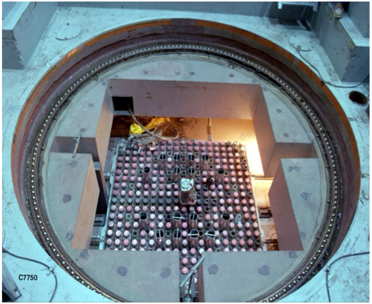


TREAT Mission Supporting Problem
Improved Quasi-Static Method in RattleSnake

Jean C. Ragusa, Zach M. Prince

Department of Nuclear Engineering, Texas A&M University, College Station, TX

November 30, 2015

email: jean.ragusa@tamu.edu

Outline

Time-dependent Multigroup Transport

Fluxes $(1 \le g \le G)$

$$\frac{1}{v^g}\frac{\partial \Psi^g(\vec{r},\vec{\Omega},t)}{\partial t} = \sum_{g'=1}^{G}\int_{4\pi} d\Omega' \left[\Sigma_s^{g'\to g}(\vec{r},\vec{\Omega}'\cdot\vec{\Omega},t) + \frac{\chi_p^g}{4\pi}\nu_p\Sigma_f^{g'}(\vec{r},t)\right]\Psi^{g'}(\vec{r},\vec{\Omega}',t)$$

$$- div\left[\vec{\Omega}\Psi^g(\vec{r},\vec{\Omega},t)\right] - \Sigma^g(\vec{r},t)\Psi^g(\vec{r},\vec{\Omega},t) + \sum_{i=1}^{I}\frac{\chi_{d,i}^g(\vec{r})}{4\pi}\lambda_i C_i(\vec{r},t)$$

+ IC + BC

In operator notation:

$$\boxed{\frac{1}{v}\frac{\partial \Psi}{\partial t} = (H + P_p - L)\Psi + S_d}$$

Precursors C_i $(1 \le i \le I)$

$$\boxed{\frac{dC_i}{dt} = \sum_{g=1}^{G}\nu_{d,i}^g\Sigma_f^g(\vec{r},t)\Phi^g(\vec{r},t) - \lambda_i(\vec{r},t)C_i(\vec{r},t)}$$

Flux Factorization

Factorization

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

$$\boxed{\Psi^g(\vec{r},\vec{\Omega},t) = p(t)\psi^g(\vec{r},\vec{\Omega},t)}$$

and, for the scalar flux,

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

with, obviously,

$$\varphi^g(\vec{r},t) = \int_{4\pi} d\Omega' \, \psi^g(\vec{r},\vec{\Omega}',t)$$

- Factorization is not an approximation
- When reporting these in the previous equations, one obtains to the so-called shape equations.
- Note that factorization is not unique:

$$\Psi = p \times \psi = \frac{p}{a} \times (a\psi)$$

Shape equations

Shape equations

The shape equations are similar to the original transport equations:

$$\boxed{\frac{1}{v}\frac{\partial\psi}{\partial t} = (H + P_p - \tilde{L})\psi + \frac{1}{p}S_d}$$

$$\boxed{\frac{dC_i}{dt} = \sum_{g=1}^{G} \nu_{d,i}^{g}\Sigma_f^g(\vec{r},t)p\varphi^g(\vec{r},t) - \lambda_i(\vec{r},t)C_i(\vec{r},t)}$$

Differences with original transport equation

1. An additional removal term based on $\frac{1}{v^g}\frac{1}{p}\frac{dp}{dt}\psi^g$

$$\boxed{\tilde{L}^g = L^g + \frac{1}{v^g}\frac{1}{p}\frac{dp}{dt}}$$

2. Delayed neutron source term scaled by $\frac{1}{p}$
3. No change in the precursor equations but we have re-written them to show explicitly $p \times \varphi^g$

Shape equations: Implementation within the MOOSE framework

Shape equations → FEM solver + implicit time integration

$$\boxed{\frac{1}{v}\frac{\psi^{n+1}-\psi^{n}}{\Delta t}=\left(H^{n+1}+P_p^{n+1}-L^{n+1}-\frac{1}{v}\frac{1}{p^{n+1}}\left.\frac{dp}{dt}\right|_{n+1}\right)\psi^{n+1}+\frac{1}{p^{n+1}}S_d^{n+1}}$$

Modification to the original transport equation

1. An additional removal term based on $\frac{1}{v^g}\frac{1}{p}\frac{dp}{dt}\psi^g$
   → add a new kernel
2. Delayed neutron source term scaled by $\frac{1}{p}$
   → scale the delayed neutron source kernel
3. nonlinear coupling between shape variable ψ^{n+1} and amplitude variable p^{n+1}

Amplitude equations (PRKE)

Principle

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

Notation

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

$$\int_{4\pi}\int_D \Psi^{*g}(\vec{r},\vec{\Omega}) f^g(\vec{r},\vec{\Omega}) d^3r d\Omega = (\Psi^{*g}, f^g)$$

Uniqueness of the factorization

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

$$K_0 = \sum_{g=1}^{G}\left(\Psi^{*g}, \frac{1}{v^g}\psi^g\right) = constant$$

PRKE (continued)

PRKE

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

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

PRKE Coefficients

$$\frac{\rho - \bar{\beta}}{\Lambda} = \frac{(\Psi^*, (H + P_p - L)\psi)}{K_0}$$

$$\frac{\bar{\beta}}{\Lambda} = \sum_{i=1}^{I} \frac{\bar{\beta}_i}{\Lambda} = \sum_{i=1}^{I} \frac{(\Psi^*, P_{d,i}\psi)}{K_0}$$

$$\bar{\lambda}_i = \frac{(\Psi^*, \chi_{d,i}\lambda_i C_i)}{(\Psi^*, \chi_{d,i} C_i)}$$

Coupling

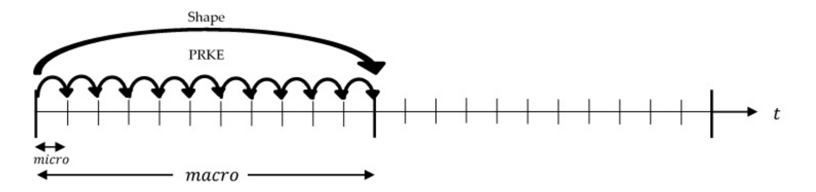
Factorization leads to a nonlinear system

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 over time !

$$K_0 = \sum_{g=1}^{G} \left(\Psi^{*g}, \frac{1}{v^g} \psi^g(t=0) \right) = constant$$

Thus, we employ $$\left| \frac{\sum_{g=1}^{G} \left(\Psi^{*g}, \frac{1}{v^g} \psi^g(t=t^{n+1}) \right)}{K_0} - 1 \right| = \left| \frac{K_{n+1}}{K_0} - 1 \right| < tol$$

Note that we have seen in practice ...

$$\frac{\left\| \psi^{g,t_{n+1}^{\ell+1}} - \psi^{g,t_{n+1}^{\ell}} \right\|}{\left\| \psi^{g,t_{n+1}^{\ell+1}} \right\|} < tol \quad \text{or even} \quad \frac{\left\| \psi^{g,t_{n+1}^{\ell+1}} - \psi^{g,t_{n+1}^{0}} \right\|}{\left\| \psi^{g,t_{n+1}^{0}} \right\|} < tol$$

where ℓ =IQS iteration index over a given macro time step $[t_n, t_{n+1}]$

These empirical criteria must be followed by a renormalization before starting the next time step $[t_{n+1}, t_{n+2}]$

$$\psi^{g,t_{n+1}^{converged}} \leftarrow \psi^{g,t_{n+1}^{converged}} \times \frac{K_{n+1}^{converged}}{K_0}$$

Precursors time-discretization

A simple ODE:

$$\boxed{\frac{dC_i}{dt} = \sum_{g=1}^{G} \nu_{d,i}^{g} \Sigma_f^g(\vec{r},t) p \varphi^g(\vec{r},t) - \lambda_i(\vec{r},t) C_i(\vec{r},t)}$$

Numerical integration: Theta-scheme (already in RattleSnake)

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

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

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

RattleSnake currently implements both implicit ($\theta = 1$) and Crank-Nicholson ($\theta = 1/2$) as options for precursor evaluation.

Analytical Integration

Analytical Integration

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

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:

$$C^{n+1} = C^n e^{-\lambda \Delta t} + \left[a_3 (\nu_d \Sigma_f)^{n+1} + a_2 (\nu_d \Sigma_f)^n\right] \varphi^{n+1} + \left[a_2 (\nu_d \Sigma_f)^{n+1} + a_1 (\nu_d \Sigma_f)^n\right] \varphi^n$$

where the integration coefficients are defined as:

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

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

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

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

Changes to RATTLESNAKE

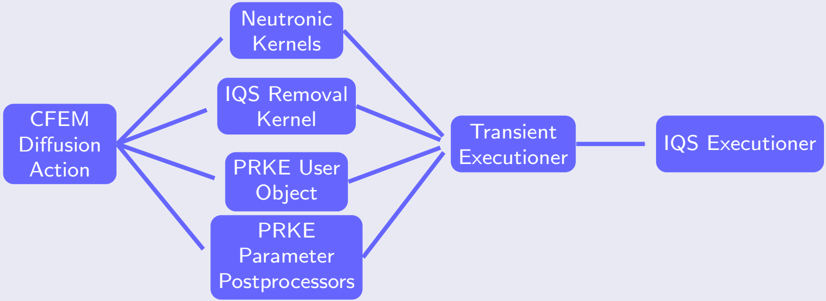
Action Systems

- Continuous FEM Diffusion (completed)
- Discontinuous FEM Diffusion
- Discontinuous FEM Sn Transport (first-order form)
- Discontinuous FEM Sn Transport (SAAF form)

Four action systems $\neq$ four times the work!

- Action System (adding IQS as an option)
- Post-processors (element integrals) for PRKE coefficients: $\rho - \bar{\beta}$, $\bar{\beta}_i$, $\bar{\lambda}_i$
  Note that the numerator of $\rho - \bar{\beta}$, i.e., $(\Psi^*, (H + P_p - L)\psi)$, is particularly easy thanks to the residual `save_in` option of MOOSE.
- IQS userobject (PRKE solve using updated PRKE coefficients)
- IQS executioner derived from MOOSE executioner (use of the existing Picard iteration loop in the `transient` executioner; this can seamlessly enable IQS in multiphysics simulations without any further changes)

Break down of IQS Changes



TWIGL benchmark

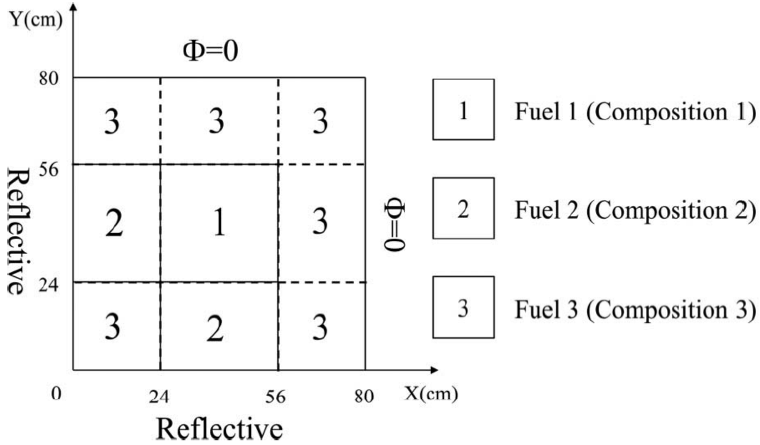
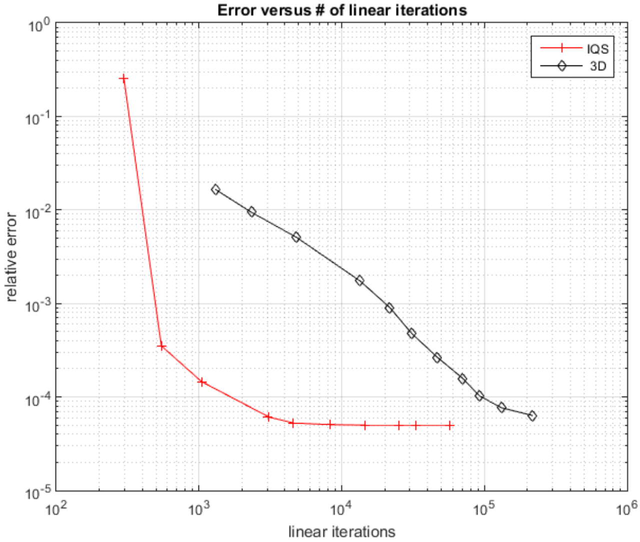


Figure 3. The TWIGL benchmark problem.

TWIGL: Computational Efficacy

Relative error versus # of linear iterations:



2-D Mach 3 flow over a forward facing step

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)
- Study of a JFNK-based algorithm to resolve the IQS nonlinearity between the amplitude/shape equations

Questions ?

Thanks

- Yaqi Wang (INL, RattleSnake code lead)
- Mark DeHart (INL, TREAT M&S lead)
- NEAMS