Multiphysics Reactor-core Simulations Using the Improved Quasi-Static Method

Zachary M. Prince[a], Jean C. Ragusa[a]

[a] *Texas A&M University, Department of Nuclear Engineering, College Station, TX 77840, USA*

Abstract

The improved quasi-static method (IQS) is a rigorous space/time multiscale approach whereby the neutron flux is represented by a time-dependent amplitude and a time-space-energy dependent shape. The objective of the IQS factorization is to evaluate amplitude and shape on different time scales in order to reduce the computational burden associated with solving the multi-dimensional flux equations, while maintaining solution accuracy. The IQS decomposition leads to a nonlinear system of equations that requires iteration of shape and amplitude. IQS iteration techniques involve fixed-point (Picard) iteration with various convergence criteria and shape rescaling. Nonlinear convergence of each of these techniques is investigated. Verification of IQS with analysis of time step convergence is also investigated. Proper time step convergence is vital for implementation of time adaptive methods and error prediction. The time derivative of the shape function is discretized through fourth order using implicit-Euler, Crank-Nicolson, and backward difference formulae (BDF).

Keywords: Reactor dynamics, reactor kinetics, Improved quasi-static method, temporal convergence

1. Introduction

The improved quasi-static method (IQS) is a numerical technique devised for nuclear reactor transient analysis. It involves factorizing the neutron flux solution into a time-only-dependent component, the amplitude, and a space- and time-dependent component, the shape [1, 2, 3, 4, 5]. The amplitude solution satisfies the point reactor kinetic equations (PRKE) where the shape solution have been used to generate the PRKE coefficients (reactivity, effective fraction of delayed neutrons, mean generation time). The shape solution satisfies a modified time-dependent neutron balance equation. The rationale for the IQS method lies in the assumption that the shape function is weakly dependent

Email addresses: zachmprince@tamu.edu (Zachary M. Prince), jean.ragusa@tamu.edu (Jean C. Ragusa)

on time. Therefore, it is expected that the modified time-dependent neutron balance equations be less stiff than the original time-dependent neutron balance equations for the flux. As a result, the shape may not require to be solved for at the same frequency as the amplitude, but only on larger macro-time steps, which is expected to yield wall clock savings, especially in multi-dimensional geometries. The PRKE form a small system of ordinary differential equations (ODE) and solving them on a fine temporal grid is not a computational burden. As opposed to the standard PRKE approach, whereby a shape function is selected, typically once for an entire transient without updates, the IQS technique is obtained in a rigorously consistent manner from the time-dependent neutron balance equation.

Due to the factorization of the flux into a shape and an amplitude, the latter two variables are nonlinearly coupled. Ott in [1] first investigated the coupling of shape and amplitude in a quasi-static nature, but did not include the time derivative in the shape equation. Later, in [6], Ott incorporated the time derivative of shape in the equation, yielding better results but requiring a fixed-point approach to resolve the nonlinear coupling between the amplitude and shape equations. This also led to the technique's name: the Improved Quasi-Static method although one may argue that such a denomination does not make it immediately clear that the technique rigorously solves a time-dependent problem.

Nonlinear problems are solved in an iterative manner, typically using either a fixed-point (Picard) approach or Newton's iterations. Sissaoui et al. [4], Koclas et al. [7], Devooght et al. [2], and Monier [3] all use fixed-point iterative techniques for their IQS simulations, the main difference among them being their criteria for convergence. Devooght et al. in [2] proposed a Newton-based iteration technique, but this type of iteration in IQS is not investigated in this paper. Another approach is to linearize the equations, so that no nonlinear iteration is necessary. IQS can be linearized using the IQS Predictor-Corrector method (IQS-PC) [8]. IQS-PC entails evaluating the flux equation then correcting its amplitude using an amplitude evaluation. This method has proven to be effective for problems requiring a significant amount of shape updates when IQS is implemented [5]. Dulla also investigates shape time adaptation with IQS with Caron in [9].

For use in multiphysics reactor physics applications, the interplay of the IQS solution technique (neutronics) and other physics components (e.g., fuel temperature) needs to be resolved effectively. Meneley and Ott first implemented IQS into the one-dimensional fast-reactor code QX1 [10]. This implementation does simple adiabatic heat up with tabular cross-section feedback. Later, Keresztúri et al. describe IQS implementation into KIKO3D, a three-dimensional pressurized water reactor code, in [11]. This implementation computes fuel heat transfer and thermal hydraulic feedback. Ikeda and Takeda developed in the nodal expansion method code EPISODE which uses IQS-PC with adiabatic fuel heat and Doppler feedback, as well as full neutronics and thermal-hydraulics coupling [12]. These multiphysics applications range in complexity and problem type, but seem to neglect the advantage of including the the non-neutronics

physics into the quasi-static process. Although Meneley and Ikeda hint at using an intermediate time-scale for fuel temperature evaluation, their numerical experiments show that the evaluation is on the same scale as shape. Additionally, these applications only use heat source values at the shape time-scale from the neutronics calculation, while much more information about the transient heat source can be extracted from the amplitude scale.

This paper (i) discusses the different nonlinear iteration techniques for IQS and tests the rigor of their implementation, (ii) investigates high-order temporal discretization of the shape equation, and (iii) analyzes IQS coupled with adiabatic heat up and cross-section feedback. Prior IQS publications do not go beyond first-order time discretization for the shape equation. Here, we test IQS with higher-order schemes (up to fourth order) and investigate the method's effectiveness with such temporal discretizations. Step doubling time adaptation is also implemented to test IQS and IQS-PC performance with adaptation of shape/flux evaluation; although, Caron et al. in [9] performs a more comprehensive shape time adaptation analysis. Step doubling is also applied to amplitude evaluation. In regards to multiphysics simulation, we develop a semi-analytical approach to evaluate adiabtic fuel temperature and introduce an intermediate time scale for temperature evaluation. To test IQS and IQS-PC nonlinear iteration, time step convergence, and time adaptation performance, four different test cases are employed: two problems are purely neutronics and involve varying magnitudes of reactor size and complexity; two problems involve temperature feedback and test an intermediate time scale for temperature evaluation.

2. Background Theory on IQS

In this Section, we recall the equations for the IQS method, starting from multi-group neutron conservation statements in operator form:

$$\frac{1}{v^g}\frac{\partial \phi^g}{\partial t} = \sum_{g'=1}^{G} \left(H^{g'\to g} + P_p^{g'\to g} \right) \phi^{g'} - L^g \phi^g + S_d^g \,, \tag{1a}$$

$$\frac{dC_i}{dt} = \sum_{g=1}^{G} P_{d,i}^g \phi^g - \lambda_i C_i \;, \quad 1 \le i \le I \,. \tag{1b}$$

where $H^{g'\to g}$ is the scattering operator, $P_p^{g'\to g}$ is the prompt fission operator, L^g is the leakage and interaction operator, S_d^g is the delayed neutron source, and $P_{d,i}^g$ is the delayed-neutron fission operator. We assumed G neutron groups and I precursors groups. The operator L^g is block diagonal in energy groups: for neutron transport, $L^g = \mathbf{\Omega}\cdot\nabla + \Sigma_t^g$ whereas for neutron diffusion, $L^g = -\nabla\cdot D^g\nabla + \Sigma_r^g$, with Σ_t^g and Σ_r^g the total and removal macroscopic cross sections. In the following, we specialize the IQS derivation for the diffusion approximation without loss of generality. Obtaining the IQS formulation for neutron transport is similarly straightforward.

The flux factorization in IQS 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):

$$\phi^g(\boldsymbol{r},t) = p(t)\varphi^g(\boldsymbol{r},t)\,. \tag{2}$$

After reporting the above factorization in the balance equations, the shape diffusion equations result (the main differences are highlighted using boxes around modified/new terms):

$$\frac{1}{v^g}\frac{\partial\varphi^g}{\partial t} = \sum_{g'=1}^{G}\left(H^{g'\to g} + P_p^{g'\to g}\right)\varphi^{g'} - \left(L^g + \boxed{\frac{1}{v^g}\frac{1}{p}\frac{dp}{dt}}\right)\varphi^g + \boxed{\frac{1}{p}}S_d^g \tag{3a}$$

$$\frac{dC_i}{dt} = \boxed{p}\sum_{g=1}^{G}P_{d,i}^g\varphi^g - \lambda_i C_i\ , \quad 1\le i\le I \tag{3b}$$

Note that the time-dependent shape equations are similar to the time-dependent flux equations. One may introduce a new block diagonal operator $\tilde{L}^g = L^g + \frac{1}{v^g}\frac{1}{p}\frac{dp}{dt}$ where the reaction term (total/removal cross section) simply needs to be augmented by $\frac{1}{v^g}\frac{1}{p}\frac{dp}{dt}$. Note that the shape equations are now nonlinearly coupled (boxed terms) to the amplitude equations.

To obtain the amplitude equations, the multigroup shape equations are multiplied by a time-independent weighting function, typically the initial adjoint multigroup flux (ϕ^{*g}), and then integrated over the phase-space domain. For brevity, the inner product over space will be represented with parenthetical notation ($(\phi^{*g}, f^g) = \int_D \phi^{*g}(\boldsymbol{r})f^g(\boldsymbol{r})d^3r$). In order to impose uniqueness of the factorization, one requires that

$$K(t) = \sum_{g=1}^{G}\left(\phi^{*g}, \frac{1}{v^g}\varphi^g(t)\right) \tag{4}$$

be constant (hence $K(t) = K(t_0) = K_0$). After some manipulations, the point reactor kinetics equations (PRKE) for the amplitude solution are obtained:

$$\frac{dp}{dt} = \left[\frac{\rho-\bar{\beta}}{\Lambda}\right]p + \sum_{i=1}^{I}\bar{\lambda}_i\xi_i \tag{5a}$$

$$\frac{d\xi_i}{dt} = \frac{\bar{\beta}_i}{\Lambda}p - \bar{\lambda}_i\xi_i \quad 1\le i\le I \tag{5b}$$

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

$$\frac{\rho-\bar{\beta}}{\Lambda} = \frac{\sum_{g=1}^{G}\left(\phi^{*g}, \sum_{g'}(H^{g'\to g} + P_p^{g'\to g} - L^{g'}\delta_{g'g})\varphi^{g'}\right)}{\sum_{g=1}^{G}\left(\phi^{*g}, \frac{1}{v^g}\varphi^g\right)} \tag{6a}$$

$$\frac{\bar{\beta}}{\Lambda} = \sum_{i=1}^{I} \frac{\bar{\beta}_i}{\Lambda} = \sum_{i=1}^{I} \frac{\sum_{g=1}^{G} (\phi^{*g}, P_{d,i}^{g} \varphi^{g})}{\sum_{g=1}^{G} \left(\phi^{*g}, \frac{1}{v^{g}} \varphi^{g}\right)} \tag{6b}$$

$$\bar{\lambda}_i = \frac{\sum_{g=1}^{G} (\phi^{*g}, \chi_{d,i}^{g} \lambda_i C_i)}{\sum_{g=1}^{G} (\phi^{*g}, \chi_{d,i}^{g} C_i)} \tag{6c}$$

Solving for the shape in Eq. (3a) can become expensive, especially in two or three spatial dimensions, and even more so when using the transport equations in lieu of the diffusion equations. Using IQS, one expects the time dependence of the shape to be weaker than that of the flux itself, thus allowing for larger time step sizes in updating the shape. The PRKE equations form a small ODE system and can be solved using a much smaller time step size. In transients where the shape varies much less than the flux, IQS can be very computationally effective. Note that the PRKE parameters are evaluated at each macro step and interpolated for the PRKE evaluation. In order to preserve the error convergence rate of high order temporal discretization schemes for shape, higher order interpolation of the parameters will be required. Third order Runge-Kutta discretization with step doubling adaptation was used for the PRKE evaluation, to insure a negligibly small error in the amplitude for a given set of parameters.

2.1. IQS Iterative Schemes

As noted in the previous section, shape-PRKE equations form a nonlinear system and must be solved in a iterative manner. Over each macro time step, one can use the latest end-time shape iterate to compute/interpolate the PRKE coefficents over the micro time step intervals. Sissaoui et al. from [4], Koclas et al. from [7], Devooght et al. from [2], and Monier from [3] all use iterative techniques for their IQS implementations. They all undergo a similar process:

Step 1: Compute the PRKE parameters at the end of the macro step using the last computed shape

Step 2: Linearly interpolate the computed PRKE parameters over the macro step

Step 3: Solve the PRKE on micro steps over the entire macro step

Step 4: Solve the shape equation on the macro step using the computed values of p and dp/dt.

Step 5: Check if the shape solution has converged:

- *No:* Repeat the same macro time step
- *Yes:* Move on to the next macro time step

The major difference between the methods of these authors is the convergence criteria used. Sissaoui and Koclas [4, 7] use fixed point iteration where the criteria is the simply the normalized difference between the last two computed shapes. Monier in [3] also employs fixed point iterations with the same criteria, except that the solution is rescaled by K_n/K_{n+1} after each iteration. Dulla in [5] does the same fixed-point iteration, but the shape is scaled by K_0/K_{n+1} at

the end of each macro time step. Renormalizing the shape is essential for preserving the uniqueness condition ($K_{n+1} = K_0$); this condition is not inherently conserved, even when the solution has converged from shape iterations.

These techniques are by no means an exhaustive list of the possible iteration techniques for IQS. Dulla et al. in [5] provide an in-depth analysis of the fixed point iteration technique most similar to Sissaoui and Koclas, involving convergence rates and solution results. However, no comprehensive analysis of iteration techniques exists. The following describes each iteration convergence criterion investigated in this paper:

- L^∞ norm of shape [3]:

$$\frac{\max\left|\varphi_{n+1}^{(k+1)} - \varphi_{n+1}^{(k)}\right|}{\max\left|\varphi_{n+1}^{(k+1)}\right|} < \epsilon_\varphi$$

- L^2 norm of shape:

$$\frac{\left\|\varphi_{n+1}^{(k+1)} - \varphi_{n+1}^{(k)}\right\|_{L^2}}{\left\|\varphi_{n+1}^{(k+1)}\right\|_{L^2}} < \epsilon_\varphi$$

- Reactivity convergence [3]:

$$(\rho/\Lambda)_{n+1}^{(k+1)} - (\rho/\Lambda)_{n+1}^{(k)} < \epsilon_\rho$$

- Amplitude convergence [3]:

$$p_{n+1}^{(k+1)} - p_{n+1}^{(k)} < \epsilon_p$$

- Uniqueness consistency [3]:

$$\frac{K_{n+1}^{(k+1)} - K_0}{K_0} < \epsilon_K$$

  where k denotes the nonlinear iteration index within a given macro time step interval.

In order to ensure that uniqueness criteria is preserved, some authors [3, 5] explicitly scale the shape solution, either at each nonlinear iteration or upon exiting the nonlinear loop on the macro time time step. This re-scaling is simply

$$\varphi_n^g \leftarrow \varphi_n^{g,(\text{last})} \frac{K_0}{K_n^{(\text{last})}} \,. \tag{7}$$

2.2. IQS Predictor-Corrector Scheme

The Predictor-Corrector version of the IQS method (IQS-PC) is a linearized version of IQS, where no nonlinear iterations are required. The method relies on the shape and amplitude factorization of the flux and uniqueness condition. The PRKE derivation is identical to that of the standard (nonlinear) version of the IQS scheme, in the sense that shape solutions at the beginning and end times of the macro time interval are used (with interpolation in between). However, the manner in which the shape function is obtained is different. In the IQS-PC version, the flux equations (not the shape equations) are first solved (represented by Eqs. (1a) and (1b)) in order to obtain a *predicted* flux solution at the end of the macro time step. This predicted flux is then converted to a shape by normalizing it as follows:

$$\varphi_{n+1}^g = \underbrace{\phi_{n+1}^g}_{\text{predicted}} \frac{K_0}{\mathcal{K}_{n+1}}, \tag{8}$$

where the flux scaling factor is given by

$$\mathcal{K}_{n+1} = \sum_{g=1}^{G} \left(\phi^{*g}, \frac{1}{v^g} \phi_{n+1}^g \right), \tag{9}$$

and K_0 is the shape normalization constant (see Eq. (4)).

The PRKE parameters are then computed with this shape using Eqs. (6a)-(6c) and interpolated over the macro step, then the PRKE ODE system is solved on the micro time scale. With the newly computed amplitude, the shape is rescaled into a flux and the final *corrected* flux is given by:

$$\underbrace{\phi_{n+1}^g}_{\text{corrected}} = p_{n+1} \times \varphi_{n+1}^g . \tag{10}$$

The advantage to the predictor-corrector method is that no nonlinear iterations are necessary in this method; it is simpler to implement and can be faster than the standard IQS. Ikeda et al. in [12] and Goluoglu et al. in [13] both use IQS-PC for complex, three-dimensional problems. Their results prove IQS-PC to be effective. Dulla et al. in [5] also describes an in depth comparison of IQS-PC with traditional IQS, which originally brought the method to light.

2.3. Temperature Feedback Treatment

Here, we consider multiphysics reactor physics simulations. Fuel temperature feedback mechanisms are included using an adiabatic heat conservation model given in Eq. (11).

$$\rho c_p \frac{\partial T(\boldsymbol{r}, t)}{\partial t} = \kappa_f \sum_{g=1}^{G} \Sigma_f^g \phi^g(\boldsymbol{r}, t) \tag{11}$$

The thermal-range cross section's dependence on temperature is described by Eq. (12).

$$\Sigma_a^{thermal}(\boldsymbol{r},t) = \Sigma_a^{thermal}(\boldsymbol{r},0)\left[1+\gamma\left(\sqrt{T}-\sqrt{T_0}\right)\right] \tag{12}$$

A standard time-implicit solver for the heat equation would simply employ the flux values at the extremities of a time step interval. However, in an IQS neutronic solve, much more information about space/time distribution of the flux is known: the shape distributions are typically known at the beginning and end of a macro time step interval but the amplitude is known on the micro-step time scale, providing a richer space/time information for the neutron flux over the macro-step interval. Thus, it is possible to solve for temperature in Eq. (11) using a semi-analytical approach, shown in Eq. (13).

$$T_{n+1} = T_n + \frac{\kappa_f}{\rho c_p}\left(a_2\varphi_{n+1} + a_1\varphi_n\right) \tag{13}$$

where n corresponds to the beginning of the temperature step. a_1 and a_2 are integration coefficients given by Eq. (15a) and Eq. (15b). The shape function was assumed to vary linearly in time:

$$\varphi(\boldsymbol{r},t) = \frac{t_{n+1}-t}{\Delta t}\varphi_n(\boldsymbol{r}) + \frac{t-t_n}{\Delta t}\varphi_{n+1}(\boldsymbol{r})\,, \tag{14}$$

which leads to the coefficients' definitions:

$$a_1 = \int_{t_n}^{t_{n+1}} \left(\frac{t_{n+1}-t'}{\Delta t}\right) p(t')dt' \tag{15a}$$

$$a_2 = \int_{t_n}^{t_{n+1}} \left(\frac{t'-t_n}{\Delta t}\right) p(t')dt' \tag{15b}$$

Because the amplitude p is known on a fine time scale, the integrals Eq. (15a) and Eq. (15b) are carried out along the micro steps, using a linear interpolant for the amplitude.

Temperature feedback affects both the shape equation and the reactivity coefficients of the PRKE; thus, it is an additional nonlinear component to the already coupled shape-amplitude equations. In foresight to the application of this component, we expect temperature to be more rapidly varying than the shape, but less so than the amplitude. Therefore, the evaluation of temperature will have its own time scale, intermediary between the amplitude's fine time scale and the shape's coarse time scale. Hence, a possible solution process for a problem with temperature feedback will have three time scales, as portrayed in Fig. 1. The first time scale is the shape solve, the second is the temperature evaluation as well as the computation of PRKE parameters, and the third is the PRKE scale. It is important to note that the number of time steps in each scale is arbitrary and the number chosen in Fig. 1 are only meant for illustrative purposes.

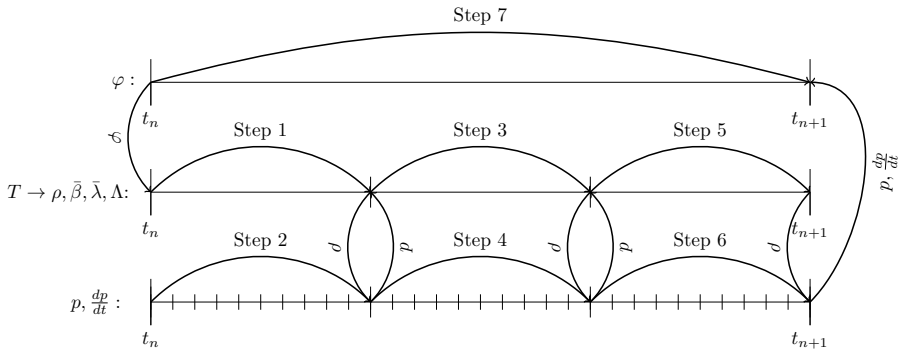


Figure 1: Time scales and process of IQS solve with temperature feedback

Iteration processes are needed in between each time scale. The amplitude and temperature need to be iterated on the middle time scale until convergence on each temperature step. Then another iterative process needs to occur in the shape time scale on all three variables. Fig. 2 shows the programming diagram implement to execute this process. The time increment of $\Delta t/3$ for the temperature solve is arbitrary and is meant only to match Fig. 1 where three temperature updates have been used as illustration.

2.4. Delayed Neutron Precursor Treatment

The precursors' equations, Eq. (3b), form a system of ODES. A theta-scheme is often employed to evaluate the precursor concentrations:

$$C_{n+1} = \frac{1-(1-\theta)\lambda\Delta t}{1+\theta\lambda\Delta t}C_n + \frac{(1-\theta)\beta\Delta t}{1+\theta\lambda\Delta t}S_{f,n}p_n + \frac{\theta\beta\Delta t}{1+\theta\lambda\Delta t}S_{f,n+1}p_{n+1} \quad (16)$$

where S_f is the fission source computed using the shape solution ($S_{f,n} = (\nu\Sigma_f)_n\varphi_n$). With $\theta = 1$, this yields the implicit Euler method, while with $\theta = \frac{1}{2}$, we obtain the Crank-Nicolson technique. In doing so, however, no micro-scale temporal information for the amplitude is used. One can also solve the precursors' equations analytically, yielding:

$$C_{n+1} = C_n e^{-\lambda(t_{n+1}-t_n)} + \int_{t_n}^{t_{n+1}} \beta(t')S_f(t')p(t')e^{-\lambda(t_{n+1}-t')}dt' \quad (17)$$

The shape function (using in S_f) is not known continuously over the time step but one may use the beginning/end times values to linearly interpolate the fission source over the macro time step,

$$S_f(t) = \frac{t_{n+1}-t}{\Delta t}S_{f,n} + \frac{t-t_n}{\Delta t}S_{f,n+1} \quad t_n \le t \le t_{n+1}\,. \quad (18)$$

Furthermore, a very accurate representation of $p(t)$ over the macro step is available from the PRKE solve (micro time scale). Using this fine scale information,

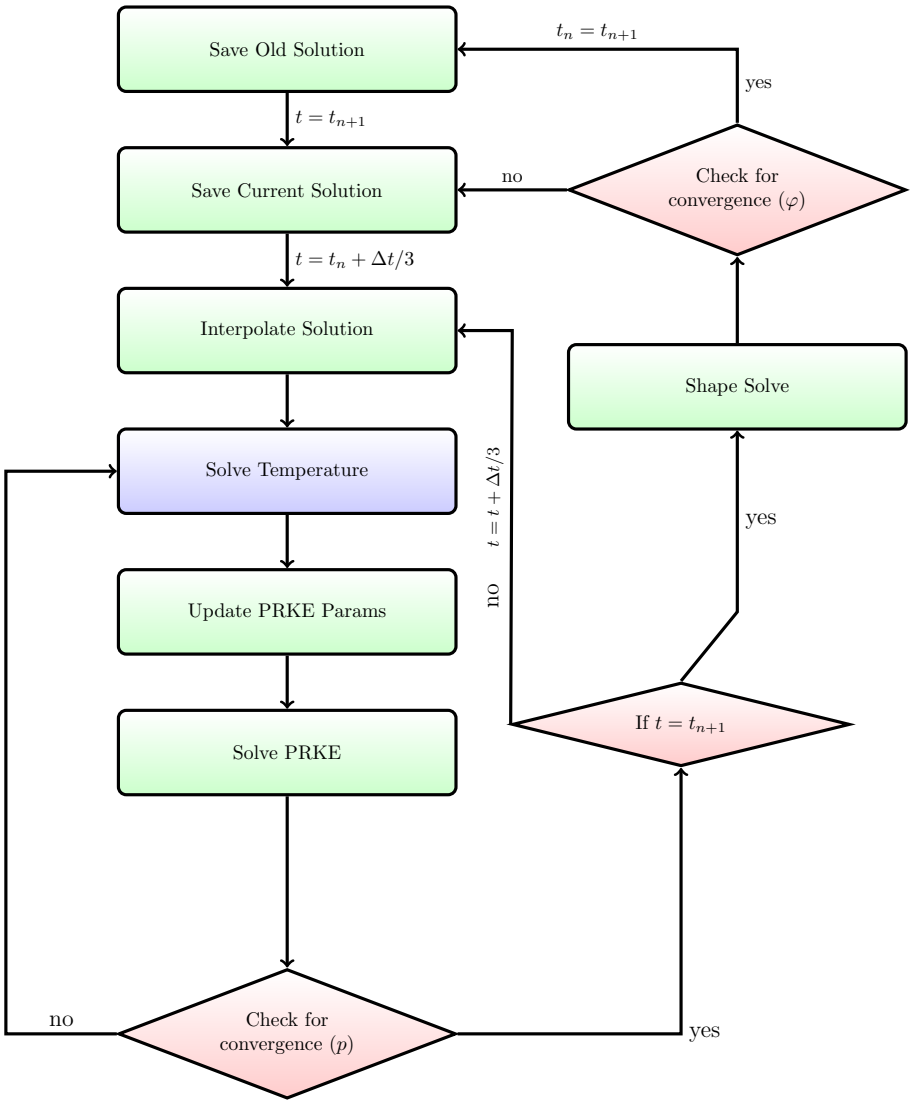


Figure 2: Visualization of fixed-point iteration and temperature update process for IQS

the analytical solve for the precursor values yields

$$C_{n+1} = C_n e^{-\lambda \Delta t} + \left(\hat{a}_2 S_{f,n+1} + \hat{a}_1 S_{f,n}\right) \beta \,, \tag{19}$$

with integration coefficients defined as:

$$\hat{a}_1 = \int_{t_n}^{t_{n+1}} \frac{t_{n+1} - t'}{\Delta t} p(t') e^{-\lambda (t_{n+1} - t')} dt' \,, \tag{20a}$$

and

$$\hat{a}_2 = \int_{t_n}^{t_{n+1}} \frac{t' - t_n}{\Delta t} p(t') e^{-\lambda (t_{n+1} - t')} dt' \,. \tag{20b}$$

2.5. Step Doubling Time Adaptation

Further enhancements to the performance of the IQS methods can be gained by using time adaptation (or time step control) in order to increase or reduce the macro time step size for the shape evaluation, depending on error estimates. A step-doubling technique is chosen as the time adaptation technique [14]. The step doubling technique involves estimating the local error for a certain time step by taking the difference between a solution with one full step ($\varphi^g_{\Delta t}$) and a solution with two half steps ($\varphi^g_{\Delta t/2}$). Note: φ is changed to ϕ in the case of the IQS-PC tehcnique. The relative error is computed as follows:

$$e_n = \frac{\left\| \sum_{g=1}^{G} \varphi^g_{\Delta t/2} - \sum_{g=1}^{G} \varphi^g_{\Delta t} \right\|_{L^2}}{\max\left(\left\| \sum_{g=1}^{G} \varphi^g_{\Delta t/2} \right\|_{L^2}, \left\| \sum_{g=1}^{G} \varphi^g_{\Delta t} \right\|_{L^2} \right)} \,. \tag{21}$$

If the error is smaller than the user-specified tolerance, e_{tol}, the time step is accepted. In addition, a new time step size is estimated as follows:

$$\Delta t_{new} = s \Delta t \left(\frac{e_{tol}}{e_n} \right)^{\frac{1}{1+q}} \,, \tag{22}$$

where q is the convergence order of the time integration scheme being used and $s \simeq 0.8$ is a safety factor. If the error is larger than the user-specified tolerance, the time step is rejected and new (smaller) macro time step size is estimated using Eq. (21) as well. This process can be visualized by Figs. 3 and 4, where a step involves a full convergence of shape, amplitude, and any multiphysics on the respective time step.

To investigate IQS's performance with step-doubling time adaptation, the adaptation will be applied to direct temporal discrezation of the flux equation (no IQS), the standard IQS method, and the IQS-PC version. Each of these methods will be applied to several diffusion problems; the number of time steps taken and the resulting error will be used to compare the methods.

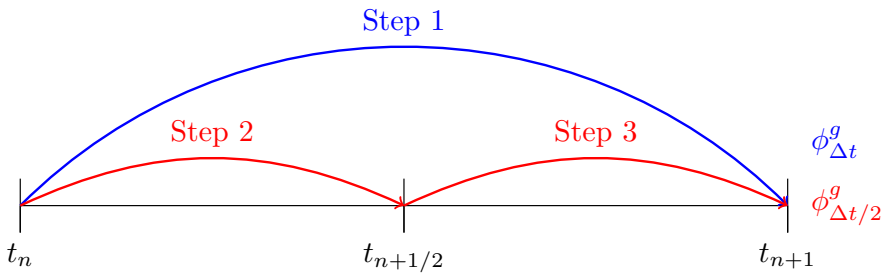


Figure 3: Visualization of step doubling process on time-line

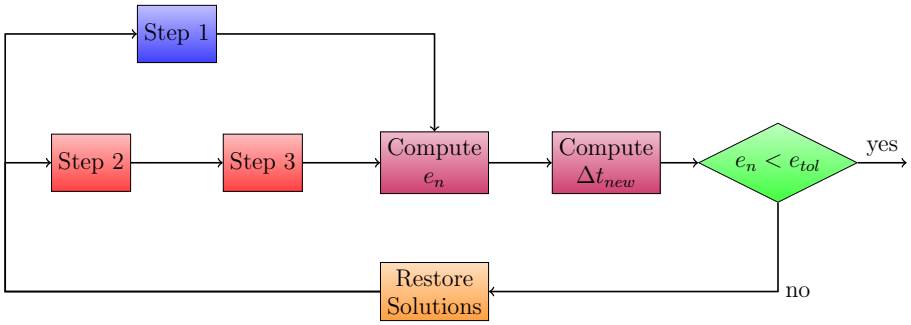


Figure 4: Visualization of step doubling process with coding logic

3. Implementation in a multiphysics simulation environment

Both the standard IQS and IQS-PC methods have been implemented in the Multiphysics Object-Oriented Simulation Envrionement, MOOSE [15], as part of its radiation transport solver, Rattlesnake [16]. All multi-dimensional results presented here have been computed using the diffusion solver of Rattlesnake. MOOSE calculation flows naturally, which allows for a relatively straightforward implementation of the IQS methodologies. MOOSE's nonlinear solvers (fixed-point iterations and Jacobian-free Newton-Krylov) are well suited to handle the nonlinear solve of the traditional IQS method during a macro time step. IQS-PC's implementation is MOOSE is even simpler. We briefly describe the their implementation next.

First, the IQS shape equation was represented by the gathering of appropriate kernels. Most of the shape equation kernels are the same as the flux kernels already implemented in Rattlesnake, so only two kernels were modified. Eq. (23) shows which kernels needed to be written to make sure MOOSE is solving the shape equation. The PRKE ODE was written as a sub-function in the Transient executioner, which progresses the system in time and tells when to evaluate the main shape/flux solve. This executioner has breakpoints that allows specified calls to the PRKE, PRKE parameter evaluation, and other multiphysics. The PRKE parameters are written as user-objects that loop over elements to

execute the proper integration when the executioner calls them. Similarly, the semi-analytic treatment of fuel temperature and precursors are auxiliary kernels that do a node-by-node evaluation of the variable when the executioner calls it. For a more detailed description of the MOOSE application of IQS, see [17].

$$\frac{1}{v^g}\frac{\partial \varphi^g}{\partial t} = \underbrace{\frac{\chi_p^g}{k_{eff}}\sum_{g'=1}^{G}(1-\beta)\nu^{g'}\Sigma_f^{g'}\varphi^{g'}}_{FluxKernel} + \underbrace{\sum_{g'\neq g}^{G}\Sigma_s^{g'\to g}\varphi^{g'}}_{FluxKernel} - \underbrace{\left(-\nabla\cdot D^g\nabla\right)\varphi^g}_{FluxKernel} - \underbrace{\Sigma_r^g\varphi^g}_{FluxKernel}$$
$$\underbrace{-\frac{1}{v^g}\boxed{\overbrace{\frac{1}{p}\frac{dp}{dt}}^{FromPRKE}}\varphi^g}_{IQSKernel} + \underbrace{\frac{1}{p}\sum_{i=1}^{I}\chi_{d,i}^g\lambda_i C_i}_{ModifiedFluxKernel} \tag{23}$$

4. Neutronics-only Transient Results

In this section, we analyze various convergence criteria proposed in the non-linear solution technique for the IQDS method. We also investigate the use of higher-temporal discretization for the shape equations. Both series of tests are carried out using neutronics-only problems. The first test case uses a one-dimensional problem. The second test is from the ANL Benchmark Problem Book (BPB) [18].

4.1. One-Dimensional Problem

This one-dimensional problem uses a 400-cm slab. This initial configuration is homogeneous and the transient is initiated by a perturbation in absorption cross section. Fig. 5 shows problem layout and Table 1 contains the material properties; one-group values are used for this problem. Regions 2, 3, and 4 have linear ramp perturbations at different moments in time, Table 2 shows the values of the absorption cross-section in each region at the times of interest. The values of Σ_a between these times of interest vary linearly between the given values.

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

Figure 5: 1-D slab: Region identification number

Fig. 6 shows the reference flux solution at different instants as well as total power as a function of time. The baseline reference solution was computed by solving the flux equations (not the IQS formulation) using time-step control

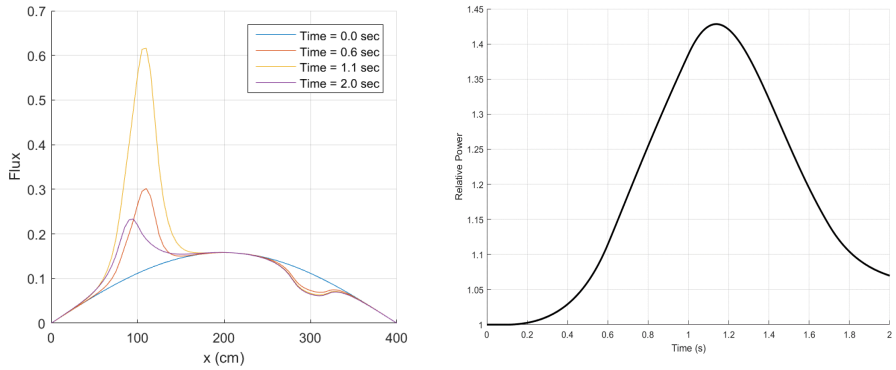
Table 1: 1-D slab material properties and problem parameters

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

Table 2: 1-D slab absorption cross-section at times of interest

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

with a tight relative error tolerance of 10^{-12}. The problem being relatively large, the various zones are weakly coupled and the flux perturbation is not global. This baseline computation is used to compute the error of the other time discretization methods. Fig. 7 shows the shape profile at the same instants during the transient as in Fig. 6a; that solution was also obtained using time-step control with tight tolerance. Both flux solve and IQS solve yield the same answer, as expected.



(a) Flux profile at various times

(b) Power profile over transient

Figure 6: Baseline flux and power distribution

In the standard IQS technique, a nonlinear system of equations is to be solved for the shape and the amplitude solutions. Section 2.1 lists various iteration stopping criteria typically employed with IQS. Fig. 8 gives the number of fixed-point iterations required to reach tolerance of 10^{-11} as a function of time during the transient. A theta-scheme is employed for the precursors equations (Eq. (16)). This figures shows that criteria based on the L^∞ norm of the shape, the L^2 norm of the shape, the reactivity change, and the amplitude

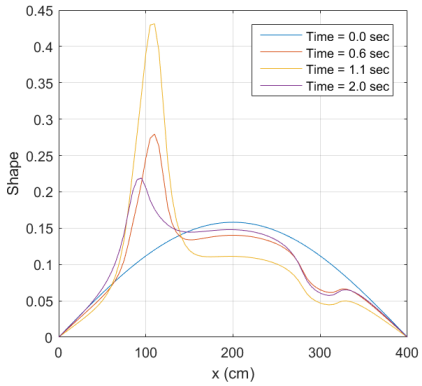


Figure 7: IQS Shape profile at various times

have approximately the same convergence behavior, but the criteria based on the uniqueness constant K never reaches the tight tolerance and runs up to the maximum number of iterations allowed (20 in this case). Fig. 9a shows the error in the factor K saturates to different levels, whether the shape is rescaled or not. Rescaling (described by Eq. (7)) However, switching the precursors solve to be performed semi-analytically (Eq. (19)) significantly improves the convergence in the uniqueness factor K, see Fig. 9b.

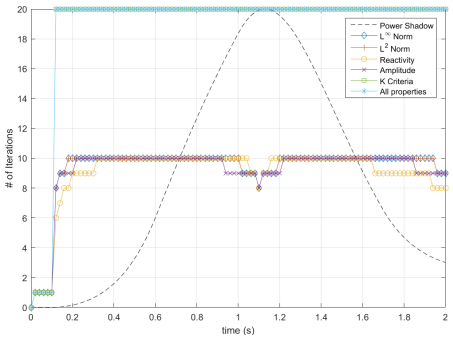
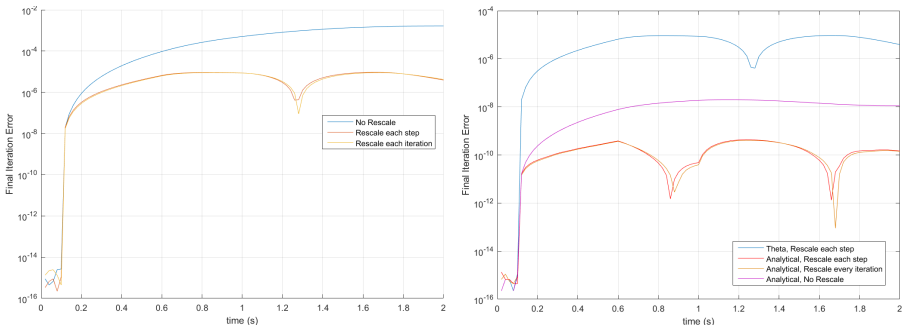


Figure 8: # of iterations for various convergence criteria, tolerance= 10^{-11}, max iterations= 20



(a) Precursors solved using a theta-scheme (b) Analytical precursor solve

Figure 9: Final iteration error for K factor convergence criterion

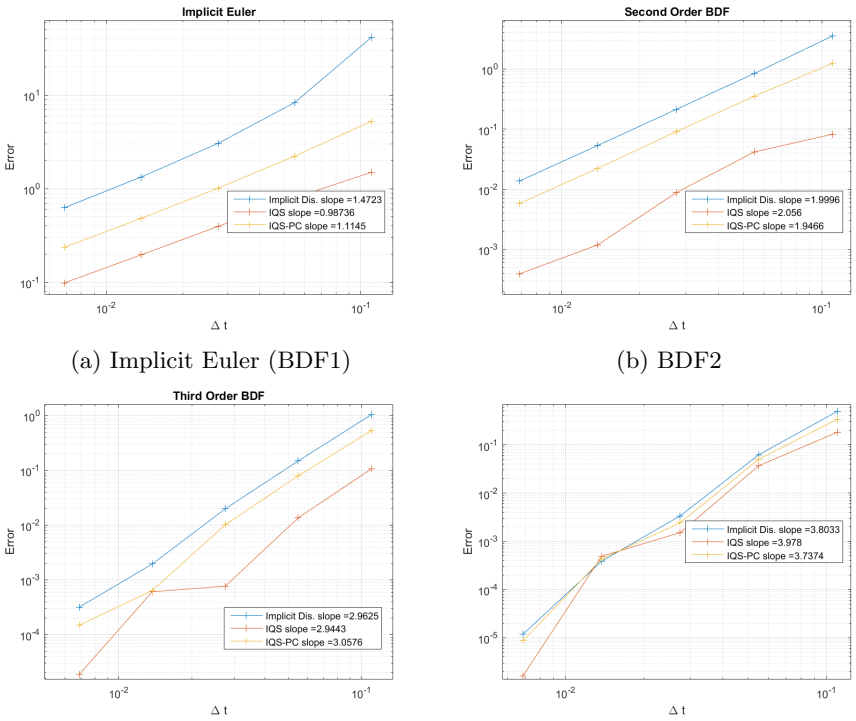
Next, we investigate the use of higher-order temporal methods to solve the shape equations (standard IQS and IQS-PC methods). We compare these results with the same high-order discretization for the flux equations. Fig. 10 shows the convergence results for four different time-implicit discretization: backward difference formulae, from order 1, which is the same as the implicit Euler, to order 4. For a detailed description on BDF methods, see [19]. The plots show that IQS and IQS-PC converge with the expected order (1 through 4) using BDF1 through BDF4. Recall that in the amplitude equations, the PRKE parameters are interpolated using the shape solutions. We used higher-order interpolants in time for the shape solution when a higher-order method was used (the convergence rates were degraded for higher-order BDF methods when the PRKE parameters were only linearly interpolated over the macro step). Lagrange and Hermite interpolants yielded the same results. In addition, we used semi-analytical integration of the precursors equations, along with the same higher-order interpolation of the shape values when a higher-order method was used. Without using the same order interpolants as the BDF formula, the error in the PRKE parameters and the precursors led to reduction in the observed convergence rate.

The convergence rates for the straightforward discretization of the flux equations are also shown in Fig. 10 and they followed the same expected convergence rates. These flux-solve results are indicated by the legend "Implicit Dis.". It is worthy to note that, for lower-order discretizations, the error in the flux equations is always significanrtly higher than that of the IQS results. However, as the order of the temporal scheme is increased, the gap between the discretization error in the flux equations and the IQS equations vanishes. In previous publications, the shape solve in the IQS method always employed a lower-order temporal discretization (often implicit Euler). Our results seem to indicate that a high-order discretization of the flux equations may reduce the usefulness of the IQS approach (at least for large domains with weakly coupled spatial regions).

4.2. TWIGL Benchmark

This benchmark problem originates from the ANL Benchmark Problem Book [18]. It is a 2D, 2-group reactor core model with no reflector region [20]. The core is smaller (hence the spatial regions are more tightly coupled) and the spatial variation of the flux varies moderately over the transient. Therefore, IQS is expected to perform significantly better than an implicit flux solve. Fig. 11 shows the IQS solution as compared with the implicit flux solution. This illustrates that IQS can yield a much more accurate solution, even at a significantly larger time step than the implicit flux discretization.

In order to demonstrate asymptotic convergence of IQS, implicit Euler (IE) and second order BDF (BDF2) were applied to solve the TWIGL problem. Fig. 12 plots the error convergence of IQS and the implicit discretization methods. The curves show the superior convergence of IQS. The observed convergence rates are indicated in the legend. They show orders 1 for IE and 2 for BDF2 for the flux solves. However, the convergence rates for IQS show higher values, order 2 for IE solves for the shape, and above 2.5 for BDF2 solve of



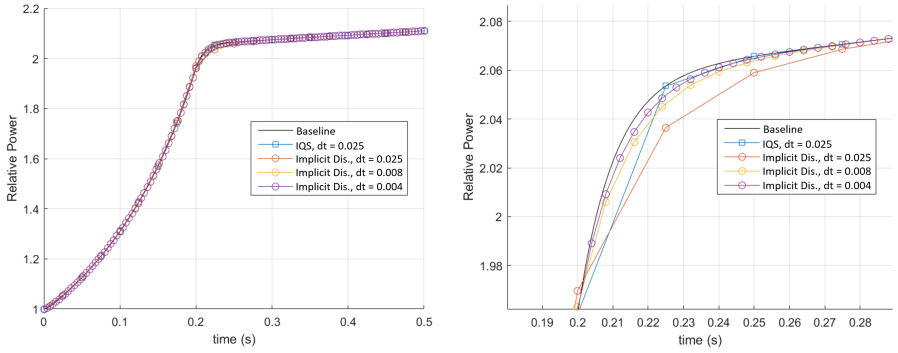
(a) Implicit Euler (BDF1)

(b) BDF2

(c) BDF3

(d) BDF4

Figure 10: Error convergence plots of implicit discretization, IQS, and IQS-PC with various time discretization schemes



(a) Power profile for entire transient

(b) Power at cusp of profile

Figure 11: Power level comparison of TWIGL Benchmark

the shape. When there is moderate variation in the shape over the transient, the PRKE parameters are accurately estimated, and the fine-scale PRKE solve provides a well resolved amplitude solution.

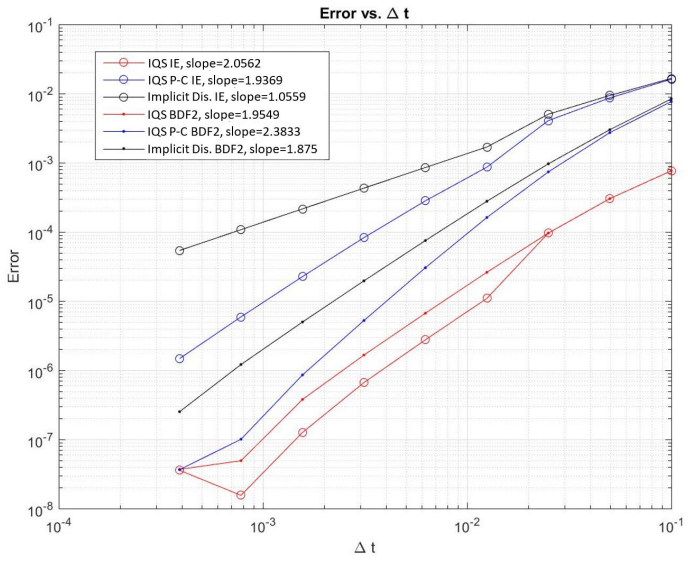


Figure 12: Error convergence comparison of TWIGL Benchmark

Table 3 and Fig. 13 show the results for the TWIGL benchmark with time adaptation. The results show that both IQS methods perform exceptionally well compared to implicit flux discretization. It also shows that traditional IQS performed better with large e_{tol}, while IQS-PC was better with smaller e_{tol}.

Table 3: TWIGL step doubling results

		Implicit Discretization			IQS			IQS-PC		
Test	e_{tol}	Error	Steps	Solves	Error	Steps	Solves	Error	Steps	Solves
1	0.05	0.00012677	9	29	0.03380433	4	20	0.00323100	4	9
2	0.01	3.5555e-05	11	35	0.00166991	5	40	0.00263068	5	12
3	0.005	4.0364e-05	11	31	0.00886584	5	40	0.00160486	6	21
4	0.001	0.00294822	33	122	0.02976305	5	36	1.7527e-05	10	35
5	0.0005	0.00099778	39	131	0.00143781	6	55	1.4185e-05	16	74
6	0.0001	0.00019510	78	236	0.00016175	8	65	6.2903e-06	19	78
7	5.0e-05	0.00018372	112	342	6.0328e-05	12	163	1.5247e-06	24	92

5. Multiphysics Results

This section describes two multiphysics examples with adiabatic heat up: the LRA benchmark from the ANL Benchmark Problem Book [18] and a modelisation of an experiment carried out at the Transient Reactor Test Facility, TREAT [21, 22].

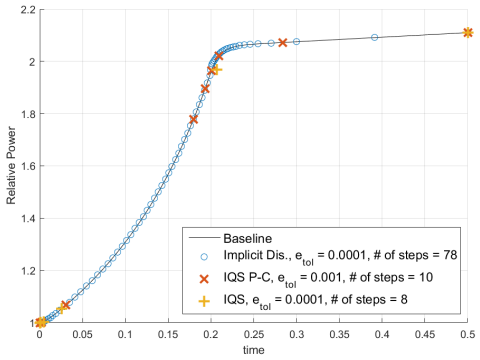


Figure 13: Power level comparison of TWIGL Benchmark with time step control

5.1. LRA Benchmark

The LRA benchmark is a two-dimensional, two-group neutron diffusion problem with adiabatic heat-up and Doppler feedback in thermal reactor. It is a super prompt-critical transient. The execution of the benchmark was performed by the Rattlesnake/MOOSE framework at Idaho National Laboratory (INL) [16]. The spatial discretization uses continuous finite elements with first-order Lagrangian basis functions. The mesh consists of blocks 11×11 with five uniform refinements, totaling $165,165$ elements and $124,609$ nodes. Three different temporal techniques are applied: implicit discretization of the flux equation, IQS, and IQS-PC. Crank-Nicolson time discretization scheme is used for the diffusion evaluation of each technique. The performance of IQS and the temperature updates were measured by its improvement in accuracy at peak power over the implicit discretization method.

Fig. 14 shows the baseline power and temperature transient profile for the LRA benchmark. The baseline results are compared to the results achieved by Sutton and Aviles in [23] and presented in Table 4. The relative difference in the magnitude of the peak power ($t \approx 1.44s$) from the baseline was used for error comparison.

Table 4: LRA baseline verification

Calculation	Baseline Rattlesnake	Sutton (Spandex 1936)
No. of Spatial Nodes	3872	1936
Eigenvalue	0.99637	0.99637
No. of Time Steps	6000	23,890
Time to Peak Power (s)	1.441	1.441
Peak Power (W/cm^3)	5456	5461

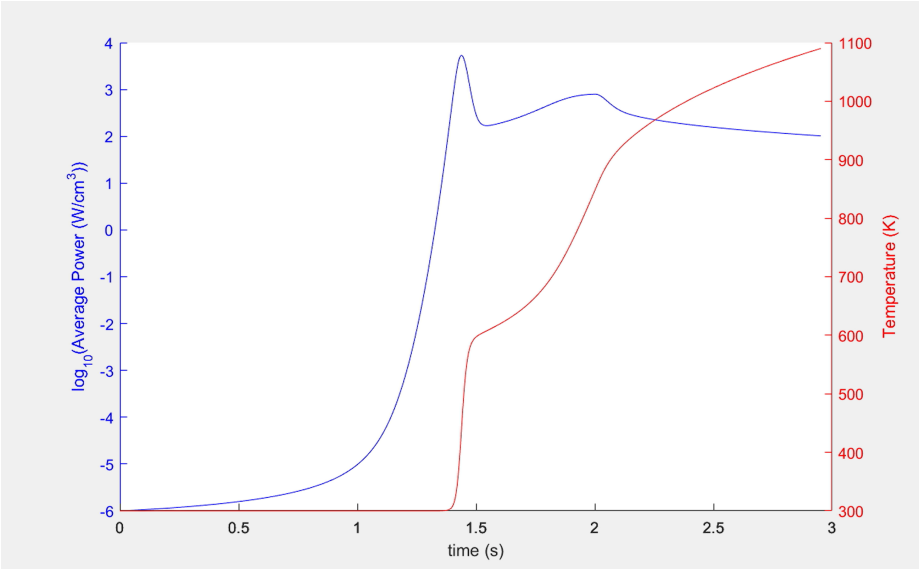
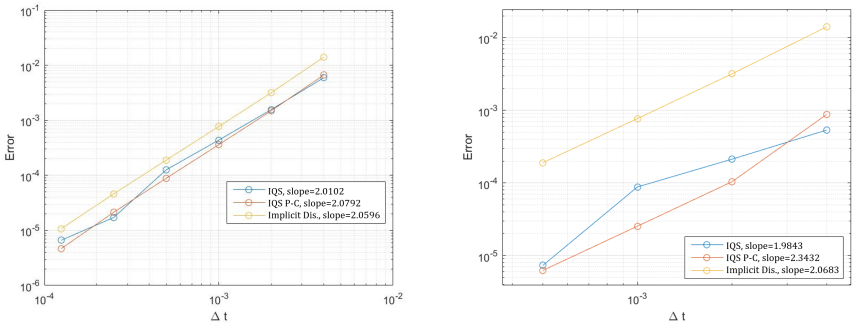


Figure 14: LRA baseline temperature and power profile

5.1.1. LRA Results

This section shows the time step error convergence of IQS for the LRA benchmark, as well as the effect of the intermediate temperature time scale. Fig. 15a is an error convergence plot comparing the three techniques where temperature is evaluated only on the macro step (1 temperature update). Fig. 15b is an error convergence plot comparing the three techniques where temperature is evaluated 5 times within a macro step (5 temperature updates). Finally, Fig. 16 shows the effect of various temperature updates. The dashed lines correspond to implicit discretization at different flux step sizes, while the IQS macro step size is kept constant.

The convergence plots show that updating temperature and the PRKE parameters within a macro step has a significant effect on the performance of IQS. With only one update, IQS was only slightly better than implicit discretization, implicit discretization required about 150% more time steps than IQS for the same error. While 5 temperature updates showed a much more significant IQS performance, implicit discretization required about 400% more time steps than IQS for the same error. Fig. 16 shows that error has a convergent behavior for the number of temperature updates. This convergence makes sense because temperature can only be so accurate before the error in shape is dominating. Table 5 shows the run time results for the implicit discretization calculations. The number of GMRES linear iterations is included because it is proportional measure of the computational effort. Tables 6 and 7 present the IQS run-times with various numbers of temperature updates. These run-times are based on total alive time of the execution where the diffusion evaluation is distributed



(a) Only one temperature update per macro step

(b) Five temperature updates per macro step

Figure 15: LRA error convergence plots

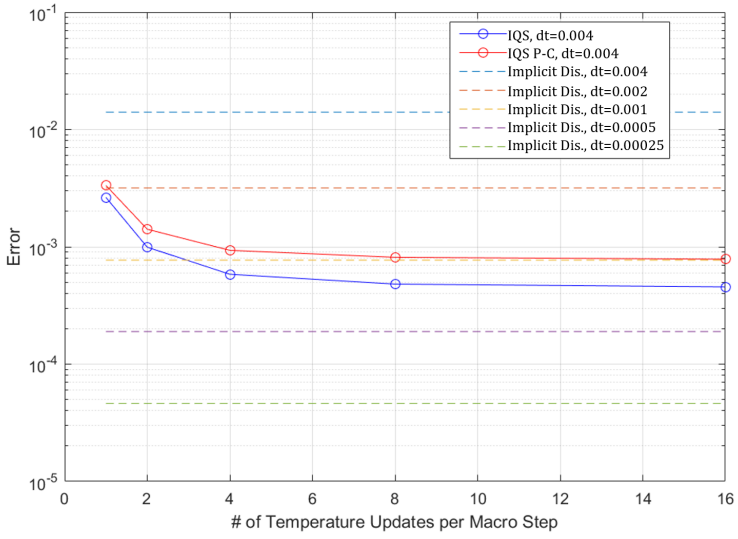


Figure 16: Error plot with various temperature updates per macro step

over 24 processors. These run-times show a marginal performance for IQS and impressive performance for IQS-PC. Some of the execution times were able to decrease from implicit discretization with the same number of macro steps because IQS is better equipped to resolve the nonlinearity between temperature and amplitude. Furthermore, there does seem to be an ideal number of temperature updates to optimize execution time: IQS only needs one and IQS-PC seems to be ideal at 4 updates. This discrepancy in the number of updates shows that a adaptive type implementation of the updates would be ideal, and could enforce a constant error over the transient. It is also important to compare the error of implicit discretization with IQS at one update and IQS-PC at 4 updates. IQS shows an error comparable to implicit discretization at $\Delta t = 0.002$, signifying an actual increase in runtime by -34.1%. IQS-PC shows an error less than implicit discretization at $\Delta t = 0.002$, signifying an actual increase in runtime by $< -34.9\%$.

Table 5: Implicit discretization run time results

Run	Δt	Error	Runtime (hr)	Linear Iter.
1	4.0e-3	1.407e-2	4.11	7.13e4
2	2.0e-3	3.174e-3	6.01	9.49e4
3	1.0e-3	7.690e-4	10.38	1.45e5
4	5.0e-4	1.892e-4	21.91	2.08e5
5	2.5e-4	4.590e-5	25.23	3.16e5

Table 6: IQS run time results with $\Delta t = 0.004$

Run	Temperature Updates	Error	Runtime (hr)	% Increase in Runtime*
1	1	2.612e-3	3.96	-3.18%
2	2	9.893e-4	6.02	47.1%
3	4	5.796e-4	7.87	92.3%
4	8	4.772e-4	12.61	207.9%
5	16	4.516e-4	22.14	440.7%

* difference in runtime from $\Delta t = 0.004$ implicit discretization

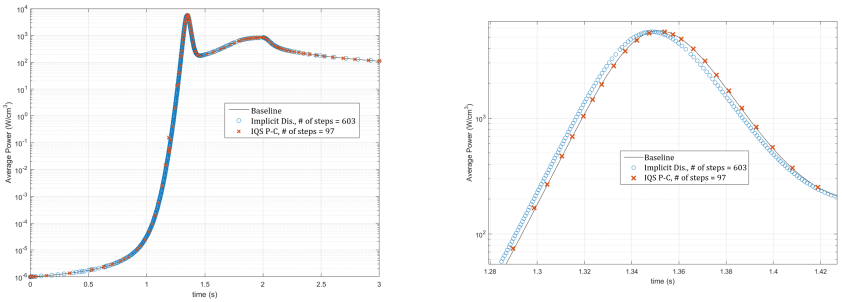
5.1.2. LRA Time Adaptation

Fig. 17 shows the power profile of the LRA with time adaptation of implicit discretization and IQS-PC, and Table 8 compiles the results. These time adaptation results show the significant decrease in macro time steps required for IQS-PC. These profiles were obtaining with only one temperature update per macro step; so based on previous results, the IQS-PC performance would improve even more with more updates.

Table 7: IQS PC run time results with $\Delta t = 0.004$

Run	Temperature Updates	Error	Runtime (hr)	% Increase in Runtime*
1	1	3.488e-3	2.91	-28.9%
2	2	1.349e-3	3.73	-9.00%
3	4	9.161e-4	3.97	-3.04%
4	8	8.052e-4	5.39	31.7%
5	16	7.905e-4	8.19	100%

* difference in runtime from $\Delta t = 0.004$ implicit discretization



(a) Full power profile

(b) Peak power profile at peak

Figure 17: LRA power profile with time adaptation of implicit discretization and IQS-PC

Table 8: LRA step doubling adaptation results with implicit discretization and IQS-PC

	Implicit Dis.			IQS-PC		
Event	Power (W/cm^3)	Error	Steps	Power (W/cm^3)	Error	Steps
Max Power	5567.3	0.019454	423	5568.3	0.019274	47
End (3 s)	109.66	2.3650e-4	603	109.65	3.0622e-4	97

5.2. TREAT Transient-15 Problem

Transient 15 is a test case based on the TREAT core. The purpose of the original creation of this simulation in Rattlesnake is to test the model's fidelity with the thermal feedback of TREAT, but it is not meant to exactly match any previous experiments. Nevertheless, the goal of the following simulations is to test IQS and its time scale based treatment of temperature with a more complex model. Transient 15 involves an 11-energy group diffusion approximation and is discretized into $355,712$ hexahedral continuous finite elements totaling $4,109,523$ degrees of freedom. The three-second transient involves a linear ramp decrease in the absorption cross section throughout the control rod region. Fig. 18 shows a visualization of the flux profile within the core, hidden is the massive amount of graphite surrounding the core.

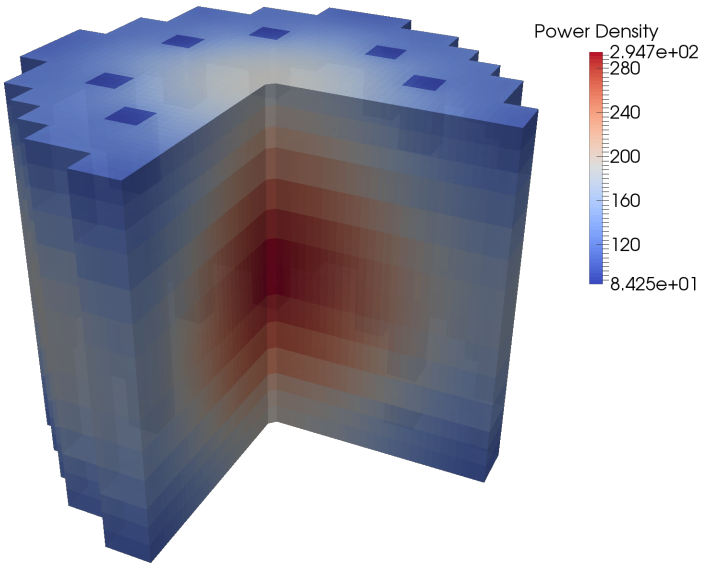


Figure 18: Transient 15 core power profile at peak power

The Transient-15 model uses a adiabatic temperature feedback mechanism, similar to the one explored by the LRA. Eq. (24) describes the heat up of the fuel. It is very similar, except the specific heat is now dependent on temperature is described by Eq. (25). The temperature evaluation is identical to the one described in LRA section, except a Newton iteration process is employed to resolve the nonlinearity from the specific heat term. The feedback to the cross-sections are applied using linear interpolation of tabular data provided by INL.

$$\rho c_p(T)\frac{\partial T(\boldsymbol{r},t)}{\partial t} = \kappa_f \sum_{g=1}^{G} \Sigma_f^g \phi^g(\boldsymbol{r},t) \tag{24}$$

$$c_p = -5.8219e-10T^3 - 4.3694e-7T^2 + 2.8369e-3T - 1.009e-2 \tag{25}$$

5.2.1. Transient-15 Multiphysics Time Scale Results

In order to test the temperature feedback treatment, six different scenarios were run: a baseline with a very small time step, implicit discretization, IQS with one and 5 temperature updates per macro step, and IQS-PC with one and 5 updates. Fig. 19 shows the baseline power and temperature profile for the Transient-15 example. Table 9 shows the error and runtime results.

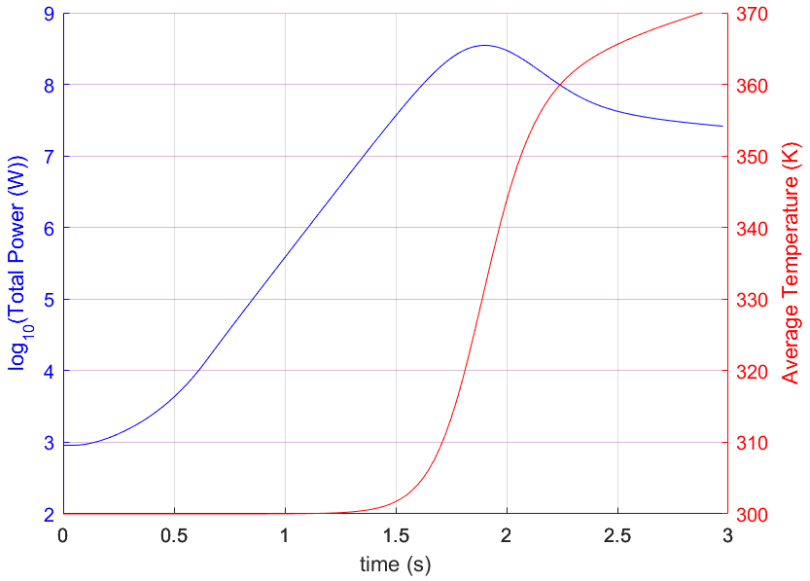


Figure 19: Transient-15 total power and average temperature profile during transient

Method	No. of Steps	Max Power (W)	Time at Max Power (s)	Max Average Temperature (K)	% Increase Runtime*	Max Power Error	Linear Iterations
Baseline	3000	3.5039e+08	1.901	371	—	—	—
Implicit Dis.	300	3.5011e+08	1.90	371	—	7.875e-4	41020
IQS	300	3.5036e+08	1.90	371	-11.9%	8.385e-5	23949
IQS (5 updates)	300	3.5040e+08	1.90	371	49.7%	3.687e-5	24035
IQS-PC	300	3.5065e+08	1.90	371	-2.1%	7.527e-4	39020
IQS-PC (5 updates)	300	3.5043e+08	1.90	371	26.5%	1.227e-4	37866

* difference in runtime from implicit discretization

Table 9: Transient-15 Error and Runtime Results

The results from Table 9 show similar performance of IQS with the temperature updates as the LRA. Again, the number of linear GMRES iterations is shown as a measure of computational expense. However, these iterations do not consider the temperature updates, so the iterations of the simulations with multiple updates should be taken with a grain of salt. IQS with 1 temperature update shows a performance that reduces the error to approximately a tenth of the implicit discretization error, and reduces the execution time by about 12%. This shows that IQS was able to resolve the nonlinearity between flux and

temperature with significantly fewer diffusion evaluations. Having IQS with 5 updates significantly increased the execution time for the same time step, but the error was reduced. Comparing this error to a similar implicit discretization error at a smaller time step could show that the runtime was reduced. IQS-PC performed not nearly as well as it did with the LRA benchmark, but still proved to be effective. Having 5 updates for IQS-PC increased the runtime marginally, but decreased the error significantly.

6. Conclusions

References

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

[2] J. Devooght, B. Arien, E. H. Mund, A. Siebertz, Fast reactor transient analysis using the generalized quasi-static approximation, Nuclear Science and Engineering 88 (1984) 191–199.

[3] A. Monier, Application of the collocation technique to the spatial discretization of the generalized quasistatic method for nuclear reactors, Ph.D. thesis, Université de Montréal (1991).

[4] M. Sissnoui, J. Koclas, A. Hébert, Solution of the improved and generalized quasistatic methods by kaps and rentrop integration scheme with stepsize control, Annals of Nuclear Energy 22 (1995) 763–774.

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

[6] K. Ott, D. Meneley, Accuracy of the quasistatic treatment of spatial reactor kinetics, Nuclear Science and Engineering 36 (1969) 381–419.

[7] J. Koclas, M. Sissnoui, A. Hébert, Solution of the improved and generalized quasistatic methods by kaps and rentrop integration scheme with stepsize control, Annals of Nuclear Energy 23 (1996) 901–907.

[8] S. Dulla, E. H. Mund, P. Ravetto, Accuracy of a predictor-corrector quasi-static method for space-time reactor dynamics, PHYSOR 2006.

[9] D. Caron, S. Dulla, P. Ravetto, Adaptive time step selection in the quasi-static methods of nuclear reactor dynamics, Annals of Nuclear Energy 105 (2017) 266 – 281. doi:http://dx.doi.org/10.1016/j.anucene.2017.03.009.
URL http://www.sciencedirect.com/science/article/pii/S030645491630994X

[10] D. Meneley, K. Ott, E. Wiener, FAST-REACTOR KINETICS: THE QX1 CODE., 1971.

[11] A. Keresztúri, G. Hegyi, C. Marázcy, M. Telbisz, I. Trosztel, C. Hegedűs, Development and validation of the three-dimensional dynamic codekiko3d, Annals of Nuclear Energy 30 (2003) 93–120.

[12] H. IKEDA, T. TAKEDA, Development and verification of an efficient spatial neutron kinetics method for reactivity-initiated event analyses, Journal of Nuclear Science and Technology 38 (2001) 496–515. doi:10.1080/18811248.2001.9715059.
URL http://dx.doi.org/10.1080/18811248.2001.9715059

[13] S. Goluoglu, H. L. Dodds, A time-dependent, three-dimensional neutron transport methodology, Nuclear Science and Engineering 139 (2001) 248–261.

[14] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, 1992.

[15] D. Gaston, C. Newman, G. Hansen, , D. Lebrun-Grandie, Moose: A parallel computational framework for coupled systems of nonlinear equations, Nucl. Engrg. Design 239 (2009) 1768 – 1778.

[16] Y. Wang, Nonlinear diffusion acceleration for multigroup transport equation discretized with sn and continuous fem with rattlesnake, in: Proc. International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering, Idaho, 2013.

[17] Z. M. Prince, J. C. Ragusa, Y. Wang, Improved quasi-static method: Iqs method implementation for cfem diffusion in rattlesnake, Tech. Rep. INL/EXT-16-38059, Idaho National Laboratory, Idaho Falls, Idaho (Feb 2016).

[18] Argonne Code Center, Benchmark problem book, anl- 7416, suppl. 2, Tech. rep., Argonne National Laboratory (1977).

[19] B. Gear, Backward differentiation formulas, Scholarpedia 2 (2007) 3162, revision 91024.

[20] L. Hageman, 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 (1969) 8 – 32.

[21] J. Ortensi, M. D. DeHart, F. N. Gleicher, Y. Wang, A. L. Alberti, T. S. Palmer, Full core treat kinetics demonstration using rattlesnake/bison coupling within mammoth, Tech. Rep. INL/EXT-15-36268, Idaho National Laboratory, Idaho Falls, Idaho (Sep 2015).

[22] J. F. Kirn, J. Boland, H. Lawroski, R. Cook, Reactor physics measurements in treat, Tech. Rep. ANL-6173, Agonne National Laboratory, Argonne, IL (Oct 1960).

[23] T. M. Sutton, B. N. Aviles, Diffusion theory methods for spatial kinetics calculations, Progress in Nuclear Energy 30 (1996) 119–182.