McMaster University

Engineering Physics 4H04

Special Studies in Engineering Physics

Final Report
G4-STORK Heat Transfer/SLOWPOKE

Author
Guillaume Gauthier

Supervisor
Dr. Adriaan Buijs

April 29th, 2014

Abstract

Reactor transients caused by loss of coolant, or ejection of the control rod are important because they lead to fast and significant variations in nuclear reactor properties which can lead to catastrophic results. That is why they need to be studied and simulated such that a better understanding of their cause and effect can lead to a better mitigation of the damage caused when such scenario occurs. Developing codes able to simulate transients can not only help improve existing reactors, but also allow to foresee weaknesses in future reactors.

G4-STORK was developed to simulate, study, and provide a better understanding of transients in nuclear reactors. It has been bench-marked against mostly theoretical situations/geometries and found to give results that fit the theoretical results within statistical uncertainty. The program must first be able to predict known experimentally measured transient behavior of existing reactors in order to be able to predict transient behaviors of new reactors. That is the why the SLOWPOKE-2 reactor has been chosen to be modeled, there is a well known transient response that occurs when the control rod, of the reactor, is taken out of a previously shut down SLOWPOKE-2 that had its control rod fully inserted. G4-STORK was in need of an upgrade in terms of heat tracking inside the reactor geometry to be able to replicate this response.

This report details the fabrication of the SLOWPOKE-2 reactor in G4-STORK; as well as, the physics and implementation of a heat transfer class to keep track of the energy deposition due to the fission sites inside the reactor and the heat diffusion away from the fission site.

Contents

List of Tables

List of Figures

1 Introduction

During the simulation of transient conditions inside a nuclear reactor, one is particularly interested in the time evolution of the neutron population overtime. G4-STORK does just that, it is a program that uses a Monte-Carlo process to simulate the interaction of a neutron population with its surrounding. It falls short in one aspect and that is that the energy transfer between the neutrons and the different geometries inside the reactor as they undergo nuclear interaction such as fission is not being taken into consideration. It is important to account for energy deposition due to nuclear interaction inside the reactor since during transients the neutron population is usually increasing which generates high amounts of heat inside the reactor which will change the geometry's temperature substantially. Leading to additional feedback during the nuclear reactor transient due to changes in nuclear cross sections with temperature.

This report details the implemented heat transfer class which allows the user to specify a region of the nuclear reactor on which the temperature will be kept track of based on the heat equation and the location of the fission sites inside the geometry. The heat tracking is done by defining a grid over the region in question effectively dividing it into many nodes and applying the heat transfer equation over these nodes.

The goal of this heat transfer algorithm is to permit more accurate reactor transient analysis and to allow for the simulation of transients such as the one that occurs in the SLOWPOKE-2 nuclear reactor when going from having the control rod fully inserted into the reactor core to having the control rod fully pulled out of the reactor. The reason the SLOWPOKE-2 was chosen is because it has a well documented and experimentally measured transient behavior which goes something like this once the control rod has been pulled out. The reactivity of the reactor goes above critical, the power output of the reactor increases; this leads to a temperature of the geometry which leads modifies the cross sections of the geometries leading to decrease of the reactivity of the reactor, and the rate of increase of the power. This process keeps happening until the power increase reaches 0, the reactivity of the reactor is critical, and the temperatures are stable. Without temperature changes inside the geometry; there is no negative feedback to bring the reactor reactivity back down to the critical level after the rod has been pulled out which means that one would obtain that the neutron population is ever increasing, warranting the need for heat deposition tracking. The transients are not yet simulated due to massive computational time that would be needed; although attempts were made at determining the steady state temperature distribution of the reactor with the control rod pulled out and the results are presented in this report, the SLOWPOKE-2 geometry construction is also discussed in details; as well as, changes made to increase the navigation speed through the reactor geometries.

2 Heat Transfer Inside Nuclear Reactor

The standard heat equation that can be derived from Fourier's law assuming a constant thermal diffusivity α. And assuming conservation of energy. The equation is as follows:

$$\frac{\partial T(x,y,z,t)}{\partial t} = \alpha\left(\frac{\partial^2 T(x,y,z,t)}{\partial x^2} + \frac{\partial^2 T(x,y,z,t)}{\partial y^2} + \frac{\partial^2 T(x,y,z,t)}{\partial z^2}\right) = \alpha\vec{\nabla}^2 T(x,y,z,t) \tag{1}$$

2.1 Generalized Heat Equation Without Heat Sources

In this case we are especially interested in the diffusion of heat from one material to another with sharp interfaces. In that case, the thermal diffusivity coefficient $\alpha(x,y,z)$ is not constant through space and this change needs to be accounted for in the heat equation. The derivation of the exact equation for the none constant case is done as follows. First, Fourier's law (Eq. 2) is needed to understand the heat transfer from a macroscopic point of view.

$$\vec{q}(x,y,z,t) = -k(x,y,z,t)\vec{\nabla}T(x,y,z,t) \tag{2}$$

Where $T(x,y,z,t)$ is the temperature at specific location given by coordinates (x,y,z) at time t, k is the thermal conductivity of the material at that same location and time, and $\vec{q}$ is a vector filed describing the flux of heat at that same location and its direction in space. Let us now look at an infinitesimal volume where heat is allowed to flow in and out of the surface. The total flux entering the surface is given by the integral of the inner product of the Flux field with the surface normal over the whole surface ($\mathcal{S}$).

$$U_{surface} = \oiint_{\mathcal{S}} -\vec{q}(x,y,z,t) \cdot d\vec{S} = \oiint_{\mathcal{S}} -\vec{q}(x,y,z,t) \cdot \hat{n} dS \tag{3}$$

Where $\hat{n}$ is the surface normal unit vector and dS is an infinitesimal surface area. $U_{surface}$ represents the heat entering the surface at that given instant in time. One can convert the equation to a temperature transfer through the surface by dividing by c_p specific heat capacity ($\frac{J}{K*kg}$) and the density (ρ) of the material ($\frac{kg}{m^3}$). Eq. 3 becomes:

$$T_{surface} = \oiint_{\mathcal{S}} \frac{-\vec{q}(x,y,z,t)}{c_p(x,y,z,t) \times \rho(x,y,z,t)} \cdot \hat{n} dS \tag{4}$$

Another way to calculate the total change in the amount of heat inside the surface at that instant in time is to take the integral over the volume of the derivative heat inside the surface with time.

$$T_{volume} = \iiint_{\mathcal{V}} \frac{\partial T(x,y,z,t)}{\partial t} dV \tag{5}$$

Where $\mathcal{V}$ represents the volume enclosed by the surface ($\mathcal{S}$). Applying conservation of energy and assuming that none of the heat radiates and that no heat is added to the system (for now), one obtains that the temperature transfers in Eq. 5 and Eq. 4 must be equal.

$$\iiint_{\mathcal{V}} \frac{\partial T(x,y,z,t)}{\partial t} dV = \oiint_{\mathcal{S}} \frac{-\vec{q}(x,y,z,t)}{c_p(x,y,z,t) \times \rho(x,y,z,t)} \cdot \hat{n} dS \tag{6}$$

Now the divergence theorem is needed to make both integrals in terms of the enclosed volume. The theorem is as follows:

$$\iiint_{\mathcal{V}} \left(\vec{\nabla} \cdot \vec{F}\right) dV = \oiint_{\mathcal{S}} \left(\vec{F} \cdot \hat{n}\right) dS \tag{7}$$

This means that Eq. 4 is equal to:

$$\oiint_{\mathcal{S}} \frac{-\vec{q}(x,y,z,t)}{c_p(x,y,z,t) \times \rho(x,y,z,t)} \cdot \hat{n} dS = \iiint_{\mathcal{V}} -\vec{\nabla} \cdot \left[\frac{\vec{q}(x,y,z,t)}{c_p(x,y,z,t) \times \rho(x,y,z,t)}\right] dV \tag{8}$$

Substituting Eq. 8 into Eq. 6.

$$\iiint_{\mathcal{V}} \frac{\partial T(x,y,z,t)}{\partial t} dV = \iiint_{\mathcal{V}} -\vec{\nabla} \cdot \left[\frac{\vec{q}(x,y,z,t)}{c_p(x,y,z,t) \times \rho(x,y,z,t)} \right] dV \tag{9}$$

Substituting Eq. 2 into Eq. 9:

$$\iiint_{\mathcal{V}} \frac{\partial T(x,y,z,t)}{\partial t} dV = \iiint_{\mathcal{V}} \vec{\nabla} \cdot \left[\frac{k(x,y,z,t)\vec{\nabla}T(x,y,z,t)}{c_p(x,y,z,t) \times \rho(x,y,z,t)} \right] dV \tag{10}$$

Eq. 10 needs to stand for any volume, any time (t) and any temperature distribution. For this to be true the integrands must be the equal. This leads to the following more general form of the heat equation:

$$\frac{\partial T(x,y,z,t)}{\partial t} = \vec{\nabla} \cdot \left[\frac{k(x,y,z,t)\vec{\nabla}T(x,y,z,t)}{c_p(x,y,z,t) \times \rho(x,y,z,t)} \right] = \vec{\nabla} \cdot \left[\alpha(x,y,z,t)\vec{\nabla}T(x,y,z,t) \right] \tag{11}$$

Where $\alpha(x,y,z,t) = \frac{k(x,y,z,t)}{c_p(x,y,z,t) \times \rho(x,y,z,t)}$. Using the distributive property of the divergence operator:

$$\vec{\nabla} \cdot (\psi \vec{A}) = \vec{A} \cdot \vec{\nabla}\psi + \psi \vec{\nabla} \cdot \vec{A} \tag{12}$$

Eq. 11 can be simplified as follow:

$$\frac{\partial T(x,y,z,t)}{\partial t} = \vec{\nabla}\alpha(x,y,z,t) \cdot \vec{\nabla}T(x,y,z,t) + \alpha(x,y,z,t)\vec{\nabla}^2 T(x,y,z,t) \tag{13}$$

As a self check one can simply assume the thermal diffusivity (α) is constant in space and see what comes out of Eq. 13 and as expected one obtains the original heat equation in Eq. 1.

2.2 Generalized Heat Equation With Heat Sources

Now the heat equation needs to be derived assuming that there is some heat being added or removed from inside the surface because of some other process than diffusion such a fissions, other some other process. Now to determine the impact of this heat generation term, one must add up the amount of heat that was introduced into the volume or taken out of the volume at time t. This is done by integrating the heat generation term over the whole volume at time t; which gives the following result:

$$U_{Generated} = \iiint_{\mathcal{V}} h(x,y,z,t) dV \tag{14}$$

Where $U_{Generated}$ is the total heat generated in the volume per unit time, $h(x,y,z,t)$ is the heat generated per unit time at location (x,y,z). To transform Eq. 14, into a temperature transfer one must simply divide by the specific heat capacity (c_p) and the density of the material (ρ). This leads to the following equation:

$$T_{Generated} = \iiint_{\mathcal{V}} \frac{h(x,y,z,t)}{\rho(x,y,z,t)c_p(x,y,z,t)} dV \tag{15}$$

To get the new heat equation with added external heat sources, it is sufficient to reapply the energy conservation criterion as follows:

$$T_{Volume} - T_{Generated} = T_{Surface} \tag{16}$$

The temperature gained by the volume minus the energy generated by the heat sources inside the volume must equal the energy gained through the surface flux. Performing substitutions of Eq. 4, 5, and 15 into 16, one obtains the following result.

$$\iiint_{\mathcal{V}} \left[\frac{\partial T(x,y,z,t)}{\partial t} - \frac{h(x,y,z,t)}{\rho(x,y,z,t)c_p(x,y,z,t)}\right] dV = \iiint_{\mathcal{V}} \vec{\nabla} \cdot \left[\alpha(x,y,z,t)\vec{\nabla}T(x,y,z,t)\right] dV \tag{17}$$

Using the previous argument that the equation must be valid for any volume size and at all times, this can only be satisfied if the integrands are equal at every point in space and time as follows:

$$\frac{\partial T(x,y,z,t)}{\partial t} = \vec{\nabla} \cdot \left[\alpha(x,y,z,t)\vec{\nabla}T(x,y,z,t)\right] + \frac{h(x,y,z,t)}{\rho(x,y,z,t)c_p(x,y,z,t)} \tag{18}$$

It is again possible to use the distrubitivity of the divergence operator (Eq. 12) to simplify Eq. 18 as follows:

$$\frac{\partial T(x,y,z,t)}{\partial t} = \vec{\nabla}\alpha(x,y,z,t) \cdot \vec{\nabla}T(x,y,z,t) + \alpha(x,y,z,t)\vec{\nabla}^2 T(x,y,z,t) + \frac{h(x,y,z,t)}{c_V(x,y,z,t)} \tag{19}$$

Where $c_V(x,y,z,t)$ is the heat capacity per unit volume of the material. This is the form of the heat equation that will be used to analyzed the transfer of heat transfer inside the nuclear reactor.

3 Finite Element Approximation to Heat Equation

Eq. 19 is the heat equation to be used to keep track of the heat transfer inside the nuclear reactor, but the exact heat generation formula $h(x,y,z,t)$ is not known and cannot be known exactly due to its statistical nature and neither is $T(x,y,z,t)$. Even if they were know there would be no analytically formula to solve equation; therefore, one must resort to numerical formulas such as finite elements, or finite differences.

3.1 Finite Difference Approximation of the Heat Equation

To approximate the heat equation in Eq. 19, finite difference will be used. To approximate first order derivative in space, the central difference will be used and goes as follows:

$$\frac{\partial f(x,y,z,t)}{\partial x} = \frac{f(x+h,y,z,t) - f(x-h,y,z,t)}{2h} + \mathcal{O}(h^2) \tag{20}$$

In the case of second order differential equation in space the 2^{nd} order central difference will be used which goes as follows:

$$\frac{\partial^2 f(x,y,z,t)}{\partial x^2} = \frac{f(x+h,y,z,t) - 2f(x,y,z,t) + f(x-h,y,z,t)}{h^2} + \mathcal{O}(h^2) \tag{21}$$

And finally, first order time derivatives are approximated using a forward difference:

$$\frac{\partial f(x,y,z,t)}{\partial t} = \frac{f(x,y,z,t+\Delta t) - f(x,y,z,t)}{\Delta t} + \mathcal{O}(\Delta t) \tag{22}$$

Eq. 19 can be equivalently expressed as follows:

$$\frac{\partial T}{\partial t} = \left(\hat{x}\frac{\partial}{\partial x} + \hat{y}\frac{\partial}{\partial y} + \hat{z}\frac{\partial}{\partial z}\right)\alpha \cdot \left(\hat{x}\frac{\partial}{\partial x} + \hat{y}\frac{\partial}{\partial y} + \hat{z}\frac{\partial}{\partial z}\right)T + \alpha\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right)T + \frac{h}{c_V} \tag{23}$$

Eq, 23 can be further simplified by completing the dot product to obtain the following.

$$\frac{\partial T}{\partial t} = \left(\frac{\partial \alpha}{\partial x}\frac{\partial T}{\partial x} + \frac{\partial \alpha}{\partial y}\frac{\partial T}{\partial y} + \frac{\partial \alpha}{\partial z}\frac{\partial T}{\partial z}\right) + \alpha\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}\right) + \frac{h}{c_V} \tag{24}$$

3.2 Cross Term Approximation

Using Eq. 20 , 21 and Eq. 22 the previous differential can be transformed into its finite difference approximation. First, let's start with the $\frac{\partial \alpha}{\partial x}\frac{\partial T}{\partial x}$ term.

$$\frac{\partial \alpha}{\partial x}\frac{\partial T}{\partial x} = \left(\frac{\alpha(x+h_x,y,z,t) - \alpha(x-h_x,y,z,t)}{2h_x}\right)\left(\frac{T(x+h_x,y,z,t) - T(x-h_x,y,z,t)}{2h_x}\right) + \mathcal{O}(h_x) \tag{25}$$

This process can be repeated with the y and the z-direction with finite difference steps of size h_y and h_z, respectively, to obtain the following results.

$$\frac{\partial \alpha}{\partial y}\frac{\partial T}{\partial y} = \left(\frac{\alpha(x,y+h_y,z,t) - \alpha(x,y-h_y,z,t)}{2h_y}\right)\left(\frac{T(x,y+h_y,z,t) - T(x,y-h_y,z,t)}{2h_y}\right) + \mathcal{O}(h_y) \tag{26}$$

$$\frac{\partial \alpha}{\partial z}\frac{\partial T}{\partial z} = \left(\frac{\alpha(x,y,z+h_z,t) - \alpha(x,y,z-h_z,t)}{2h_z}\right)\left(\frac{T(x,y,z+h_z,t) - T(x,y,z-h_z,t)}{2h_z}\right) + \mathcal{O}(h_z) \tag{27}$$

3.3 Laplacian Approximation

Let us now take a look at the second term of Eq. 24, $\alpha\frac{\partial^2 T}{\partial x^2}$ it can be approximated using finite difference as follows.

$$\alpha\frac{\partial^2 T}{\partial x^2} = \alpha(x,y,z,t)\frac{T(x+h_x,y,z,t) - 2T(x,y,z,t) + T(x-h_x,y,z,t)}{h_x^2} + \mathcal{O}(h_x^2) \tag{28}$$

The same process can be applied in the y and z directions with space grid discretization h_y and h_z, respectively, to obtain the following.

$$\alpha\frac{\partial^2 T}{\partial y^2} = \alpha(x,y,z,t)\frac{T(x,y+h_y,z,t) - 2T(x,y,z,t) + T(x,y-h_y,z,t)}{h_y^2} + \mathcal{O}(h_y^2) \tag{29}$$

$$\alpha\frac{\partial^2 T}{\partial z^2} = \alpha(x,y,z,t)\frac{T(x,y,z+h_z,t) - 2T(x,y,z,t) + T(x,y,z-h_z,t)}{h_z^2} + \mathcal{O}(h_z^2) \tag{30}$$

3.4 Approximating the Heat Generation Term

When simulating nuclear reactor through Monte Carlo, one has access to a subset of the total distribution of neutrons in reactor; in the sense that for each neutron in the simulation there are usually more than a billion times more neutrons in a real nuclear reactor. The hope is that the subset of neutrons is big enough to given a good approximation of the behavior of real neutron system. Assuming this approximation to be valid, it is possible to approximate the amount of heat added in the reactor at a certain instant in time. This is done by first using the real thermal power generation of the nuclear reactor and by keeping track of the location and time of the fissions in the nuclear reactor. It is a simple matter to convert the reactor power to reactor energy by multiplying it by the time of the interval over which the fission were kept track of and the energy can then be divided evenly among the fissions which equates to assuming that the only interaction where energy is transferred is during fission events. To express the process mathematically:

$$h(\vec{r},t) \approx \sum_{i=0}^{N-1} \frac{P_{Reactor}(t)}{N}\delta(\vec{r}-\vec{r}_i, t-t_i) \tag{31}$$

Where N is the total number of fissions that occurred between time ($t_{initial}$) and the final time in the interval being considered ($t_{final} = t_{initial} + \Delta t$), $\vec{r}_i$ is the physical position of the i^{th} fission site, and t_i is the time of the i^{th} fission site. Eq. 31 only approximates the heat generated between time interval $t_{initial}$ and t_{final}. Note the reactor power only needs to be known at one instant in time since we are keeping track of keff which is the number of neutron from new generation produced by neutron from previous generation, we can expect the reactor power to grow proportionally with the neutron population as follows:

$$P_{Reactor}(t+\Delta t) = \int_t^{t+\Delta t} k_{eff}(t)P_{Reactor}(t)dt \approx k_{eff}(t)P_{Reactor}(t) \tag{32}$$

To obtain the final term in equation, it suffices to divide Eq. 31 by $c_v(x,y,z,t)$.

$$\frac{h(\vec{r},t)}{c_v(x,y,z,t)} \approx \frac{1}{c_v(x,y,z,t)}\sum_{i=0}^{N-1} \frac{P_{Reactor}(t)}{N}\delta(\vec{r}-\vec{r}_i, t-t_i) \tag{33}$$

3.5 Full Finite Difference Heat Equation

Finally one can solve for the heat at a time step greater the t, by approximating the time derivative, in Eq. 24 , using forward difference and isolating for $T(x,y,z,t+\Delta t)$. Once Eq. 25, 26, 27, 28, 29, 30, 33 have been substituted into Eq. 24, the final heat equation is obtained and is as follows.

$$
\begin{aligned}
&T(x,y,z,t+\Delta T) \approx T(x,y,z,t)+\\
&\Delta t\left(\frac{\alpha(x+h_x,y,z,t)-\alpha(x-h_x,y,z,t)}{2h_x}\right)\left(\frac{T(x+h_x,y,z,t)-T(x-h_x,y,z,t)}{2h_x}\right)+\\
&\Delta t\left(\frac{\alpha(x,y+h_y,z,t)-\alpha(x,y-h_y,z,t)}{2h_y}\right)\left(\frac{T(x,y+h_y,z,t)-T(x,y-h_y,z,t)}{2h_y}\right)+\\
&\Delta t\left(\frac{\alpha(x,y,z+h_z,t)-\alpha(x,y,z-h_z,t)}{2h_z}\right)\left(\frac{T(x,y,z+h_z,t)-T(x,y,z-h_z,t)}{2h_z}\right)+\\
&\Delta t\alpha(x,y,z,t)\frac{T(x+h_x,y,z,t)-2T(x,y,z,t)+T(x-h_x,y,z,t)}{h_x^2}+\\
&\Delta t\alpha(x,y,z,t)\frac{T(x,y+h_y,z,t)-2T(x,y,z,t)+T(x,y-h_y,z,t)}{h_y^2}+\\
&\Delta t\alpha(x,y,z,t)\frac{T(x,y,z+h_z,t)-2T(x,y,z,t)+T(x,y,z-h_z,t)}{h_z^2}+\\
&\frac{\Delta t}{c_v(x,y,z,t)}\sum_{i=0}^{N-1}\frac{P_{Reactor}(t)}{N}\delta(\vec{r}-\vec{r}_i,t-t_i)+\\
&\mathcal{O}(\Delta t\left[h_x+h_y+h_z+h_x^2+h_y^2+h_z^2\right]+\Delta t^2)
\end{aligned}
\tag{34}
$$

As can be seen in one calculation of a cross term involves two subtractions, one multiplication and one division. This means that there is a potential to save a lot of time if one is to take time to check whether the neighboring points are the same material or not. Also, another interesting result is that the present algorithm is order $h_i\Delta t$ ($i = \{x, y, z\}$) when sitting at a point where there is transition between two materials and Δt^2 in terms of time. Whereas, it goes as $h_i^2\Delta t$ when the temperature is projected at a point where it is surrounded by the same material, the time error does not change and stays Δt^2.

3.6 Updating Material Temperature

The way Geant4 handles temperature is that a certain material pointer is assigned to a certain logical volume and a temperature is assigned to each material. This means that every logical volume consists of only one temperature, but the grid split all the logical volume into many squares each with their respective temperature which are updated based on the heat equation. Now it might happen that some of the logical volume lies outside the specified grid in which case the temperature calculated for that specific material inside the grid is not representative of it actual temperature. This means we need a way to determine if the grid temperature is representative of the temperature of the material everywhere inside the reactor. This can be achieved by comparing the actual heat capacity associated with the different materials inside the reactor and the one that can be calculated as seen from inside the specified grid. The heat capacity of the materials inside the specified heat transfer region is calculated using the following:

$$C_{Grid} = h_x h_y h_z \times c_p \times \rho \times N \tag{35}$$

Where in N is the number of regions inside the grid associated with the material we are trying to calculate the heat capacity of. To calculated the actual heat capacity associated with a given material can be calculated directly from the volume of the logical volumes inside the world as follows:

$$C_{actual} = c_p \times \rho \times V \tag{36}$$

To determine whether the grid is a good representation of the actual temperature of the material after heat transfer; the actual and grid calculated heat capacities are compared and the selection criterion to determine whether they are close enough to one another is as follows:

$$\frac{|C_{Grid} - C_{Actual}|}{C_{Actual}} \leq 10\% || C_{Actual} \leq C_{Grid} \tag{37}$$

If one of the two previous criterion is meet, the temperature on the grid is considered a good indicative of the material temperature and is calculated as follows:

$$T_{Mat} = \frac{1}{N}\sum_{i=1}^{N} T_i \tag{38}$$

In the case where the criterion is not meet the following is used to determine the temperature of the material after the heat has been transferred:

$$T_{Mat} = \frac{C_{Grid}\left[\frac{1}{N}\sum_{i=1}^{N} T_i - T_{Before}\right] + C_{Actual}T_{Before}}{C_{Actual}} = T_{Before} + \frac{C_{Grid}}{C_{Actual}}\left[\frac{1}{N}\sum_{i=1}^{N} T_i - T_{Before}\right] \tag{39}$$

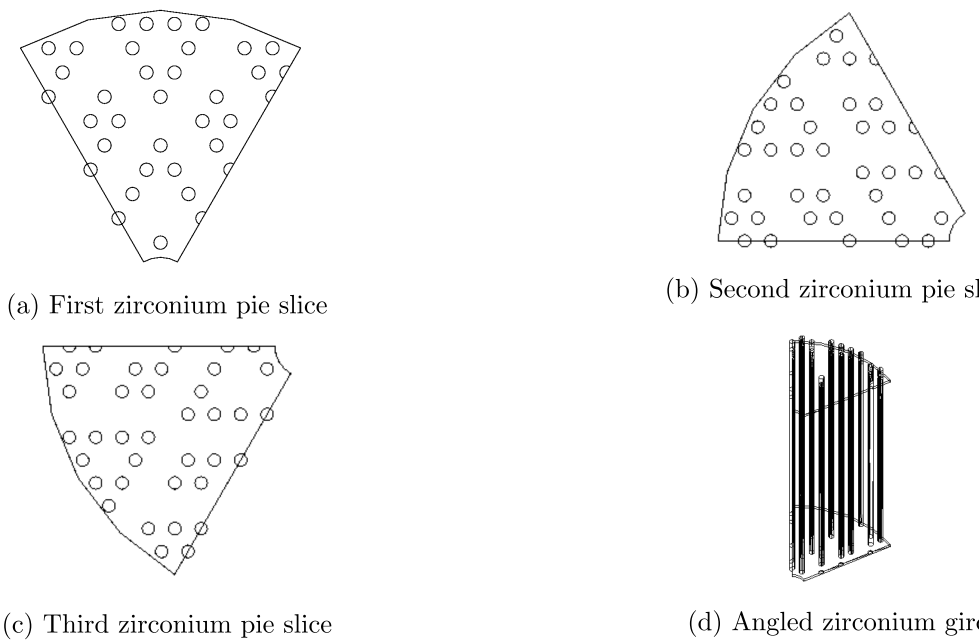
4 SLOWPOKE-2 Reactor

The way the SLOWPOKE-2 reactor was built is from the inside out and the geometries were taken from J.R.M. Pierres thesis [1]. First, the zirconium fuel assembly was built and this was done by splitting it into 6 pie slices due to the symmetry of the reactor, the reasons behind this will be discussed. Then the air gaps and the fuel rods were added inside the zirconium fuel rods, this was followed by the control rod being added and finally the beryllium reflector as well as some irradiation rods and a D_2O container (which is an added feature of the SLOWPOKE-2 at Polytechnique de Montreal). Finally, the reactor is put inside a poll of water enclosed by an aluminum tube. The geometries used to designing the SLOWPOKE-2 reactor were mostly derived from the MCNP code presented by J. R. M. Pierre in his thesis [1].

4.1 Geometries

4.1.1 Zirconium Grid

The SLOWPOKE-2 reactor core geometry was split into 6 pie slices. So as to make the computational load smaller. The speed boost comes from the fact that the neutrons spend most of their time inside the fuel rods and if all of the water circulation holes and zirconium rods were made to be one big volume; the navigation manager, to determine which of the fuel rod the particle is sitting inside of, would need to go through and make sure the neutron is not sitting inside any of the many subtracted geometries before confirming that it is actually inside the mother volume (Zirconium Grid) and then moving on to determine which one of the ≈216 fuel rods the neutron is actually inside of. In the case where the zirconium grid is split into 6 pie slices the work needed to pinpoint the location of the neutron inside the reactor is cut down by almost a factor of 6. The image of a zirconium plate with the added zirconium rods for the fuel is shown in Figure 1.


(a) First zirconium pie slice

(b) Second zirconium pie slice

(c) Third zirconium pie slice

(d) Angled zirconium gird

Figure 1: Zirconium grid pieces

These three geometries are combined and then mirrored to form the full zirconium assembly which is shown in Figure 2.

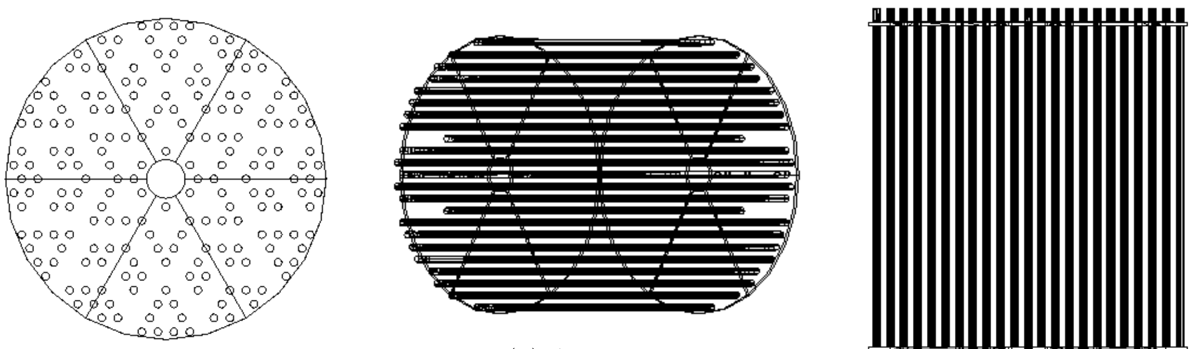
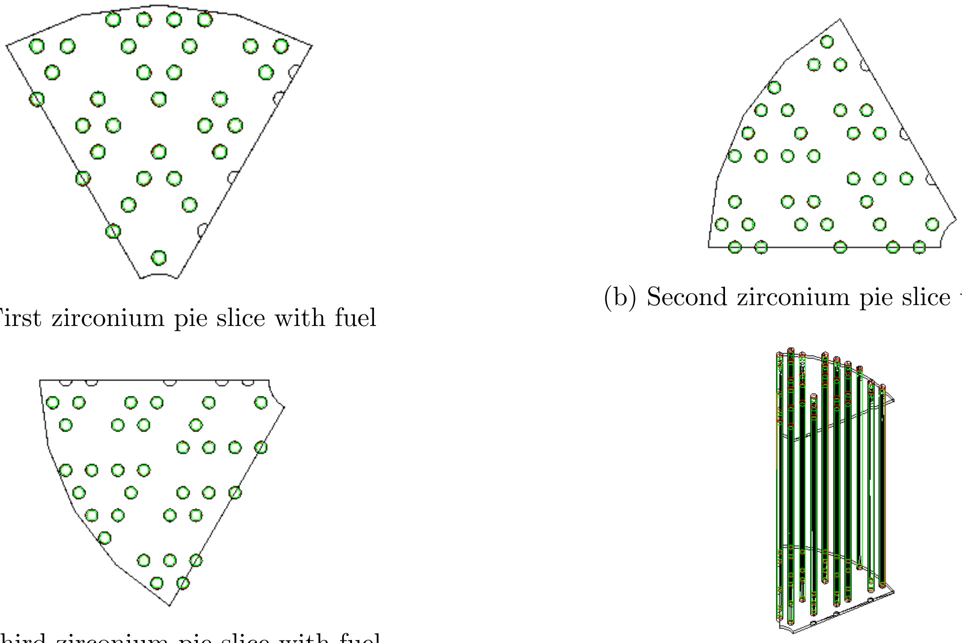

(a) Bottom View (b) Angle View (c) Side View

Figure 2: Zirconium grid final after mirroring

4.1.2 Fuel Rods

Before the creation of the grid, fuel rods with their respective air gaps are added to the pie slice pieces as in Figure 3. Once they are combined the following core assembly is finished as shown in Figure 4, note: Red = Air, Green = Fuel.


(a) First zirconium pie slice with fuel (b) Second zirconium pie slice with fuel

(c) Third zirconium pie slice with fuel (d) Angled zirconium pie slice with fuel

Figure 3: Zirconium grid pieces with added fuel

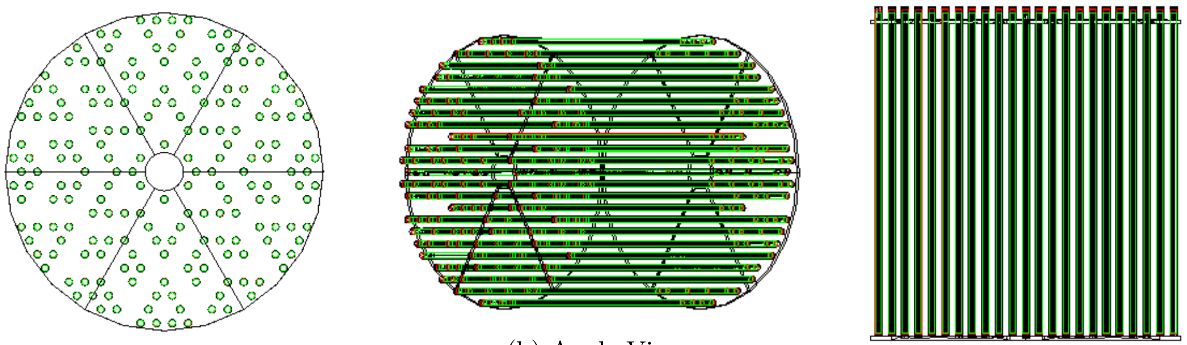

(a) Bottom View (b) Angle View (c) Side View

Figure 4: Final fuel core assembly

The zirconium grid with water circulation holes is depicted in the Figure 5.

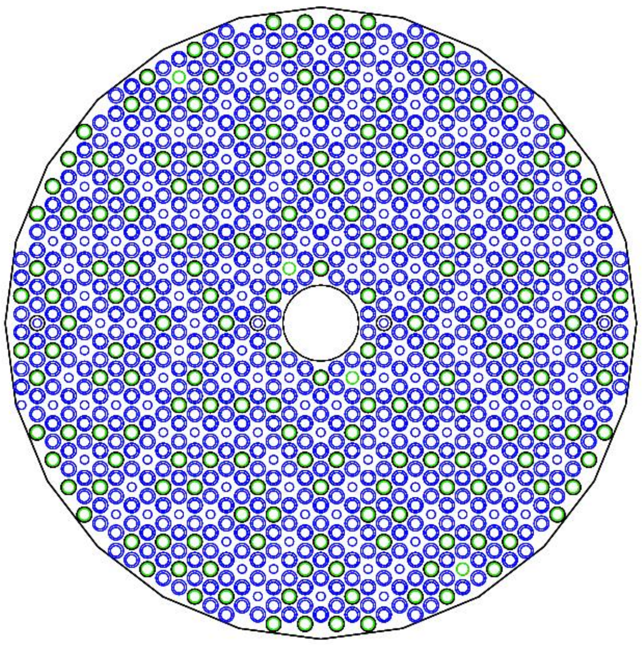


Figure 5: Final fuel core assembly with water circulation holes

4.1.3 Reflector

The next geometry to be added is the beryllium shield with one added compensating plate. It simply consists of a lower reflecting beryllium plate, a reflecting ring inside which the fuel can be added. Finally, some compensating beryllium plates can be added on top to increase the reactivity as the fuel burns up, but in this case only one is put one since the composition of our fuel is that of fresh fuel. This is shown in Figure 6 where the beryllium is depicted in brown.

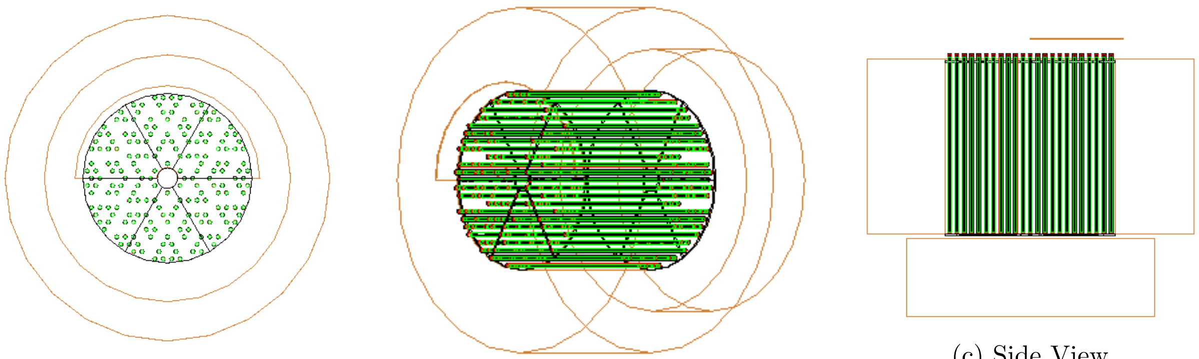

(a) Bottom View (b) Angle View (c) Side View

Figure 6: Fuel core assembly with added beryllium shield

4.1.4 Control Rod

The next piece of the reactor is the control rod. It is made up of a cadmium rod inside an aluminum hold which all sites inside a zirconium rods as depicted in Figure 7. It is simply used to control the reactivity of the reactor by bringing it closer, or moving it further away from the core center.

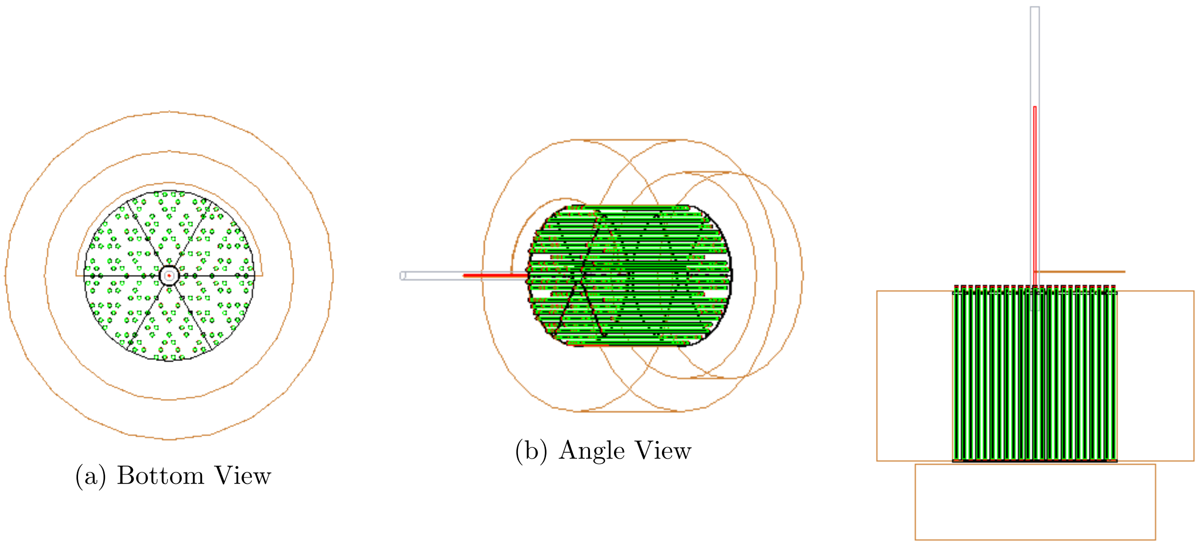
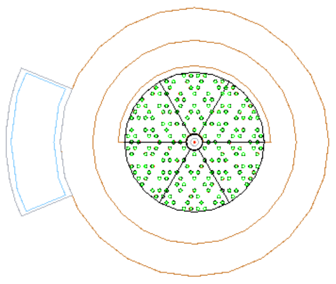
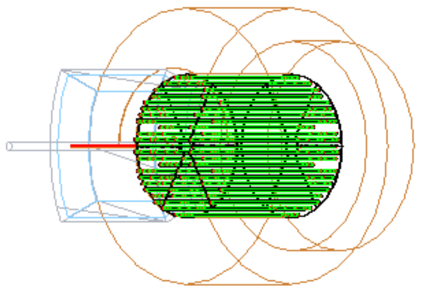

(a) Bottom View (b) Angle View (c) Side View

Figure 7: Fuel core assembly with added beryllium shield and control rod

4.1.5 Heavy Water Container

This part is unique to the SLOWPOKE reactor at the Polytechnique de Montreal. It consists simply of a of an aluminum container which is filled with heavy water as shown in Figure 8. The container spans an angle of $\frac{\pi}{3}$, which means it spans only $1/6^{th}$ of the circle.


(a) Bottom View


(b) Angle View

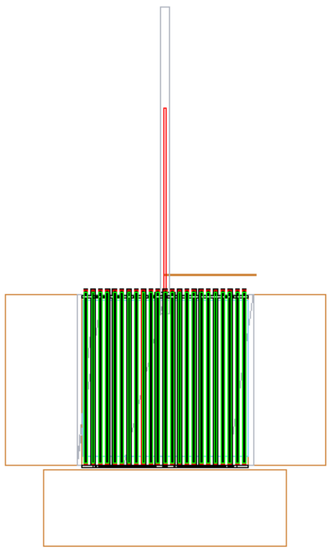
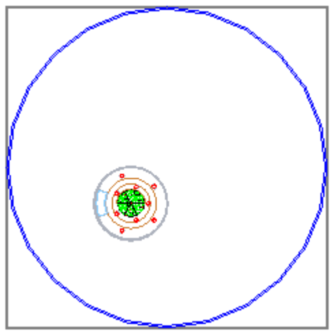
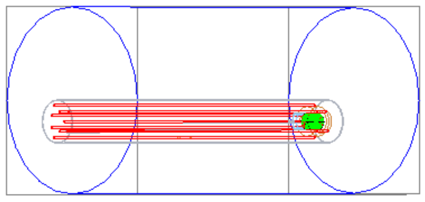

(c) Side View

Figure 8: Fuel core assembly with added beryllium shield, control rod, and heavy water container

4.1.6 Final Reactor

To finalize the reactor, irradiation tubes are added into the beryllium shield, and outside the shield. Finally, the water is added and the pool to hold the water is added to obtain the final SLOWPOKE reactor as shown in Figure 9.


(a) Bottom View


(b) Angle View

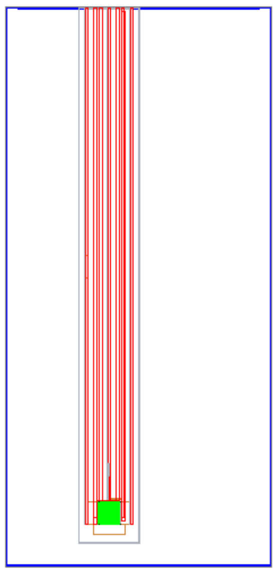

(c) Side View

Figure 9: SLOWPOKE2 Reactor

4.2 Materials

Table 1 shows all the materials that were used to fabricate the present SLOWPOKE reactor.

Table 1: List of materials used to fabricate the reactor

Material Name	Density $(g \times cm^{-3})$	Number of elements	Specific heat capacity $(\frac{J}{g \times K})$	Thermal conductivity $(\frac{W}{m \times s})$	Temperature K
Galactic	10^{-25}	1	0.0	0.0	-271
Light Water	0.998	2	4.1813	0.5984	30.6
Heavy Water	1.105	2	4.224211211	0.589	20.5
Zirconium	6.49	1	0.278	8.625	52.14
Aluminum Alloy	2.70	5	0.897	205	20-40
Cadmium	8.65	1	0.231	92	50.0
Fuel	10.6	2	0.2411519506	21.5	57.32
Air	5.0807E-5	2	1.0035	0.024	18.0
Reflector	1.85	17	1.83	218	22.5

4.2.1 Element breakdown of materials

The materials can be broken down into their respective elements as follows:

Galactic simple filler material for the world volume which contains the reactor geometry.

Light Water is composed of 2 ^1_1H combined with an oxygen atom which can be any of the naturally occurring isotopes assigned according to their natural abundance.

Heavy Water is composed of 2 ^2_1H combined with an oxygen atom which can be any of the naturally occurring isotopes assigned according to their natural abundance.

Zirconium is simply made of naturally occurring zirconium.

Aluminum Alloy is composed by weight of naturally occurring Aluminum at 97.92%, Silicon at 0.60%, Copper at 0.28%, Magnesium at 1.0%, and Chromium at 0.20%.

Cadmium is simply naturally occurring Cadmium.

Air is broken down into Oxygen at 21.174% and Nitrogen at 78.826%.

Reflector is, by weight fraction, mostly made up of naturally occurring Beryllium at 99.53863%, Oxygen at 9.70113×10^{-4}%, and the following trace elements Aluminum at 0.1010534%, Iron at 0.131695%, Silicon at 0.06063207%, Cadmium at 7.376901×10^{-5}%. There are also some trace single isotopes which are: Manganese at 0.01515802%, $^{10}_4B$ with mass fraction of 4.345298×10^{-7}, $^{11}_4B$ at 1.616855×10^{-6}, ^6_3Li at 1.313695×10^{-7}, ^7_3Li at 1.92001^{-6}, $^{149}_{62}Sm$ at 6.497736^{-7}, $^{155}_{64}Gd$ at 3.53687×10^{-8}, $^{157}_{64}Gd$ at 3.132657×10^{-8}, $^{151}_{63}Eu$ at 2.425283×10^{-7}, and $^{153}_{63}Eu$ at 2.627389×10^{-7}.

Fuel is made of 2 atoms of naturally occurring oxygen atoms and one atom of low enriched uranium (19.89%) which can either be $^{235}_{92}U$ with a 19.89% probability, or $^{238}_{92}U$ with a 80.11% probability.

5 Simulation Results

5.1 Testing SLOWPOKE geometries

After optimizing the SLOWPOKE-2 geometry such that the run times were decent enough that real simulations could be performed. It is needed to test the actual set-up to determine if everything is as one would expect it to be in this case, one should obtain that the reactor is critical when the control rod is fully pulled out and below critical when the control rod is fully inserted. The results obtained from these tests are shown in Table 2. These results also permit to see how fast the standard deviation of the reactivity of the reactor decreases. So as to be able to make predictions as to how long a run should be to obtain a certain uncertainty on the output reactivity.

5.1.1 Reactivity Results

Table 2: Results obtained for the different runs (ran using 25 processors on the computer cluster)

Control Rod Position		**In**	**Out**
Results	Mean keff (k)	0.9908	1.0068
	Standard Deviation (k)	0.00146	0.00136
	Standard Deviation of Mean (k)	0.000318	0.000351
	Total Simulation Time (hours)	32.7	24.6
Inputs	Number of Primaries per Event	3000	3000
	Number of Events per Run	400	400
	Run Duration (ms)	0.5	0.5
	Number of runs simulated after convergence	20	25
	Instantaneous Delays	Yes	Yes
	Total Number of Primaries Simulated	24000000	18000000

The reactivity worth of the control rod can be obtained by taking the difference of the reactivity of the reactor with the rod fully inserted and with the rod fully pulled out.

$$\text{Control Rod Simulated Reactivity Worth} = (15.045 \pm 0.669)\ \text{mk}$$

From experiments, it is known that the reactivity worth of the control rod is 5.45mk [2]. The theoretical result is about 14.3 standard deviations away from the actual answer.

There is clearly something wrong with the theoretical simulation and from the standard deviation, it is not the number of samples that is too low, but we are somehow simulating a different situation that the actual one. The first thing that came to mind was that the geometry was wrong, but after rechecking with J.R.M. Pierres thesis , I am fairly certain that the two models are the same. With same geometry the MCNP model obtained a control worth reactivity worth of (7.85 ± 0.33) mk which is not the experimental reactivity worth, but is closer.

Another discarded error source are the cross section libraries which could be less accurate, but thats very unlikely since J.R.M. Pierres simulations [1] were ran in 1996 and there has been a lot of refinement of cross section data since then.

Not all of our results are negative, the first one being that the reactor is below critical when the control rod is fully inserted and above critical when the control rod is fully taken out as is expected for the reactor to function properly.

5.1.2 Standard Deviation Versus Simulation Time

The standard deviation of the mean of a measurement decrease as the number of measurement taken increases according to $\sigma_N = \frac{\sigma}{\sqrt{N}}$ where N is the number of samples. Using the data from Table 1, one can plot the standard deviation of the result of a run as a function of the number of primaries simulated. This can be converted to calculation time using the total computation time in Table 4. The result is shown in Figure 10.

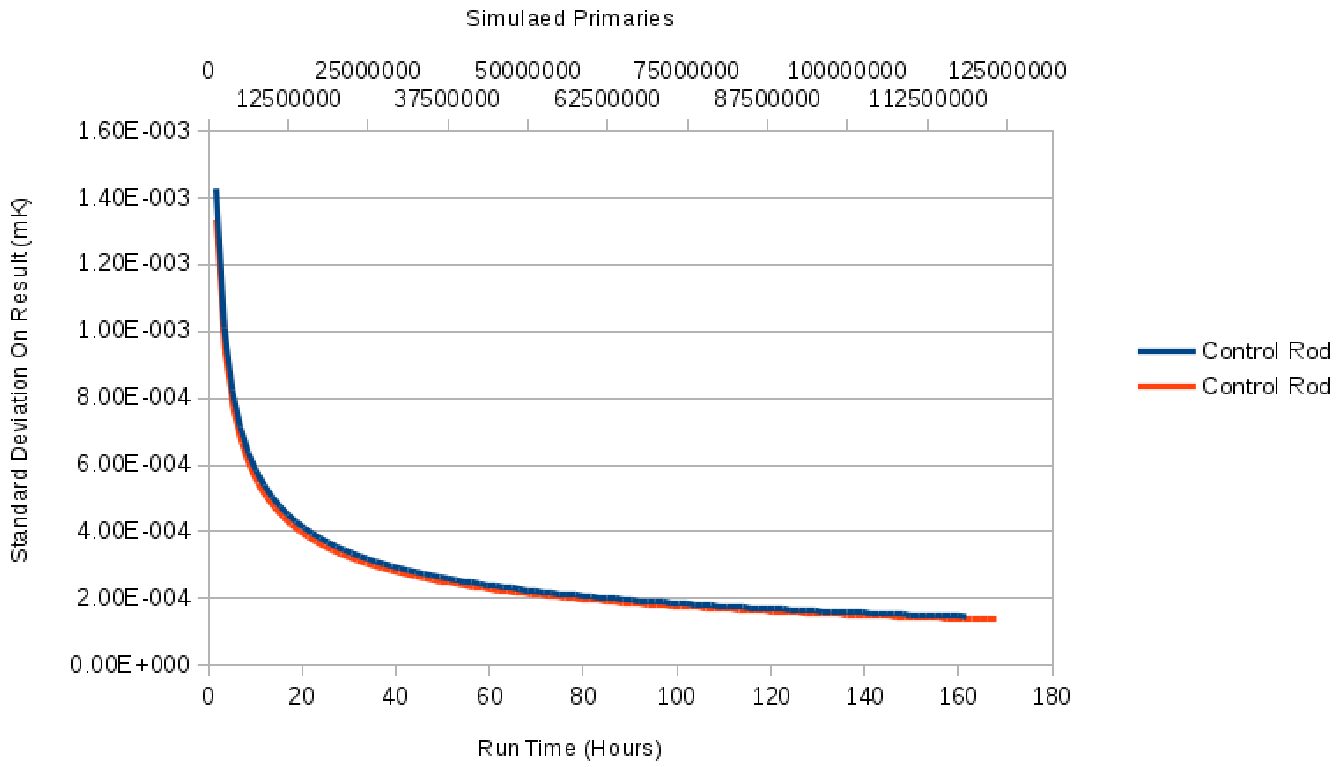


Figure 10: Standard deviation of the mean versus simulation time estimated with both results set

From this we can see that to get uncertainties under 1mk the simulation time per run would need to be higher than 3.43 hours and to be under 0.5mk more than 13.7 hours.

This means that either very little runs need to be made, or the process needs to be speed up if the program is to be run for time periods in the range of minutes and not milliseconds.

5.1.3 Is Further Improvement On the Union Class Needed?

The following simulations were performed to determine if optimizing the union volume further would significantly decrease the simulation time of the SLOWPOKE-2 reactor, since this volume type is only used in the core for the water circulation holes:

First, the holes in the zirconium grid were deleted and 1152000 neutrons were simulated. The total computation time was 28196.4 seconds. This means an average of 0.024475 seconds per simulated primary.

Second, with the holes in the zirconium 108000 neutrons were simulated. The total computation time was 2810.8 seconds. This means an average of 0.026025 seconds per simulated primary. An improvement of 6%.

This could reduce the computation down from 3.43 hours to 3.22 hours, but this is in no way where most of the computation time is being spent.

5.1.4 Entropy versus number of runs

The following results show the Shannon entropy of the reactor as a function of the run number.

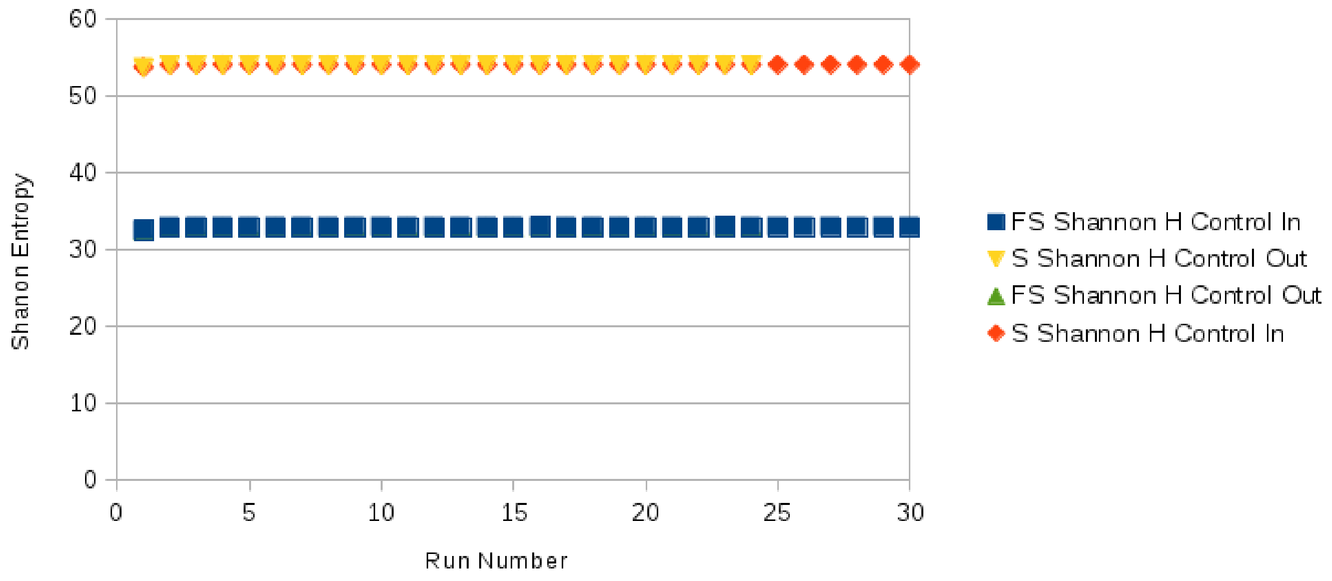


Figure 11: Shannon Entropy as a function of the run number

From Figure 11, it can clearly be seen that the neutron population was already converged to within 1% after only 2 runs.

5.1.5 keff versus run number

Figure 12, shows the k effective as a function of run number which seems to have reached a steady state after less than 5 runs. There is still a clear uncertainty in the calculated reactivity of the SLOWPOKE reactor from one run to another.

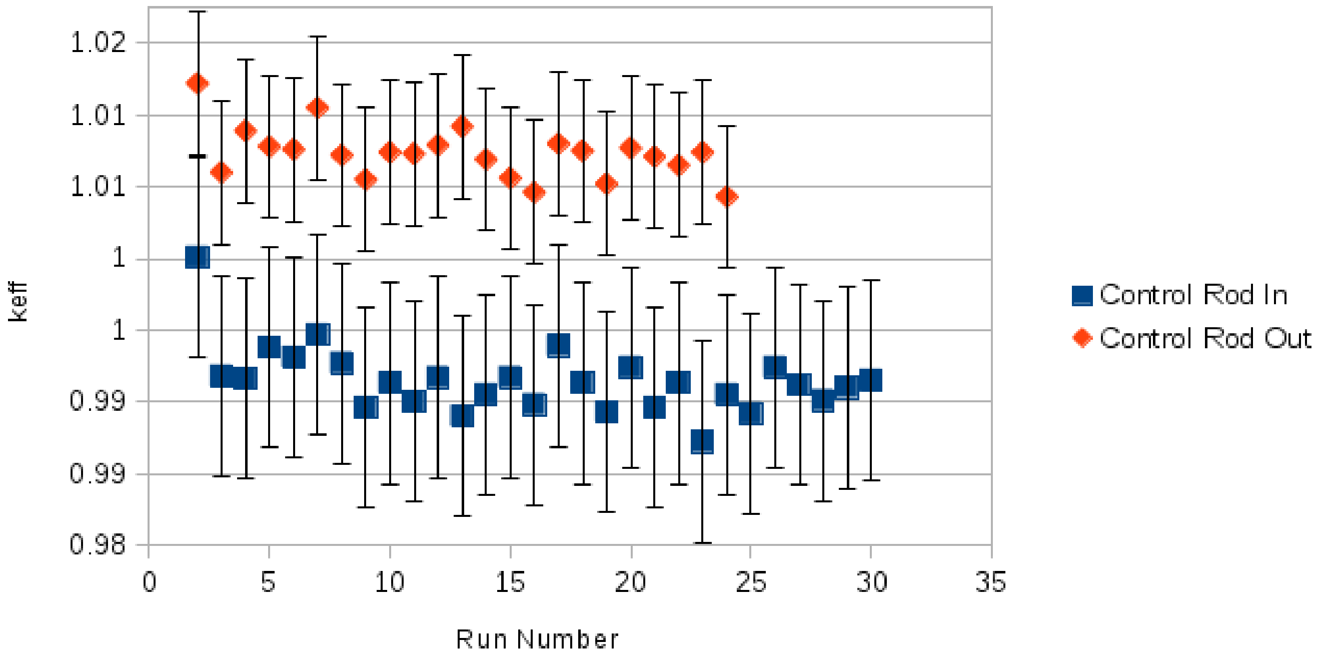


Figure 12: SLOWPOKE-2 reactivity versus run number

5.1.6 Material Temperatures

The temperature of the different geometries inside the reactor during the simulation is shown in Table 3.

Table 3: Material temperatures

Material	Temperature (K)	Temperature (^0X)
Be Reflector	295.65	22.5
Light Water	303.75	30.6
Air Gaps	291.15	18.0
Zirconium Support Gird	325.29	52.14
Aluminum Alloy	295.15	22.0
Cadmium Rod	323.15	50.0
Heavy Water	293.65	20.5
Fuel Rod	330.47	57.32

5.1.7 Neutron Population Distribution

The vertical population distribution inside the reactor is shown in Figure 13. Note that the center of the reactor core is located at around -2300mm. This is where the neutron density is highest which is not unexpected.

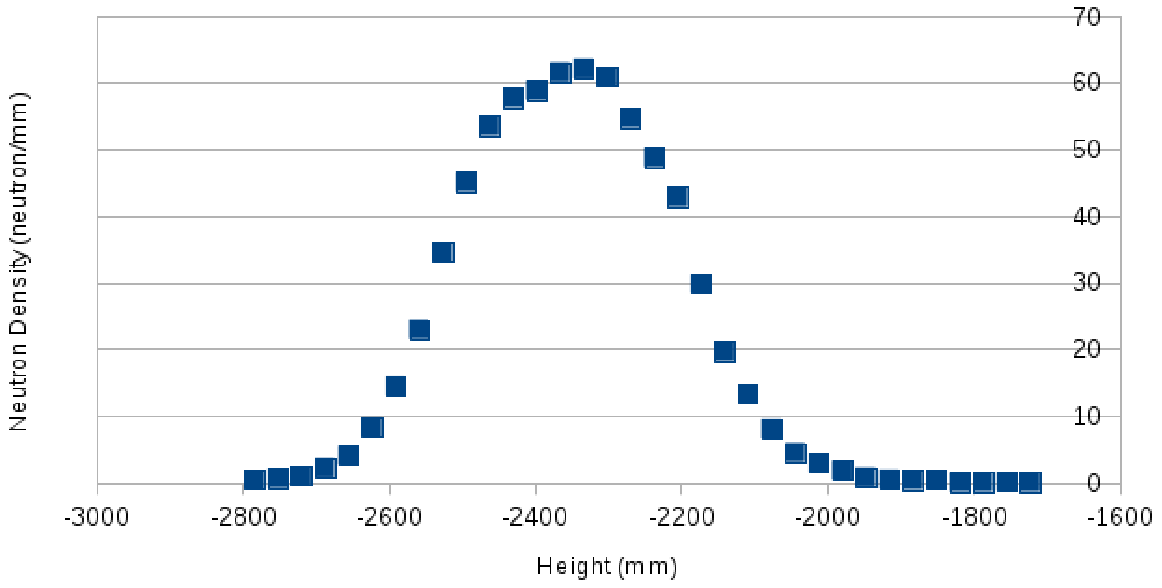


Figure 13: Vertical neutron density

The radial neutron distribution of the reactor using the center of the core, not to be confused with the reactor pool, as the origin gives the radial distribution density plotted in Figure 14.

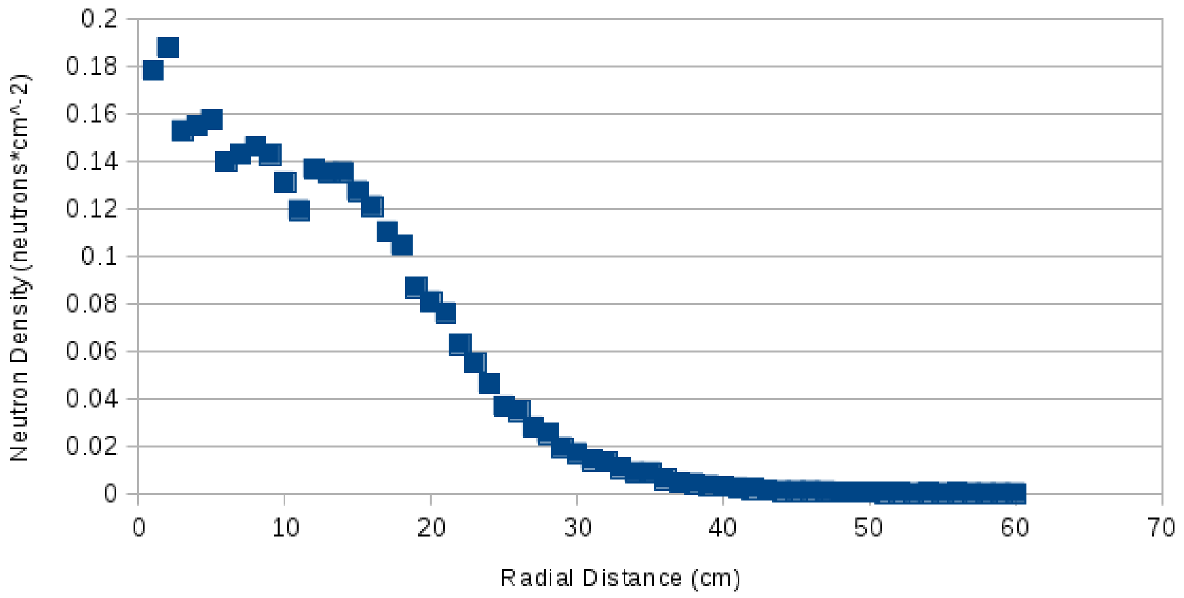


Figure 14: Radial neutron density distribution

Figure 14 shows an expected neutron distribution which decreases as the distance from the center of the reactor core increases. Figure 15 is a plot of x-coordinate versus y-coordinate of 25000 neutrons with respect to the center of the reactor core.

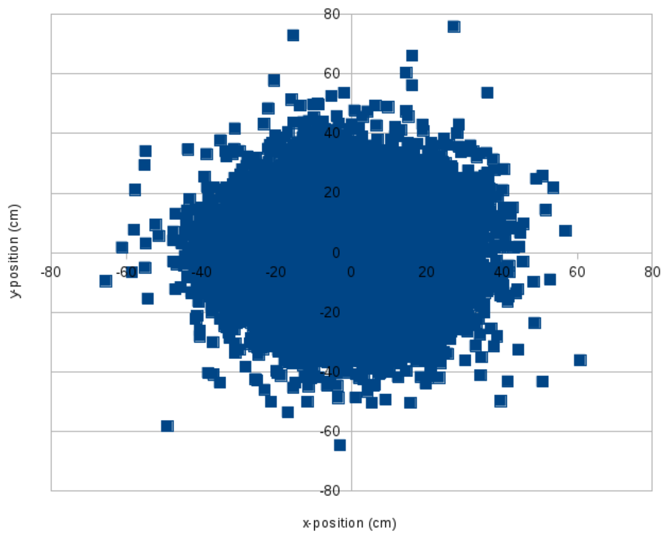


Figure 15: Neutron distribution

Knowing that the beryllium reflector spans from 11.049cm to 21.234cm and looking at Figure 14, it can be seen that the density starts dropping rapidly around 11 cm which is what is expected and the neutron density is very small once the outside of the reactor has bee reached. The same can be observed from Figure 15 which show that most of the neutrons are contained inside the reactor core.

5.1.8 Neutron Energy Distribution

Figure 16 shows the energy distribution of the neutrons inside the reactor. It looks like somewhat like a Boltzmann energy distribution, which is what is expected since the neutron are in thermal equilibrium with their surroundings and act as a gas, but the energy is way to high 1keV most probable energy coresponds to a temperature of about 1.16×10^7K. The fuel bundles being at 57^0C, one would expect an energy distribution closer to 0.0284eV. No explanation as of now.

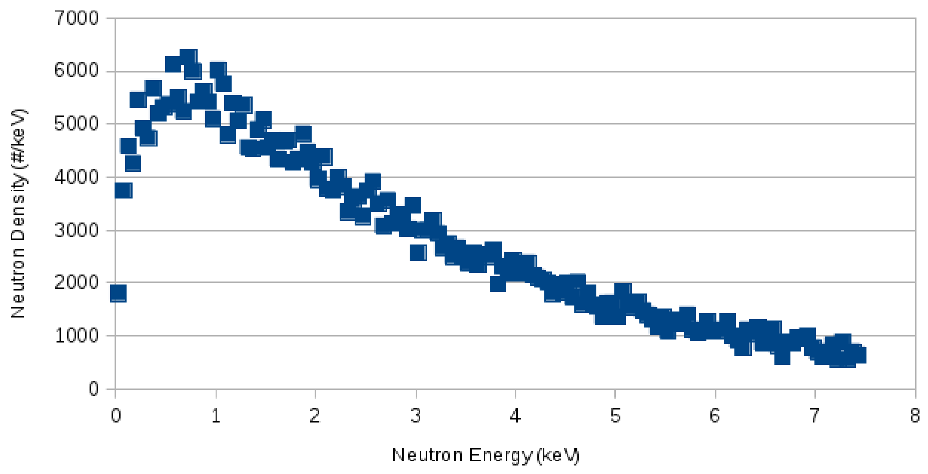


Figure 16: Neutron energy distribution

5.1.9 Momentum Distribution

The simulated distribution of the neutron momentum's inside the SLOWPOKE-2 reactor is shown in Figure 17. As expected, the neutron momentum's in all three directions are symetric about 0 and have the same relative values. The reason this is important is that if the neutrons are in thermal equilibrium with their surroundings and act as a gas there should be no preferential direction which is exactly what is observed. The shape looks like a Gaussian which is also to be expected.

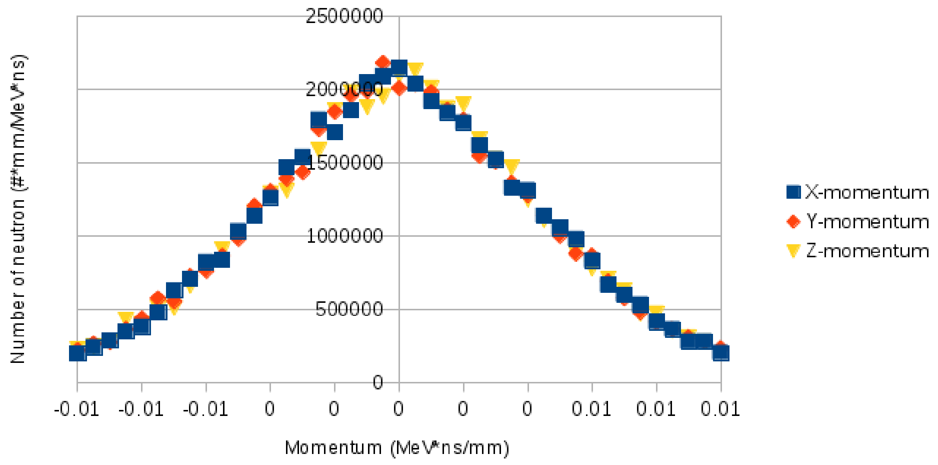


Figure 17: Neutron momentum distribution

The neutron source distributions were saved in source files such that pre-converted neutron distributions could be used during to decrease computational times during the following simulations.

5.2 Verifying the old temperature tracking code

5.2.1 Attempt #1

The goal of this simulation was to verify that the piece of code that was added, to change the temperature of the geometries based on the number of fissions that occurred inside the geometry during the last run and the power generated by the reactor at that point in time, works. The results from the simulation are presented below.

Table 4: Results obtained for the different runs (ran using 25 processors on the computer cluster)

Run Number		**1**	**2**
Results	Mean keff (k)	1.0067	1.0064
	Standard Deviation (k)	0.00123	0.00123
	Standard Deviation of Mean (k)	0.000329	0.000329
	Total Simulation Time (hours)	31.9	31.6
Inputs	Number of Primaries per Event	5000	5000
	Number of Events per Run	4408	4408
	Run Duration (ms)	0.5	0.5
	Instantaneous Delays	Yes	Yes
	Total Number of Primaries Simulated	22040000	22040000

Note: the standard deviation was estimated using runs done to determine the reactivity worth of the control rod since the standard deviation of one run cannot be calculated. Although one could calculate the keff of each event and return the STD of the mean to get an better estimate this might be worth implementing. Also one last thing to note is that the simulation was performed with the control rod pulled out of the reactor. The reactor power was set to 100,000kW, this might seem like a high power, but the goal was not to be realistic, but to see whether the temperature changing algorithm worked, or not. The number was worked out a follows:

$$\text{Fuel specific heat capacity (298K)} = 0.24115195\ \frac{J}{g \times K}$$

$$\text{Fuel density (298K)} = 10.6\ \frac{g}{cm^3}$$

$$\text{Fuel rod volume (298K)} = 3.0377\ cm^3$$

$$\text{Fuel heat capacity (298K)} = 7.765\ \frac{J}{K}$$

$$\text{Number of fuel rods} = 198$$

$$\text{Total fuel heat capacity} = 1537.47\ \frac{J}{K}$$

$$\text{Average fuel rod temperature change from one run to another due to 100,000kW power} = 32.5\ \text{K}$$

To put this power into perspective, the SLOWPOKE settles at about 70kW when the control rod is fully pulled out which means that in 1 second the 70 kJ of energy is deposited into the reactor. In this case a run lasts 0.5ms which means that at 100,000kW the energy deposited in one run is 50kJ. This is saying that in one run the equivalent energy of 1 second of real operation was added to the reactor. The difference is that in the simulation, the neutron only get 0.5 ms of transient time and the change is not as incremental as would be seen with a smoother change in temperature. The present results are inconclusive as to whether there really was a change in the temperature of the rods and the reason is that the keff are to close to one another. The difference between them is 0.3mk, but the approximated standard deviation of each of the results is about 0.32mk . One good thing about the results is that the reactivity of the second run (the one where the fuel temperature should have increased) is lower than the first one which is what one would expect (i.e. negative feedback from the temperature change).

Table 5: Material temperatures

Material	Run #1		Run #2	
	Temperature		Temperature	
	(K)	(^0C)	(K)	(^0C)
Be Reflector	295.65	22.5	295.65	22.5
Light Water	303.75	30.6	303.75	30.6
Air Gaps	291.15	18.0	291.15	18.0
Zirconium Support Gird	325.29	52.14	325.29	52.14
Aluminum Alloy	295.15	22.0	295.15	22.0
Cadmium Rod	323.15	50.0	323.15	50.0
Heavy Water	293.65	20.5	293.65	20.5
Fuel Rod	330.47	57.32	362.97	89.82

5.2.2 Attempt #2

Since the change in reactivity of the reactor was too small during the last run, the reactor power was increased to 250,000kW which in one run (0.5ms) is the equivalent to 125kJ. This is equivalent to 2.5s of real reactor operation at 50kW power level.

Table 6: Results obtained for the run before and after heat injection (ran using 25 processors on the computer cluster)

Run Number		**1**	**2**
Results	Mean keff (k)	1.0066	1.0050
	Standard Deviation (k)	0.00296	0.00296
	Standard Deviation of Mean (k)	0.000419	0.000419
	Total Simulation Time (hours)	20.17	19.99
Inputs	Number of Primaries per Event	5000	5000
	Number of Events per Run	2900	2900
	Run Duration (ms)	0.5	0.5
	Instantaneous Delays	Yes	Yes
	Total Number of Primaries Simulated	14500000	14500000

Note that the standard deviation was estimated using the same technique as for attempt #1. Also, the simulation was performed with the control rod pulled out of the reactor.

The difference in the reactivity worth before and after heat injection is 1.6 ± 0.84 mk. This means that the separation of the reactivity before and after heat injection is about twp times the sum of their respective standard deviation. This means that there is a very high probability that their actual values are different and that this difference is not due to statistical variation. With this result it is possible to conclude that the temperature changing algorithm actually works in changing the temperature of the geometries.

No full simulation verification of the full code added perform heat transfer as well as temperature tracking

5.3 Testing reactivity for steady state temperatures according to AECL report [3]

In AECL NUCLEAR REVIEW can be found an article which simulated [3] the steady state temperature of the SLOWPOKE-2 when the control rod is fully pulled out. These number will be tested with the current simulation to determine if they lead to a keff which is closer to 1, which is what one would expect. The results are presented in table 7 and the temperatures used for the geometries are shown in Table 8.

Table 7: Results obtained when temperature of materials were changed to steady state temperature according to [3] (ran using 25 processors on the computer cluster)

Results	Mean keff (k)	1.009486
	Standard Deviation (k)	0.002110
	Standard Deviation of Mean (k)	0.000298
	Total Simulation Time (hours)	39.58
Inputs	Number of Runs	50
	Number of Primaries per Event	2000
	Number of Events per Run	270
	Run Duration (ms)	0.5
	Instantaneous Delays	Yes
	Total Number of Primaries Simulated	27000000

A very interesting result arises and it is that increasing the temperature of the geometries increases the reactivity worth of the reactor. The reactivity of the reactor as a function of the run number is shown in Figure 18.

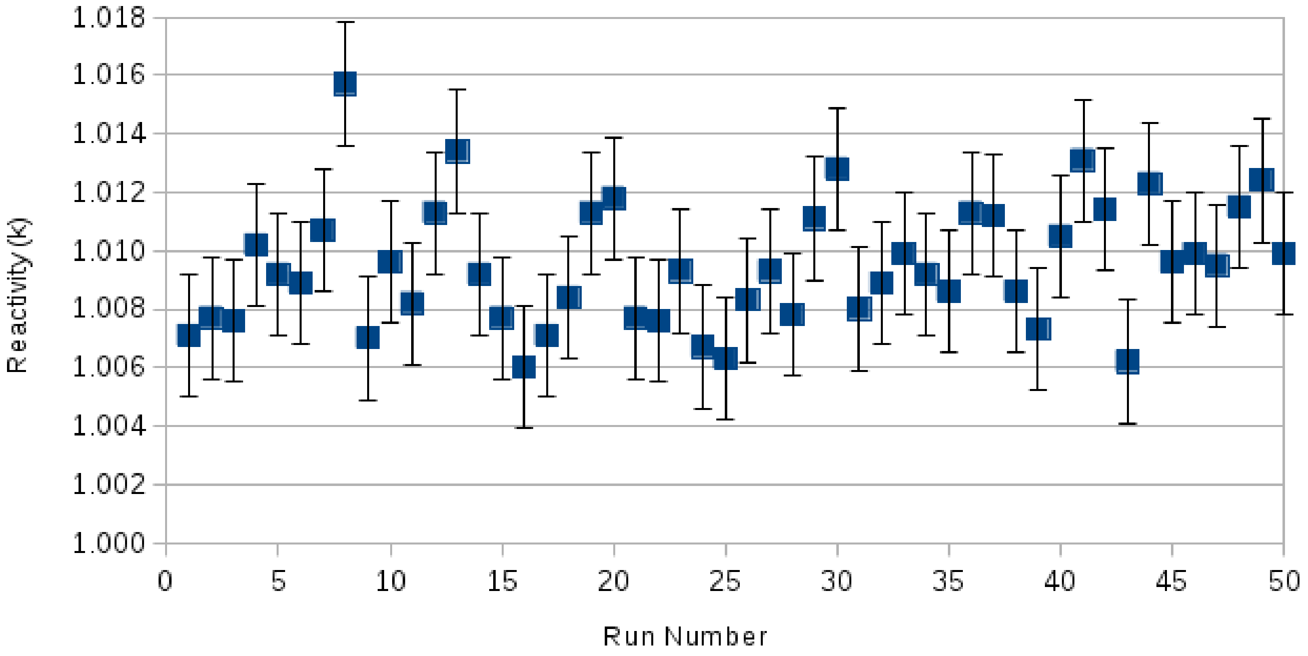


Figure 18: Reactivity of the reactor as a function of the run number

The temperature of the different geometries is shown below in Table 8.

Table 8: Material temperatures

Material	Temperature (K)	Temperature (^0C)
Be Reflector	328.15	55.0
Light Water	330.15	57.0
Air Gaps	383.15	110.0
Zirconium Support Gird	380.15	107.0
Aluminum Alloy	315.15	42.0
Cadmium Rod	338.15	65.0
Heavy Water	303.65	30.5
Fuel Rod	398.15	125.0

5.4 Testing change in reactivity for change in temperature of different material

To determine where most of the reactivity increase comes from the temperature of the two main geometries is changed one at a time so as to determine where the reactivity increase comes from. The results are shown in Table 9.

Table 9: Results obtained from changing the temperatures of the different materials one at a time (ran using 25 processors on the computer cluster)

Geometry Temperature Change		Fuel	Water
Results	Mean keff (k)	1.004488	1.00895
	Standard Deviation (k)	0.003076	0.00295
	Standard Deviation of Mean (k)	0.000435	0.00042
	Total Simulation Time (hours)	26.05	26.51
Inputs	Number of Runs	50	50
	Number of Primaries per Event	1440	1440
	Number of Events per Run	250	250
	Run Duration (ms)	0.5	0.5
	Instantaneous Delays	Yes	Yes
	Total Number of Primaries Simulated	18000000	18000000

The reactivity of the reactor as a function of the run number is shown in Figure 19 for the two different material temperature change and as can be seen the simulation converges within the first two runs. From these results it is easy to see where the increase in reactivity of the SLOWPOKE reactor came from. As the temperature of the water increases, the reactivity increases as well. In the case of the fuel, as the temperature increases the reactivity decreases. One odd thing is that in table 7 the reactor reactivity is 1.0094 which is higher than the two calculated above, but as the temperature increases, one would expect the two opposite thermal feedback mechanism to somehow cancel one another.

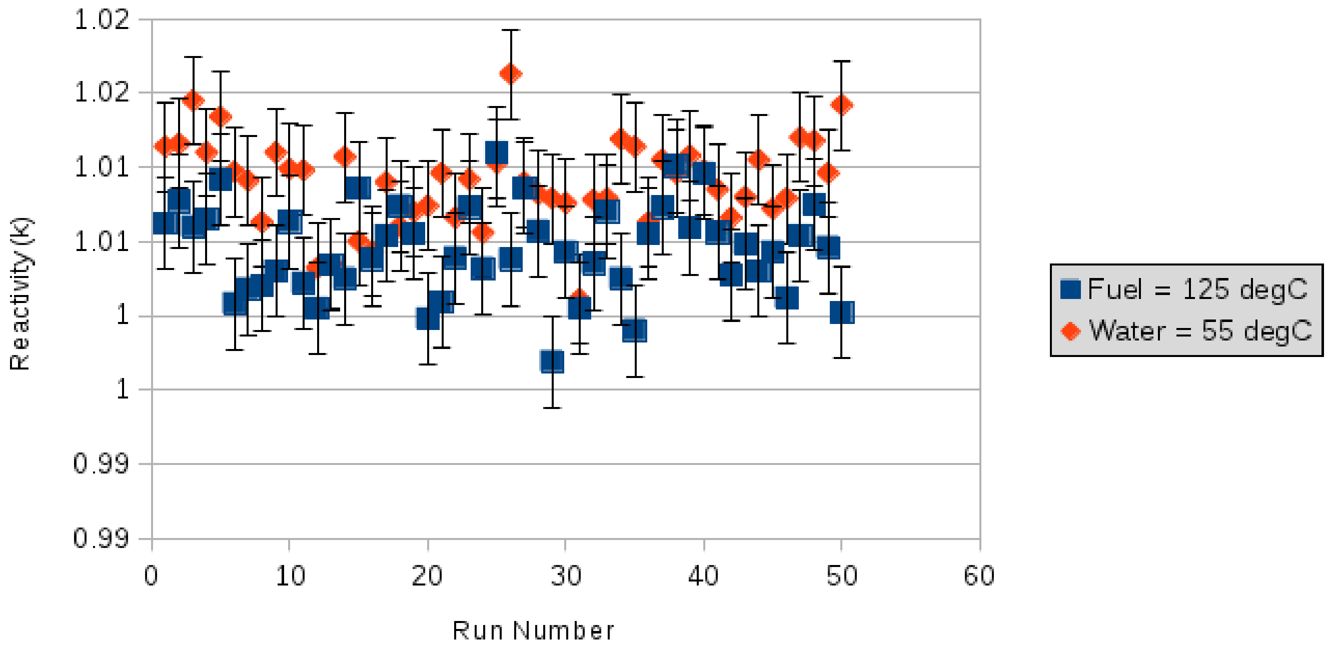


Figure 19: Reactivity of the reactor as a function of the run number for changing the temperature of different geometries

Table 10: Material temperatures

Material	**Fuel temperature change**		**Water temperature change**	
	Temperature		**Temperature**	
	(K)	(^0C)	**(K)**	(^0C)
Be Reflector	295.65	22.5	295.65	22.5
Light Water	303.75	30.6	330.15	57.0
Air Gaps	291.15	18.0	291.15	18.0
Zirconium Support Gird	325.29	52.14	325.29	52.14
Aluminum Alloy	295.15	22.0	295.15	22.0
Cadmium Rod	323.15	50.0	323.15	50.0
Heavy Water	293.65	20.5	293.65	20.5
Fuel Rod	398.15	125.0	330.47	57.32

6 Conclusion

The present report presented a heat transfer algorithm which has been implemented in G4-STORK and can be used to keep track of the fission events inside the reactor. These events are then used to keep track of the temperature changes of the different geometries inside the nuclear reactor on the specified grid. One thing to note is that this algorithm has not been fully tested in that a complete verification using an actual full scale run has yet to be performed yet, but it has been tested using small runs and found to work. The method on how to use the new heat transfer algorithm is depicted in Appendix A1. An earlier version of the code denoted old thermal tracking code, which only account for where the fission energy was deposited in the reactor, will also be given as a backup since it has been tested by performing full runs and it was found to work. The set-up procedure for the old thermal tracking code is given in Appendix A2.

The way the SLOWPOKE-2 reactor was built was detailed such that the design philosophy was well laid out. Mainly, the reactor was built from the inside out to make it easier to catch mistakes, and the reactor core was split into 6-pie slices to increase computational speed in the mother/daughter hierarchy. Simulation of the SLOWPOKE-2 geometry as presented were performed and encouraging results were obtained. The reactivity was higher than one when the control rod is fully pulled out and smaller than one when the control rod wais fully inserted into the reactor. The neutron distribution in energy, space and momentum were also obtained. The temperature of the various material was then increases to determine how the reactivity changes with changing temperature. It was found that the reactivity of the reactor increases with increasing water temperature and decreases with increasing fuel temperature.

6.1 Further Improvement

It is highly recommended to find a way to time how much total computational time, number of times called times time per call, the program spends in different functions to be able to decide what needs to be looked at further. That way one would be able to determine what needs to be optimized is it the geometry, the cross section broadening, or the fission generation process, or something else altogether.

The SLOWPOKE-2 geometry is highly optimized as of now, but there is no doubt changes which could be made to increase the geometry navigation efficiency, but the gains are probably going to be small. The low efficiency gain can be guessed at by the fact that removing the water circulation holes altogether only decreases the computational time by 6% approximately. By number of geometries, there are more than 1176 water circulation holes in the reactor accounting for more than 84% of the geometries (by number) making up the reactor. This means that the maximum efficiency increase from rearranging/deleting geometries is probably in the 6-10% range.

There are some improvements that could be made to increase the computation speed of the heat transfer class. The first stems from the fact that the heat tranfer is only computed by one processor (the master) instead of being paralyzed and computed by many processor (including the slaves); therefore, speed could be increased here since the slaves represent a lot more computational power than the master alone. Another imperfection was mentioned during the finite difference approximation and it is that much time could be gained by determining if the calculation of the cross term is necessary beforehand; therefore, preventing its calculation altogether if not necessary which has not yet been implemented in the code.

Unless some new method of keeping track of the neutron population inside the reactor is performed, which is 1000 times faster, there seems to be no way to simulate for longer than 0.1s without making some compromise and losing accuracy in terms of transient. A compromise could be to use linear interpolation of the material temperatures over large time periods using the short time simulation using Monte-Carlo as a reference for the way the material properties and reactor power will vary.

References

[1] J. R. M. Pierre, "Low Enrichement Uranium (LEU)-Fueled SLOWPOKE-2 Nuclear Reacotr Simulation with the Monte-Carlo Based MCNP 4A Code," *Royal Military College of Canada,* Kingston, Ontario, May, 1996.

[2] G. A. Burbidge, R. T. Jones and B. M. Townes, "Commissioning of the SLOWPOKE-2 (RMC) Reactor," *Atomic Energy of Canada Limited Report,* RCC/TR-85-004, Rev. 1, 1985.

[3] S. Yue, S. Polugari, V. Anghel, J. Hilborn and B. Sur, "A New Safety Principle for the SLOWPOKE-2 Reactor," *AECL NUCLEAR REVIEW,* Vol. 2, No. 1, June 2013, Chalk River, Ontario, Canada, p. 113-118.

Appendices

A Using Heat Transfer Function

A.1 New Thermal Transfer Code

The first things to do when one wants to use the new heat transfer module is to activate it inside the input file, this is done by specifying the following command.

Step 1:
FISSION_ENERGY_TRACKING 1

This tells G4-STORK that the energy deposited through fissions needs to be kept track of. The second piece of information needed to perform heat transfer is the initial thermal power of the reactor which can be specified using the following.

Step 2:
REACTOR_POWER XXX

Where XXX will be the initial thermal power of the reactor in units of kW; in other words, if XXX = 100, the thermal power of the reactor is 100kW. The last piece of information to be specified in the input file is the temperature grid divisions. This tells the program how many division to do in the x, y, z, and time direction respectively. This is done as follows:

Step 3:
TEMP_GRID_DIVISION XXX YYY ZZZ TTT

Where XXX will be the number of divisions in the x-direction, YYY will be the number of divisions in the y-direction, ZZZ the number in the z-direction, and finally TTT will be the number of time divisions. The time taken to compute the heat transfer is proportional to the product of all three of these numbers, but the higher they are the more precise the heat distribution inside the nuclear reactor will be.

The user must the specify the grid over which the heat transfer must occur inside the world constructor as follows:

Step 4:

```
WolrdConstructor::WorldConstructor() : StorkVWorldConstructor()
{
    ... Normal world initiation ...

    // Setting up the thermal grid
    thermalGrid[0] = G4ThreeVector(200*mm,200*mm,-1500*mm);
    thermalGrid[1] = G4ThreeVector(-900*mm,-900*mm,-2800*mm);

}
```

The thermal transfer will happen on a grid starting at x=-900mm and ending at x=200mm, y=-900mm to 200mm, and z=-2800mm to 1500mm. The order of the two vectors do not matter, but they must lie on opposite corners of the cube one is trying to define. The last step is to define material which will be affected by the heat transfer and those who will be affected but can be overwritten by the interpolation vector. This is done as follows:

Step 5:

```
void WORLDConstructor::ConstructMaterials()
{
        ... Normal Material Definition ...

    matMap["Galactic"] = World;
    matMap["H2O"] = LW;
    matMap["D2O"] = HW;
    matMap["Zirconium"] = Zr;
    matMap["AlAlloy1"] = AlAlloy1;
    matMap["AlAlloy2"] = AlAlloy2;
    matMap["AlAlloy3"] = AlAlloy3;
    matMap["AlAlloy4"] = AlAlloy4;
    matMap["Reflector"] = Reflector;
    matMap["Cadmium"] = Cadmium;
    matMap["Air"] = Air;

    // (Material #3) Fuel Rods (19.95% Enriched Uranium in (UO2))
    std::stringstream matName;
    if(Initiation)
    {
        for(G4int i = 0; i <34; i++)
        {
            matName.str("");
            matName << "Fuel" << i;
            Fuel = new StorkMaterial(matName.str(), FuelDensity, 2, 0.2411519506*joule/g/kelvin,\
                                            21.5*joule/(s*m*kelvin), kStateSolid, FuelTemp);
            Fuel->AddElement(Oxygen, 2);
            Fuel->AddElement(LEU,    1);
            matMap[matName.str()] = Fuel;
        }
        Initiation = false;
    }
}
```

In the example above the only material which is exclusively affected by the heat transfer module is the fuel. If the material properties vector was to be flagged because the user input a material property change vector, the fuel would not be affected by the vector (even if it is targeted directly); on the other hand, all the other material would regain their original values and only the one targeted by the vector would see changes in its original properties.

A.2 Old Thermal Transfer Code

In the case of the old thermal transfer code only step 1-2, 5, shown in Appendix A1, need to be executed since no grid set-up is needed. The smaller set-up time is offset by the fact that no heat transfer is simulated in this case.