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

Multiphysics Core Dynamics Simulation Using the Improved Quasi-Static Method

Zachary M. Prince, Jean C. Ragusa

Department of Nuclear Engineering, Texas A&M University, College Station, TX
zachmprince@tamu.edu, jean.ragusa@tamu.edu

Abstract - *In this paper, we present a tightly coupled technique to solve multi-physics transient reactor simulations, whereby the neutronic solution is based on the improved quasi-static method (IQS). In IQS, the flux is typically factored into components, shape and amplitude, have greatly varying time scales. When multiphysic simulations are considered, an additional intermediate time scale to evaluate feedback (temperature and PRKE parameters) is used. The rationale for the intermediate time scale is based on the observation that temperature and point-reactor parameters may vary quickly in time than the shape itself. The method was implemented in the Rattlesnake/MOOSE framework of Idaho National Laboratory (INL) and tested with the LRA benchmark and the Transient-15 TREAT problem. The results show that the addition of this time scale was effective in reducing execution time for these problems while yielding accurate answers.*

I. INTRODUCTION

The purpose of this paper is to introduce several techniques in dealing with multi-physics feedback for transient neutron diffusion calculations. In dynamic simulations, the neutron temporal distribution in a nuclear core can be strongly influenced by non-neutronics variables, e.g., temperature. The improved quasi-static method (IQS) is an effective technique for simulating the kinetic behavior of the neutron flux in reactors. Here, we present a study combining the IQS method with multi-physics solution techniques for coupled transient calculations.

IQS is a transient spatial kinetic method that involves factorizing the neutron flux into a space- and time-dependent component (shape) and a time-dependent component (amplitude) [1, 2]. The technique relies on the shape being less rapidly varying in time compared to the flux, hence requiring fewer shape computations or updates. IQS has largely been applied to neutron kinetics, without other feedback mechanisms. This paper presents the application of multi-physics feedback with IQS and analyzes performance with temperature feedback problems.

The majority of publicized applications of IQS involve purely neutron kinetics, with several exceptions. Devooght et. al. discusses the application of the generalized quasi-static method with thermal hydraulic feedback within a newton iteration scheme in [3]. This implementation is not ideally efficient for IQS because, theoretically, the shape is less variant in time than temperature feedback and temperature is more closely coupled with amplitude. Other references such as [4] and [5] apply IQS to multi-physics feedback problems, but are ambiguous on their coupling treatment.

This paper presents a new approach to multi-physics feedback with IQS involving inclusion of feedback mechanisms in the quasi-static process. The intention of this implementation is to further optimize solution accuracy with computational effort. In order to evaluate the performance of the methodology, it will be tested with two temperature feedback problems: the LRA benchmark and an example from the Transient Reactor Testing Facility (TREAT) reactor. The performance is quantified by comparing accuracies with traditional implementations of neutron dynamics and IQS.

II. THEORY

In this Section, we recall the equation for the IQS method, starting from the standard multi-group diffusion equations with delayed neutron precursors in operator form:

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

$$\frac{dC_i}{dt} = \sum_{g=1}^{G} P_{d,i}^g \phi^g - \lambda_i C_i\,, \quad 1 \le i \le I \qquad (2)$$

where $H^{g'\to g}$ is the scattering operator, $P_p^{g'\to g}$ is the prompt fission operator, L^g is the diffusion operator, S_d^g is the delay source, and $P_{d,i}^g$ is the delayed-neutron fission operator.

The flux 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):

$$\phi^g(\boldsymbol{r}, t) = p(t)\varphi^g(\boldsymbol{r}, t) \qquad (3)$$

After implementing the factorization, the shape diffusion equations result:

$$\frac{1}{v^g}\frac{\partial \varphi^g}{\partial t} = \sum_{g'=1}^{G}\left(H^{g'\to g} + P_p^{g'\to g}\right)\varphi^{g'} - \left(L^g - \boxed{\frac{1}{v^g}\frac{1}{p}\frac{dp}{dt}}\right)\varphi^g + \boxed{\frac{1}{p}}S_d^g \qquad (4)$$

$$\frac{dC_i}{dt} = \boxed{p}\sum_{g=1}^{G} P_{d,i}^g \varphi^g - \lambda_i C_i\,, \quad 1 \le i \le I \qquad (5)$$

Note that the time-dependent shape equation is similar to the time-dependent flux equation. However, the shape equations are now non-linearly coupled (boxed terms) to the amplitude equations.

To obtain the amplitude equations, the multigroup shape equations are multiplied by a weighting function, typically the initial adjoint flux (ϕ^{*g}), and then integrated over the 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 $\sum_{g=1}^{G} \left(\phi^{*g}, \frac{1}{v^g}\varphi^g\right)$ to be constant. After some manipulation, the point reactor kinetics equations (PRKE) for the amplitude solution are obtained:

$$\frac{dp}{dt} = \left[\frac{\rho - \bar{\beta}}{\Lambda}\right] p + \sum_{i=1}^{I} \bar{\lambda}_i \xi_i \tag{6}$$

$$\frac{d\xi_i}{dt} = \frac{\bar{\beta}_i}{\Lambda} p - \bar{\lambda}_i \xi_i \quad 1 \le i \le I \tag{7}$$

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

$$\frac{\rho - \bar{\beta}}{\Lambda} = \frac{\sum_{g=1}^{G} \left(\phi^{*g}, \sum_{g'} (H^{g' \to g} + P_p^{g' \to g} - L^{g'} \delta_{g'g}) \varphi^{g'}\right)}{\sum_{g=1}^{G} \left(\phi^{*g}, \frac{1}{v^g}\varphi^g\right)} \tag{8}$$

$$\frac{\bar{\beta}}{\Lambda} = \sum_{i=1}^{I} \frac{\bar{\beta}_i}{\Lambda} = \sum_{i=1}^{I} \frac{\sum_{g=1}^{G} (\phi^{*g}, P_{d,i}^g \varphi^g)}{\sum_{g=1}^{G} \left(\phi^{*g}, \frac{1}{v^g}\varphi^g\right)} \tag{9}$$

$$\bar{\lambda}_i = \frac{\sum_{g=1}^{G} (\phi^{*g}, \chi_{d,i}^g \lambda_i C_i)}{\sum_{g=1}^{G} (\phi^{*g}, \chi_{d,i}^g C_i)} \tag{10}$$

Solving for the shape in Eq. (4) can become expensive, especially in two or three 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. The two-time scale solution process, a micro scale for the PRKE and a macro scale for the shape, is illustrated in Fig. 1.

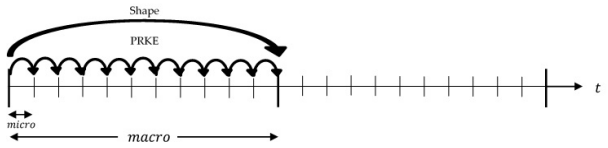

Fig. 1: IQS method visualization

It is important to note that the PRKE parameters are evaluated at each macro step and (linearly) interpolated for the PRKE evaluation. In order to preserve the error convergence of high order discretization schemes for shape, higher order interpolation of the parameters is required.

1. IQS Predictor-Corrector (IQS P-C)

To avoid performing the IQS non-linear solve, a predictor-corrector version of IQS has been derived in [2]. One first solves the neutron flux (represented by Equations 1 and 2) to obtain a predicted flux. The predicted flux is then converted to a shape through a rescaling argument :

$$\varphi_{n+1}^g = \underbrace{\phi_{n+1}^g}_{\text{predicted}} \frac{K_0}{K_{n+1}} \tag{11}$$

Where:

$$K_{n+1} = \sum_{g=1}^{G} \left(\phi^{*g}, \frac{1}{v^g}\phi_{n+1}^g\right) \tag{12}$$

$$K_0 = \sum_{g=1}^{G} \left(\phi^{*g}, \frac{1}{v^g}\varphi_{n+1}^g\right) = \sum_{g=1}^{G} \left(\phi^{*g}, \frac{1}{v^g}\phi_0^g\right) \tag{13}$$

The PRKE parameters are then computed with this shape using Equations (8)-(10) and interpolated over the macro step for the solution of the PRKE equations. With the newly computed amplitude, the shape is rescaled and the corrected flux is evaluated:

$$\underbrace{\phi_{n+1}^g}_{\text{corrected}} = p_{n+1} \times \varphi_{n+1}^g \,. \tag{14}$$

2. IQS Solution Process with Multiphysics

Other physical quantities, such as temperature, are affected by reaction rates and subsequently affect the operators of the flux equations. For IQS, this feedback affects both the shape equation and the parameters of the PRKE; thus, it is an additional nonlinear component to the already nonlinear shape-amplitude equations. Each of these components have different temporal behaviors; so, it may be beneficial for efficiency to evaluate them on different time scales. The amplitude is more rapidly varying than the shape which is computationally expensive to evaluate and is evaluated only on macro-time steps. In multiphysics simulations, one may take advantage of a fine-scale power distribution in the coupled physics components (temperature). Figure 2 shows a such a solution process for temperature feedback.

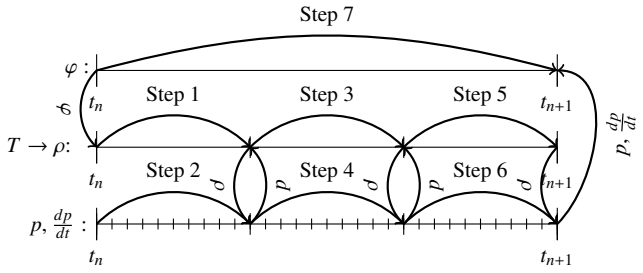

Fig. 2: Time scales and process of IQS

The top time scale represents a shape diffusion evaluation on a macro step, the middle has an arbitrary three steps (for illustration) within the macro step where temperature and the PRKE parameters are evaluated, and the bottom one represents the PRKE evaluation on micro steps. The shape is linearly interpolated within the macro step for the temperature and PRKE parameter evaluation, and the parameters are interpolated within the temperature step for the PRKE evaluation. Since there is a nonlinear coupling between all these components, each temperature step is iterated until amplitude has converged and the macro step is iterated until shape has converged. Fig. 3 further visualizes this process as a programming structure.

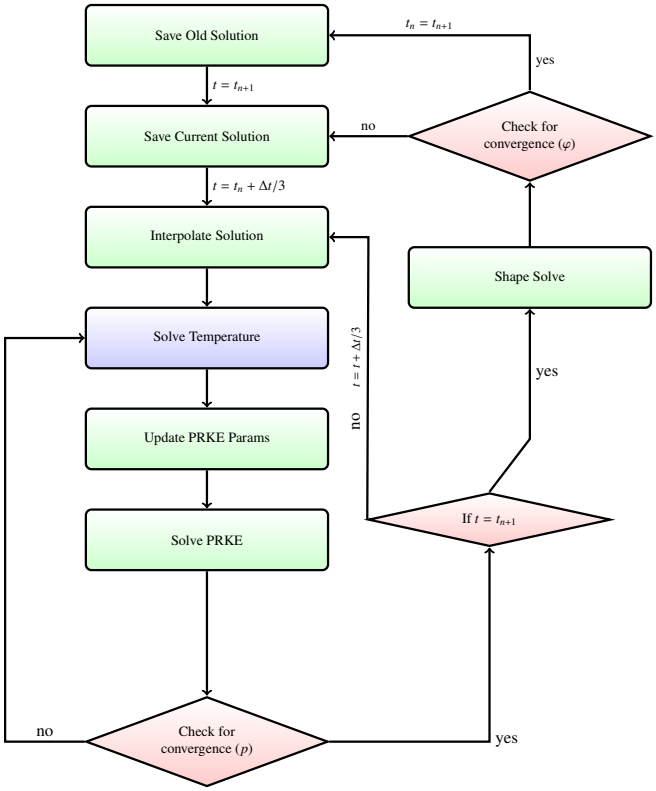

Fig. 3: Visualization of shape iteration and temperature update process for IQS

3. Dynamical Time Scale Analysis

The temporal variance of flux, shape, temperature, and amplitude can be quantified by defining a dynamical time scale (τ) for each physics. A small value τ means the variable is quickly varying in time and consequently needs relatively small time steps for accuracy, vice-versa for large τ. The general definition of τ is defined by Eq. (15), where θ is the physic component of interest.

$$\tau = \frac{1}{\left|\frac{1}{\theta}\frac{d\theta}{dt}\right|} \tag{15}$$

Since each variable in discretized in time, a finite difference approximation will be made for the $\frac{d\theta}{dt}$ term and the average between the two corresponding time steps will be made for the $\frac{1}{\theta}$ term. Additionally, τ is spatially dependent for flux and temperature, but only the time dependent behavior of this quantity is of interest. Therefore, the L^2 norm of each term will be used to compute the approximate time scale ($\tilde{\tau}$), formally defined by Eq. (16). θ represents a summation over groups for flux and shape.

$$\tilde{\tau}_{n+1} = \frac{\|\theta_{n+1} + \theta_n\|_{L^2}}{2} \frac{\Delta t}{\|\theta_{n+1} - \theta_n\|_{L^2}} \tag{16}$$

According to the a priori hypothesis from previous sections, τ is large for shape, somewhat smaller for temperature, and much smaller for amplitude and flux.

III. RESULTS AND ANALYSIS

To test the multiple time scale implementation of IQS, the LRA benchmark [6] and a TREAT reactor example have been selected. The performance of the implementation is evaluated comparing errors in power at peak time with standard flux solution techniques ("brute force") and IQS without quasi-static temperature treatment. Execution times were also compared at given error values to judge the computational efficiency of adding more temperature evaluations. These examples were executed using INL's MOOSE/Rattlesnake framework, which uses a PJFNK (preconditioned Jacobian-free Newton Krylov) method to evaluate the nonlinear problems. Each time step evaluation consists of three iteration loops: GMRES iterations for the linear system, Newton iterations for the temperature-flux nonlinearity, and Picard iterations for the shape-amplitude nonlinearity. Another judge of computational expense is the total number of linear iterations during the execution, these are shown specifically for the brute force LRA simulations and all the simulations for the TREAT example.

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) [7]. The spacial discretization was performed using continuous finite element method with first order Lagrangian basis functions. The mesh consisted of blocks 11×11 with five uniform refinements, totaling 165, 165 elements and 124, 609 nodes. Three different temporal techniques were applied: implicit discretization of the flux equation ("brute force"), IQS, and IQS-PC. Crank-Nicholson time discretization scheme was used for the diffusion evaluation of each technique. Third order Runge-Kutta discretization with step doubling adaptation was used for the PRKE evaluation. The performance of IQS and the temperature updates were measured by its improvement in accuracy at peak power over the Brute force technique.

A. LRA Temperature Feedback

The heat up is described by Eq. (17) and the feedback is described by Eq. (18).

$$\rho c_p \frac{\partial T(\boldsymbol{r},t)}{\partial t} = \kappa_f \sum_{g=1}^{G} \Sigma_f^g \phi^g(\boldsymbol{r},t) \tag{17}$$

$$\Sigma_a^{thermal}(\boldsymbol{r},t) = \Sigma_a^{thermal}(\boldsymbol{r},0)\left[1 + \gamma\left(\sqrt{T} - \sqrt{T_0}\right)\right] \tag{18}$$

In the temperature evaluation, a typical implicit solver would simply use the interpolated flux at end of the temperature time step for the right-hand-side of the equation. However, IQS has much more information about the profile of the flux along the time step because of the micro-step amplitude evaluation. Therefore, it is possible to solve for temperature using a semi-analytical approach, shown by Eq. (19).

$$T^{n+1} = T^n + \frac{\kappa_f}{\rho c_p}\left(a_2\varphi^{n+1} + a_1\varphi^n\right) \tag{19}$$

where n corresponds to the beginning of the temperature step. a_1 and a_2 are integration coefficients defined by Eq. (20) and Eq. (21).

$$a_1 = \int_{t_n}^{t_{n+1}} \left(\frac{t_{n+1} - t'}{\Delta t} \right) p(t')dt' \tag{20}$$

$$a_2 = \int_{t_n}^{t_{n+1}} \left(\frac{t' - t_n}{\Delta t} \right) p(t')dt' \tag{21}$$

Any interpolation of the amplitude function $p(t)$ along the micro steps is possible to carry out the above macro-step integrals. Here, we use piece-wise linear interpolation.

B. LRA Results

Fig. 4 shows the baseline power and temperature transient profile for the LRA benchmark. Fig. 5 shows the spacial power distribution at the peak power. The baseline results are compared to the results achieved by Sutton and Aviles in [8] and presented in Table I. The relative difference in the magnitude of the peak power ($t \approx 1.44s$) from the baseline was used for error comparison. Fig. 6 is an error convergence plot comparing the three techniques where temperature is evaluated only on the macro step (1 temperature update). Fig. 7 is an error convergence plot comparing the three techniques where temperature is evaluated 5 times within a macro step (5 temperature updates). Finally, Fig. 8 shows the effect of various temperature updates. The dashed lines correspond to brute force at different flux step sizes, while the IQS macro step size is kept constant.

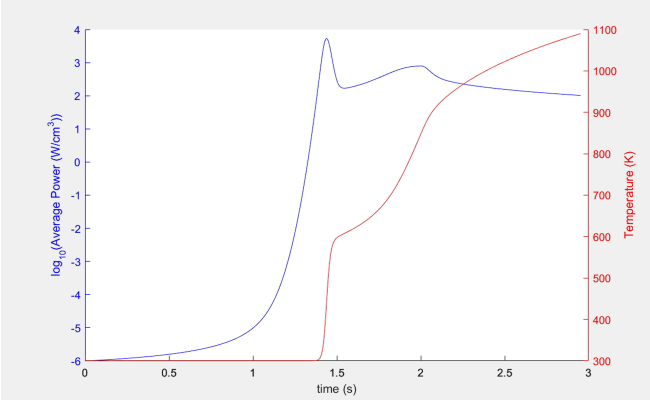

Fig. 4: LRA baseline temperature and power profile

Calculation	Baseline	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

TABLE I: LRA baseline verification

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,

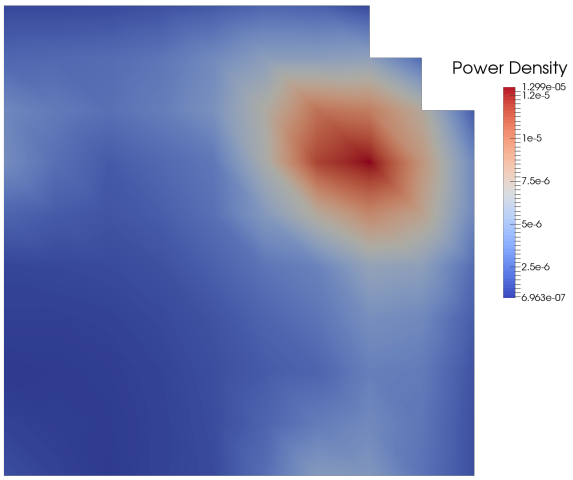

Fig. 5: LRA baseline spacial power profile at $t = 1.44s$

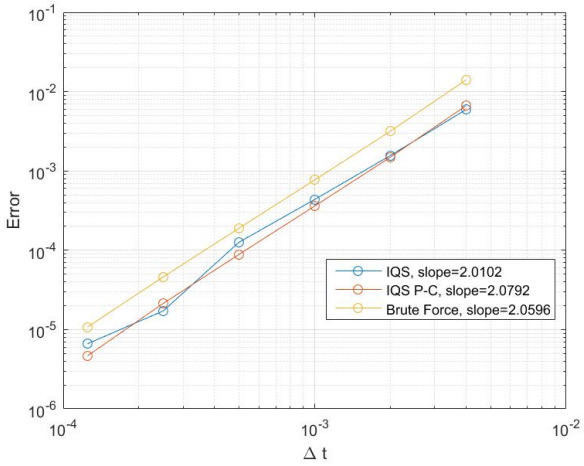

Fig. 6: LRA convergence plots with only one temperature update per macro step

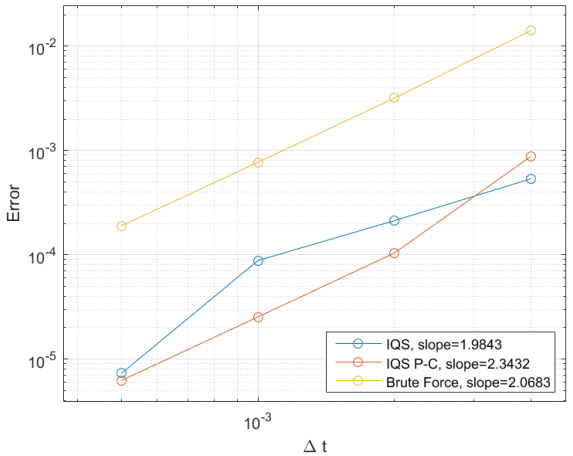

Fig. 7: Error convergence plot with 5 temperature updates per macro step

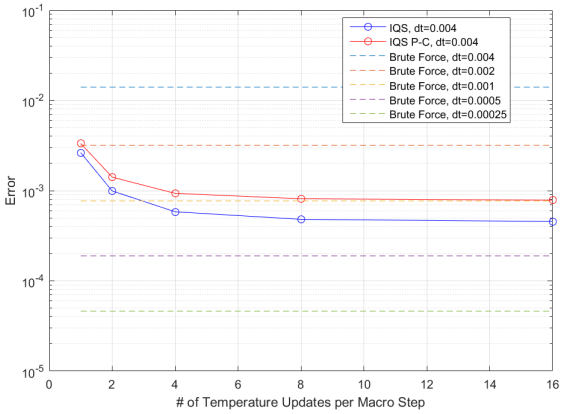

Fig. 8: Error plot with various temperature updates per macro step

IQS was only slightly better than brute force, brute force required about 150% more time steps than IQS for the same error. While 5 temperature updates showed a much more significant IQS performance, brute force required about 400% more time steps than IQS for the same error. Fig. 8 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 II shows the run time results for the brute force calculations. Tables III and IV 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 over 24 processors. These run-times show a marginal performance for IQS and impressive performance for IQS P-C. Some of the execution times were able to decrease from brute force 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 P-C 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 brute force with IQS at one update and IQS P-C at 4 updates. IQS shows an error comparable to brute force at Δt = 0.002, signifying an actual increase in runtime by -34.1%. IQS P-C shows an error less than brute force at Δt = 0.002, signifying an actual increase in runtime by <-34.9%.

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 II: Brute force run time results

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 Δt = 0.004 brute force

TABLE III: IQS run time results with Δt = 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 Δt = 0.004 brute force

TABLE IV: IQS PC run time results with Δt = 0.004

The performance of the quasi-statics can also be explained by the computation of the dynamical time scale described by Section 3.. Fig. 9 shows the time scale profile over the transient, computed using Eq. (16). This plot shows that in a brute force simulation, the flux dominates the time dependent behavior, while temperature lags in its variance for the majority of the transient. In an IQS simulation, the time scale behavior of amplitude almost exactly matches the flux, while shape is more varying than temperature throughout most of the transient. The large τ for temperature during the beginning of the transient shows that adaptation of the number of updates is important; computational expense on temperature evaluations is being wasted during this time.

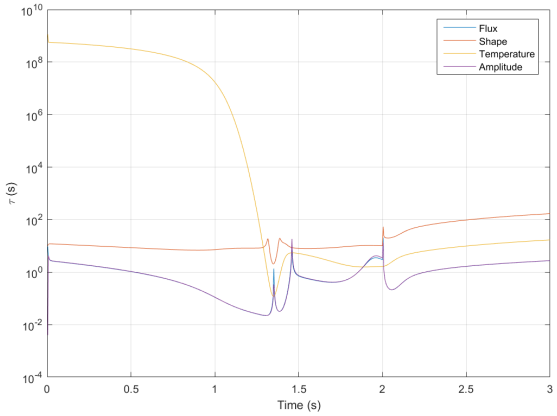

Fig. 9: Dynamical time scale for LRA benchmark

2. Transient-15 TREAT Example

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. 10 shows a visualization of the flux profile within the core, hidden is the massive amount of graphite surrounding the core.

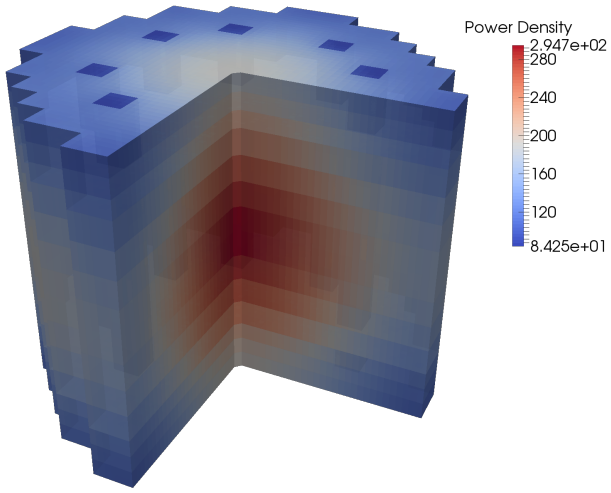

Fig. 10: Transient 15 core power profile at peak power

A. Transient-15 Temperature Feedback

The Transient-15 model uses a adiabatic temperature feedback mechanism, similar to the one explored by the LRA. Eq. (22) describes the heat up of the fuel. It is very similar, except the specific heat is now dependent on temperature is described by Eq. (23). 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.

$$\rho c_p(T)\frac{\partial T(\boldsymbol{r},t)}{\partial t} = \kappa_f \sum_{g=1}^{G} \Sigma_f^g \phi^g(\boldsymbol{r},t) \tag{22}$$

$$c_p = -5.8219e{-}10T^3 - 4.3694e{-}7T^2 + 2.8369e{-}3T - 1.009e{-}2 \tag{23}$$

B. Transient-15 Results

In order to test the temperature feedback treatment, six different scenarios were run: a baseline with a very small time step, brute force, IQS with one and 5 temperature updates per macro step, and IQS P-C with one and 5 updates. Fig. 11 shows the baseline power and temperature profile for the Transient-15 example. Table V shows the error and runtime results.

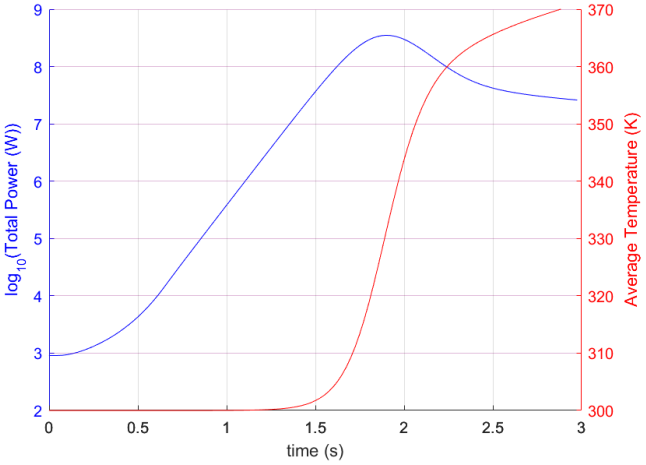

Fig. 11: Transient 15 total power and average temperature profile during transient

The results from Table V show similar performance of IQS with the temperature updates as the LRA. IQS with 1 temperature update shows a performance that reduces the error to approximately a tenth of the brute force 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 brute force error at a smaller time step could show that the runtime was reduced. IQS P-C performed not nearly as well as it did with the LRA benchmark, but still proved to be effective. Having 5 updates for IQS P-C increased the runtime marginally, but decreased the error significantly. The transient profile of the variables' dynamical time scales is shown in Fig. 12. This plot exhibits a similar response to that of the LRA. The response of temperature shows that the updates are a computational frugal treatment of the feedback and adaptation of the number of updates is vital for optimization.

IV. CONCLUSIONS

IQS is a powerful tool for reactor simulation, it has the capability of greatly reducing computation time while retaining accuracy of relevant quantities. The goal of this paper is to demonstrate IQS's performance with multi-physics simulations, as well as techniques to improve its performance. The technique focused on incorporating a new time scale in IQS for feedback quantities and PRKE parameters. The rationale is that for problems involving prompt-critical transients, temperature is more quickly varying than shape, but still slower than the amplitude itself. Therefore, an intermediate time scale was implemented, as well as a fixed-point iteration between amplitude and temperature. The attempt is to reduce the number of multi-group diffusion evaluations in a transient by reducing the number of time steps required for a given accuracy, and reduce the number of nonlinear iterations between flux/shape and temperature. Since multi-group diffusion evaluations are

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	—	—	—
Brute Force	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 P-C	300	3.5065e+08	1.90	371	-2.1%	7.527e-4	39020
IQS P-C (5 updates)	300	3.5043e+08	1.90	371	26.5%	1.227e-4	37866

* difference in runtime from brute force

TABLE V: Transient-15 Error and Runtime Results

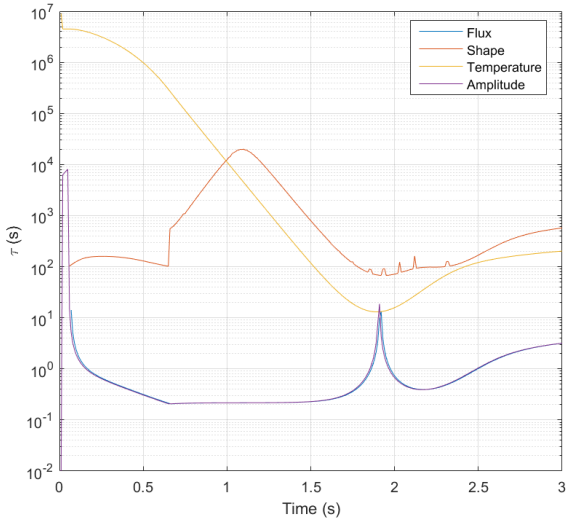

Fig. 12: Dynamical time scale for the Transient 15 example

significantly more computationally expensive than temperature and amplitude evaluations, this result will show a decrease in execution time for a given accuracy.

Two examples were chosen to test the implementation of the temperature feedback treatment: the LRA benchmark and the Transient-15 TREAT problem. Both problems involve adiabatic heat up of the core, the LRA has a constant specific heat, while Transient-15's is dependent on temperature. The LRA showed clear results concluding a marginal improvement in performance for IQS and a impressive performance for IQS P-C. The results proved that the updates improved the performance for both IQS and IQS P-C, but optimization of the number of updates is important, possibly requiring adaptation. The implementation to the Transient-15 problem gave similar results. IQS proved to marginally benefit from the updates, decreasing error but increasing execution time. IQS P-C was benefited more by the updates in this example. The updates were able to reduce the error with a marginal increase in execution time.

In conclusion, the multi-physics treatment with IQS presented in this paper is worth further investigation, optimization, and testing with additional transient reactor problems with other feedback mechanisms.

V. ACKNOWLEDGMENTS

This work was supported by the Department of Energy, Idaho National Laboratory, and the Integrated University Program Fellowship. We thank Mark Dehart, Yaqi Wang, the NEAMS program, and INL's MOOSE/Rattlesnake team for their support.

REFERENCES

1. K. OTT, "Quasi-static treatment of spatial phenomena in reactor dynamics," *Nuclear Science and Engineering*, **26**, 563 (1966).
2. S. DULLA, E. H. MUND, and P. RAVETTO, "The quasi-static method revisited," *Progress in Nuclear Energy*, **50**, *8*, 908 – 920 (2008).
3. J. DEVOOGHT, B. ARIEN, E. H. MUND, and A. SIEBERTZ, "Fast reactor transient analysis using the generalized quasi-static approximation," *Nuclear Science and Engineering*, **88**, 191–199 (1984).
4. J. E. BANFIELD, *Semi-Implicit Direct Kinetics Methodology for Deterministic, Time-Dependent, Three-Dimensional, and Fine-Energy Neutron Transport Solutions*, Ph.D. thesis, University of Tennesee (2013).
5. A. KERESZTÚRI, G. HEGYI, C. MARÁZCY, M. TELBISZ, I. TROSZTEL, and C. HEGED´US, "Development and validation of the three-dimensional dynamic code - KIKO3D," *Annals of Nuclear Energy*, **30**, 93–120 (2003).
6. ARGONNE CODE CENTER, "Benchmark Problem Book, ANL- 7416, Suppl. 2," Tech. rep., Argonne National Laboratory (1977).
7. 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).
8. T. M. SUTTON and B. N. AVILES, "Diffusion Theory Methods for Spatial Kinetics Calculations," *Progress in Nuclear Energy*, **30**, 119–182 (1996).