiments concerning space time reactor kinetics behavior." *Nuclear Science and Engineering*, **22**: p. 171 (1965).

[5] L. Hageman and J. Yasinsky. "Comparison of alternating direction time differencing method with other implicit method for the solution of the neutron group diffusion equations." *Nucl. Sci. Eng.*, **38**: pp. 8 – 32 (1969).

[6] S. Langenbuch *et al.* "Coarse-mesh flux-expansion method for the analysis of space- time effects in large light water reactor cores." *Nucl. Sci. Eng.*, **63**: pp. 437 – 456 (1969).

[7] Argonne Code Center. *Benchmark Problem Book, ANL- 7416, Suppl. 2. Technical report*, Argonne National Laboratory (1977).