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 - *The purpose of this paper is to present a new technique in dealing with multi-physics feedback in transient reactor simulations, specifically utilizing the improved quasi-static method (IQS). the technique involves the addition of an intermediate time scale to evaluate temperature and PRKE parameters. The theory is that temperature and the parameters are more quickly varying in time than the shape, thus further reducing the number of multi-group diffusion evaluations and execution time for a given accuracy. The technique was implemented into the Rattlesnake/MOOSE framework at Idaho National Laboratory (INL) and tested with the LRA benchmark and the Transient-15 TREAT problem. The results proved that the addition of this time scale was effective in reducing error and execution time for these problems.*

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 temporal 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 kinetics 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 benchmark 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 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:

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

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

Where $H^{g'\to g}$ is the scattering operator, $P_p^{g'\to g}$ is the prompt fission operator, L^g is the streaming and interaction 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):

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

After implementing the factorization, the shape diffusion equations result:

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

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

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

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

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

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

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.

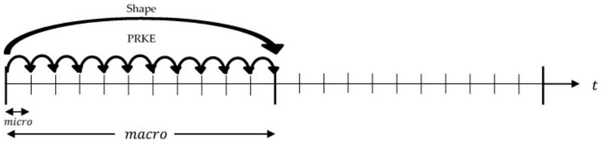

Fig. 1: IQS method visualization

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 dependencies; 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.

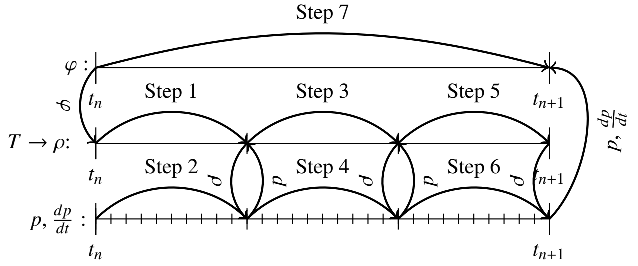

Fig. 2: Time scales and process of IQS

The top time scale represents a shape diffusion evaluation on a macro step, the middle has an arbitrary three steps 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.

III. RESULTS AND ANALYSIS

To test the multiple time scale implementation of IQS, the LRA benchmark was chosen for reference and a TREAT

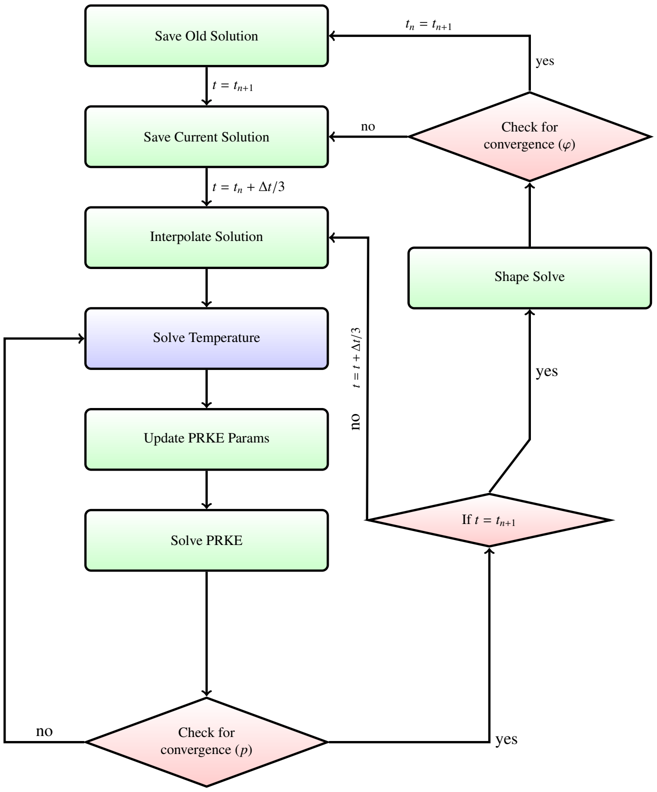

Fig. 3: Visualization of shape iteration and temperature update process for IQS

example was employed.

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 [6]. 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 11X11 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. (15) and the feedback is described by Eq. (16).

$$\rho c_p \frac{\partial T(\boldsymbol{r},t)}{\partial t} = \kappa_f \sum_{g=1}^{G} \Sigma_f^g \phi^g(\boldsymbol{r},t) \qquad (15)$$

$$\Sigma_a^{thermal}(\boldsymbol{r},t) = \Sigma_a^{thermal}(\boldsymbol{r},0)\left[1+\gamma\left(\sqrt{T}-\sqrt{T_0}\right)\right] \qquad (16)$$

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. (17).

$$T^{n+1} = T^n + \frac{\kappa_f}{\rho c_p}\left(a_2\varphi^{n+1} + a_1\varphi^n\right) \qquad (17)$$

Where n corresponds to the beginning of the temperature step. a_1 and a_2 are integration coefficients defined by Eq. (18) and Eq. (19).

$$a_1 = \int_{t_n}^{t_{n+1}} \left(\frac{t_{n+1} - t'}{\Delta t}\right) p(t')dt' \qquad (18)$$

$$a_2 = \int_{t_n}^{t_{n+1}} \left(\frac{t' - t_n}{\Delta t}\right) p(t')dt' \qquad (19)$$

Any interpolation of the amplitude along the micro steps is possible for the integration, this application uses piece-wise linear.

B. LRA Results

Fig. 4 shows the baseline power and temperature profile for the LRA benchmark. 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. 5 is an error convergence plot comparing the three techniques where temperature is evaluated only on the macro step (1 temperature update). Fig. 6 is an error convergence plot comparing the three techniques where temperature is evaluated 5 times within a macro step (5 temperature updates). Finally, Fig. 7 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.

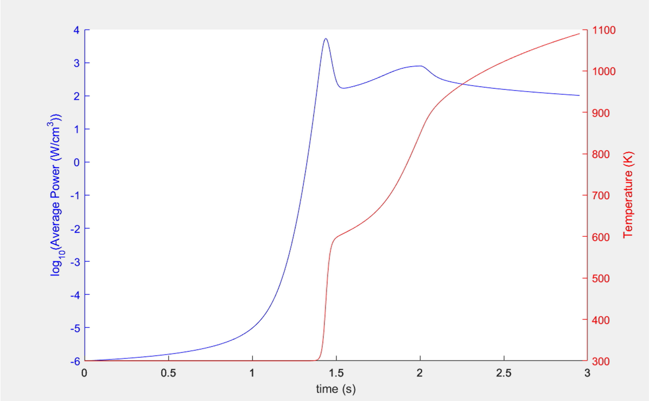

Fig. 4: LRA baseline temperature and power profile

Calculation	Baseline	Sutton (Spandex 1936)
No. of Mesh	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

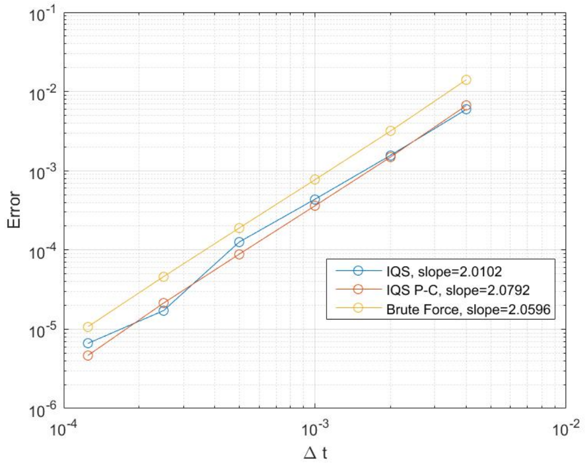

Fig. 5: LRA convergence plots with only one temperature update each macro step

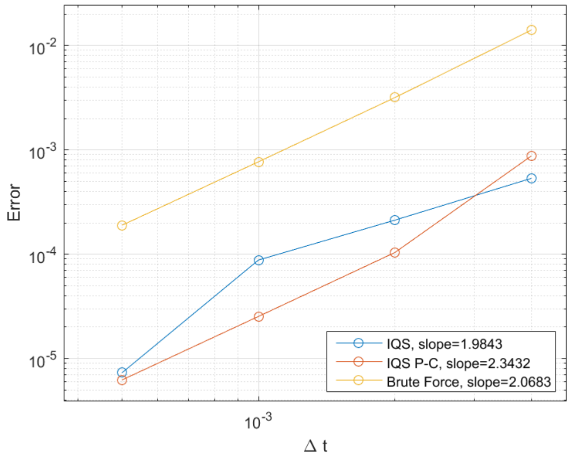

Fig. 6: Error convergence plot with 5 temperature updates per macro step

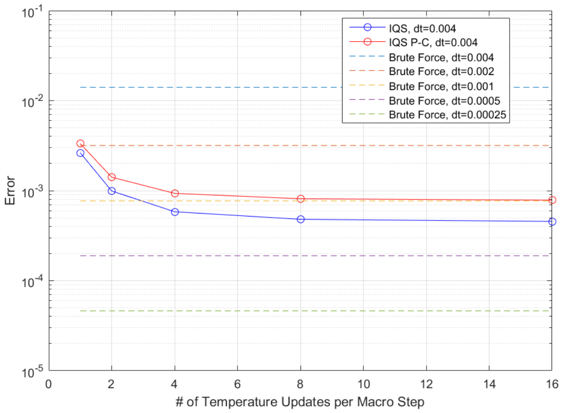

Fig. 7: Error plot with various temperature updates per macro step

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 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. 7 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 implicit discretization 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 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 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.

Run	Δt	Error	Runtime (hr)
1	4.0e-3	1.407e-2	4.11
2	2.0e-3	3.174e-3	5.68
3	1.0e-3	7.690e-4	5.55
4	5.0e-4	1.892e-4	13.57
5	2.5e-4	4.590e-5	23.81

TABLE II: Implicit discretization 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 $\Delta t = 0.004$ implicit discretization

TABLE III: IQS 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

TABLE IV: IQS PC run time results with $\Delta t = 0.004$

2. Transient-15 TREAT Example

Transient 15 is a small core TREAT experiment that was modeled by the Rattlesnake group at INL for demonstrating their model's fidelity to TREAT's feedback mechanisms. The goal of the simulations is to obtain a power peak whose position and magnitude that sufficiently matches experimental results. The model 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 boron content throughout the fuel region. Fig. 8 shows a visualization of the flux profile within the core, hidden is the massive amount of graphite surrounding the core.

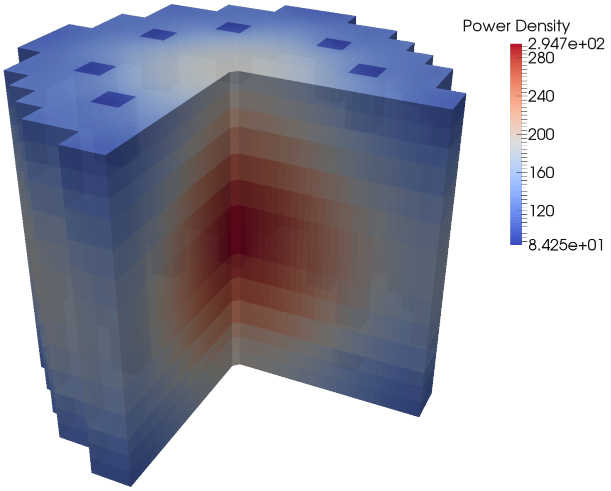

Fig. 8: Transient 15 core power profile at peak power

A. Transient-15 Temperature Feedback

The Transient-15 model uses a adiabatic temperature feedback mechanism, similar to the one explored by the LRA. Eq. (20) describes the heat up of the fuel. It is very similar, except the specific heat is now dependent on temperature is described by Eq. (21). 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.

$$\rho c_p(T)\frac{\partial T(\boldsymbol{r},t)}{\partial t} = \kappa_f \sum_{g=1}^{G} \Sigma_f^g \phi^g(\boldsymbol{r},t) \tag{20}$$

$$c_p = -5.8219e{-}10T^3 - 4.3694e{-}7T^2 + 2.8369e{-}3T - 1.009e{-}2 \tag{21}$$

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, implicit discretization, IQS with one and 5 temperature updates per macro step, and IQS P-C with one and 5 updates. Fig. 9 shows the baseline power and temperature profile for the Transient-15 example. Table V shows the error and runtime results.

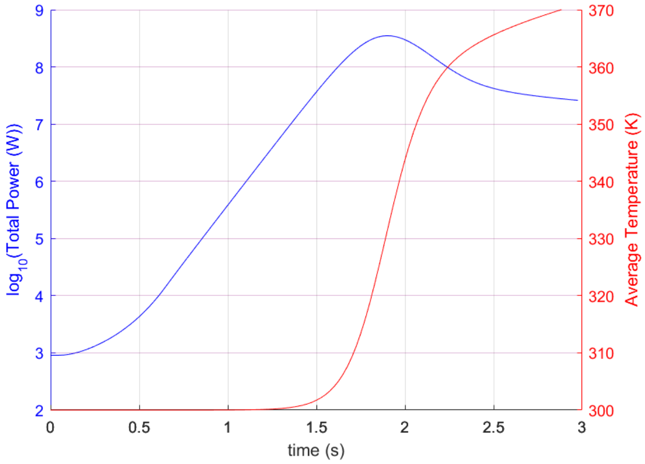

Fig. 9: Transient 15 total power and average temperature profile during transient

The results from Table V prove the performance of IQS with the temperature updates is slightly more ambiguous than the results from the LRA. IQS with 5 temperature updates shows a performance that reduces the error to approximately a tenth of the implicit discretization error, and reduces the execution time by about 20%. This shows that the 5 updates were able to resolve the nonlinearity between flux and temperature with significantly less diffusion evaluations. However, IQS P-C did not perform nearly as well and proved almost detrimental to the problem. With only one update, IQS P-C achieved a less accurate solution and increased the execution time. Only with 5 updates was the error reduced from the implicit discretization error, but the increase in

Method	No. of Steps	Max Power (W)	Time at Max Power (s)	Max Average Temperature (K)	% Increase Runtime*	Max Power Error
Baseline	3000	3.504e+08	1.9	370	—	—
Implicit Dis.	65	3.424e+08	1.875	370	—	0.0230
IQS	61	3.527e+08	1.9	371	17.9%	0.00643
IQS (5 updates)	61	3.514e+08	1.9	371	-19.2%	0.00339
IQS P-C	66	4.340e+08	1.9	374	20.3%	0.238
IQS P-C (5 updates)	66	3.537e+08	1.875	372	81.9%	0.00935

* difference in runtime from implicit discretization

TABLE V: Transient-15 Error and Runtime Results

run-time was significant.

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 methodology is that for problems involving prompt-critical transients, temperature and its feedback to amplitude is more quickly varying than shape, but less time-dependent than 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 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 more ambiguous results. IQS proved to significantly benefit from the updates, increasing accuracy and decreasing execution time. IQS P-C was able to reduce the error with the updates, but the execution time was significantly greater.

In conclusion, the multi-physics treatment with IQS presented in this paper is worth further investigation, optimization, and implementation into other 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 codeâĂŤKIKO3D," *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).