UPPSALA
UNIVERSITET

Point Kinetics with Temperature Feedbacks

Authors: Victoria Sedig and John Danielsson
Supervisor: Zsolt Elter

Project in Computational Science: Report

February 2022

PROJECT REPORT

Department of Information Technology

Contents

Abstract

The goal of this project is to make a simulator of the behaviour of a simplified nuclear reactor. This includes solving a system of ODE:s consisting of 10 equations using numerical methods. The implementation is done in Python. Some extra features of the implementation for specific circumstances for the reactor are added. The resulting simulator works well within realistic conditions.

Nomenclature

A_i	Area of the cladding inner surface, for the whole reactor core (m^2)
α_f	Doppler coefficient (\$/K)
α_m	Moderator temperature coefficient (\$/K)
β	Total delayed neutron fraction (1)
β_i	Partial delayed neutron fraction of the i:th group (1)
η	Ratio of the power released in the fuel to the total power (1)
λ	Average neutron generation time (s)
λ_i	Decay constant of the delayed neutrons belonging to the i:th group (1/s)
$\rho(t)$	The reactivity over time(\$)
ρ	Reactivity which is the sum of $\rho_r + \rho_f + \rho_m$ (\$)
ρ_r	Reactivity which can be inserted by for example removing the control rods
$\rho_f(t)$	Reactivity feedback from the change of the fuel temperature over time
$\rho_m(t)$	Reactivity feedback from the change of the moderator temperature over time
A_o	Area of the cladding outer surface, for the whole reactor core (m^2)
c_0	Zero order coefficient (1)
c_1	First order coefficient (1/°C)
c_2	Second order coefficient ($1/{}^\circ\mathrm{C}^2$)
C_i	Number of precursor nuclei of the delayed neutrons belonging to the i:th group (1/cm3)
C_{pc}	Specific heat of the cladding (J/kg/K)

C_{pf}	Specific heat of the fuel (J/kg/K)
C_{pm}	Specific heat with the cladding and coolant temperatures
c_{x0}	Zero order coefficient (J/kg/K);
c_{x1}	First order coefficient (J/kg/K /°C)
c_{x2}	Second order coefficient (J/kg/K /°C^2)
d	Effective thickness of the gap between the fuel and cladding (m)
f_0	Zero order coefficient (W/K/m)
f_1	First order coefficient (W/K/m/°C)
f_2	Second order coefficient (W/K/m/°C^2)
m_0	Zero order coefficient (W/K/m^2)
m_1	First order coefficient (W/K/m^2/°C)
m_2	Second order coefficient (W/K/m^2/°C^2
M_c	Total mass of the cladding present in the reactor core (kg)
M_f	Total mass of the fuel present in the reactor core (kg)
M_m	Total mass of the coolant present in the reactor core (kg)
n	Number of neutrons (1/$cm3$), which is proportional to the power generated in the fuel
P_d	Decay heat power (J)
t	Time (s)
$T_c(t)$	Average temperature of the cladding (°C)
$T_f(t)$	Average time dependent fuel temperature (°C)
$T_{ave,0}$	$T_{ave}(t)$ at time $t = 0$ (°C)
$T_{ave}(t)$	Average moderator (cooler) temperature (°C)
$T_{f,0}$	$T_f(t)$ at time $t = 0$ (°C)
T_{in}	Inlet coolant temperature (°C)

U_{fc}	Coefficient of the heat transfer from the fuel into the cladding, for the whole reactor core (W/K)
W_{core}	Coolant mass flow rate (through the reactor core) (kg/s)

The descriptions and the parameters in Nomenclature are taken from Khaled Sayed Mahmoud Ph.D dissertation ”NUMERICAL MODELING OF REACTIVITY EXCURSION ACCIDENTS IN SMALL LIGHT WATER REACTORS” ([6]).

1 Introduction

In this project a model for point kinetics with temperature feedback for a nuclear reactor is built. It is important for educational purposes to be able to show and simulate the behaviour of a simplified nuclear reactor. With the model one can follow the development of the temperature and power in the reactor with respect to different circumstances and input parameters. The model is based on the equation system seen in Section 2.3 "Point Kinetics Model". The simulator is verified using the ***ReMeg*** software discussed in Section 4.

2 Theory

2.1 Nuclear Reactor

A large scale nuclear reactor is in general used to collect electricity through fission. There are also small scale nuclear reactors that do not produce electricity which are used in research and experiments. Fission in a nuclear reactor is the event when an atom's core is split into two or more parts by being hit by a neutron, called neutron induced fission. During the split particles and energy are released. Neutrons could be released, which then can cause the split of more atoms, leading to a chain reaction. Atoms fit for fission are unstable atoms that would likely split if a neutron is fired upon them. The energy released from the event is the kinetic energy with the released particles and the resulting atoms and electromagnetic radiation. The atoms, particles and radiation can then collide with other atoms in the reactor, transforming the kinetic energy into heat ([1]).

The unstable element in a reactor is called fuel. The fuel is in the most common case in the format of a rod and typically Uranium 235 is used (although there exists other possible fuels). Around the fuel there is a layer of cladding. Cladding is a material such as Aluminum, stainless steel, and zirconium which is there to make a barrier from the fuel to the other parts of the reactor and to protect from the release of radioactive materials. Outside of the cladding there is a moderator which is a liquid, for example water. The heat from the fission is absorbed by the moderator. As the moderator is heated up it is transported to a pressuriser and then to a stem generator where the moderator is turned into steam. The steam then enables a turbine to spin and produce electricity.

To ensure control over how the system increases or slows down in fission events there are control-rods (most common to use rods as shape) that can be put in and out of the system. The control rods are made from a material that absorbs neutrons such as boron, cadmium or hafnium. When the rod is inserted it will absorb neutrons that otherwise could have caused a fission event, which means the number of fission events will be less than without the rod.

Around the whole system there is a metre-thick concrete and steel container to protect the system from intrusion and to protect the people outside the system from radiation. In Figure 1 there is a simple model of a nuclear reactor ([1], [3], [2], [4]).

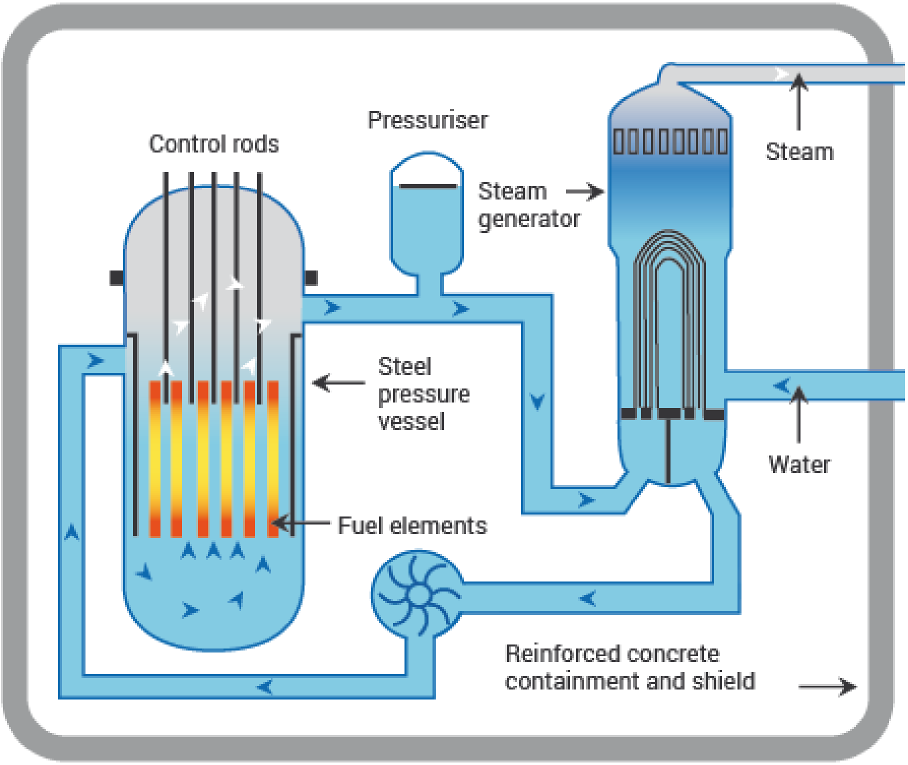


Figure 1: A model of a nuclear reactor. The figure was collected at the website https://world-nuclear.org/ ([2]).

In this report a part of the reactor is referred to as a shutdown rod. It is the same as a control rod except it is not initially in the system. It can be inserted to decrease or shut down the power in the reactor.

2.2 Delayed Neutrons

During the fission process the nuclide splits into smaller fragments as described previously. If these atoms are unstable they can in turn go through a beta chain decay where they release a neutron after a time delay that depends on the half-life of the $\beta-$ decay. There are many processes that can occur to release the so-called delayed neutrons by traversing the decay chain in different ways. This means neutrons continue to be generated due to the original fission event with a time delay. The fission products that can result in later emitted neutrons are called neutron precursors. For practicality these delayed neutrons can be arranged into a few distinct groups characterized by the same decay constant λ and fraction β. In this project the delayed neutrons are divided into six groups, also referred to as generations. Delayed neutrons are only a small fraction of the overall neutron population but because they appear late they make the reactor more controllable. The time scale for the half lives for delayed neutrons can vary up to several seconds for uranium-235, compared to prompt neutrons which are born within 10^{-14} seconds.

2.3 Criticality and Reactivity

The criticality of a reactor is a parameter to describe how the population of the neutrons propagates in the reactor. The criticality depends on a multiplication factor k. To simplify the presentation, this factor can be seen describing the number of neutrons in the current *i*th generation compared to neutrons in previous generation *i-1*. The criticality has different states depending on what value the multiplication factor has. The criticality states are:

- *subcritical* if $k < 1$, meaning that the number of neutrons is decreasing.
- *critical* if k = 1, meaning that the number of neutrons is stable.
- *supercritical* if $k > 1$, meaning that the number of neutrons is increasing.

The reactivity ρ in the reactor can also be described with k according to:

$$\rho(t) = \frac{k(t) - 1}{k(t)}, \tag{1}$$

which describes how far the reactor is from being in a critical state. Inserting reactivity means increasing the criticality of the reactor, and reversely removing reactivity means lowering the criticality. It can be seen from the equation that ρ is unitless, however due to reactivity often being very small compared to the criticality it is often written in pcm (percent mille or thousands of a percent).

One unit to measure reactivity is called the $ unit. One $ is defined as the interval between the reactor being prompt critical and critical only with delayed neutrons. This gives a measure for reactivity that can be safely added. A reactivity insertion smaller than one $ only makes the reactor possibly super-critical with the addition of delayed neutrons, which gives a longer time for safety mechanisms to intervene. An increase larger than a $ would instead make the reactor super-critical with only prompt neutrons, giving a criticality change and power change faster than can be stopped by safety mechanisms.

2.4 Point Kinetics Model

A point kinetics model is a type of model of a nuclear reactors kinetics where the modulations are handled as if the reactor was a point instead of a 3D space. For example in a real reactor the temperatures are different in different locations of the moderator or the rods. This is ignored in the point kinetics model.

The behaviour of a nuclear reactor can be modeled by a point kinetics model in the form of a system of ordinary differential equations. The model is not a representation of a full scale and real reactor. It is a system describing some basic functioning for a small test nuclear reactor. The system of equations is the following:

$$\frac{dP_k(t)}{dt} = \frac{\beta}{\ell}[\rho(t) - 1] \cdot P_k(t) + \sum_{i=1}^{6} \lambda_i C_i(t), \tag{2}$$

$$\frac{dC_i(t)}{dt} = \frac{\beta_i}{\ell} P_k(t) - \lambda_i C_i(t), \quad i = 1, 2, ...6, \tag{3}$$

$$\frac{dT_f(t)}{dt} = \frac{\eta(P_k(t) + P_d(t)) - U_{fc}[T_f(t)](T_f(t) - T_c(t))}{M_f C_{pf}[T_f(t)]}, \tag{4}$$

$$\frac{dT_c(t)}{dt} = \frac{U_{fc}[T_f(t)](T_f(t) - T_c(t)) - U_{cm}[T_c(t), T_{ave}(t)](T_c(t) - T_{ave}(t))}{M_c C_{pc}[T_c(t)]}, \tag{5}$$

$$\frac{dT_{ave}(t)}{dt} = \frac{U_{cm}[T_c(t), T_{ave}(t)](T_c(t) - T_{ave}(t)) + (1 - \eta)(P_k(t) + P_d(t))}{M_m C_{pm}[T_{ave}(t)]} - \\ - \frac{W_c C_{pm}[T_{ave}(t)] \cdot 2 \cdot (T_{ave}(t) - T_{in})}{M_m C_{pm}[T_{ave}(t)]}, \tag{6}$$

Equations (2) and (3) are taken from the one-speed neutron diffusion equation, while equations (4-6) are based on energy balance equations.

P_k is the power generated in the reactor core, $T_f(t)$ is the average time dependent fuel temperature (°C), $T_c(t)$ is the average time dependent clad temperature (°C), $T_{ave}(t)$ is the average time dependent moderator temperature (°C), and C_i is the number of precursor nuclei of the delayed neutrons belonging to the i:th group (1/cm3). A list of all the parameters and constants can be seen in Nomenclature ([6]).

In addition to these differential equations the heat transfer coefficients and heat capacities have to be computed. The heat transfer coefficient between the fuel and cladding is written as

$$U_{fc} = (f_0 + f_1 \cdot T_f(t) + f_2 \cdot T_f(t)^2)\frac{A_i}{d}. \tag{7}$$

The heat transfer coefficient between the cladding and moderator is written as:

$$U_{cm} = \frac{(m_0 + m_1 \cdot T_{av}(t) + m_2, T_{ave}(t)^2) \cdot (T_c - T_{ave})^{0.1} \cdot A_o}{(c_0 + c_1 \cdot T_{ave}(t) + c_2 \cdot T_{ave}(t)^2)^{0.24}}. \tag{8}$$

The specific heat capacities for fuel, cladding and moderator are written as:

$$C_{px} = c_{x0} + c_{x1} \cdot T_x + C_{x2} \cdot T_x^2 \tag{9}$$

with x being f, c, or ave.

The function $\rho(t)$ which describes the reactivity insertion is composed of three parts, with the user defined added reactivity as ρ_r, the feedback from the fuel being ρ_f and the feedback from the moderator being ρ_m so that

$$\rho = \rho_r + \rho_f + \rho_m, \tag{10}$$

with the feedback being modelled as

$$\rho_f = \alpha_f(T_f(t) - T_{f,0}), \tag{11}$$

$$\rho_m = \alpha_m(T_{ave}(t) - T_{ave,0}). \tag{12}$$

Here, $T_{f,0}$ and $T_{av,0}$ are the initial conditions for the fuel and moderator. A of all the parameters and constants can be seen in Nomenclature ([6]).

2.5 Only Prompt Neutrons

In the case that there are no delayed neutrons the ODE system of (2) - (6) does not work correctly since it assumes that there are delayed neutrons. A way to model the system without delayed neutrons is:

$$\frac{dn}{dt} = \frac{k-1}{l}n(t), \tag{13}$$

$$n(t) = n_0 exp\left(\frac{k-1}{l}t\right) = n_0 exp\left(\frac{-t}{T}\right), \tag{14}$$

There is a linear connection between the number of neutrons n(t) and power P(t), where $P_k(t) = c_k \cdot n(t)$ ([5]). Furthermore the relation between criticality and reactivity as in equation (1) allows us to rewrite the equation as:

$$\frac{dP_k}{dt} = \frac{\rho}{l(1-\rho)}P_k(t), \tag{15}$$

where the parameters and constants are the same as explained in Section 2.3.

2.6 Stationary States

When modelling for example how inserting control rods will affect the running of a reactor at a given stationary initial state other methods need to be implemented to find correct stable initial conditions for all temperatures. The equations for initial conditions to be solved can be written as

$$T_{av}(0) = \frac{P}{2W_{core}C_{pm}} + T_{in}, \tag{16}$$

$$T_c(0) = \frac{\eta P}{U_{cm}} + T_{av}(0), \tag{17}$$

$$T_f(0) = \frac{\eta P}{U_{fc}} + T_c(0). \tag{18}$$

Since the variables C_{pm}, U_{cm} and U_{fc} depend on the sought temperatures, these equations need to be solved iteratively until they converge to an acceptably stable value.

3 Implementations of the Model

3.1 Shut Down Rods

An additional possible input to the model is inserting a shutdown rod at a given start time with a specified magnitude and speed. This results in a decrease in reactivity and can be used to model a manual response to a sudden reactivity insertion.
The negative reactivity of the shutdown rod is modelled as

$$\rho_s = \frac{M}{1 + \exp\left(C(t_0 - t)\right)}, \tag{19}$$

where M is the magnitude of the insertion, C is the speed of the insertion and t_0 is the start time.

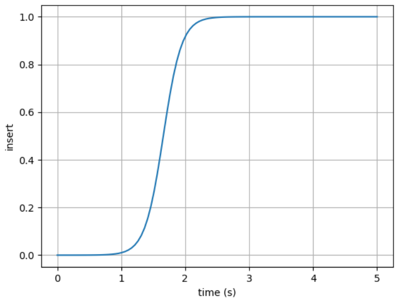


Figure 2: Example of magnitude of negative reactivity from a shutdown rod

3.2 Incursion Function

The reactivity is inserted into the system with an incursion function. The default function for this is a linear function that increases with time. However it is possible to define other functions and use those instead.

3.3 No Delayed Neutrons

In the case that there are no delayed neutrons the ODE in Section 2.5 (equations (13)- (14)) is used. The reactor has an input where it is possible to specify if there should be delayed neutrons or not.

4 Numerical Experiments

The experiments are run with different conditions and states such as:

- different initial values,
- different incursion functions,
- shutdown rods,
- delayed neutrons.

The plots from the experiments contain information about the reactivity, thermal power and temperature over time.

To check and compare that the experiments produce reasonable results the executable version of the software *ReMeg* is used. It is developed by Dr. Sandor Feher.

5 Python Package

The code is made as a package on GitHub to enable to download and run the simulator on a computer for free.

There are some files necessary to make a proper python package. These are a readme file, a test file and an example file. A readme file has relevant information and an introduction to the package. A test file runs tests on the code to make sure that everything is working as it is intended to do. An example file illustrates examples for how the package can be used.

6 Method for Solving the System of Equations

6.1 Scipy

The python library SciPy is used to solve the model in section 2.3. SciPy offers solvers for a large range of problems such as optimization, statistics and differential equations. In this project the SciPy functions solve_ivp and simps are used.

Solve_ivp solves one dimension ODEs given an initial value. As input it takes the function to solve, the start time, the end time, the numerical method and optional arguments [8].

Simps stands for Simpson's method and is a method to numerically approximate integrals. This method is used to integrate the power over the solution interval, giving the total power emitted during the given time [9].

6.2 Numerical Methods

As mentioned in Section 3.1 the solve_ivp takes a numerical method as input. The methods that are used in this project are the explicit methods RK45 (Runge Kutta 45) and the implicit methods lsoda and BDF (backward differentiation formula).

In the implicit calculation the Jacobian is calculated. The implicit methods are not optimal at calculating the Jacobian as they use approximations to find it. Therefor a function calculating the exact Jacobian is developed. The Jacobian is calculated by:

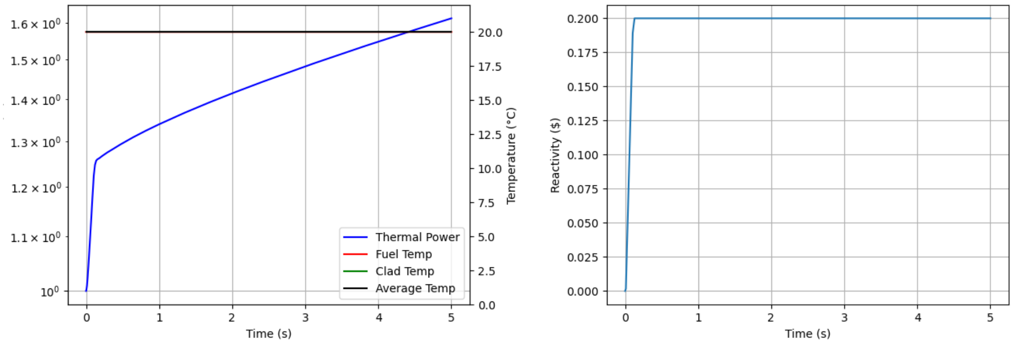
$$J = \begin{pmatrix} \frac{\beta}{\ell}[\rho(t)-1] & \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 & \lambda_6 & 0 & 0 & 0 \\ \frac{\beta_1}{l} & -\lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\beta_2}{l} & 0 & -\lambda_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\beta_3}{l} & 0 & 0 & -\lambda_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\beta_4}{l} & 0 & 0 & 0 & -\lambda_4 & 0 & 0 & 0 & 0 & 0 \\ \frac{\beta_5}{l} & 0 & 0 & 0 & 0 & -\lambda_5 & 0 & 0 & 0 & 0 \\ \frac{\beta_6}{l} & 0 & 0 & 0 & 0 & 0 & -\lambda_6 & 0 & 0 & 0 \\ \frac{\eta}{M_f C_{pf}} & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{U_f}{M_f C_{pf}} & \frac{U_f}{M_f C_{pf}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{U_{fc}}{M_c C_{pc}} & \frac{-U_{fc}-U_{cm}}{M_c C_{pc}} & \frac{U_{cm}}{M_c C_{pc}} \\ \frac{1-\eta}{M_m C_{pm}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{U_{cm}}{M_m C_{pm}} & \frac{-U_{cm}-2W_{core}C_{pm}}{M_m C_{pm}} \end{pmatrix}, \tag{20}$$

where J is the Jacobian and the other parameters are explained in Section 2.3.

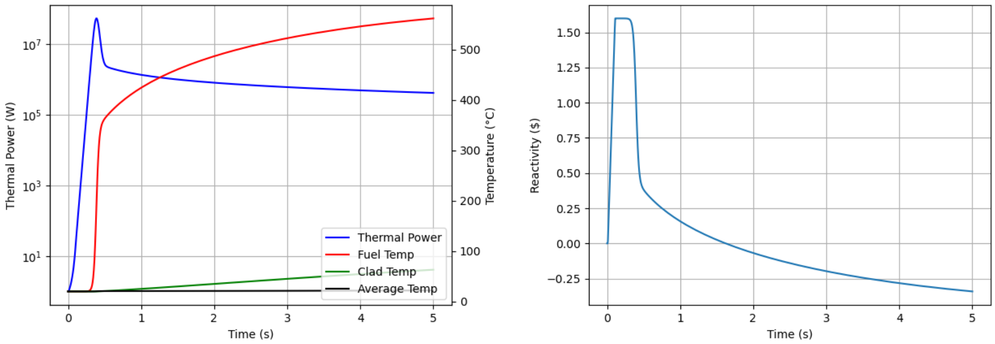
7 Results

7.1 Simulations

In figures 3, 4 and 5 simulations of the systems thermal power, temperatures and reactivity is shown. There are no shutdown rods. Figure 5 demonstrates the behaviour of the system for different neutron generation times.

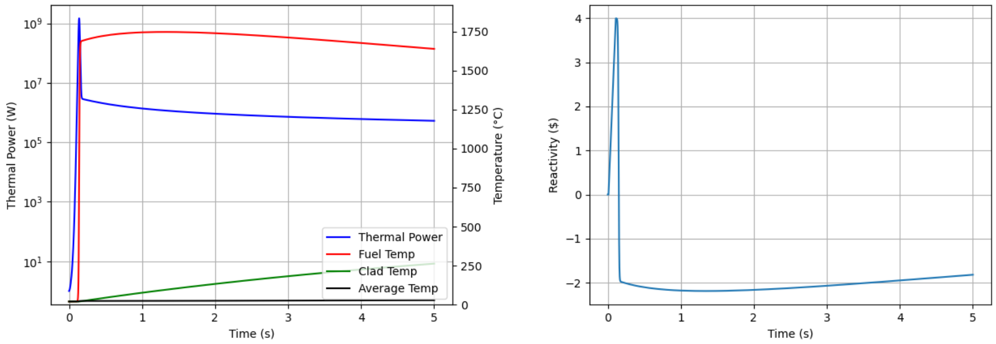


(a) Power for $\rho = 2, \delta\rho = 0.1$ (b) Reactivity for $\rho = 2, \delta\rho = 0.1$

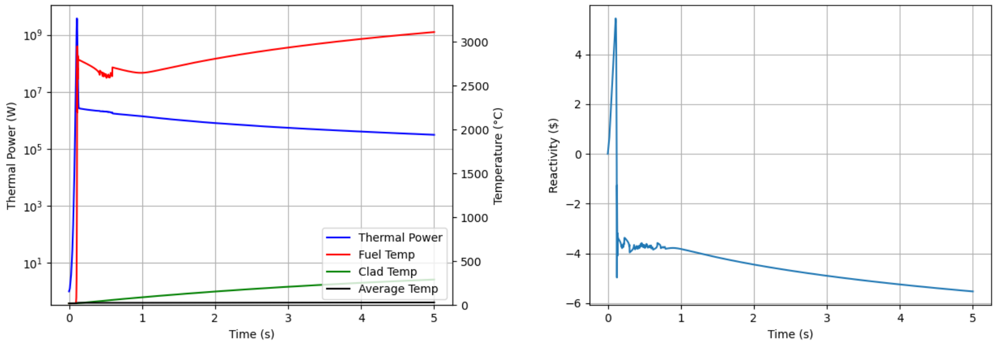


(c) Power for $\rho = 16, \delta\rho = 0.1$ (d) Reactivity for $\rho = 16, \delta\rho = 0.1$

Figure 3: Examples of power evolution due to reactivity inserted over 0.1 seconds for the system with some initial values. Plot b) is the same system as plot a) and plot d) is the same system as plot c). The x-axis is time (s) for each subplot. For subplot a and c the y-axis is thermal power (W) For subplot b and d the y-axis is reactivity ($).

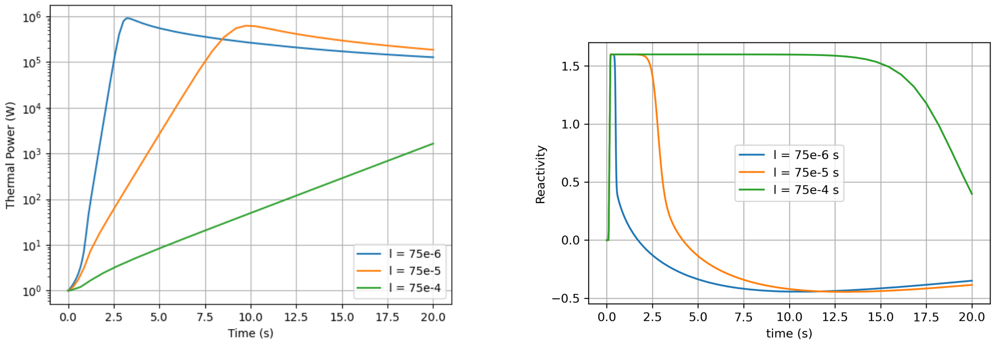


(a) Power for $\rho = 40, \delta\rho = 0.1$ (b) Reactivity for $\rho = 40, \delta\rho = 0.1$



(c) Power for $\rho = 52, \delta\rho = 0.1$ (d) Reactivity for $\rho = 52, \delta\rho = 0.1$

Figure 4: Examples of simulations for the system with some initial values. Plot b) is the same system as plot a) and plot d) is the same system as plot c). The x-axis is time (s) for each subplot. For subplot a and c the y-axis is thermal power (W) For subplot b and d the y-axis is reactivity ($).

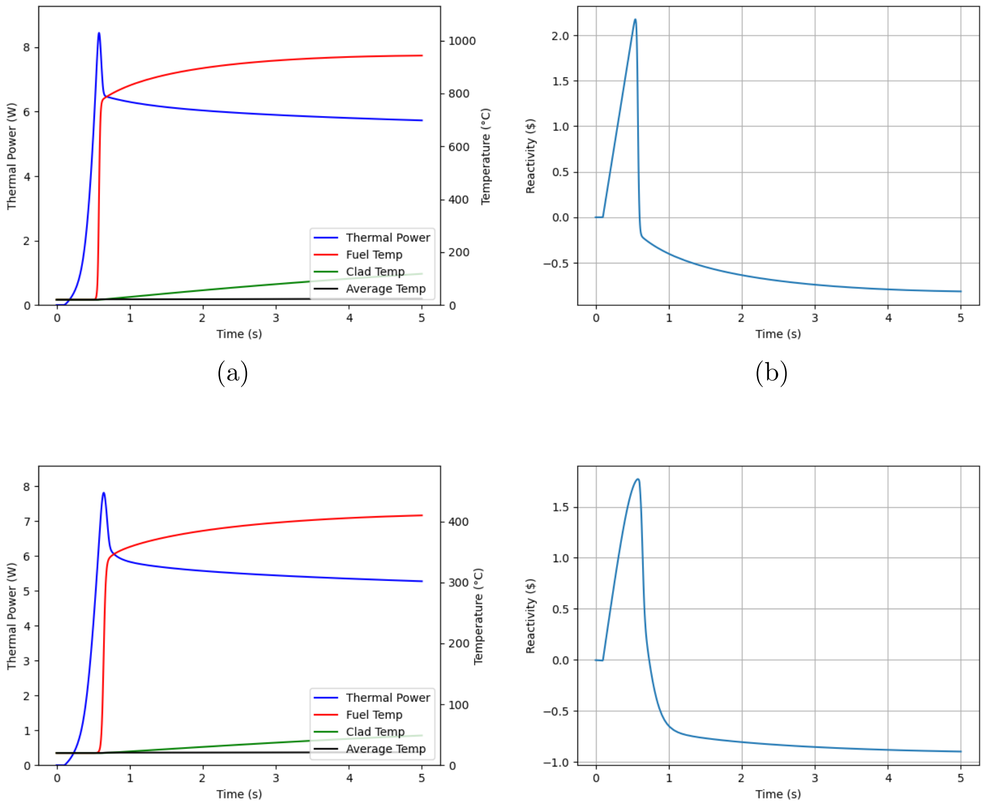


(a) Power for neutrons with generation times l (b) ρ for neutrons with generation times l

Figure 5: Examples of simulations of the reactor with the same reactivity insertion for different neutron generation times. For both plots the x-axis is time (s). For plot a the y-axis is Power (W) and for plot b it is reactivity (\$).

7.2 Simulations with Shutdown Rods

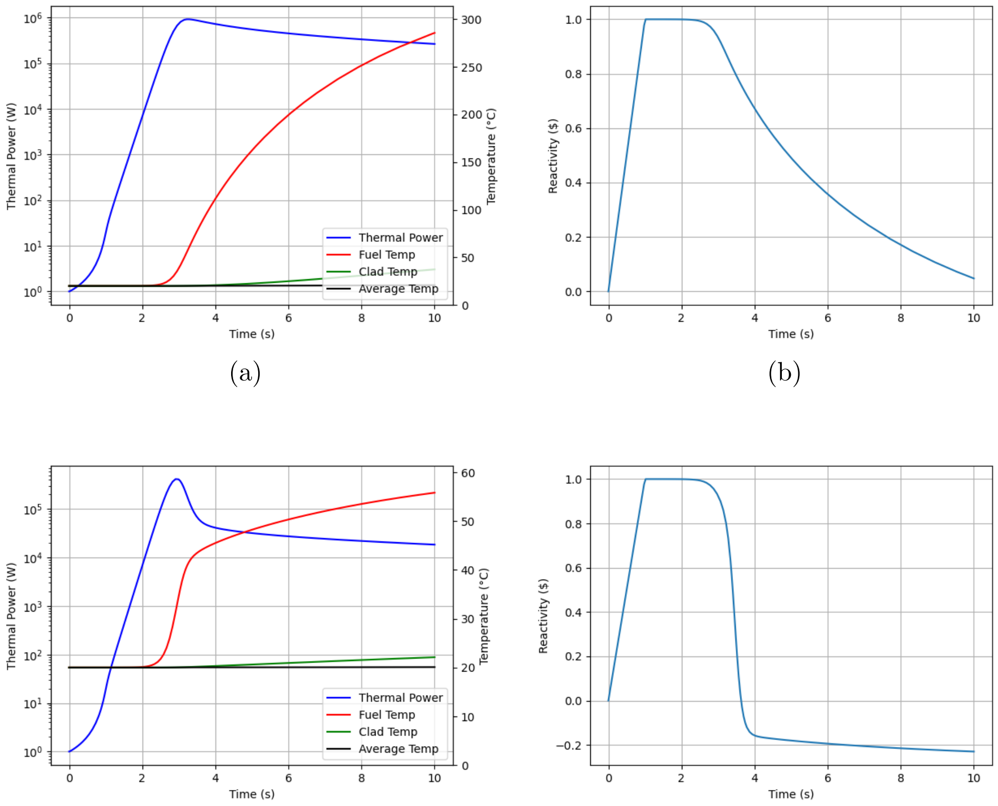
Figure 6 and 7 visualizes simulations of the systems thermal power, temperatures and reactivity when shutdown rods are inserted. In figure 6 the shutdown rods are inserted around the same time as the reactivity incursion. In figure 7 the shutdown rods are inserted slightly after the reactivity incursion.



(a) (b)

(c) (d)

Figure 6: Examples of simulations for the system with some initial value. Plot b) is the same system as plot a) and plot d) is the same system as plot c). System a) and b) is without shutdown rods insertion and system c) and d is with. The x-axis is time (s) for each subplot. For subplot a and b the y-axis is thermal power (W) For subplot c and d the y-axis is reactivity ($). In subplot b and d shutdown rods are inserted.



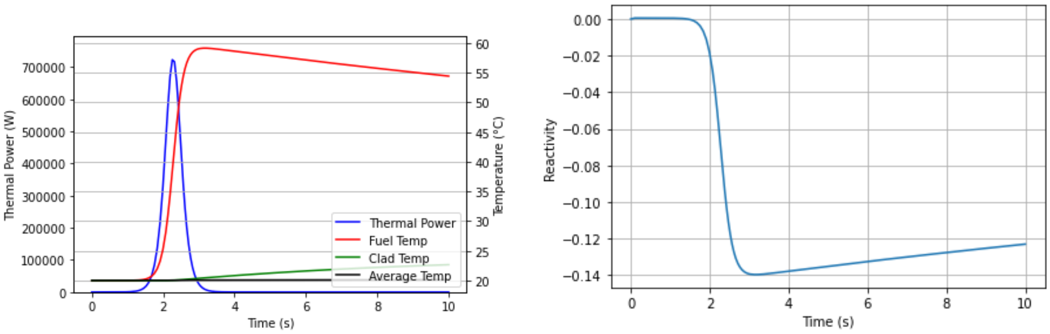
(a) (b)

(c) (d)

Figure 7: Examples of simulations for the system with some initial value. Plot b) is the same system as plot a) and plot d) is the same system as plot c). System a) and b) is without shutdown rods insertion and system c) and d is with. The x-axis is time (s) for each subplot. For subplot a and b the y-axis is thermal power (W) For subplot c and d the y-axis is reactivity ($). In subplot b and d shutdown rods are inserted.

7.3 Simulations with only Prompt Neutrons

Figure 8 and shows simulations of the systems thermal power, temperatures, reactivity and power when there are no delayed neutrons.



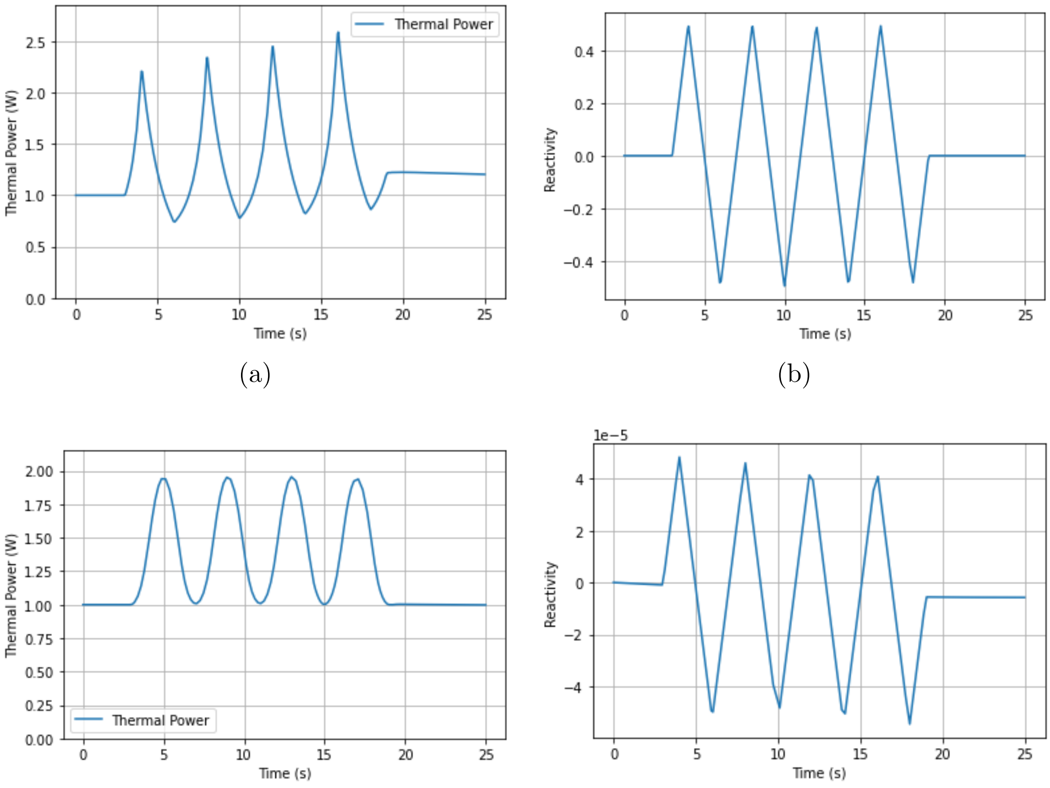
(a) Power and Temperature with only prompt neutrons

(b) Reactivity with only prompt neutrons

Figure 8: Examples of simulations of the reactor when there a only prompt neutrons. For both plots the x-axis is time (s). For plot a the y-axis is thermal power (W) and for plot b it is temperature (°C).

7.4 Using custom functions

Figure 9 shows simulations of the systems' thermal power when a custom reactivity function is implemented. The magnitude is 0.5 for the case with delayed neutrons and 5×10^{-5} for the case with only prompt neutrons.



(a) (b)

(c) (d)

Figure 9: Examples of simulations for the system with custom reactivity function. Plot b) is the reactivity for the system in plot a) and plot d) is the reactivity for system in plot c). System a) and b) is with delayed neutrons and system c) and d is without. The x-axis is time (s) for each subplot. For subplot a and b the y-axis is thermal power (W) For subplot c and d the y-axis is reactivity.

8 Discussion

8.1 Behaviour of the System

Some behaviors of the system can be highlighted. When a small reactivity is inserted as in Figure 3 a) a small prompt jump in power is induced since the criticality is above 1. After the jump there is a constant positive reactivity as demonstrated in Figure 3 b) which upholds an exponential increase. As the reactivity insertion is small there are no major temperature changes. Therefore no feedback that kicks in.

In Figure 3 c) on the other hand there is a larger reactivity insertion which pushes the power to several megawatts. At this stage the fuel temperature rapidly increases. This results in the feedback forcing down the reactivity as in Figure 3 d) and leaving the power in an equilibrium.

The higher the reactivity insertion, the sharper the increase in power and temperature will be which can be seen in Figure 4. Inserting a reactivity like in a) and b) gives an increase of several magnitudes for both fuel temperature and thermal power in less than 0.2 seconds.

When the increase in reactivity is too large and too rapid it is no longer possible for the numerical solvers to compute a realistic solution, and large nonphysical fluctuations appear in the temperature, power, and reactivity. These errors occur at a level of reactivity insertion far above anything that would be realistic in a reactor of this kind. This behaviour is demonstrated in Figure 4 in c) and d).

In Figure 5 the behaviour for power and reactivity with different neutron generation times are visualized. As seen in plot b) larger generation time gives a higher maintained reactivity. This leads to a slower increase of power, seen in plot a). The system can handle more reactivity for the longer neutron generation times. This shows how the timescale of reactions in the reactor will depend on many parameters, among them the neutron generation time. Reactors can have different power alterations with the same added reactivity because of different neutron generation times.

8.2 Behaviour with Shutdown Rods

In Figure 6, plot a) and b) represents a system without shutdown rods and plot c) and d) represents a system with shutdown rods. The system with a shutdown rod does not reach the same peak or height as the system without.

When the shutdown rods are inserted after the reactivity incursion as in Figure 7 the difference between having shutdown rods or not is clearer. Plot a) and b) represents a system without shutdown rods and plot c) and d) represents a system with shutdown rods.

8.3 Behaviour without Delayed Neutrons

In the case of a system consisting only of prompt neutrons the system will act differently since there are no delayed neutrons slowing down the system rate. The results for a small reactivity insertion of 0.001 can be seen in 8 a) and b).

The behavior from the prompt neutrons is more rapid than when including delayed neutrons, which can be seen in the large difference in the magnitude of reactivity insertion. It also drops down to a power close to zero quicker compared to when delayed neutrons continue to be generated. Furthermore since the reactivity is at a much smaller scale than with the delayed neutrons the temperature change needed to decrease the system criticality is also much smaller.

8.4 Custom Functions

As seen in Figure 9 the total inserted reactivity is zero but the system response is different depending on parameters. For the scenario with the delayed neutrons each peak pushes the power higher than the previous and it remains at a higher power level than the initial power even after the function input has ceased. This allows the system to be pushed to a higher power even while no cumulative excess reactivity has been inserted.

When modelling with only prompt neutrons the behavior instead returns to zero after each reactivity insertion has been added and subtracted. The resulting power is kept at the same as the initial power.

9 Conclusion

The simulator behaves in typical circumstances as expected. It gives in general a realistic propagation for the basic behaviour of a nuclear reactor. The comparisons to the *ReMeg* gives alike plots.

At a certain point, for example when a very large amount of reactivity is inserted, it does not give reasonable results. However the kind of input needed for this to happen are not reasonable or realistic. A real reactor could not have these type of extreme inputs.

10 Acknowledgements

Thanks to Z. Elter for his commitment and suggestions for improvements of this project and report.

Thanks to Dr. Sandor Feher from the Budapest University of Technology for providing us the executable version of the ***ReMeg*** software, which was used for comparison reasons.

References

[1] How Does a Nuclear Reactor Work?, World Nuclear Association, `https://www.world-nuclear.org/nuclear-essentials/how-does-a-nuclear-reactor-work.aspx`, collected 16/11/2021

[2] Nuclear Power Reactors?, World Nuclear Association, `https://world-nuclear.org/information-library/nuclear-fuel-cycle/nuclear-power-reactors/nuclear-power-reactors.aspx`, updated Oct 2021, collected 16/11/2021

[3] NUCLEAR 101: How Does a Nuclear Reactor Work?, Office of NUCLEAR ENERGY, `https://www.energy.gov/ne/articles/nuclear-101-how-does-nuclear-reactor-work`, updated 29/03/2021, collected 16/11/2021

[4] Cladding, U.S. Nuclear Regulatory Commission, `https://www.nrc.gov/reading-rm/basic-ref/glossary/cladding.html`, updated 09/03/2021, collected 16/11/2021

[5] Reactor physics with Python Lecture Notes, Zs. Elter. E. Branger, M. Preston, Division of Applied Nuclear Physics, Uppsala University, edition 18/06/2021, collected 17/11/2021

[6] NUMERICAL MODELING OF REACTIVITY EXCURSION ACCIDENTS IN SMALL LIGHT WATER REACTORS, Khaled Sayed Mahmoud, Ph.D. Dissertation, Department of Nuclear Techniques, Institute of Nuclear Techniques, Budapest University of Technology and Economics, released 2006, collected 22/11/2021

[7] Implicit vs. Explicit Numerical Methods, FLOW 3D, `https://www.flow3d.com/resources/cfd-101/numerical-issues/implicit-versus-explicit-numerical-methods/`, collected 07/12/2021

[8] scipy.integrate.solve_ivp, SciPy, `https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html`, collected 07/12/2021

[9] scipy.integrate.simps, SciPy, `https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html`, updated 11/05/2014, collected 07/12/2021