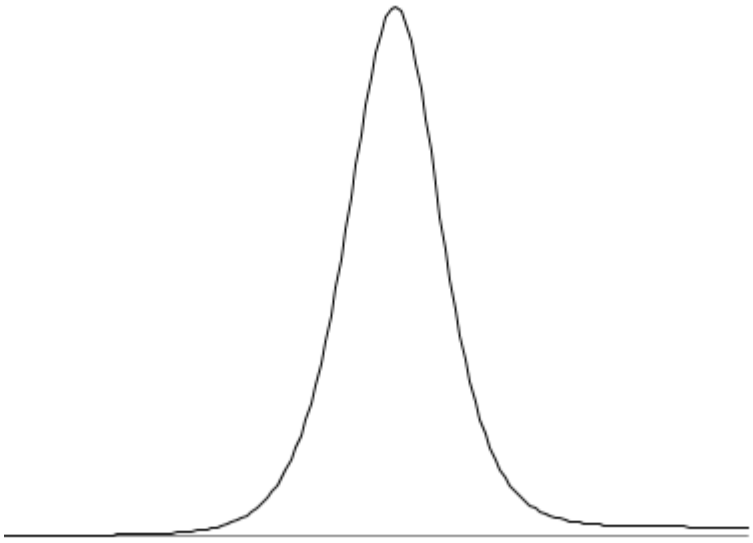
TECHNICAL REPORT:

$PKINETICS$ - Nuclear Fission Reactor Point Kinetics in JIC^{Lib2}

March, 2016

Jan Vermaak & Guillermo Villanueva

Rev 1.01



Contents

List of Figures

List of Tables

1 Derivation of the Point Kinetics Equations

A very simple approximation of the kinetics of a reactor can be made by integrating the neutron transport equation over the entire core volume. This results in the reduced transport equation as shown below:

$$\begin{aligned} \frac{1}{v}\frac{d\phi}{dt} + \nabla\vec{J} + \Sigma_a\phi(t) &= (1-\beta_{\text{eff}})\bar{\nu}\Sigma_f\phi(t) + \sum_{i=1}^{6}\lambda_i C_i(t) \\ \frac{dC_i}{dt} &= \beta_{i,\text{eff}}\bar{\nu}\Sigma_f\phi(t) - \lambda_i C_i(t) \quad i = 1,\ldots,6 \end{aligned} \tag{1}$$

Where:

ν	, Neutron velocity [m.s^{-1}]
ϕ	, Average neutron flux [cm^{-2}.s^{-1}]
$\vec{J}$	, Neutron current [cm^{-2}.s^{-1}]
Σ_a	, Effective macroscopic absorption cross-section [cm^{-1}]
β_{eff}	, Effective delayed neutron fraction
$\bar{\nu}$	, Average amount of neutrons released per fission
Σ_f	, Effective macroscopic fission cross-section [cm^{-1}]
λ_i	, Decay constant for delayed neutron precursor i [s^{-1}]
C_i	, Average concentration of delayed neutron precursor i [cm^{-3}]
$\beta_{i,\text{eff}}$	, Effective delayed neutron fraction of precursor i

In order to simplify equation 1 we need to define three additional parameters, the multiplication factor, k_{eff}, the prompt neutron lifetime, l_p, and reactivity, ρ.

1.1 The effective multiplication factor, k_{eff}

To get some context on where we will use this parameter we first lump all the components of criticality together as follows:

$$\begin{aligned} \frac{1}{v}\frac{d\phi}{dt} &= (1-\beta_{\text{eff}})\bar{\nu}\Sigma_f\phi(t) - \nabla\vec{J} - \Sigma_a\phi(t) + \sum_{i=1}^{6}\lambda_i C_i(t) \\ &= \bar{\nu}\Sigma_f\phi(t)\left((1-\beta_{\text{eff}}) - \left(\frac{\nabla\vec{J} + \Sigma_a\phi(t)}{\bar{\nu}\Sigma_f\phi(t)}\right)\right) + \sum_{i=1}^{6}\lambda_i C_i(t) \end{aligned} \tag{2}$$

We then define the effective multiplication factor as the ratio of sources to losses (i.e. $\frac{\text{born}}{\text{lost}}$):

$$k_{\text{eff}} = \frac{(1-\beta_{\text{eff}})\bar{\nu}\Sigma_f\phi(t) + \sum_{i=1}^{6}\lambda_i C_i(t)}{\nabla\vec{J} + \Sigma_a\phi(t)} \tag{3}$$

The next parameter, the prompt neutron lifetime, l_p, is a parameter that is both defined and measured at steady state. Therefore, at steady state, all the $\frac{dC_i}{dt}$ terms in equation 1 reduce to zero and we can deduce:

$$\lambda_i C_i(t) = \beta_{i,\text{eff}} \bar{\nu} \Sigma_f \phi(t)$$

Therefore:

$$\begin{aligned}
\sum_{i=1}^{6} \lambda_i C_i(t) &= \sum_{i=1}^{6} \beta_{i,\text{eff}} \bar{\nu} \Sigma_f \phi(t) \\
&= \bar{\nu} \Sigma_f \phi(t) \left(\sum_{i=1}^{6} \beta_{i,\text{eff}} \right) \\
\sum_{i=1}^{6} \lambda_i C_i(t) &= \beta_{\text{eff}} \bar{\nu} \Sigma_f \phi(t)
\end{aligned}$$

Equation 3 then becomes:

$$\begin{aligned}
k_{\text{eff}} &= \frac{(1-\beta_{\text{eff}}) \bar{\nu} \Sigma_f \phi(t) + \beta_{\text{eff}} \bar{\nu} \Sigma_f \phi(t)}{\nabla \vec{J} + \Sigma_a \phi(t)} \\
\therefore k_{\text{eff}} &= \frac{\bar{\nu} \Sigma_f \phi(t)}{\nabla \vec{J} + \Sigma_a \phi(t)}
\end{aligned}$$

Substituting this value of k_{eff} into equation 2, one obtains:

$$\frac{1}{v} \frac{d\phi}{dt} = \bar{\nu} \Sigma_f \phi(t) \left((1-\beta_{\text{eff}}) - \frac{1}{k_{\text{eff}}} \right) + \sum_{i=1}^{6} \lambda_i C_i(t) \tag{4}$$

Next we will remove the $\bar{\nu} \Sigma_f$ term by defining the prompt neutron lifetime, l_p.

1.2 The prompt neutron lifetime, l_p

The average time from the birth of a prompt neutron to the time it is absorbed or lost is termed the prompt neutron lifetime. In order to derive a more intuitive definition, let us first define the rate of neutron loss (either through absorption or leaking):

$$\text{rate of loss} = \text{leakage rate} + \text{absorption rate}$$

This, however, is familiar:

$$\text{rate of loss} = \nabla \vec{J} + \Sigma_a \phi(t)$$

It features in the equation for k_{eff} above and therefore we can write:

$$\text{rate of loss} = \frac{\bar{\nu} \Sigma_f \phi(t)}{\nabla \vec{J} + \Sigma_a \phi(t)} \cdot \left(\frac{1}{\bar{\nu} \Sigma_f \phi(t)} \right) = k_{\text{eff}} \cdot \left(\frac{1}{\bar{\nu} \Sigma_f \phi(t)} \right)$$

The prompt neutron lifetime is then defined as the time it would take to lose a certain neutron population $n(t)$. And this is defined as the total amount lost ($n(t)$) multiplied by the rate of loss:

$$\begin{aligned} l_p &= n(t) \times \text{rate of loss} \\ &= n(t) \times k_{\text{eff}} \cdot \left(\frac{1}{\bar{\nu}\Sigma_f \phi(t)} \right) \end{aligned}$$

Including the definition of flux, $\phi(t) = n(t).v$, we obtain the final form of the prompt neutron lifetime:

$$\begin{aligned} l_p &= n(t) \times k_{\text{eff}} \cdot \left(\frac{1}{\bar{\nu}\Sigma_f n(t) v} \right) \\ &= k_{\text{eff}} \cdot \left(\frac{1}{v.\bar{\nu}\Sigma_f} \right) \\ \therefore l_p &= \frac{k_{\text{eff}}}{v.\bar{\nu}\Sigma_f} \end{aligned}$$

Now extraction from this equation an expression for $\bar{\nu}\Sigma_f$ we get:

$$\bar{\nu}\Sigma_f = \frac{k_{\text{eff}}}{v.l_p}$$

We can then substitute this into equation 4 to obtain:

$$\begin{aligned} \frac{1}{v}\frac{d\phi}{dt} &= \frac{k_{\text{eff}}}{v.l_p}\phi(t)\left((1 - \beta_{\text{eff}}) - \frac{1}{k_{\text{eff}}} \right) + \sum_{i=1}^{6} \lambda_i C_i(t) \\ &= \frac{\phi(t)}{v.l_p}\left(k_{\text{eff}}(1 - \beta_{\text{eff}}) - 1 \right) + \sum_{i=1}^{6} \lambda_i C_i(t) \end{aligned} \tag{5}$$

1.3 Reactivity ρ

In practical reactor facilities the value of k_{eff} is never determined in the classical sense because it involves numbers that are hard to comprehend (i.e. 1.001 is controllable whereas 1.01 is a massive excursion). In order to center the concept of criticality around the value of zero (i.e. positive means power increases and negative means power decreases). For this, the term "reactivity", ρ, was defined as:

$$\rho = \frac{k_{\text{eff}} - 1}{k_{\text{eff}}}$$

However, the nature of this number was still unsatisfactory since it depended on the kinetic parameter β_{eff}. When normalizing by this factor, reactivity, ρ', with units of dollars (\$) becomes analogous between reactor shapes and sizes and is defined by:

$$\rho' = \frac{1}{\beta_{\text{eff}}} \cdot \frac{k_{\text{eff}} - 1}{k_{\text{eff}}} \tag{6}$$

Solving for k_{eff} from this equation we get:

$$\begin{aligned}
\rho' &= \frac{1}{\beta_{\text{eff}}} \cdot (1 - 1/k_{\text{eff}}) \\
\rho'.\beta_{\text{eff}} &= 1 - 1/k_{\text{eff}} \\
1/k_{\text{eff}} &= 1 - \rho'.\beta_{\text{eff}} \\
k_{\text{eff}} &= \frac{1}{1 - \rho'.\beta_{\text{eff}}}
\end{aligned}$$

Substituting this into equation 5 we get:

$$\begin{aligned}
\frac{1}{v}\frac{d\phi}{dt} &= \frac{\phi(t)}{v.l_p}\left(\frac{1}{1-\rho'.\beta_{\text{eff}}}(1-\beta_{\text{eff}}) - 1\right) + \sum_{i=1}^{6}\lambda_i C_i(t) \\
&= \frac{\beta_{\text{eff}}(\rho'-1)}{(1-\rho'\beta_{\text{eff}})} \cdot \frac{\phi(t)}{v.l_p} + \sum_{i=1}^{6}\lambda_i C_i(t)
\end{aligned}$$

1.4 Final simplified form

As a final modification to equation 1 we need to transform the differential equations for the delayed neutron precursors:

$$\frac{dC_i}{dt} = \beta_{i,\text{eff}}\bar{\nu}\Sigma_f\phi(t) - \lambda_i C_i(t) \quad i = 1, \ldots, 6$$

We do this by first substituting the derived expression for $\bar{\nu}\Sigma_f$:

$$\frac{dC_i}{dt} = \beta_{i,\text{eff}}\frac{k_{\text{eff}}}{v.l_p}\phi(t) - \lambda_i C_i(t) \quad i = 1, \ldots, 6$$

And then the expression for k_{eff}:

$$\frac{dC_i}{dt} = \beta_{i,\text{eff}}\frac{\frac{1}{1-\rho'.\beta_{\text{eff}}}}{v.l_p}\phi(t) - \lambda_i C_i(t) \quad i = 1, \ldots, 6$$

To finally get:

$$\frac{dC_i}{dt} = \frac{\beta_{i,\text{eff}}}{(1-\rho'.\beta_{\text{eff}})} \cdot \frac{\phi(t)}{v.l_p} - \lambda_i C_i(t) \quad i = 1, \ldots, 6$$

Therefore the final simplified form of the point kinetics equations, after substituting the expression for flux ($\phi(t) = n(t).v$) become:

$$\begin{aligned}
\frac{dn}{dt} &= \frac{\beta_{\text{eff}}(\rho'-1)}{(1-\rho'\beta_{\text{eff}})} \cdot \frac{n(t)}{l_p} + \sum_{i=1}^{6}\lambda_i C_i(t) \\
\frac{dC_i}{dt} &= \frac{\beta_{i,\text{eff}}}{(1-\rho'.\beta_{\text{eff}})} \cdot \frac{n(t)}{l_p} - \lambda_i C_i(t) \quad i = 1, \ldots, 6
\end{aligned} \tag{7}$$

2 Discretized forms of the Point Kinetics Equations

For any method of discretization, the time derivatives need to be linearized:

$$\frac{dn}{dt} = \frac{n^{j+1} - n^j}{\Delta t}$$
$$\frac{dC_i}{dt} = \frac{C_i^{j+1} - C_i^j}{\Delta t}$$

Where j and $j+1$ refers to time steps in the solution domain.

2.1 Initial first-order discretization by uncoupling

As an initial approach, let us consider calculating the future time step by strictly using values from the previous time step. This is known as Euler's Method for numerically solving ordinary differential equations:

$$\frac{n^{j+1} - n^j}{\Delta t} = \frac{\beta_{\text{eff}}(\rho' - 1)}{(1 - \rho'\beta_{\text{eff}})} \cdot \frac{n^j}{l_p} + \sum_{i=1}^{6} \lambda_i C_i^j$$
$$\frac{C_i^{j+1} - C_i^j}{\Delta t} = \frac{\beta_{i,\text{eff}}}{(1 - \rho'.\beta_{\text{eff}})} \cdot \frac{n^j}{l_p} - \lambda_i C_i^j \quad i = 1, \ldots, 6$$

In this form, each of the 7 equations can be calculated individually (uncoupled form) and the implementation in computer code becomes fairly easy:

$$\begin{aligned} n^{j+1} &= n^j + \Delta t\left(\frac{\beta_{\text{eff}}(\rho' - 1)}{(1 - \rho'\beta_{\text{eff}})} \cdot \frac{n^j}{l_p} + \sum_{i=1}^{6} \lambda_i C_i^j\right) \\ C_i^{j+1} &= C_i^j + \Delta t\left(\frac{\beta_{i,\text{eff}}}{(1 - \rho'.\beta_{\text{eff}})} \cdot \frac{n^j}{l_p} - \lambda_i C_i^j\right) \quad i = 1, \ldots, 6 \end{aligned} \tag{8}$$

The detriment of this method however is that it is fairly unstable and induces large errors when the time step is not extremely small.

2.2 Problems with the coupled version

The set of equations 7, is a system of coupled linear ordinary differential equations. By applying the first order discretization, as we did above, where we solve each equation by using the previous time step values we essentially make them uncoupled and hence easy to solve. However, this approach introduces instability and an error between time steps that can only be minimized by using very small time steps. In order to

improve the stability and the cumulative error we must use the system of equations in their coupled form:

$$\frac{n^{j+1}-n^j}{\Delta t} = \frac{\beta_{\text{eff}}(\rho'-1)}{(1-\rho'\beta_{\text{eff}})}\cdot\frac{n^{j+1}}{l_p} + \sum_{i=1}^{6}\lambda_i C_i^{j+1}$$
$$\frac{C_i^{j+1}-C_i^j}{\Delta t} = \frac{\beta_{i,\text{eff}}}{(1-\rho'.\beta_{\text{eff}})}\cdot\frac{n^{j+1}}{l_p} - \lambda_i C_i^{j+1} \quad i=1,\ldots,6$$

In order to make this equation a little easier to read let us group some values that are constant over the time step:

$$A_0 = \Delta t.\frac{\beta_{\text{eff}}(\rho'-1)}{(1-\rho'\beta_{\text{eff}})}\cdot\frac{1}{l_p}$$
$$A_i = \Delta t.\frac{\beta_{i,\text{eff}}}{(1-\rho'.\beta_{\text{eff}})}\cdot\frac{1}{l_p} \quad i=1,\ldots,6$$

And:

$$\Lambda_i = \Delta t.\lambda_i$$
$$\Lambda_i^* = (1+\Lambda_i) = 1+\Delta t.\lambda_i$$

And therefore we have a slightly more readable version:

$$n^{j+1}-n^j = A_0.n^{j+1} + \sum_{i=1}^{6}\Lambda_i C_i^{j+1}$$
$$C_i^{j+1}-C_i^j = A_i.n^{j+1} - \Lambda_i C_i^{j+1} \quad i=1,\ldots,6$$

Rearranging:

$$n^{j+1}(1-A_0) - \Lambda_1.C_1^{j+1} - \cdots - \Lambda_6.C_6^{j+1} = n^j$$
$$-A_i.n^{j+1} + \Lambda_i^*.C_i^{j+1} = C_i^j \quad i=1,\ldots,6$$

We can write this in matrix form as:

$$\begin{bmatrix} (1-A_0) & -\Lambda_1 & -\Lambda_2 & -\Lambda_3 & -\Lambda_4 & -\Lambda_5 & -\Lambda_6 \\ -A_1 & \Lambda_1^* & 0 & 0 & 0 & 0 & 0 \\ -A_2 & 0 & \Lambda_2^* & 0 & 0 & 0 & 0 \\ -A_3 & 0 & 0 & \Lambda_3^* & 0 & 0 & 0 \\ -A_4 & 0 & 0 & 0 & \Lambda_4^* & 0 & 0 \\ -A_5 & 0 & 0 & 0 & 0 & \Lambda_5^* & 0 \\ -A_6 & 0 & 0 & 0 & 0 & 0 & \Lambda_6^* \end{bmatrix} \begin{bmatrix} n^{j+1} \\ C_1^{j+1} \\ C_2^{j+1} \\ C_3^{j+1} \\ C_4^{j+1} \\ C_5^{j+1} \\ C_6^{j+1} \end{bmatrix} = \begin{bmatrix} n^j \\ C_1^j \\ C_2^j \\ C_3^j \\ C_4^j \\ C_5^j \\ C_6^j \end{bmatrix}$$

In simple matrix form this is:

$$Ax = b$$

This can then easily be solved using a suitable numerical solver.

3 Thermal feedback

Thermal feedback has been modeled by including the heat capacity equation:

$$\int_{t_i}^{t_f} \dot{Q}.dt = m.C_p.\Delta T$$

Where the total fuel mass, m, and its associated heat capacity, C_p, is user supplied. The energy integral $\int_{t_i}^{t_f} \dot{Q}.dt$ is calculated after each time step from:

$$\left[\int_{t_i}^{t_f} \dot{Q}.dt\right]^{timestep\ i+1} = \left[\int_{t_i}^{t_f} \dot{Q}.dt\right]^{timestep\ i} + \dot{Q}^{i+1}.\Delta t$$

The reactor power, $\dot{Q}$, is assumed to be associated with the neutron population through a constant, C_{power}, and can be altered from its default unity input by the user. In other words, the following equation is applied:

$$\dot{Q}(t) = C_{power} \cdot n(t)$$

Where $C_{power} = 1.0$ by default. The temperature of the fuel is then calculated from:

$$T^{i+1} = T^i + \frac{\left[\int_{t_i}^{t_f} \dot{Q}.dt\right]^{timestep\ i+1}}{m.C_p}$$

The default fuel mass in the core corresponds to 90 TRIGA fuel elements ($90 \times 2.461\ kg.element^{-1} = 221.49\ kg$). For the heat capacity, the heat capacity as a function of temperature have been obtained by weighting the individual heat capacities of pure uranium and zirconium hydride. This approach is justified in NUREG-1282 [1]. The heat capacity of Uranium has been obtained from [2] and that for Zirconium-hydride has been obtained from [3] and are combined in Table 1 below.

Table 1: Heat capacity of Texas A&M TRIGA reactor fuel.

Temperature	**Heat Capacity**	**Temperature**	**Heat Capacity**
[K]	**[J/kg/K]**	**[K]**	**[J/kg/K]**
323.15	216.8	773.15	452.3
373.15	268.8	873.15	1577.0
473.15	321.1	973.15	535.6
573.15	346.3	1073.15	430.1
673.15	399.4	1173.15	414.5

The temperature feedback coefficients have been determined from the end-of-life (EOL) data of the Texas A&M TRIGA Reactor's Safety Analysis Report and the effective temperature feedback is given in Table 2.

Table 2: Reactivity feedback from fuel temperature.

Temperature [K]	**Reactivity, ρ [$]**	**Temperature [K]**	**Reactivity, ρ [$]**
293.15	0.000	823.15	-5.917
323.15	-0.143	873.15	-6.692
373.15	-0.442	923.15	-7.505
423.15	-0.819	973.15	-8.354
473.15	-1.273	1023.15	-9.208
523.15	-1.865	1073.15	-10.095
573.15	-2.505	1123.15	-11.015
623.15	-3.133	1173.15	-11.968
673.15	-3.816	1223.15	-12.954
723.15	-4.479	1273.15	-13.972
773.15	-5.179		

4 Input Guide

The ***pkinetics*** code by default runs a critical system based on the kinetic parameters of the Texas A&M Nuclear Science Center TRIGA reactor at a neutron population of 0.03 $neutrons.cm^{-3}$. In order to run the code with different input conditions, an input file can be specified that will alter the conditions of the simulation. As an initial example, consider the input file shown below:

4.1 Running the code

The point kinetics code is run by executing ***JIC_LIB2.exe*** with the option ***pkinetics***, followed by the input file name as shown below:

```
JIC_LIB2.exe pkinetics input.txt
```

4.2 Input Cards

The default values for all the input cards are for the Texas A&M Nuclear Science Center TRIGA reactor.

beff int float
Specifies the delayed neutron fraction for the delayed neutron precursor group.

- Word 1: int Delayed neutron precursor group number.
- Word 2: float Delayed neutron fraction for precursor group.

Default values:

```
Group 1  2.6457461646E-04
Group 2  1.4937238494E-03
Group 3  1.3179916318E-03
Group 4  2.8312412831E-03
Group 5  9.0404463040E-04
Group 6  1.8842398884E-04
```

lambda int float
Specifies the decay constant for the delayed neutron precursor group.

- Word 1: int Delayed neutron precursor group number.
- Word 2: float Decay constant for precursor group [s^{-1}].

Default values:

```
Group 1  0.01273
Group 2  0.03171
Group 3  0.11670
Group 4  0.31215
Group 5  1.39880
Group 6  3.85100
```

lifetime float
Specifies the prompt neutron lifetime for the simulation.

- Word 1: float Prompt neutron lifetime [s].

Default values:

```
0.026178e-3
```

precursor int float
Specifies the delayed neutron precursor concentration for the indicated group. If omitted, or values are 0.0, the zero reactivity steady state values associated with the given population will be used as initial values.

- Word 1: int Delayed neutron precursor group number.
- Word 2: float Concentration for precursor group [cm^{-3}].

Default values:

```
Calculate from steady state
```

population float
Specifies the initial neutron population for the simulation.

- Word 1: float Neutron population [cm^{-3}].

Default values:

```
30.0e-3
```

rhopoint float float
Specifies the reactivity versus time dependence. Each time this card appears, the data point goes into an accumulator that will eventually build a time table.

- Word 1: float Time [s].
- Word 2: float Reactivity [$].

Default values:

```
0.0  0.0
```

timestep float
Specifies the outer timestep size.

- Word 1: float timestep [ms].

Default values:

```
0.0166666666
```

simtime float
Specifies the total simulation time.

- Word 1: float simulation time [s].

Default values:

```
10.0
```

solver int
Specifies the solver selection.

- Word 1: int Solver option. Can be 1:Coupled CG, 2:Uncoupled Euler.

Default values:

```
1
```

mass float
Specifies the total fuel mass.

- Word 1: float Fuel mass $[kg]$.

Default values:

```
221.49
```

cpower float
Specifies the power relation to neutron population.

- Word 1: float ratio.

Default values:

```
1.0
```

temperature float
Specifies the initial fuel temperature.

- Word 1: float Fuel temperature $[K]$.

Default values:

```
293.15 K
```

cppoint float float
Specifies the heat capacity dependence on temperature. If so much as a single point is entered, the entire default table is deleted and the new data is accepted.

- Word 1: float Temperature $[K]$.
- Word 2: float Heat Capacity $[J.kg^{-1}.K^{-1}]$.

Default values:

```
T       Cp      T       Cp
323.15 216.8 773.15 452.3
373.15 268.8 873.15 1577.0
473.15 321.1 973.15 535.6
573.15 346.3 1073.15 430.1
673.15 399.4 1173.15 414.5
```

alphapoint float float

Specifies the temperature feedback of the reactor. If so much as a single point is entered, the entire default table is deleted and the new data is accepted.

- Word 1: float Temperature $[K]$.
- Word 2: float Reactivity [$].

Default values:

```
T [K] rho          T [K] rho
293.15 0.000        823.15 -5.917
323.15 -0.212       873.15 -6.692
373.15 -0.566       923.15 -7.505
423.15 -0.919       973.15 -8.354
473.15 -1.273       1023.15 -9.208
523.15 -1.865       1073.15 -10.095
573.15 -2.505       1123.15 -11.015
623.15 -3.133       1173.15 -11.968
673.15 -3.816       1223.15 -12.954
723.15 -4.479       1273.15 -13.972
773.15 -5.179
```

5 Verification & Validation

5.1 Test 1 - no thermal feedback

The response of the solver has been measured against the Texas A&M TRIGA reactor at a power of 300 W. The results are shown in Figure 1. The code matches the reactor data well during withdrawal and insertion of the rod, however, the transition period where withdrawal is reversed the code cannot capture the reactor related transient. Therefore, the code matches well before and after the point where the rod withdrawal direction is changed.

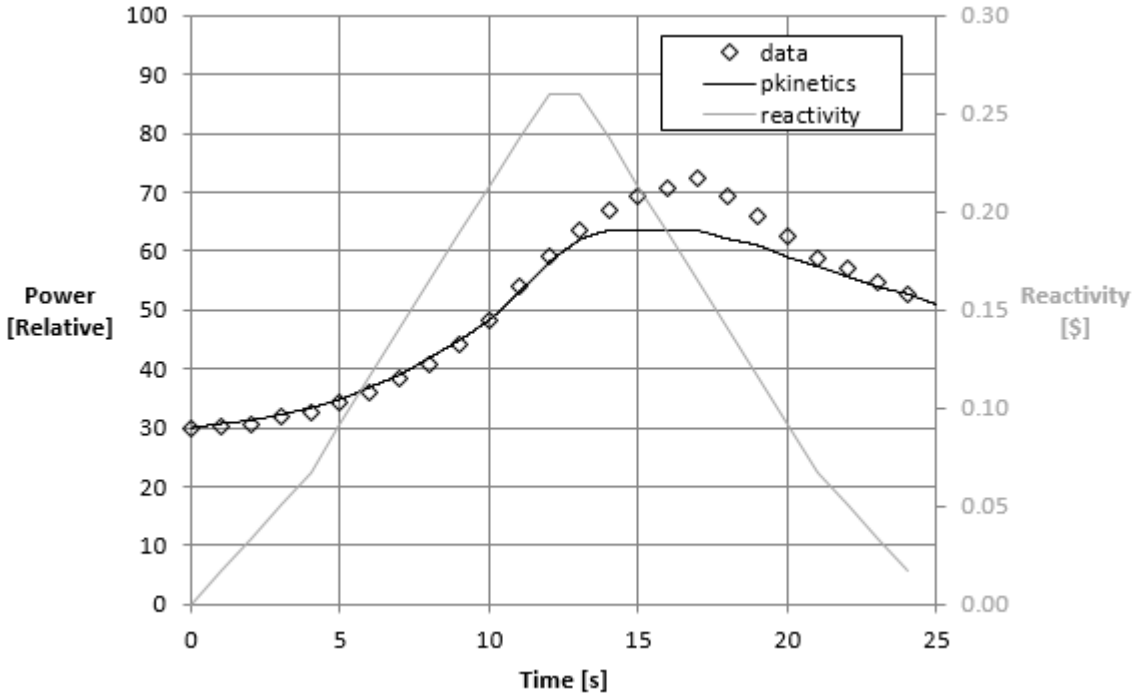


Figure 1: Results of a simple rod oscillation.

5.2 Test 2 - Pulsing

Assessing the performance of the code against measured data is a more strenuous task. The major input parameter(s) is the temperature feedback coefficient. For coefficients obtained from the Texas A&M TRIGA safety analysis report, for End-of-Life conditions, Table 3 below compares the measured versus calculated values. The values compare within the expected nominal magnitudes, however, given the uncertainty in feedback coefficients this comparison is harder to judge absolute accuracy.

Table 3: Comparison of peak powers.

Reactivity [$]	**Peak NSCR** [MW]	**Peak Code** [MW]	**Deviation** [%]
1.15	33.2	36.1	8.9
1.24	64.2	81.2	26.6
1.33	129.9	145.5	12.0
1.56	423.8	400.6	-5.5

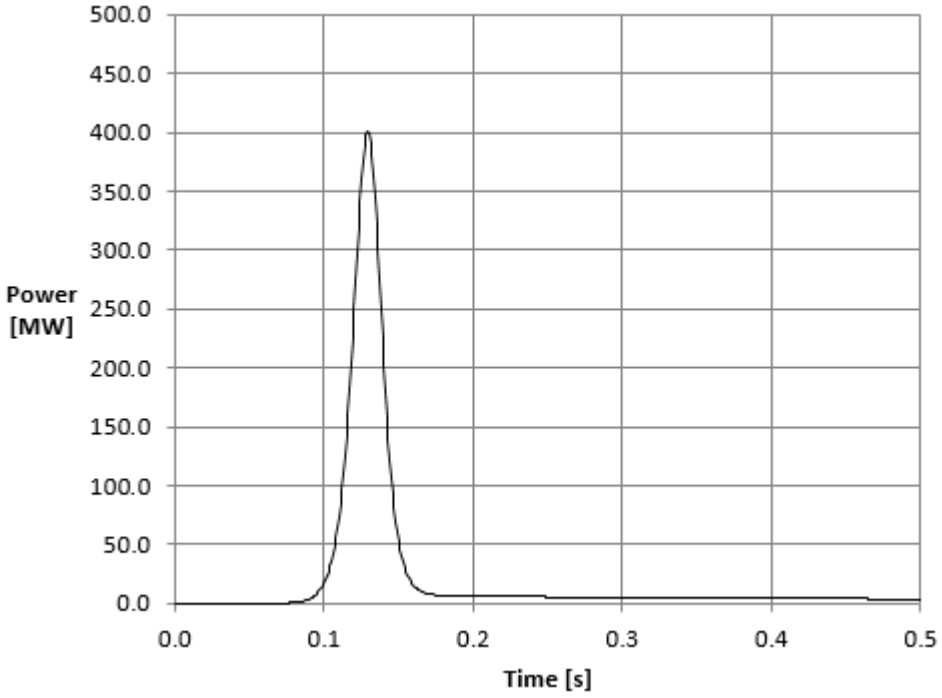


Figure 2: Results of a pulse with a reactivity of $1.56.

6 Example input file

```
// ========================== Kinetic parameters
beff 1  2.6457461646E-04
beff 2  1.4937238494E-03
beff 3  1.3179916318E-03
beff 4  2.8312412831E-03
beff 5  9.0404463040E-04
beff 6  1.8842398884E-04

lambda 1  0.01273
lambda 2  0.03171
lambda 3  0.11670
lambda 4  0.31215
lambda 5  1.39880
lambda 6  3.85100

lifetime 0.026178e-3

// ========================== Simulation initial conditions

precursor 1  0.000
precursor 2  0.000
precursor 3  0.000
precursor 4  0.000
precursor 5  0.000
precursor 6  0.000
population 30.0e-3

// ========================== Reactivity table
rhopoint  0.0    0.0
rhopoint  1.0    0.0
rhopoint  1.0001 0.21

// ========================== Simulation settings
timestep 0.0166666
simtime 10.0
solver 1
```

References

[1] *Safety Evaluation Report on High-Uranium Content, Low-Enriched Uranium-Zirconium Hydride Fuels for TRIGA Reactors*, NUREG-1282, Docket number 50-163, GA Technologies, August 1987.

[2] Ginnings D.C., Corruccini R.J., *Heat Capacities at High Temperatures of Uranium, Uranium Trichloride, and Uranium Tetrachloride*, Journal of Research of the National Bureau of Standards, Research Paper RP1831, Volume 39, October 1947.

[3] Douglas T.B., Victor A.C., *Heat Content of Zirconium and of Five Compositions of Zirconium Hydride from* 0° *to* $990^\circ C$, Journal of Research of the National Bureau of Standards, Research Paper RP2878, Volume 61, July 1958.