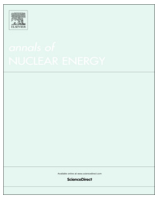
Annals of Nuclear Energy 146 (2020) 107630

Contents lists available at ScienceDirect

Annals of Nuclear Energy

journal homepage: www.elsevier.com/locate/anucene



PC-based JSI research reactor simulator

Jan Malec [a,b], Dan Toškan [a], Luka Snoj [a,b,*]

[a] Jožef Stefan Institute, Reactor Physics Division, Jamova cesta 39, 1000 Ljubljana, Slovenia
[b] University of Ljubljana, Faculty of Mathematics and Physics, Jadranska cesta 19, 1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 14 February 2020
Received in revised form 31 May 2020
Accepted 2 June 2020
Available online 20 June 2020

Keywords:
Research reactor
Reactor simulator
Nuclear education and training
Reactor physics

ABSTRACT

A real-time research reactor simulator was developed at the Jožef Stefan Institute for education and training of students and future reactor operators, especially in countries with limited or no access to a research reactor. The research reactor simulator (RRS) simulates time behaviour of reactor power, fuel temperature and reactivity by using the 6-group point kinetics equation with feedback. It features temperature feedback mechanisms and xenon poisoning. Using graphics acceleration to render simulation results and a simple integration scheme with a short simulation step to propagate physics achieves low latency between a user's input and display of simulation results, which is important for research reactors that can perform fast transients. The simulator targets TRIGA-type reactors, but all physical parameters can be changed on the fly. Thus, the simulator can be easily adapted to simulate other reactors.

1. Introduction

Research nuclear reactors are commonly used for education and training of nuclear power plant (NPP) personnel, research reactor (RR) operators and students of physics, nuclear engineering and related fields (Böck et al., 2015; Böck, 2014; Snoj et al., 2012, 2014). However, since not all countries and educational institutions have direct access to research reactors, students can carry out exercises in other countries or use simulators. Several NPP simulators are available on the market, from PC-based to full scope, but there are only a few research reactor simulators. The world's first PC-based TRIGA research reactor simulator was developed by the Microsimtech company (Microsimtech triga simulator, 2019). It is intended for education on the concepts of delayed neutron effect, multiplication factor, criticality, reactivity control by rods and boron concentration (although the simulator features boron concentration changes, it is important to note that in all real TRIGA-type reactors, reactivity control is done using control rods and demineralised water as a coolant and moderator), feedback on fuel (Doppler) and moderator temperatures and Xenon and Samarium poisoning. Another type of research reactor simulator features a console for moving the control rods and a small display for reactor data developed by the Czech Technical University. A simple simulator of a nuclear reactor is the McMaster Nuclear Simulator (Gilbert, n.d.) built by Dave Gilbert. One disadvantage of these simulators is the relatively low degree of flexibility in adjusting parameters such as delayed neutron fraction, mean neutron generation time, control rod worth curves, control rod insertion velocity and various modes of operation. In other words, they work well as tools to familiarise students with reactor operation and behaviour, but they pay less attention to reactor physics.

At the Jožef Stefan Institute (JSI) in Ljubljana, Slovenia, we developed a research reactor simulator that focuses on reactor physics. Our goal was to develop a PC-based low power ($P < 1$ MW) research reactor simulator for countries/institutions with no or limited access to real RR. In countries with easy access to RR, it can be used for classroom presentation of RR experiments prior to operating the real reactor. The requirements were that it should be robust, fast, flexible and user-friendly.

The JSI operates a 250 kW TRIGA research reactor, which was used as a target for start-up configuration and validation purposes. However, due to the flexibility requirement, the simulator is general and can be applied to other RRs as well.

The paper is structured as follows. Section 2 presents the physics of the simulator together with validation of the models. Section 3 presents the technical implementation of the simulator and an analysis of the graphical user interface (GUI) design from the user interface design perspective. In Section 4, the team presents plans for future simulator development. Section 5 describes the conclusions and some uses for the simulator.

[*] Corresponding author at: Jožef Stefan Institute, Reactor Physics Division, Jamova cesta 39, 1000 Ljubljana, Slovenia.

E-mail addresses: jan.malec@ijs.si (J. Malec), luka.snoj@ijs.si (L. Snoj).

https://doi.org/10.1016/j.anucene.2020.107630
0306-4549/© 2020 The Author(s). Published by Elsevier Ltd.

2. Physical models

The time behaviour of the neutron population in a nuclear reactor can be described by point kinetic equations (Duderstadt and Hamilton, 1976):

$$\frac{dN(t)}{dt} = \frac{\rho(t) - \beta_{eff}}{l} N(t) + \sum_{i=1}^{6} \lambda_i C_i(t) + N_s(t)$$

$$\frac{dC_{i(t)}}{dt} = \frac{\beta_i}{l} N(t) - \lambda_i C_i(t) i = 1, \cdots, 6 \tag{1}$$

$N(t)$ represents the total time-dependent population of the neutrons in the reactor core, and $C_i(t)$ represents the neutron populations of individual groups. l and β_{eff} denote the prompt neutron lifetime and the effective delayed neutron fraction, respectively. The delayed neutron fractions and the decay constants for individual groups are denoted by β_i and λ_i, respectively. The default values are listed in Table 1. $N_s(t)$ is the neutron source activity, which is by default a constant, user-defined value. The neutron source function $N_s(t)$ can also follow a user-defined waveform in a similar fashion as the control rod position, which we will describe later.

The reactor's physical properties such as the kinetic parameters in Eq. (1) are adjustable in the simulator. The default simulator configuration mimics the behaviour of the TRIGA Mark II reactor operated by the Jožef Stefan Institute in Slovenia (Štancar et al., 2018). The default configuration is also important to allow validation, a procedure during which the developers verify how closely the behaviour of the simulator matches the behaviour of the real reactor.

The delayed neutron fractions β_i, the decay constants λ_i and the average delayed neutron lifetime for the TRIGA Mark II reactor in Slovenia are available from the source code of the digital reactivity metre (Lengar et al., 2012). The decay constants λ_i are expressed in sec^{-1} and are connected to the mean time τ_i as $\tau_i = \frac{1}{\lambda_i}$. Delayed neutron fractions β_i are directly used as a sum to calculate β_{eff}. Time-dependent reactivity $\rho(t)$is calculated as

$$\rho(t) = \rho_0(t) - \alpha(T(t))T(t) - \rho_{Xe}(t), \tag{2}$$

where ρ_0 is the reactivity of a cold reactor core free of any neutron poison, calculated based on the control rod positions, $\alpha(T)$ is the temperature-dependent fuel temperature feedback coefficient and $\rho_{Xe}(t)$ is the reactivity change due to ^{135}Xe poisoning.

The delayed neutron populations $C_{i(t)}$ are calculated as defined in paragraph 2.2 plotted on a separate graph for all six groups. The simulator offers an option to disregard the delayed neutron production by setting $\beta_i = 0, i = 1, \cdots, 6$. The time evaluation of the kinetic equations (1) is an iterative process with a fixed step size dt.

Point kinetic equations are solved using an explicit, fourth order Runge-Kutta solver with a fixed step size (Saidu and Waziri, 2010). One equation is evaluated to propagate the neutron population $N(t)$; and another six equations are used to evaluate the neutron concentration for each individual group C_i.

2.1. Reactor power and flux normalisation

The quantity $N(t)$ in Eq. (1) is the total number of neutrons in the reactor core. This quantity is convenient for the propagation of kinetic equations and is clearly defined, but further calculations require the thermal reactor power and the neutron flux.

The reactor power can be calculated by multiplying the macroscopic cross-section for fission, average neutron velocity and the energy released by a single fission:

$$P(t) = \mathrm{N}(t)\Sigma_f v E_f \tag{3}$$

The default macroscopic cross-section for fission Σ_f was calculated using OpenMC (Romano and Harper, 2017) with core configuration no. 132 (Štancar et al., 2018). Default neutron velocity v was chosen to be 2200 m/s, which is the velocity of thermal neutrons with an energy of 0.025 eV, and the average energy released by fission E_f is set to 200 MeV (Duderstadt and Hamilton, 1976).

Following its definition, the neutron flux can be calculated by normalising the total number of neutrons N(t) to the core volume V_c and multiplying it with the velocity of thermal neutrons v :

$$\Phi = N * \frac{v}{V_c} \tag{4}$$

2.2. Delayed neutron group normalisation

The graphic display of delayed neutron fractions $C_i(t)$ from Eq. (1) in an absolute scale is not practical due to large concentration variations between the groups. For this reason, the delayed group concentrations are normalised to their stationary concentrations. The stationary concentrations of delayed neutrons are calculated by assuming stationary conditions $\frac{dN(t)}{dt} = \frac{dC_{i(t)}}{dt} = 0$ in Eq. (1) and solving for delayed neutron concentrations. The resulting formula for stationary concentrations is:

$$C_{i,stationary}(t) = \frac{1}{\sum_j^6 C_j(t)} \frac{\lambda_i}{\beta_i} \left(\frac{\sum_j^6 \lambda_j}{\sum_j^6 \beta_j} \right) \tag{5}$$

2.3. Temperature feedback effects

The simulator can model the temperatures of fuel and cooling water and calculate the feedback effects of fuel temperature on reactivity. The temperature model describing the fuel heating is inspired by the Fuchs-Hansen (Duderstadt and Hamilton, 1976) adiabatic model with the addition of fuel cooling. The fuel temperature is affected by the heat generated by fission, temperature conduction between the fuel elements and the surrounding water and thermal radiation. The heat released by fission can be calculated by converting the neutron population to thermal power and integrating it over the simulation time interval.

The total heat exchange between the fuel elements and the surroundings is described using a single semiempirical polynomial from the TRIGLAV code (Peršič et al., 2017, 2000) that is used for calculation of the stationary temperature of standard TRIGA fuel elements cooled by natural convection:

$$T_{festat.} = a_1 P_{el} + a_2 P_{el}^2 + a_3 P_{el}^3 + T_w \tag{6}$$

where T_{festat} is the temperature of the fuel element in the stationary conditions, P_{el} is the power produced by each fuel element (Štancar et al., 2018; Snoj and Ravnik, 2008; Štancar and Snoj, 2017; Žerovnik et al., 2014; Goričanec et al., 2018); a_1, a_3 and a_3 are constants fitted to experimental data and T_w is the temperature of the cooling water. In the stationary conditions, the heat exchange between the fuel elements and the surroundings is equal to the power produced in the fuel. In the research reactor simulator, the roots of the polynomial (6) are calculated analytically. The normalised power P_{el} is calculated by dividing the total reactor power from Eq. (3) by the number of fuel elements in the core. The resulting function tells us the stationary reactor power $P_{fe,stat}$ in relationship to the fuel temperature, which is equal to the power exchanged between the fuel and the environment. The total change in temperature in one simulation step is calculated as

$$T'_{fe} = T_{fe} + \int \frac{Pdt}{c_p(t)} + \int \frac{P_{fe,stat}dt}{c_p(t)} \tag{7}$$

T_{fe} and T'_{fe} are fuel element temperatures at the previous and next iterations, and temperature-dependent fuel capacityC_f is calculated using the formula from (Simnad, 1981):

$$c_p(t) = 333\frac{J}{kgK} + 0.678\frac{J}{kgK^2}(T_{fe}[\mathrm{K}] - 273\mathrm{K}). \tag{8}$$

The cooling water is heated by the fuel elements and cooled by conduction to the surroundings and active cooling. In the pulse operation, where the term $P_{fe,stat}$ is small compared to the reactor power, Eq. (7) describes the behaviour in the adiabatic approximation commonly used for pulse analysis (Vavtar and Snoj, 2019).

T_w and T'_w describe the reactor tank water temperature before and after each simulation step. During each simulation step, the reactor water is heated by fission and cooled by the heat convection to air, heat transfer to the concrete vessel (Štancar and Snoj, 2017) and an active cooling system, described by

$$T'_w = T_w + \frac{Pdt}{C_w} - Q_{air}dt - Q_{concrete}dt - \frac{A}{C_w}dt \tag{9}$$

The heating is represented by the second term in Eq. (9). $C_w$is the heat capacity of the cooling water expressed in the units of J/K and calculated from the water volume V_w as $C_w = V_w\rho_w c_w$. Default water volume V_w is approximated from the technical drawings, available in (Ravnik, 2019), whereas water density $\rho_w = 998.2\frac{kg}{m^3}$ and specific heat capacity $c_w = 4.185\frac{kJ}{kg\,K}at\,20^\circ\mathrm{C}$ were taken from (Thermophysical Properties of Materials for Nuclear Engineering, 2019).

Q_{air} represents the heat that escaped the water tank by convection to air. After evaluating the thermophysical properties of air at T_{air} = 20 °C, the following relationship can be derived (Žagar et al., 1999):

$$Q_{air} = 13.6(T_{air} - T_w)^{\frac{4}{3}}\frac{W}{K^{\frac{4}{3}}} \tag{10}$$

It is assumed that when the air temperature is higher than the water temperature, no convection occurs. The heat exchanged between the water in the reactor tank and the concrete wall of the reactor can be described by (Žagar et al., 1999):

$$Q_{concrete} = 250(T_{concrete} - T_w)\frac{W}{K} \tag{11}$$

Active cooling removes a constant amount of heat from the system and lowers the temperature for Adt during each step until the water reaches room temperature and is not used in reactor simulator validation. The default value for A is 240 kW and can be adjusted by the simulator user. The ambient air and the concrete reactor wall are assumed to behave as reservoirs with infinite heat capacity.

The effect of fuel temperature on reactivity ρ_T is calculated using a temperature-dependent feedback coefficient $\alpha_T(T)$:

$$\rho_T = \alpha_T(T)\mathrm{T}. \tag{12}$$

Mimicking the temperature dependence measured at the JSI TRIGA, a function consisting of two straight lines with adjustable parameters is used:

$$\alpha_T(T) = \begin{cases} aT + b; aT + b > cT + d \\ cT + d; aT + b < cT + d. \end{cases} \tag{13}$$

The parameters a,b,c,d, and d' are chosen by the simulator user. The default parameters that are valid for the JSI TRIGA are chosen from (Ravnik, 1991).

2.4. Xenon poisoning

The RR simulator simulates poisoning by ^{135}Xe with a cross-section for neutron absorption of $2,651 \cdot 10^6$ barn. Simulation of poisoning by the strongest contributor, samarium ^{149}Sm, with the cross-section lower by two orders of magnitude is not implemented. ^{135}Xe is generated directly from uranium fission and the decay of ^{135}I. The iodine is a daughter product of ^{135}Te. In the next equations, the decay of ^{135}Te to ^{135}I is neglected due to short half-life of 19 s. The ^{135}I production is simulated with a differential equation under the assumption that it is generated directly from uranium (Duderstadt and Hamilton, 1976):

$$\frac{\partial I}{\partial t} = \gamma_I\Sigma_f\Phi(r,t) - \lambda_I I(r,t), \tag{14}$$

where λ_I is the β decay constant for iodine,γ_I I is the iodine fission yield and Σ_f is the macroscopic

cross-section for fission. The reactor flux is calculated using Eq. (4). With the iodine concentration under control, the differential equation for changes in the ^{135}Xe concentration can be written as:

$$\frac{\partial X}{\partial t} = \gamma_X\mathrm{X}(\mathrm{r},\mathrm{t})\Phi(r,t) + \lambda_I I(r,t) - \lambda_X X(r,t) - \sigma_a^X\mathrm{X}(\mathrm{r},\mathrm{t})\Phi(r,t). \tag{15}$$

γ_X and λ_I are the xenon generation probability and the xenon decay constant, and σ_a^X is the microscopic cross-section for absorption. Eqs. (14) and (15) are solved at each computation step using a first order solver of differential equations. Since changes of the poison concentrations in the reactor core occur on a much slower time scale than the changes in neutron concentrations, using a fourth order differential equation solver would introduce unnecessary computational overhead with such a small step size. The step size was kept constant to preserve a consistent number of data points for all time-dependent physical quantities and to avoid the need for interpolation between points in time.

The effect of ^{135}Xe on reactivity is calculated as:

$$\Delta\rho_{Xe} \cong \frac{\sigma_a^X\mathrm{X}(\mathrm{t})}{\Sigma_a} \cong \frac{\sigma_a^X\mathrm{X}(\mathrm{t})}{\nu\Sigma_f} \tag{16}$$

The approximation $\Sigma_a \cong \nu\Sigma_f$ derives from the equation $k = \frac{\nu\Sigma_f}{\Sigma_c+\Sigma_f+L}$ under the assumption that fission cross section is larger than capture or in other words fission presents majority of the absorption.

2.5. Constant power initial conditions

The simulator allows the user to set the reactor in a stationary power condition in which the net reactivity is zero. Steady-state neutron populations can be calculated by setting the derivatives on the left side of the point kinetics equations (1) to zero and solving the equations for $N(t)$ and C_i. The resulting equations can be written as:

$$N(t) = \frac{P(t)\mathrm{c}}{\Sigma_{total}v_{avg}} \tag{17}$$

$$C_i = \frac{N(t)\beta_i}{\lambda_i l} \tag{18}$$

The stationary temperature of the fuel is calculated from the semi-empirical polynomial (6), and the stationary water temperature of the water in the reactor tank is calculated by balancing the reactor power with the amount of heat exchanged between the water tank and the concrete pool in Eq. (9).

Since the stationary water temperature can be higher than the temperature that initiates the automatic shutdown system, the steady-state algorithms do not set the water temperature higher than 40 °C.

2.6. Operational limits

The operational limits are checked during every iteration of the simulation so the emergency shutdown signals can be triggered when necessary. The power scram signal, fuel temperature scram signal and water temperature scram signal shut down the reactor when the corresponding values surpass the limit values that can be set from the user interface. The period scram activates only if the calculated reactor period surpasses the limit value for more than a certain amount of time. The reactor period is numerically calculated as an exponentially weighted moving average of N_T points. The moving average is calculated as:

$$T_s = \sum_{i=1}^{N_T} \frac{T_k^{N_T-i-2}}{T_i T_w}, \tag{19}$$

where T_i is the reactor period corresponding to one calculation cycle, computed as:

$$T_i = \frac{dt}{\log \frac{N_{i+1}}{N_i}}. \tag{20}$$

$T_w$is the sum of weights that represent a geometric series, computed as:

$$T_w = \frac{1 - T_k^{N_T-2}}{T_k} \tag{21}$$

T_k is a weighting factor. Automatic emergency shutdown 6 s after the reactor pulse was also implemented to mimic the behaviour of the JSI TRIGA. However, it is important to note that all the parameters can be adjusted.

2.7. Control rods

The reactivity of the reactor is controlled by three control rods: the safety control rod, the regulating control rod and the Shim or compensating control rod. The purpose of using more than one control rod is to allow the user to reproduce the experiments commonly executed at research reactors such as the rod swap experiment. The third control rod was added to educate the user about the purpose of the safety control rod, which needs to be fully withdrawn for the reactor to reach criticality.

When the simulator is started, the control rods are operated in manual mode. To move the control rods, the operator can change the target rod position using the graphic widgets, a computer keyboard or a compatible USB controller. During each simulation step, the control rods move towards the target rod position with a constant speed and infinite acceleration. The default rod speeds were chosen to mimic the control speeds of JSI TRIGA in terms of inserted reactivity per unit of time, i.e. x pcm/min.

Another way of moving the control rods is to use one of the automatic rod control modes. In the automatic mode, the simulator checks whether the difference between the current reactor power and the desired reactor power is greater than a pre-defined threshold and moves the target position up or down until the threshold condition is satisfied. The target position moves at a fixed, predefined rate of 1 step per second. The automatic mode has an additional option that avoids changing the rod position when the reactor period comes within 90% of the reactor period limit.

The automatic modes 'square wave', 'sine wave' and 'saw tooth' move the control rod position to follow the corresponding user-configurable waveforms. An option is available to limit the control rod movement velocity while following the pre-defined waveforms. When this option is turned off, the square wave, sine wave and saw tooth modes only change the target rod position, just as in the manual and automatic modes.

When the pulse mode is enabled, the positions of the safety and Shim rods behave as they do in the manual mode, but the regulating rod is fully inserted. In this mode, the position controls of the regulating rod will move the regulating rod magnet, which defines the final position that will be reached when the regulating rod is ejected. During the firing sequence, the regulatory rod is moved towards the end position with a constant acceleration 'a'.

During runtime, the simulator reads the control rod values from the lookup table. The table is dynamically generated each time the rod parameters are changed. Configurable parameters are total rod worth and parameters a_1 and a_2 of the following Bezier curves in the parametric form:

$$B_x(t) = t^3 + 3t^2(1-t)a_0 + 3t(1-t)^2 a_1 \text{ and } B_y(t) = t^3 + 3t^2(1-t) \tag{22}$$

Fig. 1 shows the integral curve of the pulsing rod belonging to JSI TRIGA core '231′ and Eq. (22) with different combinations of parameters $\boldsymbol{a}_0$ and $\boldsymbol{a}_1$. The core numbers are assigned to JSI TRIGA cores each time the configuration is changed. By setting $\boldsymbol{a}_0 = 1$ and $\boldsymbol{a}_1 = 0$, the user can configure the control rod to have a linear calibration curve. The curve with $\boldsymbol{a}_0 = 0.57$ and $\boldsymbol{a}_1 = 0.68$ closely mimics the calibration curve of the pulsing control rod. The Bezier curves approximate the worth of real control rods accurately.

The neutron source is represented by the term $N_s(t)$. In the start-up configuration, the neutron source provides a constant supply of 10^5 neutrons per second. In addition to changing the source intensity, the user can chose to configure a time-dependent neutron source that changes intensity according to the sine wave, square wave and saw tooth wave, which behave similarly to the corresponding automatic control modes for the control rods. The time-dependent modes can be used to simulate active neutron sources, like those that are used at accelerator-driven systems.

2.8. Pulse analysis

Each time one of the control rods is ejected with a pulsing mechanism, an automatic timer starts counting five seconds, and the simulator starts integrating the reactor power using Newton's method (Širca and Horvat, 2011). As soon as the timer runs out, the integration of the pulse energy stops, and an automatic fast shutdown procedure (SCRAM) is initiated. The pulse, pulse shape and parameters are displayed on the GUI. The algorithm iterates through the temperature buffer and the power buffer to find the power and temperature maximums and once more through the power buffer to find the full width at half maximum (FWHM).

2.9. Validation of the kinetic equations

Efforts were made to validate the simulator against experiments or other software. Firstly, we validated modelling of point kinetics equations using the digital reactivity metre (DRM) (Lengar et al., 2012), a computer code used at the JSI TRIGA and the NPP Krško that employs inverse kinetics to calculate the reactor reactivity. For validation, the simulator was programmed to start with a critical reactor, insert 100 pcm of reactivity 10 s after the start of the simulation, and return to the initial state 20 s after the start of the simulation. The resulting flux was passed to the digital reactivity metre that used inverse kinetic equations to calculate reactivity. Fig. 2 presents a comparison of the inserted and DRM calculated reactivity for step reactivity change. In stationary conditions, the difference is lower than 1 pcm, which is an expected result because the digital reactivity meter reverses the

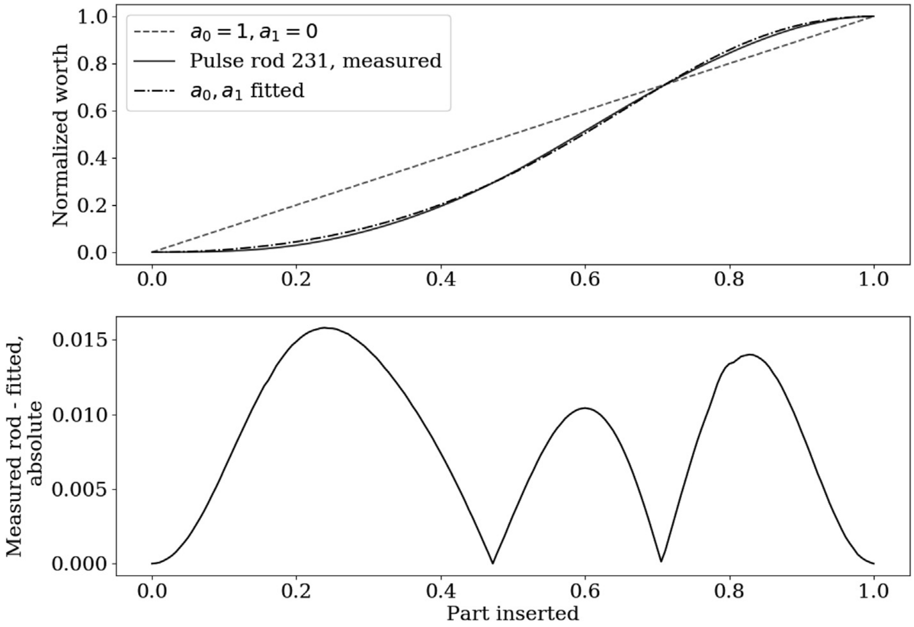


Fig. 1. The top graph shows the worth of two control rod models based on Bezier curves (from Eq. (22)) and compares them with the worth of a real control rod. The bottom graph shows the absolute difference of the normalised control worth of a model based on a Bezier curve and a real control rod. The fitted parameters obtained using multivariable the minimisation method Broyden–Fletcher–Goldfarb–Shanno have values of $\boldsymbol{a}_0 = 0.57$ and $\boldsymbol{a}_1 = 0.68$. The units on x axes are dimensionless and represent the length of the control rod inserted divided by the total control length.

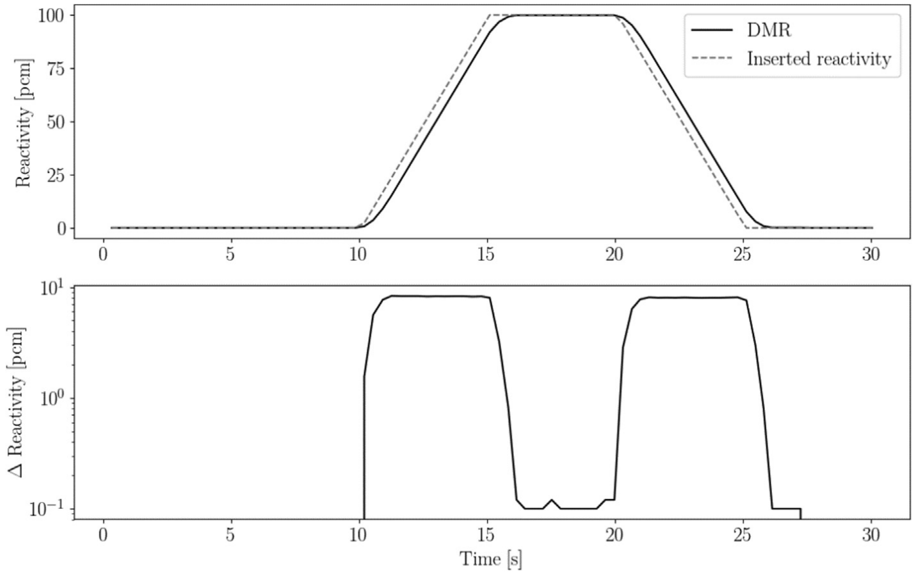


Fig. 2. Comparison of the inserted reactivity with the calculated reactivity using DRM.

equations used to calculate flux. During the transient, the differences are mainly due to different sampling times of the simulator and DRM. The latter has sampling time of 1.43 s.

2.10. Validation of the temperature feedback effects

When the TRIGA Mark II reactor in Ljubljana operates above the point of heating, which is around 1000 W, the response to the changes in reactivity is mainly governed by the temperature feedback effects. In stationary operating conditions above 1000 W, the reactivity inserted by the movement of control rods is cancelled out by the thermal feedback effects, which results in zero net reactivity. Any increase in reactivity results in the rise of reactor power, which raises the fuel temperature and consequently further increases the temperature feedback effects. Due to the finite heat capacity, the fuel elements do not heat up instantly, which causes the reactor power to rise above the stationary temperature. This effect is called 'power overshoot' and is difficult to model precisely because it depends heavily on the thermal parameters, kinetic parameters and control rod calibration. The goal of a simulator used for education is to qualitatively reproduce the power response to a step reactivity change when operating between 1 kW and 250 kW. Fig. 2 displays the response of the JSI TRIGA reactor and the simulator to a step reactivity change of 135 pcm, starting from stationary power conditions of 82 kW. When the dependence of the thermal feedback coefficient with the parameters a, b, c, d and d' in Eq. (13) is set according to (Ravnik, 1991); the shape of the simulated response matches the measured one, but qualitatively, it is narrower and lower. By choosing a constant temperature feedback coefficient of $\alpha = 5.2$, the simulated response matches the experimental data very well.

2.11. Validation of pulse operation

The operation in pulse mode was tested by comparing the maximum values of the reactor power time dependence during a pulse. Fig. 3 presents a comparison of the maximum power reached during a pulse at given reactivity between the simulator and the reactor. The peak powers achieved during the simulated pulses are in a good agreement with the data obtained during measurements between 1998 and 2005, but do not match the measurements performed between 2012 and 2018. No pulse data are available between 2005 and 2012. A known difference between the reactor cores from 2005 and 2012 is that the core in 2012 has one additional fuel element, which slightly increases the heat capacity of the fuel, but this effect does not explain the difference. Analysis of TRIGA pulse experiments was performed in (Pungerčič et al., 2018) and (Vavtar and Snoj, 2018).

3. Implementation

The simulator uses a feedback loop to synchronise the simulation time with the computer system time. The main application loop updates the graphic widgets that display the simulation results and propagates the simulation. Fig. 4 presents a flowchart depicting the operation of the simulation.

3.1. Simulation scheduling

The number of steps needed to propagate the simulation is $t_f \frac{S}{dt}$, where dt is a pre-defined, fixed time step and S is a multiplier that can be used to speed up the simulation. On a personal computer with an Intel i7-7700 K CPU and Nvidia GeForce 1050 Ti graphics card, the time required to draw the simulation results is longer than the time required to propagate the physics simulation by at least two orders of magnitude at normal simulation speed, so the effect of the physics simulation on the graphics refresh rate is negligible. The simulation results are added to a circular buffer that is implemented as a fixed, allocated block of memory in the heap. An index with periodic boundary conditions is used to keep the current position in the buffer.

Each simulation step consists of the following operations.

1. Pulse handling routine: Check to see if the reactor is currently pulsing, and if it is, advance the pulse timer. When the pulse timer expires, scram the reactor and calculate pulse peak power, peak temperature, full width at half maximum (FWHM) and the amount of energy that was released.
2. Recalculate the fuel temperature based on the power from the previous iteration.
3. Move the control rods if the current rod position does not match the set position.
4. Propagate the xenon and iodine concentrations.
5. Recalculate the reactivity.
6. Numerically evaluate the point kinetic equations (1).
7. Propagate the temperature of the water in the fuel tank.
8. Check operational limits and initiate emergency shutdown if those limits have been violated.
9. Advance the simulation timer and push new results to the buffer.

After each simulation step, it is necessary to recalculate all the simulation results that are displayed using the graphic widgets but that are not saved in the buffer. The average reactor period is calculated using a running average of the last 700 reactivity points following Eq. (19). Calculating the average is necessary to avoid quick jumps that cannot be observed at JSI TRIGA instrumentation due to the nature of the analogue metre. Just before the next frame

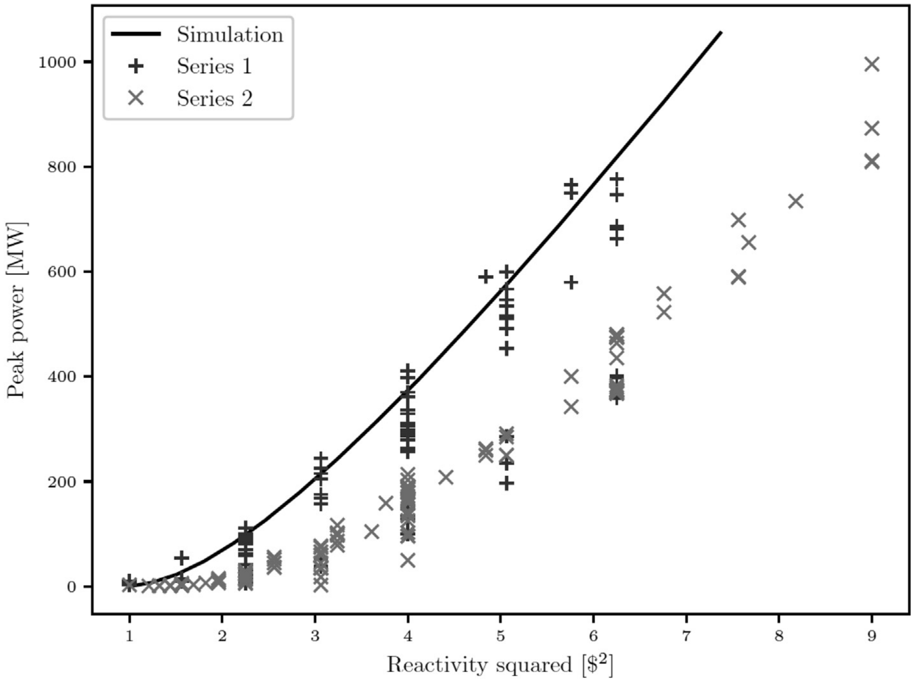


Fig. 3. Comparison of the maximum power achieved during simulated pulses with experimental data. Data from 'Series 1' correspond to measurements performed between 1998 and 2015, whereas data from 'Series 2' correspond to measurements taken between 2012 and 2018. Further analysis is available in (Vavtar et al., 2019) and (Vavtar and Snoj, 2019).

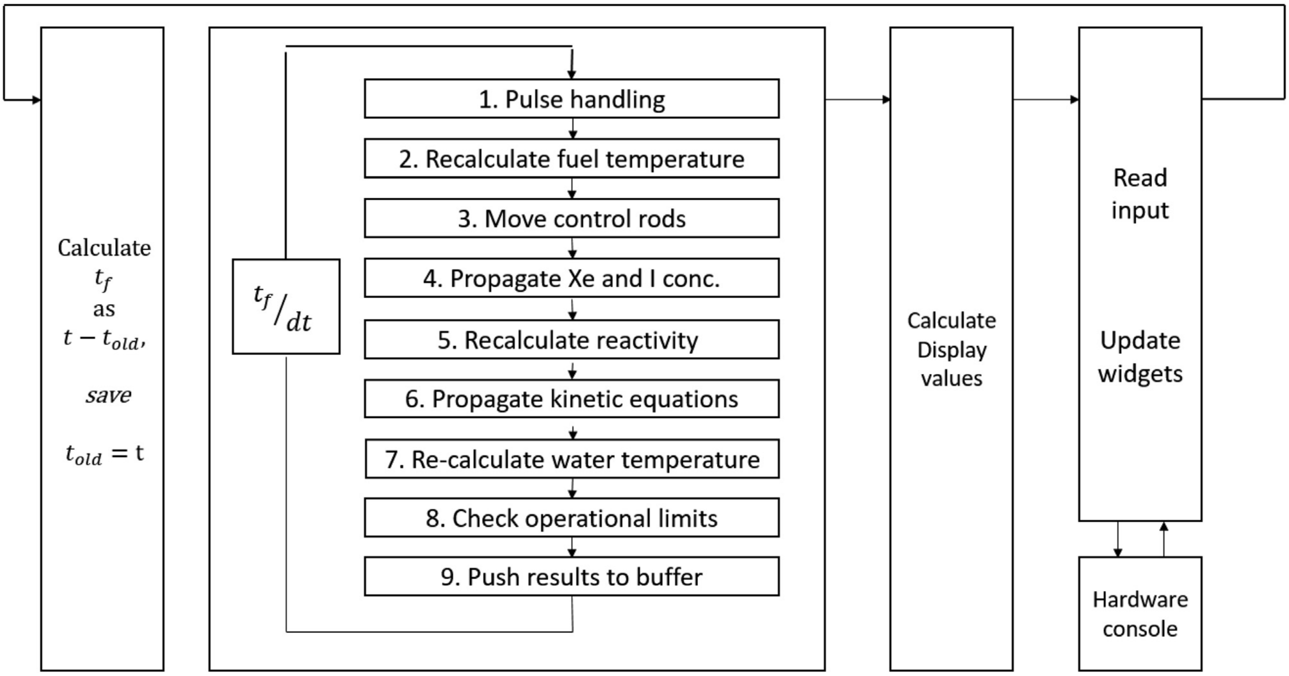

Calculate
t_f
as
$t - t_{old}$,
save
$t_{old} = $ t
t_f/dt
1. Pulse handling
2. Recalculate fuel temperature
3. Move control rods
4. Propagate Xe and I conc.
5. Recalculate reactivity
6. Propagate kinetic equations
7. Re-calculate water temperature
8. Check operational limits
9. Push results to buffer
Calculate Display values
Read input
Update widgets
Hardware console

Fig. 4. Flowchart depicting operation of the simulator. The current time is marked with t, the time expired from the previous frame is marked with t_f and the time step used to solve the kinetic equations is dt.

is drawn, an algorithm iterates through the elements and updates an array with a new maximum power value if such a value has been found and pushes any maximum values that correspond to simulation results that are no longer in memory out of the array. The array of maximum values is used to scale the reactor power plot.

3.2. Software architecture

The research reactor simulator was designed to give the operator a feeling of operating a real research reactor. One requirement was developing a graphic user interface (GUI) that can display information using smooth scrolling graphs and react to a user's input quickly. Quantitatively, the simulator is required to read a user's input and propagate the physics simulation in a time that is faster than typical screen draw times. This is possible by using a GUI library that allows the use of graphics acceleration and the low-level control needed to tweak performance. Research reactor simulator graphic elements are rendered using NanoVG (NanoVG source repository, 2019), a lightweight vector graphics rendering library for OpenGL. The graphic widgets were borrowed from NanoGUI (Nanogui source code, 2019), a cross platform library that uses Graphics Library Framework (GLFW) (GLFW source repository, 2019) to create OpenGL context and event handling, GLAD to allow the usage of OpenGL on Windows, Eigen for linear algebra and NanoVG for 2D vector graphics. Following the coding patterns from the NanoGUI library, the research reactor simulator is written almost entirely in C++, which allows a suitable combination of modern programming practices and low-level control needed for optimisation. The research reactor simulator can be built with the CMake build system on Windows and Linux computers.

3.3. GUI design principles

Development and validation of the reactor physics model is mandatory for a simulator that will be used for education, but the simulator will never achieve its goals without a user interface that is both useful and attractive. The target user population for the research reactor simulator are students of physics, nuclear engineering and related fields, professionals working in the nuclear field and enthusiasts who want to learn about reactor physics. The goal is to allow these users to control a virtual research reactor with minimal help from the manual. To achieve this, the following design principles are addressed.

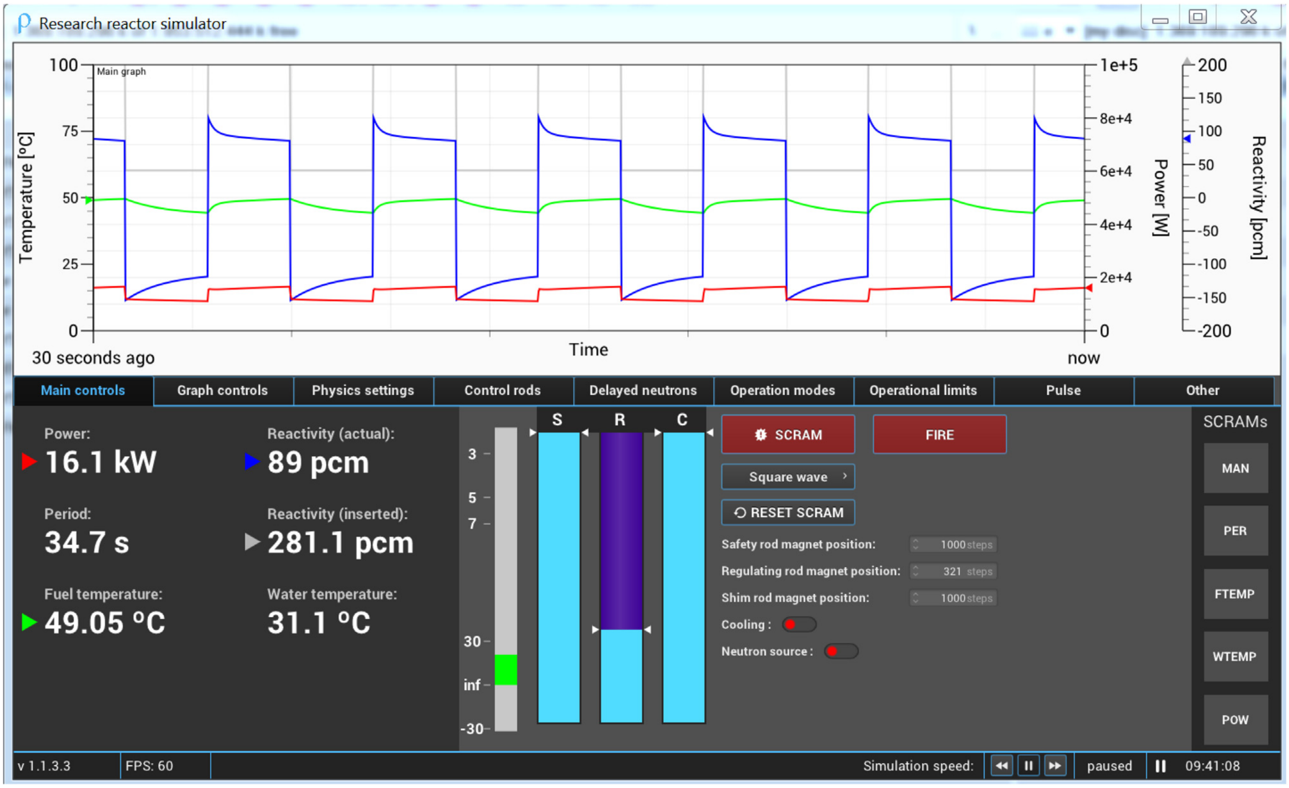
3.3.1. Availability of information

The most important information about the simulator is most visible. In the case of the research reactor simulator, the user is expected to follow a scrolling graph that displays the reactor power in watts with a red line, the fuel temperature with a green line, the reactivity with a blue line and the inserted reactivity with a grey line. The inserted reactivity, which does not consider any feedback effects, is provided to visualise effects of thermal feedback and xenon poisoning. Fig. 5 presents a screenshot of the research reactor simulator. The graph with reactor power, fuel temperature and water temperature is displayed on the top portion of the screen and is always visible. The bottom part of the simulator is tabbed and reserved for reactor controls and settings, which are also sorted by importance. Apart from the tabbed control interface, the users will notice the lack of multi-layer menus in the simulator.

All the controls on the screen are self-descriptive or labelled. If the simulator is controlled with the computer keyboard, the users need to consult the manual to learn about keyboard shortcuts. Otherwise, the simulator can be operated without a manual.

3.3.2. Natural mapping

According to (Norman, 2002), natural mapping can be used to apply physical analogies and cultural standards to form design elements in a way that will be immediately understandable to the users with minimal additional explanation. Applying natural mapping to the design of scientific interfaces is not trivial because many of the quantities do not have obvious analogies to real-world



Power: 16.1 kW

Reactivity (actual): 89 pcm

Period: 34.7 s

Reactivity (inserted): 281.1 pcm

Fuel temperature: 49.05 °C

Water temperature: 31.1 °C

Safety rod magnet position: 1000 steps

Regulating rod magnet position: 321 steps

Shim rod magnet position: 1000 steps

Cooling :

Neutron source :

v 1.1.3.3 FPS: 60 Simulation speed: paused 09:41:08

Fig. 5. The simulator GUI displaying the simulator in the square wave mode. The main controls tab gives information about the basic parameters required to safely operate a reactor (power, period, fuel temperature, water temperature, reactivity, inserted reactivity and rod positions) and allows the user to manipulate the control rods.

situations. The research reactor simulator uses natural mapping for the controls that are commonly used. Two examples are the display of the reactor period and the control rod display. The reactor period is displayed using a white bar graph overlaid by a green graph that marks the current period. Research reactor operators will immediately recognise such a graphic element because it mimics a gauge display used by some research reactors like JSI TRIGA. Users who have never operated a research reactor are also expected to intuitively understand the reactor period control, since the green bar extends upwards when the power is going up and downwards when the power is going down. The control rod display is built from three labelled bar graphs that represent control rods channels. When the control rods are moved, the bottom part of the channel is coloured light blue, which means that the control rod has been withdrawn. This control is designed with natural mapping in that the widget animation represents the control rod movement. Another example of natural mapping that is derived from cultural standards is the scram button, which is big and red. Similarly, the layout of the main graph displaying the reactor power, fuel temperature and reactivity uses a similar layout to the Yokogawa (Yokogawa Electric Corporation, 2019) recorders commonly used in nuclear facilities.

In specialised software like the research reactor simulator, it is not practical to ensure natural mapping of all the controls to the software. Some controls, e.g. the kinetic parameters, have a lot of settings that do not have obvious analogies in everyday life or have too many parameters to make useful widgets with natural mapping.

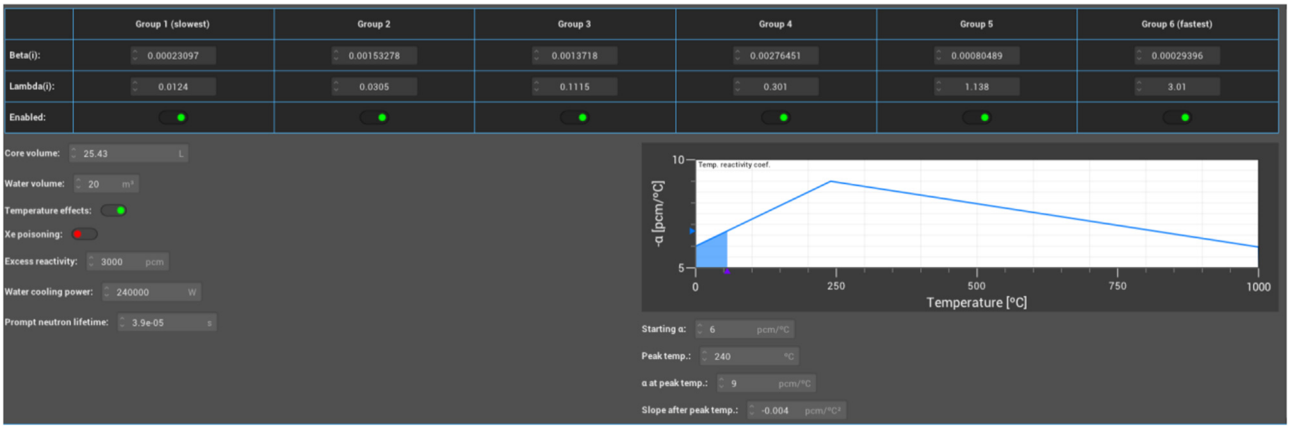
3.3.3. Constraints

Constraints are powerful tools that simplify usage of the software and prevent new users from running into trouble. On the other hand, if too many constraints are applied, functionality and flexibility are automatically reduced. Therefore, a compromise between simplicity and flexibility needs to be reached with any type of software. The main research reactor simulator does not limit a user's input except for basic controls like rod positions. This means that the simulator is flexible, but an inexperienced user can operate it without risk of running in a non-operational or non-physical state as long as the physics settings have not been tweaked. Instead of providing more filters to the user's input, the simulator provides a convenient way of resetting itself.

3.4. GUI controls and displays

All the information needed to 'safely' operate the virtual research reactor is displayed on the main panel. These include displays for numerical values of the quantities displayed on the main graph and information about the reactor period that allows the operator to move the rods, change operating modes, stop the reactor, enable active cooling and operate the neutron source. The graph controls allow the user to change the plot ranges, resize the graph and switch between the linear and logarithmic graphs for reactor power. An option to hide the reactivity plot is available to allow students to perform an approach to criticality exercise by drawing the $1/M$ diagram (Duderstadt and Hamilton, 1976).

The physics tab provides a table that allows the users to edit kinetic parameters and individual controls for the core volume, water volume, temperature effects, xenon poisoning, excess reactivity, power of water cooling, prompt neutron lifetime and parameters for negative fuel temperature feedback on reactivity defined by Eq. (13) (Fig. 6). The physics settings tab provides a sub-panel for configuring the neutron source using a widget that displays the current waveform and allows the user to edit the waveform parameters. In the editing panel in Fig. 7, the reactor operator defined the neutron source waveform as a square wave with a 50% cycle.



	Group 1 (slowest)	Group 2	Group 3	Group 4	Group 5	Group 6 (fastest)
Beta(i):	0.00023097	0.00153278	0.0013718	0.00276451	0.00080489	0.00029396
Lambda(i):	0.0124	0.0305	0.1115	0.301	1.138	3.01
Enabled:	●	●	●	●	●	●

Core volume: 25.43 L

Water volume: 20 m³

Temperature effects: ●

Xe poisoning: ●

Excess reactivity: 3000 pcm

Water cooling power: 240000 W

Prompt neutron lifetime: 3.9e-05 s

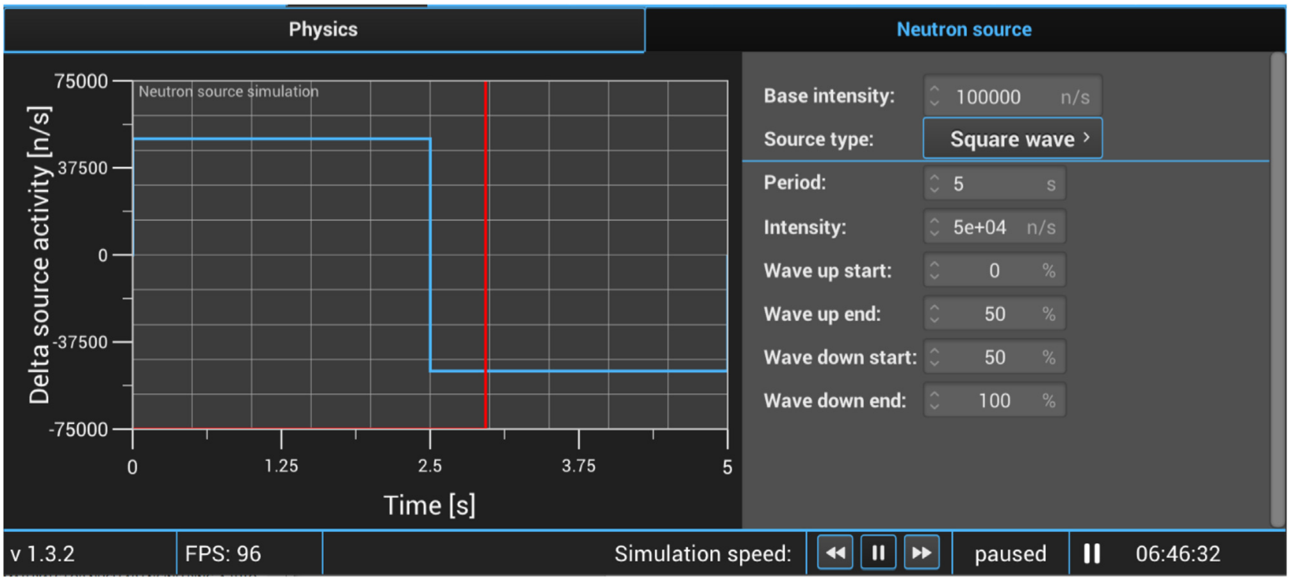
Starting α: 6 pcm/°C

Peak temp.: 240 °C

α at peak temp.: 9 pcm/°C

Slope after peak temp.: -0.004 pcm/°C²

Fig. 6. Screenshot of the physics settings tab. The temperature dependence of the temperature feedback coefficient is displayed with a graph.



Base intensity: 100000 n/s

Source type: Square wave ›

Period: 5 s

Intensity: 5e+04 n/s

Wave up start: 0 %

Wave up end: 50 %

Wave down start: 50 %

Wave down end: 100 %

v 1.3.2 FPS: 96 Simulation speed: ◀◀ ❚❚ ▶▶ paused ❚❚ 06:46:32

Fig. 7. Waveform editing interface for the neutron source.

The control rods panel hosts three widgets displaying the integral reactivity worth curves and three widgets displaying the differential curves corresponding to the three control rods. Fig. 8 presents a screenshot displaying the integral control rod curve with text input fields for the total number of steps the control rod can travel, total control rod worth, movement speed and two sliders that define parameters a_0 and a_1 from Eq. (22). When the sliders are moved, the curves change shape dynamically. The blue area under the curve is the reactivity integral from zero to the rod insertion point.

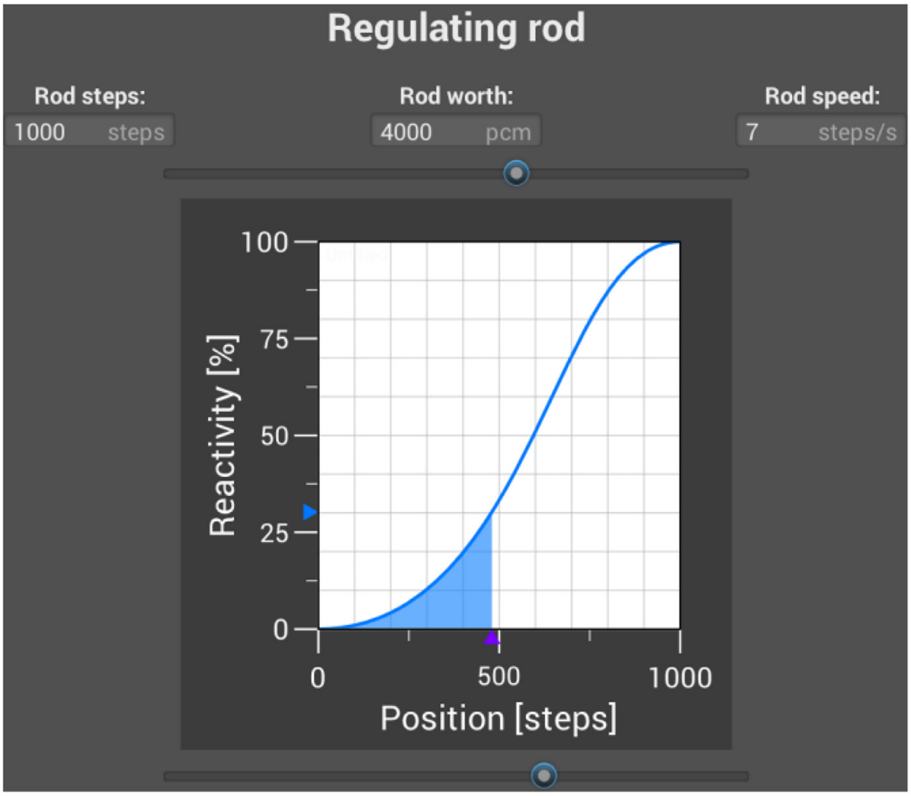
The operation modes tab serves as a waveform editor for the automatic modes and shares a similar interface as the neutron source in Fig. 7, apart from the additional tab reserved for the automatic mode. The operational limits tab allows the users to change the conditions that trigger the automatic fast shutdown (SCRAM). Concentrations of the delayed group are displayed in the delayed neutrons tabbed panel in Fig. 9. The scrolling graph shares the horizontal axes with the main graph displaying reactor power, fuel temperature and the reactivity, but the vertical axes are normalised to 1 and go from 0% to 300% for each of the groups.

Fig. 10 presents a screenshot of the pulse analysis tab 7 after a pulse with peak power of 211.1 MW was simulated. The slider below the simulation results can be used to control the horizontal axis of the graph. The 'other' tab includes an option to save the simulator settings to a binary file and load them from a file, export the operating history to a text file, export the control rod calibration curves to a text file or load a script of commands to run predefined scenarios.

All settings can be changed on the fly without affecting the stability of the system. The simulation time can be adjusted during runtime from 0.001x to 250x real-time speed without lowering the graphic refresh rate. The simulator offers an option to save and load user-defined settings to save the entire buffer of displayed values to a file and export rod curves. The operation limits are adjustable by the user with the help of a graphic interface (see Fig. 11).

3.5. Script interface

Using the script interface, the users can supply pre-defined commands to the simulator. The main purpose of the script interface is to run automated tests on new versions of the simulator, but the scripting interface has uses outside of the development environment. The commands are supplied to the simulator in a text file.



Regulating rod

Rod steps: 1000 steps

Rod worth: 4000 pcm

Rod speed: 7 steps/s

Fig. 8. A custom widget used to define the control rod parameters. The upper slider controls the parameter $\boldsymbol{a}_0$, and the lower slider controls.$\boldsymbol{a}_1$.

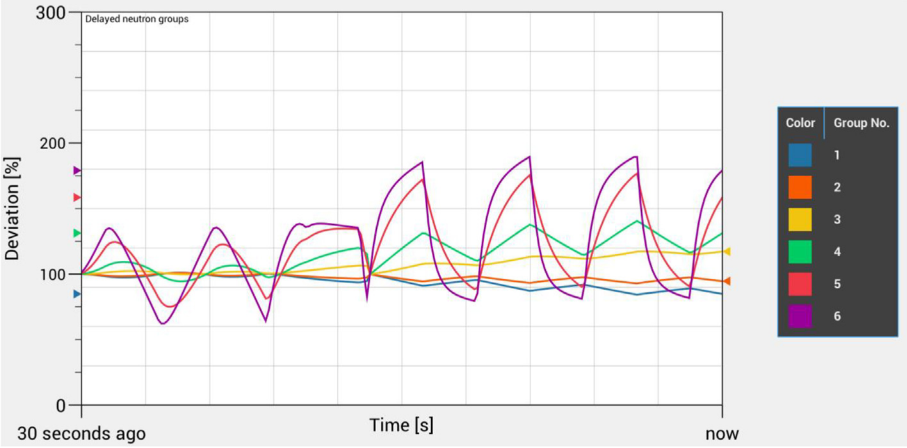


Fig. 9. Display of the six groups of delayed neutrons during a switch from the saw tooth automatic mode to the square wave mode. The displayed quantity is $\boldsymbol{C_i(t)/C_{i,stationary}(t)}$,where $\boldsymbol{C_{i,stationary}(t)}$ is calculated according to Eq. (5).

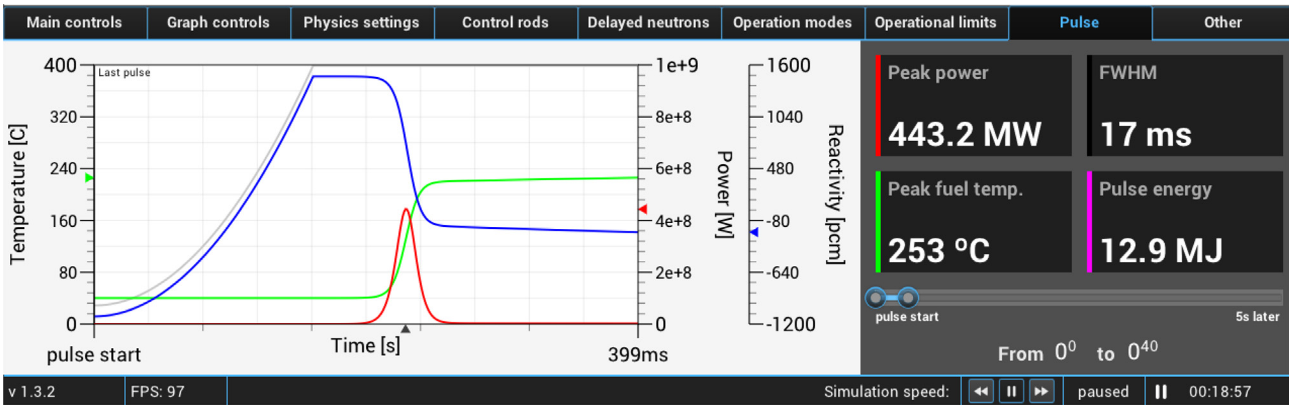
Each line of the script requires a timestamp in seconds, a command and a value separated by a space. Some commands do not require that a value be entered, in which case, the value field will be ignored. A typical use case for the script interface is the pulse benchmark in Fig. 3, which was created by dynamically generating input scripts with the following template.

```
0.1 setStablePower 1.0
0.2 setSimulationSpeed 100
0.2 setSimulationMode Pulse
0.2 setDataLogDivider 1
0.3 setRegulatingSteps<no_steps>
0.4 firePulse 1
3 saveToFile E:\Projekti\Comparison_reactor_DRM\pulseOut
3.1 exitSimulator 0.
```

The simulator manual includes the full list of the supported commands (Toškan et al., 2017).

3.6. Simulator hardware console

A hardware console with buttons (mode selector and control rod drive buttons) has been developed to allow for a more authentic experience of operating a research reactor. Fig. 12 shows a hardware console with a case machined at the JSI workshop. The button layout was inspired by the control panel of the JSI TRIGA



Peak power 443.2 MW

FWHM 17 ms

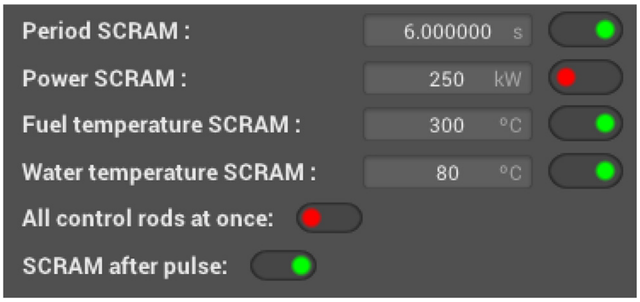
Peak fuel temp. 253 °C

Pulse energy 12.9 MJ

From 0^0 to 0^{40}

v 1.3.2 FPS: 97 Simulation speed: paused 00:18:57

Fig. 10. A screenshot of the pulse analysis tab after a pulse with the peak power of 443.2 MW.



Period SCRAM : 6.000000 s ●

Power SCRAM : 250 kW ●

Fuel temperature SCRAM : 300 °C ●

Water temperature SCRAM : 80 °C ●

All control rods at once: ●

SCRAM after pulse: ●

Fig. 11. Controls used to adjust the operation limits.

Mark II reactor in Podgorica. The console uses USB to connect to the simulator and acts as a streaming device that receives commands from and sends commands to a PC running the research reactor simulator.

4. Future plans

In the near future, we plan to include the simulation of water boiling and evaporation from the tank, which would lead to increased dose rates above the reactor.

Currently, the spatial effects of neutron flux redistribution due to control rod movement are not considered. In the future, we plan to couple two-dimensional deterministic and later even three-dimensional neutron transport calculations with the RR simulator

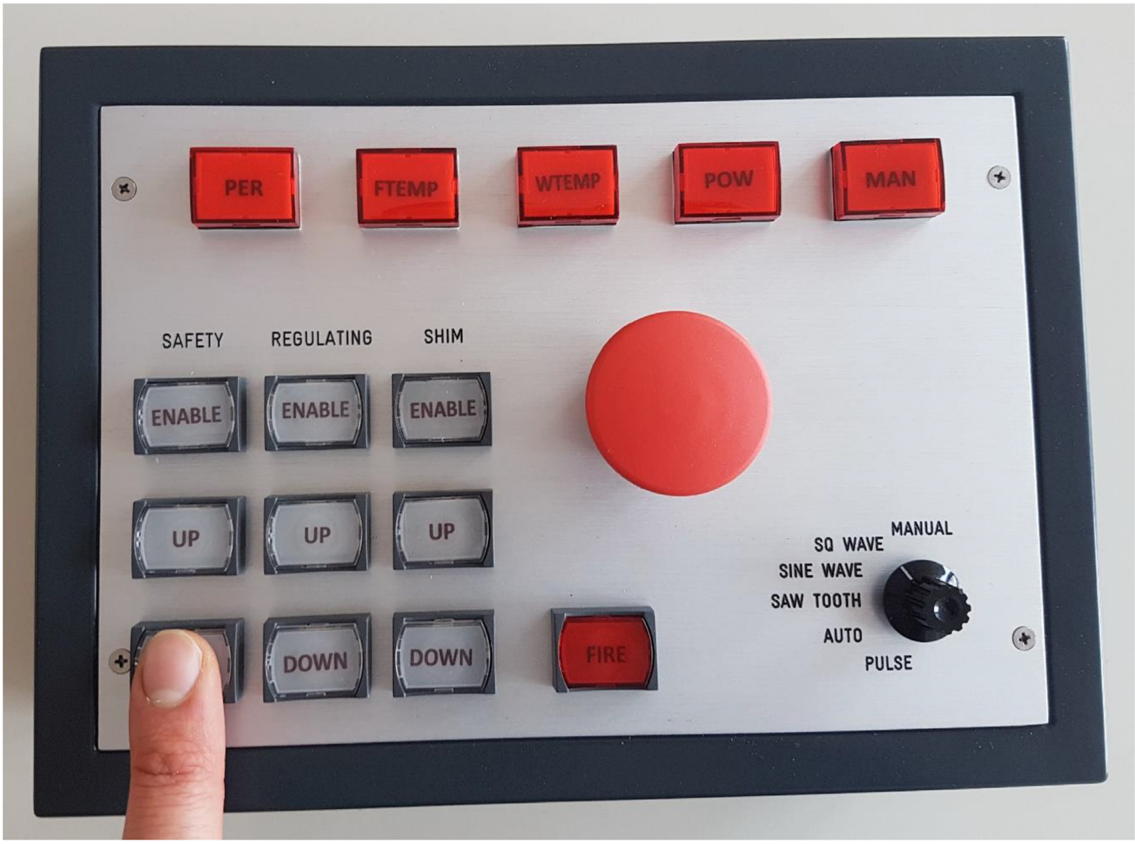


Fig. 12. A hardware control panel for the Research Reactor Simulator.

Table 1
Default values for physical quantities used in the Research Reactor Simulator.

Quantity	Default value
Nominal power	250 kW
Core volume	25.43 L
Water volume	20 m^3
Excess reactivity	3000 pcm
Water cooling power	240000 W
Control rod worth	4000 pcm per rod
SCRAM period	6 s
Fuel temperature scarm	300 °C
Water temperature scarm	80 °C
Neutron source intensity	1000n/s
Delayed neutron fractions	β_1 = 0.00023097, β_2 = 0.00153278, β_3 = 0.0013718, β_4 = 0.00276451, β_5 = 0.00080489, β_6 = 0.00029396
Group decay times	λ_1 = 0.0124, λ_2 = 0.03041, λ_3 = 0.1115, λ_4 = 0.301, λ_5 = 0.138, λ_6 = 3.01
Fuel temperature feedback coefficients	$\alpha_T$at 0 °C: 6 pcm/°C, peak temperature: 240 °C, α_T at peak temperature: 9 pcm/°C, slope after peak: −0.004 pcm/°C^2

and to visualise three-dimensional neutron flux and power distributions (Snoj et al., 2011). We are currently considering the feasibility of using a precalculated fission matrix method such as RAPID (Mascolino et al., 2018) to calculate the three-dimensional neutron distribution in real time. After a three-dimensional flux distribution simulation is developed, it will be possible to include burn-up simulations in the code.

5. Conclusions

The RR simulator is an excellent tool for studying reactor physics of research reactors. It can be used as an educational and training (E&T) tool for better preparation before real reactor practical exercises or as an E&T tool instead of a real reactor. Having the possibility of changing so many reactor parameters makes this simulator very versatile. Although it was designed based on the experience on TRIGA reactors, it can be easily applied to other research reactors.

The RR simulator can be used for performing basic reactor physics exercises such as:

- subcritical multiplication, 1/M diagram;
- reactor response to reactivity changes (step, sine, saw tooth);
- reactor response to source variation;
- reactor feedback;
- control rod worth calibration (rod in, rod drop and rod swap method);
- pulse experiment;
- power calibration;
- Xe poisoning; and
- delayed neutron studies.

Since 2016, the RR simulator has undergone 1 year of testing and development and was released for testing in October 2017 (Research Reactor Simulator, 2019). It can be obtained free of charge by applying at the RR simulator webpage http://reactorsimulator.ijs.si/getit.php. The source code is shared on GitHub (https://github.com/ijs-f8/Research-Reactor-Simulator) under GPL3 software licence (gnu, 2020).

CRediT authorship contribution statement

Jan Malec: Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing, Validation. **Dan Toškan:** Software, Methodology. **Luka Snoj:** Resources, Validation, Supervision, Writing - review & editing, Investigation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0073). Special thanks go to the operators of the TRIGA Mark II reactor in Ljubljana who have supported the development by giving feedback on the research reactor simulator and assisted the project with data and experiments.

Appendix A

Appendix B. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.anucene.2020.107630.

References

Böck, H., et al., 2015, Human resource development and nuclear education through research reactors: successful approach to build up the future generation of nuclear professionals, presented at the Proceedings, p. 10 str.

Böck, H., 2014, Human resources development by the Eastern European research reactor initiative (EERRI). In: International Conference on Human Resource Development for Nuclear Power Programmes, Building and Sustaining Capacity, Vienna.

Duderstadt, J.J., Hamilton, L.J., 1976. Nuclear Reactor Analysis. WILEY, New York.

Gilbert, D., McMaster Nuclear Simulator. Hamilton Ontario, Canada: McMaster University.

"GLFW source repository," GLFW source repository. [Online]. Available: https://github.com/glfw/glfw (accessed: 25-Oct-2019).

"gnu.org." [Online]. Available: https://www.gnu.org/licenses/gpl-3.0.html. (accessed: 12-Feb-2020).

Goričanec, T. et al., 2018. Evaluation of neutron flux and fission rate distributions inside the JSI TRIGA Mark II reactor using multiple in-core fission chambers. Ann. Nucl. Energy 111, 407–440. https://doi.org/10.1016/j.anucene.2017.08.017.

Lengar, I., Trkov, A., Kromar, M., Snoj, L., 2012. Digital meter of reactivity for use during zero-power physics tests at the Krško NPP = Uporaba digitalnega merilnika reaktivnosti pri zagonskih testih na ničelni moči v NE Krško. J. Energy Tech. 5, 13–26.

Mascolino, V., Roskoff, N.J., Haghighat, A., 2018. Haghighat, Benchmarking of the RAPID code system using the GBC-32 cask with variable burnups, In: PHYSOR 2018: Reactor Physics paving the way towards more efficient systems, Cancun, Mexico, vol. 2018.

"Microsimtech triga simulator," Microsimtech triga simulator. [Online]. Available: http://microsimtech.com/triga/. (Accessed: 05-Aug-2019).

"Nanogui source code," Nanogui source code. [Online]. Available: https://github.com/wjakob/nanogui (accessed: 01-Sep-2019).

"NanoVG source repository." [Online]. Available: https://github.com/memononen/nanovg (accessed: 01-Sep-2019).

Norman, D.A., 2002. The Design of Everyday Things. Basic Books Inc., New York, NY, USA.

Peršič, A. et al., 2017. TRIGLAV: a program package for TRIGA reactor calculations. Nucl. Eng. Des. 318, 24–34. https://doi.org/10.1016/j.nucengdes.2017.04.010.

Peršič, A., Ravnik, M., Slavič, S., Žagar, T., 2000. TRIGLAV A Program Package for Research Reactor Calculations, IJS-DP-7862, Jožef Stefan Institute.

Pungerčič, A., Vavtar, I., Snoj, L., 2018. Analysis of the JSI TRIGA pulse experiments. In: European Research Reactor Conference, RRFM, p. 10.

Ravnik, M., 1991. Nuclear safety parameters of mixed TRIGA cores. In: Workshop on reactor physics calculations for applications in nuclear technology.

Ravnik, M., Description of TRIGA Reactor." [Online]. Available: http://www.rcp.ijs.si/~ric/description-s.html (accessed: 09-Aug-2019).

"Research Reactor Simulator official website," Research Reactor Simulator official website. [Online]. Available: http://reactorsimulator.ijs.si/ (accessed: 01-Oct-2019).

Romano, P., Harper, S., 2017. Nuclear data processing capabilities in OpenMC. EPJ Web Conf. 146, 06011. https://doi.org/10.1051/epjconf/201714606011.

Saidu, M.H.I., Waziri, M.Y., 2010. Article: a simplified derivation and analysis of fourth order Runge Kutta method. Int. J. Computer Appl. 9 (8), 51–55.

Simnad, M.T., 1981. The U-ZrHx alloy: Its properties and use in TRIGA fuel. Nucl. Eng. Des. 64 (3), 403–422. https://doi.org/10.1016/0029-5493(81)90135-7.

Širca, S., Horvat, M., 2011. Računske metode za fizike, vol. 46. DMFA - založništvo, Ljubljana.

Snoj, L., Ravnik, M., 2008. Power peakings in mixed TRIGA cores, Nuclear Eng. Design, (238), 2473–2479.

Snoj, L., Sklenka, L., Rataj, J., Böck, H., 2012. Eastern Europe research reactor initiative nuclear education and training courses - current activities and future challenges. In: Proceedings PHYSOR 2012, Advances in reactor physics, p. 7 str.

Snoj, L., Rupnik, S., Jazbec, A., New practical exercises at the JSI TRIGA Mark II reactor. In: PHYSOR 2014 International conference, p. 12 str.

Snoj, L., Kromar, M., Žerovnik, G., Ravnik, M., 2011. Advanced methods in teaching reactor physics. Nucl. Eng. Des. 241 (4), 1008–1012. https://doi.org/10.1016/j.nucengdes.2010.02.040.

Štancar, Ž., Barbot, L., Destouches, C., Fourmentel, D., Villard, J.-F., Snoj, L., 2018. Computational validation of the fission rate distribution experimental benchmark at the JSI TRIGA Mark II research reactor using the Monte Carlo method. Ann. Nucl. Energy 112, 94–108. https://doi.org/10.1016/j.anucene.2017.09.039.

Štancar, Ž., Snoj, L., 2017. An improved thermal power calibration method at the TRIGA Mark II research reactor. Nucl. Eng. Des. 325, 78–89. https://doi.org/10.1016/j.nucengdes.2017.10.007.

"Thermophysical Properties of Materials for Nuclear Engineering: A Tutorial and Collection of Data," 28-Feb-2019. [Online]. Available: https://www.iaea.org/publications/7965/thermophysical-properties-of-materials-for-nuclear-engineering-a-tutorial-and-collection-of-data (accessed: 09-Aug-2019).

Toškan, D., Malec, J., Snoj, L., 2017. Research reactor simulator : User manual, vol. 12332. Ljubljana: Inštitut Jožef Stefan.

Vavtar, I., Snoj, L., 2018. Analiza pulznih sistemov na reaktorju TRIGA, presented at the Zbornik 5. konference mladih jedrskih strokovnjakov, pp. 6–7.

Vavtar, I., Snoj, L., 2019. Pulse experiments overview - In publication, TBD.

Vavtar, I., Pungerčič, A., Snoj, L., 2019. Utilisation of JSI TRIGA Pulse Experiments for Testing of Nuclear Instrumentation and Validation of Transient Models. In: Anima 2019.

"Yokogawa Electric Corporation released the μr and VR - PDF." [Online]. Available: https://docplayer.net/43944522-Yokogawa-electric-corporation-released-the-ur-and-vr.html (accessed: 09-Aug-2019).

Žagar, T., Ravnik, M., Peršič, A., 1999. Analysis of TRIGA reactor thermal power calibration method. In: Proceedings, pp. 91–98.

Žerovnik, G., Snoj, L., Trkov, A., Barbot, L., Fourmentel, D., Villard, J.-F., 2014. Measurements of thermal power at the TRIGA Mark II reactor in Ljubljana using multiple detectors. IEEE Trans. Nuclear Sci. 5 (61), 2527–2531.