NERS551
Nuclear Reactor Dynamics

Numerical Solution of the Exact Point Kinetics Equations

3-8-2022

Prof T. Downar

Outline

- **Review Exact P.K. Equations (EPKE)**
  - **Derivation (See Backup Slides)**
  - **Comparison w/ “Conventional” P.K.**
- Numerical Solution of P.K. Equations
  - Theta method for temporal discretization
  - Exponential transformation
  - Precursor time integration
  - Solution of EPK
  - Linear feedback model
  - Solution of EPK with feedback
- Application to NEACRP PWR Rod Ejection

Compare Exact and Conventional Point Kinetics

- Exact Point Kinetics

$$\frac{dp(t)}{dt} = \frac{\rho(t) - \beta^{eff}(t)}{\Lambda(t)} p(t) + \frac{1}{\Lambda_0} \sum_k \lambda_k(t) \zeta_k(t)$$

$$\frac{d\zeta_k(t)}{dt} = \frac{\Lambda_0}{\Lambda(t)} \beta_k^{eff}(t) p(t) - \lambda_k(t) \zeta_k(t),\ k = 1, 2...$$

- EPK Parameters

$$\Lambda(t) = \frac{1}{p(t)F(t)} < \phi_g^*(\mathbf{r}) \frac{1}{v_g} \phi_g(\mathbf{r}, t) >$$

$$\beta_k^{eff}(t) = \frac{1}{p(t)F(t)} < \phi_g^*(\mathbf{r}) \chi_{dk,g}\left(\mathbf{r}\right) \beta_k(\mathbf{r}) S^F(\mathbf{r}, t) >,$$

$$\lambda_k(t) = \frac{< \phi_g^*(\mathbf{r}) \lambda_k(\mathbf{r}) \chi_{dk,g}\left(\mathbf{r}\right) C_k(\mathbf{r}, t) >}{< \phi_g^*(\mathbf{r}) \chi_{dk,g}\left(\mathbf{r}\right) C_k(\mathbf{r}, t) >}$$

$$\rho(t) = \frac{1}{p(t)F(t)} < \phi^*(r)\left(F - M\right)\phi(r, t) >$$

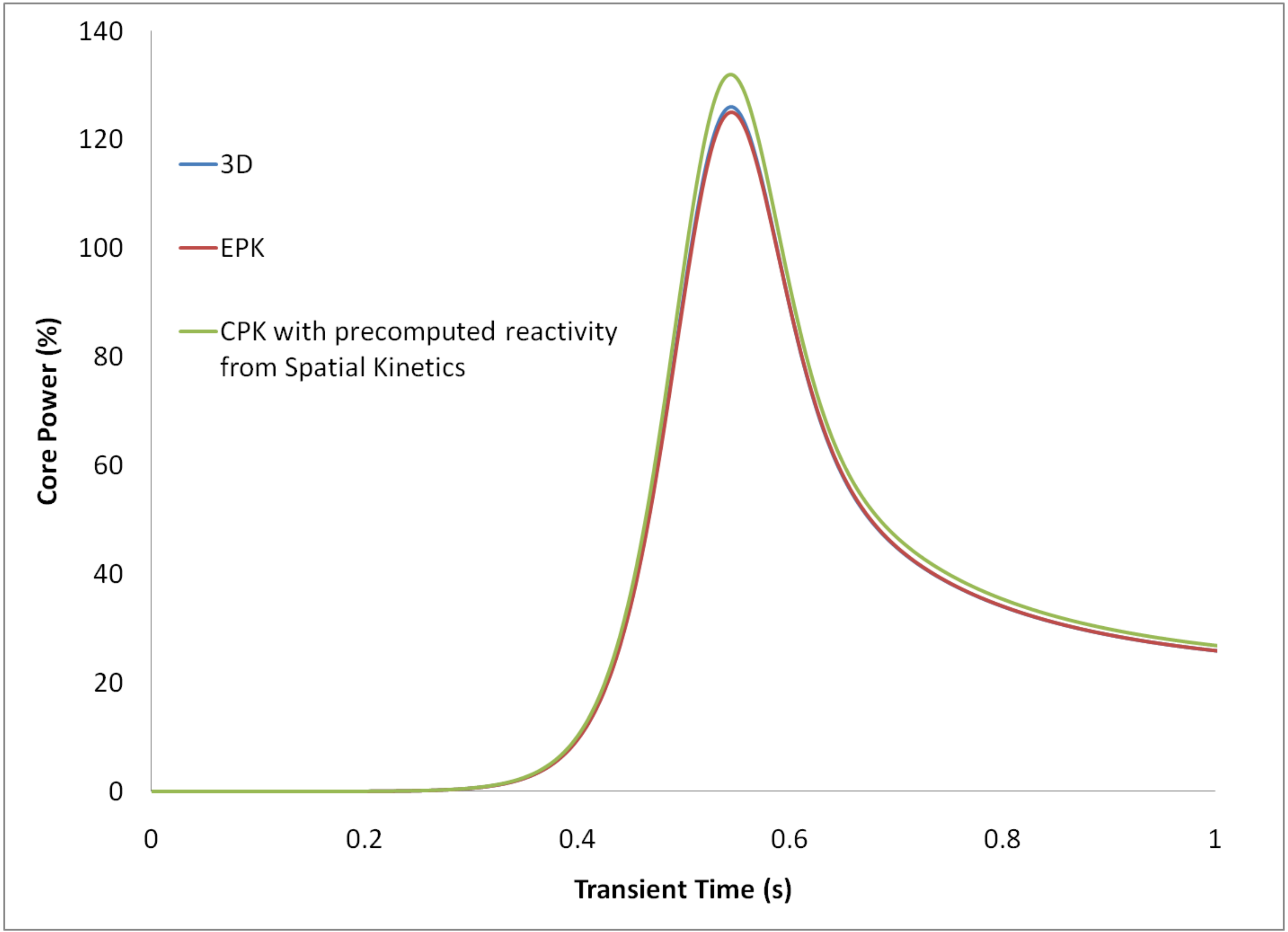
- Conventional Point Kinetics

$$\frac{dp(t)}{dt} = \frac{\rho(t) - \beta^{eff}}{\Lambda_0} p(t) + \frac{1}{\Lambda_0} \sum_k \lambda_k \zeta_k(t)$$

$$\frac{d\zeta_k(t)}{dt} = \beta_k^{eff} p(t) - \lambda_k \zeta_k(t),\ k = 1, 2...$$

- CPK Parameters
  - Initial Λ, β, λ are used for entire transient,
  - The reactivity are often evaluated with core average parameters and reactivity coefficients which were evaluated near steady state condition.
  - It is hard to obtained accurate reactivity for CPK.

EPK and CPK for Rod Eject Benchmark(A1)



Benefits from Exact Point Kinetics

- As EPK parameters rely on solutions of spatial kinetics, in order to obtain EPK solution, the spatial kinetics solution is always needed. So, what are the Benefits from EPK?

1. Help user to analyze the mechanism of transient results from spatial kinetics,
2. Identify the error sources of conventional point kinetics,
3. Accelerate the spatial kinetics solution without impacting the accuracy (e.g. improved quasi-static method).

Outline

- Review Exact P.K. Equations
  - Derivation
  - Comparison w/ “Conventional” P.K.
- **Numerical Solution of Exact P.K. Equations**
  - **Theta method for temporal discretization**
  - **Exponential transformation**
  - **Precursor time integration**
  - **Solution of EPK**
  - **Linear feedback model**
  - **Solution of EPK with feedback**
- Application to NEACRP Rod Eject

Temporal Discretization Method for Point Kinetics

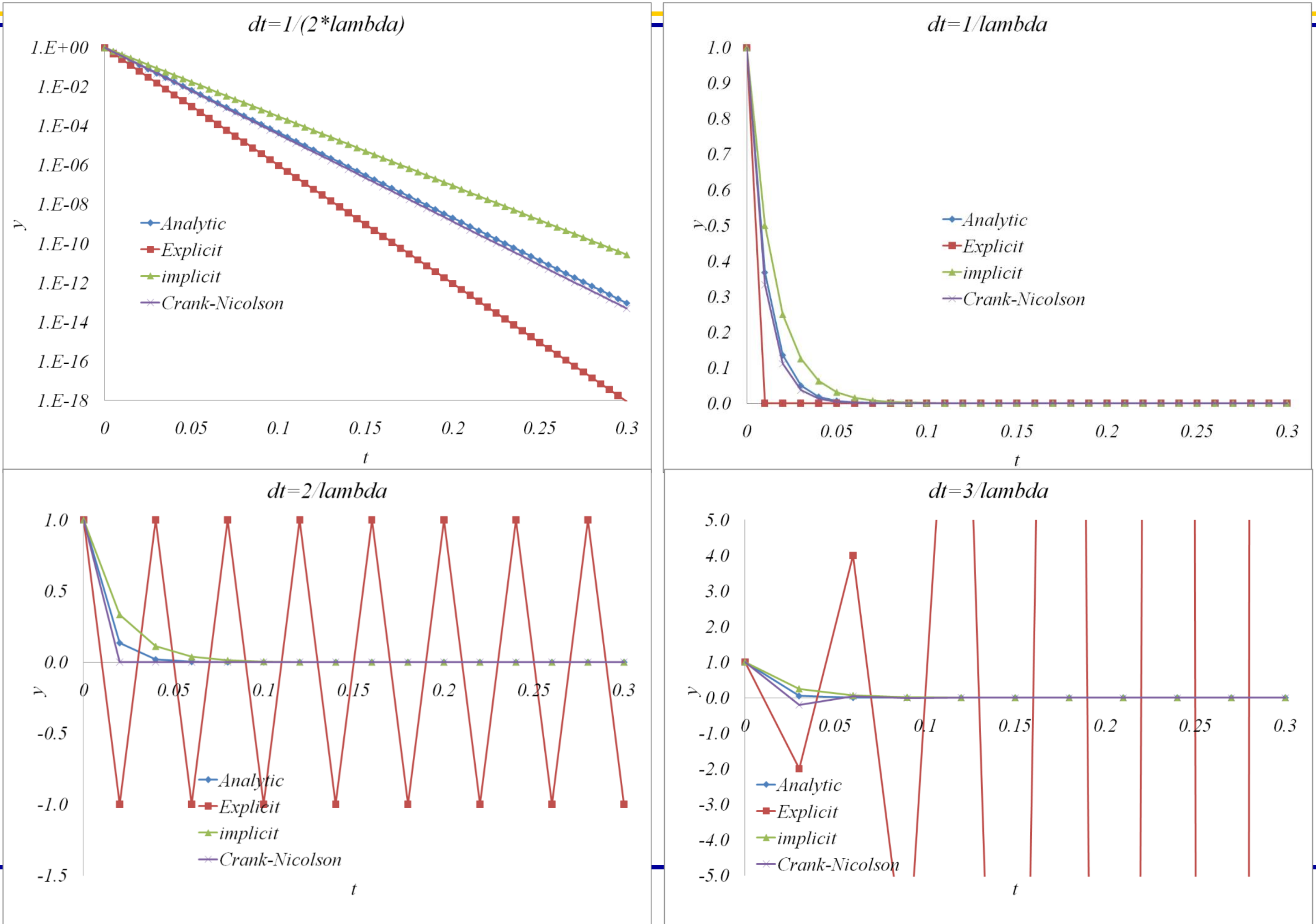
- **Theta method** is used for flux level equation. Explicit, implicit and Crank-Nicolson methods can be realized with 0, 1 and ½ as theta values.
- **Exponential transformation** is used to reduce the stiffness of point kinetic equations.
- The **precursors will be analytically** integrated over time step with linear approximation of flux level during the time step.

Theta method (Review?)

- System of ordinary differential Equations $\dfrac{dy(t)}{dt} = f\left(t, y(t)\right)$
- Theta method $y^n - \theta \Delta t^n f(t_n, y^n) = y^{n-1} - (1-\theta)\Delta t^n f(t_{n-1}, y^{n-1})$
- Test problem $\dfrac{dy(t)}{dt} = -\lambda y(t)$ $\qquad TE = FDE - PDE$
- Explicit method: $\theta = 0:\ y^n = (1-\lambda\Delta t^n)y^{n-1}, \quad TE = \dfrac{1}{2}y''(t_{n-1})\Delta t^n + \dfrac{1}{6}y'''(t_{n-1})\left(\Delta t^n\right)^2 + \dots$
- Implicit method: $\theta = 1:\ y^n = \dfrac{1}{1+\lambda\Delta t^n}y^{n-1}, \quad TE = -\dfrac{1}{2}y''(t_n)\Delta t^n + \dfrac{1}{6}y'''(t_n)\left(\Delta t^n\right)^2 + \dots$
- Crank-Nicolson: $\theta = \dfrac{1}{2}:\ y^n = \dfrac{2-\lambda\Delta t^n}{2+\lambda\Delta t^n}y^{n-1}, \quad TE = \dfrac{1}{24}y'''(t_n + \dfrac{\Delta t^n}{2})\left(\Delta t^n\right)^2 + \dots$
- Explicit and implicit methods are first order in time, Crank-Nicolson is second order in time which is more accurate for small time step size, i.e.

$$\Delta t < \left|\frac{y''(t)}{y'''(t)}\right|, \quad \textit{which is} \left|\frac{1}{\lambda}\right| \textit{ for test problem}$$

Theta method solution for test problem



Stability of Theta Method

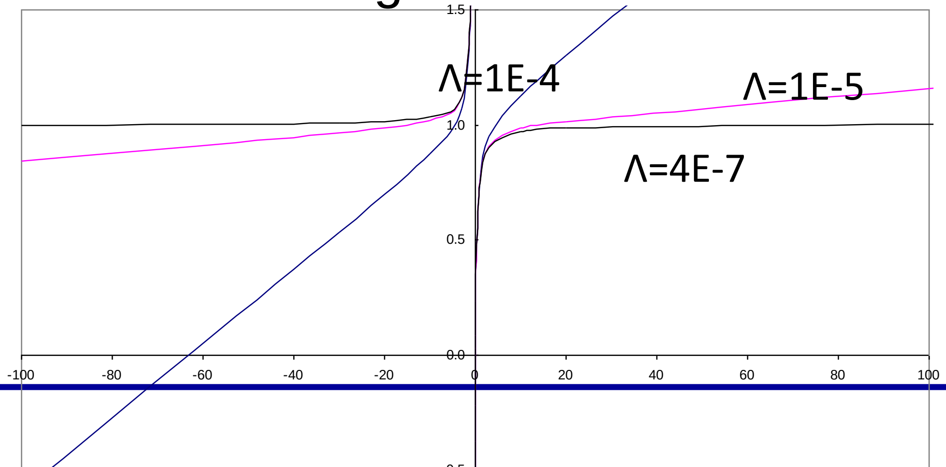
- Implicit method is unconditionally stable, Crank-Nicolson is unconditionally neural stable, explicit method is conditionally stable.

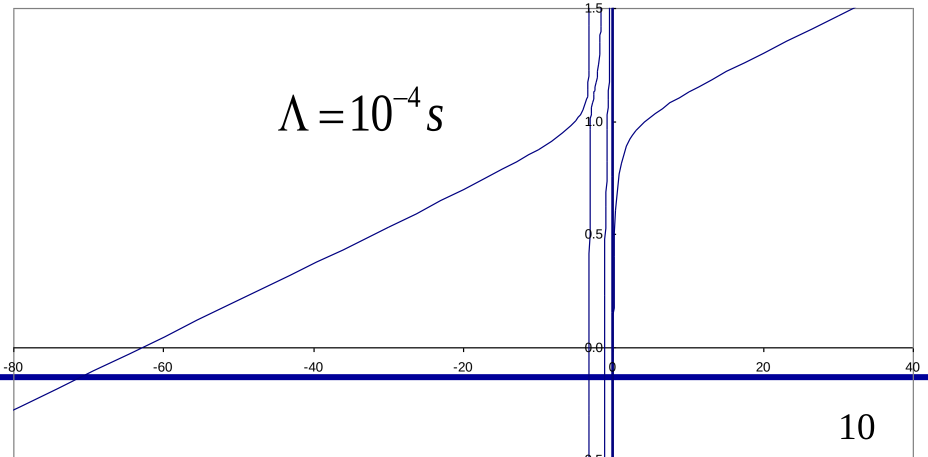
- For the test problem, explicit method is stable when $\Delta t < \left|\frac{1}{\lambda}\right|$

- λ is called the stiffness of test problem.

- For system of linear ordinary differential equations $\frac{dy(t)}{dt} = Ay(t),$ the stiffness is $\max|\lambda_i|,$ where λ_i is eigenvalues of A

- The max eigenvalues of PKE:





Theta method applied to EPKE

- Theta method for flux amplitude equation

$$e^{-\alpha^{n}\Delta t_{n}} p^{n} - p^{n-1} = \theta \Delta t_{n} e^{-\alpha^{n}\Delta t_{n}} \left(\left(\frac{\rho^{n} - \beta^{eff,n}}{\Lambda^{n}} - \alpha^{n} \right) p^{n} + \frac{1}{\Lambda_{0}} \sum_{k} \lambda_{k} \zeta_{k}^{n} \right)$$

$$+ \left(1-\theta\right) \Delta t_{n} \left(\left(\frac{\rho^{n-1} - \beta^{eff,n-1}}{\Lambda^{n-1}} - \alpha^{n} \right) p^{n-1} + \frac{1}{\Lambda_{0}} \sum_{k} \lambda_{k} \zeta_{k}^{n-1} \right) \qquad (7)$$

Where θ is user input parameter.
If θ=0, the finite differencing is explicit,
If θ=1, the finite differencing is implicit,
If θ=0.5, the finite differencing is called Crank-Nicolson scheme, which has high accuracy (2nd order)

If same method is apply to precursor equations (6), then we get K+1 equations for K+1 unknowns.

As the precursors depend only on power, and are independent from each other, we can get solution without introducing finite different error to precursor equations.

Exponential Transformation

- Test problem: $\frac{dy(t)}{dt} = -\lambda y(t)$

- Analytic Solution: $y(t) = y(0)e^{-\lambda t}$

- Stiffness: λ

- Exponential Transformation: $y(t) = \tilde{y}(t)e^{\alpha t}$

$$\frac{dy(t)}{dt} = \frac{d\tilde{y}(t)}{dt}e^{\alpha t} + \alpha\tilde{y}(t)e^{\alpha t} = -\lambda\tilde{y}(t)e^{\alpha t}$$

$$\frac{d\tilde{y}(t)}{dt} = -\left(\lambda + \alpha\right)\tilde{y}(t)$$

- Stiffness after Exponential Transformation $|\lambda + \alpha|$

- If estimated alpha is close to negative of lambda, then the stiffness is “softened”.

Exponential Transformation for PKE

- Exact Point Kinetic Equations

$$\frac{dp(t)}{dt} = \frac{\rho(t) - \beta^{eff}(t)}{\Lambda(t)} p(t) + \frac{1}{\Lambda_0} \sum_k \lambda_k \zeta_k(t), \quad (1)$$

$$\frac{d\zeta_k(t)}{dt} = \frac{\Lambda_0}{\Lambda(t)} \beta_k^{eff}(t) p(t) - \lambda_k \zeta_k(t), \quad (2)$$

- Initial Condition $\zeta_k(0) = \frac{\beta_k^{eff}(0)}{\lambda_k} p(0), \quad (3)$

- Exponential Transformation

$$p(t) = \tilde{p}(\tilde{t}) e^{\alpha_n \tilde{t}} \ , \quad \tilde{t} = t - t_{n-1} \in [-\Delta t_{n-1}, \Delta t_n], \quad (4) \qquad \alpha^n = \frac{1}{\Delta t_{n-1}} \ln \frac{p^{n-1}}{p^{n-2}}, \qquad (23)$$

$$\tilde{p}^{n-1} = p^{n-1}$$

$$\frac{d\tilde{p}(\tilde{t})}{d\tilde{t}} = \left(\frac{\rho(\tilde{t}) - \beta^{eff}(\tilde{t})}{\Lambda(\tilde{t})} - \alpha^n \right) \tilde{p}(\tilde{t}) + \frac{e^{-\alpha^n \tilde{t}}}{\Lambda_0} \sum_k \lambda_k \zeta_k(\tilde{t}), \quad (5)$$

$$\frac{d}{d\tilde{t}} \left(\zeta_k(\tilde{t}) e^{\lambda_k \tilde{t}} \right) = e^{\lambda_k \tilde{t}} \frac{d\zeta_k(\tilde{t})}{d\tilde{t}} + \lambda_k e^{\lambda_k \tilde{t}} \zeta_k(\tilde{t}) = \frac{\Lambda_0}{\Lambda(\tilde{t})} \beta_k^{eff}(\tilde{t}) p(\tilde{t}) e^{\lambda_k \tilde{t}} \qquad (6)$$

Analytic Integration of Precursors

- **Theta method** is used for flux level equation. Explicit, implicit and Crank-Nicolson methods can be realized with 0, 1 and ½ as theta values.

- **Exponential transformation** is used to reduce the stiffness of point kinetic equations.

- The **precursors will be analytically** integrated over time step with linear approximation of flux level during the time step.

- Exact Point Kinetics

$$\frac{dp(t)}{dt} = \frac{\rho(t) - \beta^{eff}(t)}{\Lambda(t)} p(t) + \frac{1}{\Lambda_0} \sum_k \lambda_k(t)\zeta_k(t)$$

$$\frac{d\zeta_k(t)}{dt} = \frac{\Lambda_0}{\Lambda(t)} \beta_k^{eff}(t) p(t) - \lambda_k(t)\zeta_k(t),\ k = 1, 2...$$

Analytic Integration of Transformed Linear Precursor

$$\frac{d}{d\tilde{t}}\left(\zeta_k(\tilde{t})e^{\lambda_k\tilde{t}}\right)=e^{\lambda_k\tilde{t}}\frac{d\zeta_k(\tilde{t})}{d\tilde{t}}+\lambda_k e^{\lambda_k\tilde{t}}\zeta_k(\tilde{t})=\frac{\Lambda_0}{\Lambda(\tilde{t})}\beta_k^{eff}(\tilde{t})p(\tilde{t})e^{\lambda_k\tilde{t}} \qquad (6)$$

- Assume $\frac{\Lambda_0}{\Lambda(\tilde{t})}\beta_k^{eff}(\tilde{t})p(\tilde{t})e^{-\alpha\tilde{t}}$ to be linear during step

$$G(\tilde{t})\equiv\frac{\Lambda_0}{\Lambda(\tilde{t})}\beta_k^{eff}(\tilde{t})p(\tilde{t})e^{-\alpha\tilde{t}}\approx G^n w+G^{n-1}(1-w) \qquad (8)$$

$$w=\tilde{t}/\Delta t_n$$

- Integrate eq. (6) over step

$$\zeta_k^n=e^{-\lambda_k\Delta t_n}\zeta_k^{n-1}+e^{\alpha\Delta t_n}\Delta t_n\left\{G^n\kappa_1\left(\tilde{\lambda}_k^n\right)+G^{n-1}\left[\kappa_0\left(\tilde{\lambda}_k^n\right)-\kappa_1\left(\tilde{\lambda}_k^n\right)\right]\right\} \qquad (9)$$

$$\tilde{\lambda}_k^n=(\lambda_k+\alpha)\Delta t_n$$

Exponential Integration

$$\kappa_m(x) = e^{-x}\int_0^1 w^m e^{xw}\,dw = \frac{1 - m\kappa_{m-1}(x)}{x} \qquad (10)$$

$|x| < 0.06$

$\kappa_0(x) = (1 - e^{-x})/x$

$$\kappa_0(x) = 1 - \frac{x}{2} + \frac{x^2}{6} - \frac{x^3}{24} + \frac{x^4}{120} - \frac{x^5}{720} + \frac{x^6}{5040} + O[x]^7 \qquad (11)$$

$\kappa_1(x) = [1 - \kappa_0(x)]/x$

$$\kappa_1(x) = \frac{1}{2} - \frac{x}{6} + \frac{x^2}{24} - \frac{x^3}{120} + \frac{x^4}{720} - \frac{x^5}{5040} + \frac{x^6}{40320} + O[x]^7 \qquad (12)$$

$$\kappa_0(x) - \kappa_1(x) = \frac{1}{2} - \frac{x}{3} + \frac{x^2}{8} - \frac{x^3}{30} + \frac{x^4}{144} - \frac{x^5}{840} + \frac{x^6}{5760} + O[x]^7 \qquad (13)$$

Integrated Precursor Transformed Linear Equation

- Integrated precursor $\zeta_k^n = p^n \Omega_k^n + \hat{\zeta}_k^n,$ (14)

$$\Omega_k^n = \frac{\Lambda^0}{\Lambda^n} \beta_k^{eff,n} \Delta t_n \kappa_1 \left(\tilde{\lambda}_k^n \right) \quad (15)$$

$$\hat{\zeta}_k^{n-1} = e^{-\lambda_k \Delta t_n} \zeta_k^{n-1} + e^{\alpha \Delta t_n} \Delta t_n G^{n-1} [\kappa_0 (\tilde{\lambda}_k^n) - \kappa_1 (\tilde{\lambda}_k^n)] \quad (16)$$

- Delay Neutron terms

$$\sum_k \lambda_k \zeta_k^n = p^n \sum_k \lambda_k \Omega_k^n + \sum_k \lambda_k \hat{\zeta}_k^{n-1} = \tau^n p^n + \hat{S}_d^n \quad (17)$$

$$\tau^n = \sum_k \lambda_k \Omega_k^n, \qquad \hat{S}_d^n = \sum_k \lambda_k \hat{\zeta}_k^n, \quad (18)$$

$$S_d^{n-1} = \sum_k \lambda_k \zeta_k^{n-1}, \quad (19)$$

Solution of EPKE

- Theta method for flux level equation

$$e^{-\alpha^n \Delta t_n} p^n - p^{n-1} = \theta \Delta t_n e^{-\alpha^n \Delta t_n} \left(\left(\frac{\rho^n - \beta^{eff,n}}{\Lambda^n} - \alpha^n \right) p^n + \frac{\tau^n}{\Lambda_0} p^n + \frac{1}{\Lambda_0} \hat{S}_d^n \right)$$

$$+ (1-\theta)\Delta t_n \left(\left(\frac{\rho^{n-1} - \beta^{eff,n-1}}{\Lambda^{n-1}} - \alpha^n \right) p^{n-1} + \frac{S_d^{n-1}}{\Lambda_0} \right) \qquad (20)$$

- Flux Level Solution

$$p^n = \frac{e^{\alpha^n \Delta t_n} \Bigg(p^{n-1} + (1-\theta)\Delta t_n \Bigg(\Bigg(\dfrac{\rho^{n-1} - \beta^{eff,n-1}}{\Lambda^{n-1}} - \alpha^n \Bigg) p^{n-1} + \dfrac{S_d^{n-1}}{\Lambda_0} \Bigg) \Bigg) \Bigg) + \theta \Delta t_n \dfrac{\hat{S}_d^n}{\Lambda_0}}{1 - \theta \Delta t_n \left(\dfrac{\rho^n - \beta^{eff,n}}{\Lambda^n} - \alpha^n + \dfrac{\tau^n}{\Lambda_0} \right)} \quad (21)$$

- Fission Power

$$H^n = f_{fp} p^n, \qquad (22)$$

Exponential transformation parameter

- First take parameter, $\alpha^n = \frac{1}{\Delta t_{n-1}} \ln \frac{p^{n-1}}{p^{n-2}},$ (23)

- Evaluate the power.

- The transformed solution is closer to linear than original solution, and the transformation parameter is accepted, if

$$\left| p^n - e^{\alpha^n \Delta t_n} p^{n-1} \right| \le \left| p^n - p^{n-1} - (p^{n-1} - p^{n-2}) / \gamma \right|, \qquad (24)$$

- Otherwise re-evaluate solution without transformation, i.e.

$$\alpha^n = 0.$$

Outline

- Review Exact P.K. Equations
  - Derivation (See Backup Slides)
  - Comparison w/ “Conventional” P.K.
- Numerical Solution of P.K. Equations
  - Theta method for temporal discretization
  - Exponential transformation
  - Precursor time integration
  - Solution of EPK
  - **Linear feedback model**
  - Solution of EPK with feedback
- Application to NEACRP PWR Rod Ejection

Linear Feed Back Model

- Reactivity feed back

$$\rho(t) = \rho_{im}(t) + \gamma_d \int_0^t (H(t') - P_0)\, exp\big(\lambda_H (t' - t)\big) dt' \quad (25)$$

imposed reactivity   feed back constant   conduction time constant

- Doppler feedback $$\gamma_d = \frac{H \bullet 1s}{mc_p} \frac{d\rho}{dT_f} \qquad (26)$$

- Feed back Reactivity

$$\rho_d(t) = \rho(t) - \rho_{im}(t) = \gamma_d \int_0^t (H(t') - P_0) \exp\big(\lambda_H (t' - t)\big) dt' \quad (27)$$

Feedback Reactivity at Next Step

- Feed back Reactivity at new step

$$\rho_d^n = \gamma_d \int_0^{t_n} (H(t') - P_0) \exp\big(\lambda_H (t' - t_n)\big) dt'$$

$$\hat{\lambda}_H^n = \lambda_H \Delta t_n$$

$$\rho_d^n = e^{-\widehat{\lambda}_H^n} \rho_d^{n-1} - P_0 \gamma_d \Delta t_n \kappa_0(\hat{\lambda}_H^n) + \gamma_d e^{-\widehat{\lambda}_H} \int_0^{\Delta t_n} H(\tilde{t}) \exp(\lambda_H \tilde{t}) d\tilde{t} \quad (28)$$

- Linear approximation

$$\tilde{\lambda}_H^n = (\lambda_H + \alpha) \Delta t_n$$

$$Q(\tilde{t}) = f_{fp}(\tilde{t}) p(\tilde{t}) e^{-\alpha \tilde{t}} = Q^n w + Q^{n-1}(1 - w) \qquad (29)$$

$$\int_0^{\Delta t_n} Q(\tilde{t}) \exp(\lambda_H + \alpha) \tilde{t} d\tilde{t} = e^{\widetilde{\lambda}_H^n} \Delta t_n \{Q^n \kappa_1(\tilde{\lambda}_H^n) + Q^{n-1}[\kappa_0(\tilde{\lambda}_H^n) - \kappa_1(\tilde{\lambda}_H^n)]\} \ (30)$$

$$\rho_d^n = e^{-\widehat{\lambda}_H^n} \rho_d^{n-1} - \eta P_0 \gamma_d \Delta t_n \kappa_0(\hat{\lambda}_H^n) + e^{\alpha \Delta t_n} \gamma_d \Delta t_n \{Q^n \kappa_1(\tilde{\lambda}_H^n) + Q^{n-1}[\kappa_0(\tilde{\lambda}_H^n) - \kappa_1(\tilde{\lambda}_H^n)]\}$$

Linear Expression of Next Step Reactivity

$$\rho^n = a_1 p^n + b_1 \qquad (31)$$

$$a_1 = f_{fp}^n \gamma_d \Delta t_n \kappa_1(\tilde{\lambda}_H^n) \qquad (32)$$

$$\begin{aligned} b_1 &= \rho_{im}^n + e^{-\widehat{\lambda}_H^n} \rho_d^{n-1} - P_0 \gamma_d \mathit{\Delta} t_n \kappa_0\left(\hat{\lambda}_H^n\right) \\ &+ e^{\alpha \mathit{\Delta} t_n} \gamma_d \mathit{\Delta} t_n f_{fp}^{n-1} p^{n-1} \left[\kappa_0\left(\tilde{\lambda}_H^n\right) - \kappa_1(\tilde{\lambda}_H^n)\right] \end{aligned} \qquad (33)$$

Solution with Feedback

- Substitute (31) into (20)

$$e^{-\alpha^n \Delta t_n} p^n - p^{n-1} = \theta \Delta t_n e^{-\alpha^n \Delta t_n} \left(\left(\frac{a_1 p^n + b_1 - \beta^{eff,n}}{\Lambda^n} - \alpha^n \right) p^n + \frac{\tau^n}{\Lambda_0} p^n + \frac{1}{\Lambda_0} \hat{S}_d^n \right)$$

$$+ (1-\theta) \Delta t_n \left(\left(\frac{\rho^{n-1} - \beta^{eff,n-1}}{\Lambda^{n-1}} - \alpha^n \right) p^{n-1} + \frac{S_d^{n-1}}{\Lambda_0} \right) \qquad (34)$$

- Rearrange eq. (34) $\quad a\left(p^n\right)^2 + bp^n + c = 0 \qquad (35)$

$$a = \frac{\theta \Delta t_n a_1}{\Lambda^n} \qquad (36)$$

$$b = \left[\theta \Delta t_n \left(\left(\frac{b_1 - \beta^{eff,n}}{\Lambda^n} - \alpha^n \right) + \frac{\tau^n}{\Lambda_0} \right) - 1 \right] \qquad (37)$$

$$c = \theta \Delta t_n \frac{1}{\Lambda_0} \hat{S}_d^n + e^{\alpha^n \Delta t_n} \left[(1-\theta) \Delta t_n \left(\left(\frac{\rho^{n-1} - \beta^{eff,n-1}}{\Lambda^{n-1}} - \alpha^n \right) p^{n-1} + \frac{S_d^{n-1}}{\Lambda_0} \right) + p^{n-1} \right] \qquad (38)$$

- Flux Level Solution $\quad p^n = \dfrac{-b - \sqrt{b^2 - 4ac}}{2a}, \quad if \;\; a < 0 \qquad (39)$

- No feedback or explicit: $\quad a = 0, and \;\; p^n = \dfrac{c}{-b}, \qquad (40)$

P.K. Solution Scheme

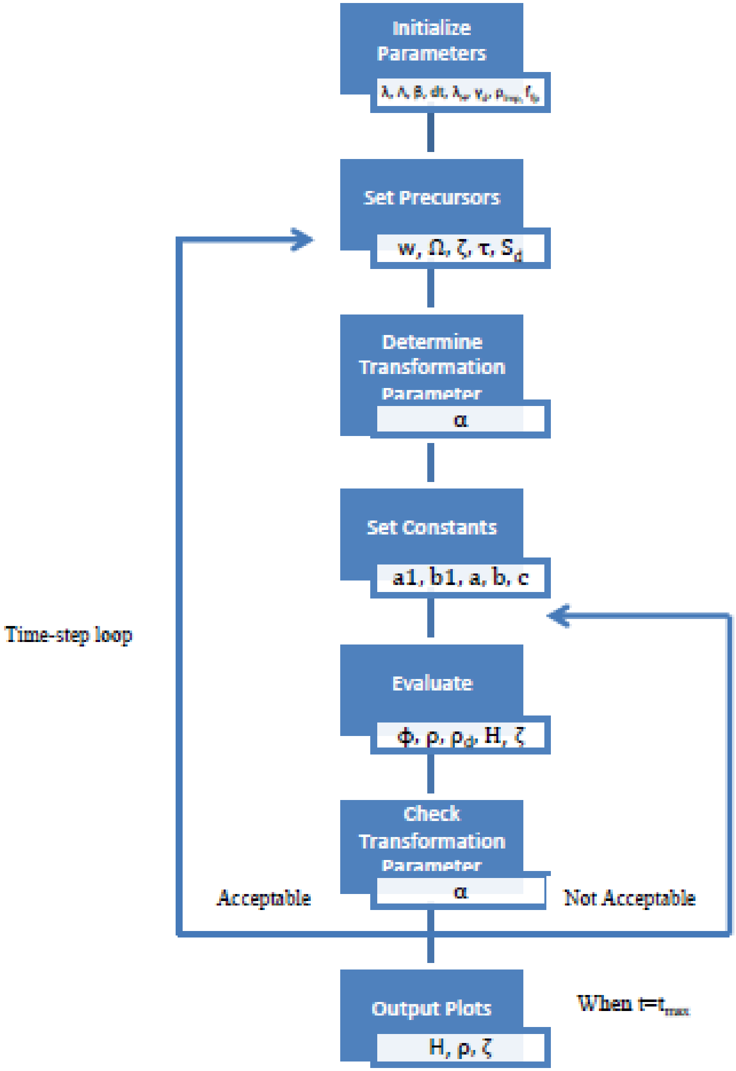
Only formula with red numbers need to be coded, the sequence of these formula in your program should be as follows:

1. Prepare Ω_k^n and $\hat{\zeta}_k^n$ with eq. (15),(16)

   the exponential integration functions (10)~(13) are called

1. Prepare $\tau^n, \hat{S}_d^n, S_d^{n-1}$ with eq. (18),(19)
2. Prepare a1,b1 with eq. (32) and (33)
3. Determine transformation parameter with eq. (23)
4. Prepare a,b,c with eq. (36), (37), and(38)
5. Evaluate flux level with eq. (39) or (40)
6. Check if exponential transformation is acceptable with eq. (24), if not, redo step 5 and 6 with $\alpha^n = 0.$
7. Evaluate power level, precursors and reactivity with eq. (22), (14) and (31)

E.P.K.E. Solution Scheme


Initialize Parameters
$\lambda, \Lambda, \beta, dt, \lambda_w, \gamma_d, \rho_{imp}, f_{fp}$
Set Precursors
$w, \Omega, \zeta, \tau, S_d$
Determine Transformation Parameter
α
Set Constants
a1, b1, a, b, c
Time-step loop
Evaluate
$\phi, \rho, \rho_d, H, \zeta$
Check Transformation Parameter
α
Acceptable
Not Acceptable
Output Plots
H, ρ, ζ
When $t=t_{max}$

Outline

- Review Exact P.K. Equations
  - Derivation
  - Comparison w/ “Conventional” P.K.
- Numerical Solution of P.K. Equations
  - Theta method for temporal discretization
  - Exponential transformation
  - Precursor time integration
  - Solution of EPK
  - Linear feedback model
  - Solution of EPK with feedback
- **Application to NEACRP Rod Eject**

Kinetics options implemented in PARCS

Transient card in CNTL block

transient (tran) (kinopt)

Kinopt, integer, kinetic option, default 0

0: 3d spatial kinetics

1: point kinetics with adjoint weighted cross section change as reactivity exclude control rod reactivity component,

2: point kinetics with adjoint weighted cross section change as reactivity,

3: Conventional point kinetics with power weighted core average parameter feedback,

4: point kinetics with adjoint fission weighted core average parameter feedback,

5: point kinetics without feedback (user input reactivity),

6: no kinetics,(user input power levels)

Impose reactivity are acceptable for option 1 to 5

Impose Reactivity for point kinetics

- IMPOSERHO cardsin TRAN block

```
 imposerho     t1   r1   t2  r2 ...
```

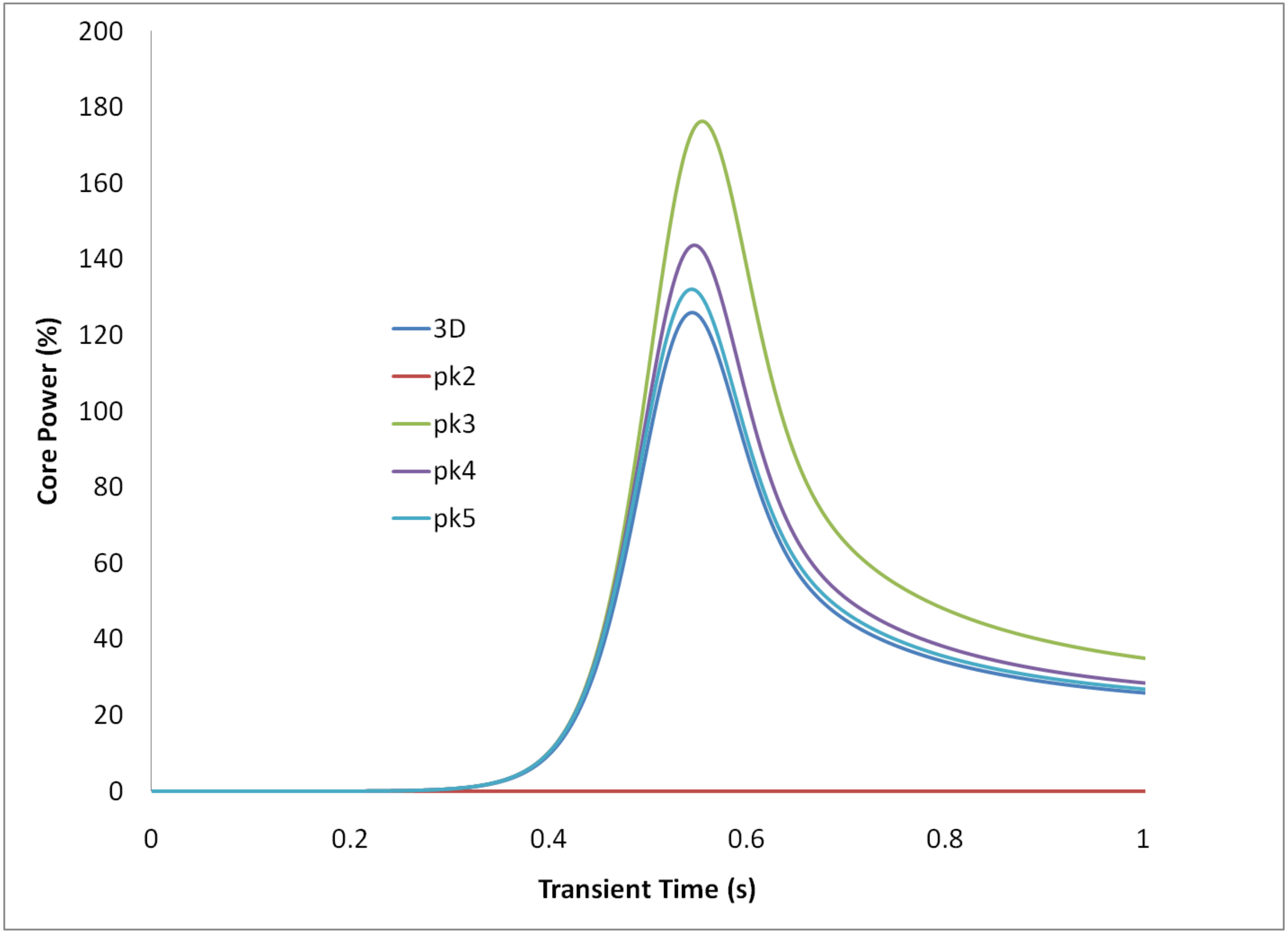
ti: real, time point for impose reactivity

ri: real, impose reactivity (or power level if kinopt=6)

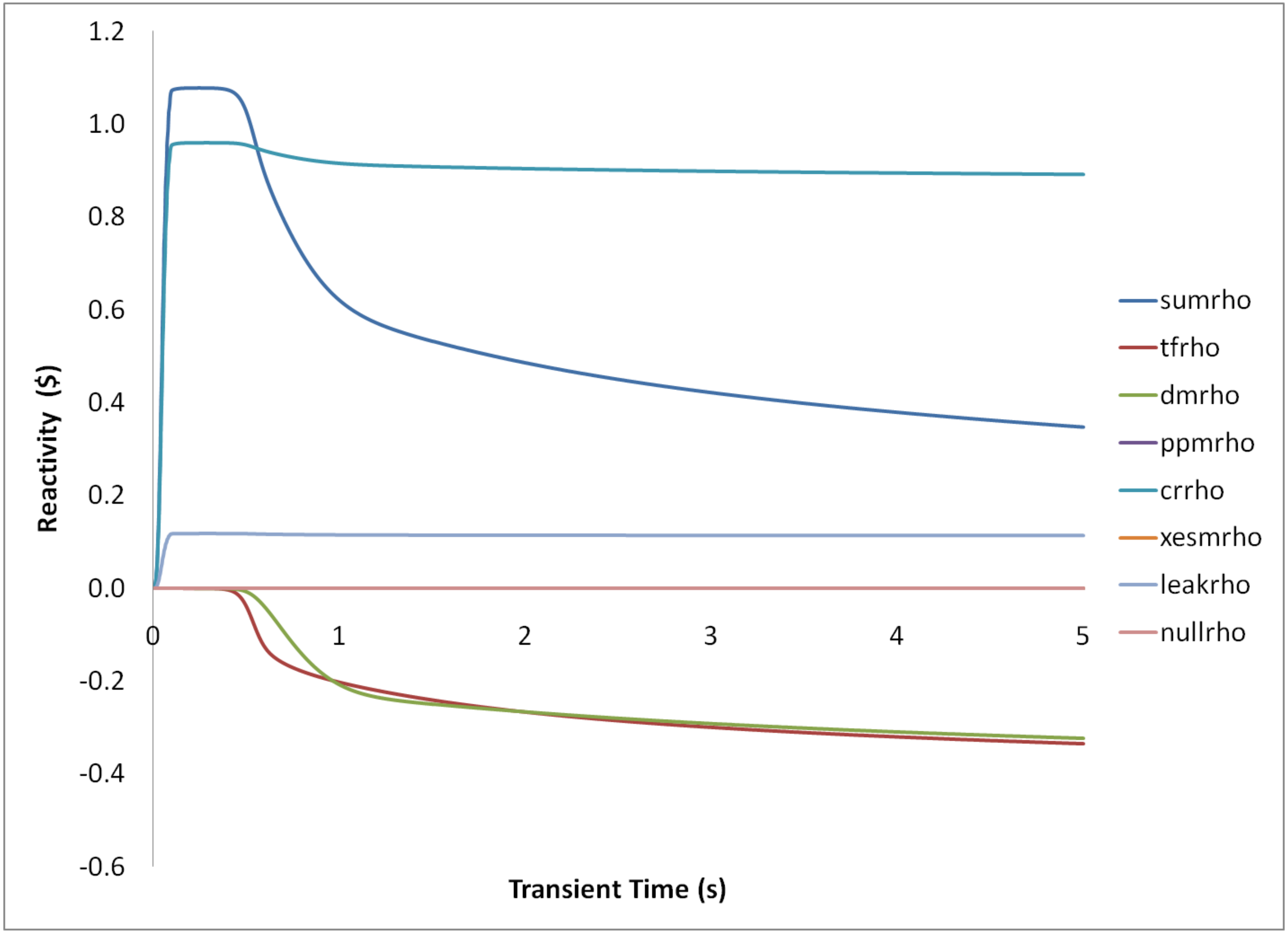
Example:

```
  imposerho   0.000000E+00   0.000000E+00
  imposerho   1.000000E-03   2.532442E-04
  imposerho   2.000000E-03   5.697343E-04
  imposerho   3.000000E-03   9.885548E-04
```

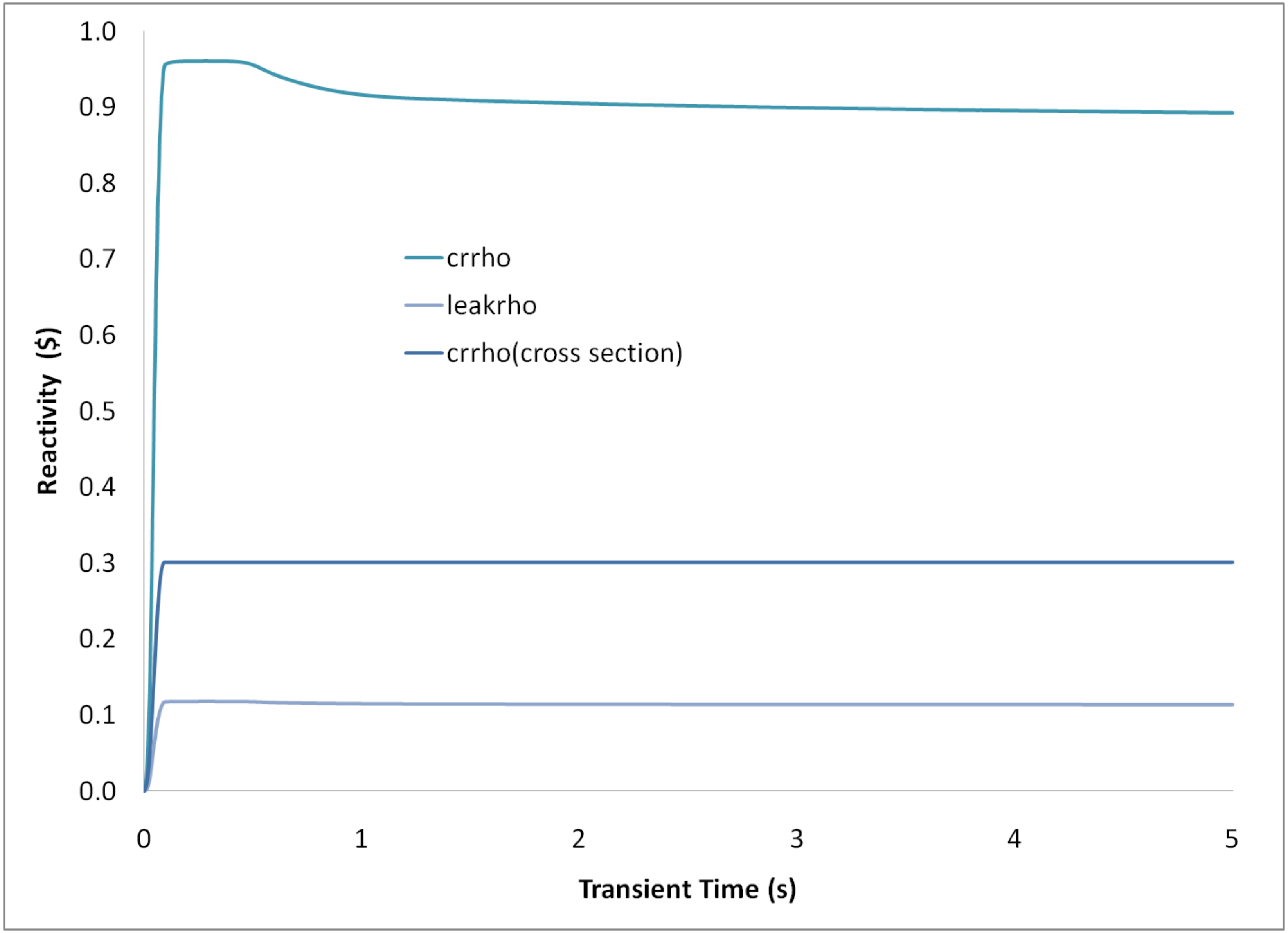
Point kinetic solutions for NEACRP-A1



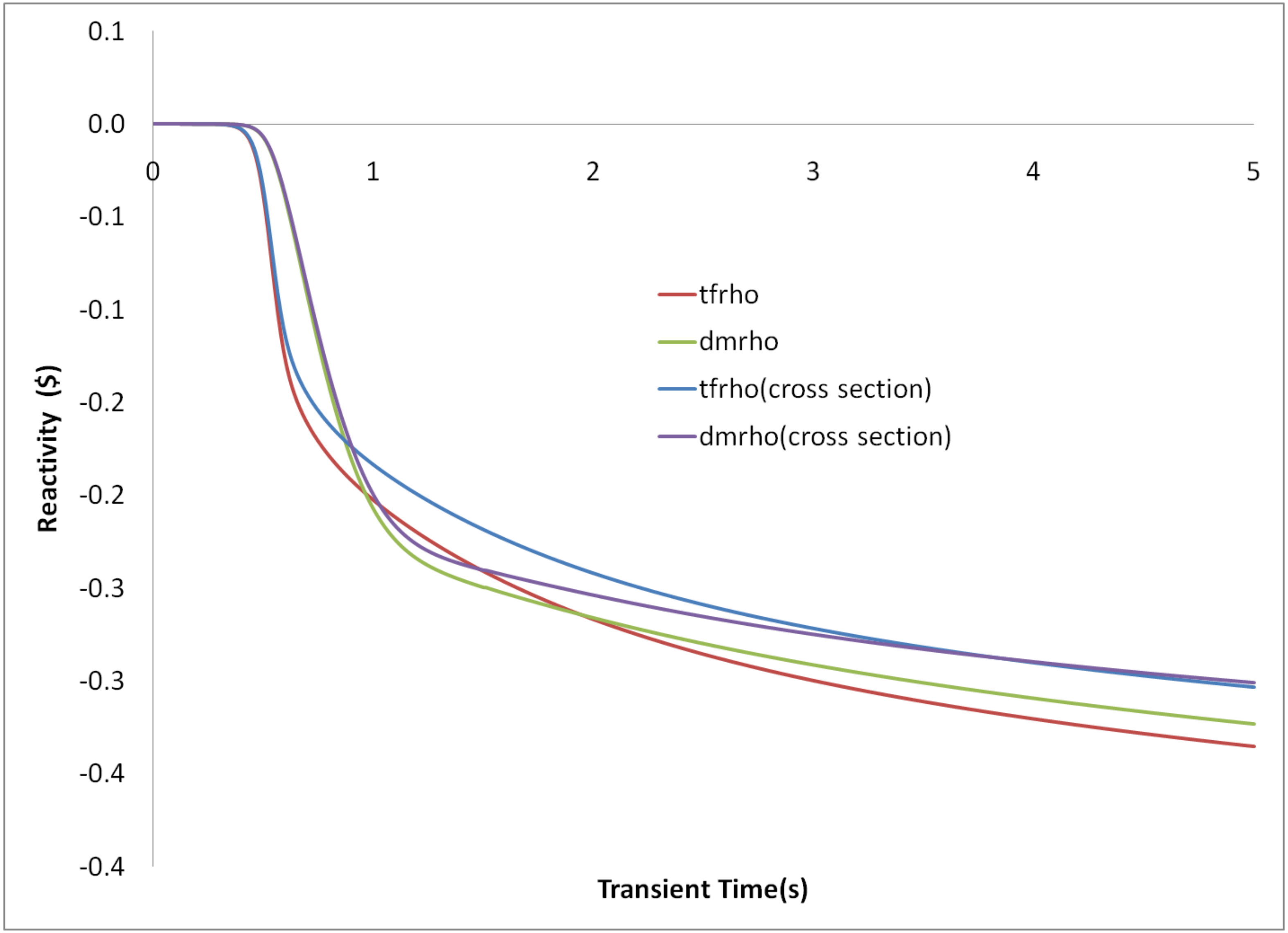
Reactivity Components from spatial kinetics



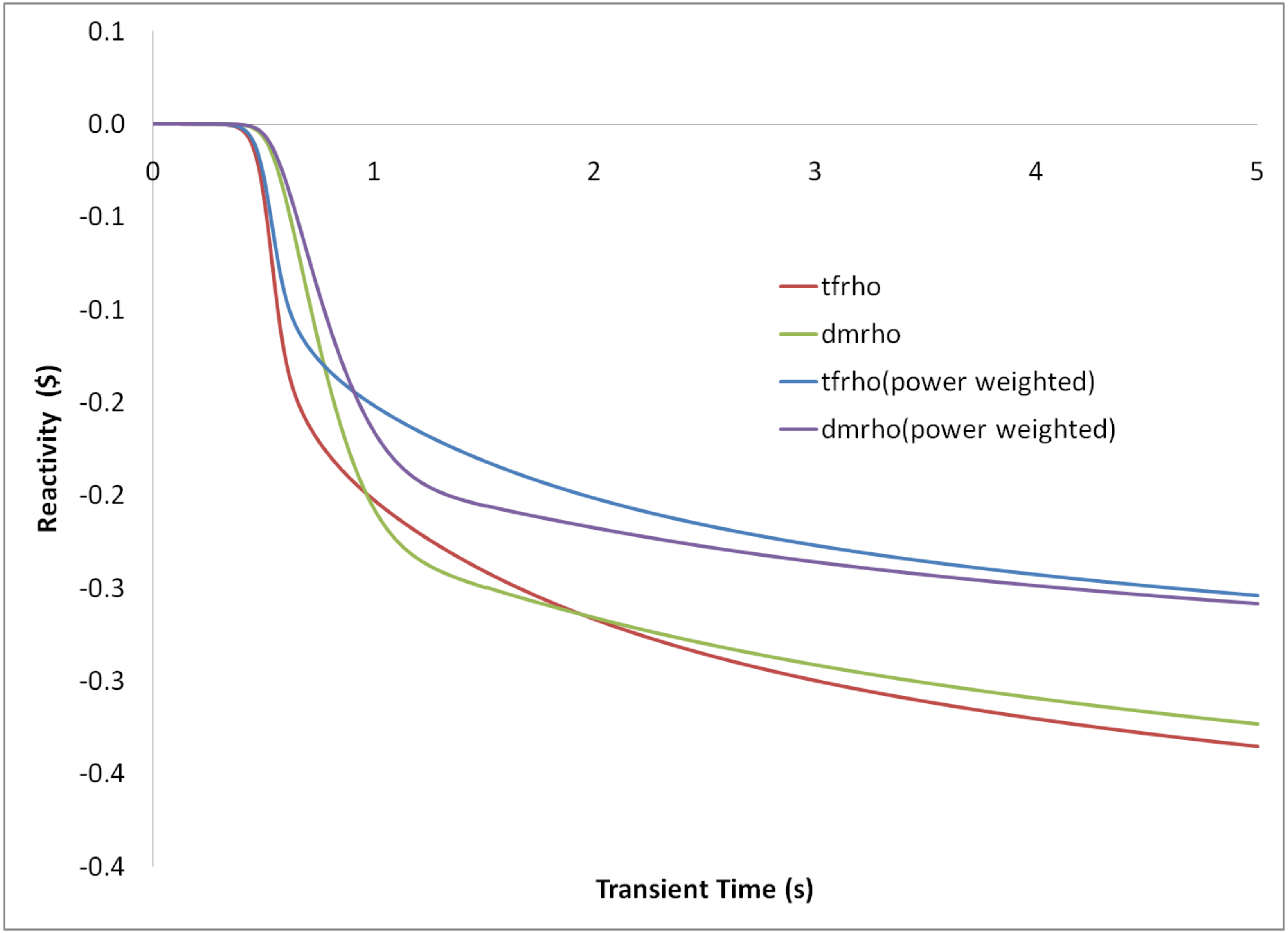
Control Rod Reactivity from Cross Section with initial flux shape



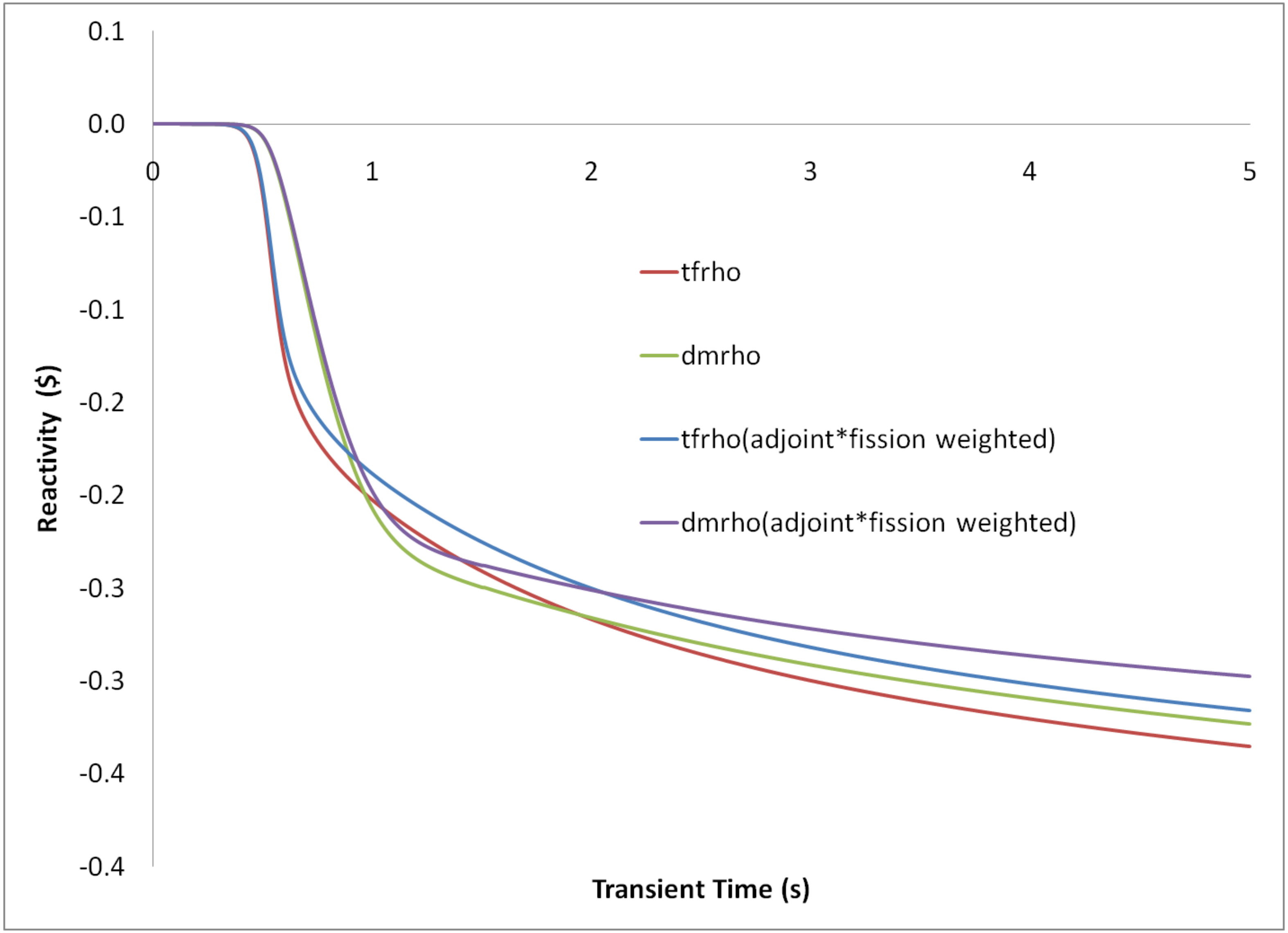
Feed back Reactivity from Cross with initial flux shape



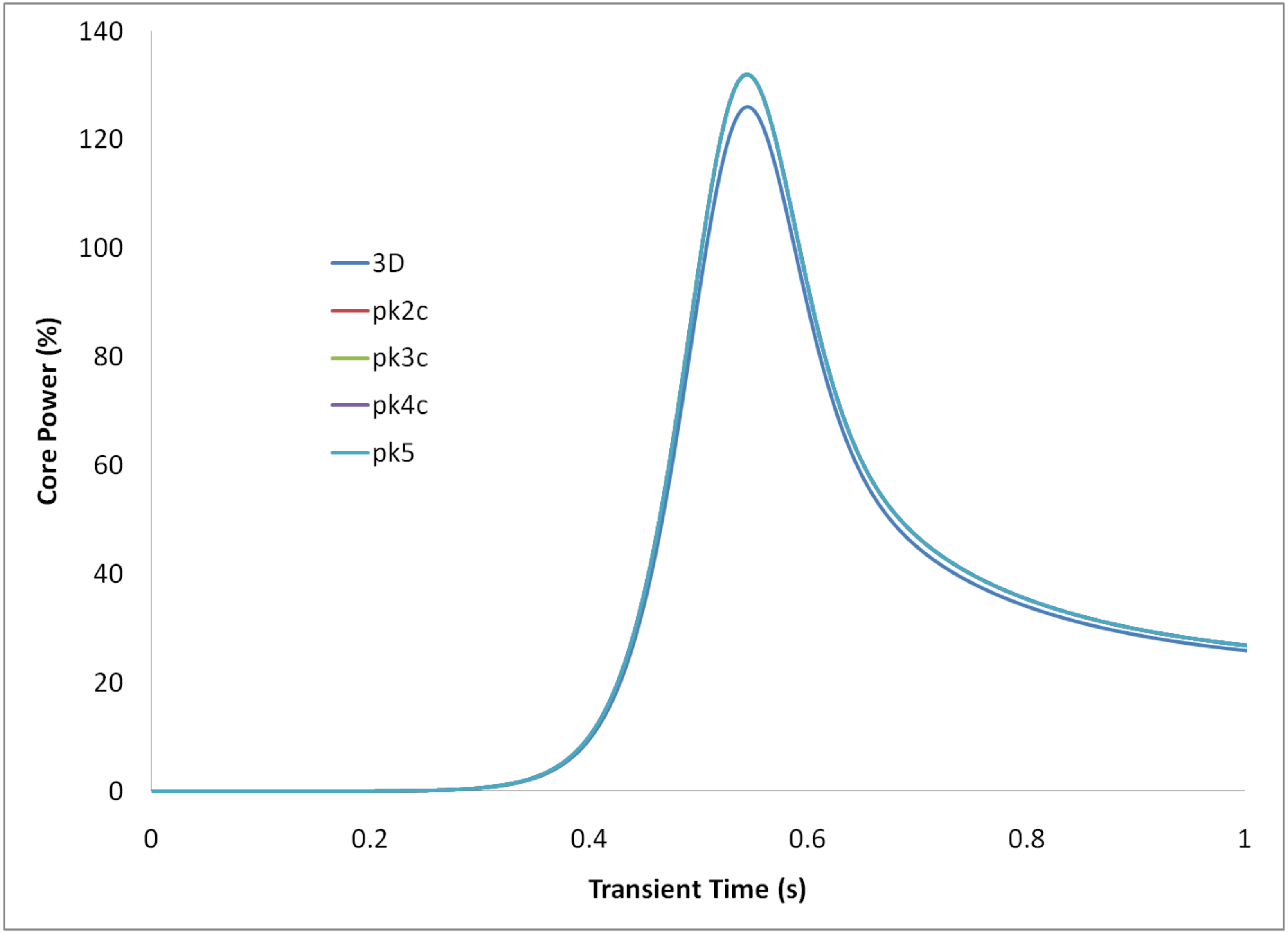
Feed back Reactivity from power weighted core average parameters



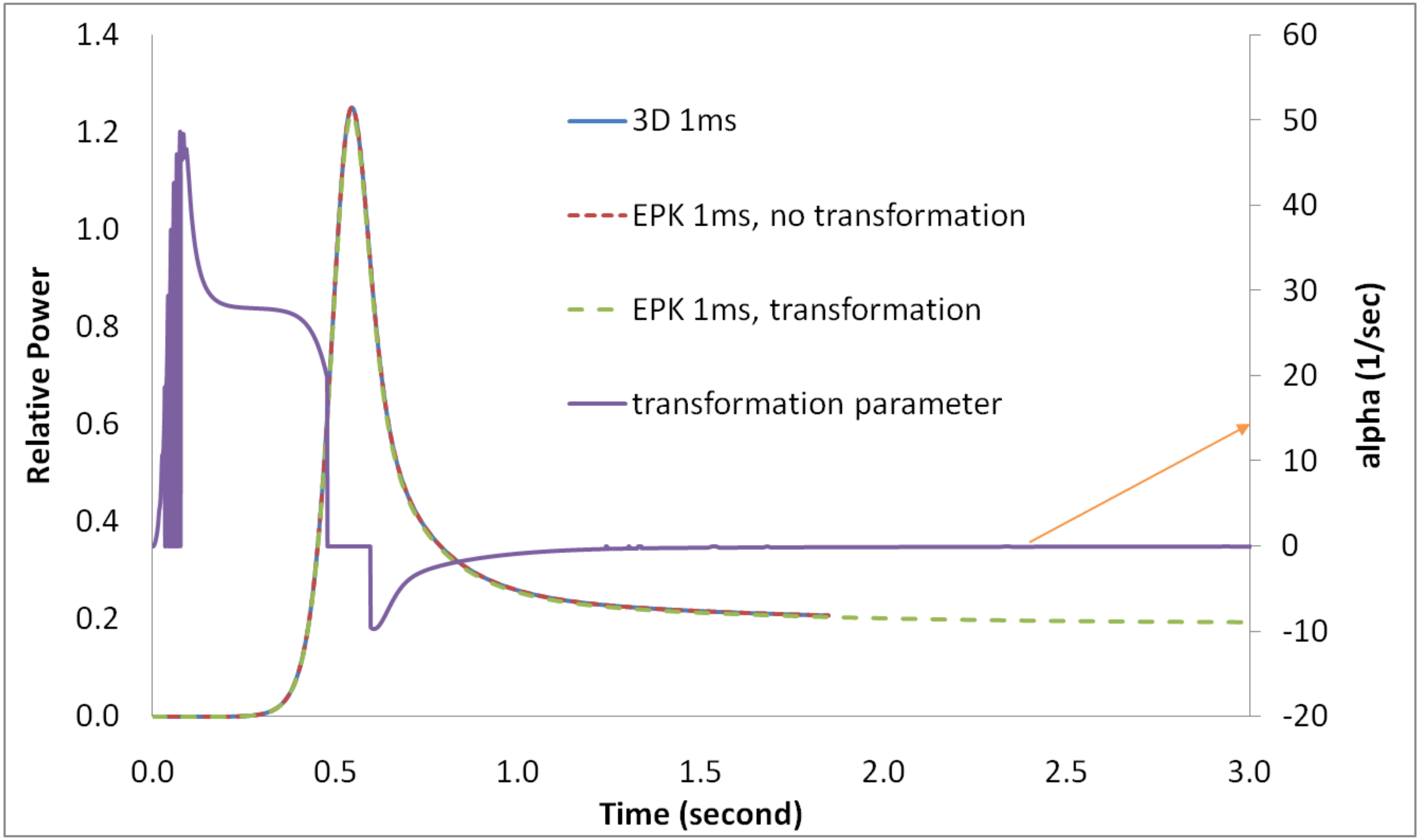
Feed back Reactivity from Adjoint*Fission weighted core average parameters



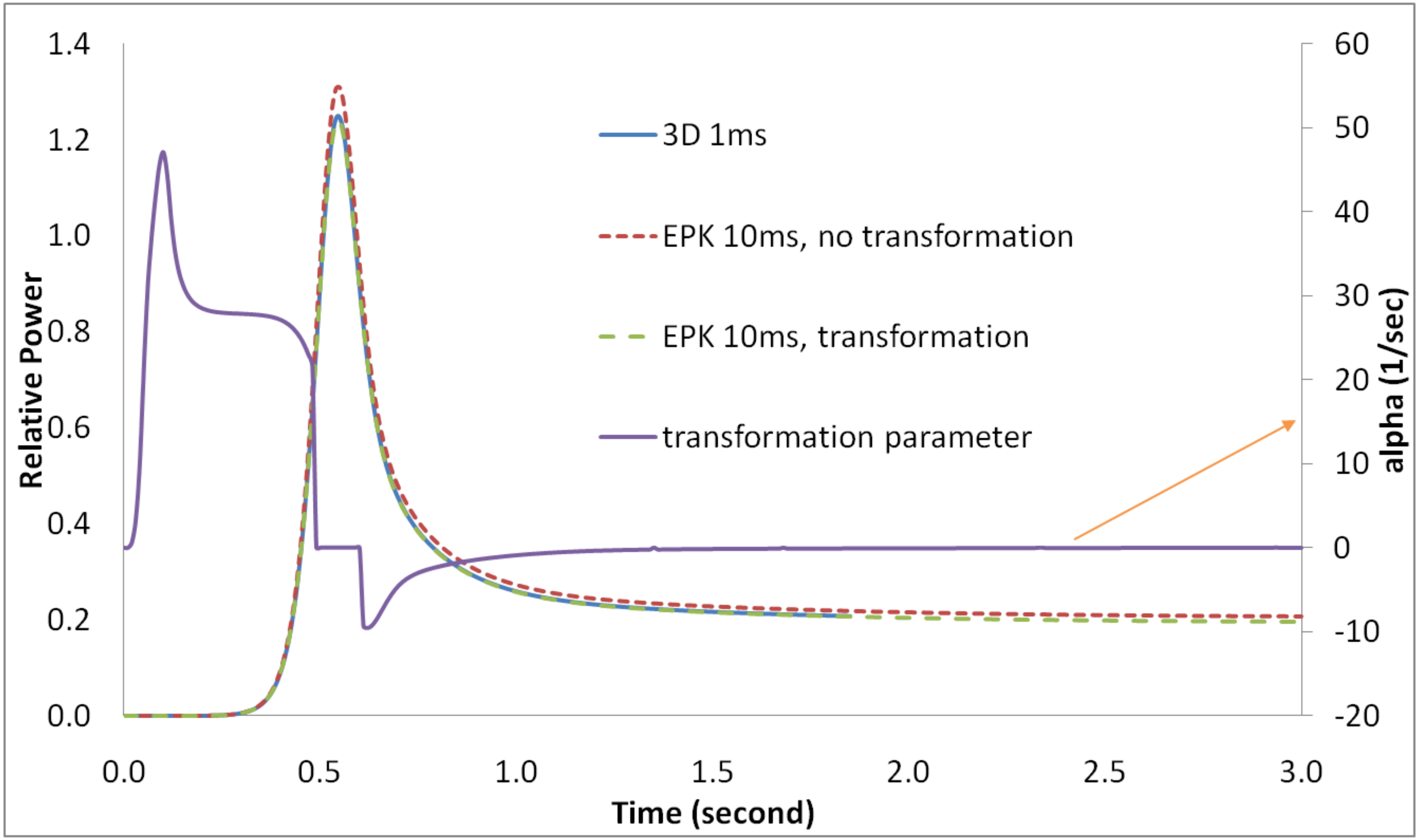
Point kinetic solutions with corrected imposed reactivity



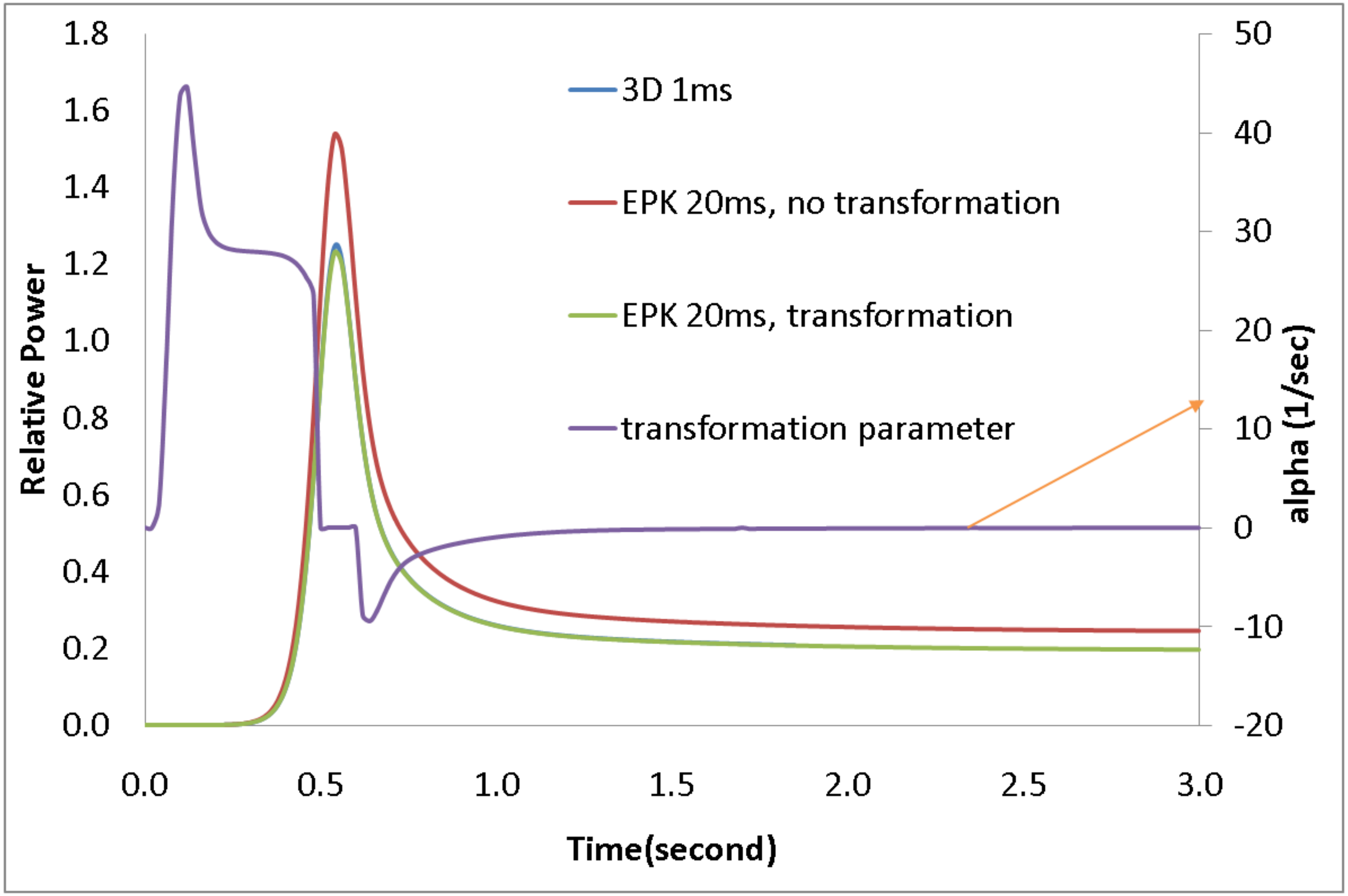
Analysis of Exponential Transportion: EPK solution for NEACRP-A1 with 1ms steps



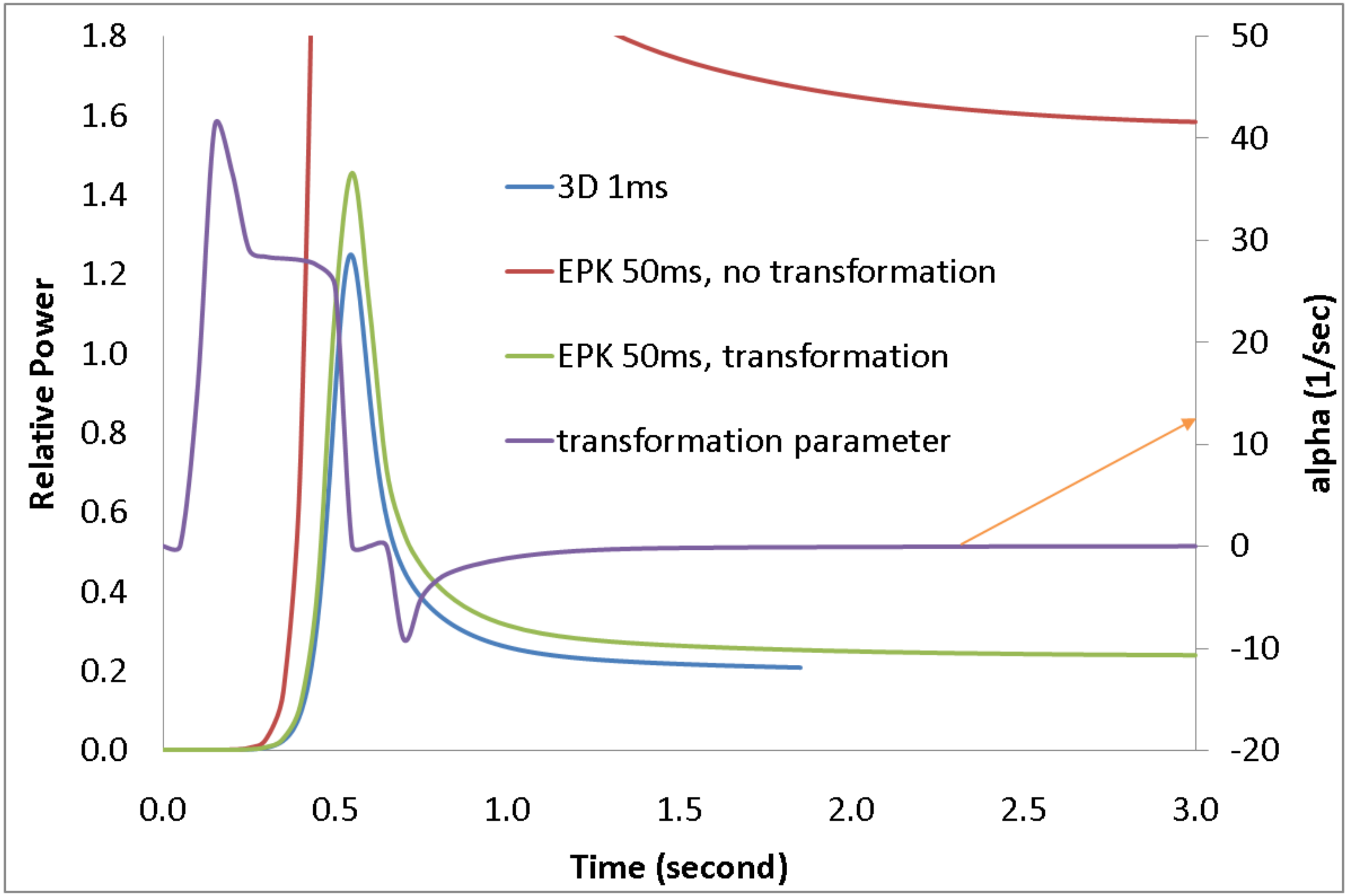
EPK solution for NEACRP-A1 with 10ms steps



EPK solution for NEACRP-A1 with 20ms steps



EPK solution for NEACRP-A1 with 50ms steps



Summary of Applications

- Exact point kinetics can reproduce exactly the same power solution from the spatial kinetics solution with the pre-computed reactivity and time dependent kinetics parameters.

- Point kinetics with pre-computed reactivity and only the initial kinetics parameters predicts power solution slightly different from spatial kinetics solution.

- There is a large error in control rod reactivity using only the initial flux shape.

- There are errors in the feed back reactivity evaluated with cross section or core average parameters. The reactivity evaluated with power weighted core average parameters has the largest error among the three methods.

Backup Slides

Spatial Kinetics Equations

- Transient Neutron Diffusion Equation

$$\frac{1}{v_g(\mathbf{r})}\frac{\partial\phi_g(\mathbf{r},t)}{\partial t} = -\nabla\bullet J_g(\mathbf{r},t) + \sum_{g'}\Sigma_{g,g'}\left(\mathbf{r},t\right)\phi_{g'}(\mathbf{r},t) - \Sigma_{tg}(\mathbf{r},t)\phi_g(\mathbf{r},t) + \chi_g\left(\mathbf{r}\right)S^F(\mathbf{r},t)$$

$$+\sum_k \chi_{dk,g}\left(\mathbf{r}\right)\left(\lambda_k\left(\mathbf{r}\right)C_k(\mathbf{r},t) - \beta_k(\mathbf{r})S^F(\mathbf{r},t)\right)$$

$$S^F(\mathbf{r},t) = \frac{1}{k^s_{eff}}\sum_{g'}\nu\Sigma_{f,g'}\left(\mathbf{r},t\right)\phi_{g'}(\mathbf{r},t), \qquad J_g(\mathbf{r},t) = -D_g(\mathbf{r},t)\nabla\phi_g(\mathbf{r},t)$$

$$\chi_g = \chi_{p,g} + \sum_k \beta_k(\chi_{dk,g} - \chi_{p,g}),$$

- Precursor equations

$$\frac{\partial C_k(\mathbf{r},t)}{\partial t} = \beta_k(\mathbf{r})S^F(\mathbf{r},t) - \lambda_k(\mathbf{r})C_k(\mathbf{r},t), \qquad k = 1, 2....$$

Magnitude and Form Functions

- Magnitude and Form Functions

$$\phi_g(\mathbf{r},t) = p(t)\psi_g(\mathbf{r},t),$$

- Normalization of Form Function

$$< \ \ > \equiv \sum_g \iiint_v dv$$

: Integration notation

$$< \phi_g^*(\mathbf{r}) \frac{1}{v_g(\mathbf{r})} \psi_g(\mathbf{r},t) > = < \phi_g^*(\mathbf{r}) \frac{1}{v_g(\mathbf{r})} \psi_g(\mathbf{r},0) >$$

Adjoint flux

- Alternative Normalization?

$$< \kappa\Sigma_{f\,g}(r,t)\psi_g(\mathbf{r},t) > = < \kappa\Sigma_{f\,g}(r,0)\psi_g(\mathbf{r},0) >$$

- Power Form Factor

$$f_{fp}(t) = \frac{< \kappa\Sigma_{f\,g}(r,t)\phi_g(r,t) > p(0)}{p(t) < \kappa\Sigma_{f\,g}(r,0)\phi_g(r,0) >} = \frac{< \kappa\Sigma_{f\,g}(r,t)\psi_g(r,t) >}{< \kappa\Sigma_{f\,g}(r,0)\psi_g(r,0) >}, \qquad f_{fp}(0) = 1$$

- Fission Power Level

$$H(t) = f_{fp}(t)p(t)$$

Integrate Flux Equation with Adjoint weighting

$$<\phi_g^*(\mathbf{r})\frac{1}{v_g}\frac{\partial\phi_g(\mathbf{r},t)}{\partial t}>=<\phi_g^*(\mathbf{r})\left(-\nabla\bullet J_g(\mathbf{r},t)+\sum_{g'}\Sigma_{g,g'}(\mathbf{r},t)\phi_{g'}(\mathbf{r},t)-\Sigma_{tg}(\mathbf{r},t)\phi_g(\mathbf{r},t)+\chi_g(\mathbf{r})S^F(\mathbf{r},t)\right)>$$

$$+<\phi_g^*(\mathbf{r})\sum_k\chi_{dk,g}(\mathbf{r})\left[\lambda_k(\mathbf{r})C_k(\mathbf{r},t)-\beta_k(\mathbf{r})S^F(\mathbf{r},t)\right]>$$

- Time Derivative Term

$$<\phi_g^*(\mathbf{r})\frac{1}{v_g(\mathbf{r})}\phi_g(\mathbf{r},t)> = p(t)<\phi_g^*(\mathbf{r})\frac{1}{v_g(\mathbf{r})}\psi_g(\mathbf{r},t)> = p(t)<\phi_g^*(\mathbf{r})\frac{1}{v_g(\mathbf{r})}\psi_g(\mathbf{r},0)>$$

- Generation Time

$$\Lambda(t)=\frac{<\phi_g^*(\mathbf{r})\frac{1}{v_g(\mathbf{r})}\psi_g(\mathbf{r},t)>}{F(t)}$$

- Adjoint weighted quasi-stationary fission source

$$F(t)=<\phi_g^*(\mathbf{r})\chi_g(\mathbf{r})S^F(\mathbf{r},t)>/\,p(t)=<\phi_g^*(\mathbf{r})\chi_g(\mathbf{r})\widehat{S}^F(\mathbf{r},t)>,\qquad \widehat{S}^F(\mathbf{r},t)=\frac{1}{k_{eff}^s}\sum_{g'}\nu\Sigma_{f,g'}(\mathbf{r},t)\psi_{g'}(\mathbf{r},t),$$

- Time invariance

$$\Lambda(t)F(t)=\Lambda(0)F(0)=\Lambda_0F_0$$

$$<\phi_g^*(\mathbf{r})\frac{1}{v_g}\frac{\partial\phi_g(\mathbf{r},t)}{\partial t}>=\frac{\partial}{\partial t}<\phi_g^*(\mathbf{r})\frac{1}{v_g}\phi_g(\mathbf{r},t)>=\frac{\partial}{\partial t}\left(p(t)<\phi_g^*(\mathbf{r})\frac{1}{v_g(\mathbf{r})}\psi_g(\mathbf{r},0)>\right)=\Lambda_0F_0\frac{dp(t)}{dt}$$

Integrate Flux Equation (cont.)

$$<\phi_g^*(\mathbf{r})\frac{1}{v_g}\frac{\partial\phi_g(\mathbf{r},t)}{\partial t}>=<\phi_g^*(\mathbf{r})\left(-\nabla\bullet J_g(\mathbf{r},t)+\sum_{g'}\Sigma_{g,g'}(\mathbf{r},t)\phi_{g'}(\mathbf{r},t)-\Sigma_{tg}(\mathbf{r},t)\phi_g(\mathbf{r},t)+\chi_g(\mathbf{r})S^F(\mathbf{r},t)\right)>$$

$$+<\phi_g^*(\mathbf{r})\sum_k\chi_{dk,g}(\mathbf{r})\left[\lambda_k(\mathbf{r})C_k(\mathbf{r},t)-\beta_k(\mathbf{r})S^F(\mathbf{r},t)\right]>$$

- Reactivity Term

$$\rho(t)=\frac{<\phi_g^*(\mathbf{r})\left(-\nabla\bullet J_g(\mathbf{r},t)+\sum_{g'}\Sigma_{g,g'}(\mathbf{r},t)\phi_{g'}(\mathbf{r},t)-\Sigma_{tg}(\mathbf{r},t)\phi_g(\mathbf{r},t)+\chi_g(\mathbf{r})S^F(\mathbf{r},t)\right)>}{p(t)F(t)}$$

$$\rho(t)=\frac{1}{p(t)F(t)}<\phi^*(r)\left(F-M\right)\phi(r,t)>$$

Integrate Flux Equation (cont.)

$$<\phi_g^*(\mathbf{r})\frac{1}{v_g}\frac{\partial\phi_g(\mathbf{r},t)}{\partial t}>=<\phi_g^*(\mathbf{r})\left(-\nabla\bullet J_g(\mathbf{r},t)+\sum_{g'}\Sigma_{g,g'}\left(\mathbf{r},t\right)\phi_{g'}(\mathbf{r},t)-\Sigma_{tg}(\mathbf{r},t)\phi_g(\mathbf{r},t)+\chi_g\left(\mathbf{r}\right)S^F(\mathbf{r},t)\right)>$$

$$+<\phi_g^*(\mathbf{r})\sum_k\chi_{dk,g}\left(\mathbf{r}\right)\left[\lambda_k\left(\mathbf{r}\right)C_k(\mathbf{r},t)-\beta_k(\mathbf{r})S^F(\mathbf{r},t)\right]>$$

- Beta Terms

$$\beta_k^{eff}(t)=\frac{1}{p(t)F(t)}<\phi_g^*(\mathbf{r})\chi_{dk,g}\left(\mathbf{r}\right)\beta_k(\mathbf{r})S^F(\mathbf{r},t)>,\qquad \beta^{eff}(t)=\sum_k\beta_k^{eff}(t)$$

$$<\phi_g^*(\mathbf{r})\sum_k\chi_{dk,g}\left(\mathbf{r}\right)\beta_k(\mathbf{r})S^F(\mathbf{r},t)>=p(t)F(t)\sum_k\beta_k^{eff}(t)=p(t)F(t)\beta^{eff}(t)$$

- Delay neutron Terms

$$\lambda_k(t)=\frac{<\phi_g^*(\mathbf{r})\lambda_k(\mathbf{r})\chi_{dk,g}\left(\mathbf{r}\right)C_k(\mathbf{r},t)>}{<\phi_g^*(\mathbf{r})\chi_{dk,g}\left(\mathbf{r}\right)C_k(\mathbf{r},t)>},\qquad \zeta_k(t)=\frac{<\phi_g^*(\mathbf{r})\chi_{dk,g}\left(\mathbf{r}\right)C_k(\mathbf{r},t)>}{F_0},$$

$$<\phi_g^*(\mathbf{r})\sum_k\chi_{dk,g}\left(\mathbf{r}\right)\lambda_k\left(\mathbf{r}\right)C_k(\mathbf{r},t)>=\sum_k\lambda_k(t)\zeta_k(t)F_0$$

Integrate Flux Equation (cont.)

$$<\phi_g^*(\mathbf{r})\frac{1}{v_g}\frac{\partial\phi_g(\mathbf{r},t)}{\partial t}>=<\phi_g^*(\mathbf{r})\left(-\nabla\bullet J_g(\mathbf{r},t)+\sum_{g'}\Sigma_{g,g'}(\mathbf{r},t)\phi_{g'}(\mathbf{r},t)-\Sigma_{tg}(\mathbf{r},t)\phi_g(\mathbf{r},t)+\chi_g(\mathbf{r})S^F(\mathbf{r},t)\right)>$$

$$+<\phi_g^*(\mathbf{r})\sum_k\chi_{dk,g}(\mathbf{r})\left[\lambda_k(\mathbf{r})C_k(\mathbf{r},t)-\beta_k(\mathbf{r})S^F(\mathbf{r},t)\right]>$$

- Put All Terms Together $\Lambda_0 F_0\frac{dp(t)}{dt}=\rho(t)p(t)F(t)-\beta^{eff}(t)p(t)F(t)+\sum_k\lambda_k(t)\zeta_k(t)F_0$

$$\frac{dp(t)}{dt}=\frac{\rho(t)-\beta^{eff}(t)}{\Lambda(t)}p(t)+\frac{1}{\Lambda_0}\sum_k\lambda_k(t)\zeta_k(t)$$

- Why adjoint weighting? Impact on reactivity by form function error

$$\frac{1}{F(t)}<\phi^*(r)\left(\overline{\overline{F}}-\overline{\overline{M}}\right)\left(\psi(r,t)+\delta\psi(r,t)\right)>-\frac{1}{F(t)}<\phi^*(r)\left(\overline{\overline{F}}-\overline{\overline{M}}\right)\psi(r,t)>$$

$$=\frac{1}{F(t)}<\phi^*(r)\left(\overline{\overline{F}}-\overline{\overline{M}}\right)\delta\psi(r,t)>=\frac{1}{F(t)}<\phi^*(r)\left(\overline{\overline{F}}_0+\Delta\overline{\overline{F}}-\overline{\overline{M}}_0-\Delta\overline{\overline{M}}\right)\delta\psi(r,t)>$$

$$=\frac{1}{F(t)}\left(<\phi^*(r)\left(\overline{\overline{F}}_0-\overline{\overline{M}}_0\right)\delta\psi(r,t)>+<\phi^*(r)\left(\Delta\overline{\overline{F}}-\Delta\overline{\overline{M}}\right)\delta\psi(r,t)>\right)$$

- Select weight function so that:

$$<\phi^*(r)\left(\overline{\overline{F}}_0-\overline{\overline{M}}_0\right)\delta\psi(r,t)>\equiv 0\quad \textit{for any }\delta\psi(r,t)\ \Rightarrow\ \left(\overline{\overline{F}}_0^{\,*}-\overline{\overline{M}}_0^{\,*}\right)\phi^*(r)=0$$

Integrate Precursor Eqns with Adjoint weighting

$$< \phi_g^*(\mathbf{r})\chi_{dk,g}(\mathbf{r})\frac{\partial C_k(\mathbf{r},t)}{\partial t} >=< \phi_g^*(\mathbf{r})\chi_{dk,g}(\mathbf{r})\beta_k(\mathbf{r})S^F(\mathbf{r},t) > - < \phi_g^*(\mathbf{r})\chi_{dk,g}(\mathbf{r})\lambda_k(\mathbf{r})C_k(\mathbf{r},t) >$$

- Time Derivative Term $\zeta_k(t) = \frac{< \phi_g^*(\mathbf{r})\chi_{dk,g}(\mathbf{r})C_k(\mathbf{r},t) >}{F_0},$

$$< \phi_g^*(\mathbf{r})\chi_{dk,g}(\mathbf{r})\frac{\partial C_k(\mathbf{r},t)}{\partial t} >= \frac{\partial}{\partial t} < \phi_g^*(\mathbf{r})\chi_{dk,g}(\mathbf{r})C_k(\mathbf{r},t) >= \frac{d}{dt}\left(\zeta_k(t)F_0\right) = F_0\frac{d}{dt}\zeta_k(t)$$

- Source term $\beta_k^{eff}(t) = \frac{< \phi_g^*(\mathbf{r})\chi_{dk,g}(\mathbf{r})\beta_k(\mathbf{r})S^F(\mathbf{r},t) >}{p(t)F(t)},$

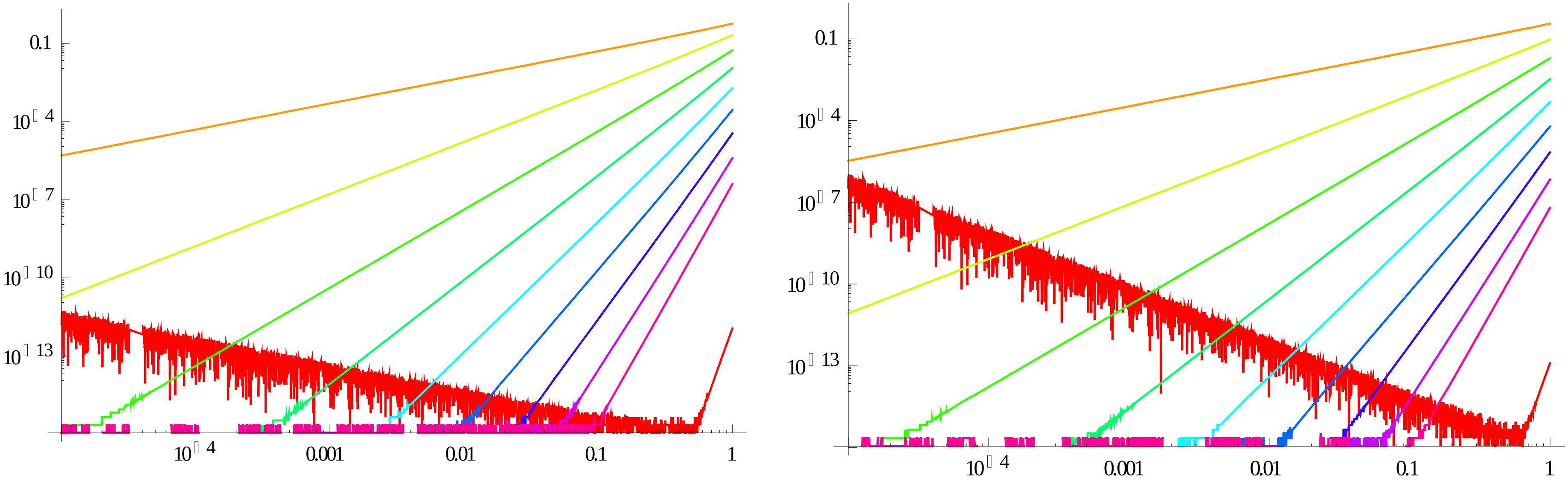
$$< \phi_g^*(\mathbf{r})\chi_{dk,g}(\mathbf{r})\beta_k(\mathbf{r})S^F(\mathbf{r},t) >= \beta_k^{eff}(t)p(t)F(t)$$

- Decay term $< \phi_g^*(\mathbf{r})\chi_{dk,g}(\mathbf{r})\lambda_k(\mathbf{r})C_k(\mathbf{r},t) >= \lambda_k(t)\zeta_k(t)F_0$

- Put All Terms Together $F_0\frac{d\zeta_k(t)}{dt} = \beta_k^{eff}(t)p(t)F(t) - \lambda_k(t)\zeta_k(t)F_0$

$$\frac{d\zeta_k(t)}{dt} = \frac{\Lambda_0}{\Lambda(t)}\beta_k^{eff}(t)p(t) - \lambda_k(t)\zeta_k(t)$$

Numerical error of evaluating Kappa0 and Kappa1



Differences of 0-8th order Taylor expansions and direct evaluation of Kappa0 and Kappa1 from 13th order Taylor expansions value. The Taylor expansions functions have significant truncation error for large input. On the other hand, for small input, there is large round off error in numerical evaluation of kappa2 due to subtract two close numbers.