Point Kinetics Module - Numerical Methods

K. Łuszczek

July 3, 2019

Abstract

This short document outlines a set of simple numerical methods used in the point kinetics module for the solution of the neutron density equation. In addition a benchmark against selected published results is presented at the end of the document.

1 Point Kinetics - General Form

The point kinetics equation that will be solved is of the following form[1]:

$$\frac{dn(t)}{dt} = \frac{\rho(t) - \beta(t)}{\Lambda(t)} n(t) + \sum_{l=1}^{L} \lambda_l(t) c_l(t) + S(t) \tag{1.1}$$

$$\frac{dc_l(t)}{dt} = \frac{\beta_l(t)}{\Lambda(t)} n(t) - \lambda_l(t) c_l(t) \tag{1.2}$$

where:

- n – neutron density $\left[{}^{\text{neutrons}}/_{cm^3}\right]$
- ρ – reactivity
- β – total delayed neutron fraction $\left[{}^{\text{delayed neutrons}}/_{\text{total number of neutrons}}\right]$
- β_l – fraction of delayed neutrons emitted by the l-th precursor family $\left[{}^{\text{delayed neutrons}}/_{\text{total number of neutrons}}\right]$
- Λ – mean neutron generation time $[s]$
- λ_l – decay constant of the l-th precursor family $\left[{}^{1}/_{s}\right]$
- c_l – concentration of the l-th precursor family $\left[{}^{\text{precursors}}/_{cm^3}\right]$
- S – external neutron source $\left[{}^{\text{neutrons}}/_{(cm^3 s)}\right]$

[1] The derivation of the point kinetics equation can be easily found in literature. Refer to [Anglart, 2013] for an example.

Let us define a point kinetics time step, $i+1$. Time at the beginning and at the end of the time step will be denoted as t_i and t_{i+1}, respectively. The length of the time step will be defined as:

$$\Delta t = t_{i+1} - t_i \tag{1.3}$$

The following sections will discuss the solution of the point kinetics equations for any given kinetics time step $i+1$. Prior to proceeding to the solutions themselves, a couple of assumptions are made.

1. Within the time step $i+1$, the kinetic parameters $\beta_l, \beta, \Lambda, \lambda_l$ are time-independent. This assumption is reasonable for the length of time-steps that will be dealt with in practice.

2. The initial conditions for any given time-step $i+1$, $n(t_i), c_l(t_i)$, are known at the time of solving the point kinetics equation for that step. For the very first time step this means that $n(t_0 = 0), c_l(t_0 = 0)$ are known.

2 Theta Integration

Eq. 1.1 and Eq. 1.2 are identified as first-order linear differential equations. These equations can be quite easily solved numerically by means of the so-called theta integration. As this method is often referenced in this text, it seemed appropriate to briefly introduce it before the neutron density and precursor density equations are dealt with. Consider the following form of the time-dependent differential equation:

$$\frac{d}{dt}y(t) = f(t, y(t)) \tag{2.1}$$

Now, one can integrate over $\Delta t = t_{i+1} - t_i$

$$y(t_{i+1}) = y(t_i) + \int_{t_i}^{t_{i+1}} dt f(t, y(t)) \tag{2.2}$$

The integral can be approximated by:

$$\int_{t_i}^{t_{i+1}} dt' f(t', y(t')) \approx \Delta t \left[(1-\theta) f(t_i, y(t_i)) + \theta f(t_{i+1}, y(t_{i+1})) \right] \tag{2.3}$$

Varying theta in the range $(0, 1]$ results in different non-linear procedures (since the solution at the end of time step is required to determine $f(t_{i+1}, y(t_{i+1}))$. For purposes of this document two specific theta values are of interest. Namely:

- $\theta = 1$; this theta value yields the implicit Euler method, which is first-order accurate. This means that the error to the exact solution is proportional to the first power of the time step: $O(\Delta t)$. This method is unconditionally stable.
- $\theta = 0.5$; this theta value yields the Crank-Nicholson method, which is second order accurate. This means that the error to the exact solution is proportional to the second power of the time step: $O(\Delta t^2)$. This method is conditionally stable (stability discussed in section 5.1.1).

3 The Precursor Equation

In this section the solution to the time-dependent precursors concentration equation will be discussed. We are looking for the form of the equation that could be plugged into the neutron balance equation 1.1. As will be shown later, all solution methods presented in this section can be written in the following generic form:

$$c_l(t) = \widetilde{x}_l c_l(t_i) + \left(\frac{\beta_l}{\lambda_l \Lambda}\right) \widetilde{y}_l n(t_i) + \left(\frac{\beta_l}{\lambda_l \Lambda}\right) \widetilde{z}_l n(t); \text{ for } t \geq t_i \tag{3.1}$$

3.1 The Backward Euler Method

Applying the theta integration 2.3 to the precursor density equation 3.1 yields

$$c_l(t_{i+1}) = c_l(t_i) + \Delta t \left[(1-\theta)\left(\frac{\beta_l}{\Lambda} n(t_i) - \lambda_l c_l(t_i)\right) + \theta\left(\frac{\beta_l}{\Lambda} n(t_{i+1}) - \lambda_l c_l(t_{i+1})\right)\right] \tag{3.2}$$

By setting $\theta = 1$, the theta integration becomes the so called fully implicit backward Euler method. Rewriting Eq. 3.2 as the backward Euler method and grouping terms in order to match the the form of Eq. 3.1 results in the following:

$$c_l(t_{i+1}) = c_l(t_i)\left[\frac{1}{1+\Delta t \lambda_l}\right] + n(t_{i+1})\frac{\beta_l}{\Lambda}\left[\frac{\Delta t}{1+\Delta t \lambda_l}\right] \tag{3.3}$$

When solving the precursor equation with the implicit backward Euler method, the coefficients in Eq. 3.1 become:

$$\begin{aligned} \widetilde{x}_l &= \frac{1}{1+\Delta t \lambda_l} \\ \widetilde{y}_l &= 0 \\ \widetilde{z}_l &= \frac{\Delta t \lambda_l}{1+\Delta t \lambda_l} \end{aligned} \tag{3.4}$$

It is noted that the coefficients are independent of the solution itself.

3.2 An Analytic Solution

The precursor equation 1.2 can also be solved analytically. The equation is an inhomogeneous linear differential equation. There are many established methods of solving the equation of this type. Here, an arbitrary choice is made to apply the "variations of parameters" method. Firstly, the solution to the complementary homogeneous equation (also known as the associated homogeneous equation) is sought for. The associated homogenous equation is of the following form:

$$\frac{dc_l(t)}{dt} + \lambda_l c_l(t) = 0 \tag{3.5}$$

The solution can be found in the following manner:

- Grouping variables: $\frac{dc_l(t)}{c_l} = -\lambda_l dt$
- Integrating: $\ln(c_l(t)) = -\lambda_l t + B_1$
- Exponenting: $|c_l(t)| = e^{B_1} e^{-\lambda_l t}$

The absolute value can be dropped (since the precursor concentration can not be negative) and e^{B_1} is renamed as B. Thus, the solution to the complementary equation is:

$$c_l(t) = Be^{-\lambda_l t} \tag{3.6}$$

The constant B in Eq. 3.6 will be replaced by an arbitrary function $B(t)$ (the variation of parameter). Using this form of $c_l(t)$ in the inhomogeneous Eq. 1.2, one gets:

$$\frac{dB(t)}{dt}e^{-\lambda_l t} - \lambda_l B(t)e^{-\lambda_l t} = \frac{\beta_l}{\Lambda}n(t) - \lambda_l B(t)e^{-\lambda_l t} \tag{3.7}$$

$$\frac{dB(t)}{dt} = \frac{\beta_l}{\Lambda}n(t)e^{\lambda_l t} \tag{3.8}$$

Integration over the time step $\Delta t = t_{i+1} - t_i$ yields:

$$B(t_{t+1}) = B(t_i) + \frac{\beta_l}{\Lambda}\int_{t_i}^{t_{i+1}} n(t')e^{\lambda_l t'}dt' \tag{3.9}$$

$$c_l(t_{i+1}) = e^{-\lambda_l \Delta t}\left[c_l(t_i) + \frac{\beta_l}{\Lambda}\int_{t_i}^{t_{i+1}} n(t')e^{\lambda_l (t'-t_i)}dt'\right] \tag{3.10}$$

Eq. 3.10 can be solved analytically, if an analytic representation of the time dependence of the neutron density in the integrand, $n(t')$, is known. This representation needs to be approximated. Three choices are given in the following subsections.

3.2.1 Constant Neutron Density Approximation

The constant time independence approximation of the neutron density within a limited time period $t \in [t_i, t_{i+1}]$:

$$n(t) \approx n(t_{i+1});\ \ t \in [t_i, t_{i+1}] \tag{3.11}$$

leads Eq. 3.10 to take the following form:

$$c_l(t_{i+1}) \approx e^{-\lambda_l \Delta t} c_l(t_i) + \frac{1}{\lambda_l}(1 - e^{-\lambda_l \Delta t})\frac{\beta_l}{\Lambda} n(t_{i+1}) \tag{3.12}$$

The constant approximation makes the coefficients in Eq. 3.1 become:

$$\begin{aligned} \widetilde{x}_l &= e^{-\lambda_l \Delta t} \\ \widetilde{y}_l &= 0 \\ \widetilde{z}_l &= (1 - e^{-\lambda_l \Delta t}) \end{aligned} \tag{3.13}$$

Note that the coefficients are independent of the solution itself.

3.2.2 Linear Neutron Density Approximation

The linear time dependence approximation:

$$n(t) \approx n(t_i) + \left(\frac{n(t_{i+1}) - n(t_i)}{\Delta t_{i+1}}\right) \cdot (t - t_i)\,;\ \ t \in [t_i, t_{i+1}] \tag{3.14}$$

plugged into Eq. 3.10 yields

$$c_l(t_{i+1}) \approx e^{-\lambda_l \Delta t} c_l(t_i) + \left[\frac{1 - e^{-\lambda_l \Delta t}(1 + \lambda_l \Delta t)}{\lambda_l^2 \Delta t}\right] \frac{\beta_l}{\Lambda} n(t_i) + \left[\frac{\lambda_l \Delta t + e^{-\lambda_l \Delta t} - 1}{\lambda_l^2 \Delta t}\right] \frac{\beta_l}{\Lambda} n(t_{i+1}) \tag{3.15}$$

The linear approximation makes the coefficients in Eq. 3.1 become:

$$\begin{aligned} \widetilde{x}_l &= e^{-\lambda_l \Delta t} \\ \widetilde{y}_l &= \frac{1 - e^{-\lambda_l \Delta t}(1 + \lambda_l \Delta t)}{\lambda_l \Delta t} \\ \widetilde{z}_l &= \frac{\lambda_l \Delta t + e^{-\lambda_l \Delta t} - 1}{\lambda_l \Delta t} \end{aligned} \tag{3.16}$$

The coefficients $\widetilde{x}_l, \widetilde{y}_l, \widetilde{z}_l$ are independent of the solution itself.

3.2.3 Exponential Neutron Density Approximation

The exponential time dependence approximation:

$$\left.\begin{aligned} n(t) &\approx n(t_i)e^{\Omega_l[t-t_i]} \\ \Omega_l &= \frac{1}{\Delta t}\ln\left(\frac{n(t_{i+1})}{n(t_i)}\right) \quad ; \; t \in [t_i, t_{i+1}] \end{aligned}\right\} \tag{3.17}$$

yields:

$$c_l(t_{i+1}) \approx e^{-\lambda_l \Delta t} c_l(t_i) + e^{-\lambda_l \Delta t}\left[\frac{1}{\Omega_l + \lambda_l}\left(e^{(\Omega_l+\lambda_l)\Delta t} - 1\right)\right]\frac{\beta_l}{\Lambda} n(t_i) \tag{3.18}$$

The Eq. 3.18 will be re-written to be closer to the form of equation 3.1.

$$c_l(t_{i+1}) \approx e^{-\lambda_l \Delta t} c_l(t_i) + e^{-\lambda_l \Delta t}\frac{\beta_l}{\Lambda} n(t_i) + e^{-(\Omega_l+\lambda_l)\Delta t}\left[\frac{e^{(\Omega_l+\lambda_l)\Delta t} - 1}{\Omega_l + \lambda_l} - 1\right]\frac{\beta_l}{\Lambda} n(t_{i+1}) \tag{3.19}$$

The coefficients in Eq. 3.1 are now:

$$\begin{aligned} \widetilde{x}_l &= e^{-\lambda_l \Delta t} \\ \widetilde{y}_l &= \lambda_l e^{-\lambda_l \Delta t} \\ \widetilde{z}_l &= \lambda_l e^{-(\Omega_l+\lambda_l)\Delta t}\left[\frac{e^{(\Omega_l+\lambda_l)\Delta t} - 1}{\Omega_l + \lambda_l} - 1\right] \end{aligned} \tag{3.20}$$

Here, the final solution will need to be obtained through iterations, due to the fact that Ω_l depends on the end of step neutron density, $n(t_{i+1})$.

4 The Neutron Density Equation

In total, four methods of the temporal integration of Eq. 1.1 have been implemented in the point-kinetics module. They are listed below, with their corresponding order p:

- The Backward Euler Method, $p = 1$,
- The Frequency Transformed Backward Euler Method, $p = 1$,
- The Crank-Nicolson Method, $p = 2$,
- The Frequency Transformed Crank-Nicolson Method, $p = 2$.

4.1 The Backward Euler Method

As done for the precursor density equation in Section 3.1, the neutron density equation can be simply solved by means of theta-integration (Eq. 2.3). We force the method to be fully implicit by setting $\theta = 1$. Then, integrating Eq. 1.1 over the time step $\Delta t = t_{i+1} - t_i$, using backward Euler method yields :

$$n(t_{i+1}) - n(t_i) = \Delta t \left[X(t_{i+1}) n(t_{i+1}) + \sum_{l=1}^{L} \lambda_l c_l(t_{i+1}) + S(t_{i+1}) \right] \tag{4.1}$$

$$X(t_{i+1}) = \frac{\rho(t_{i+1}) - \beta}{\Lambda} \tag{4.2}$$

Regardless of the choice of the precursor solution method described in Section 3.1 we can plug the general form of the end-of-step precursor density (Eq. 3.1) into Eq. 4.1.

$$\begin{array}{c} n(t_{i+1}) \left[1 - \Delta t \left(X(t_{i+1}) - \sum_{l=1}^{L} \widetilde{z}_l \frac{\beta_l}{\Lambda} \right) \right] = \\ n(t_i) \left[1 + \Delta t \sum_{l=1}^{L} \widetilde{y}_l \frac{\beta_l}{\Lambda} \right] + \Delta t \left(\sum_{l=1}^{L} \widetilde{x}_l \lambda_l c_l(t_i) + S(t_{i+1}) \right) \end{array} \tag{4.3}$$

4.2 The Frequency Transformed Backward Euler Method

To solve Eq. 1.1 for the time step $\Delta t = t_{i+1} - t_i$ one could also apply a frequency transformation in the following form:

$$\left.\begin{array}{rl} n(t) = e^{\omega[t-t_i]} \Psi(t) & \\ \Psi(t_i) = n(t_i) & ; t \in [t_i, t_{i+1}] \end{array}\right\} \tag{4.4}$$

The left hand side of Eq. 1.1 can then be expressed as:

$$\frac{dn}{dt} = \omega n(t) + e^{\omega[t-t_i]} \frac{d}{dt} \Psi(t) \tag{4.5}$$

Eq. 4.5 is plugged into Eq. 1.1, after which implicit time integration (Backward Euler method) is applied (as in section 4.1). The expression for the end-of-step neutron density, $n(t_{i+1})$, can than be obtained:

$$\begin{array}{rcl} n(t_{i+1}) \left[1 - \Delta t \left(X(t_{i+1}) + \sum_{l=1}^{L} \left(\widetilde{z}_l \frac{\beta_l}{\Lambda} \right) - \omega \right) \right] & = & \\ n(t_i) \left[e^{\omega \Delta t} + \Delta t \sum_{l=1}^{L} \left(\widetilde{y}_l \frac{\beta_l}{\Lambda} \right) \right] & + & \\ \Delta t \left(\sum_{l=1}^{L} \left(\widetilde{x}_l \lambda_l c_l(t_i) \right) + S(t_{i+1}) \right) & & \end{array} \tag{4.6}$$

4.3 The Crank-Nicolson Method

The Crank-Nicolson method is obtained from the theta-integration by setting $\theta = \frac{1}{2}$. As opposed to the implicit Euler method, it is sensitive to numerical oscillations for very large time steps. Solving the neutron density equation with

the Crank-Nicolson method yields the following equation for the end-of-step neutron density:

$$\begin{aligned} n(t_{i+1})\left[1-\theta\Delta t\left(X(t_{i+1})+\sum_{l=1}^{L}\left(\widetilde{z}_l\frac{\beta_l}{\Lambda}\right)\right)\right] &= \\ n(t_i)\left[1-\Delta t\left\{(1-\theta)X(t_i)+\theta\sum_{l=1}^{L}\left(\widetilde{y}_l\frac{\beta_l}{\Lambda}\right)\right\}\right] &+ \\ \Delta t\left\{(1-\theta)\left(\sum_{l=1}^{L}\left(\lambda_l c_l(t_i)\right)+S(t_i)\right)+\theta\left(\sum_{l=1}^{L}\left(\widetilde{x}_l\lambda_l c_l(t_i)\right)+S(t_{i+1})\right)\right\} & \end{aligned} \tag{4.7}$$

4.4 The Frequency Transformed Crank-Nicolson Method

By performing the equivalent frequency transformation, as presented in Section 4.2, but solving with the Crank-Nicolson method (rather than with the implicit Euler) one obtains:

$$\begin{aligned} n(t_{i+1})e^{-\omega\Delta t}\left[1-\theta\Delta t\left(X(t_{i+1})+\sum_{l=1}^{L}\left(\widetilde{z}_l\frac{\beta_l}{\Lambda}\right)-\omega\right)\right] &= \\ n(t_i)\left[1+\Delta t\left\{(1-\theta)(X(t_i)-\omega)+\theta e^{-\omega\Delta t}\sum_{l=1}^{L}\left(\widetilde{y}_l\frac{\beta_l}{\Lambda}\right)\right\}\right] &+ \\ \Delta t\left\{(1-\theta)\left(\sum_{l=1}^{L}\left(\lambda_l c_l(t_i)\right)+S(t_i)\right)+\theta e^{-\omega\Delta t}\left(\sum_{l=1}^{L}\left(\widetilde{x}_l\lambda_l c_l(t_i)\right)+S(t_{i+1})\right)\right\} & \end{aligned} \tag{4.8}$$

5 Adaptive time steps

There is a case to be made for the implementation of the adaptive time step (as opposed to the constant time step) in the discussed numerical solution scheme.

First of all, the use of the constant time step small enough to properly deal with the fastest changes in $n(t)$ will result in a slower simulation than necessary. On the other hand, if the constant step is too large it may lead to unacceptable values of errors being committed during the calculations.

Intuitively, shorter time-steps should be used in the calculation points where neutron density experiences significant changes due to an addition or a subtraction of the reactivity into the system. Likewise, longer time-steps could be used during intervals in which the evolution of the neutron density is rather uneventful. The adaptive time step, selected and adjusted autonomously by the code, would prevent the user from being burdened with the task of selecting an appropriate time-step length.

It is also desireable to control the global error committed during calculations and ensure the stability of soluion. As explained in [Shampine, 2004], "controlling the error by adjusting the steps size can often stabilize the integration".

5.1 Error estimation

For convenience the general form of the initial value problem, Eq. 2.1, is recalled below:

$$\frac{d}{dt}y(t) = f(t, y(t))$$
$$y(t_0) = y_0$$

By denoting $y(t_i)$ as the exact (true) solution at a time step i and y_i as the approximate solution produced by a numerical method, one can define the global error at the step i as:

$$ge_i = y(t_i) - y_i \tag{5.1}$$

Ideally, one would like to directly control the true error. In most cases (when the exact solution is unavailable), it is an impossible task to precisely calculate the global committed error. As explained in [Shampine, 2004], it is rather unusual for the initial value problem solver to even estimate the global error, without solving the problem more than once, as the errors committed locally (at each step) will propagate. What can be done instead is to rely on some *reasonable approximation* of the local error. Provided that the method is stable (which in this context means that the propagated errors are damped [Shampine, 2004]).

At any given time step t_i, one can define the local solution as the solution of:

$$\frac{d}{dt}\hat{y}(t) = f(t, \hat{y}(t)) \tag{5.2}$$
$$\hat{y}(t_i) = y_i$$

The local error incurred over the step Δt, starting from t_i, is:

$$le_i = \hat{y}(t_i + \Delta t) - y_{i+1} \tag{5.3}$$

Essentailly, it is a difference between the exact solution at t_{i+1} with initial point set to the approximate solution y_i and the approximate solution at t_{i+1}. This way the errors propagating from previous steps are not present in the value of the local error.

It can also be shown that for a method of order p the local error is of order $p + 1$:

$$le_i^p \approx O(\Delta t^{p+1}) \tag{5.4}$$

At this point the same predicament as in the case of the global error exists: the exact solution of the differential equation is needed. Therfore, one needs to come up with some *reasonable approximation* of the local error. One of such approximations is described in [Shampine, 2004].

Assuming there are two methods of solving the same numerical problem, the first one of order p and the second one of order $\widetilde{p}$ (such that $p > \widetilde{p}$) , producing the following results:

$$\begin{aligned} \mathcal{R}_p &: y_i \mapsto y_{i+1} \\ \mathcal{R}_{\tilde{p}} &: y_i \mapsto \tilde{y}_{i+1} \end{aligned} \tag{5.5}$$

In accordance with definition in Eq. 5.4 the local errors are:

$$\begin{aligned} le_i^p &= y_{i+1} - \hat{y}(t_{i+1}) \approx O(\Delta t^{p+1}) \\ le_i^{\tilde{p}} &= \tilde{y}_{i+1} - \hat{y}(t_{i+1}) \approx O(\Delta t^{\tilde{p}+1}) \end{aligned} \tag{5.6}$$

The local error estimate will be then defined as a difference between solutions provided by the higher and the lower order methods:

$$est_i = y_{i+1} - \tilde{y}_{i+1} \tag{5.7}$$

This local error estimate is actually an asymptotically correct estimate of the local error of the lower order method [Shampine, 2004]. As laid out in Section 4, the point kinetics module has methods of the first and of the second order at its disposal. Thus, the local error of the first order method (the Backward Euler) will be controlled by the step size adjustment but the integration will be advanced with the second order (the Crank-Nicolson) method. Time steps obtained by controlling the error of the lower order method are smaller than what would be allowed by the second order's method accuracy. Such a scheme is also called the "local approximation" [Shampine, 1973].

5.1.1 Stability

Since the neutron density equation will be solved with the conditionally stable method (the Crank-Nicolson), and in the light of the stability requirement that was put forth in the preceding Section, any doubts about the stability of the implemented scheme will be dealt with here.

Simply put, by adjusting the time-step based on the local error estimate in the way described in Section 5.1 one inadvertently takes care of possible instability issues. To support this claim the following reasoning from [Shampine, 2004] is directly quoted:

> "When the step size is small enough to correspond to being inside the region of absolute stability, the numerical solution smooths out. Standard local error estimators recognize the y(t) is easy to approximate and casue the solver to increase the step size. Eventually the step size corresponds to being outside the stability region and the computation becomes unstable. Standard estimators recognize this instability as local errors that are unacceptably large and cause the solver to reduce the step size. Eventually the step size corresponds to being inside the stability region and the cycle repeats."

5.2 Kinetic time step control

Having a reasonable way of estimating the local truncation error, the solver will adjust the time-step, so that the magnitude of the error remains equal to or

below of the user-requested tolerance (error per step criterion).

$$\|est_i\| \leqslant TOL \tag{5.8}$$

If the above criterion is not met. The step is rejected and repeated with a shorter time step, estimated in Eq. 5.9.

Following [Shampine, 2004], the largest ratio by which we can multiply the time step, so that the local error still meets the criteria is:

$$\left(\frac{TOL}{\|est_i\|}\right)^{1/(1+p)} \tag{5.9}$$

The error is estimated for the first order method, $p = 1$.

In an attempt to reduce the number of rejected time-steps (and therefore the computational time), one would want to somehow limit the magnitude of the time-step change induced by the controller. A very crude way of achieving this is to introduce a safety factor, S, which would limit the time step change rate and to impose allowed minimum and maximum values on the said rate. Eventually, the new time step is calculated as:

$$\Delta t^{new} = \min\left\{\max\left\{\sqrt{\frac{TOL}{\|est_i\|}} \cdot S, Ratio_{min}\right\}, Ratio_{max}\right\} \Delta t^{old} \tag{5.10}$$

The parameters are set as follows:

$$\begin{aligned} S &= 0.9 \\ Ratio_{max} &= 2 \\ Ratio_{min} &= 0.5 \end{aligned} \tag{5.11}$$

In addition, the time-step value is not allowed to be lower than some minimum time-step (in this particular case the minimum step size is set to be $10^{-8}s$). In case the calculation step with the minimum time-step does not meet the condition from Eq. 5.8, the warning is issued by the code and the step is accepted nonetheless.

6 Benchmark tests

A several benchmark examlpes are presented in this section. The examples vary from step reactivity insertions, ramp reactivity, zigzag ramp to sine reactivity. A benchmark, such as the one presented here, should verify the methodoogy and numerical solutions.

There is a range of parameters, such as convergence criteria etc., that can be set in the module but for which no specific rigorous sensitivity analyses were carried out. Unless stated otherwise, in each benchmark's subsection, settings presented in table 6.0.1 were used. At this point they are somewhat arbitrary.

All results were obtained with frequency transformation method for neutron density and exponential approximation of the neutron source term. Combinations of other methods and approximations were tested to ensure the correct (and consistent) behaviour of the code. However, these results are not documentend here.

Table 6.0.1: Selected point kinetics module settings.

Parameter	Value	Unit	Description
ϵ_{n_eos}	1.00E-10	-	The end-of-step neutron density convergence criteria.
ϵ_{error}	1.00E-07	-	The maximum allowed relative local truncation error.
Δt_{min}	1.00E-08	s	The minimum allowed time-step.

The kinetics parameters for reactors considered in the benchamarks are lised in table 6.0.2.

Table 6.0.2: Kinetics parameters of benchmark reactors (after [Yang & Jevremovic, 2009]).

	Reactor A (thermal)		Reactor B (thermal)		Reactor C (fast)	
Family	λ_i	β_i	λ_i	β_i	λ_i	β_i
1	0.0127	0.000285	0.0127	0.000266	0.0129	0.0001672
2	0.0317	0.0015975	0.0317	0.001491	0.0311	0.001232
3	0.115	0.00141	0.115	0.001316	0.134	0.0009504
4	0.311	0.0030525	0.311	0.002849	0.331	0.001443
5	1.40	0.00096	1.40	0.000896	1.26	0.0004534
6	3.87	0.000195	3.87	0.000182	3.21	0.000154

6.1 Benchmark 1: step insertion of \$0.5 into reactor C

A constant reactivity of \$0.5 is inserted into the reactor C (kinetics parameters given in table 6.0.2). The neutron generation time is $\Lambda = 10^{-7}$. Table 6.1.1

shows the comparison between the reference solution (from [Yang & Jevremovic, 2009]) and the results produced by the point kinetics module. Figure 6.1.1 presents the evolution of the neutron density in time for this benchmark.

Table 6.1.1: Neutron density comparison for benchmark 1.

t[s]	Reference	Point kinetics module	Relative error
1.00E-01	2.075317E+00	2.075317E+00	0.00E+00
1.00E+00	2.655853E+00	2.655849E+00	1.51E-06
1.00E+01	1.274654E+01	1.274661E+01	-5.49E-06

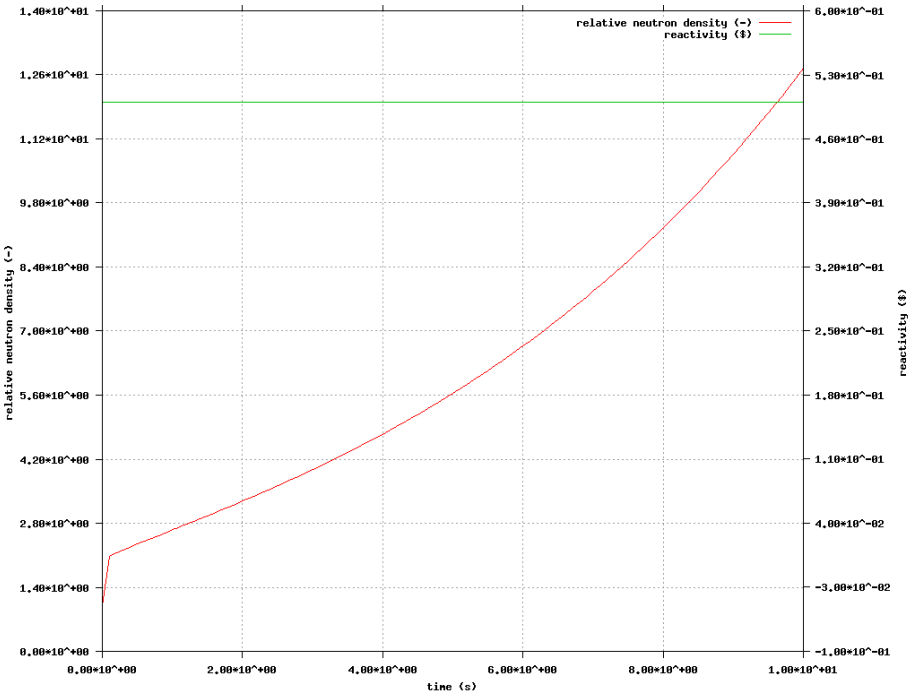

Figure 6.1.1: Neutron density evolution during benchmark 1.

6.2 Benchmark 2: step insertion of -$0.5 into reactor A

A constant reactivity of -$0.5 is inserted into the reactor A (kinetics parameters given in table 6.0.2). The neutron generation time is $\Lambda = 5 \cdot 10^{-4}$. Table 6.2.1 shows the comparison between the reference solution (from [Yang & Jevremovic, 2009]) and the results produced by the point kinetics module. Figure 6.2.1 presents the evolution of the neutron density in time for this benchmark.

Table 6.2.1: Neutron density comparison for benchmark 2.

t[s]	Reference	Point kinetics module	Relative error
1.00E-01	6.989252E-01	6.989252E-01	0.00E+00
1.00E+00	6.070536E-01	6.070536E-01	0.00E+00
1.00E+01	3.960777E-01	3.960777E-01	0.00E+00

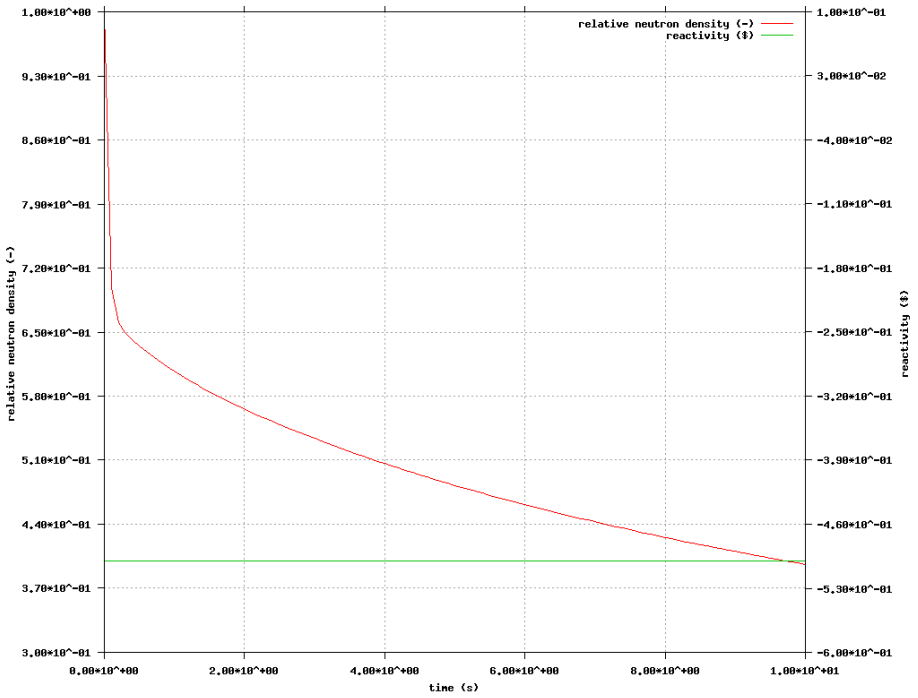

Figure 6.2.1: Neutron density evolution during benchmark 2.

6.3 Benchmark 3: ramp insertion of \$0.1/s into reactor B

A constant rate reactivty ramp of \$0.1/s is inserted into the reactor B (kinetics parameters given in table 6.0.2). The neutron generation time is $\Lambda = 2 \cdot 10^{-5}$. Table 6.3.1 shows the comparison between the reference solution (from [Yang & Jevremovic, 2009]) and the results procduced by the point kinetics module. Figure 6.3.1 presents the evolution of the neutron density in time for this benchmark. The figure is limited to the first 5 seconds of the ramp.

Table 6.3.1: Neutron density comparison for benchmark 3.

t[s]	Reference	Point kinetics module	Relative error
2.00E+00	1.338200E+00	1.338200E+00	0.00E+00
4.00E+00	2.228442E+00	2.228442E+00	0.00E+00
6.00E+00	5.582052E+00	5.582052E+00	0.00E+00
8.00E+00	4.278630E+01	4.278626E+01	9.35E-07
1.00E+01	4.511636E+05	4.511607E+05	6.43E-06

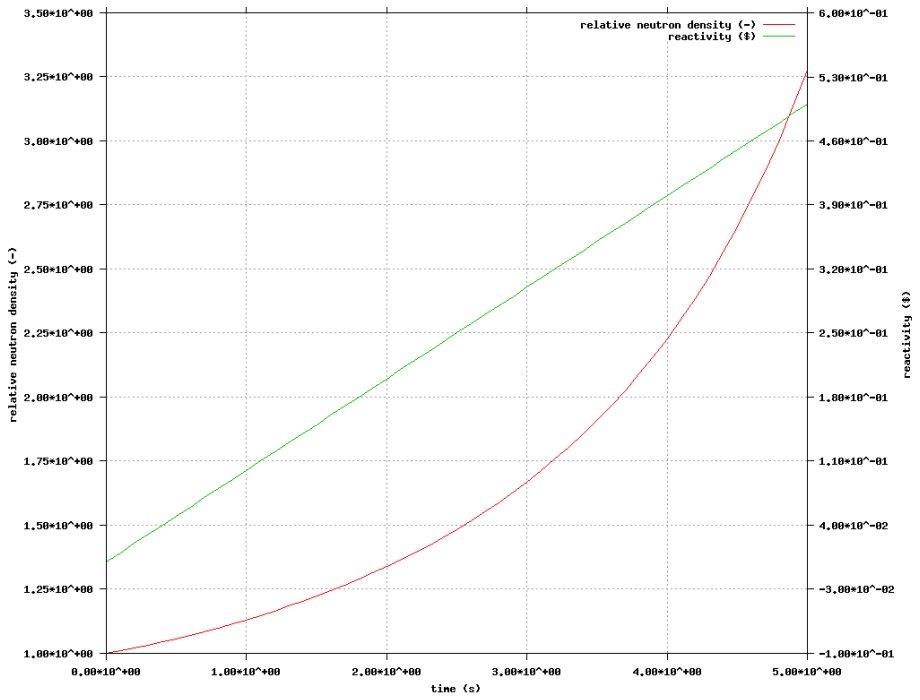

Figure 6.3.1: Neutron density evolution during benchmark 3.

6.4 Benchmark 4: ramp insertion of $1/s into reactor C.

A constant rate reactivty ramp of $1.0/s is inserted into the reactor C (kinetics parameters given in table 6.0.2). The neutron generation time is $\Lambda = 10^{-7}$. Table 6.4.1 shows the comparison between the reference solution (from [Yang & Jevremovic, 2009]) and the results produced by the point kinetics module. Figure 6.4.1 presents the evolution of the neutron density in time for this benchmark.

Table 6.4.1: Neutron density comparison for benchmark 4.

t[s]	Reference	Point kinetics module	Relative error
5.00E-01	2.136407E+00	2.136409E+00	-9.36E-07
1.00E+00	1.207813E+03	1.207787E+03	2.15E-05

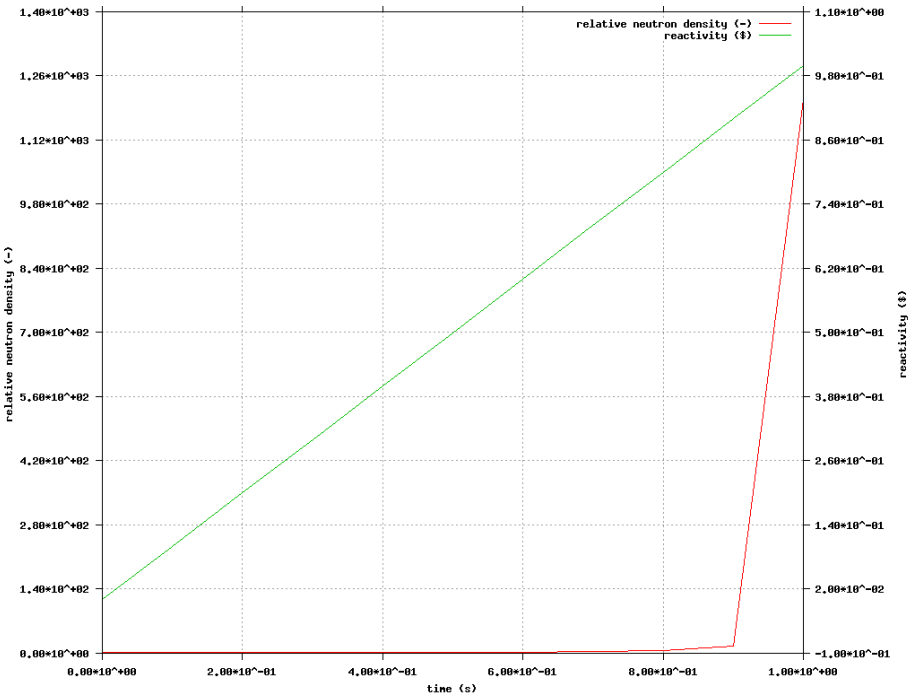

Figure 6.4.1: Neutron density evolution during benchmark 4.

6.5 Benchmark 5: zigzag ramp insertion into reactor A.

A zigzag ramp reactivity is inserted into the reactor A (kinetics parameters given in table 6.0.2). The neutron generation time is $\Lambda = 5 \cdot 10^{-4}$. The zigzag reactivity is a function of time of the following form:

$$\rho(t) = \begin{cases} 0.0075t & 0 \leqslant t \leqslant 0.5 \\ -0.0075(t-0.5) + 0.00375 & 0.5 \leqslant t < 1 \\ 0.0075(t-1) & 1 \leqslant t \leqslant 1.5 \\ 0.00375 & 1.5 \leqslant t \leqslant 10 \end{cases} \tag{6.1}$$

Table 6.5.1 shows the comparison between the reference solution (from [Yang & Jevremovic, 2009]) and the results produced by the point kinetics module. Figure 6.5.1 presents the evolution of the neutron density in time for this benchmark.

Table 6.5.1: Neutron density comparison for benchmark 5.

t[s]	Reference	Point kinetics module	Relative error
5.00E-01	1.721422E+00	1.721422E+00	0.00E+00
1.00E+00	1.211127E+00	1.211127E+00	0.00E+00
1.50E+00	1.892226E+00	1.892226E+00	0.00E+00
2.00E+00	2.521601E+00	2.521600E+00	3.97E-07
1.00E+01	1.204711E+01	1.204710E+01	8.30E-07

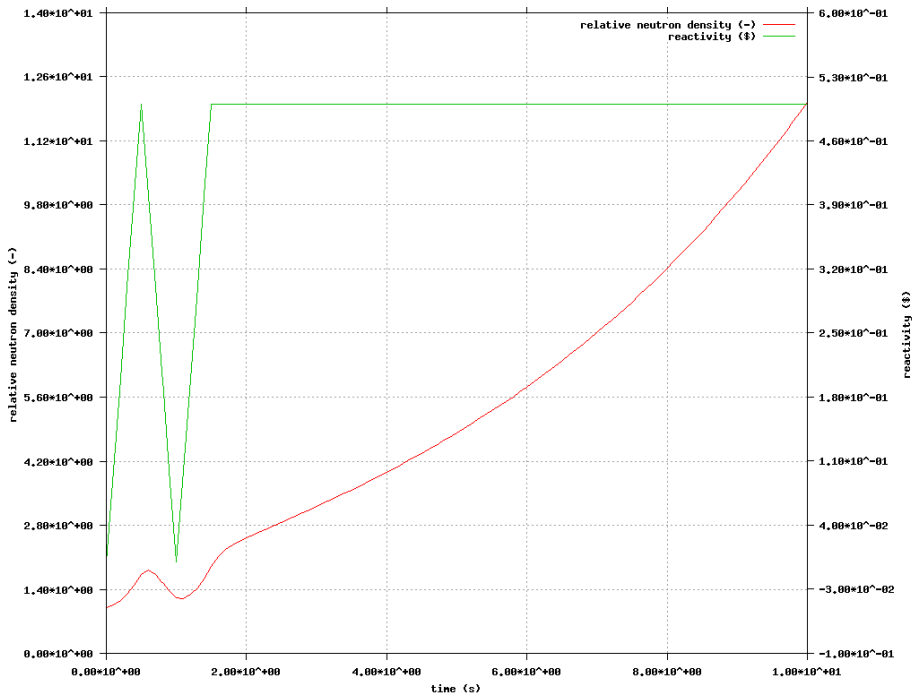

Figure 6.5.1: Neutron density evolution during benchmark 5.

6.6 Benchmark 6: sinusoidal reactivity.

The last benchmark checks the response to the sinusoidal reactivity oscillation. The reactivity over time is shown in equation 6.2.

$$\rho(t) = \frac{8\beta}{8 + \lambda T} \sin(\pi t / T) \tag{6.2}$$

For this benchmark a fast reactor with only one precursor family is considered. The following kinetics parameters are used: $\beta = 0.0079$, $\lambda = 0.077 s^{-1}$, $\Lambda = 10 s^{-8}$. Four half-periods T are considered. Table 6.6.1 shows the comparison between the reference solution (from [Ganapol, 2009]) and the results

produced by the point kinetics module. Two plots for the extreme half-periods ($50s^{-1}$ and $350s^{-1}$) are presented as figure 6.6.1 and figure 6.6.2.

Table 6.6.1: Neutron density comparison for benchmark 6.

Half-period T [1/s]	t[s]	Reference	Point kinetics module	Relative error
50	3.9108E+01	6.15339E+01	6.153249E+01	2.29E-05
150	1.3732E+02	9.58190E+01	9.581806E+01	9.81E-06
250	2.3712E+02	1.13458E+02	1.134567E+02	1.15E-05
350	3.3707E+02	1.23820E+02	1.238191E+02	7.27E-06

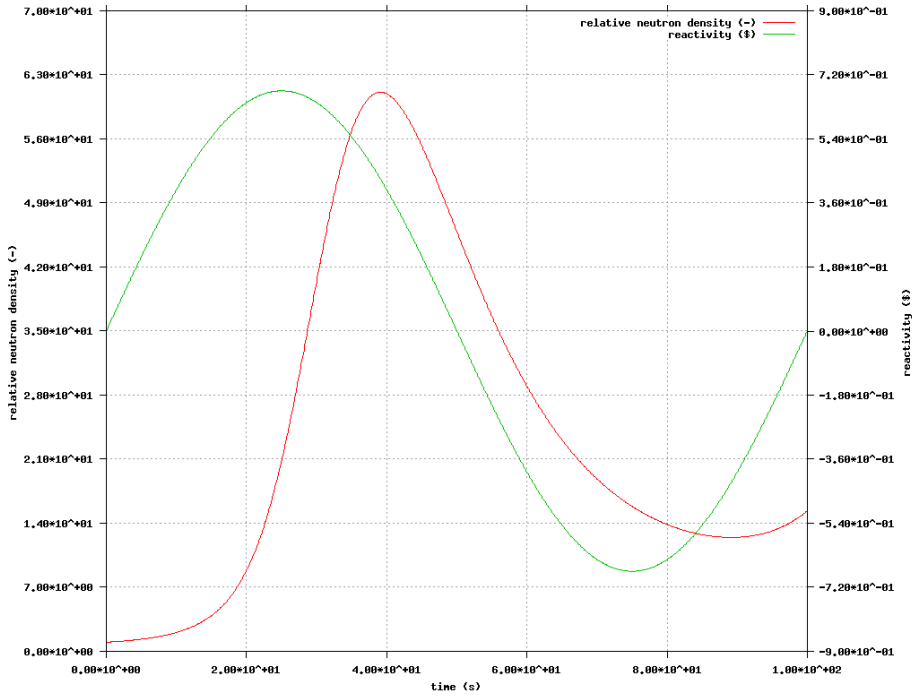

Figure 6.6.1: Neutron density evolution during benchmark 6 with half-period 50 1/s.

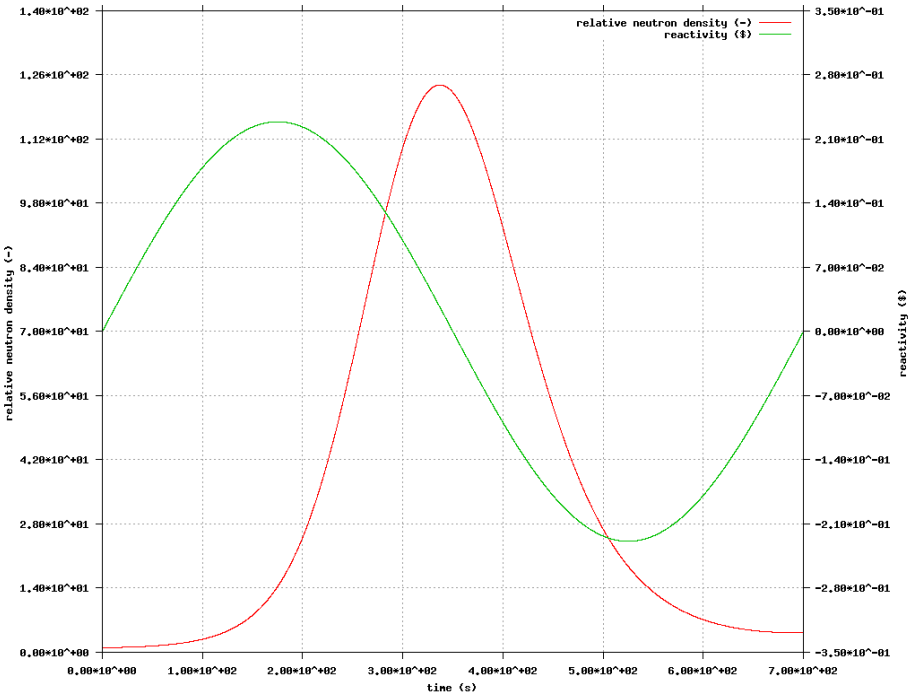

Figure 6.6.2: Neutron density evolution during benchmark 6 with half-period 350 1/s.

References

[Shampine, 2004] Shampine, L.F. (2004). Error Estimation and Control of ODEs. Journal of Scientific Computing. 25, 3-16.

[Shampine, 1973] Shampine, L.F. (1973). Local extrapolation in the solution of ordinary differential equations. Mathematics of Computation. 27, 91-97.

[Ganapol, 2009] Ganapol, B.D. (2009). A refined way of solving reactor point kinetics equations for imposed reactivity insertions. Nuclear Technology and Radiation Protection 24. 157-165.

[Yang & Jevremovic, 2009] Yang Xue, Jevremovic Tatjana (2009). Revisiting the Rosenbrock numerical solutions of the reactor point kinetics equation with numerous examples. Nuclear Technology and Radiation Protection 24. 3-12.

[Anglart, 2013] Anglart H. (2013). Nuclear Reactor Dynamics and Stability. Warsaw: Warsaw Univeisity of Technology Publishing House. ISBN 978-83-7814-089-4.