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

INTRODUCTION

The purpose of this summary is to introduce several techniques in dealing with multiphysics feedback for transient neutron diffusion calculations. In dynamic simulation, 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 multiphysics 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) [**?**, **?**]. 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 mostly been applied to neutron kinetics, without thermal-hydraulic feedback. This summary discusses the application of multiphysics feedback with IQS and analyzes performance with temperature feedback benchmark problems

THEORY

In this Section, we recall the equation for the IQS method, starting from the standard multi-group transport 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 (ψ): your notation is not consistent. psi. phi?

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

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:

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

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

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

IQS Predictor-Corrector (IQS P-C)

To avoid performing the IQS non-linear solve, a predictor-corrector of IQS has been derived in [**?**]. One first solves the neutron flux (represented by Equations 4 and 5) 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:

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

$$K_0 = \sum_{g=1}^{G} \left(\phi^{*g}, \frac{1}{v^g} \varphi^g_{n+1} \right) = \sum_{g=1}^{G} \left(\phi^{*g}, \frac{1}{v^g} \phi^g_0 \right) \tag{13}$$

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:

$$\underbrace{\phi^g_{n+1}}_{\text{corrected}} = p_{n+1} \times \varphi^g_{n+1} \,. \tag{14}$$

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 1 shows a such a solution process for temperature feedback.

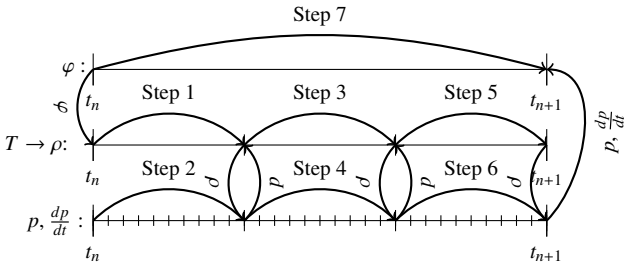

Fig. 1: Time scales and process of IQS

The top time scale represents a shape diffusion evaluation on a macro step, the middle as 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.

RESULTS AND ANALYSIS

To test the multiple time scale implementation of IQS, the LRA benchmark was chosen for reference. 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. Three different evaluation techniques were applied: temporal 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. The performance of IQS and the temperature updates was measured by its improvement in accuracy at peak power over the Brute force technique. The results show the effectiveness of multiple temperature updates along the macro step. you comment on results too early

LRA Temperature Feedback

The heat up is described by Eq. (15) and the feedback is described by Eq. (16).

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

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

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 quasi-analytical approach, shown by Eq. (17).

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

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).

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

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

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

LRA Convergence Results

Fig. 2 shows the baseline power and temperature profile for the LRA benchmark. The relative difference in the magnitude of the peak power ($t \approx 1.44s$) from the baseline was used for error comparison. Fig. 3 is an error convergence plot comparing the three techniques where temperature is evaluated only on the macro step (1 temperature update). Fig. 4 is an error convergence plot comparing the three techniques where temperature is evaluated 5 times within a macro step (5 temperature updates). Finally, Fig. 5 shows the affect of various temperature updates. The dotted lines correspond to brute force 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 affect on the performance of IQS. With only one update, IQS

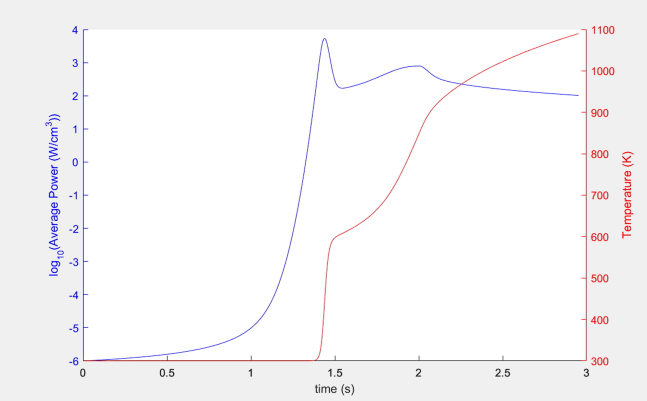

Fig. 2: LRA baseline temperature and power profile

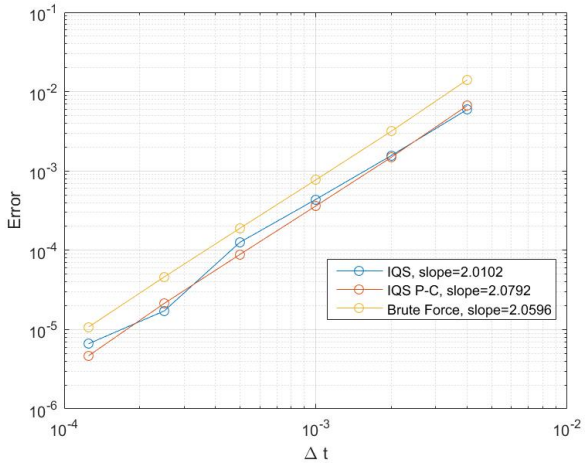

Fig. 3: LRA convergence plots with only one temperature updated each macro step

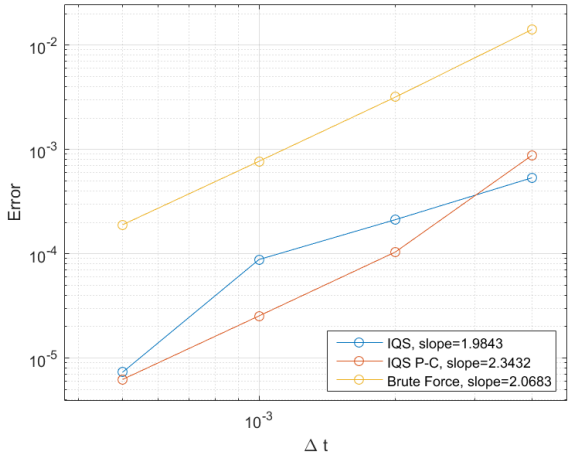

Fig. 4: Error convergence plot with 5 temperature updates per macro step

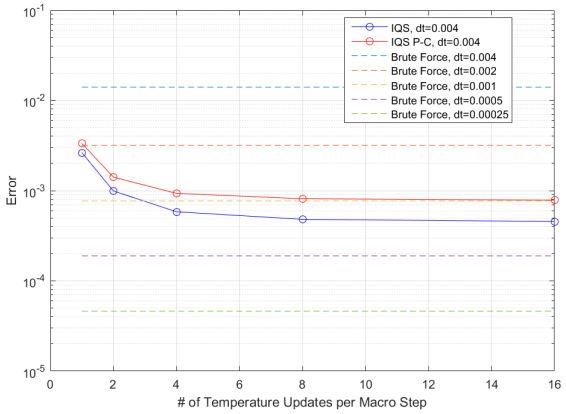

Fig. 5: Error plot with various temperature updates per macro step

was only slightly better than brute force, brute force required about 1.5x more time steps than IQS for the same error. While 5 temperature updates showed a much more significant IQS performance, brute force required about 4x more time steps than IQS for the same error. Fig. 5 shows that there is an exponential convergence of error for the number of temperature updates. This convergence makes sense because temperature can only be so accurate before the error in shape is dominating.

CONCLUSIONS

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