CERT
Center for Exascale Radiation Transport

IQS: Status Update

Jean C. Ragusa, Zach M. Prince

Department of Nuclear Engineering, Texas A&M University, College Station, TX
Institute for Scientific Computation, Texas A&M University, College Station, TX
Center for Exascale Radiation Transport, Texas A&M University, College Station, TX

October 20, 2015

email: jean.ragusa@tamu.edu

Outline

1. Theory
   - Time-dependent Multigroup Diffusion
   - Factorization approach
   - IQS equations
   - IQS method solution process
   - Precursors time-discretization

2. Implementation in YAK/RATTLESNAKE

3. Results
   - Prototype code
   - YAK/RATTLESNAKE

4. Wrap-up

Time-dependent Multigroup Diffusion

Equations

Neutron flux ϕ^g:

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

Precursors C_i

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

with

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

Factorization

Factorization

Decomposition of the multigroup flux into the product of a time-dependent amplitude (p) and a space-/time-dependent multigroup shape (φ):

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

When reporting this in the previous equations, this leads to the so-called shape equations.

Shape equations

Shape equations

The shape equations are similar to the original diffusion equations:

$$\frac{1}{v^g}\frac{\partial \varphi^g}{\partial t} = \frac{\chi_p^g}{k_{eff}}\sum_{g'=1}^{G}(1-\beta)\nu^{g'}\Sigma_f^{g'}\varphi^{g'} - \left(-\vec{\nabla}\cdot D^g\vec{\nabla} + \Sigma_r^g + \frac{1}{v^g}\frac{1}{p}\frac{dp}{dt}\right)\varphi^g + \sum_{g'\neq g}^{G}\Sigma_s^{g'\to g}\varphi^{g'} + \frac{1}{p}\sum_{i=1}^{I}\chi_{d,i}^g\lambda_i C_i \ , \quad 1 \le g \le G \tag{5}$$

$$\frac{dC_i}{dt} = \frac{\beta_i}{k_{eff}}\sum_{g=1}^{G}\nu^g\Sigma_f^g p\varphi^g - \lambda_i C_i \ , \quad 1 \le i \le I \tag{6}$$

Differences

1. An additional removal term based on $\frac{1}{v^g}\frac{1}{p}\frac{dp}{dt}\varphi^g$
2. Delayed neutron source term scaled by $\frac{1}{p}$
3. No change in the precursor equations but we re-write them to show explicitly $p \times \varphi^g$

Amplitude equations (PRKE)

Principle

To obtain the amplitude equation, we multiply the shape equations with a weighting function (initial adjoint flux, ϕ^*), then integrate over phase-space.

Notation

For brevity, the adjoint flux product and integration over space will be represented with parenthetical notation:

$$\int_D \phi^{*g}(\vec{r}) f^g(\vec{r}) d^3r = (\phi^{*g}, f^g) \tag{7}$$

Uniqueness of the factorization

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

$$K_0 = \sum_{g=1}^{G} \left(\phi^{*g}, \frac{1}{v^g} \varphi^g \right) = constant \tag{8}$$

PRKE (continued)

PRKE

$$\frac{dp}{dt} = \left[\frac{\rho - \bar{\beta}}{\Lambda}\right] p + \sum_{i=1}^{I} \bar{\lambda}_i \xi_i \tag{9}$$

$$\frac{d\xi_i}{dt} = \frac{\bar{\beta}_i}{\Lambda} - \bar{\lambda}_i \xi_i \quad 1 \le i \le I \tag{10}$$

Coefficients

$$\frac{\rho}{\Lambda} = \sum_{g=1}^{G} \frac{\left(\phi^{*g}, \sum_{g'=1}^{G} \frac{\chi_p^g}{k_{eff}} \nu^{g'} \Sigma_f^{g'} \varphi^{g'} + \sum_{g' \neq g}^{G} \Sigma_s^{g' \to g} \varphi^{g'} - \left(-\vec{\nabla} \cdot D^g \vec{\nabla} + \Sigma_r^g\right) \varphi^g\right)}{\left(\phi^{*g}, \frac{1}{v^g} \varphi^g\right)} \tag{11}$$

$$\frac{\bar{\beta}}{\Lambda} = \sum_{i=1}^{I} \frac{\bar{\beta}_i}{\Lambda} = \frac{1}{k_{eff}} \sum_{i=1}^{I} \sum_{g=1}^{G} \frac{\left(\phi^{*g}, \beta_i \nu^g \Sigma_f^g \varphi^g\right)}{\left(\phi^{*g}, \frac{1}{v^g} \varphi^g\right)} \tag{12}$$

$$\bar{\lambda}_i = \sum_{g=1}^{G} \frac{\left(\phi^{*g}, \chi_{d,i}^g \lambda_i C_i\right)}{\left(\phi^{*g}, \chi_{d,i}^g C_i\right)} \tag{13}$$

Coupling

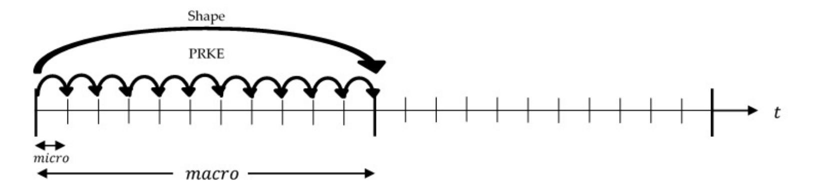
Factorization leads to a nonlinear system

The amplitude and shape equations form a system of nonlinear coupled equations:

1. the coefficients appearing in the PRKEs depend upon the shape solution,
2. the shape equation has a kernel dependent on amplitude and its derivative,
3. the delayed neutron source term is scaled by the amplitude.

Time scales and IQS method solution process

Because solving for the shape can be expensive, especially in two or three dimensions, it is attractive to make the assumption that the shape is weakly time-dependent so the shape can be computed after a multitude of PRKE calculations which is the root of IQS:



Convergence criteria

Ideally

The normalization constant should not change

$$K_0 = \sum_{g=1}^{G} \left(\phi^{*g}, \frac{1}{v^g} \varphi^g(t = 0) \right) \tag{14}$$

So we use

$$\left| \frac{\sum_{g=1}^{G} \left(\phi^{*g}, \frac{1}{v^g} \varphi^g(t = t^{n+1}) \right)}{K_0} - 1 \right| = \left| \frac{K_{n+1}}{K_0} - 1 \right| < tol \tag{15}$$

We have seen in practice

$$\frac{\| \varphi^{\ell+1}_{g,t_{n+1}} - \varphi^{\ell}_{g,t_{n+1}} \|}{\| \varphi^{\ell+1}_{g,t_{n+1}} \|} < tol \quad \text{or even} \quad \frac{\| \varphi^{\ell+1}_{g,t_{n+1}} - \varphi^{0}_{g,t_{n+1}} \|}{\| \varphi^{0}_{g,t_{n+1}} \|} < tol \tag{16}$$

where ℓ =IQS iteration index over a given macro time step $[t_n, t_{n+1}]$
followed by a renormalization before starting the next time step $[t_{n+1}, t_{n+2}]$

$$\varphi^{converged}_{g,t_{n+1}} \leftarrow \varphi^{converged}_{g,t_{n+1}} \times \frac{K_{n+1}}{K_0} \tag{17}$$

Theta-scheme

Theta-scheme

$$C^{n+1} = \frac{1-(1-\theta)\Delta t\lambda}{1+\theta\Delta t\lambda}C^n + \frac{(1-\theta)\Delta t\beta(\nu\Sigma_f)^n}{1+\theta\Delta t\lambda}\varphi^n p^n + \frac{\theta\Delta t\beta(\nu\Sigma_f)^{n+1}}{1+\theta\Delta t\lambda}\varphi^{n+1}p^{n+1} \quad (18)$$

Reporting this value of C^{n+1}, one can solve for the shape φ^{n+1} as a function of φ^n and C^n (and p^n, p^{n+1}, $dp/dt|_n$ and $dp/dt|_{n+1}$).

Once φ^{n+1} has been determined, C^{n+1} is updated.

YAK currently implements both implicit and Crank-Nicholson as options for precursor evaluation.

Analytical Integration

Analytical Integration

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

Assuming a linear in time variation over the macro time step $[t_n, t_{n+1}]$ for the shape and the fission cross section, we get:

$$C^{n+1} = C^n e^{-\lambda\Delta t} + \left(a_3(\beta\nu\Sigma_f)^{n+1} + a_2(\beta\nu\Sigma_f)^n\right)\varphi^{n+1} + \left(a_2(\beta\nu\Sigma_f)^{n+1} + a_1(\beta\nu\Sigma_f)^n\right)\varphi^n \tag{20}$$

Where the integration coefficients are defined as:

$$a_1 = \int_{t_n}^{t_{n+1}} \left(\frac{t_{n+1}-t'}{\Delta t}\right)^2 p(t')e^{-\lambda(t_{n+1}-t')}dt' \tag{21}$$

$$a_2 = \int_{t_n}^{t_{n+1}} \frac{(t'-t_n)(t_{n+1}-t')}{(\Delta t)^2} p(t')e^{-\lambda(t_{n+1}-t')}dt' \tag{22}$$

$$a_3 = \int_{t_n}^{t_{n+1}} \left(\frac{t'-t_n}{\Delta t}\right)^2 p(t')e^{-\lambda(t_{n+1}-t')}dt' \tag{23}$$

Analytical Integration (continued)

Recall that we just need to compute coefficients of the form:

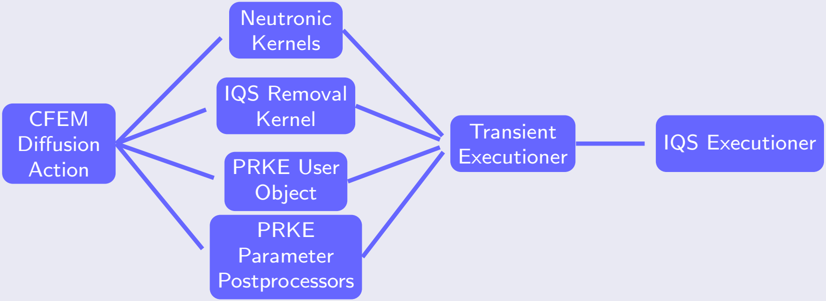
$$\int_{t_n}^{t_{n+1}} \left(\delta_2 t'^2 + \delta_1 t' + \delta_0\right) p(t') e^{-\lambda(t_{n+1}-t')} dt' \tag{24}$$

The amplitude (p) is contained in the a_i's integration coefficients.
$p(t)$ has been highly accurately calculated at the micro time step level.

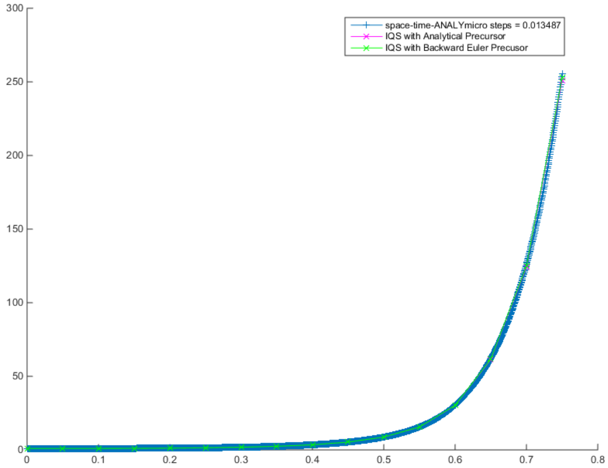
Changes to MOOSE

- Diffusion Action System (adding IQS as an option to solve CFEM diffusion, collecting and initializing variables for postprocessor computation, and rearranging hierarchy of classes)
- postprocessors to compute ρ, $\bar{\beta}$, $\bar{\lambda}$ numerator and denominator (created four postprocessors that use element integrals to compute PRKE parameters)
- IQS userobject (combining postprocessor computed values for PRKE solve using updated reactivity coefficients, ρ, $\bar{\beta}$, $\bar{\lambda}$)
- IQS executioner (use of the existing Picard iteration loop of the executioner; this can seamlessly enable IQS in multiphysics simulations without any further changes)

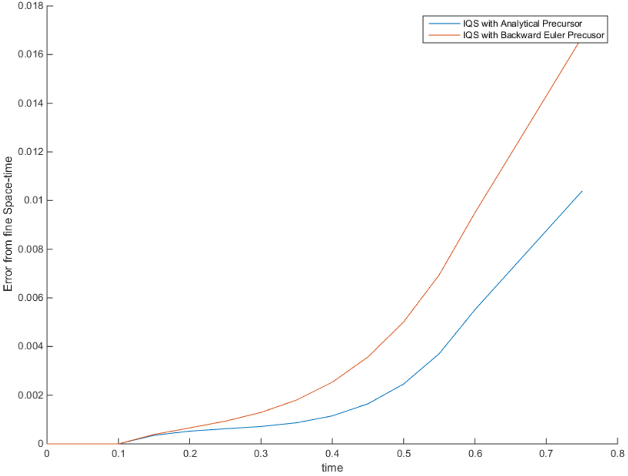
Break down of IQS Action



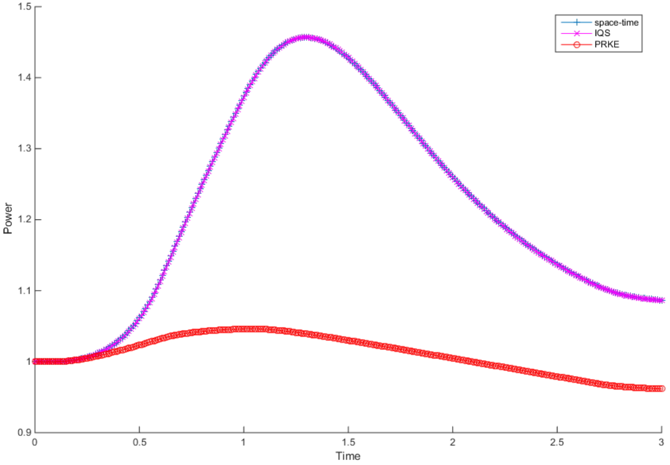
Analytical precursors versus time-discrete precursors



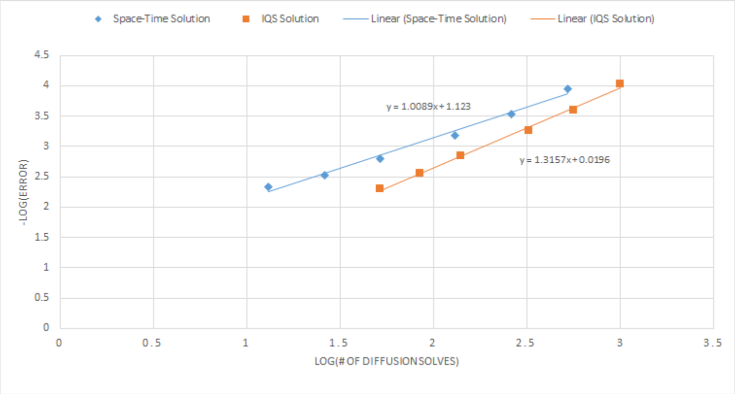
Analytical precursors versus time-discrete precursors



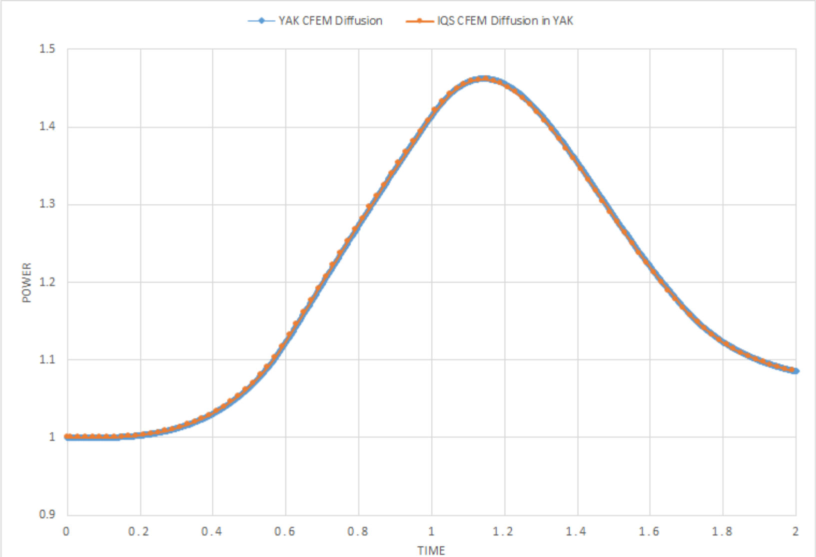
1D Heterogeneous test case



1D Heterogeneous test case



1D Heterogeneous test case with YAK



Conclusion and Outlook

Completed

- Theoretical understanding of IQS convergence and selection of proper convergence criteria
- 1D prototype Matlab code for MOOSE comparison/verification
- IQS userobject and executioner (using Picard iterations)
- IQS for CFEM Diffusion action system

In progress

- Implementation of analytical precursor integration in YAK
- Further YAK verification
- YAK documentation

Next Steps

- DFEM Diffusion action system
- DFEM SN Transport action system
- Kinetics benchmarks (neutronics only, e.g., TWIGL, LMW)
- Dynamics benchmarks (with feedback, e.g., the LRA test case)