NUCLEAR REACTOR KINETICS

Lecture 7

More PKE Solutions from Spatial Solutions:

Better Shape Functions, Quasi-Static Methods

Massachusetts
Institute of
Technology

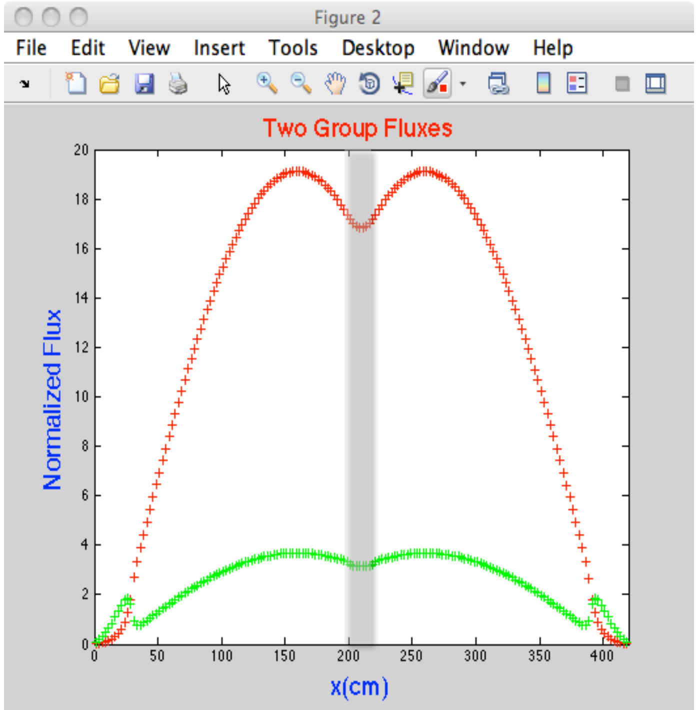
Today's Lecture: Goals

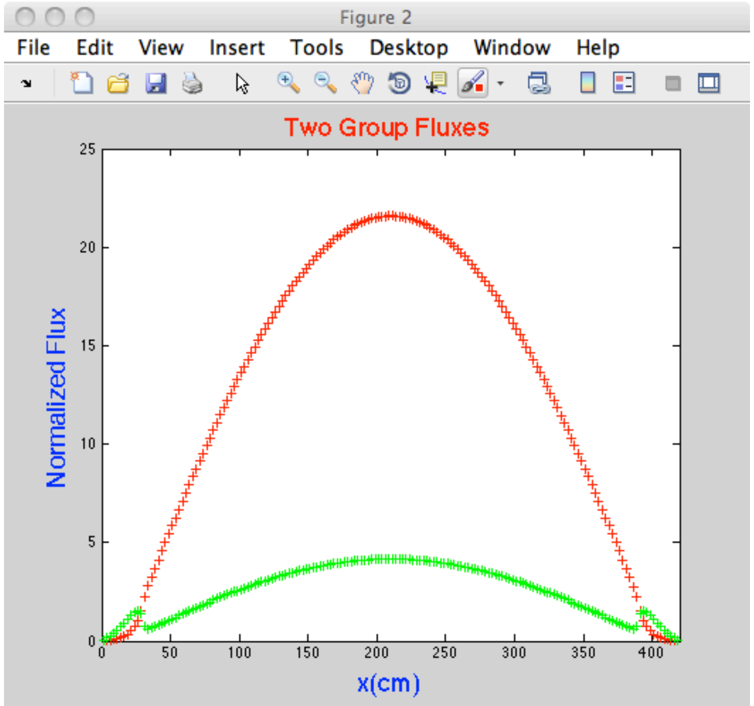
- Re-baseline our homework and lectures schedule?
- Answer any questions about Pset 3 assignment: 1-D, 2-group, transient finite-difference solutions
- Understand relationship between dynamic and static reactivity
- Understand adjoint flux shape questions from last lecture
- Reiterate lessons from last lecture (IKEs vs. dynamic reactivity)
- Understand 2-group PKEs and its implications
- Examine Quasi-Static reactivity approximations
- Discuss upcoming lectures and paths forward

Course Outline

22.213 (22.S904) Calendar					
Lecture #	Date	Topic	LECTURER	Read	Assignment Handed Out
1	5-Sep	Course Overview and First Day Exam	Smith		
2	10-Sep	Review of Delayed Neutrons and Point Kinetics Equations	Smith		PSET # 1: Point Kinetics
3	12-Sep	Review Steady-State Finite-Difference Diffusion Methods (1D, 2D)	Smith		
4	17-Sep	Generalized PKEs from Spatial Finite-Difference Diffusion	Smith		PSET # 2: 2-D Steady-State Diffusion
5	19-Sep	Basic Transient Finite-Difference Direct Solutions	Smith		
6	24-Sep	Testing Various PKE implementations	Smith		PSET # 3: 2-D Fully-Implicit Diffusion
7	26-Sep	Quasi-Static Time-Integration	Smith		
8	1-Oct	Higher-order Time Integration	Smith		PSET # 4: PKE from 2-D Diffusion
9	3-Oct	Time Stepping for Automatic Error Control	Smith		
	8-Oct	Columbus Holiday (8th and 9th)			
10	10-Oct	Smith on Travel: Time for Course Project and Homework	None		PSET # 5: PKE Time Step Control
11	15-Oct	Iterative Numerical MethodsL PJ, GS, SOR, CG, GMRES, etc.	Smith		
12	17-Oct	Coarse Mesh Rebalance & Nonlinear Diffusion Acceleration	Smith		PSET # 6: PKE with Nonlinear Feedback
13	22-Oct	Nodal Methods: Kinetic Distortion and Frequency Transformation	Smith		
	24-Oct	Midterm Exam			
14	29-Oct	Midterm Detailed Exam Solution/2D LRA SS Comparisons	Smith		PSET # 7: CMR and NDA acceleration
15	31-Oct	Multigrid Acceleration Methods	Smith		
16	5-Nov	JFNK for Non-linear Systems	Smith		2-D LRA Rod Ejection Problem: Contest
17	7-Nov	Transient Sn	Smith		
	12-Nov	Veterns Day Holiday			
	14-Nov	Special Project Work Period	ANS Meeting		
18	19-Nov	Transient MOC	Smith		
19	21-Nov	Parallel Solver Technologies (PetSc)	Herman/Roberts		
20	26-Nov	So You Want To Be A Professor? Student Lectures	?????		
21	28-Nov	So You Want To Be A Professor? Student Lectures	?????		
22	3-Dec	So You Want To Be A Professor? Student Lectures	?????		
23	5-Dec	So You Want To Be A Professor? Student Lectures	?????		
24	10-Dec	So You Want To Be A Professor? Student Lectures	?????		
25	12-Dec	Last Day of Class General Wrapup, Cats and Dogs, Critique	Smith		
	17-21 Dec	Finals Week - No Exam for 22.S904 (22.213)			

Background Reading





DIFFUSION THEORY METHODS FOR SPATIAL KINETICS CALCULATIONS

T. M. Sutton and B. N. Aviles

Knolls Atomic Power Laboratory, P. O. Box 1072, Schenectady, NY 12301-1072, U.S.A.

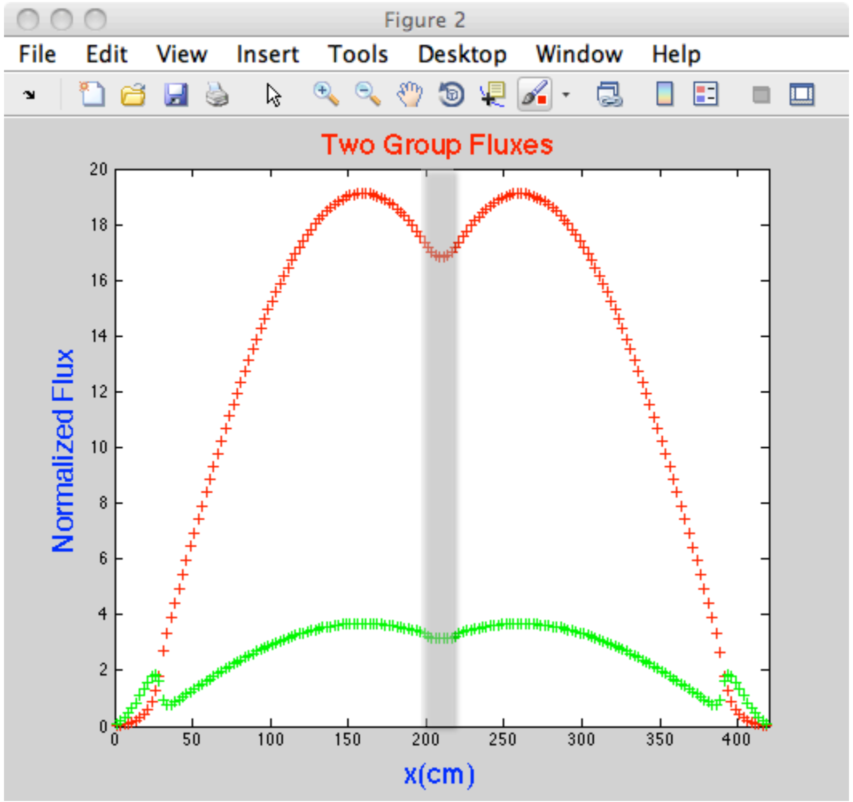
PSET # 3
SOLVING 2-GROUP TRANSIENT FINITE-DIFFERENCE EQUATIONS

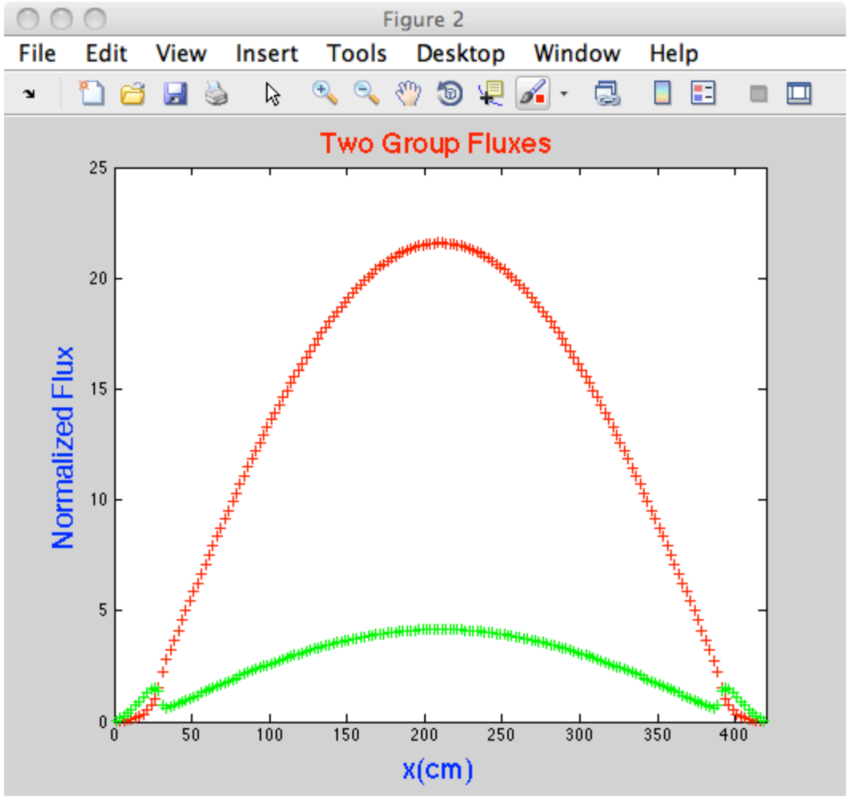
Due: Oct. 1, 2012

Massachusetts Institute of Technology

Pset 3: Reference Transient Solutions for Time Integration Testing

Transient Control Rod Withdrawal in a 1-D Reflected Core





Dynamic Simulations in 1-D
No Feedback Cases for:
Delayed Critical
Prompt Critical

Pset 3

Solve 1-D, 2-Group Diffusion Problem for Rod Withdrawal and Insertion for the Following Materials, Geometry, and Perturbations

D1, D2, Sigma-a1, Sigma-a2, Sig1→2, Nu*sigma-f1, Nu*sigma-f2

```
%
% define two group cross sections for all potential materials
%
nmat=6;
xs(1,1)=1.300; xs(2,1)=0.500; xs(3,1)=0.0098; xs(4,1)=0.114; xs(5,1)=0.022; xs(6,1)=0.006; xs(7,1)=0.1950;  % 3% enriched fuel
xs(1,2)=1.300; xs(2,2)=0.500; xs(3,2)=0.0105; xs(4,2)=0.134; xs(5,2)=0.022; xs(6,2)=0.008; xs(7,2)=0.2380;  % 4% enriched fuel
xs(1,3)=1.500; xs(2,3)=0.500; xs(3,3)=0.0002; xs(4,3)=0.010; xs(5,3)=0.032; xs(6,3)=0.000; xs(7,3)=0.0000;  % water
xs(1,4)=1.300; xs(2,4)=0.500; xs(3,4)=999.99; xs(4,4)=999.9; xs(5,4)=0.020; xs(6,4)=0.000; xs(7,4)=0.0000;  % black absorber
xs(1,5)=1.300; xs(2,5)=0.500; xs(3,5)=0.0098; xs(4,5)=0.118; xs(5,5)=0.022; xs(6,5)=0.006; xs(7,5)=0.1950;  % 3% enriched + rod1
xs(1,6)=1.300; xs(2,6)=0.500; xs(3,6)=0.0150; xs(4,6)=0.145; xs(5,6)=0.022; xs(6,6)=0.006; xs(7,6)=0.1950;  % 3% enriched + rod2
%
% define problem geometry and assign materials:
%
%   #zones=NZONE w(nzone),n(nzone), mat(nzone)
%
%  bc | slab 1| slab 2| slab 3| ..... slab(NZONE) | bc
%
NZONE=5;                                                  % numer of material zones
w(1)= 30; w(2)=170; w(3)=20;  w(4)=170; w(5)=30;          % width    per zone
n(1)= 3;  n(2)=17;  n(3)=2;   n(4)=17;  n(5)=3;           % mesh     per zone
m(1)= 3;  m(2)=1 ;  m(3)=5;   m(4)=1;   m(5)=3;           % material per zone problem 1
m(1)= 3;  m(2)=1 ;  m(3)=6;   m(4)=1;   m(5)=3;           % material per zone problem 2

n=n*10;                                                    % mesh refinement factor
bc=2;                                                     % 0=zero flux, 1=zero incoming, 2=reflective bc
```

Rod for problem #1
Rod for problem #2

1. b.c. | Reflector | Fuel-1 | Rod | Fuel-1| Reflector | b.c.
2. Cross sections, zone widths, and cross section given above
3. Reflective b.c. on outer surfaces (same as steady-state in Pset 2)

Pset 3
Fix the velocity error from lecture 4

```
%
% kinetics parameters
%
betai  = [.000218 .001023 .000605 .00131 .00220 .00060 .000540 .000152];   % no units, 8-group data
halfli = [55.6      24.5     16.3    5.21   2.37  1.04    0.424    0.195 ];  % half-life in seconds
decayi = log(2)./halfli;                                                   % sec-1
vel1=2200.*100.*(.100e4/.0253)^.5; % cm/sec
vel2=2200.*100.*(.100  /.0253)^.5; % cm/sec
beta= sum(betai);  fprintf ('\n%s',' beta-effective - '); fprintf ('%14.7f',beta);

   if ido==5
```

Pset 3 (Part A)

Evaluate

Pset 3 (Part A)

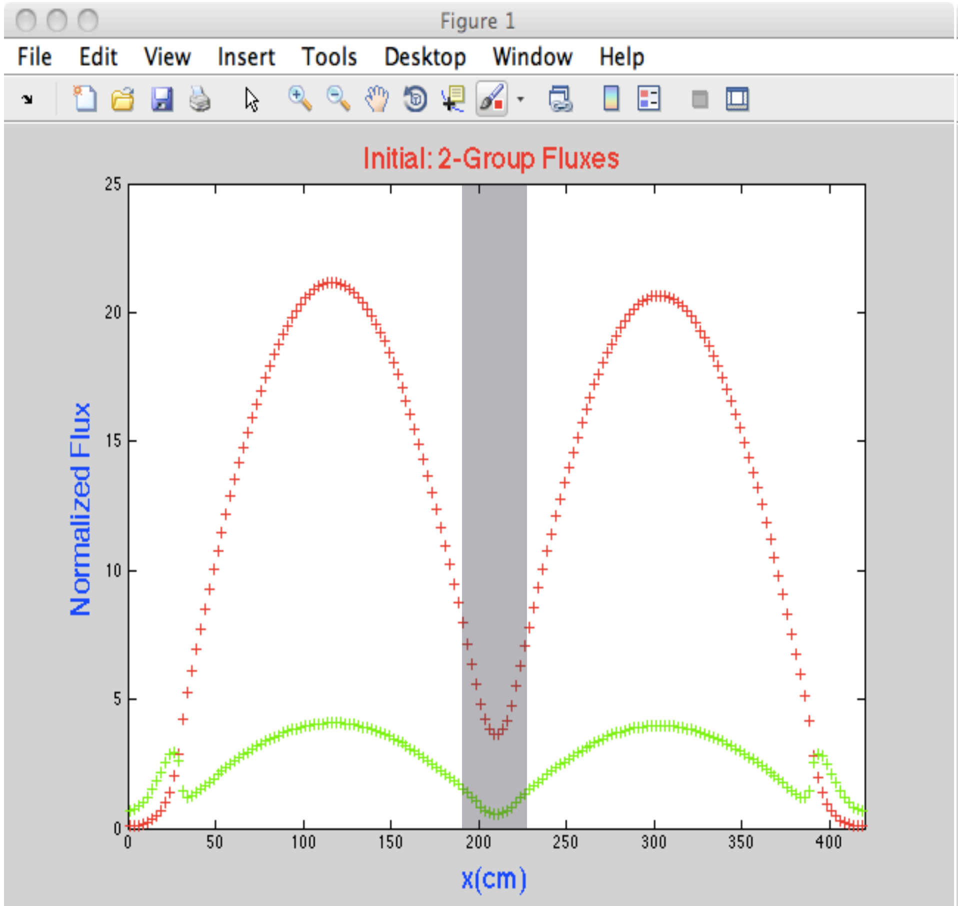
Write your own transient diffusion solver (in any language you choose) using MATLAB inversion, P-J, or G-S iterative flux inversion

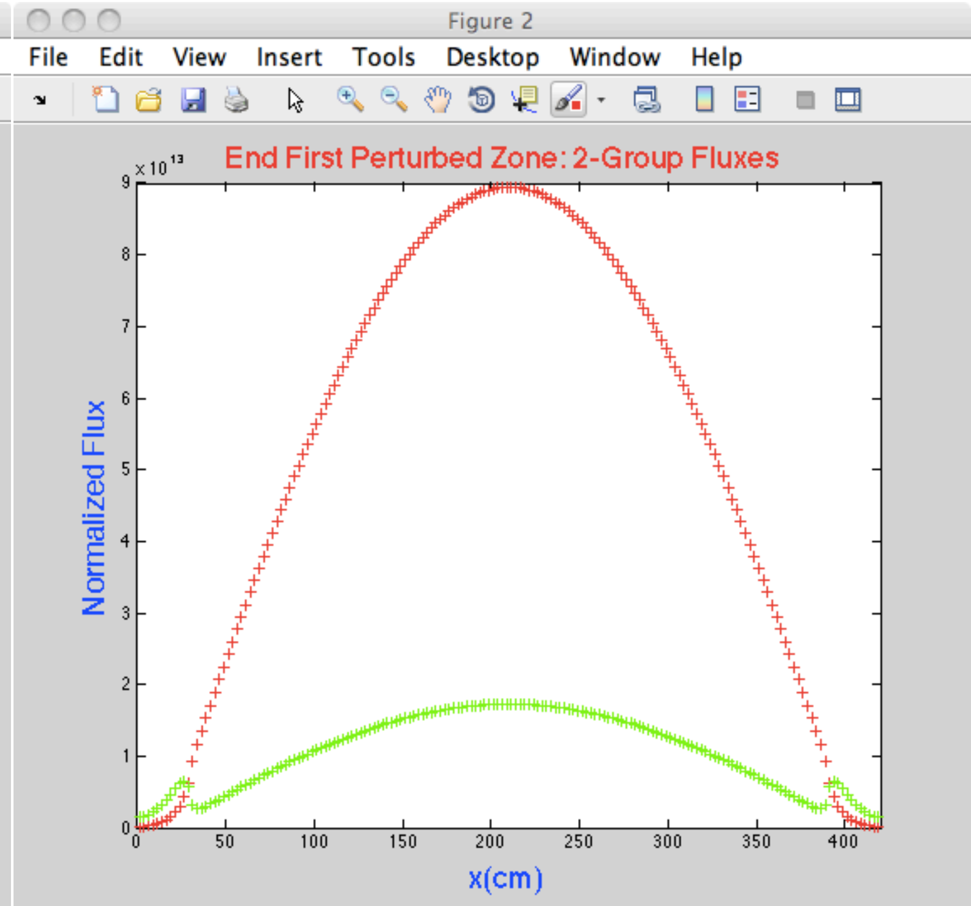
PART A: Delayed Critical Bank Withdrawal

1. Assume solution is spatially converged with 1.0 cm spatial mesh
2. Withdraw the control rod uniformly over t = [2.00,4.00] seconds
3. Insert the control rod uniformly over t = [10.0,12.0] seconds
4. Follow the transient until t=50.0 sec.

- What is the static rod worth (in fraction of beta)?
- What uniform time step size is required to converge the peak core power to 1%?
- What uniform time step size is required to converge the final core power to 1%?
- Plot normalized core power vs. time for the converged time-step (both powers).
- Plot fractional error in peak and final core powers vs. time step when the converged time step is multiplied by 1, 2, 4, 8, 16, 32, and 64.
- Plot fractional error in peak and final core powers vs. spatial mesh (using converged time step) for mesh of 1.0, 2.0, 5.0, and 10.0 cm.

Pset 3 (Part B)





Super Prompt Critical Rod Bank Withdrawal

Pset 3 (Part B)

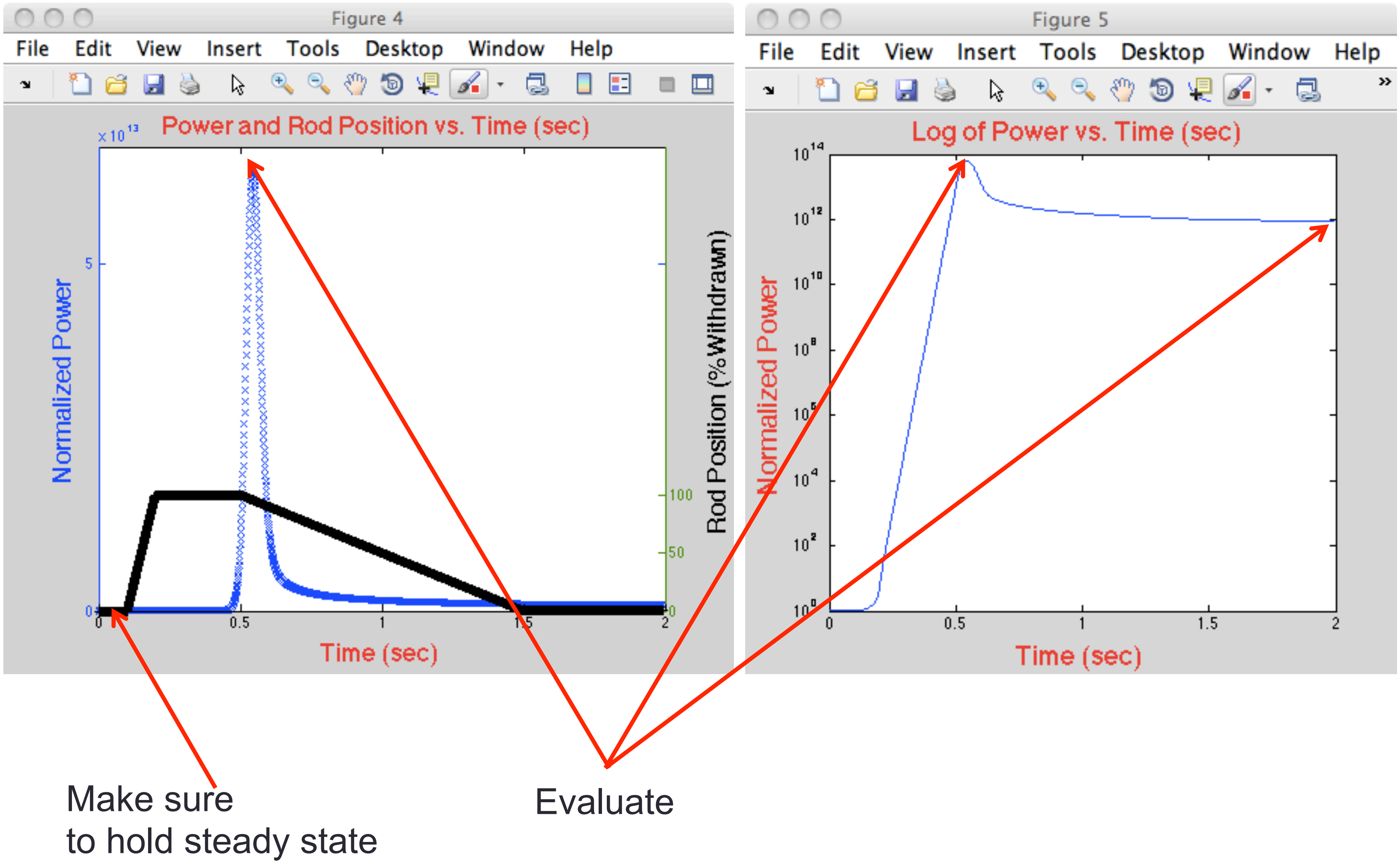


Figure 4
File Edit View Insert Tools Desktop Window Help
Power and Rod Position vs. Time (sec)
x 10^13
5
0
Normalized Power
Rod Position (% Withdrawn)
100
50
0
0.5
1
1.5
2
Time (sec)
Figure 5
File Edit View Insert Tools Desktop Window Help
Log of Power vs. Time (sec)
10^14
10^12
10^10
10^8
10^6
10^4
10^2
10^0
Normalized Power
Time (sec)
Make sure
to hold steady state
Evaluate

Pset 3 (Part B)

Write your own transient diffusion solver (in any language you choose) using MATLAB inversion, P-J, or G-S iterative flux inversion

PART B: Super-Prompt Critical Rod Withdrawal and Scram

1. Assume solution is spatially converged with 1.0 cm spatial mesh.
2. Withdraw the control rod uniformly over t = [0.10,0.20] seconds
3. Scram the control rod uniformly over t = [0.50,1.50] seconds
4. Follow the transient until t=2.0 sec.

- What is the static rod worth (in fraction of beta)?
- What is the dominance ratio of the static rod-inserted base case?
- What uniform time step size is required to converge the peak core power to 10%?
- What uniform time step size is required to converge the final core power to 1%?
- Plot normalized core power vs. time for the converged time-step (both powers).
- Plot fractional error in peak and final core powers vs. time step when the converged time step is multiplied by 1, 2, 4, 8, 16, 32, and 64.
- Plot fractional error in peak and final core powers vs. spatial mesh (using converged time step) for mesh of 1.0, 2.0, 5.0, and 10.0 cm.

Point Kinetics Solutions

- Compare true time-dependent solution to:
  - PKE solution using unity-weighted initial steady-state fluxes
  - PKE solution using unity-weighted final steady-state fluxes
  - PKE solution using adjoint-weighted initial steady-state fluxes
  - PKE solution using adjoint-weighted final steady-state fluxes
  - Geometrically-interpolated shape- and weight-function PKEs
  - 2-group PKE solution using unity-weighted initial fluxes
  - 2-group PKE solution using adjoint-weighted initial fluxes
  - Adiabatic, alpha-mode, and omega-mode methods
  - Quasi-Static, Improved Quasi-Static, etc.

Numerical Solution of Inverse Kinetics Equations

- First, let's examine reactivity from IKE to see what spatial kinetics reactivity is:

$$[C(t)] = [e^{-At}][C_0] + [e^{-At}][A]^{-1}\{[e^{At}][Y] - [Y_0]\}$$

$$where \quad Y(t) \equiv [\beta_i]\frac{T(t)}{\Lambda}$$

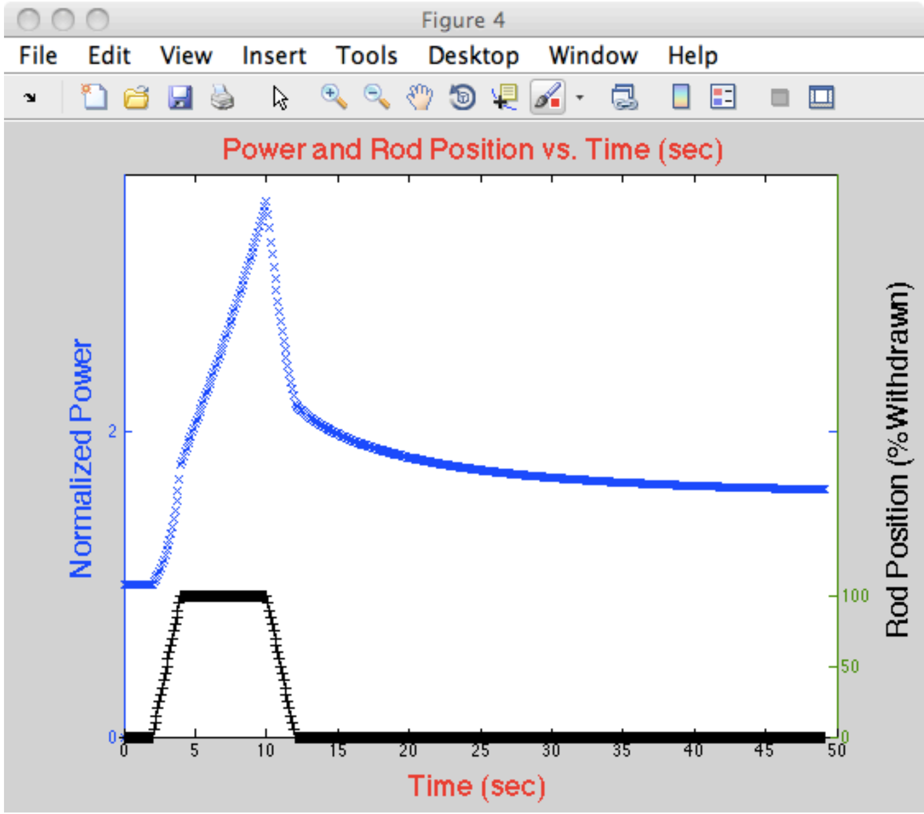
- Solve for time-varying concentrations of precursors for reactor power shape vs. time by applying this equation for discrete steps.

- From the PKEs, we solve for reactivity in terms of precursor concentration and reactor power vs. time by making finite-difference approximation

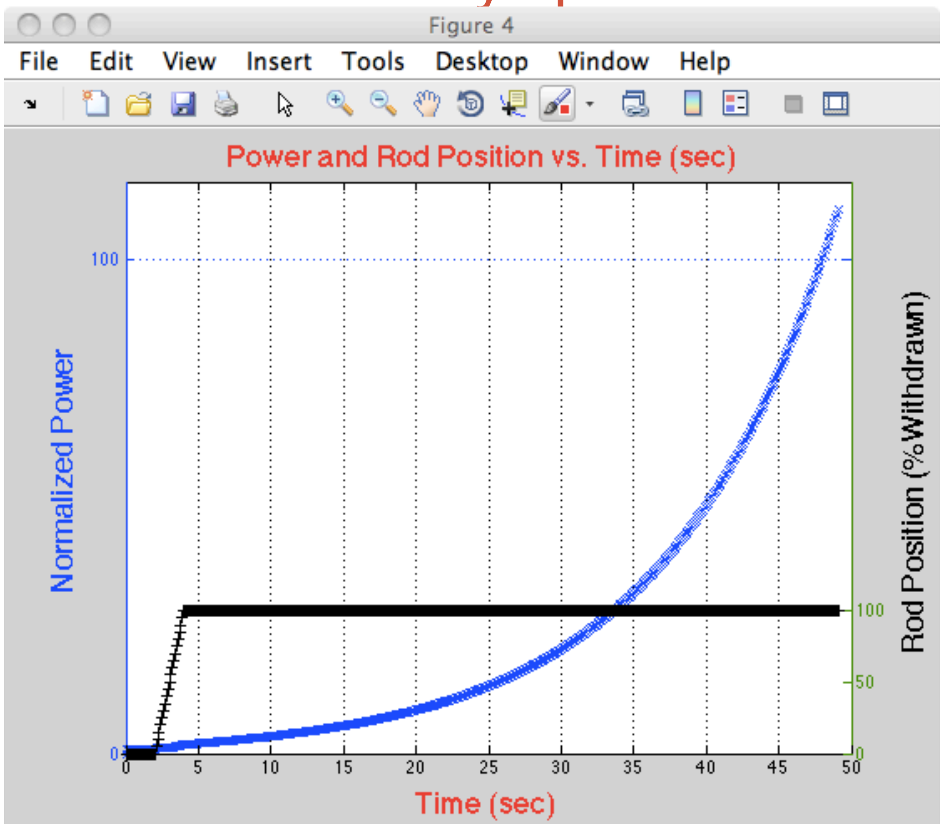
$$\frac{d}{dt}T(t) = \frac{\rho(t)-\beta}{\Lambda}T(t) + \sum_i \lambda_i C_i(t)$$

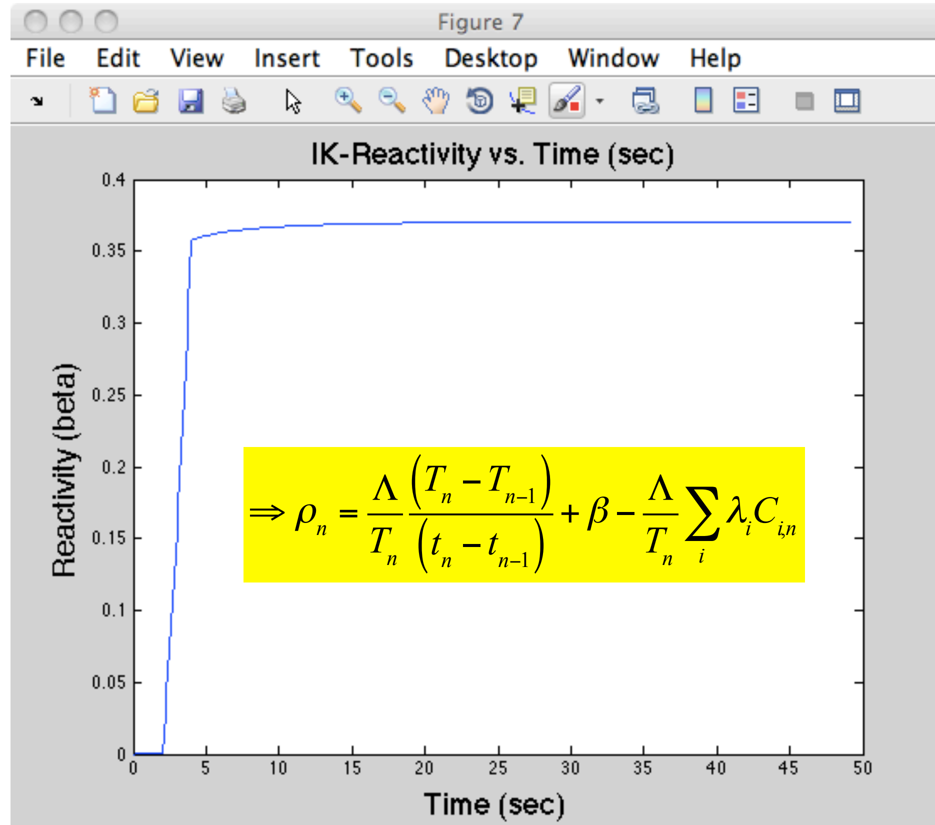
$$\Rightarrow \rho(t) = \frac{\Lambda}{T(t)}\frac{d}{dt}T(t) + \beta - \frac{\Lambda}{T(t)}\sum_i \lambda_i C_i(t)$$

$$\Rightarrow \rho_n = \frac{\Lambda}{T_n}\frac{(T_n - T_{n-1})}{(t_n - t_{n-1})} + \beta - \frac{\Lambda}{T_n}\sum_i \lambda_i C_{i,n}$$



Asymptotic Inverse Point Kinetics Solution





$$\rho_{IK} = 0.37230\beta$$

$$k_{eff}^{rodded} = 1.36460 \qquad k_{eff}^{unrodded} = 1.36799$$

$$\rho_{static} = \frac{1.36799 - 1.36460}{1.36460(\beta)} = 0.37265\beta$$

Adjoint Fluxes for Critical Reactor Systems

Expressing the transport (or diffusion) equation in operator notation:

$$A\psi = \frac{1}{k}M\psi \qquad \Rightarrow A\psi - \frac{1}{k}M\psi = 0$$

$$A^*\psi^* = \frac{1}{k^*}M^*\psi \quad \Rightarrow A^*\psi^* - \frac{1}{k^*}M^*\psi = 0$$

Multiplying first equation by adjoint flux/integrate over phase space, multiplying second equation by real flux/integrate over phase space, subtract later from former:

$$\left\langle \psi^*, A\psi \right\rangle - \left\langle \psi, A^*\psi^* \right\rangle - \frac{1}{k}\left\langle \psi^*, M\psi \right\rangle + \frac{1}{k^*}\left\langle \psi, M^*\psi^* \right\rangle = 0$$

But recalling the definition of the adjoint operating on any operator f

$$\left\langle \psi^*, f\psi \right\rangle = \left\langle \psi, f^*\psi^* \right\rangle$$

Consequently, $$\left\langle \frac{1}{k} - \frac{1}{k^*} \right\rangle \left\langle \psi^*, M\psi \right\rangle = 0$$

The fundamental mode solution (in which we are interested) has everywhere positive real and adjoint fluxes, and since the fission operator is an everywhere positive operator, we find that the real and adjoint eigenvalues must be identical:

$$k^* = k$$

Perturbation Theory Expression for Reactivity

If we start from an unperturbed critical reactor system

$$A_0\psi_0 - \frac{1}{k_0}M_0\psi_0 = 0 \qquad A_0^*\psi_0^* = \frac{1}{k_0^*}M_0^*\psi_0^*$$

and perturb the operators such that

$$A = A_o + \delta A \qquad M = M_o + \delta M \qquad \psi = \psi_o + \delta\psi$$

Then,

$$(A_0 + \delta A)(\psi_0 + \delta\psi) - \frac{1}{(k_0 + \delta k)}(M_0 + \delta M)(\psi_0 + \delta\psi) = 0$$

Expanding

$$\frac{1}{(k_0 + \delta k)} = \frac{1}{k_0(1 + \delta k / k_0)} = \frac{1}{k_0}(1 - \delta k / k_0) + O(\delta k)^2$$

and ignoring terms $O(\delta k)^2$

$$-\frac{\delta k}{k_0^2}M_0\psi_0 = \underbrace{(A_0 - \frac{1}{k_0}M_0)\psi_0}_{0} + (\delta A - \frac{1}{k_0}\delta M)\psi_0 + (A_0 - \frac{1}{k_0}M_0)\delta\psi + O(\delta)^2$$

Multiplying by ψ_0^* and integrating over phase space

$$-\frac{\delta k}{k_0^2}\left\langle \psi_0^*, M_0\psi_0 \right\rangle = \left\langle \psi_0^*, (\delta A - \frac{1}{k_0}\delta M)\psi_0 \right\rangle + \left\langle \psi_0^*, (A_0 - \frac{1}{k_0}M_0)\delta\psi \right\rangle + O(\delta)^2$$

First-Order Perturbation Theory

But since the definition of adjoint operators is for any operator

$$\left\langle \psi_0^*, (A_0 - \frac{1}{k_0} M_0)\delta\psi \right\rangle = \left\langle \delta\psi, (A_0^* - \frac{1}{k_0} M_0^*)\psi_0^* \right\rangle$$

0, for any $\delta\psi$

we can obtain

$$-\frac{\delta k}{k_0^2} = \frac{\left\langle \psi_0^*, (\delta A - \frac{1}{k_0}\delta M)\psi_0 \right\rangle}{\left\langle \psi_0^*, M_0 \psi_0 \right\rangle} + O(\delta)^2$$

There are only second order errors in reactivity

Which is clearly much more accurate than our previous expression:

$$-\frac{\delta k}{k_0^2} = \frac{(\delta A - \frac{1}{k_0}\delta M)\psi_0 + (A_0 - \frac{1}{k_0} M_0)\delta\psi}{M_0 \psi_0} + O(\delta)^2$$

Without knowing $\delta\psi$ there is a first order error in reactivity

Finally, making use of definition of reactivity perturbed from critical

$$\rho = \frac{k-1}{k} \quad \Rightarrow \rho - \rho_0 = \frac{k-1}{k} - \frac{k_0 - 1}{k_0} = \frac{\delta k}{k k_0} \Rightarrow \delta\rho = \frac{\delta k}{(k_0)^2} + O\left(\delta k\right)^2$$

one obtains the first order perturbation (FOP) expression for reactivity:

$$\boxed{\delta\rho \cong \frac{\left\langle \psi_0^*, (\frac{1}{k_0}\delta M - \delta A)\psi_0 \right\rangle}{\left\langle \psi_0^*, M_0 \psi_0 \right\rangle}}$$

Definition of Static Reactivity From Eigenvalue Calculations?

Classic text book definition of reactivity change from state 0 to state 1:

$$\rho_0 = \frac{k_0 - 1}{k_0} \quad \text{and} \quad \rho_1 = \frac{k_1 - 1}{k_1}$$

$$\rho_1 - \rho_0 = \frac{k_1 - 1}{k_1} - \frac{k_0 - 1}{k_0} = \frac{k_0\left(k_1 - 1\right) - k_1\left(k_0 - 1\right)}{k_1 k_0} = \frac{k_1 - k_0}{k_1 k_0}$$

$$\boxed{\Delta\rho_{0\to 1} \equiv \frac{k_1 - k_0}{k_1\, k_0}}$$

Prof Smith asserted in class, that to be consistent with asymptotic reactivity or in-hour equation:

$$\boxed{\Delta\rho_{0\to 1} \equiv \frac{k_1 - k_0}{k_1}}$$

Which definition is correct???

Definition of Static Reactivity From Eigenvalue Calculations?

Professor Smith makes up two cross section libraries for OpenMC that are perfectly consistent except:

1. library "A" has nu = 2.50 for all fissioning species
2. library "B" has nu = 2.75 for all fissioning species (i.e., 10% larger)

Paul runs OpenMC cases for base case 0 and perturbed case 1 with both cross section libraries, and give results to Bryan and Kord:

With Library "A"

$$A_1\psi_1 = \frac{1}{k_1} M_1\psi_1 \qquad A_0\psi_0 = \frac{1}{k_0} M_0\psi_0$$

$$\Delta\rho_{0\to1}^{Bryan} = \frac{k_1 - k_0}{k_1\, k_0} \qquad \Delta\rho_{0\to1}^{Kord} = \frac{k_1 - k_0}{k_1}$$

With Library "B"

$$A_1\psi_1 = \frac{1}{1.1k_1} M_1\psi_1 \qquad A_0\psi_0 = \frac{1}{1.1k_0} M_0\psi_0$$

$$\Delta\rho_{0\to1}^{Bryan} = \frac{1.1\,k_1 - 1.1\,k_0}{(1.1\,k_1)(1.1\,k_0)} = \frac{k_1 - k_0}{1.1\,k_1\, k_0} \qquad \Delta\rho_{0\to1}^{Kord} = \frac{1.1\,k_1 - 1.1\,k_0}{1.1\,k_1} = \frac{k_1 - k_0}{k_1}$$

If we perturb any cross sections and run a transient, we must get a calculated dynamic response that is identical with Library "A" and "B".

What about Reference Reactivity Relative to 1.0 ?

Classic text book definition of reactivity change from state 0 to state 1:

$$\rho_0 = \frac{k_0 - k_{ref}}{k_0} \quad \text{and} \quad \rho_1 = \frac{k_1 - k_{ref}}{k_1}$$

$$\rho_1 - \rho_0 = \frac{k_1 - k_{ref}}{k_1} - \frac{k_0 - k_{ref}}{k_0} = \frac{k_0\left(k_1 - k_{ref}\right) - k_1\left(k_0 - k_{ref}\right)}{k_1 k_0} = \frac{\left(k_1 - k_0\right)k_{ref}}{k_1 k_0}$$

$$\boxed{\Delta\rho_{0\to1} \equiv \frac{\left(k_1 - k_0\right)k_{ref}}{k_1\, k_0}}$$

So if we use the reference state as the initial condition, $k_{ref} = k_0$

$$\boxed{\Delta\rho_{0\to1} \equiv \frac{k_1 - k_0}{k_1}}$$

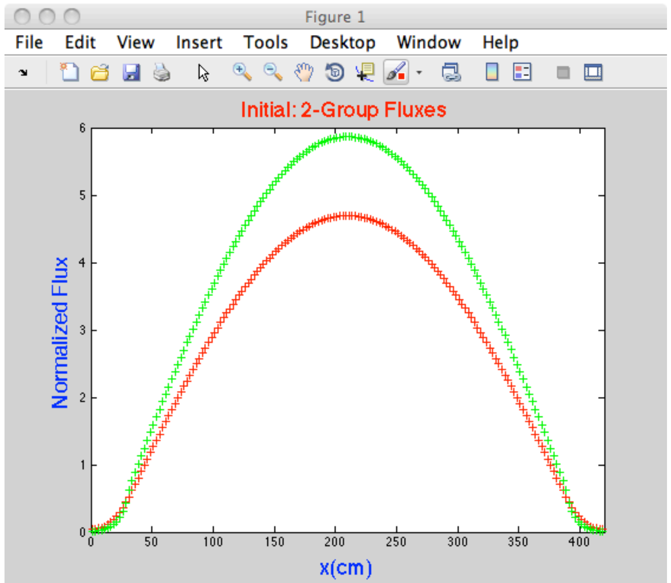
Q.E.D. classic definition is only correct if and only if $k_{ref} \equiv k_0$

So classic definition is only correct if and only if $\rho_0 = 0$, starting condition is critical

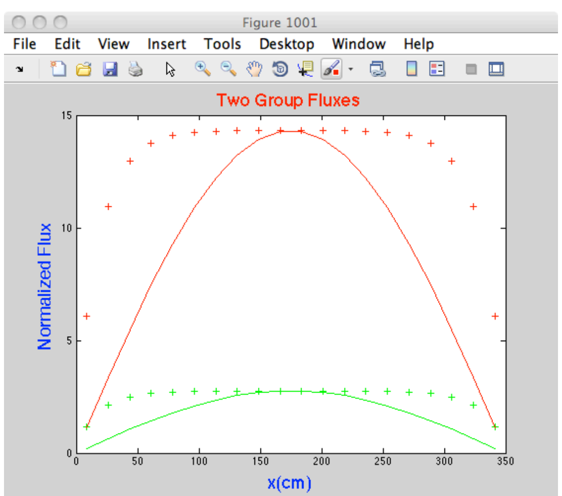
Vlad's Question about Shape of Thermal Adjoint

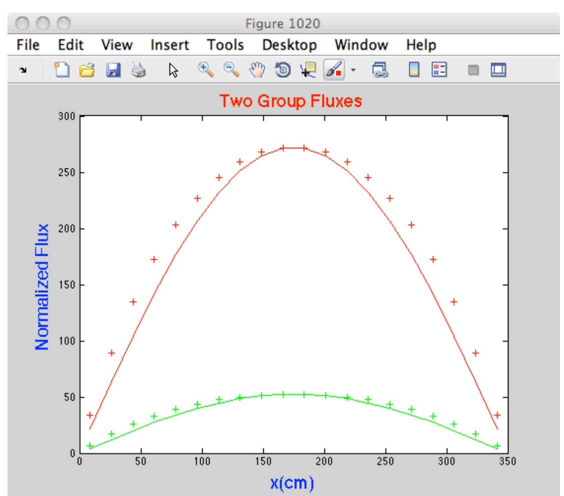
Since mean free path of thermal neutron is short, why is thermal adjoint not flat at the center of the core?

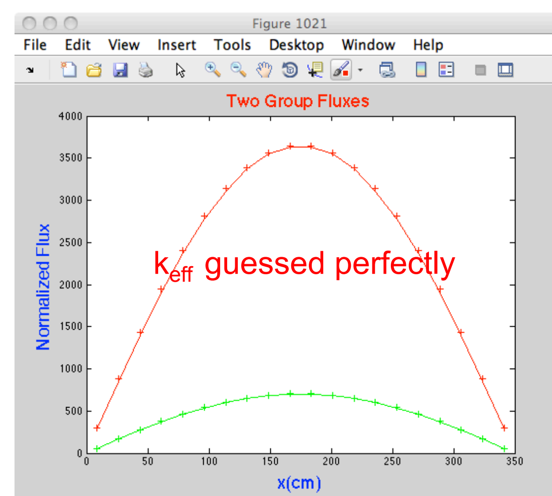
The adjoint flux is defined as the asymptotic increase in total neutron population of a critical reactor per neutron introduced a position r, direction omega, and energy E.



- Let nu=2.5, $D_2 \rightarrow 0.0$, $k_{inf}=1.0+\varepsilon$, $K_{eff}=1.0$
- Start 5 thermal neutrons at center of core
- 3 neutrons are absorbed
- 2+ fission are produced
- 5+ new fission neutrons are emitted
- These 5+ neutrons are born in fast group
- They travel some (much larger) distance
- They cause 5++ more fission neutrons to be emitted
- . . . .
- Remember fixed-source calculation as reactor approaches critical, fission distribution is that of a critical reactor, regardless of starting point of source!



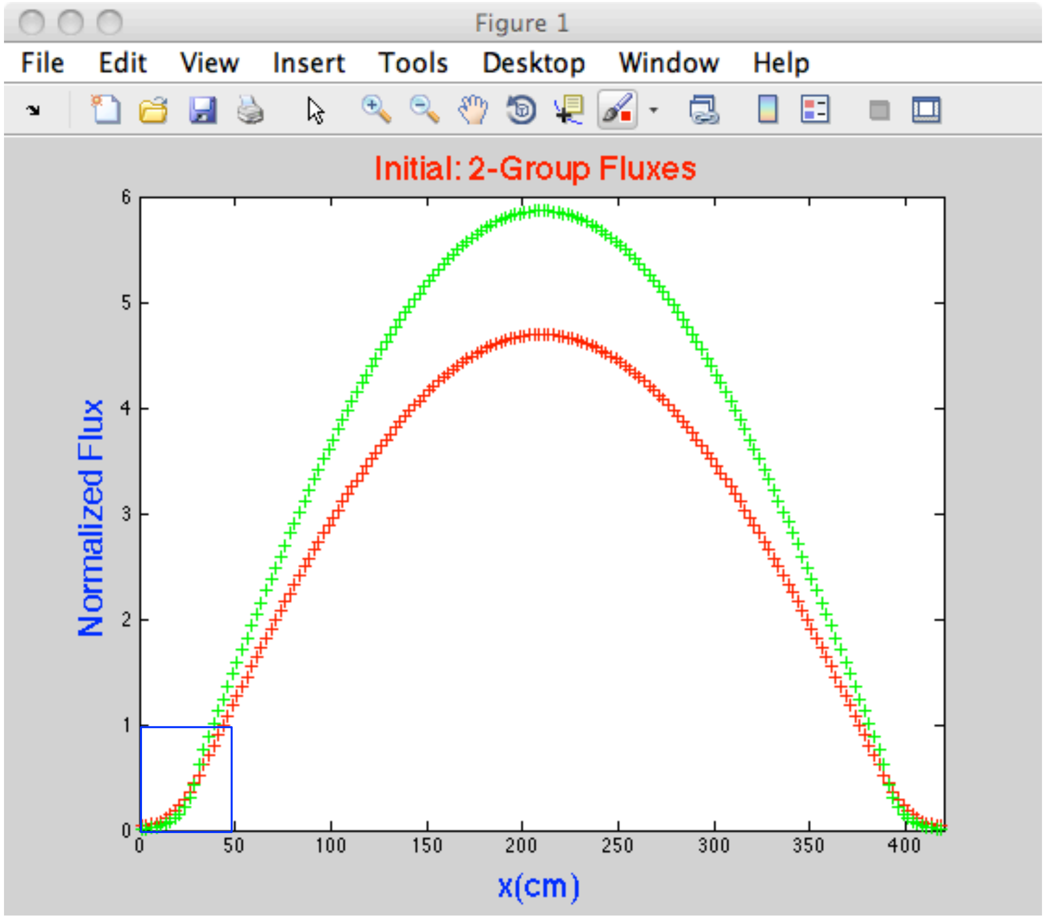




Vlad's Question about Gradient of Thermal Adjoint

Why is gradient of thermal adjoint steeper than the fast adjoint, particularly near the reflector?

The adjoint flux is defined as the asymptotic increase in total neutron population of a critical reactor per neutron introduced a position r, direction omega, and energy E.



- Start a thermal neutron (isotropically) at the center of the reflector
- Start a fast neutron (isotropically) at the center of the reflector
- Fast neutron is more likely to reach fuel.

- Start a thermal neutron (isotropically) at the reflector/core interface
- Start a fast neutron (isotropically) at the reflector/core interface
- Both equally likely to reach fuel.

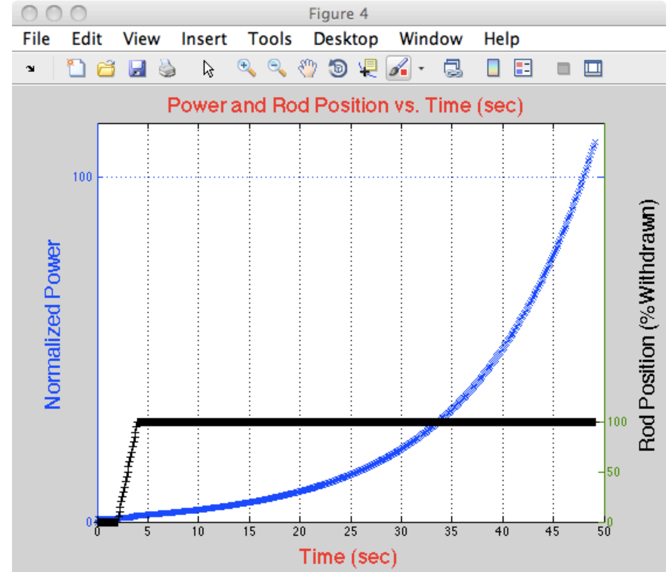
- So gradient of thermal adjoint must be greater than the gradient of fast adjoint in (and near?) the reflector.

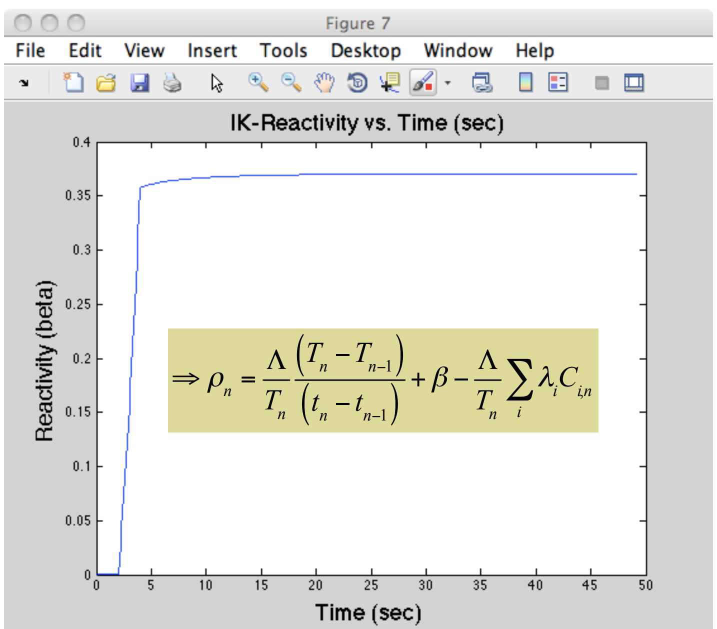
Reactivity: Static, Inverse Kinetics, Dynamic, Shape -Function Estimated

$$\rho(t) \equiv \frac{\left[\begin{array}{c}\int d\vec{r}\; \phi_1^*(\vec{r},t)\left\langle -\hat{\Sigma}_{r,1}(\vec{r},t)\,\phi_1(\vec{r},t) + \left[\nu\Sigma_{f,1}(\vec{r},t)\,\phi_1(\vec{r},t) + \nu\Sigma_{f,2}(\vec{r},t)\,\phi_2(\vec{r},t)\right]\right\rangle \\ + \int d\vec{r}\; \phi_2^*(\vec{r},t)\left\langle -\Sigma_{a,2}(\vec{r},t)\phi_2(\vec{r},t) + \hat{\Sigma}_{1\to 2}(\vec{r},t)\,\phi_1(\vec{r},t)\right\rangle\end{array}\right]}{\int d\vec{r}\; \phi_1^*(\vec{r},t)\left[\nu\Sigma_{f,1}(\vec{r},t)\,\phi_1(\vec{r},t) + \nu\Sigma_{f,2}(\vec{r},t)\,\phi_2(\vec{r},t)\right]}$$

$$\beta_i(t) = \frac{\int d\vec{r}\; \phi_1^*(\vec{r})\beta_i(\vec{r},t)\left[\nu\Sigma_{f,1}(\vec{r},t)\,\phi_1(\vec{r},t) + \nu\Sigma_{f,2}(\vec{r},t)\,\phi_2(\vec{r},t)\right]}{\int d\vec{r}\; \phi_1^*(\vec{r},t)\left[\nu\Sigma_{f,1}(\vec{r},t)\,\phi_1(\vec{r},t) + \nu\Sigma_{f,2}(\vec{r},t)\,S_2(\vec{r},t)\right]\phi_2(\vec{r},t)} = \beta_i$$

$$\Lambda(t) = \frac{\int d\vec{r}\; \phi_1^*(\vec{r},t)\left[\frac{1}{v_1}\,\phi_1(\vec{r},t) + \frac{1}{v_2}\,\phi_2(\vec{r},t)\right]}{\int d\vec{r}\; \phi_1^*(\vec{r},t)\left[\nu\Sigma_{f,1}(\vec{r},t)\,\phi_1(\vec{r},t) + \nu\Sigma_{f,2}(\vec{r},t)\,\phi_2(\vec{r},t)\right]}$$

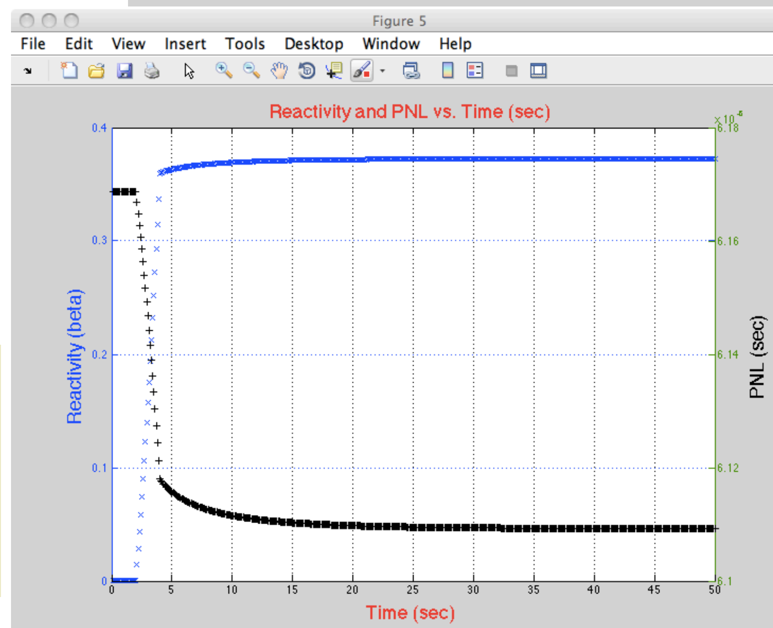




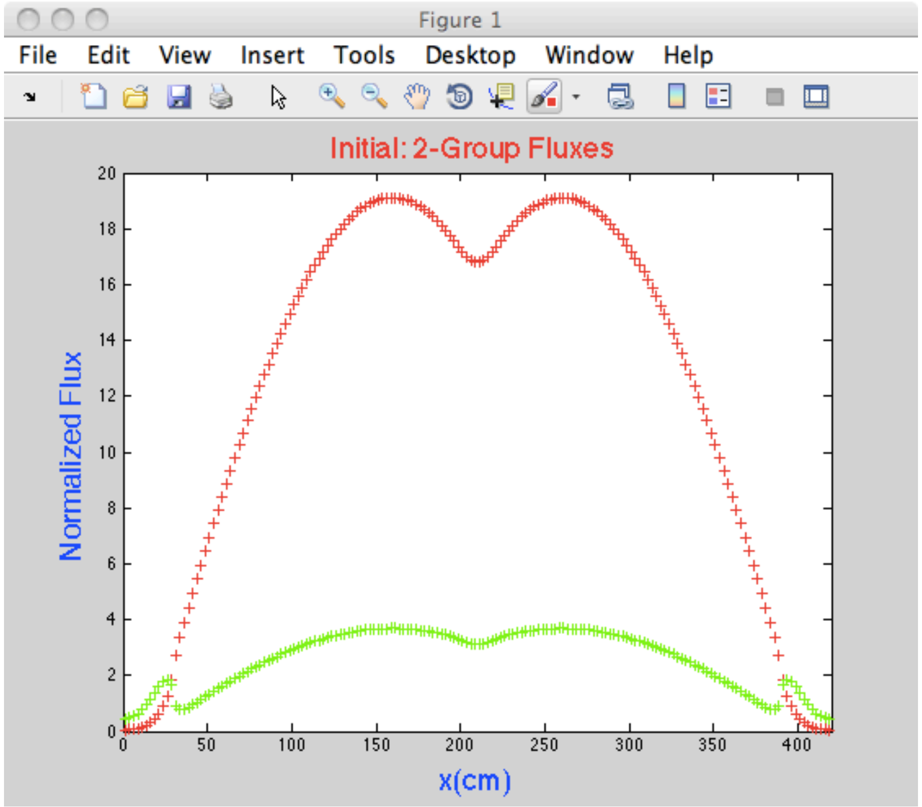
$$\Rightarrow \rho_n = \frac{\Lambda}{T_n}\frac{\left(T_n - T_{n-1}\right)}{\left(t_n - t_{n-1}\right)} + \beta - \frac{\Lambda}{T_n}\sum_i \lambda_i C_{i,n}$$

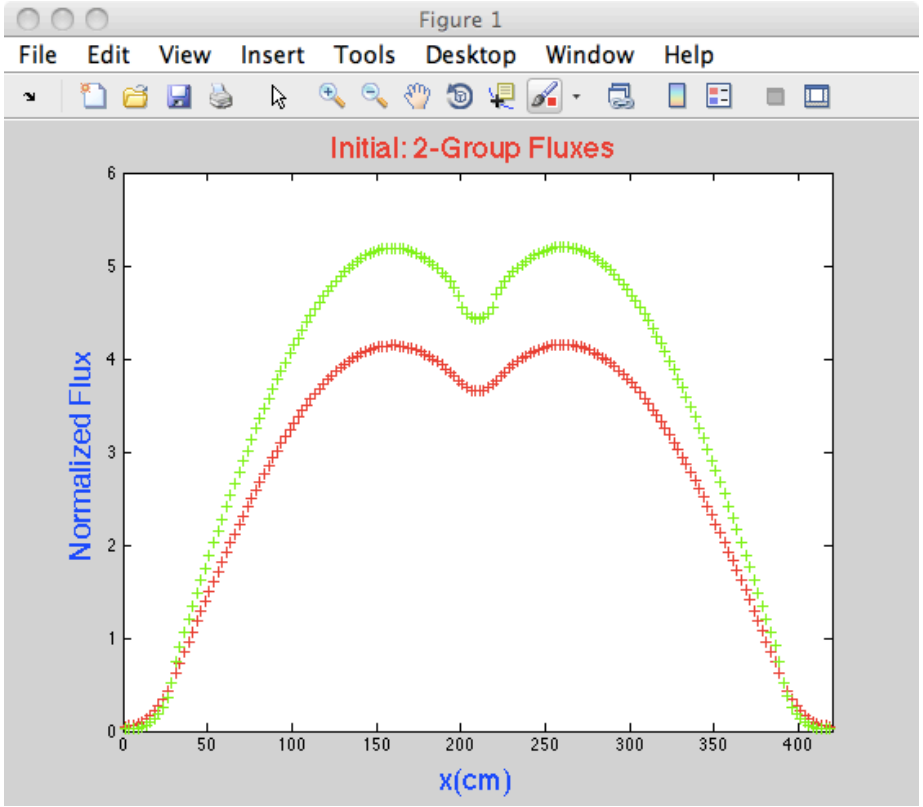
$$\rho_{static} = 0.37265\beta$$
$$\rho_{dynamic} = 0.37263\beta$$
$$\rho_{IK} = 0.37230\beta$$

Note: small time step sensitivity for IK's rho



PKE Parameters Dependence on Assumed Shape Function





Dynamic Fluxes

$$\rho_{static} = 0.37265\beta$$
$$\rho_{dynamic} = 0.37263\beta$$

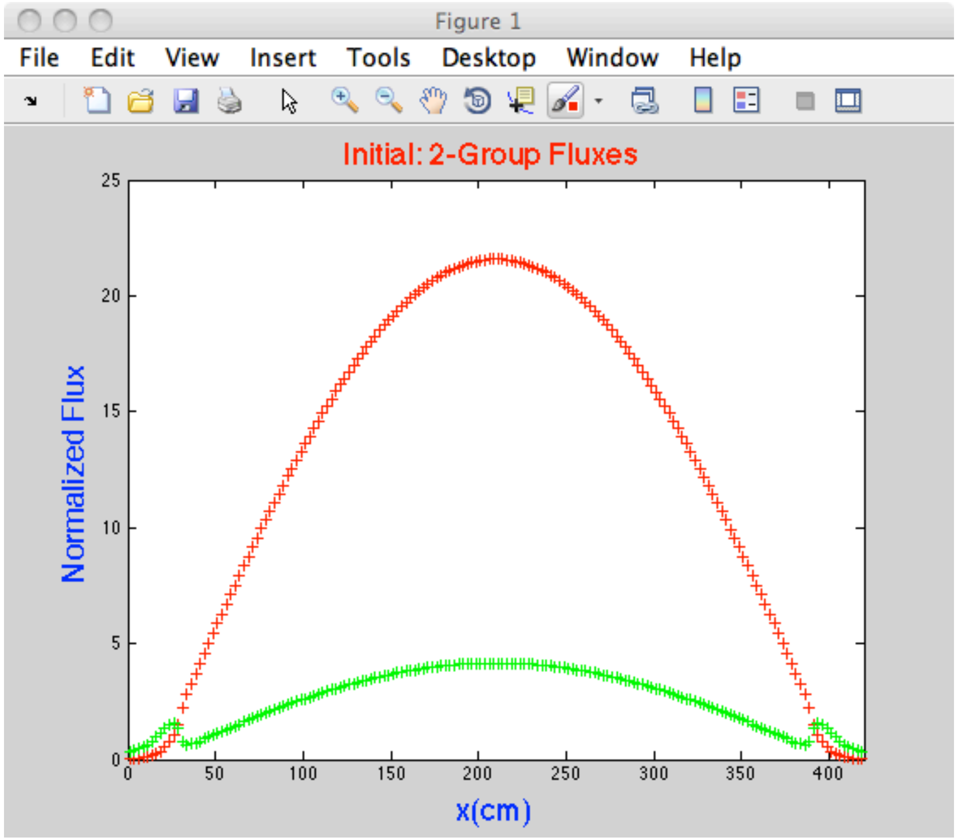
Rod-In Flux/Unity Adjoint

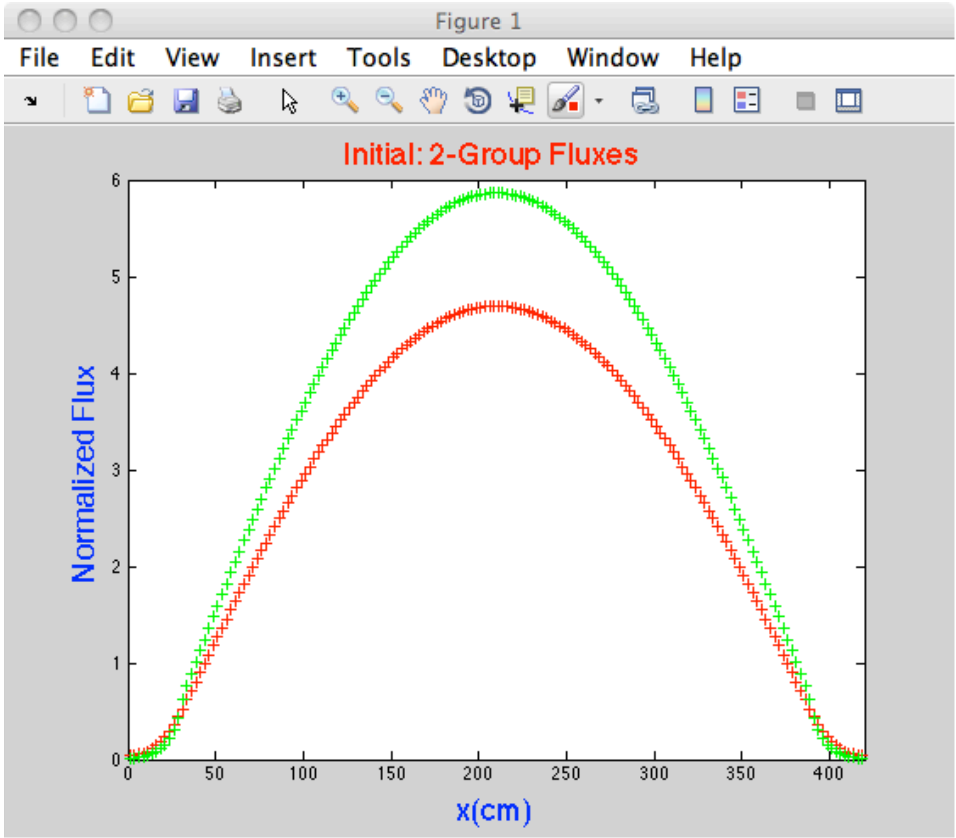
$$\rho_{static} = 0.37265\beta$$
$$\rho_{rod-in} = 0.22760\beta$$

Rod-in Flux and Adjoint

$$\rho_{static} = 0.37265\beta$$
$$\rho_{rod-in} = 0.29040\beta$$

PKE Parameters Dependence on Shape Function





Dynamic Fluxes

$$\rho_{static} = 0.37265\beta$$
$$\rho_{dynamic} = 0.37263\beta$$

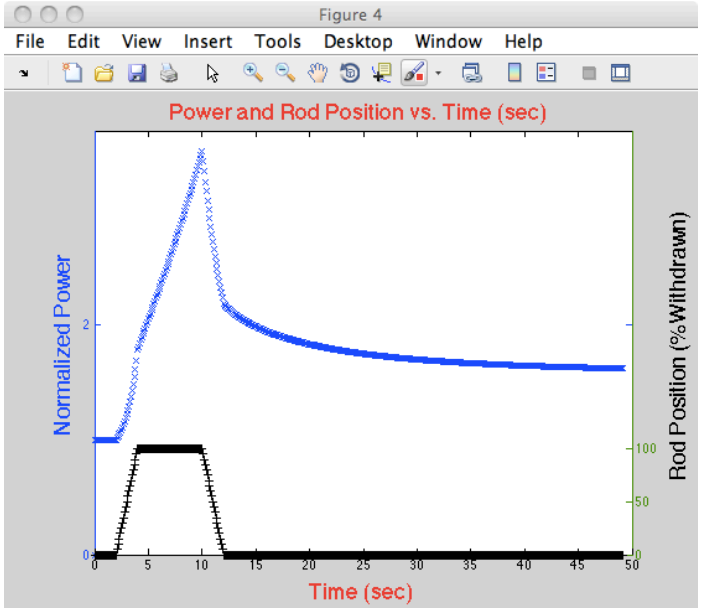
Rod-out Flux/Unity Adjoint

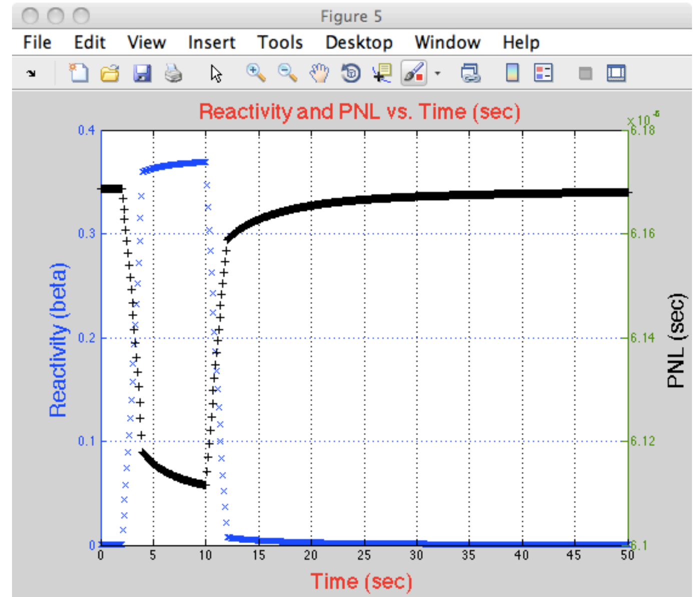
$$\rho_{static} = 0.37265\beta$$
$$\rho_{rod-out} = 0.37265\beta$$

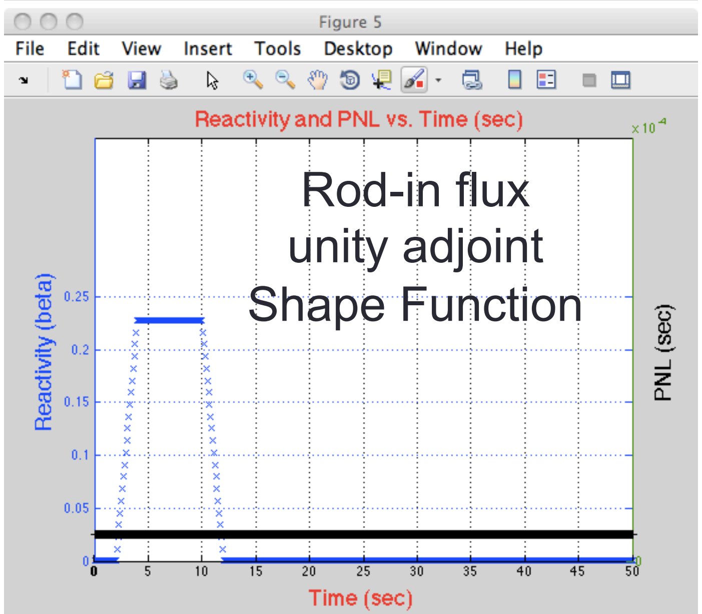
Rod-out Flux and Adjoint

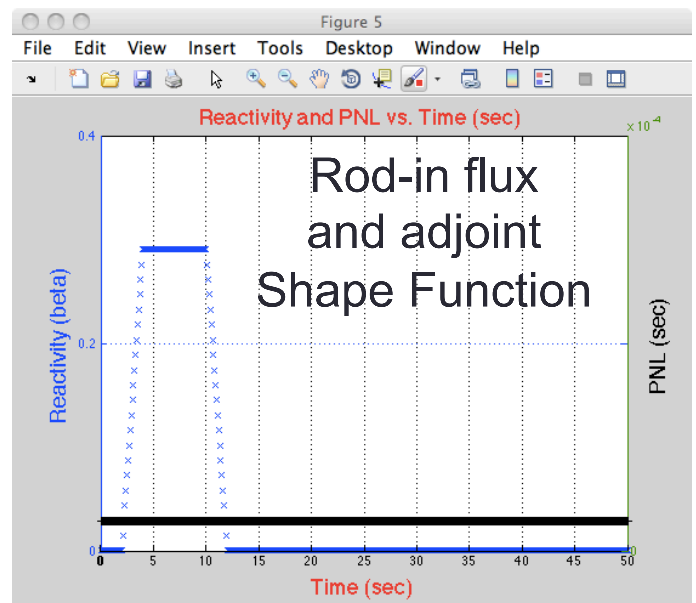
$$\rho_{static} = 0.37265\beta$$
$$\rho_{rod-out} = 0.37265\beta$$

Rod in/out Transient Edits of Point Point Kinetics Reactivity vs. Time

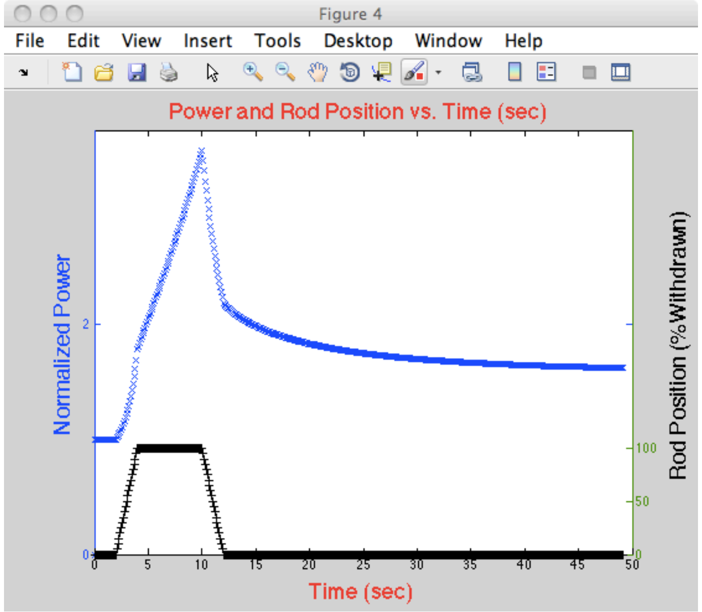


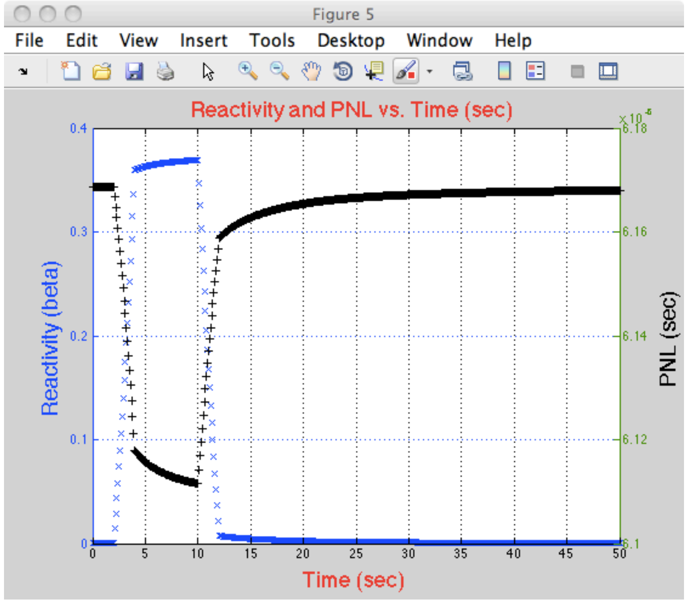


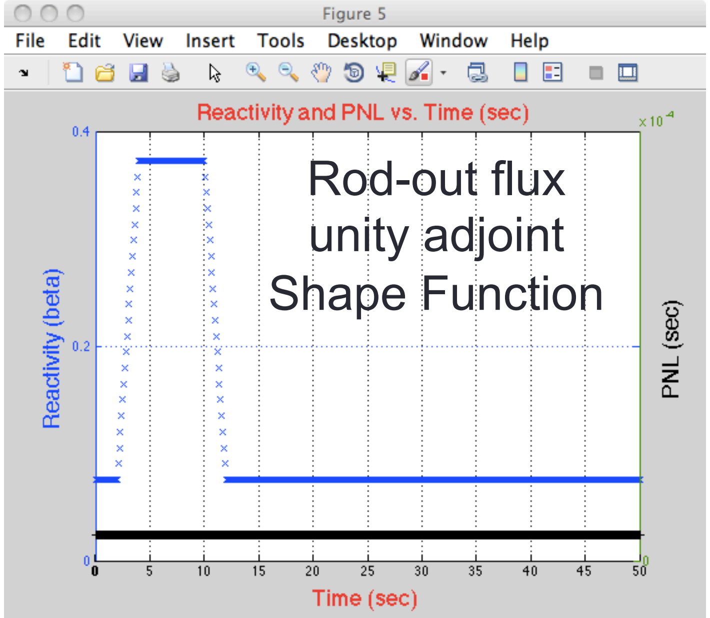


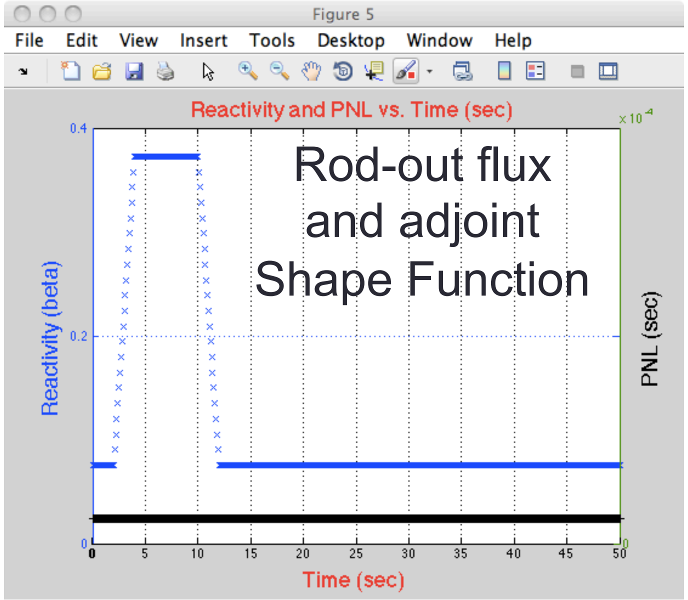


Rod in/out Transient Edits of Point Point Kinetics Reactivity vs. Time









Matrix 2-group PKE Diffusion Equations: Unity Weighting

We can get PKEs directly from the 2-group neutron balance equations:

$$\frac{d}{dt}\begin{bmatrix} \int d\vec{r}\, \frac{1}{v_1} S_1(\vec{r})T_1(t) \\ \int d\vec{r}\, \frac{1}{v_2} S_2(\vec{r})T_2(t) \end{bmatrix} = \begin{bmatrix} \int d\vec{r}\, \left[-\hat{\Sigma}_{r,1}(\vec{r},t) + \left[1-\beta(\vec{r})\right]\nu\Sigma_{f,1}(\vec{r},t)\right] S_1(\vec{r})T_1(t) & \int d\vec{r}\left[\left[1-\beta(\vec{r})\right]\nu\Sigma_{f,2}(\vec{r},t)\right]S_2(\vec{r})T_2(t) \\ \int d\vec{r}\, \Sigma_{s,1\to 2}(\vec{r},t)\, S_1(\vec{r})T_1(t) & \int d\vec{r}\left[-\Sigma_{a,2}(\vec{r},t)\right] S_2(\vec{r})T_2(t) \end{bmatrix} + \begin{bmatrix} \int d\vec{r}\, \left[\sum_i \lambda_i C_i(\vec{r},t)\right] \\ 0 \end{bmatrix}$$

$$\frac{d}{dt}\left[\int d\vec{r}\;\; C_i(\vec{r},t)\right] = \begin{bmatrix} \int d\vec{r}\, \beta_i(\vec{r})\nu\Sigma_{f,1}(\vec{r},t)S_1(\vec{r})T_1(t) & \int d\vec{r}\, \beta_i(\vec{r})\nu\Sigma_{f,2}(\vec{r},t)S_2(\vec{r})T_1(t) \\ 0 & 0 \end{bmatrix} - \lambda_i \int d\vec{r}\;\; C_i(\vec{r},t), \quad i = 1, I$$

Slightly simplifying with space-independent beta, lambda, and velocities (not necessary for the derivation):

$$\begin{bmatrix} \frac{1}{v_1}\int d\vec{r}\, S_1(\vec{r}) & 0 \\ 0 & \frac{1}{v_2}\int d\vec{r}\, S_2(\vec{r}) \end{bmatrix} \frac{d}{dt}\begin{bmatrix} T_1(t) \\ T_2(t) \end{bmatrix} = \begin{bmatrix} \int d\vec{r}\, \left[-\hat{\Sigma}_{r,1}(\vec{r},t) + \left(1-\beta\right)\nu\Sigma_{f,1}(\vec{r},t)\right] S_1(\vec{r}) & \int d\vec{r}\left[\left(1-\beta\right)\nu\Sigma_{f,2}(\vec{r},t)\right]S_2(\vec{r}) \\ \int d\vec{r}\, \Sigma_{s,1\to 2}(\vec{r},t)\, S_1(\vec{r}) & \int d\vec{r}\left[-\Sigma_{a,2}(\vec{r},t)\right] S_2(\vec{r}) \end{bmatrix}\begin{bmatrix} T_1(t) \\ T_2(t) \end{bmatrix} + \begin{bmatrix} \left[\sum_i \lambda_i \int d\vec{r}\, C_i(\vec{r},t)\right] \\ 0 \end{bmatrix}$$

$$\frac{d}{dt}\left[\int d\vec{r}\, C_i(\vec{r},t)\right] = \beta_i \begin{bmatrix} \int d\vec{r}\, \nu\Sigma_{f,1}(\vec{r},t)S_1(\vec{r}) & \int d\vec{r}\, \nu\Sigma_{f,2}(\vec{r},t)S_2(\vec{r}) \\ 0 & 0 \end{bmatrix}\begin{bmatrix} T_1(t) \\ T_2(t) \end{bmatrix} - \lambda_i \int d\vec{r}\, C_i(\vec{r},t) \quad ,i = 1, I$$

Matrix 2-group PKE Diffusion Equations: Adjoint Weighting

Collecting terms and recognizing the amplitude function is not space-dependent:

$$\begin{bmatrix} \frac{1}{v_1}\int d\vec{r}\,\phi_1^*(\vec{r})\,S_1(\vec{r}) & 0 \\ 0 & \frac{1}{v_2}\int d\vec{r}\phi_2^*(\vec{r})\;S_2(\vec{r}) \end{bmatrix} \frac{d}{dt}\begin{bmatrix} T_1(t) \\ T_2(t) \end{bmatrix} = \begin{bmatrix} \left[\sum_i \lambda_i \int d\vec{r}\,\phi_1^*(\vec{r})\,C_i(\vec{r},t)\right] \\ 0 \end{bmatrix} +$$

$$\begin{bmatrix} \int d\vec{r}\,\phi_1^*(\vec{r})\left[-\hat{\Sigma}_{r,1}(\vec{r},t)+(1-\beta)\nu\Sigma_{f,1}(\vec{r},t)\right]S_1(\vec{r}) & \int d\vec{r}\,\phi_1^*(\vec{r})\left[(1-\beta)\nu\Sigma_{f,2}(\vec{r},t)\right]S_2(\vec{r}) \\ \int d\vec{r}\,\phi_2^*(\vec{r})\,\Sigma_{s,1\to 2}(\vec{r},t)\,S_1(\vec{r}) & \int d\vec{r}\,\phi_2^*(\vec{r})\left[-\Sigma_{a,2}(\vec{r},t)\right]S_2(\vec{r}) \end{bmatrix}\begin{bmatrix} T_1(t) \\ T_2(t) \end{bmatrix}$$

$$\frac{d}{dt}\left[\int d\vec{r}\,\phi_1^*(\vec{r})\,C_i(\vec{r},t)\right] = \beta_i \begin{bmatrix} \int d\vec{r}\,\phi_1^*(\vec{r})\,\nu\Sigma_{f,1}(\vec{r},t)S_1(\vec{r}) & \int d\vec{r}\,\phi_1^*(\vec{r})\,\nu\Sigma_{f,2}(\vec{r},t)S_2(\vec{r}) \\ 0 & 0 \end{bmatrix}\begin{bmatrix} T_1(t) \\ T_2(t) \end{bmatrix} - \lambda_i \int d\vec{r}\,\phi_1^*(\vec{r})\,C_i(\vec{r},t)\,, i=1, I$$

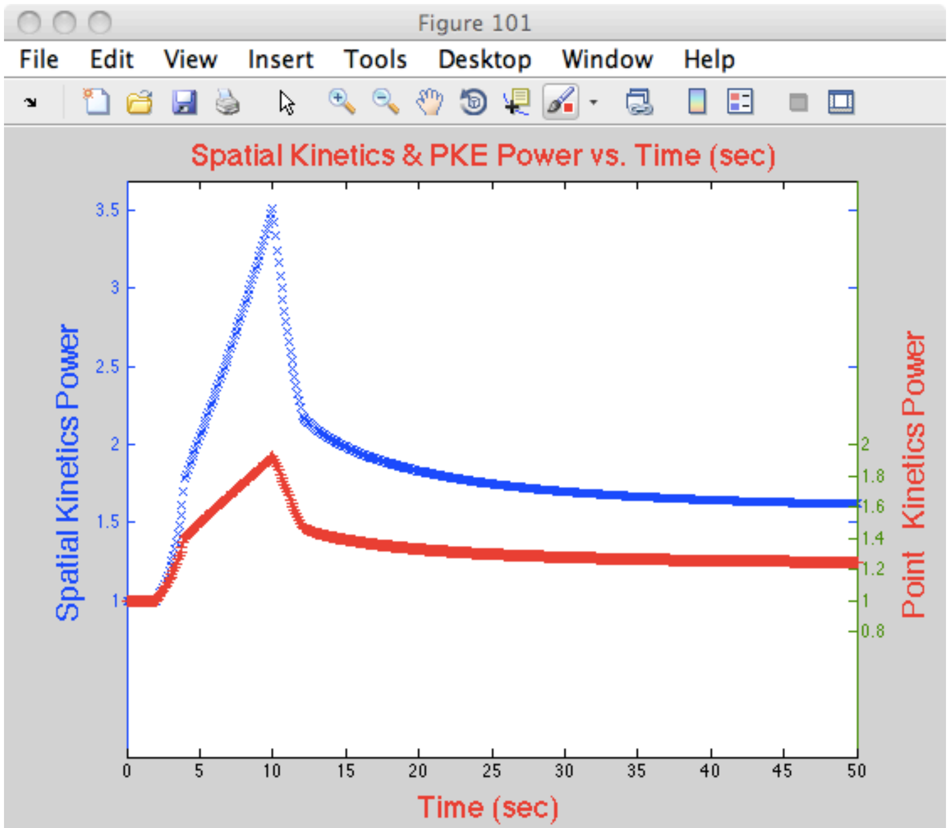
Simplifying:

$$\frac{d}{dt}\begin{bmatrix} T_1(t) \\ T_2(t) \end{bmatrix} = \begin{bmatrix} \frac{v_1}{\int d\vec{r}\,\phi_1^*(\vec{r})\,S_1(\vec{r})}\left[\sum_i \lambda_i \int d\vec{r}\,\phi_1^*(\vec{r})\,C_i(\vec{r},t)\right] \\ 0 \end{bmatrix} +$$

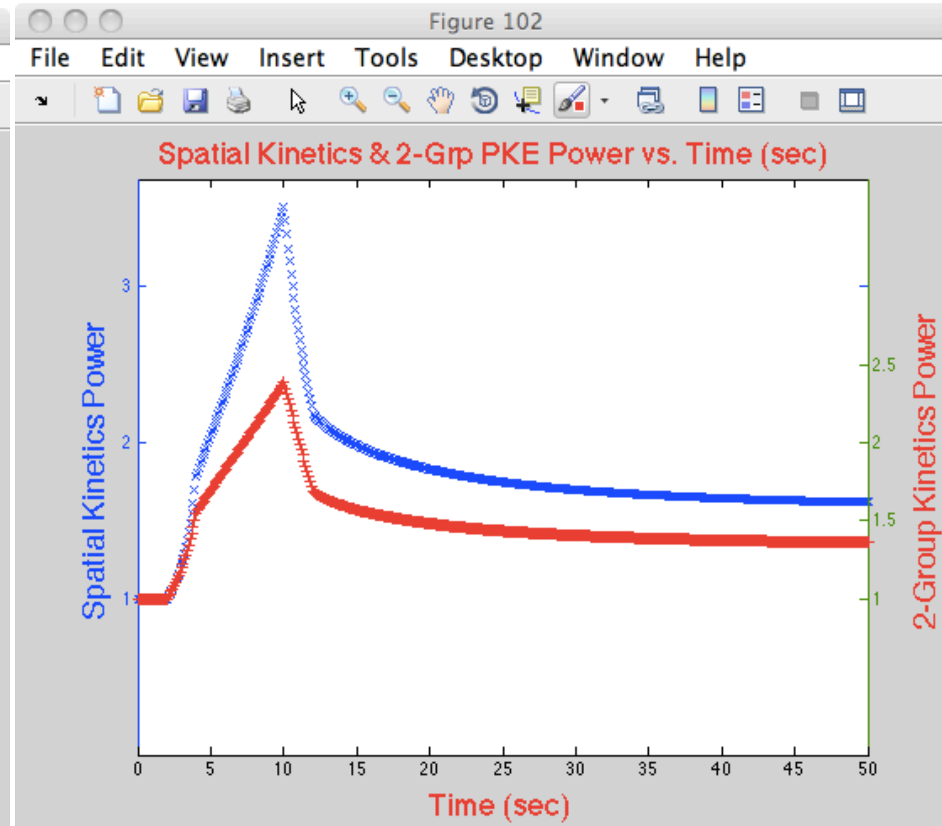
$$\begin{bmatrix} \frac{v_1}{\int d\vec{r}\,\phi_1^*(\vec{r})\,S_1(\vec{r})} & 0 \\ 0 & \frac{v_2}{\int d\vec{r}\,\phi_2^*(\vec{r})\,S_2(\vec{r})} \end{bmatrix}\begin{bmatrix} \int d\vec{r}\,\phi_1^*(\vec{r})\left[-\hat{\Sigma}_{r,1}(\vec{r},t)+(1-\beta)\nu\Sigma_{f,1}(\vec{r},t)\right]S_1(\vec{r}) & \int d\vec{r}\,\phi_1^*(\vec{r})\left[(1-\beta)\nu\Sigma_{f,2}(\vec{r},t)\right]S_2(\vec{r}) \\ \int d\vec{r}\,\phi_2^*(\vec{r})\,\Sigma_{s,1\to 2}(\vec{r},t)\,S_1(\vec{r}) & \int d\vec{r}\,\phi_2^*(\vec{r})\left[-\Sigma_{a,2}(\vec{r},t)\right]S_2(\vec{r}) \end{bmatrix}\begin{bmatrix} T_1(t) \\ T_2(t) \end{bmatrix}$$

$$\frac{d}{dt}\left[\int d\vec{r}\,\phi_1^*(\vec{r})\,C_i(\vec{r},t)\right] = \beta_i \begin{bmatrix} \int d\vec{r}\,\phi_1^*(\vec{r})\,\nu\Sigma_{f,1}(\vec{r},t)S_1(\vec{r}) & \int d\vec{r}\,\phi_1^*(\vec{r})\,\nu\Sigma_{f,2}(\vec{r},t)S_2(\vec{r}) \\ 0 & 0 \end{bmatrix}\begin{bmatrix} T_1(t) \\ T_2(t) \end{bmatrix} - \lambda_i \int d\vec{r}\,\phi_1^*(\vec{r})\,C_i(\vec{r},t)\,, i=1, I$$

PKE solutions: Rod-In Shape Function

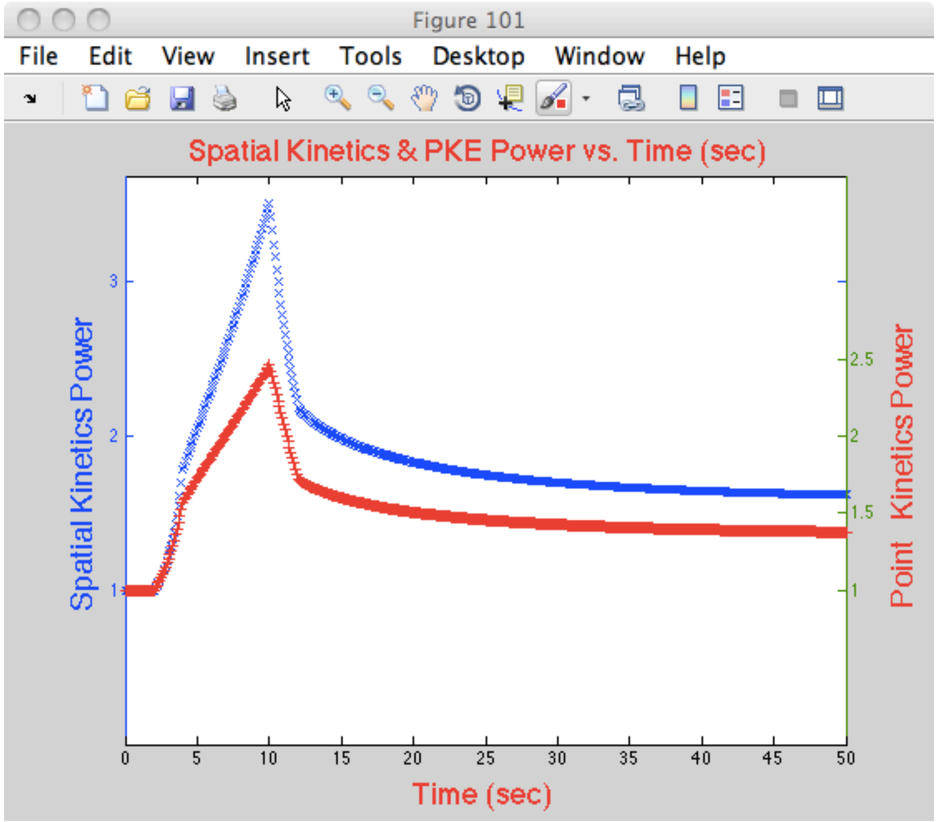


Under predicts rod out reactivity

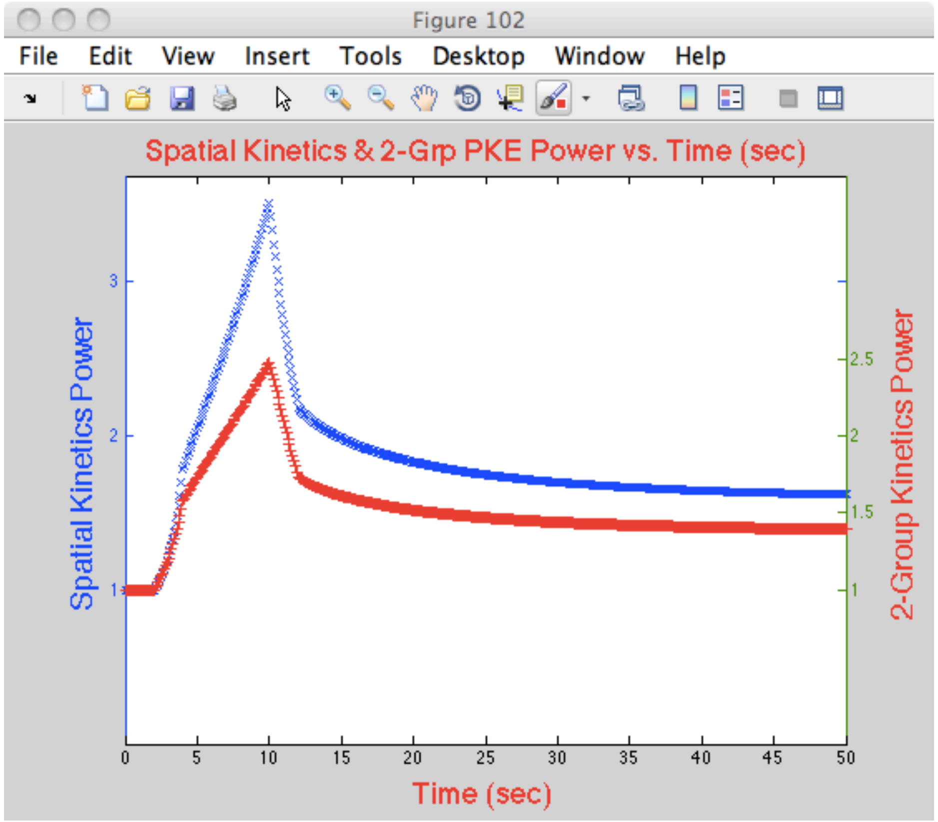


Improved but still under predicts rod out reactivity

PKE solutions: Rod-In Shape Function & Adjoint

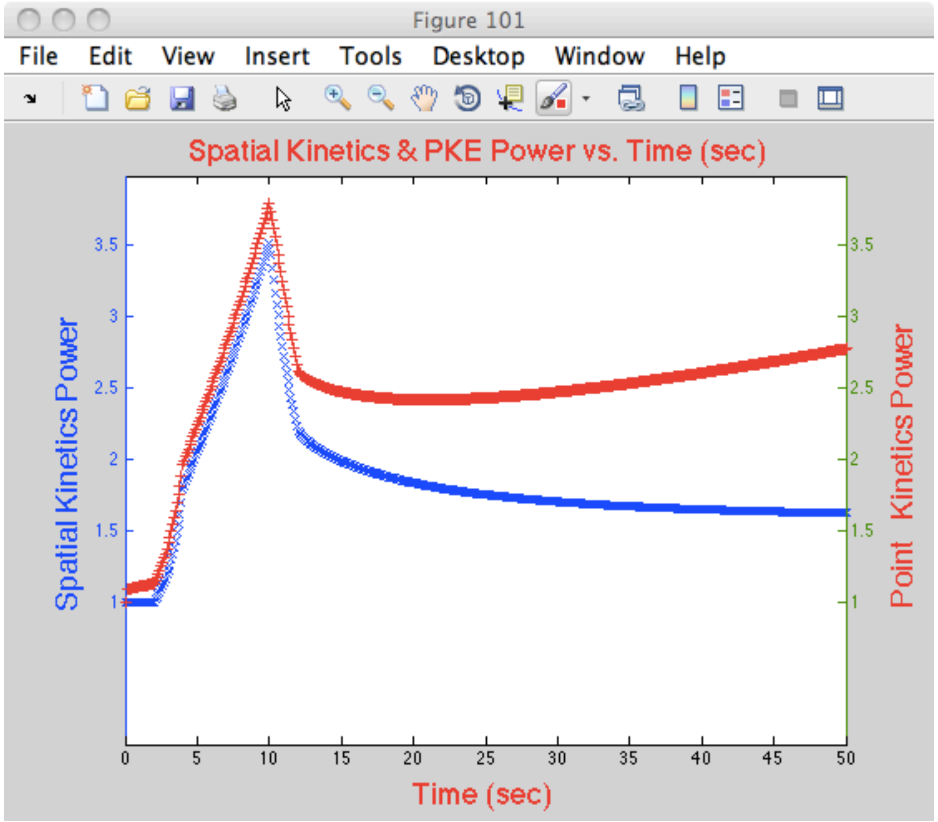


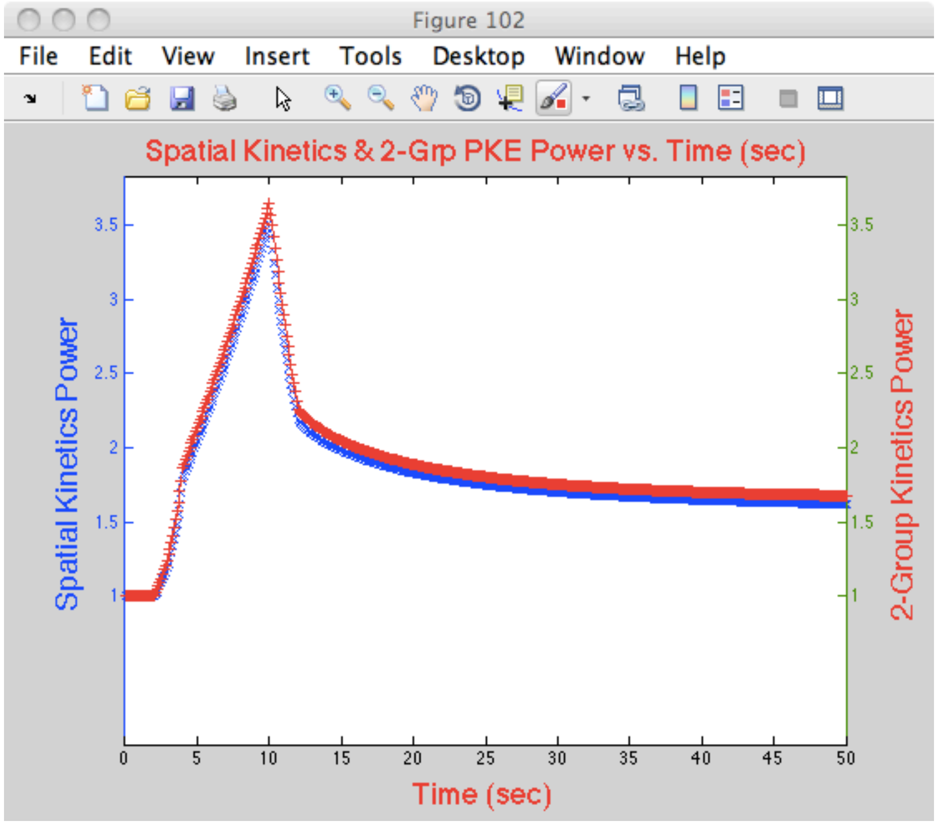
Adjoint significantly improves reactivity



Adjoint barely improves reactivity

PKE solutions: Rod-out Shape Function

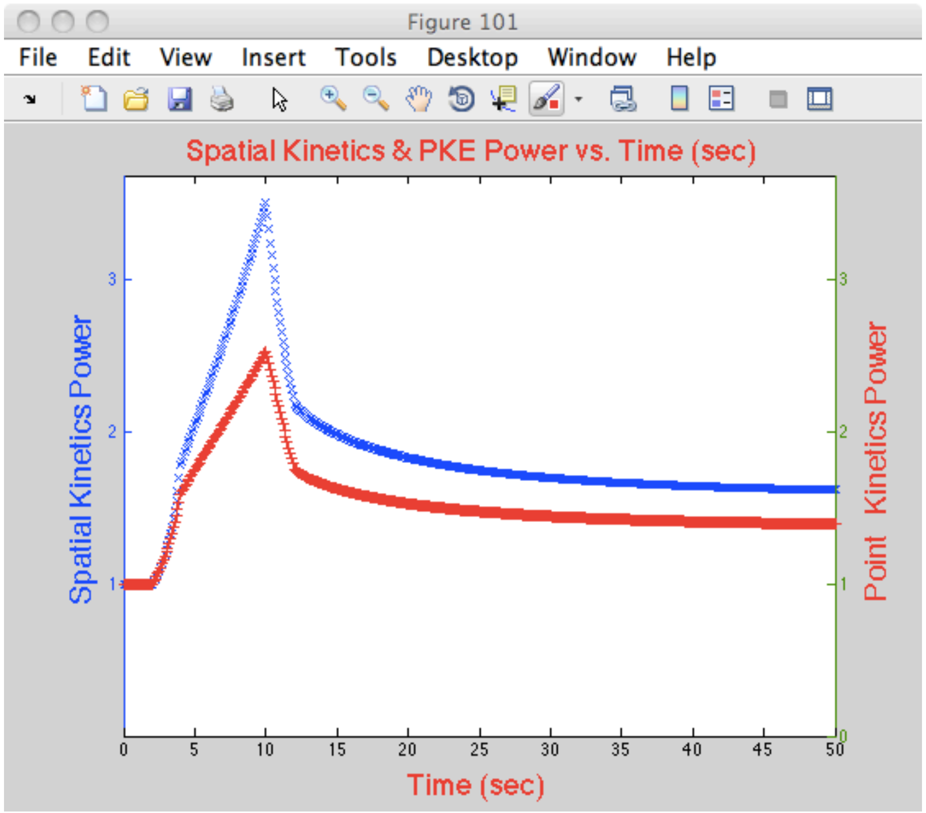


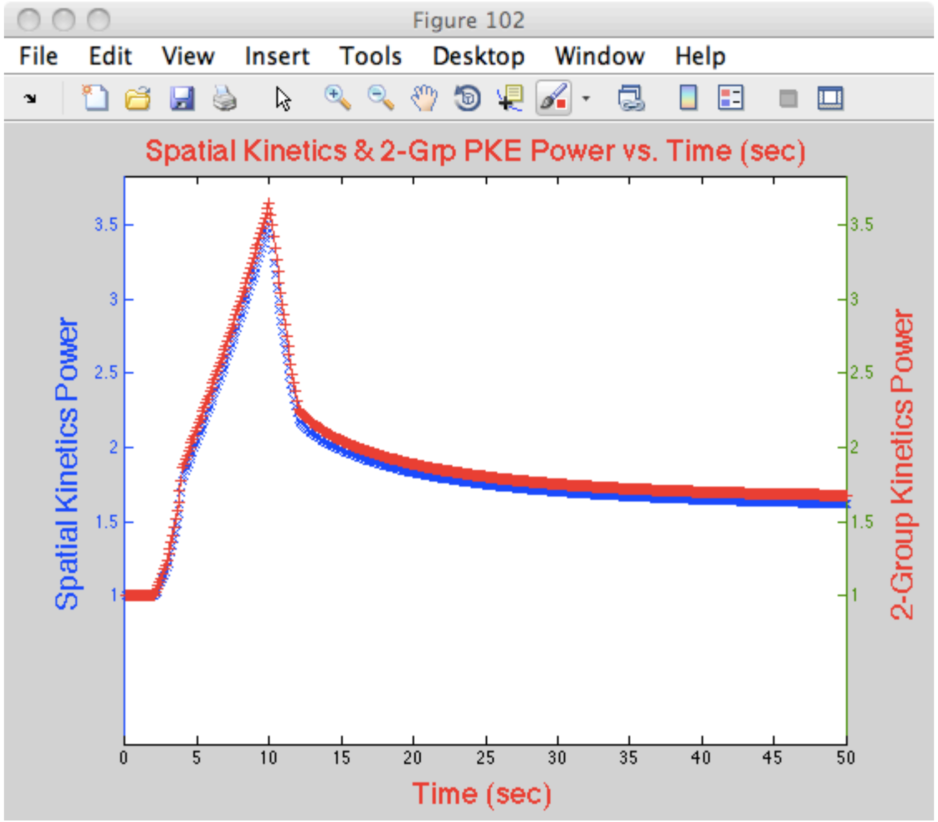


Does not hold steady-state
Nice transient shape while rod out
Fails to return to critical

Holds steady-state
Nice transient shape while rod out
Returns to critical

PKE solutions: Rod-out Shape Function & Step Renormalized Rho

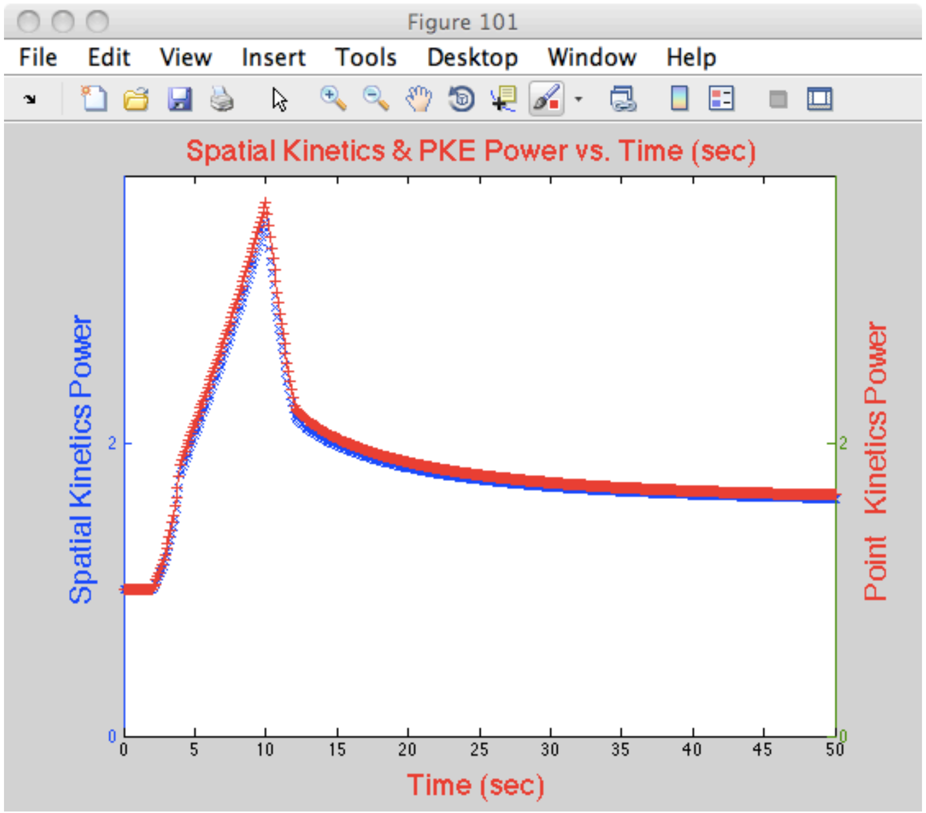


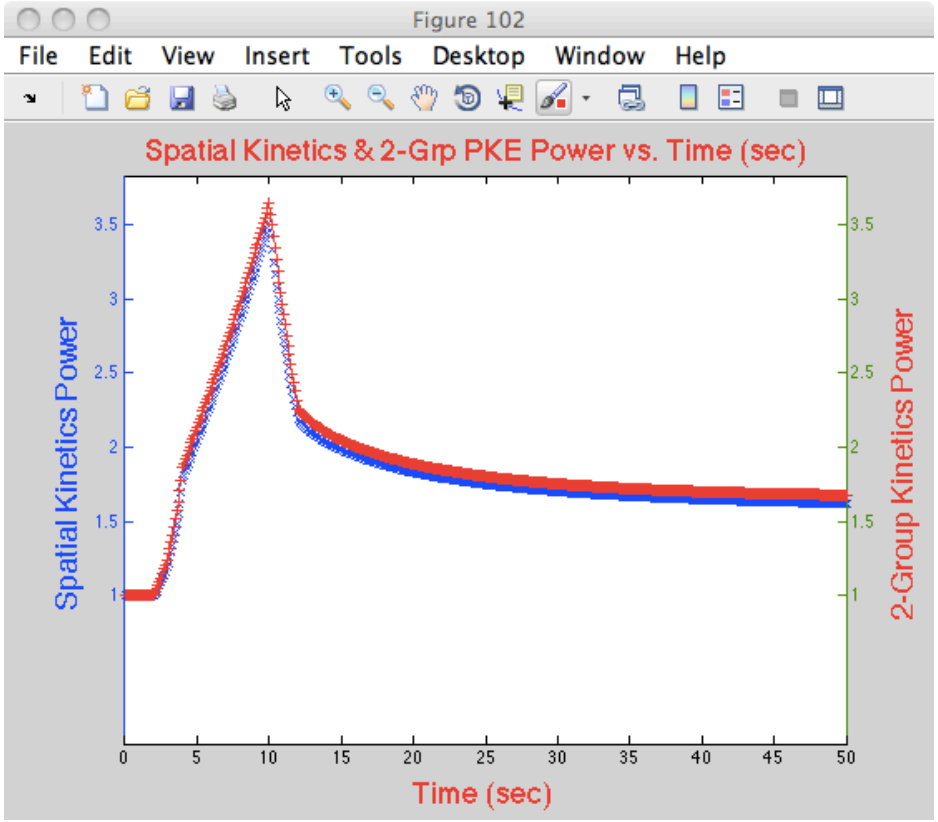


Holds steady-state
Transient shape wrong when rod is out
Returns to critical

Holds steady-state
Nice transient shape while rod out
Returns to critical

PKE solutions: Rod-out Shape Function & Bryan's Stretch Renormalization

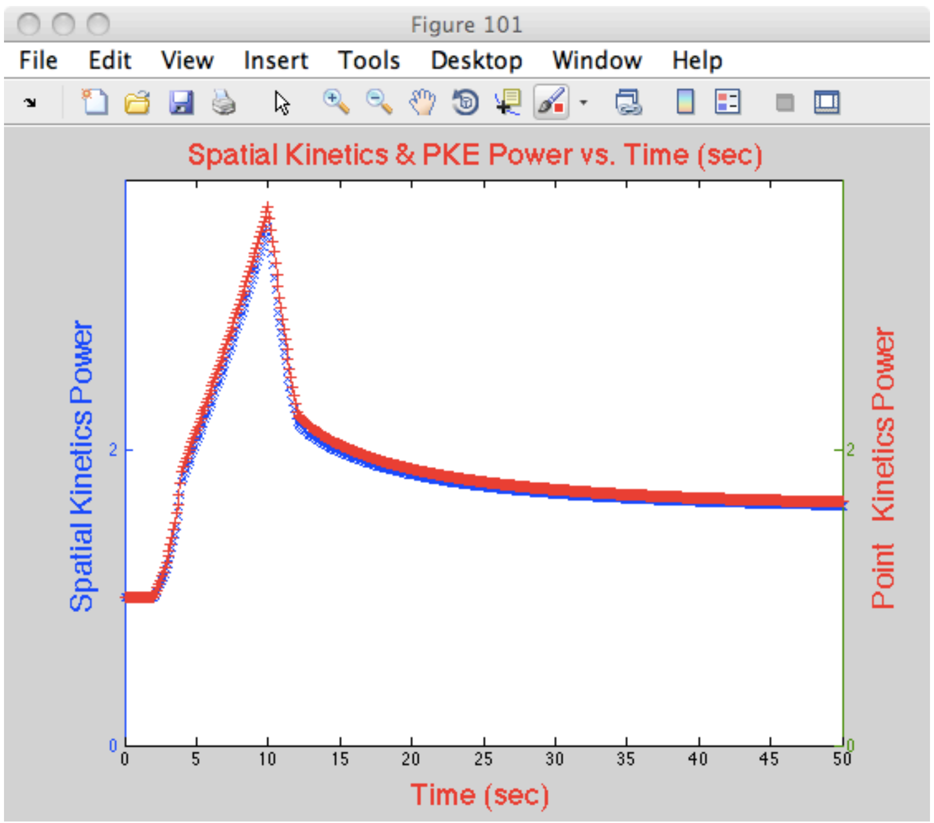


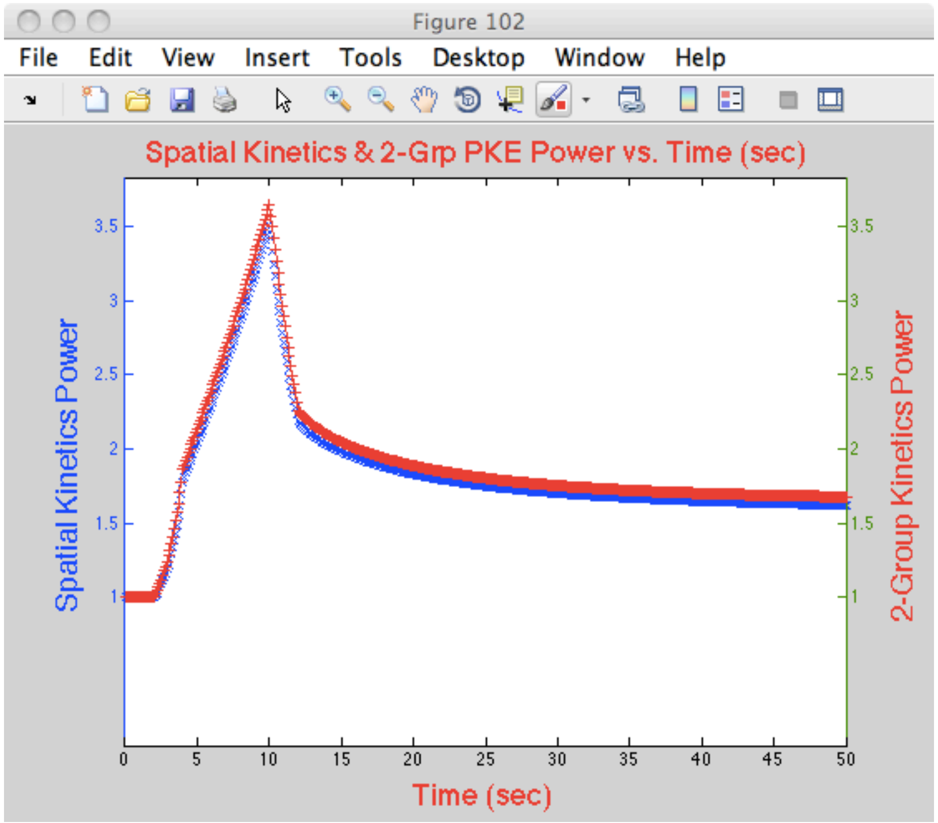


Holds steady-state
Nice transient shape while rod out
Returns to critical

Holds steady-state
Nice transient shape while rod out
Returns to critical

PKE solution: Rod-out Shape Function/Adjoint, Bryan's Stretch Renormalization

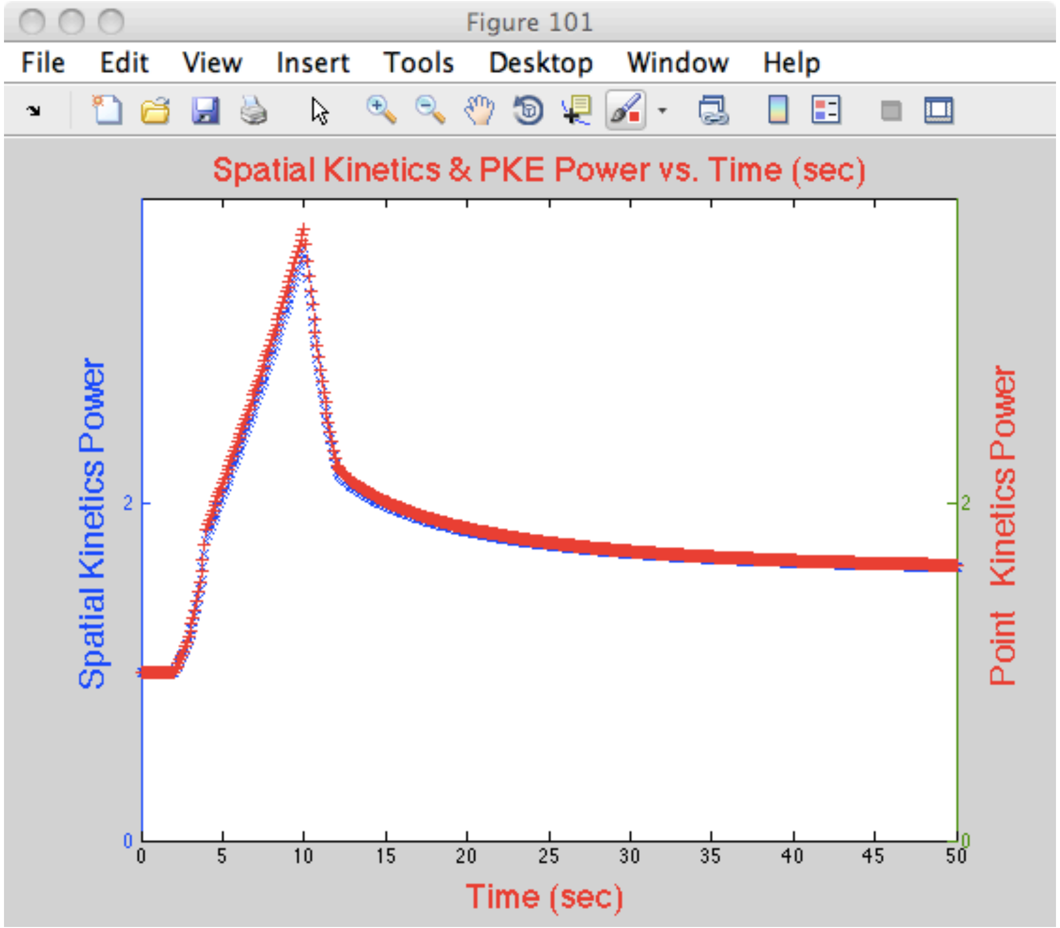




No change with Adjoint

Holds steady-state
Nice transient shape while rod out
Returns to critical

PKE solutions: Interpolation of Shape Function With Rod Insertion Fraction



Bottom Line

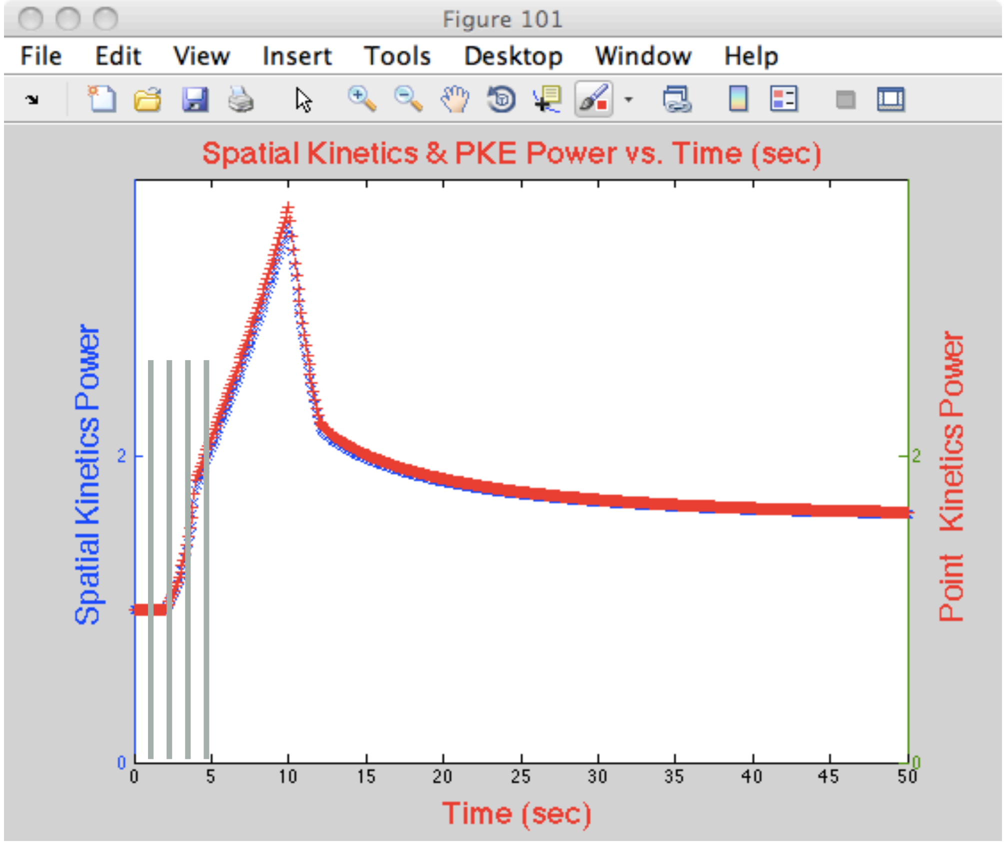
1. We need time-dependent shape functions
2. We are always better off to use multi-group PKEs

How Do We Get Time-dependent Shape Functions?

- Fixed k-eigenvalue shapes (BOT, EOT, etc.)
- Geometrical interpolation between k-eigenvalue shape function
- Adiabatic shape functions during transient
- α-eigenmode shape functions during transient
- ω-mode shape functions during transient
- Quasi-static shape functions during transient
- Improved Quasi-static shape functions during transients
- Coarse time integration of shape functions during transient
- Frequency-transform time integration of shape functions during transient

Adiabatic Method

- compute k-eigenvalue shapes periodically as you solve PKEs



- If you have thermal feedback, this too will change shape functions
- Linearly interpolate between shape functions

α-mode method

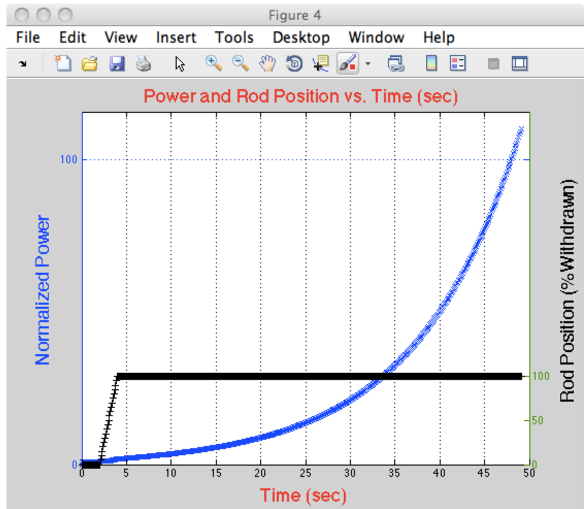
Use asymptotic period to get pseudo-steady-state equations to compute α-eigenmode shapes periodically as you solve PKEs

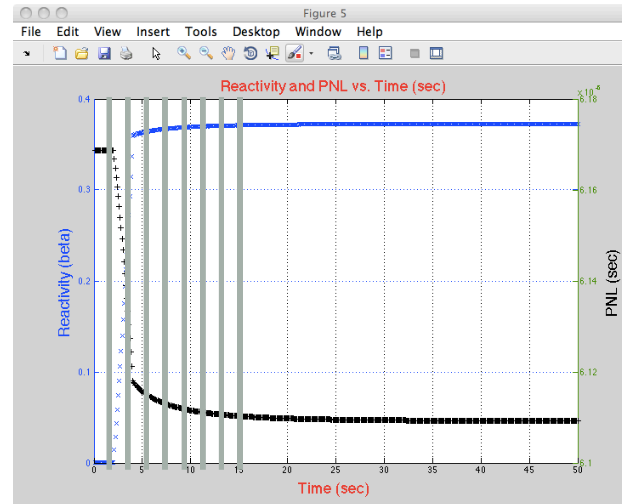
Static and Dynamic Criticality: Are they different?

by
Dermott E. Cullen
Christopher J. Clouse
Richard Procassini
Lawrence Livermore National Laboratory
and
Robert C. Little
Los Alamos National Laboratory

contact
Dermott E. Cullen
University of California
Lawrence Livermore National Laboratory
L-159
P.O. Box 808
Livermore, CA 94550
E.Mail: cullen1@llnl.gov
Website: http://www.llnl.gov/cullen1

November 22, 2003





$$\frac{\partial n}{\partial t} + \Omega * \nabla(v * n) + \Sigma t * (v * n) = \iint [< v > * \Sigma f + \Sigma s] F(E', \Omega' -> E, \Omega)(v * n) dE' d\Omega' + S$$

Assume an asymptotic behavior of fluxes $n(R,\Omega,E,t) = n(R,\Omega,E) * Exp[\alpha * t]$

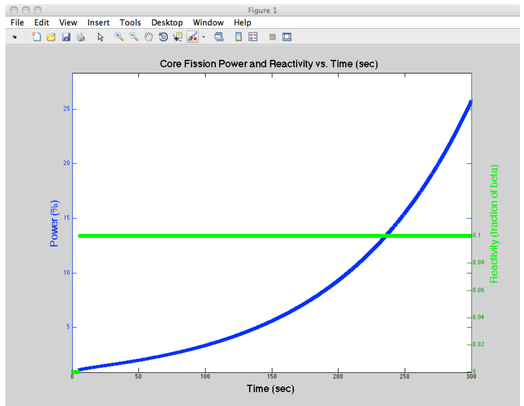
For a **super-critical** system ($\alpha > 0$) the lead term is combined with the cross section term to give us the $\frac{\alpha}{v}$ **time absorption** equation,

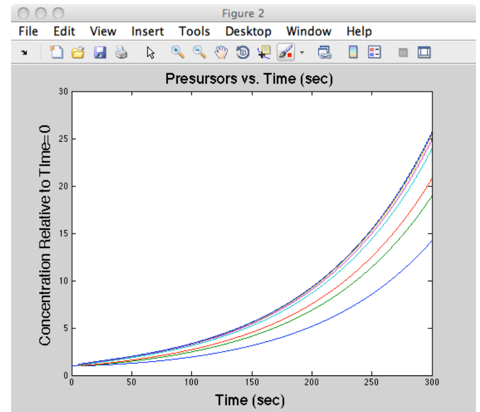
$$\Omega * \nabla N + [\Sigma t + \frac{\alpha}{v}] * N = \iint [< v > * \Sigma f + \Sigma s] F(E', \Omega' -> E, \Omega) N dE' d\Omega' \qquad [15]$$

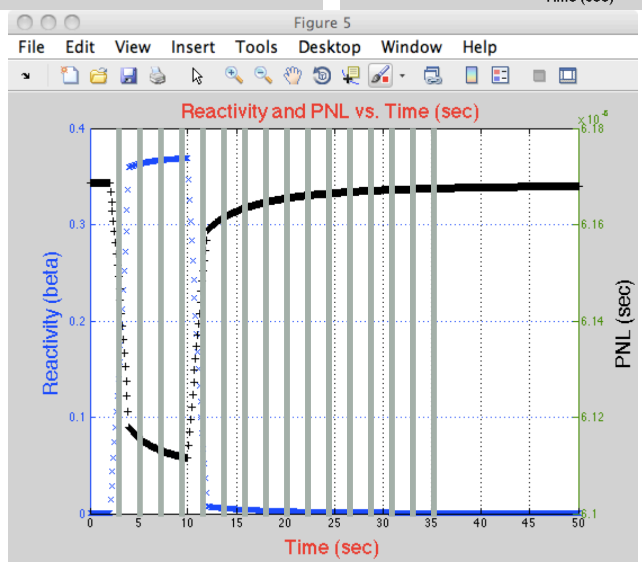
α-eigenmode shape function is different from a k-eigenmode shape function (easy to do just search for α that makes the system critical)

ω-mode Method

Use secular equilibrium concept to get pseudo-steady-state equations to compute ω-eigenmode shapes periodically as you solve PKEs







ω-eigenmode shape function is different from a k-eigenmode shape function (easy to do just search for ω that makes the system critical)

ω-mode Method

$$\frac{\partial}{\partial t}\left\langle \frac{1}{v_g(\vec{r},t)}\phi_g(\vec{r},t)\right\rangle = \nabla\cdot D_g(\vec{r},t)\nabla\phi_g(\vec{r},t) - \Sigma_{r,g}(\vec{r},t)\,\phi_g(\vec{r},t) + \sum_{g'\neq g}^{G}\Sigma_{s,g'\to g}(\vec{r},t)\,\phi_{g'}(\vec{r},t)$$

$$+[1-\beta(\vec{r},t)]\frac{\chi_g^p}{k_{crit}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r},t)\,\phi_{g'}(\vec{r},t) + \sum_{i}^{I}\chi_{i,g}^d\,\lambda_i C_i(\vec{r},t), \quad g=1,\dots,G$$

$$\frac{\partial}{\partial t}C_i(\vec{r},t) = -\lambda_i C_i(\vec{r},t) + \frac{\beta_i(\vec{r},t)]}{k_{crit}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r},t)\,\phi_{g'}(\vec{r},t), \quad i=1,\dots,I$$

Assume

$$\frac{\partial}{\partial t}\phi_g(\vec{r},t) = \omega\, S_g(\vec{r}) \qquad \frac{\partial}{\partial t}C_i(\vec{r},t) = \omega\, C_i(\vec{r})$$

$$\frac{\omega}{v_g}\frac{\partial}{\partial t}\left\langle S_g(\vec{r})\right\rangle = \nabla\cdot D_g(\vec{r})\nabla S_g(\vec{r}) - \Sigma_{r,g}(\vec{r})\,S_g(\vec{r}) + \sum_{g'\neq g}^{G}\Sigma_{s,g'\to g}(\vec{r})\,S_{g'}(\vec{r})$$

$$+[1-\beta]\frac{\chi_g^p}{k_{crit}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r})\,S_{g'}(\vec{r}) + \sum_{i}^{I}\chi_{i,g}^d\,\lambda_i C_i(\vec{r}), \quad g=1,\dots,G$$

$$\omega\, C_i(\vec{r}) = -\lambda_i C_i(\vec{r}) + \frac{\beta_i]}{k_{crit}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r})\,S_{g'}(\vec{r}), \quad i=1,\dots,I$$

ω-mode Method

- Collecting terms, we get

$$-\nabla\cdot D_g(\vec{r})\nabla S_g(\vec{r})+\left[\Sigma_{r,g}(\vec{r})+\frac{\omega}{v_g}\right]S_g(\vec{r})-\sum_{g'\neq g}^{G}\Sigma_{s,g'\to g}(\vec{r})\,S_{g'}(\vec{r})$$
$$=\left[[1-\beta]\frac{\chi_g^p}{k_{crit}}+\sum_i^I\frac{\chi_{i,g}^d}{k_{crit}}\frac{\beta_i\lambda_i}{\omega+\lambda_i}\right]\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r})\,S_{g'}(\vec{r}),\qquad g=1,\ldots,G$$

- Which is just a modified steady-state eigenvalue equation
- Search for ω that makes system critical

- Modified cross sections similar to those from 2-group fully-implicit difference case

$$\hat{\Sigma}_{r,1}(\vec{r})\equiv\left\langle\Sigma_{r,1}(\vec{r})+\frac{1}{v_1(\vec{r})\Delta_t}\right\rangle\qquad\hat{\Sigma}_{r,2}(\vec{r})\equiv\left\langle\Sigma_{a,2}(\vec{r})+\frac{1}{v_2(\vec{r})\Delta_t}\right\rangle$$
$$\upsilon\hat{\Sigma}_{f,g}(\vec{r})\equiv\left(\frac{[1-\beta(\vec{r})]}{k_{crit}}+\sum_i^I\frac{\beta_i(\vec{r})\lambda_i\Delta_t}{(1+\lambda_i\Delta_t)k_{crit}}\right)\upsilon\Sigma_{f,g}(\vec{r})$$

- This is a more general version of the α-eigenmode equations
- More useful than α-eigenmode when the reactor is delayed critical or subcritical

An Improved ω-mode Method

$$\frac{\partial}{\partial t}\left\langle \frac{1}{v_g(\vec{r},t)}\phi_g(\vec{r},t)\right\rangle = \nabla\cdot D_g(\vec{r},t)\nabla\phi_g(\vec{r},t) - \Sigma_{r,g}(\vec{r},t)\,\phi_g(\vec{r},t) + \sum_{g'\neq g}^{G}\Sigma_{s,g'\to g}(\vec{r},t)\,\phi_{g'}(\vec{r},t)$$

$$+[1-\beta(\vec{r},t)]\frac{\chi_g^p}{k_{crit}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r},t)\,\phi_{g'}(\vec{r},t) + \sum_{i}^{I}\chi_{i,g}^d\,\lambda_i C_i(\vec{r},t), \qquad g=1,\ldots,G$$

$$\frac{\partial}{\partial t}C_i(\vec{r},t) = -\lambda_i C_i(\vec{r},t) + \frac{\beta_i(\vec{r},t)]}{k_{crit}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r},t)\,\phi_{g'}(\vec{r},t)\,, \qquad i=1,\ldots,I$$

Assume

$$\frac{\partial}{\partial t}\phi_g(\vec{r},t) = \omega_g\, S_g(\vec{r}) \qquad \frac{\partial}{\partial t}C_i(\vec{r},t) = \omega_i\, C_i(\vec{r})$$

$$-\nabla\cdot D_g(\vec{r})\nabla S_g(\vec{r}) + \left[\Sigma_{r,g}(\vec{r}) + \frac{\omega_g}{v_g}\right]S_g(\vec{r}) - \sum_{g'\neq g}^{G}\Sigma_{s,g'\to g}(\vec{r})\,S_{g'}(\vec{r})$$

$$= \left[[1-\beta]\frac{\chi_g^p}{k_{crit}} + \sum_{i}^{I}\frac{\chi_{i,g}^d}{k_{crit}}\frac{\beta_i\lambda_i}{\omega_g+\lambda_i}\right]\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r})\,S_{g'}(\vec{r})\,, \qquad g=1,\ldots,G$$

- Use ω_g and ω_i from PKE time integration, and solve k-eigenvalue problem (or solve for a single multiplier on the prompt omegas)
- If omegas are correct, k_{eff} will be unity.
- Much more useful when the reactor is not yet asymptotic

Quasi-Static Methods for Shape Functions

$$\frac{1}{v_g(\vec{r},t)}\frac{\partial}{\partial t}\left\langle S_g(\vec{r},t)T(t)\right\rangle = \nabla\cdot D_g(\vec{r},t)\nabla S_g(\vec{r},t)T(t) - \Sigma_{r,g}(\vec{r},t)\,S_g(\vec{r},t)T(t) + \sum_{g'\neq g}^{G}\Sigma_{s,g'\to g}(\vec{r},t)\,S_{g'}(\vec{r},t)T(t)$$
$$+[1-\beta(\vec{r},t)]\frac{\chi_g^p}{k_{crit}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r},t)\,S_{g'}(\vec{r},t)T(t) + \sum_{i}^{I}\chi_{i,g}^{d}\,\lambda_i C_i(\vec{r},t), \qquad g=1,\dots,G$$

$$\frac{\partial}{\partial t}C_i(\vec{r},t) = -\lambda_i C_i(\vec{r},t) + \frac{\beta_i(\vec{r},t)]}{k_{crit}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r},t)\,S_{g'}(\vec{r},t)T(t)\,, \qquad i=1,\dots,I$$

$$\frac{1}{v_g(\vec{r},t)}\left[\frac{\partial}{\partial t}S_g(\vec{r},t) + \frac{S_g(\vec{r},t)}{T(t)}\frac{\partial}{\partial t}T(t)\right] = \nabla\cdot D_g(\vec{r},t)\nabla S_g(\vec{r},t) - \Sigma_{r,g}(\vec{r},t)\,S_g(\vec{r},t) + \sum_{g'\neq g}^{G}\Sigma_{s,g'\to g}(\vec{r},t)\,S_{g'}(\vec{r},t)$$
$$+[1-\beta(\vec{r},t)]\frac{\chi_g^p}{k_{crit}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r},t)\,S_{g'}(\vec{r},t) + \frac{1}{T(t)}\sum_{i}^{I}\chi_{i,g}^{d}\,\lambda_i C_i(\vec{r},t), \qquad g=1,\dots,G$$

$$S_g(\vec{r},t_{n+1}) + \nabla\cdot D_g(\vec{r},t)\nabla S_g(\vec{r},t_{n+1}) - \left\{\Sigma_{r,g}(\vec{r},t) + \frac{1}{v_g(\vec{r},t)}\left[\frac{1}{\Delta t} + \frac{1}{T(t)}\frac{\partial}{\partial t}T(t)\right]\right\}S_g(\vec{r},t_{n+1}) + \sum_{g'\neq g}^{G}\Sigma_{s,g'\to g}(\vec{r},t)\,S_{g'}(\vec{r},t_{n+1})$$
$$+[1-\beta(\vec{r},t)]\frac{\chi_g^p}{k_{crit}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r},t)\,S_{g'}(\vec{r},t_{n+1}) + \frac{1}{T(t)}\sum_{i}^{I}\chi_{i,g}^{d}\,\lambda_i C_i(\vec{r},t_{n+1}) - \frac{1}{v_g(\vec{r},t)\Delta t}S_g(\vec{r},t_n), \qquad g=1,\dots,G$$

Improved Quasi-Static Methods for Shape Functions

$$S_g(\vec{r},t_{n+1}) + \nabla\cdot D_g(\vec{r},t)\nabla S_g(\vec{r},t_{n+1}) - \left\{\Sigma_{r,g}(\vec{r},t) + \frac{1}{v_g(\vec{r},t)}\left[\frac{1}{\Delta t} + \frac{1}{T(t)}\frac{\partial}{\partial t}T(t)\right]\right\} S_g(\vec{r},t_{n+1}) + \sum_{g'\neq g}^{G}\Sigma_{s,g'\to g}(\vec{r},t)\, S_{g'}(\vec{r},t_{n+1})$$
$$+[1-\beta(\vec{r},t)]\frac{\chi_g^p}{k_{crit}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r},t)\, S_{g'}(\vec{r},t_{n+1}) + \frac{1}{T(t)}\sum_i^I \chi_{i,g}^d\, \lambda_i C_i(\vec{r},t) - \frac{1}{v_g(\vec{r},t)\Delta t} S_g(\vec{r},t_n), \quad g=1,\dots,G$$

- Integrate the shape functions with very coarse time steps
- Use PKEs and shape functions to really produce space-dependent precursor concentrations
- Spatial details available from (shape function interpolation) x (amplitude function)

- Original QS method ignored the time derivative of the shape function

$$\nabla\cdot D_g(\vec{r},t)\nabla S_g(\vec{r},t_{n+1}) - \left\{\Sigma_{r,g}(\vec{r},t) + \frac{1}{v_g(\vec{r},t)}\left[\frac{1}{\Delta t} + \frac{1}{T(t)}\frac{\partial}{\partial t}T(t)\right]\right\} S_g(\vec{r},t_{n+1}) + \sum_{g'\neq g}^{G}\Sigma_{s,g'\to g}(\vec{r},t)\, S_{g'}(\vec{r},t_{n+1})$$
$$+[1-\beta(\vec{r},t)]\frac{\chi_g^p}{k_{crit}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r},t)\, S_{g'}(\vec{r},t_{n+1}) + \frac{1}{T(t)}\sum_i^I \chi_{i,g}^d\, \lambda_i C_i(\vec{r},t), \quad g=1,\dots,G$$

Coarse Time Step Fully-implicit Integration for Shape Functions

$$\frac{\partial}{\partial t}\left\langle \frac{1}{v_g(\vec{r},t)}\phi_g(\vec{r},t)\right\rangle = \nabla\cdot D_g(\vec{r},t)\nabla\phi_g(\vec{r},t) - \Sigma_{r,g}(\vec{r},t)\,\phi_g(\vec{r},t) + \sum_{g'\neq g}^{G}\Sigma_{s,g'\to g}(\vec{r},t)\,\phi_{g'}(\vec{r},t)$$

$$+[1-\beta(\vec{r},t)]\frac{\chi_g^p}{k_{crit}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r},t)\,\phi_{g'}(\vec{r},t) + \sum_{i}^{I}\chi_{i,g}^d\,\lambda_i C_i(\vec{r},t), \qquad g=1,\dots,G$$

$$\frac{\partial}{\partial t}C_i(\vec{r},t) = -\lambda_i C_i(\vec{r},t) + \frac{\beta_i(\vec{r},t)]}{k_{crit}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r},t)\,\phi_{g'}(\vec{r},t), \qquad i=1,\dots,I$$

Assume

$$\frac{\partial}{\partial t}\phi_g(\vec{r},t) = \frac{\phi_g(\vec{r},t_{n+1})-\phi_g(\vec{r},t_n)}{\Delta t} \qquad \frac{\partial}{\partial t}C_i(\vec{r},t) = \frac{C_i(\vec{r},t_{n+1})-C_i(\vec{r},t_n)}{\Delta t}$$

$$-\nabla\cdot D_g(\vec{r},t)\nabla\phi_g(\vec{r},t_{n+1}) + \left[\Sigma_{r,g}(\vec{r},t)+\frac{1}{v_g\Delta_t}\right]\phi_g(\vec{r},t_{n+1}) - \sum_{g'\neq g}^{G}\Sigma_{s,g'\to g}(\vec{r},t)\,\phi_{g'}(\vec{r},t_{n+1}) +$$

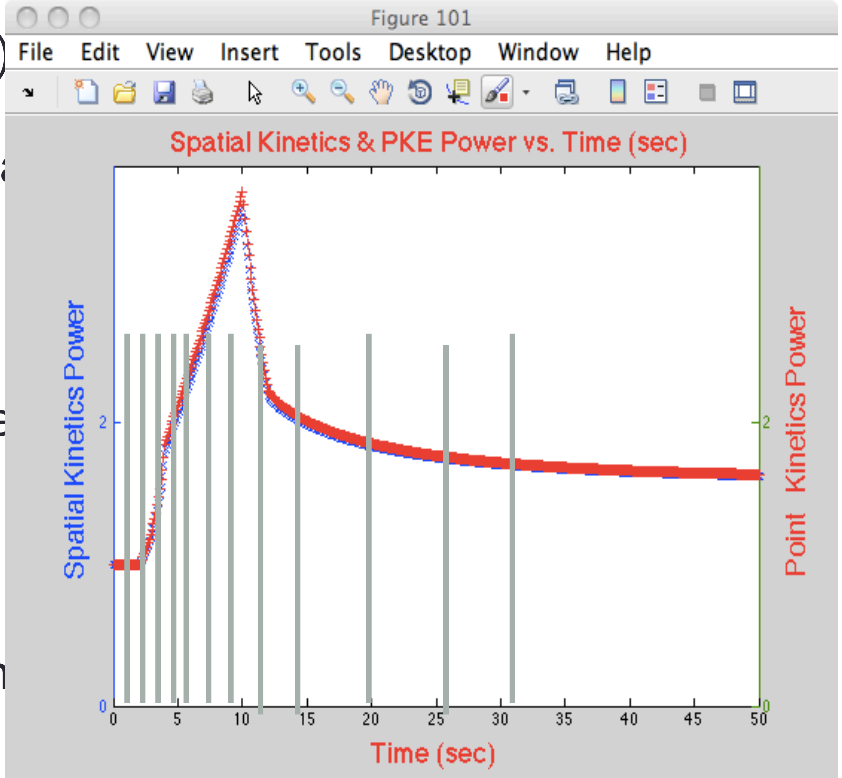
$$\left[[1-\beta(\vec{r},t)]\frac{\chi_g^p}{k_{crit}} + \sum_{i}^{I}\frac{\chi_{i,g}^d}{k_{crit}}\frac{\beta_i(\vec{r},t)\lambda_i\Delta_t}{1+\lambda_i\Delta_t}\right]\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r},t)\,\phi_{g'}(\vec{r},t_{n+1}) = \frac{\phi_1(\vec{r},t_n)}{v_1(\vec{r})\Delta_t} + \sum_{i}^{I}\lambda_i\frac{C_i(\vec{r},t_n)}{(1+\lambda_i\Delta_t)} \qquad g=1,\dots,G$$

$$C_i(\vec{r},t_{n+1}) = \frac{\beta_i(\vec{r},t)\Delta_t}{(1+\lambda_i\Delta_t)k_{crit}}\left[\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\vec{r},t)\,\phi_{g'}(\vec{r},t_{n+1})\right] + \frac{C_i(\vec{r},t_n)}{(1+\lambda_i\Delta_t)}, \qquad i=1,\dots,I$$

- Think of this as a crude time integration just to get on-the-fly shape functions
- PKEs are still used for the actual integration of the amplitude function
- Spatial details are available from (shape function interpolation) x (amplitude function)

Time-dependent Shape Functions

- Fixed k-eigenvalue shapes (BOT, EOT, etc.)
- Geometrical interpolation between k-eigenva
- Adiabatic shape functions during transient
- α-eigenmode shape functions during transie
- ω-mode shape functions during transient
- Quasi-static shape functions during transien



- Improved Quasi-static shape functions during transients
- Coarse time integration of shape functions during transient
- Frequency-transform time integration of shape functions during transient

Assignment for Next Class

- Solve Pset 3: transient1-D, 2-group finite-difference problems
- Think about how you will compute generalized PKE parameters from your 1-D, 2-group solutions.
- Study the kinetics review paper by Sutton and Aviles (KAPL)
- Start thinking about time-dependent solutions to 2-D, 2-group finite-difference problems.