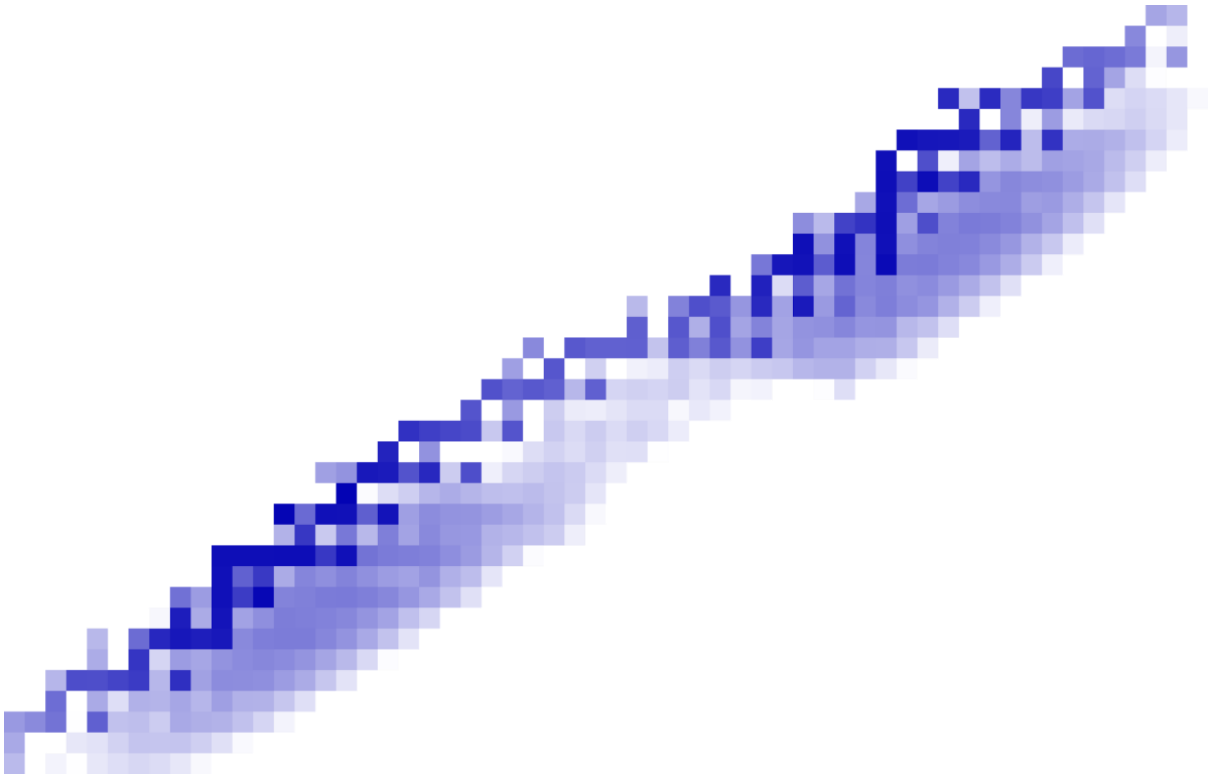
THORIUM:
THE NEXT REACTOR

What are the characteristics of the products of the thorium cycle in a Liquid Fluoride Thorium Reactor?



Rembrandt Klazinga
Noor Schroen
6Vb, Coornhert Lyceum

“Goodness, I know nothing about nuclear energy.”

James Dyson

Table of Contents

Preface

When we came up with thorium as a subject, the reactions we got were an interesting mix of 'what is that?' and 'where did you hear of it?'.
The former will hopefully be answered in depth in the following pages, and the latter we can shed some light on now.

To put it plainly: we both stumbled on thorium by complete accident. Perhaps a mention of it in an interesting article or an unexpected YouTube link or something similar. What we can say for certain, is that we were both hooked from the moment we found it.
The fact that the entire subject is surrounded by a cloud of internet hype and that basically no large scale experiments have ever been run did not curb our enthusiasm much.
When the actual PWS came along, we at least knew what we wanted, although there was a fair amount of resistance from others.

"Too experimental", "too theoretical" and "too broad" were all mentioned multiple times, and on two occasions we were forced to temporarily switch away from thorium, only to take some new opportunity to reintroduce it.

What finally allowed us to actually choose thorium was the partnership with Fluor, an engineering, procurement and construction (EPC) company that has partnered with our school in the past, including supplying employees for guidance with students' projects. When we heard that PWSs at Fluor are allowed to be far more theoretical than usual, we applied as soon as possible.

After getting their support on the project, one of our counsellors even managed to get in touch with a manager at the Nuclear Research and Consultancy Group (NRG) in Petten, which culminated in a visit and an hour long discussion with two employees at Petten who have worked with NRG's own research reactor.

Using a combination of our own research and the wonderful help provided by Fluor and NRG, we created this paper. The writing went with surprising ease, and we were very happy with how educative the project has been for us.

Introduction

Many people are searching for new ways to produce energy after they have been convinced that burning fossil fuels is not sustainable. New sources include all green energy such as from windmills and solar panels. Alternatively, nuclear energy has often been proposed as a viable large scale power source. It has been around for quite a while, but has made very little scientific progress and stayed nearly unchanged over the course of more than 50 years.

The vast majority of current commercial reactors still use uranium as their fuel, even though, theoretically speaking, it is not the best choice, partially due to its rarity: the type of uranium that is used in reactors makes up less than a hundredth of all uranium, which in itself is extremely rare.

During the last couple of years, people have started to consider the element thorium as a new source for producing energy, because of the theoretical advantages it has over uranium. Using thorium should be safer than using uranium, because of a safer reactor design. Also, the same amount of thorium should produce more energy than uranium, and thorium is more plentiful than uranium. Depending on who you are, the fact that thorium reactors don't produce material for nuclear weapons can be a pro or a con.

Most of this is theoretical, however, as a large scale self-sustaining thorium reactor is yet to be built. Companies like FLiBe Energy, led by ex-NASA engineer Kirk Sorensen are trying to attract a bigger audience and to fundraise in order to build commercial thorium-based reactors.

In this PWS we will discuss thorium by attempting to answer the question: "What are the characteristics of the products of the thorium cycle in a Liquid Fluoride Thorium Reactor?".

This question can be divided into the following subtopics:

What does the thorium cycle look like?
In chapter 3 this is covered with an explanation of the thorium fuel cycle and the reactions involved.

What is a Liquid Fluoride Thorium Reactor?
This requires a broad look at nuclear reactors in chapter 4, which then focuses on the category of reactors that the LFTR falls into, finally settling on the LFTR itself.

In what ratios are the products formed?
We answer this question with chapter 5, which describes our experiment: simulating a LFTR.

What are further characteristics of the products?
Using the results from our experiment we analyse the products and determine some of their other characteristics in chapter 6.

One of the main objections people have when the construction of a new nuclear reactor is planned, is that the fission products are dangerous because they are radioactive. However, not all fission products are radioactive and not all radioactive material is undesirable. With this research, we hope to clarify which products are created, which ones are hazardous and which ones are useful for other industries.

But before we can start working on the subtopics, we need to lay a foundation in the form of background information on nuclear physics as well as the terms that are used when discussing it.

Chapter 1: Definitions

1.1: The Atom

The atom is the smallest 'unique' building block in physics, meaning that it still has different chemical properties than other objects of a similar size. Each atom is constructed of two parts: the nucleus and a surrounding cloud of electrons. All atoms with the same number of protons are grouped under the same element.

The Nucleus

The nucleus is made up of neutrons and protons and makes up virtually all of the mass of an atom. Protons have a positive charge (of +1e, or 1.602×10^{-19} C), while neutrons have no charge. Both neutrons and protons weigh about 1u, equal to 1.672×10^{-27} kg.

The chemical characteristics of atoms are determined by the number of protons (and the resulting numbers of electrons). Elements with different numbers of neutrons are called isotopes of that element. Nuclide is a more general term for a nucleus configuration with a certain number of protons and neutrons. While the terms nuclide and isotope are often used interchangeably, the correct usage would be: two isotopes of an element are different nuclides.

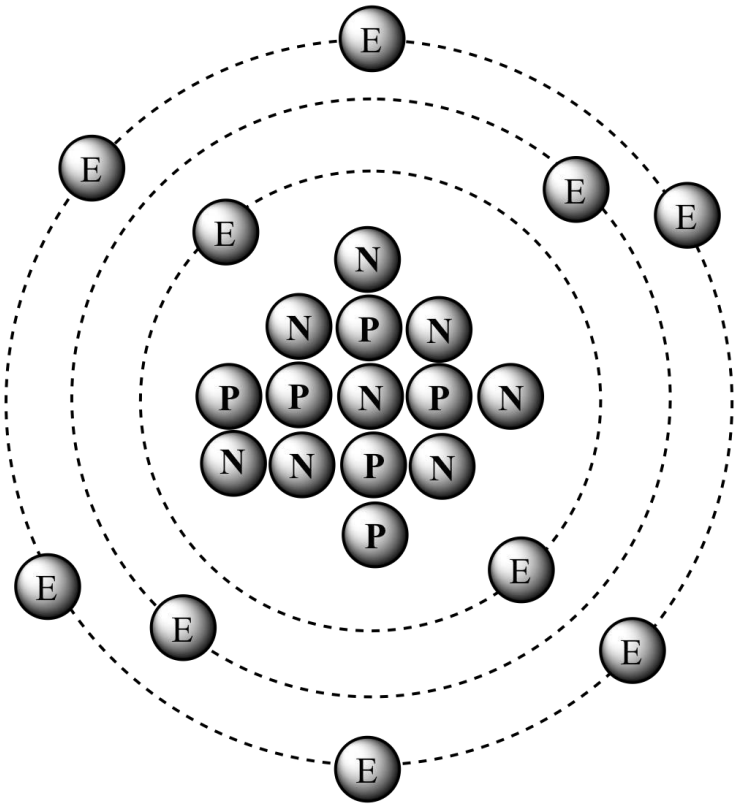


Figure 1.1: Bohr model of a C-14 atom

The Electron

Electrons swarm around the nucleus and have a charge of -1e, to counter the charge of a proton. Because an atom as a whole does not have a charge, the number of electrons and protons must be the same. Electrons have almost no mass at 5.486×10^{-4} u.

The Ion

When an atom loses or gains any number of electrons, and as a result the number of protons and electrons are not the same, it gets a charge, and it is now specifically defined as an ion. When an ion is positively charged, it has one or more electrons less than it has protons, and when it has a negative charge, it has more electrons than protons.

Notation

There are a couple of ways to note an atom. The most used way is:

${}^{A}_{Z}\mathrm{X}$, where

X= the chemical symbol for the element
Z= the number of protons
A= the mass number, or N+Z
N= the number of neutrons

As an example, a helium atom with 2 protons and 2 neutrons: ${}^{4}_{2}\mathrm{He}$

Because each element always has the same number of protons, you can also note it as:

${}^{A}\mathrm{X}$

Here the previous helium atom is noted as: ${}^{4}\,\mathrm{He}$

When writing it is easier to use the notation: X-A
With this you can abbreviate the official method to helium-4 or just He-4.
The latter two notations are the ones that we will use the most in regular text.

Typically, atoms are grouped by the number of protons in their nucleus on the periodic table of elements, where each rectangle represents an element.

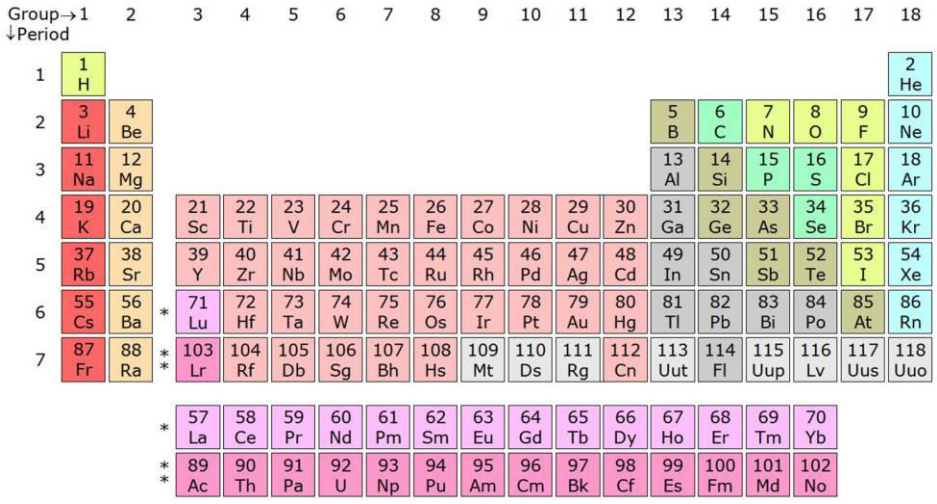


Figure 1.2: the periodic table of elements

1.2: Free neutrons

While neutrons exist in the nuclei of atoms, in reactors they are also common as loose particles and play a massive role in reactor function, in part by instigating nuclear fission: the process of splitting a nuclide. Outside of reactors, however, they are virtually non-existent.

Neutron flux

The 'neutron flux' is a common term used to describe the concentration of neutrons at any given point. The flux is defined as the number of neutrons passing through a square cm every second, and can reach levels of 10^{15} neutrons * cm^{-2} * s^{-1} in a reactor specifically built to maximise neutron production.

Neutron energy

Free neutrons are classified based on their kinetic energy. Their energy is mostly given in electronvolts (eV), but their energy can also be given using the temperature.
While there are many different groups of free neutrons classified according to their energy, only three energy groups are needed for reactor physics: thermal neutrons, resonance neutrons and fast neutrons. In this division of energy ranges, thermal neutrons have an energy range of 0.025eV to 1eV, resonance neutrons have an energy between 1eV and 1keV and fast neutrons range between 1keV and 10MeV.[3]

Neutron cross section

To determine the chance that a free neutron will react with a nuclide, the neutron cross section is used. The term cross section is used for different sorts of calculations, and the general definition of cross section is the area that determines the likelihood of an interaction between an incident object and a target object. For the neutron cross section, the incident object is the neutron and the target object is the nuclide, and in this case the effective area is given in barns (=10^{-24} cm^2). Using that area, it is then possible to calculate the chance that a neutron will interact with a nuclide.
The terms 'neutron cross section' and 'cross section' are used interchangeably, as the neutron cross section is the only type of cross section discussed.

Imagine throwing darts at a target. The size of the target is the neutron cross-section, and the number of darts you throw is the neutron flux. The bigger the target and the more darts you throw, the bigger the chance that the target is hit.

There are many different types of neutron cross sections for different types of interaction, but the three most common are the scattering, capture and fission cross sections. The total of all types of cross sections is the total cross section, which measures the probability that the neutron will do anything with the nucleus at all, instead of passing it by.[4]

1.3: Radioactivity

When a nuclide of any element is unstable, it is considered radioactive, and it emits radioactive particles, including α-, β-, and γ-particles. In that process it changes into another nuclide. The process of radioactive decay is completely random and therefore it is impossible to predict when a single particle will decay. However, calculating how a group of the same nuclides will decay is possible with the use of their half-life. The half-life of a nuclide is the time it takes for half of a group of nuclides to decay.

Dangers of radioactivity

Half-lives can also give an insight as to how dangerous certain isotopes are, although a lot more information is needed to measure the danger of its radioactivity. A short-living nuclide delivers a high dose of radioactivity, but because it decays so fast, the nuclide itself will not get very far, and only people near the source will be affected. Nuclides with longer half-lives, ranging from a couple of days to a couple of months pose a different threat, as they can travel longer distances before decaying, but still decay quickly enough to deliver a pretty high dose of radiation. Very long-living nuclides pose another problem entirely: long term contamination. In a short period of time, these nuclides do not emit much radiation, but because they are so long-lasting, they deliver a constant flow of radioactive particles during a long time, which makes a contaminated area unfit for humans to live in, possibly for millennia. Beyond that, nuclides with an extremely long half-life decay so slowly that their damage becomes irrelevant.[1]

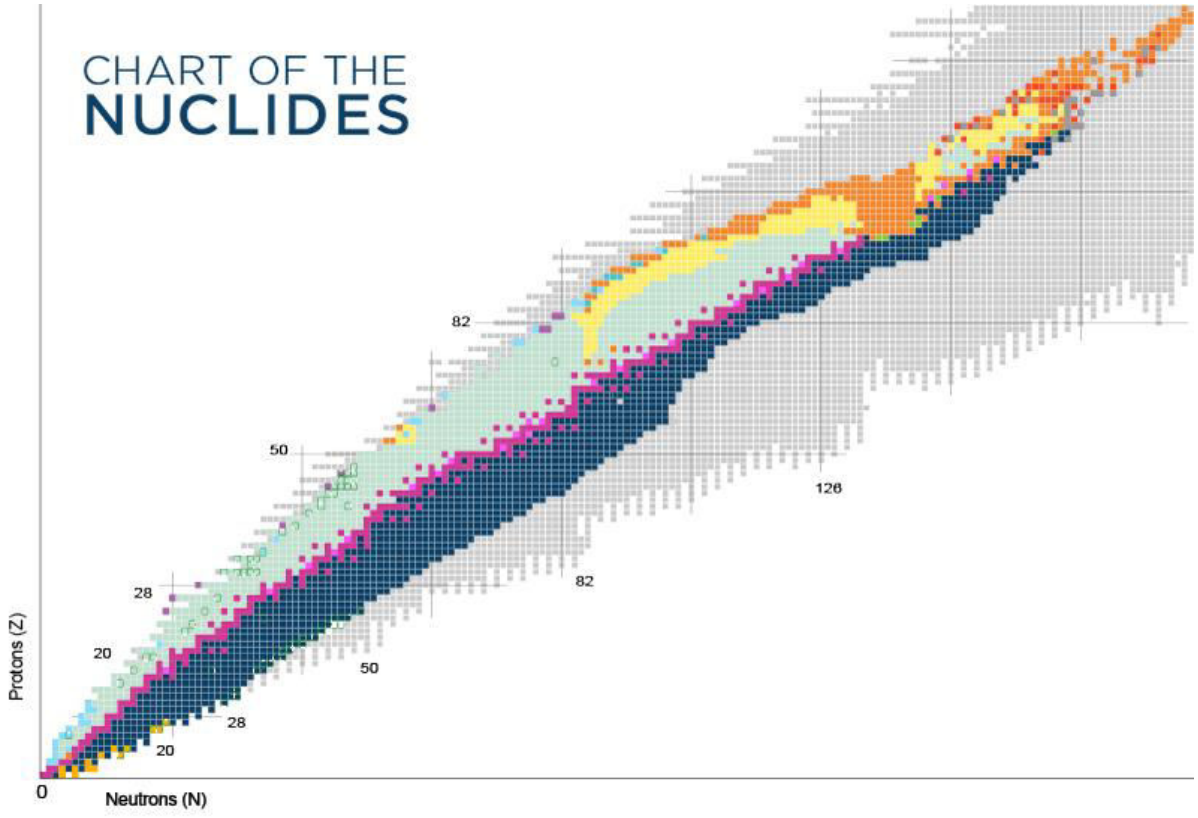


Figure 1.3: chart depicting the decay type of all isotopes.
Pink is stable, dark blue is β^-, light blue is β^+, orange is α
Note the decay types of the isotopes above and below the stability line.

α-Decay

An α-particle is made up of two neutrons and two protons. This is the same as a Helium nucleus with two neutrons, and because of that an α-particle is not only shown as ${}^{4}_{2}\alpha$, but also as ${}^{4}_{2}\mathrm{He}$. α-Decay is mostly present in extremely heavy nuclei. An example of α-decay is:

$${}^{224}_{88}\mathrm{Ra} \rightarrow {}^{4}_{2}\alpha + {}^{220}_{86}\mathrm{Rn}$$

β-Decay

A β-particle is the same as either an electron or a positron (which has the same mass as an electron and a charge of +1e, whereas an electron has a charge of -1e). A β-particle can have one of two charges corresponding with the charges of electrons and positrons: a negative and a positive charge. β-Particles are notated in a few ways:

β^+ and β^- , ${}^{0}_{1}\beta$ and ${}^{0}_{-1}\beta$, ${}^{0}_{1}\mathrm{e}$ and ${}^{0}_{-1}\mathrm{e}$

β-decay is present in almost any type of nuclide, based only on their size. Two examples of β-decay are:

${}^{14}_{6}\mathrm{C} \rightarrow {}^{14}_{7}\mathrm{Rn} + {}^{0}_{-1}\mathrm{e}$, an example of β^-

${}^{10}_{6}\mathrm{C} \rightarrow {}^{10}_{5}\mathrm{B} + {}^{0}_{1}\mathrm{e}$, an example of β^+

In the case of β^-, one neutron decays into a proton and an electron:

${}^{1}_{0}\mathrm{n} \rightarrow {}^{1}_{1}\mathrm{p} + {}^{0}_{-1}\mathrm{e}$

In the case of β^+, a proton decays into a neutron and a positron:

${}^{1}_{1}\mathrm{p} \rightarrow {}^{1}_{0}\mathrm{n} + {}^{0}_{1}\mathrm{e}$

β^--Decay is found in nuclides with a relatively large number of neutrons in the nucleus, these nuclides are found above the stability line in the chart of nuclides. Seeing as all atoms work towards being stable, these nuclides have to lose neutrons and gain protons to be stable. Likewise, β^+ decay is typically found in nuclides with relatively few neutrons, which are found under the stability line.

γ-Decay

When an atom has excess energy, it can get rid of it by emitting γ-rays, packets of electromagnetic energy, in the form of photons. Most, but not all, γ-radiation is hazardous, as it can travel through large amounts of material and still have the capacity to damage DNA enough to cause severe health risks.[2]

Activity, doses, and equivalent doses

The danger of a radioactive element cannot be judged by its mass, so to have an idea of how hazardous a nuclide is, we use its activity. The activity of a radioactive element is defined by the number of disintegrations per second in becquerel (Bq).

1Bq = 1 decay per second.

More disintegrations per second means a more radioactive element.

When interested in how much radiation a material absorbs, the dose is used to determine how much energy is absorbed per kg. For this we use the gray (Gy):

$$1\mathrm{Gy}=1\mathrm{J}/\mathrm{kg}$$

With the unit gray on its own, however, we cannot do much. Different material absorb different amounts of energy, different types of radiation do not cause the same amount of damage on one type of tissue when the amount of grays is the same. This is where the sievert (Sv) comes in. The equivalent dose takes the dose as well as a weighing factor into consideration. The weighing factor of γ- and β-decay is 1, and the factor for α-decay is 20, meaning that 1Gy of α-decay does 20 times more damage than 1Gy of either γ- or β-decay on the same tissue.

$$H = w_r \cdot D$$

H = the equivalent dose in Sv
w_r = the weighing factor
D = the dose in Gy

Chapter 2: Background Information

2.1: Energy from nuclei

It is possible to generate a lot of power using nuclei of atoms. This can be done in two different ways: nuclear fission and nuclear fusion.

Both fission and fusion work based on the same principle: in nuclear reactions, the products sometimes are slightly lighter than the atom(s) it started with. While at first glance this seems wrong or impossible, because conventional wisdom states that 'the mass of the products before and after any reaction must be same', this mass deficiency is perfectly normal according to nuclear physics and can be explained using Einstein's formula:

$$E = m \cdot c^2$$

For example,

$$1\text{kg} = (1 \cdot c^2)\text{J} = 8.988 \cdot 10^{16}\text{J}$$

This means that the 'missing' mass is released as energy, in the form of electromagnetic radiation. In nuclear power plants, this energy is usually used to turn water into steam, powering large turbines similar to those in a plant running on fossil fuels.

Nuclear Fission

In all commercial nuclear power plants, nuclear fission is used. Nuclear fission is the process in which an unstable nuclide splits, producing two fission products, which can be in the form of radioactive particles. To generate power using nuclear fission, a fissile nuclide, which is any nuclide that can easily fission, is hit with a neutron, which then captures that neutron in such a fashion that it becomes unstable. Due to this instability, the uranium splits and produces two smaller nuclides. As mentioned earlier, its products have a smaller total mass, and that mass deficiency is released as energy. Along with the two fission products, neutrons are released, which then can in turn hit other fissile nuclides, creating a chain reaction.

The number of fissions that, on average, are produced by one fission is called 'the effective neutron multiplication factor', often referred to with the letter 'k'. When:

- k<1 the chain reaction will slow down, as the number of reactions is decreasing
- k=1 the chain reaction will be sustained, this is what is wanted in a nuclear reactor
- k>1 the chain reaction will grow exponentially.
- k>>1 the chain reaction grows with such speed that it could lead to a meltdown or an explosion, if uncontrolled and when provided with enough fissile material.

Nuclear reactors are run by subtly changing the value of k using control rods. These rods, made of a material with a high absorption cross section for the present neutron energy, preventing the neutron from causing further fissions, can be extended or retracted.

Fully extended rods mean maximum neutron absorption, which in turn decreases the value of k: more of the released neutrons are absorbed by the control rods, which lowers the amount fission reactions taking place, resulting in less produced energy.
In a similar line of thought, fully retracted control rods raise the value of k, as the released neutrons can go their own way without being absorbed by the rods, which increases the likelihood of the neutrons instigating fission, in turn boosting the amount of produced energy.

Nuclear Fusion

The process in which two nuclei are so near each other that they fuse together, is called nuclear fusion. The sun uses nuclear fusion to create its energy, and it uses hydrogen to do so.

The amount of energy that is released from the reaction depends on the nucleus. Elements lighter than iron, 25 in total, release energy when fused, while all the heavier elements use energy to fuse.

Here on earth, there are a lot of experiments being done to try to replicate what happens in the sun's core. While nuclear fusion has actually been successful, it is not yet possible to create a man-made fusion reaction from which comes more energy than the energy spent to establish it in the first place.

2.2: Components of a nuclear reactor

Core

The core contains the fuel and generates the heat. The most used fuel at this moment is uranium, and it can usually be found as pellets of uranium oxide, which are arranged in tubes to form fuel rods.

Moderator

A moderator is a material, often water, which slows down neutrons released from fissions. When neutrons have a lower energy, elements have a higher neutron cross section, resulting in more fissions. Not all reactors have moderators though, as there are also reactors operating at a high neutron energy.

Control rods

Neutron absorbing materials such as boron, cadmium and hafnium are used to make control rods. These rods are inserted into or withdrawn from the core to control the neutron flux, regulating the reaction rate. When the control rods are partially inserted, they absorb the neutrons, lowering the neutron flux, resulting in fewer fissions. When the rods are fully inserted, the reactor is brought to a stop. [5]

Coolant

The primary coolant is a fluid which circulates through the core to transfer heat away from the core. When there are two cooling circuits, a secondary coolant is used to transfer the heat from the primary circuit to the secondary circuit.

Pressure vessel/pressure tubes

The reactor core, the moderator and the coolant are kept in a pressure vessel in most designs. In some, though, a series of pressure tubes holds the fuel and transports the coolant throughout the moderator, which surrounds it.

Steam generator

The steam generator is present in pressurized reactors, in the secondary circuit. This produces the steam to power the turbine.

The turbine

The turbine essentially generates electricity from the steam, just like in a fossil fuel plant. High pressure steam goes through the turbine, transferring energy and making it spin. The turbine is connected to an electricity generat, which generates electricity when spun.[9]

Containment

To contain the reactor, a structure is built around the reactor and its connected parts that is meant to separate the outside world from the dangers such as radiation in case of malfunction on the inside, but also to protect the inside from outside influences.[7][8]

2.3: Most common types of nuclear reactors

PWR

The Pressurized Water Reactor is one of the most common reactors. The nuclear fuel, the reactor rods, the moderator and the coolant are all contained in a reactor pressure vessel (rpv). The temperature in the core reaches temperatures of around 300°C, and because water is the moderator as well as the coolant, the water must be prevented from boiling. To achieve that, the pressure in the rpv must be kept at around 15MPa (150 times the atmospheric pressure). This is all the primary circuit; the coolant then flows towards the steam generator, where it heats the water in the secondary circuit, and because that is under such a pressure (6-7MPa) that it boils, it turns into steam, powering the turbine, ultimately producing power.

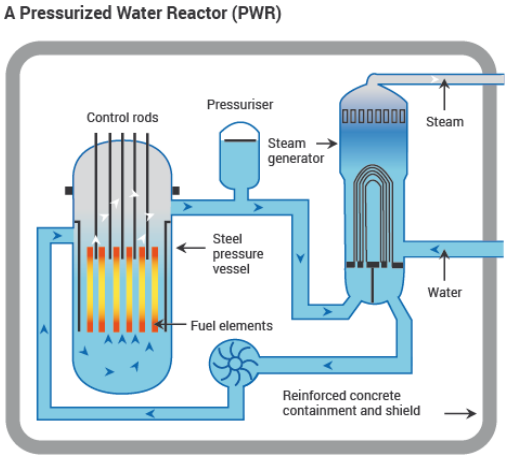


Figure 2.1: PWR

BWR

The Boiling Water Reactor is very similar to the PWR, but unlike the PWR it only consists of one circuit and the BWR does not have a steam generator. Again, water is the moderator as well as the coolant. Like the PWR, the BWR operates at a higher pressure than the atmospheric pressure, but because the water must boil, the pressure is lower, at around 7MPa. The design for the BWR is a lot simpler and it may even be a safer and a more stable design than that of the PWR. For the rest, the BWR mostly operates in the same manner.

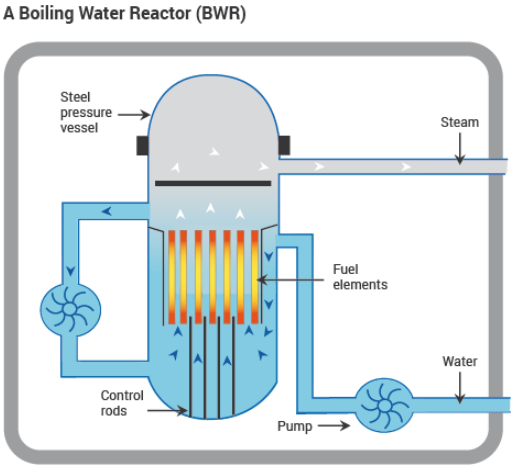


Figure 2.2: BWR

PHWR/CANDU

The Pressurized Heavy Water Reactor was developed in the 1950s in Canada where it is known as the CANDU (CANada Deuterium Uranium). Unlike both the PWR and BWR, this design uses heavy water as a coolant and moderator instead of light water. PHWRs usually use natural uranium (0.7% U-235) oxide as fuel, resulting in the need for a more efficient moderator; this is where heavy water comes in. Heavy water is water in which the hydrogen is ^{2}H instead of the more regularly occurring ^{1}H. A large tank called the calandria contains the reactor core and the heavy water as moderator. This tank is pierced by hundreds of horizontal pressure tubes which form channels for the fuel which is cooled by the heavy water under high pressure in the primary circuit, with temperatures reaching 290°C. The coolant water and the moderator water are kept separate. The coolant then flows to the steam generator, where the water in the secondary circuit is turned to steam that then powers turbines to produce energy.

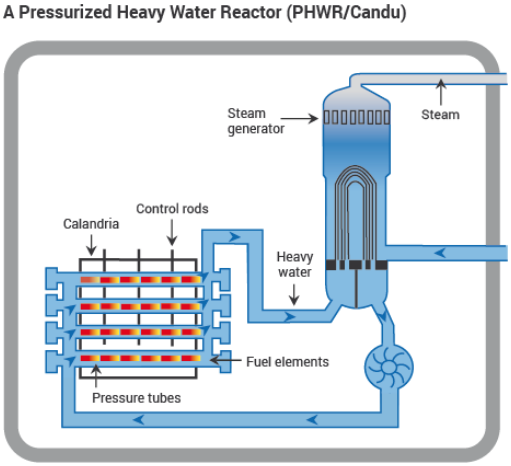


Figure 2.3: PHWR/CANDU

AGR

The Advanced Gas-cooled Reactor was designed by the British and uses graphite as the moderator and carbon dioxide as the coolant. For the fuel, uranium oxide pellets, enriched 2.5%-3.5%, are used. The carbon dioxide coolant first flows through the core, reaching temperatures of 650°C, then it goes past the steam generator. After that, the same process occurs as with every other design.[10][11]

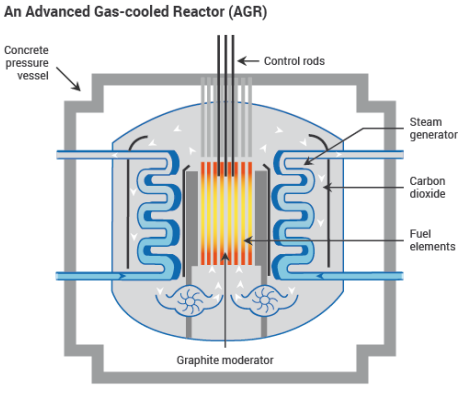


Figure 2.4: AGR

2.4: NRG (Petten)

Thanks to one of our counsellors' contacts, we were able to visit the Nuclear Research and Consultancy Group (NRG) in Petten, the biggest centre of nuclear research in the Netherlands, and Europe's largest producer of radionuclides.
While there, we met with Dr. Frodo Klaassen and Sander de Groot, two managers at NRG. We had already prepared some questions about the finer details of our experiment, including a few about protactinium separation and uranium fission products.
Our conversation can most definitely be described as fruitful, and their advice was used on multiple occasions whilst working out our experiment.
After the discussion we were given a small tour of the local facilities, which include their soon-to-be decommissioned high neutron flux research reactor. After running for over 50 years, plans are being made for it to be replaced by a new multi-purpose reactor, under the PALLAS foundation.

2.5: History of nuclear energy

Nuclear reactors were pioneered to produce fissile materials for use in nuclear weapons during the Manhattan project of World War 2.
After global nuclear disarmament gained traction to desire rose to utilise nuclear energy for power generation instead of weaponry.

Most of the experimental research in this field was done in Oak Ridge National Laboratory, Tennessee. After the very first artificial reactor in the form of the Chicago Pile-1, they made the first reactor designed for sustained operation, the X-10 Graphite Reactor. It used air cooling, graphite moderators and solid uranium oxide fuel. Subsequently they pioneered the light-water uranium-235 reactor. The 'light' refers to the type of water used: regular water, without heightened concentrations of deuterium (^{2}H).

The light-water reactor (LWR) continued to be built around the globe for power generation, mostly in the form of PWRs and BWRs, although many different reactor configurations were also used. To the researchers at Oak Ridge it became clear that they were too inefficient to be the way forward. As a next generation replacement there were two main possibilities: A 'fast breeder' reactor running on plutonium, or a liquid salt reactor running on thorium. Both of these concepts were highly experimental at the time.

The newly minted director of Oak Ridge, Alvin Weinberg, was a big proponent of the latter option. Between 1966 and 1969 an experiment successfully ran at Oak Ridge called the Molten Salt Reactor-Experiment, which proved the possibility of Molten Salt Reactors (MRSs).

In the late '60s, before this experiment could be given a follow-up, nuclear energy funding heavily decreased, and the task of developing the next generation of reactors was delegated to Argonne National Laboratory in Idaho, which had already put a large amount of effort into

plutonium fast breeders. Without direct funding, MSRs were mostly ignored in the following decades.
Argonne's plutonium fast breeder never truly took hold, as you can see today: the vast majority of currently running commercial reactors is still based on the original principle of a U-235 fed, water cooled, highly pressurized reactor.

However, nowadays public and scientific focus is shifting back to an old acquaintance...

Chapter 3: Thorium

Element number 90 of the periodic table, a soft silvery metal, was discovered in 1828 by a Swedish chemist and named thorium (symbol Th), after the Norse god of war and thunder. Thorium only has one commonly occurring isotope: Th-232. Although technically radioactive, its half-life of 14.05 billion years makes it relatively safe to handle and store in large quantities. It is notable that some of the products in its decay chain have shorter half-lives and release higher energy gamma rays, which could pose a problem for long-term storage.

In the years after thorium's discovery it remained without application, save for use in gas mantles in lamps. Only when humanity achieved nuclear fission did thorium get a place in the spotlight: scientist showed that thorium, while not fissile itself, has the unique ability to be bred into uranium-233, which is fissile. Thorium is what scientists call a fertile material.[6]

3.1: Thorium fuel cycle

The ideal thorium fuel cycle is a small, closed-loop cycle involving thorium-232 as the starting element. While Th-232 does not do much in an unmoderated (i.e. fast, high energy) neutron flux, if you slow down the neutrons to a thermal level, it gains an interesting property: a high neutron capture cross section.

1) If you bombard Th-232 with thermal neutrons, it will capture those neutrons with relative ease and turn into Th-233.

$$^{90}_{232}\mathrm{Th} + n \rightarrow {}^{90}_{233}\mathrm{Th}$$

2) Th-233 is highly unstable, with a half-life of 21.83 minutes. Through β^- decay, it turns into protactinium-233.

$$^{90}_{233}\mathrm{Th} \rightarrow {}^{91}_{233}\mathrm{Pa} + e^-$$

3) Pa-233 is still radioactive, but decays much more slowly, halving about every month. If you can avoid further neutron capture, protactinium-233 decays through β^- into the desired product: uranium-233.

$$^{91}_{233}\mathrm{Pa} \rightarrow {}^{92}_{233}\mathrm{U} + e^-$$

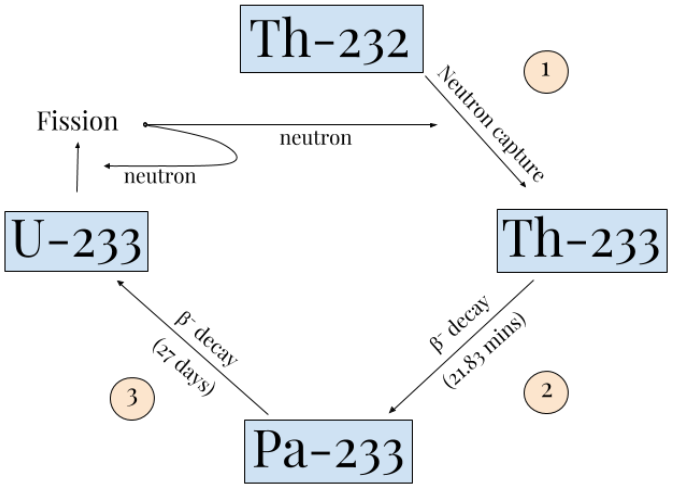


Figure 3.1: Visualisation of the thorium cycle

The uranium is then pumped into the neutron-producing core and fissioned, producing energy, fission products and yet more neutrons to continue the cycle. An ideal design would have the speed of this cycle balanced perfectly, and constantly pump in just enough thorium to keep it running, whilst syphoning off reactor products.[12]

Chapter 4: LFTR

The Liquid Fluoride Thorium Reactor is an experimental type of nuclear fission reactor. It is classified as a 'Molten Salt Reactor' (MSR), meaning that it operates with all of its contents solved in a molten salt. The LFTR is also often a breeder reactor, meaning that it produces more fissile material than it consumes (breeder reactor must not be confused with the breeding process, where thorium is transmutated to uranium, which takes place in every LFTR). No LFTR has been built yet, it is an entirely theoretical design.

In a LFTR, the molten salt is heated by the fission reaction, the salt then flows to a heat exchanger, where the heat is transferred to a gas, which then either goes to another heat exchanger to transfer that heat to yet another gas which then drives the turbines for power generation, or it goes directly to the turbines, with no second heat exchanger in between.

4.1: Advantages of the LFTR

LFTR proponents often cite a few theoretical major advantages that the LFTR has over any LWR and most other reactor designs.

LFTRs work with salts as the cooling fluids and as a result, do not need to operate at a high pressure, but instead run at atmospheric pressure (the high pressure is required in most reactors because the water would otherwise turn to steam). As a result, there is no need to worry about depressurisation.

In the case of power loss, most current reactors would start overheating because of the fission products in their core, which continue to generate decay heat from their own radioactivity. The overheating could cause a nuclear meltdown, where the core components turn hot enough to liquidize, spilling radioactive waste. In a LFTR, a so-called freeze plug would prevent this overheating.

Whereas it is said that uranium-powered reactors generate waste, the worst of which takes over 300,000 years to decay to background radiation levels, LFTRs longest lasting waste only takes 300 years to decay to background radiation levels. In terms of volume, LFTRs also generate one thirtieth of the waste that uranium-powered reactors generate. It is important to note that a new LFTR can be partially started with nuclear waste left over from current reactors, helping nuclear waste disposal.

Thorium-232 is more abundant than all uranium, let alone uranium-235, and due to its abundance, it costs significantly less. Also, a lot of thorium is already dug up because it is a by-product of the mining of rare-earth metals, and it is being stored in various places around the globe. Thorium also has a noticeably larger energy density than uranium.[21][22]

4.2: Disadvantages of the LFTR

The LFTR does have downsides, the most important of which is that it is a completely theoretical design. No full scale LFTR has ever been built, but from the plans, critics can already infer possible problems.

Corrosion could be an issue due to the aggressive molten salts that are contained in the core. Any LFTR would also take a significant amount of U-233 to start running, and U-233 is currently in extremely short supply.

The most proposed salt for the LFTR, FLiBe, is highly toxic, and as a result extra precautions must be taken and money must be spent on finding ways to safeguard people from the toxic element in FliBe, beryllium.

Another foreseeable problem is the unknown properties of the fission by-products. It is said that noble metals may eventually clog up the system, resulting in expensive repairs.[23][24]

4.3: Design

When talking about any physical concept, it helps to know what it actually looks like in real life. Especially for something as complicated as the LFTR, it was deemed necessary to show the different proposed designs for a LFTR. This knowledge is also needed when talking about the experiment and its results.

One thing all LFTRs have in common is that they have a core containing the carrier salt and some sort of moderator, like graphite rods, to keep the neutron energy at a desirable level; the neutron energy must be at thermal levels for the biggest cross section, which leads to the biggest chance for fission to occur.

Design Variations

There are three different designs for the LFTR: a single fluid reactor, a one and a half fluid design and a two fluid reactor, each with its own advantages and disadvantages.

Single fluid design

This type of reactor is made up of only one cooling fluid. It has one reactor vessel in which thorium as well as uranium are found within the same carrier salt.

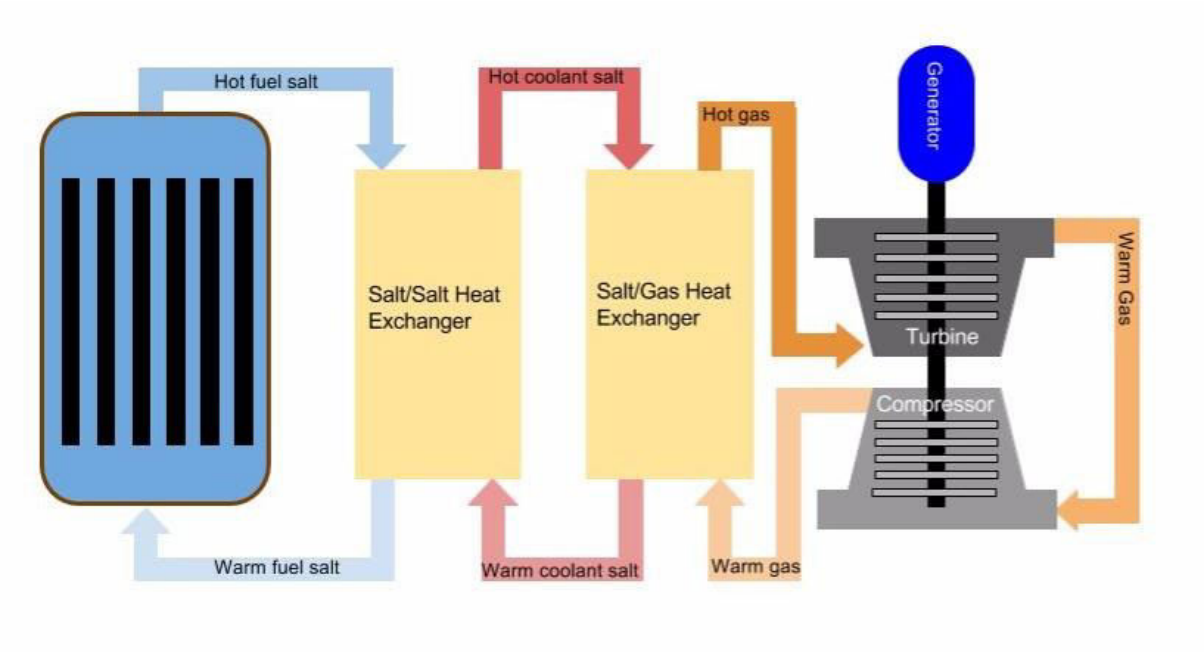


Figure 4.1: Heat transfer in a single fluid LFTR

One of the most obvious advantages of this design is its simplicity. With this design, it is also easier to use the experience gained from the MSRE, as both MSRs run at Oak Ridge National Laboratory were of the single fluid variety. Compared to the two fluid design, it has lower neutron losses in its carrier salt because it has only one salt.

Despite its advantages, it also has some noticeable disadvantages. The fission product processing, for example, is complicated by the presence of thorium because everything is put into one salt. Also, protactinium separation is almost always required because it has a large neutron cross section and it is not possible to increase the volume of the salt without having to increase the amount of fissile uranium-233. One major disadvantage is the positive temperature feedback coefficient, which is never good for a reactor. This means that when the temperature rises, more power will be delivered, which subsequently leads to a higher temperature, and so on, eventually leading to a meltdown if it is not controlled by an outside force. However, this may not be as bad as originally thought, as the positive feedback given by the graphite lags behind any increase of temperature of the salt by at least dozens of seconds, and in the future this may be solved.[15][16]

Two fluid design

This design is more complex than the single fluid design and is composed of two compartments with separate salts. The two fluid design is also often referred to as the 'standard' design nowadays; this is the design most LFTR proponents refer to by default. The fuel salt is in the core, which contains the fissile uranium-233 produced by the thorium in the so-called 'blanket' around the core. The core has a high neutron flux and some of the neutrons from the core escape into the surrounding blanket, where the thorium-232 absorbs the neutrons to become thorium-233 and then protactinium-233. After that, the protactinium-233 is either left in a region of the blanket where the neutron-flux is lower or it is taken out of the blanket entirely, to minimise the formation of hazardous nuclides.

When the protactinium has sufficiently decayed, the uranium-233 is separated by a process known as fluorination: A portion of the blanket salt is pumped out and exposed to fluorine gas, which binds to uranium-233 that is already bound to fluorine as UF_4 (tetrafluoride) in the mixture to form UF_6 (hexafluoride). Hexafluoride is gaseous at the temperature of the salt, so it bubbles out of the mixture, and rises to the top, where it can easily be skimmed off. After being separated, the hexafluoride is exposed to hydrogen gas, which steals back two of the fluorine atoms (defluorination), turning it back into liquid tetrafluoride. This UF_4 is fit to be pumped into the core to fission. The now uranium-free blanket salt can be pumped back into the blanket for further breeding.

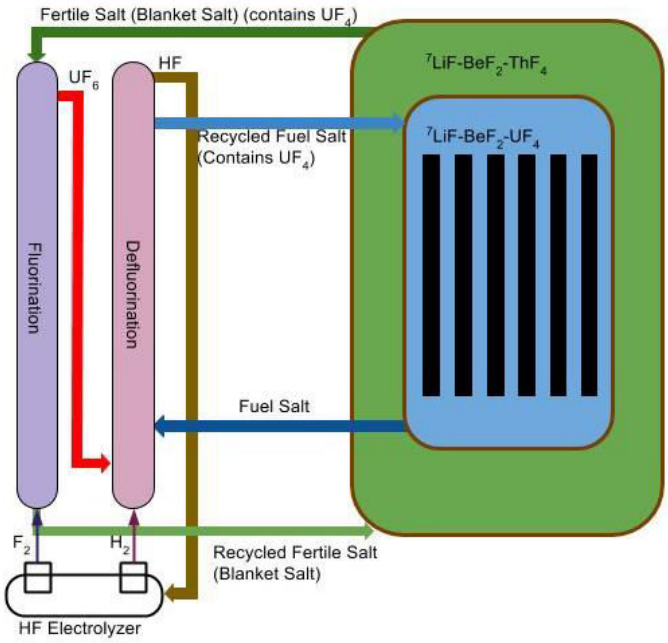


Figure 4.2: Uranium-233 separation in a two-fluid LFTR

Oak Ridge National Laboratory originally made a two fluid reactor, but they soon switched to the single fluid design, after encountering what is known as 'The Plumbing Problem'. This refers to the design, where the blanket salt does not only surround the core, but is also interlaced with the fuel salt within the core region, which is difficult to achieve.[17][19]

Whereas the single fluid design has some neutron leakage, the two fluid design has a neutron leakage near zero due to the blanket surrounding the core, which catches the neutrons that escape the core. Also the fission product processing is not nearly as complicated because there is no thorium in the fuel salt. To top it off, the removal of protactinium is not necessary in this design: increasing the volume of the blanket salt is enough to counteract the effect of the protactinium, as the neutron flux can be lower in certain areas when the volume of the blanket salt is bigger, making the chance of the protactinium absorbing a neutron smaller.

Just like the single fluid design, this design has a positive temperature feedback coefficient, but here it is only present in the salt blanket and not in the fuel salt, the fuel salt has a negative temperature feedback coefficient. Because the two fluid design has two separate fluids, it needs an extra heat transfer loop. Another disadvantage is the aforementioned Plumbing Problem.[15]

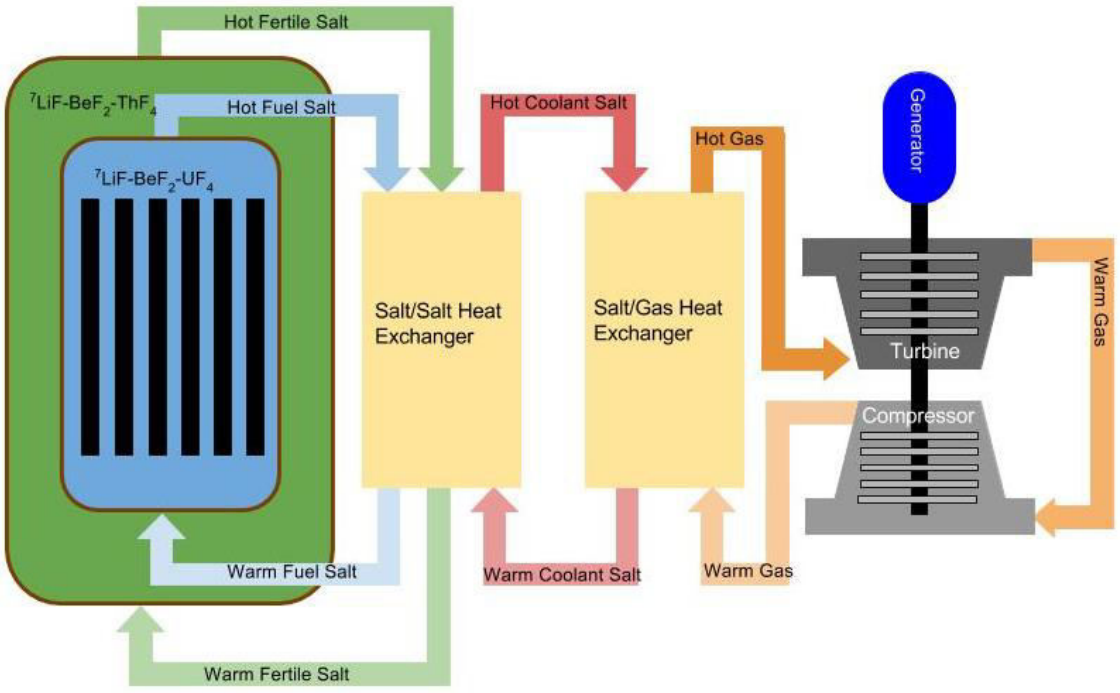


Figure 4.3: Heat transfer in a 2-salt LFTR

One and a half fluid design

In an attempt to fix the positive temperature coefficient in the single fluid design, French researchers proposed a design with a blanket around the core, resulting in a design now known as the one and a half fluid design. This is a reactor with the advantages and disadvantages of both aforementioned reactor designs. The reactor is made up of a core and a blanket surrounding it, however, the core not only contains the fissile uranium, but it also contains the fertile thorium, just like in a single fluid design, and the blanket only contains the fertile thorium, like the two fluid design.

This design indeed fixes the positive temperature coefficient and it is made easier to have a negative temperature coefficient, which is wanted in a reactor. The design still stays simple, despite its blanket, and because of its blanket the neutron leakage is lower compared to the single fluid design.

The one and a half fluid reactor does keep the same complicated fuel processing for the core salt as the single fluid design though, and it needs an extra heat transfer loop for the blanket salt like the two fluid design.

Freeze plug

An element in each design, that needs to be acknowledged separately, is the freeze plug. This feature ensures safety by draining the fuel into a separate tank where no fission is possible when something goes wrong. The plug is made of a solid salt, situated at the bottom of the core, blocking a pipe leading to a drain tank, which is kept under its freezing point by a fan. The fan, of course, needs energy to keep going. This means that when, for whatever reason, the reactor is cut off from its power source, or the temperature in the core rises to such high temperatures (this is unwanted) that the cooling effect from the fan is counteracted and the temperature around the freeze plug rises to above its freezing point, the salt melts and the fuel drains into a separate tank, keeping the fuel from fissioning. The neutron flux is not regulated anymore and will die off, and the radioactive material is kept safe.

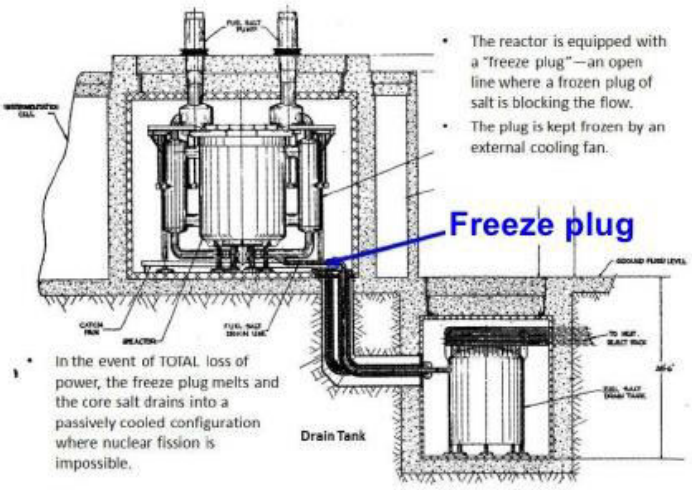


Figure 4.4: Freeze plug

The reactor is made to retain heat, which is desirable under normal circumstances, but during a meltdown the most important thing is to lose heat. The drain tank is built to keep as little heat as possible; it can be seen as a giant heatsink, having the characteristics engineers want most during a meltdown. The fact that the core retains heat, even during a meltdown proposes major engineering challenges in most current reactors. [18]

Engineers could also purposely cut off the power to the fan, draining the fuel, and in that way shut down the reactor when it is not needed. What makes this such a great safety feature is that power is needed to prevent the reactor from shutting down, instead of the other way around, where power is needed to shut the reactor down.

4.4: The Salt

A salt is an ionic compound made up of two (groups of) oppositely charged ions: the cation, which is positively charged and the anion, which is negatively charged.

When salts are used in a reactor, a salt mixture, such as $^{7}LiF\text{-}BeF_2$ and $NaF\text{-}BeF_2$, is used. This is done because the individual melting point of each salt is too high to be useful in a reactor to act as a coolant, but when the salts are combined, the resulting melting point is lower, which leads to it being a practical coolant. $^{7}LiF\text{-}BeF_2$ has a melting point between 360°C and 460°C, while the individual melting points are at 845°C for ^{7}LiF and 1169°C for BeF_2.

In all MSRs, the salt containing the fuel flows through the core, where fission occurs, it then flows through the primary heat exchangers, where it passes the heat on to a secondary coolant, which is also a salt. This makes up the primary loop, and it is why the salt is both the fuel and the coolant in a MSR.

In a LFTR, assuming it is a breeder reactor in the thermal spectrum, which most are, the biggest candidate for the carrier salt is the $^{7}LiF\text{-}BeF_2$ salt, colloquially known as FLiBe. For higher energy spectrums, others salts are taken into consideration such as $NaF\text{-}BeF_2$, with a much lower melting point.

In a single fluid design, the composition of the salt used is $^{7}LiF\text{-}BeF_2\text{-}ThF_4\text{-}UF_4$, with ratios of 72-16-12-0.4 (in approximate %mol).
In a two fluid design, the fuel salt is $^{7}LiF\text{-}BeF_2\text{-}^{233}UF_4$, where the ratios of ^{7}LiF and BeF_2 are around 67-33, and the blanket salt is $^{7}LiF\text{-}BeF_2\text{-}ThF_4$. [15]

4.5: Protactinium separation

Negatives of exposing protactinium to neutrons

Protactinium-233, a key part of the thorium fuel cycle, decays to uranium-233. What worries some LFTR proponents is the fact that it does so over the course of nearly a month, and in that time it has the capability to capture neutrons (with a cross section of 42.52 barns), turning it into Pa-234, and knocking it out of the thorium cycle entirely.

Not only is every accidentally created atom of Pa-234 a waste of multiple, otherwise useful, neutrons, it can lead to a chain of capture and decay reactions that culminates in the formation of transuranics, elements with more than 92 protons. All transuranics are radioactively unstable, and many have half-lives that make them prime candidates for

nuclear waste: long enough not to totally decay while still in the reactor, but short enough to emit enough radiation to be hazardous.
However, transuranics are only formed if the protactinium is exposed to neutrons in the first place. If one could minimise this exposure, fewer Pa-233 atoms would capture neutrons, meaning fewer wasted neutrons, and fewer transuranics.

Solution
The proposed solution in some of the more adventurous LFTR designs is protactinium separation. With this method, protactinium is pulled out of the salt as soon as possible after formation from thorium, and left to decay peacefully outside the neutron flux field.

One of the ways to do this would be by extending the system that separates U-233 from protactinium. Using liquid bismuth and lithium, the other components of the mixture can be removed, such as fission products.

Obstacles for protactinium separation

While a great concept at first glance, protactinium separation does pose some problems:
Firstly, the rate at which this filtration would take place may not be nearly enough to have a significant impact on Pa-234 production.
Secondly, a rather worrying implication of Pa-233 separation is that one could take the protactinium outside of the reactor entirely, let it decay and be left with ultra-high grade fissile uranium-233, at a concentration of over 99%. To clarify, any U-233 mixture more concentrated than 12% is considered weapons grade, meaning that protactinium separation technically has the capability to produce nuclear weapons.
As a counterargument, it is important to consider the logistics of this hypothetical weapons production: the mixture would also contain uranium-232, which emits such strong gamma rays that it is virtually impossible to handle and very easy to detect from a distance.[13][20]

Chapter 5: Experiment

Before the question “What are the characteristics of the products of the thorium cycle in a LFTR?” can be answered, the formed nuclides need to be determined, as well as the ratio in which they are formed. The experiment focuses on the calculation that determines both the nuclides and their ratios.

5.1: The Experiment

Goal

The objective of the experiment was to simulate a thorium breeder reactor over a long time span and with reasonable accuracy. The simulated reactor is a typical thorium breeder in the style of the LFTR. It is a two-salt design, is filled with FLiBe carrier salts, and has no protactinium separation. We run this reactor over a 10-year time period, with a neutron flux of 10^{10}, and a neutron energy of 0.0253 eV (thermal).

Assumptions

First of all, some assumptions were made. These would make the complexity of the simulation manageable:

- A fixed neutron flux: this assumes that the reactor is already running at the start of the simulation, and that it remains that way throughout.
- No fuel addition or waste removal, which is not realistic for a commercial reactor.
- No neutron poison simulation: due to our very basic ‘neutron economy’, there is no room to simulate the effects of some nuclides that in reality ‘consume’ neutrons. This assumption should not have a major impact on our results: the objective of the experiment is to find the ratios, so the production rate, which is influenced by the neutron flux, is not relevant.
- Calculation in large steps: instead of continuous calculation, we use ‘cycles’. Between each cycle is a time step of 1 day. Possibly, this influenced the results in terms of short lived nuclides.
- A fixed reactor volume. The simulation starts with approximately one cubic centimetre of thorium-232, and all other reactions are presumed to take place in the same space.

5.2: Formulae

When running nuclear fission calculations, two formulae are used the most. These formulae are provided here, but they are also mentioned separately when discussing the program in depth.

Reaction rate (5.1)[28]

To calculate how many particles of a starting number interact with neutrons over a given time period.

$$RR = \phi \cdot N \cdot \sigma$$

Where:
RR= the reaction rate per unit volume
Φ = neutron flux (neutrons * cm^{-2} * s^{-1})
N = atomic number density (atoms*cm^{-3})
σ = microscopic cross section (cm^2)

Decayed nuclei (5.2)

How many nuclei with a given half-life decay over a given time period?

$$D = S \cdot 0.5^{\frac{\Delta t}{t_{\frac{1}{2}}}}$$

Where:
D = number of decayed nuclei
S = number of starting nuclei
Δt = elapsed time (s)
$t_{1/2}$ = half-life (s)

5.3: Data

Neutron cross sections

An international scientific community has built up an array of libraries of nuclear knowledge. The Joint Evaluated Fission and Fusion File (JEFF), version 3.2 was chosen for the calculations. With associated software known as JANIS, or simply with the Nuclear Energy Agency's website, anyone has access to direct nuclear cross sections. Particularly interesting in this case were the neutron incident data tables on fission, the process of which is noted as (z, f), and on neutron capture, noted as (z, ɣ).

Nuclide half-lives and decay modes

With a combination of manual entry and Python-based automation we imported decay data for all the relevant nuclides: the group around Th-232, the transuranics and the area covering all fission products.

U-233 fission products

To calculate the fission products, the program uses a 'fission table'. This specifies, on average, which nuclides are formed from a single U-233 fission. The table consists of a long list of values between 1 and 0 (typically the value is around 0.001), where each value is assigned to a specific nuclide, stating what fraction of the fissioned U-233 goes to that nuclide.

A fission table with the nuclides that are directly formed from fission is called an "independent" fission table. The values in a "cumulative" table are not independent, any particular value in a cumulative table is the sum of all the independent values that precede it in the decay chain. This makes them better for analysing the eventual products, but virtually useless for our purposes.

A problem arises in finding a good source for a full independent U-233 fission table. This is because most researchers use software that directly calculates fission tables from databases like JEFF, yet we do not have access to this kind of software.
However, there are plenty of cumulative tables to be found online, so after getting some pointers from Dr. Klaassen at NRG, we can calculate the independent fission values from the cumulative ones. The code that was written to do so is in appendix II.

To form an independent fission table, as shown in table 5.1, start with all the cumulative values for nuclides with the same mass, as these form a β^--decay chain in our case. Beginning with the nuclide that is first in the chain, calculate each independent value by taking the current cumulative value minus the previous cumulative value.

Table 5.1: sample calculation of independent fission values

Nuclide	Cumulative	Calculation	Independent
Strontium-100	0.000029	0.000029-0	0.000029
Yttrium-100	0.0010304	0.0010304-0.000029	0.0010014
Zircon-100	0.029699	0.029699-0.0010304	0.0286686
Niobium-100	0.032267	0.032267-0.029699	0.002568
Molybdenum-100	0.044117	0.044117-0.032267	0.01185

Resulting data tables

The data collection specified above results in two text files that the program uses for its calculations:

- A data file, which specifies decay rate and mode, as well as cross sections where necessary. A sample line of this file appears as follows:

30 80 D:B-:0.545

  Where the first value represents the atomic number and the second value the mass. The third group shows that this nuclide **D**ecays via **B-** with a half-life of **0.545** seconds.

- A fission file, which specifies the independent fission yields of U-233. This looks like:

30 80 1.0083e-06

  Where the first value represents the atomic number and the second value the mass. The third group dicates that, on average, every fissioned U-233 atom should result in the formation of **1.0083 *10^{-6}** nuclei with these Z and A values.

5.4: Manual calculation

Now that we have formulae and data, we can use them in a manual calculation. For any nuclide you can easily calculate by hand what would happen to it over a given period of time, if you specify a few factors.
For example, this is a possible calculation:

What happens to 3.036 * 10^{22} atoms of Th-232 (about a cubic centimeter) when exposed to a fixed thermal neutron flux of 10^{19} over 1 second?

Given:
3.036 * 10^{22} atoms of Th-232
10^{19} neutrons cm^{-2} s^{-1}
1 second elapsed

From JEFF-3.2 we find that the thermal neutron absorption cross section for this material is 7.337182 * 10^{-24} cm^2, and that no other cross sections are significant at this energy level.

Since we are dealing with neutron interaction, we use the appropriate formula:

$$RR = \phi \cdot N \cdot \sigma$$

$$RR = 10^{19} \cdot 3.036 \cdot 10^{22} \cdot 7.337182 \cdot 10^{-24}$$

$$RR = 2.228 \cdot 10^{18}$$

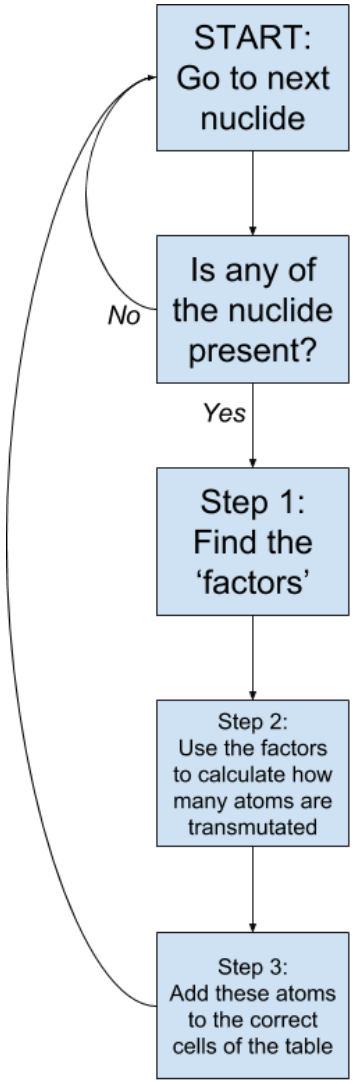
Over a period of 1 second, this means:
$2.228 \cdot 10^{18}$ atoms capture a neutron

5.5: Automation via programming (in Python)

To allow for a large amount of calculations over a long simulated time span, an alternative to manual calculation was needed, and running the formulae using Python-3.4, an object oriented programming language, was chosen as a solution. In this section we first broadly explain how the program functions, and then go through the actual code to work out the details.

Code logic

The code functions in 'cycles', where each cycle represents some elapsed time, a day in our case. The code goes through all possible nuclides to calculate what each of them does in that cycle. This single-nuclide logic has been put into a flowchart below.



Flowchart with commentary

Step 1:
Each reaction mechanism of a nuclide (decay, neutron capture or fission) has an associated 'factor' that is used by the code. This 'factor', a value between 0 and 1, represents the fraction of the starting nuclei that does not go to that reaction mechanism. E.G. if a quarter of a substance decays over some period of time, the code says there is a decay 'factor' of 0.75.
This factor is calculated using either the decay or the neutron interaction formulae, depending on the reaction mechanism.

Step 2:
If the nuclide has one reaction mechanism, it is simple to convert the factor to an actual number of atoms.
If multiple reaction mechanisms are in play, a correct distribution is required. That distribution is achieved using two formulae, which are covered below.

Step 3:
Now the code knows how many atoms are left untouched, and how many are not. These values are put into the appropriate cells of the table.

Figure 5.1: Code flowchart

Multiple reaction mechanisms

One of the most complicated matters the code has to deal with comes in the form of multiple reaction mechanisms. Even though each of these mechanisms already has a 'factor' (the fraction that does not go to that mechanism) assigned, the calculation is not done.

These factors are calculated as if only one reaction mechanism is in play, and using them directly in the case of multiple mechanisms is incorrect. Instead, two formulae were developed by us to achieve a correct distribution:

First, use this formula to calculate how many nuclei are untouched.

Unchanged nuclei after set time period (5.3)

$$U = S \cdot \prod_{i=1}^{n} f_i$$

Where:

U = number of unchanged nuclei S = number of starting nuclei

f_1 to f_n = list of factors over given timespan n = number of factors

This can be put more simply as:

$$U = S \cdot P$$

Where:

U = number of unchanged nuclei S = number of starting nuclei

p = product of factors

Use the unchanged nuclei to calculate how many do mutate, and then distribute them through a ratio. The division below results in a fraction that is multiplied by the total number of changed nuclei.

Number of nuclei affected by specific reaction mechanism (5.4)

$$M = (S - U) \cdot \frac{1 - f}{n - s}$$

Where:

M = number of changed nuclei (for specific reaction mechanism)

S = number of starting nuclei

U = number of unchanged nuclei

s = sum of all factors

n = number of factors

f = single factor (for specific reaction mechanism)

After these formulae it is known how many atoms go to each reaction mechanism, and these values can be used accordingly. We worked out these formulae ourselves, but it is likely they already exist in some more complicated form we did not find.

Sample calculation

An example of the code logic in action can be found here. These are the steps that the code would go through when calculating what happens to a single nuclide. In this case, we examine uranium-233 in the middle of the simulation, where it has already been formed.

Nuclide to examine: Z=90, A=233 (uranium-233)
Number of nuclei present: $2.418 * 10^{17}$
Time-step: 1 day (86400 seconds)
Neutron flux: 10^{10}

Step 1: Find the factors

The U-233 data table entry looks like:
"92 233 D:B-:5023862600000;N:C:45.25638;N:F:531.3016"
From this we gather it has three reaction mechanisms: decay (β^-), neutron capture and fission. We start by separately calculating the factors for each of these mechanisms:

$$f_{decay,\beta^-} = 0.5^{\frac{\Delta t}{t_{\frac{1}{2}}}} = 0.5^{\frac{86400}{5.02e12}} = 0.99999998807$$

$$f_n = 1 - (\Delta t \cdot \Phi \cdot \sigma)$$

$$f_{n,capture} = 1 - (86400 \cdot 10^{10} \cdot 45.25638 \cdot 10^{-24}) = 0.99999996089$$

$$f_{n,fission} = 1 - (86400 \cdot 10^{10} \cdot 531.3016 \cdot 10^{-24}) = 0.99999954095$$

Step 2: Distribute the atoms

We then use the factors to determine how many U-233 nuclei are untouched.

$$U = S \cdot \prod_{i=1}^{n} f_i$$

$$U = 2.418 * 10^{17} * 0.99999998807 * 0.99999996089 * 0.99999954095 = 2.4179988 * 10^{17}$$

Finally, we use the second formula in conjunction with this value to distribute the nuclei amongst the various reaction mechanisms.

$$M = (S - U) \cdot \frac{1 - f}{n - s}$$

$$M_{decay,\beta^-} = (2.418 * 10^{17} - 2.4179988 * 10^{17}) \cdot \frac{1 - 0.99999998807}{3 - 2.99999948991}$$

$$M_{decay,\beta^-} = 2.80656354 \cdot 10^{9}$$

$$M_{n,capture} = 9.20073 \cdot 10^{9}$$

$$M_{n,fission} = 1.0799 \cdot 10^{11}$$

Full code with commentary

What follows is a step-by-step analysis of the code, divided into blocks. Each code block is followed by a detailed explanation. Any code preceded by a '#' is a comment, meaning it is not run by Python, only being there to improve readability. The entire code is working towards simulating a LFTR, and outputs its results as multiple text files.

```
import math
import numpy

# Filling data from text files
data = [] # the list that contains all nuclear properties, from data.txt
empty = [] # an empty list that is used as a template
for i in range(119):
    data.append(300*[""])
    empty.append(300*[0])
for i in open("data.txt").readlines():
    i = i.split()
    data[int(i[0])][int(i[1])]=" ".join(i[2:])
fission = [x.strip("\n") for x in open("fission.txt").readlines()]
oldnumbers = empty[:]
newnumbers = empty[:]
```

The program prepares for the simulation by importing two modules that will be used for calculation, and getting data from the text files. These files describe basic behaviour like half-life, decay mode and cross sections. It reads this data into its own memory beforehand so that it will not have to in the middle of a cycle, which greatly improves speed.
It also sets up the tables that will contain the numbers of all nuclides in the reactor.

```
# Constants & settings
nflux = math.pow(10,10) # neutron flux, in neutrons per cm^2 per second
ts = 86400 # time step, in seconds
mf = 1 # multiplication factor (speed up)
limit = 3650 # set to False if limitless
newnumbers[90][232] = 3.036*(10**22) # approx one cubic cm of Th-232
```

Here the program sets a few constants that will be used throughout the program. If desired, we can change these before running to simulate different reactor conditions. In the experiment, the program was set to calculate one day at a time (86400 seconds), running a total of 3650 loops, resulting in a simulation of ten years.
We also manually set one value in the 'numbers' table: the number of thorium-232 atoms. This value represents the fuel we would put in before exposing the reactor vessel to a neutron flux.
In our case we used approximately a cubic centimeter, and all other calculations are assumed to take place in that same volume.

```
#CALCULATING
cycles = 0
while cycles <= limit:
```

Here we start the looping part of the program, so everything after this is repeated in every cycle.

```
cycles += 1
oldnumbers = newnumbers[:]
newnumbers = empty[:]
writefile = open("output/c"+str(cycles)+".txt",'w')
for i in range(len(oldnumbers)):
    for j in range(len(oldnumbers[i])):
        if oldnumbers[i][j] != 0:
            writefile.write(str(i)+" "+str(j)+
            " "+str(oldnumbers[i][j])+"\n")
writefile.close()
```

The start of each loop contains some Python 'housekeeping': some tables are shifted around, the cycles counter is increased by one and the program writes the current contents of the reactor to an output file.

```
for i in range(len(oldnumbers)):
    for j in range(len(oldnumbers[i])):
```

After housekeeping, the program can start calculating. These are two smaller loops in one another: the outer loop goes through all the columns in the 'numbers' table, meaning all the elements, while the inner loop goes through all the cells in that column, meaning all the isotopes of that element.
All the following code is run in these two loops, which are inside the main loop, so for every single possible nuclide in every single cycle. With this amount of looping required, it is easy to see why automation is necessary.

```
if oldnumbers[i][j] != 0:
```

Simply check if there is any of the current nuclide present, otherwise there is no need to run the formulae. If there does happen to be some amount of the nuclide, however:

```
start = int(oldnumbers[i][j])
factors = []
# list of factors to multiply the start value with (note: factors are
```

```
calculated individually, as if there are no other reaction mechanisms)
for q in data[i][j].split(";"):
```

We take the current amount of the nuclide we are investigating and stick it in a variable for easy usage. Then we search the 'data' table to find out what its behaviour is.

In the next two blocks we fill the 'factors' list with the different factors, one for each reaction mechanism. These factors are required for the calculation later.

```
if q.split(":")[0] == "N":
# if reaction mechanism is neutron interaction -> use neutron interaction
formula
factors.append(float(1-
(ts*mf*nflux*float(q.split(":")[2].split()[0])*(10**-24))))
# 1-the fraction of start that mutates = fraction that does NOT mutate
```

If it interacts with neutrons, we use the formula for neutron interaction: a product of the time elapsed, the neutron flux, the cross section and the number of starting atoms.

Reaction rate (5.1)

To calculate how many particles of a starting number interact with neutrons over a given time period.

$$RR = \phi \cdot N \cdot \sigma$$

Where:
Φ = neutron flux (neutrons * cm^{-2} * s^{-1})
N = atomic number density (atoms.cm^{-3})
σ = microscopic cross section (cm^2)

The result of this formula is the fraction that *does* interact with neutrons, but for the eventual calculations we need the fraction that *does not*, which is why the result is prefixed by '1-'

```
if q.split(":")[0] == "D":
# if reaction mechanism is natural decay -> use decay rate formula
factors.append(float(0.5**((ts*mf)/float(q.split(":")[2].split()[0]))))
```

If the nuclide decays, we use the formula for decay: ½ to the power of the elapsed time divided by the nuclide's half-life.

Decayed nuclei (5.2)
To calculate how many nuclei with a given half-life decay over a given time period.

$$D = S \cdot 0.5^{\frac{\Delta t}{t_{\frac{1}{2}}}}$$

Where:
D = decayed nuclei
S = starting nuclei
Δt = elapsed time (s)
$t_{1/2}$ = half-life (s)

This factor already represents the fraction that *does not* mutate, so no further editing is necessary.

If the nuclide only has one way of mutating (it only decays, for example), then the list 'factors' will only have one factor in it. However, if it has two or three (decay, neutron capture and fission when being struck by a neutron, for example), then the list 'factors' will contain multiple values. Now we use these values to calculate what actually happens in terms of the numbers of nuclei.

```
end = start*numpy.product(factors)
```

We already know that the number of unchanged atoms can be determined with the factors: we use a self-developed formula for this. This logic is explained in more detail in the "Code logic" section above.

Unchanged nuclei after set time period (5.3)

$$U = S \cdot \prod_{i=1}^{n} f_i$$

Where:
U = number of unchanged nuclei
S = number of starting nuclei
f_1 to f_n = list of factors over given timespan
n = number of factors

Which can be put more simply as:

$$U = S \cdot P$$

Where:
U = number of unchanged nuclei
S = number of starting nuclei
p = product of factors

```
newnumbers[i][j] = int(end)
```

With this knowledge we start by filling the table of new numbers with the calculated number.

Up next: the atoms that *do* mutate, in what ratios do they do so?

```
dval = []
if len(data[i][j].split(";")) == 1:
    dval.append(start-end)
```

If there is only one reaction mechanism, the question is easy to answer: they all go to that mechanism.

```
else:
    for p in range(len(data[i][j].split(";"))):
        dval.append(
        (start-end) * ((1-factors[p]) / (len(factors)-sum(factors)) )
```

If there are multiple types, we need to correctly distribute the number of changed atoms using another self-developed formula

Number of nuclei affected by specific reaction mechanism (5.4)

$$M = (S - U) \cdot \frac{1 - f}{n - s}$$

Where:
M = number of changed nuclei (for specific reaction mechanism)
S = number of starting nuclei
U = number of unchanged nuclei
s = sum of all factors
n = number of factors
f = single factor (for specific reaction mechanism)

Now we have a list of the exact numbers of changed atoms, per reaction mechanism. All that is left to do is to add these numbers to the correct cells in the table.

```
for q in data[i][j].split(";"):
    indexnum = data[i][j].split(";").index(q)
```

We start going through all the reaction mechanism for the nuclide again, now with knowledge of the numbers of changed atoms.

```
if q.split(":")[0] == "N":
    if q.split(":")[1] == "C":
        newnumbers[i][j+1] += dval[indexnum]
```

If the reaction mechanism is neutron interaction, specifically neutron capture, we add the calculated number to the cell in the same column (same element) but one cell further right (one heavier isotope)

```
if q.split(":")[1] == "F":
    for f in fission:
        newnumbers[int(f.split()[0])][int(f.split()[1])] +=
        int(float(f.split()[2]) * dval[indexnum])
```

If the reaction mechanism is neutron interaction, specifically fission, we use the fission data text file. That text file contains a list of nuclides, each of which has a value that shows (on average) how many atoms of this nuclide are formed by one fission (this number is usually somewhere in the order of 0.001). We multiply the number of changed atoms by each value in the fission file, and add the results to the correct cells in the numbers table.

```
if q.split(":")[0] == "D":
    if len(q.split(":"))<4:
    # if True, singular decay, so no further calculation needed
        if q.split(":")[1] == "B-": newnumbers[i+1][j] += dval[indexnum]
        if q.split(":")[1] == "B+": newnumbers[i-1][j] += dval[indexnum]
        if q.split(":")[1] == "EC": newnumbers[i-1][j] += dval[indexnum]
        if q.split(":")[1] == "A": newnumbers[i-2][j-4] += dval[indexnum]
```

If the reaction mechanism is decay, we go through the different kinds of decay. For B- (electron emission), we go one column further, but stay on the same row (one element up, same mass number).
For B+ (positron emission) we go one column back, and stay on the same row (one element down, same mass number).
Finally, for A (alpha emission), we go two columns back and four rows back (two elements down, four mass numbers down).

To finish, there is the complicated matter of multiple decay. Sometimes, a nuclide does not have a single way of decaying, and when it does decay, there is some chance it goes one way, and some chance it goes the other way. Most commonly multiple decay happens as a choice between regular B- emission on the one hand and B- emission plus the emission of an electron on the other.
This final problem is dealt with using the following code: essentially a repeat of the earlier neutron interaction code, but with more built in exceptions, one for each multiple decay type.

```
elif len(q.split(":"))>3:
# if True, multiple decay, use given factor to separate dval
    if q.split(":")[1] == "B-":
        if q.split(":")[3].split()[0] == "B-N":
        # if both True, B- decay and B- with neutron emission
            newnumbers[i+1][j] +=
            int(dval[indexnum]*(1-float(q.split(":")[3].split()[1])))
            newnumbers[i+1][j-1] +=
            int(dval[indexnum]*(float(q.split(":")[3].split()[1])))
        if q.split(":")[3].split()[0] == "EC":
         # if both True, B- decay and electron capture
            newnumbers[i+1][j] +=
            int(dval[indexnum]*(1-float(q.split(":")[3].split()[1])))
            newnumbers[i-1][j] +=
            int(dval[indexnum]*(float(q.split(":")[3].split()[1])))
    if q.split(":")[1] == "B+":
        if q.split(":")[3].split()[0] == "A":
        # if both True, B+ decay and alpha decay
            newnumbers[i-1][j] +=
            int(dval[indexnum]*(1-float(q.split(":")[3].split()[1])))
            newnumbers[i-2][j-4] +=
            int(dval[indexnum]*(float(q.split(":")[3].split()[1])))
```

This is where the looping code ends. The program repeats this cycle of calculation and table shifting for every isotope of every element, and then restarts into a new loop. It loops until reaching the given limit.
In the total runtime there are approximately $1.3 * 10^8$ checks to perform for all the isotopes, which Python manages to clear in just under two minutes.

Chapter 6: Results, Analysis and Discussion

Using the results taken from the experiment, both the nuclides that are formed as well as their ratios are known. With this knowledge one can determine the most relevant nuclides present in the reactor, in terms of both danger and usability.

6.1: Raw data

Due to the length of the program's output it is not here in its entirety, but a sample can be seen here.

```
52 131 722956095
52 132 34186617904
52 133 2442357144
52 134 5361312656
52 135 1994348129
52 136 1051414544
52 137 139571374
52 138 19167196
52 139 1484507
52 140 107568
53 127 2550316926234
53 129 4472272040092
53 130 6271056
53 131 60548526775
53 132 297951041
53 133 12669733561
```

This output consists of a series of lines, where each line represents one nuclide. The first value is the number of protons, the second is the atomic weight. The third value is the calculated number of atoms of the nuclide.

Appendix III consists of this file, sorted by the number of atoms.

6.2: Visualisation and interpretation of data

Mass distribution of simulated LFTR contents

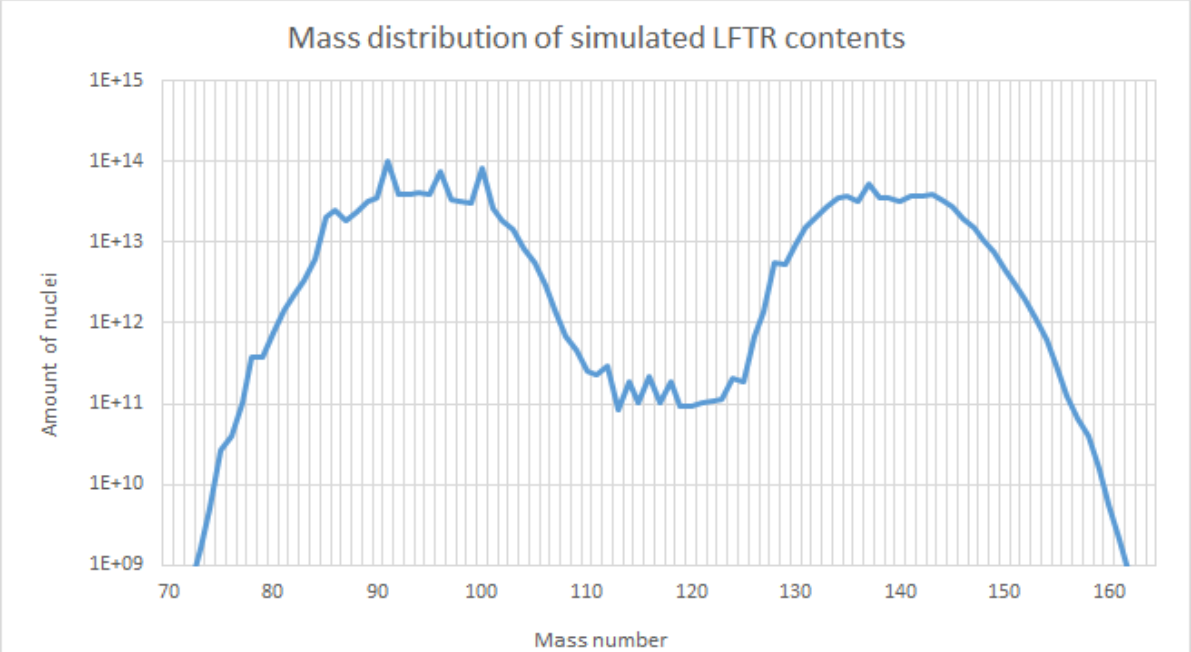


Figure 6.1: Mass distribution of simulated LFTR contents

This graph depicts the sum of all nuclides with the same mass number (known as isobars). The two peaks show how a fissile material mostly splits into a heavier and lighter particle, instead of two particles of the same size. Below is a real life visualisation made by the JAEA.

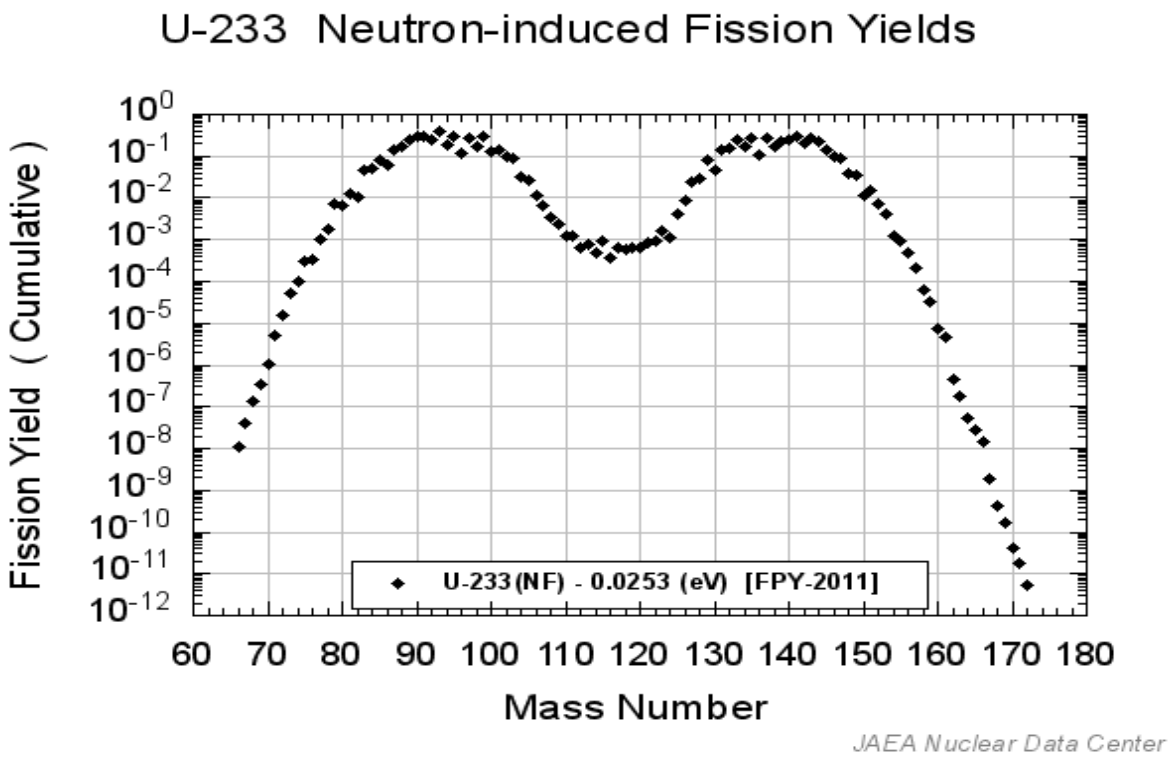


Figure 6.2: U-233 Neutron induced fission yields (JAEA)

Two-dimensional graph (chart of nuclides)

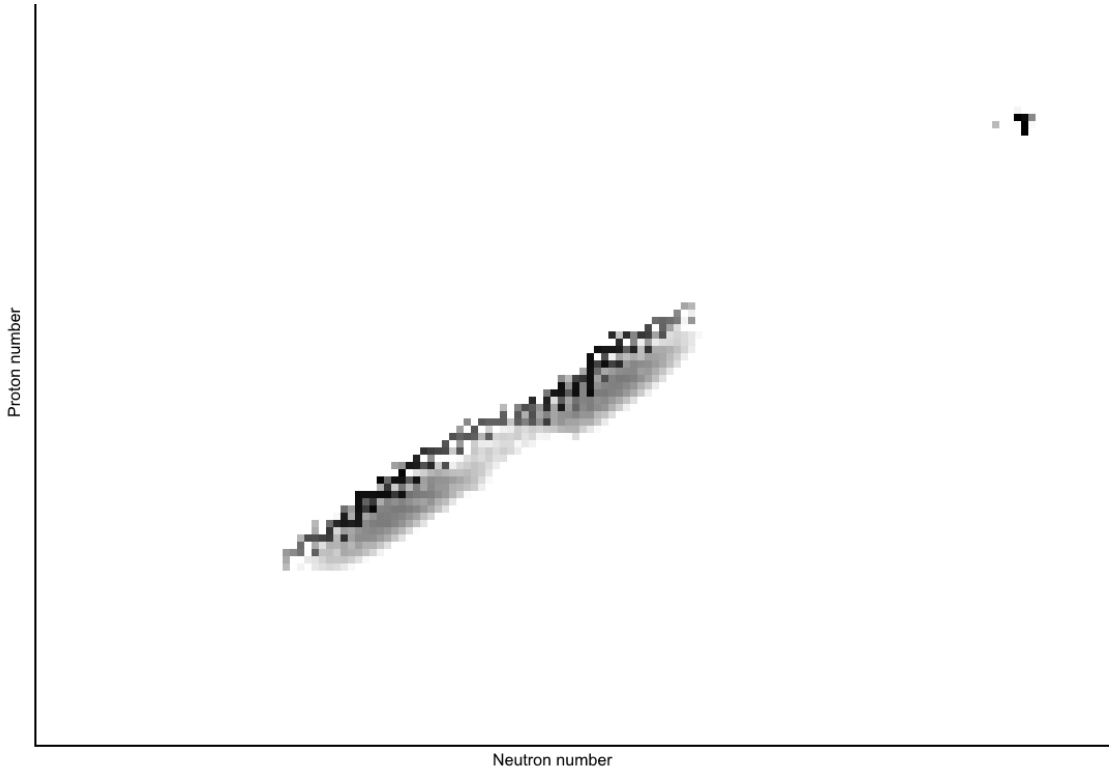


Figure 6.3: Two-dimensional graph of simulated LFTR contents

Each square pixel in the image represents one nuclide. The darker the pixel, the more of that nuclide is present in the simulation, on a logarithmic scale. The horizontal axis describes the number of neutrons, the vertical axis the number of protons. As you can see, Th-232 is still very much present in the form of one of the black squares in the top right. It is surrounded by a few of its neutron capture and decay products: protactinium and uranium.

The large cloud in the centre shows the fission products of U-232. You can see it has been divided into two parts, the lighter and heavier nuclide pair that is formed during fission. Also visible is the darker line on the top left of this cloud: the stable nuclei. Over time the less stable nuclei decayed via β^-, where each decay brought them one step up and left, until they reached a stable state.
To the right is a real life example from the Japan Atomic Energy Agency's website.

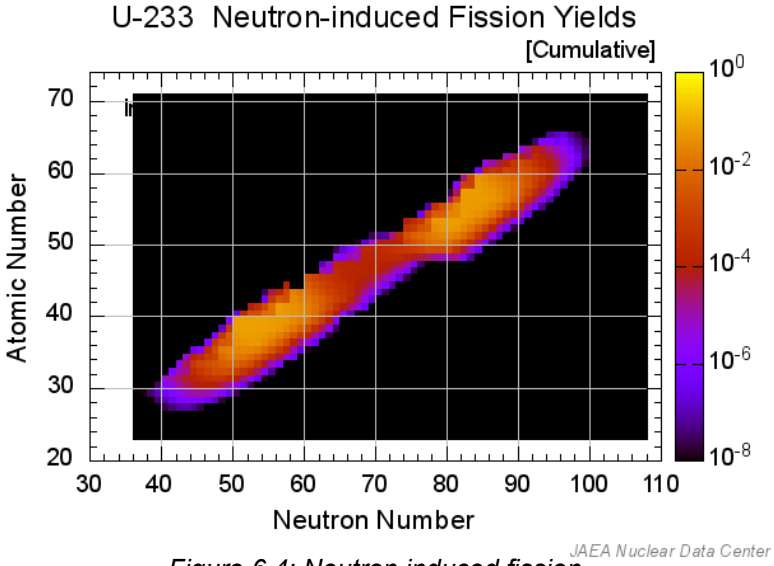


Figure 6.4: Neutron induced fission yield (JAEA)

6.3: Analysis

Not all formed nuclides are equally relevant. Many are stable, and commonly occur in nature, without being hazardous to humans. In the following section we outline a few nuclides that are particularly radioactive, but also potentially useful. Fission products with half-lives in the range of 10 - 10^6 years are also studied, as they make up the radioactive waste of the simulated LFTR.[41][42]

Long lived fission products

When fission products have a half-life of 200,000 years or more, they are considered long lived fission products, however, for waste managements a half-life of more than 30 years is seen as long enough to consider it long lived. There are seven different nuclides that are known as long lived fission products, and they are all products of the LFTR. While there are more nuclides with half-lives well above 200,000 years, these are effectively stable, as their half-lives are longer than the age of the universe.[29][30][31][32][33]

Technetium-99

Half-life: 211,000 years
General: Tc-99 emits weak β^- radiation and is the most relevant waste product of the LFTR in terms of the amount produced.
Dangers: Due to the low energy of the beta particles, Tc-99 radiation is relatively easy to contain: even a normal layer of glass is sufficient. The main danger lies in inhalation, as is it capable of doing damage while in the lungs.

Tin-126

Half-life: 230,000 years
General: A generally useless nuclide that decays through β^- but has a strong gamma decay in its decay chain.

Selenium-79

Half-life: 327,000 years
General: A weak β^- emitter.

Zirconium-93

Half-life: 1.53 million years
General: A weak β^- emitter.

Caesium 135

Half-life: 2.3 million years
General: A long lived fission product, a half-life of 2.31 million years, emits β^-, only alkaline LLFP.[39]

Palladium-107

Half-life: 6.5 million years
General: A long lived fission product that emits β^- and is relatively harmless because its decay chain contains no gamma decay.

Iodine-129

Half-life: 15.7 million years

General: The longest lived fission product that is still considered dangerously radioactive.

Dangers: Not very radioactive, but as a result hard to detect. With accumulation this radiation is still hazardous.

Uses: Geochemical tracer.[36]

Medium lived fission products

Krypton-85

Half-life: 10.76 years

General: A medium lived fission product that emits β^- particles. It is typically present in gas form.

Dangers: As it is a gas, danger to humans occurs by inhalation, in which beta and gamma radiation from its decay chain cause tissue damage and possible asphyxiation. In the case of skin contact it can cause frostbite if the gas is concentrated enough.

Uses: Lighting, inspection of components of an aircraft for small defects, detection of cracks and leaks in pipes and semiconductors.[40]

Strontium-90

General: A medium lived fission product, half-life of 30 years, emits β^- particles. After entering the bloodstream, it primarily gets absorbed into the bones and marrow.

Dangers: Cause of bone cancer or leukaemia due to high concentrations in the bones.

Uses: Radiothermal generators (RTGs), radiation therapy.[35]

Caesium-137

Half-life: 30 years

General: Medium lived fission product that emits β^-. Note that its decay chain contains strong gamma emitters.

Dangers: Cancer from β^- and gamma radiation. Due to its electropositivity, it can quickly contaminate large bodies of water.[38]

Uses: -

Other fission products

Iodine-131

Half-life: 8 days

General: A strong gamma emitter.

Dangers: Exposure, especially during childhood, leads to increased risk of thyroid disease.

Uses: Treatment for thyroid cancer and imaging the thyroid. Diagnosis of abnormal liver function, blood flow in the kidneys and urinary tract obstruction.[37]

6.4: Discussion

Validity of results

Both the mass distribution graph and the cumulative fission yield 2D graph appear valid when compared to real life results, taken from the Japan Atomic Energy Agency's website. The biggest discrepancy may be the spikes that appear on the left area of the mass distribution. If we had to pinpoint a reason for this, it would most likely be the fission yield data and the calculations we performed on it before using it.

Two-dimensional graph

It is important to note that very few transuranics were formed, even though we did not simulate protactinium separation. Our simulation is far too basic to *solely* attribute this lack of transuranics to the characteristics of a LFTR, but we can still do so in part. In the neutron-capturing chain that leads to transuranics, some of the isotopes of uranium have rather large fission cross sections. These isotopes are 'burned' in the reactor, decreasing the amount of transuranics drastically.

"LFTR's longest lasting waste only takes 300 years to decay to background radiation levels"

This was stated in paragraph 4.1, as a theoretical advantage of the LFTR. At first glance, this statement seems false, taking the results of the simulation into consideration. All seven long living fission products are formed by the LFTR, and they all take well above 300 years to decay. However, these fission products have half-lives ranging from 200,000 to 16,000,000 years, and they deliver low doses of radiation, most likely under the threshold of background radiation.

Background radiation is defined as a relatively constant low-level radiation from environmental sources such as building materials, cosmic rays, and ingested radionuclides in the body (http://www.dictionary.com/browse/background-radiation). Worldwide, the average effective dose from background radiation is around 2.4 mSv.[47]

The eighth longest living isotope produced in a considerable amount is samarium-151 with a half-life of 88.8 years. At first, it may deliver more than 2.4mSv, but after 300 years, only 9.6% of the samarium produced by the LFTR is left, and by then the effective dose from samarium will be under 2.4mSv if it was not before. Samarium-151 decays to europium-151, which is stable. The other unstable products have even shorter half-lives, meaning that they will be practically absent in 300 years.

With our limited knowledge and time, we can assume that this statement is correct. We can also infer this from the fact that numerous sources seem to agree on the issue when they summarize the advantages of the LFTR.

Protactinium

Although we didn't simulate it, protactinium separation does pose a possible problem for the thorium fuel cycle: it decays to U-233, so if one could isolate it, the production of extremely concentrated fissile uranium is a possibility, and with it the production of bombs. The existence of U-232 in the mixture would hinder this by sending out high energy gamma rays, making it deadly to handle and easy to spot. If a LFTR design kept any separated protactinium within the containment structure, and therefore within the hazardous core radiation, it would also severely hinder any nefarious plans. From this we can conclude that protactinium separation can be used to make bombs, technically speaking, but also that the properties of a LFTR make it unlikely to occur.

Improvements

We believe the first major bottleneck in our research was the fission yield data. A readily available independent U-233 fission table that was constructed by nuclear physicists would most likely be more accurate than our own version, which we created using a cumulative table, and even that cumulative table contained some possible discrepancies.

The second factor is the simplicity of our simulation. If we had more time for the PWS and more experience in the subjects we discuss, as well as in coding Python, our results would most likely have been more accurate. The addition of factors like neutron poison simulation and the capture cross sections of some nuclides is particularly needed to get the biggest improvement.

Conclusion

What are the characteristics of the products of the thorium cycle in a LFTR?

In this PWS we first researched thorium in general, and then focused on the molten salt reactors that would use it. With that knowledge, we created a simulation of a LFTR.

Conclusions to draw from PWS:

- Liquid Fluoride Thorium Reactors appear to form fewer transuranics than other reactors.
- The waste from Liquid Fluoride Thorium Reactors is less dangerous than that of other reactors because of its decay characteristics.
- Creating bombs from protactinium appears to be technically possible, but practically unlikely.

What we learned

Before the start of this PWS, we had never created anything of this magnitude. Along the way, we came to understand what it really means to work together during an extended period of time. With the help of our counsellors, we learned how to make a scientific paper, and we realized the amount of effort that goes into research.

In terms of concrete knowledge, we have learned a lot about the nuances of LFTR technology, where before we only knew of the overly hyped arguments for the LFTR that have circulated the internet since Kirk Sorensen's TEDx talk. Also, our knowledge of nuclear reactors and the associated physics has expanded vastly; we are now able to explain the basic workings of a fission plant.

Acknowledgements

In writing our PWS we relied on several people for aid in the development of almost every chapter. In no particular order, we would like to thank them for their advice and support:

- Nynke Schmersal and Jan-Willem Rensman, our counsellors from Fluor, whose commentary improved upon nearly every sentence you have read so far, and who helped us get in contact with NRG.
- Frodo Klaassen and Sander de Groot, our contacts at NRG. Their advice helped steer the PWS in the right direction, and their technical knowledge corrected us on many matters.
- Wolf Mensé, a mathematics teacher at our school and simultaneously our contact with Fluor, for giving us the opportunity to choose this subject in the first place.
- Our families, who helped with determining the subject and with proofreading.

Appendices

Appendix I: Full simulation Python code

```
import math
import numpy

# Filling data from text files
data = [] # the list that contains all nuclear properties, from data.txt
empty = [] # an empty list that is used as a template
for i in range(119):
    data.append(300*[""])
    empty.append(300*[0])
for i in open("data.txt").readlines():
    i = i.split()
    data[int(i[0])][int(i[1])]=" ".join(i[2:])
fission = [x.strip("\n") for x in open("fission.txt").readlines()]
oldnumbers = empty[:]
newnumbers = empty[:]

# Constants & settings
nflux = math.pow(10,10) # neutron flux, in neutrons per cm^2 per second
ts = 86400 # time step, in seconds
mf = 1 # multiplication factor (speed up)
limit = 3650 # set to False if limitless
newnumbers[90][232] = 3.036*(10**22) #approx one cubic cm of Th-232

print("XC v1.2")
print("Estimated time to calculate: "+str(0.03*limit)+" seconds")


#CALCULATING
cycles = 0
while cycles <= limit:
  cycles += 1
  oldnumbers = newnumbers[:]
  newnumbers = empty[:]
  writefile = open("output/c"+str(cycles)+".txt",'w')
  for i in range(len(oldnumbers)):
    for j in range(len(oldnumbers[i])):
      if oldnumbers[i][j] != 0:
        writefile.write(str(i)+" "+str(j)+" "+str(oldnumbers[i][j])+"\n")
  writefile.close()
  for i in range(len(oldnumbers)):
    for j in range(len(oldnumbers[i])):
      if oldnumbers[i][j] != 0:
        start = int(oldnumbers[i][j])
        factors = [] # list of factors to multiply the start value with
(note: factors are calculated individually, as if there are no other
mutations)
        for q in data[i][j].split(";"):
          if q.split(":")[0] == "N": # if reaction mechanism is neutron
interaction -> use neutron interaction formula
                        factors.append(float(1-
(ts*mf*nflux*float(q.split(":")[2].split()[0])*(10**-24)))) # 1-the
fraction of start that mutates = fraction that does NOT mutate
```

```
          if q.split(":")[0] == "D": # if reaction mechanism is natural
decay -> use decay rate formula

factors.append(float(0.5**((ts*mf)/float(q.split(":")[2].split()[0]))))
        end = start*numpy.product(factors)
        newnumbers[i][j] = int(end)
        dval = []
        if len(data[i][j].split(";")) == 1:
          dval.append(start-end)
        else:
          for p in range(len(data[i][j].split(";"))):
      dval.append((start-end)*((1-factors[p])/(len(factors)-
sum(factors)))
        for q in data[i][j].split(";"):
          indexnum = data[i][j].split(";").index(q)
          if q.split(":")[0] == "N":
            if q.split(":")[1] == "C":
              newnumbers[i][j+1] += dval[indexnum]
            if q.split(":")[1] == "F":
              for f in fission:

newnumbers[int(f.split()[0])][int(f.split()[1])] +=
int(float(f.split()[2]) * dval[indexnum])
          if q.split(":")[0] == "D":
            if len(q.split(":"))<4: # if True, singular decay, so no
further calculation needed
              if q.split(":")[1] == "B-": newnumbers[i+1][j] +=
dval[indexnum]
              if q.split(":")[1] == "B+": newnumbers[i-1][j] +=
dval[indexnum]
              if q.split(":")[1] == "EC": newnumbers[i-1][j] +=
dval[indexnum]
              if q.split(":")[1] == "A": newnumbers[i-2][j-4] +=
dval[indexnum]
            elif len(q.split(":"))>3: # if True, multiple decay, use
given factor to separate dval
              if q.split(":")[1] == "B-":
                if q.split(":")[3].split()[0] == "B-N": # if both True,
B- decay and B- with neutron emission
                  newnumbers[i+1][j] += int(dval[indexnum]*(1-
float(q.split(":")[3].split()[1])))
                  newnumbers[i+1][j-1] +=
int(dval[indexnum]*(float(q.split(":")[3].split()[1])))
                if q.split(":")[3].split()[0] == "EC": # if both True, B-
decay and electron capture
                  newnumbers[i+1][j] += int(dval[indexnum]*(1-
float(q.split(":")[3].split()[1])))
                  newnumbers[i-1][j] +=
int(dval[indexnum]*(float(q.split(":")[3].split()[1])))
              if q.split(":")[1] == "B+":
                if q.split(":")[3].split()[0] == "A": # if both True, B+
decay and alpha decay
                  newnumbers[i-1][j] += int(dval[indexnum]*(1-
float(q.split(":")[3].split()[1])))
                  newnumbers[i-2][j-4] +=
int(dval[indexnum]*(float(q.split(":")[3].split()[1])))
```

Appendix II: Full fission file converter code

```
def generate(iso): # generate the chain resulting from single nuclide
  chain = []
  selected = iso
  while True:
    chain.append(selected)
    if data[int(selected.split()[0])][int(selected.split()[1])] == "S":
      break
    else:
      mtype =
data[int(selected.split()[0])][int(selected.split()[1])].split(":")[1]
      if mtype == "B-":
        selected = unsort_cumul[[" ".join(x.split()[:2]) for x in
unsort_cumul].index(str(int(selected.split()[0])+1)+"
"+selected.split()[1])]
      elif mtype == "B+" or mtype == "EC":
        selected = unsort_cumul[[" ".join(x.split()[:2]) for x in
unsort_cumul].index(str(int(selected.split()[0])-1)+"
"+selected.split()[1])]
  return chain

def individualise(chain):
  global indiv
  val = float(0)
  for i in chain:
    cum = float(i.split()[2])
    if (cum-val)>0:
      indiv.append(" ".join(i.split()[:2])+" "+str(cum-val))
    elif (cum-val)<0:
      indiv.append(" ".join(i.split()[:2])+" "+str(cum))
    val = float(cum)

def compare(l1,l2):
  if set(l1).issubset(set(l2)):
    return [l2,l1]
  elif set(l2).issubset(set(l1)):
    return [l1,l2]
  else:
    return False

data = []
empty = []
# step 0: import data
for i in range(119):
  data.append(300*[""])
  empty.append(300*[0])
for i in open("data.txt").readlines():
  i = i.split()
  data[int(i[0])][int(i[1])]=" ".join(i[2:])

# step 1: get cumulative fission values and sort by atomic mass
unsort_cumul = [x.rstrip("\n") for x in
open("cumulative.txt").readlines()]
cumul = []
```

```
for i in range(300):
  cumul.append([])
for i in unsort_cumul:
  cumul[int(i.split()[1])].append(i)

# step 2: go through each mass and find the chains
fullchain = []
for i in cumul:
  chain = []
  for j in i:
    newchain = generate(j)
    inlist = False
    for q in chain:
      result = compare(newchain,q)
      if result == False:
        pass
      else:
        inlist = result
    if inlist == False:
      chain.append(newchain)
  if chain != []:
    fullchain.append(chain)

# step 3: use the chains to calculate individual values
indiv = []
for i in fullchain:
  for j in i:
    individualise(j)

writefile = open("individual.txt","w")
for i in indiv:
  writefile.write(i+"\n")
writefile.close()
```

Appendix III: Program output, sorted by number of atoms

This data can be found as a .txt file here: https://goo.gl/dVf28F

90	232	3.04E+22	34	85	1.8E+09	31	80	15367952
92	233	4.42E+17	58	147	1.8E+09	46	116	15274031
92	234	2.60E+17	56	134	1.8E+09	59	153	15067852
91	233	2.76E+14	40	101	1.74E+09	49	117	14533459
43	99	3.91E+13	39	94	1.6E+09	48	118	14376673
42	95	3.31E+13	37	89	1.57E+09	45	113	14263439
38	90	2.98E+13	34	86	1.55E+09	49	130	13381080
54	136	2.38E+13	55	138	1.53E+09	40	104	13109172
92	235	2.25E+13	39	96	1.49E+09	31	76	12578895
40	93	1.92E+13	51	130	1.45E+09	45	111	12224475
58	142	1.87E+13	58	148	1.43E+09	62	156	10633556
40	94	1.85E+13	53	136	1.42E+09	48	119	10633446
40	90	1.82E+13	56	145	1.33E+09	60	155	10500403
40	92	1.81E+13	55	134	1.33E+09	35	82	10417535
58	140	1.75E+13	50	125	1.31E+09	49	124	10343783
54	134	1.72E+13	45	105	1.25E+09	48	117	9934976
40	91	1.72E+13	51	129	1.2E+09	32	84	9692927
59	141	1.66E+13	35	85	1.19E+09	44	108	9232933
56	138	1.64E+13	36	92	1.17E+09	37	97	9022082
55	133	1.63E+13	42	103	1.16E+09	30	76	8497418
39	89	1.57E+13	44	100	1.14E+09	61	154	8441495
40	96	1.57E+13	42	104	1.13E+09	48	122	7952371
60	143	1.56E+13	35	89	1.12E+09	46	117	7850320
55	137	1.56E+13	37	94	1.11E+09	35	92	7619722
57	139	1.54E+13	39	99	1.1E+09	46	113	7385745
42	97	1.52E+13	51	133	1.07E+09	49	122	7257343
55	135	1.51E+13	52	136	1.05E+09	61	156	6962844
38	88	1.49E+13	51	132	1.03E+09	61	153	6818073
42	98	1.42E+13	58	145	1.02E+09	42	109	6794411
54	132	1.31E+13	50	128	1E+09	47	117	6689370
42	100	1.21E+13	52	127	9.81E+08	47	118	6674502
37	87	1.11E+13	38	97	9.68E+08	44	112	6552649
60	144	1.02E+13	64	160	9.57E+08	46	115	6471162
54	131	9.7E+12	41	102	9.11E+08	62	157	6440526
36	84	9.43E+12	55	143	9.11E+08	53	130	6271056
60	145	9.3E+12	41	103	8.38E+08	45	112	6252202
44	101	8.81E+12	47	111	7.92E+08	30	77	5881661
36	86	8.6E+12	32	72	7.7E+08	63	157	5780589
52	130	7.24E+12	54	141	7.43E+08	62	155	5589618

60	146	6.93E+12	52	131	7.23E+08	43	101	5556579
44	102	6.65E+12	53	138	7.19E+08	45	110	5522038
91	234	6.5E+12	57	147	7E+08	51	136	5401896
37	85	5.79E+12	40	102	6.71E+08	30	75	5254120
36	85	5.71E+12	57	142	6.51E+08	48	128	5243759
53	129	4.47E+12	34	87	6.37E+08	33	78	5184217
45	103	3.87E+12	61	151	6.35E+08	58	152	5100038
60	148	3.54E+12	35	83	6.27E+08	33	87	5031106
36	83	2.93E+12	58	149	6.19E+08	45	115	4852573
44	104	2.67E+12	39	97	6.17E+08	53	141	4675186
53	127	2.55E+12	38	87	6.08E+08	45	114	4590515
58	144	2.55E+12	42	105	5.6E+08	31	81	4561050
52	128	2.53E+12	57	146	5.49E+08	56	148	4522605
62	147	2.52E+12	56	139	5.15E+08	38	100	4488514
61	147	2.44E+12	33	83	5.06E+08	31	75	3891707
62	149	2.1E+12	42	101	5.01E+08	48	121	3779690
34	82	1.6E+12	49	113	4.91E+08	34	90	3761803
46	105	1.37E+12	44	99	4.49E+08	36	95	3580389
60	150	1.33E+12	59	149	4.46E+08	47	116	3473612
56	137	1.24E+12	34	83	4.32E+08	57	150	3223566
35	81	1.02E+12	37	88	4.05E+08	39	102	3182267
41	95	9.41E+11	52	129	4.04E+08	43	111	3176287
62	151	9.12E+11	39	98	3.91E+08	61	157	3025242
40	95	8.57E+11	41	100	3.86E+08	62	158	2917489
39	91	8.08E+11	50	129	3.84E+08	31	74	2798562
34	80	6.83E+11	50	130	3.8E+08	30	78	2718218
38	89	6.36E+11	37	95	3.74E+08	49	131	2664905
50	126	5.64E+11	59	150	3.7E+08	30	74	2654407
62	152	5.37E+11	62	153	3.7E+08	47	120	2598280
46	106	5.15E+11	56	146	3.56E+08	61	152	2207613
58	141	4.28E+11	59	145	3.51E+08	50	134	2049626
58	143	4.12E+11	63	156	3.49E+08	47	114	2025898
56	136	3.97E+11	51	126	3.34E+08	49	120	1905756
34	79	3.47E+11	35	90	3.3E+08	60	156	1885632
46	107	3.13E+11	66	161	3.25E+08	55	146	1878724
63	153	2.9E+11	51	128	3.25E+08	46	118	1875383
46	108	2.18E+11	57	141	3.09E+08	47	115	1825410
34	77	1.79E+11	33	81	3.02E+08	59	154	1810072
56	140	1.74E+11	35	84	2.98E+08	61	150	1702540
34	78	1.74E+11	53	132	2.98E+08	65	161	1695261
44	106	1.73E+11	50	127	2.98E+08	44	113	1685939
59	143	1.66E+11	42	106	2.8E+08	41	108	1681690

51	125	1.6E+11	59	151	2.58E+08	54	144	1561398
52	125	1.57E+11	57	138	2.5E+08	44	107	1531813
48	111	1.34E+11	38	98	2.47E+08	59	144	1501777
62	154	1.25E+11	31	71	2.41E+08	52	139	1484507
44	103	1.21E+11	36	93	2.4E+08	49	119	1306996
47	109	1.15E+11	32	80	2.36E+08	48	124	1280280
46	110	1.08E+11	58	150	2.25E+08	47	121	1249464
49	115	1E+11	33	77	2.18E+08	30	72	1182295
51	123	9.04E+10	53	139	2.14E+08	32	85	1130358
50	117	8.66E+10	60	152	2.1E+08	64	159	1033288
48	113	8.53E+10	34	88	2.06E+08	48	123	1018522
50	124	8.42E+10	39	92	2.05E+08	47	119	889052
52	126	7.34E+10	37	86	2.03E+08	37	84	833667
93	234	7.23E+10	60	151	1.94E+08	62	159	829695
54	133	6.35E+10	59	147	1.92E+08	42	110	817027
53	131	6.05E+10	35	79	1.91E+08	30	79	781900
50	122	5.34E+10	43	104	1.9E+08	45	109	731365
42	99	5.08E+10	54	142	1.83E+08	31	82	726680
51	121	5.07E+10	50	131	1.66E+08	49	132	673877
48	116	4.85E+10	57	148	1.65E+08	40	105	658259
50	120	4.8E+10	32	81	1.62E+08	43	100	627622
48	114	4.75E+10	50	132	1.59E+08	35	93	610562
32	76	4.74E+10	51	124	1.57E+08	63	158	567371
39	90	4.36E+10	39	100	1.5E+08	48	127	550371
50	119	4.34E+10	52	137	1.4E+08	45	116	520170
50	118	4.26E+10	33	84	1.38E+08	30	73	520095
60	147	4.19E+10	60	153	1.36E+08	32	75	508708
48	112	3.92E+10	33	79	1.33E+08	63	159	493528
63	155	3.81E+10	49	129	1.3E+08	44	114	479637
52	132	3.42E+10	55	144	1.29E+08	31	73	466323
64	156	2.95E+10	49	128	1.28E+08	46	111	458042
42	96	2.26E+10	41	104	1.27E+08	61	158	439933
64	155	2.05E+10	41	94	1.26E+08	46	119	430454
36	82	1.92E+10	33	85	1.24E+08	33	88	429944
57	140	1.89E+10	32	82	1.23E+08	40	107	425257
64	157	1.88E+10	42	107	1.07E+08	45	117	401199
33	75	1.82E+10	33	80	1.07E+08	40	106	382276
38	86	1.35E+10	43	105	1.01E+08	39	103	377155
53	133	1.27E+10	40	103	97846655	48	129	327747
32	74	1.21E+10	33	82	95984660	29	74	324969
40	97	1.15E+10	34	76	94938136	58	153	317866
54	130	8.9E+09	43	103	93260434	29	73	312248

41	97	8.14E+09	66	162	92445153	37	98	308059
92	236	8.09E+09	49	127	90193082	53	142	306452
38	94	7.39E+09	48	115	89782057	29	75	299904
64	158	7.23E+09	41	105	89756305	43	112	276086
56	142	7.11E+09	34	81	87148196	51	137	272722
40	98	6.79E+09	36	80	83583655	48	126	270875
38	93	6.69E+09	41	98	80795674	57	151	258140
54	138	6.47E+09	51	134	79232623	36	96	246966
54	137	6.33E+09	35	91	73999121	60	157	240824
36	89	6.26E+09	52	122	73892639	38	101	239683
36	90	6.07E+09	56	135	72002105	63	160	239563
56	143	6.04E+09	30	70	70984213	56	149	231558
37	90	5.98E+09	59	152	68077113	34	91	221766
36	88	5.49E+09	43	108	65846973	62	160	205578
52	134	5.36E+09	59	148	61422734	59	155	199465
40	99	5.23E+09	39	101	58952908	31	72	195332
38	95	5.06E+09	59	146	58118821	51	122	184373
37	91	5.05E+09	60	149	56978423	29	72	162372
41	99	5.01E+09	32	78	54442032	30	80	151423
53	135	5E+09	43	106	53448290	47	122	144466
50	123	4.8E+09	37	96	47338381	61	148	138164
55	140	4.77E+09	60	154	46793761	47	112	133345
38	92	4.5E+09	56	147	46055751	48	130	132070
40	100	4.31E+09	43	107	44302460	33	76	110994
55	141	4.25E+09	32	77	43680276	32	86	108116
56	141	4.17E+09	62	148	43524871	52	140	107568
37	92	4.08E+09	42	108	42837025	31	83	103167
54	139	4.08E+09	41	106	41931828	49	118	102120
56	144	3.95E+09	58	151	41892271	44	115	99088
52	124	3.91E+09	32	83	41604220	55	147	95398
51	127	3.85E+09	51	135	40481864	63	161	91042
39	95	3.76E+09	36	94	39482299	46	120	79840
38	91	3.74E+09	38	99	39167081	50	135	79665
55	139	3.64E+09	53	140	39094851	41	109	75172
53	134	3.45E+09	50	121	37690633	29	71	70514
55	136	3.45E+09	43	109	37531993	29	76	70226
54	135	3.33E+09	31	79	34843284	64	161	61774
39	93	3.16E+09	44	110	33949201	61	159	60587
61	149	3.13E+09	31	77	33171730	54	145	59643
36	91	3.11E+09	31	78	32828874	35	80	49933
62	150	3.1E+09	57	149	31473984	29	77	45406
51	131	3.03E+09	34	89	30542738	47	123	45005

37	93	2.88E+09	33	86	29871883	37	99	39844
35	87	2.87E+09	49	126	28468656	48	125	38443
38	96	2.86E+09	50	133	27059687	42	111	34062
58	146	2.83E+09	46	109	26611053	28	72	32963
91	229	2.74E+09	46	112	26240852	49	133	32392
57	145	2.71E+09	55	145	25350444	64	162	31586
65	159	2.62E+09	41	96	24338907	28	71	30044
54	140	2.59E+09	49	125	21219228	43	113	26537
41	101	2.52E+09	44	111	21080912	35	94	26485
50	115	2.52E+09	54	143	20419679	62	161	24857
57	144	2.49E+09	32	79	19957132	33	89	24457
52	133	2.44E+09	52	138	19167196	45	108	22525
36	87	2.42E+09	41	107	18944678	28	70	20373
32	73	2.33E+09	46	114	18907238	28	73	19236
55	142	2.29E+09	43	110	18856268	45	118	19109
35	86	2.15E+09	44	105	18679963	58	154	17741
53	137	2.11E+09	44	109	17943246	93	237	17277
35	88	2.08E+09	48	120	17789615	63	162	17245
42	102	2.03E+09	49	123	17687674	92	238	16104
34	84	2.01E+09	47	113	17376017	60	158	15569
52	135	1.99E+09	43	102	16970094	30	71	5835
57	143	1.82E+09	49	121	16218004	65	162	1847
			61	155	15732930	29	70	1163
						92	237	325

Sources

Tools

Java-based Nuclear Data Information System, for finding neutron cross sections.
https://www.oecd-nea.org/janis/
(indirect, reads from: https://www.oecd-nea.org/dbforms/data/eva/evatapes/jeff_32/)

Online LaTeX editor for equations
https://www.codecogs.com/latex/eqneditor.php

Large sources (whole websites and books):

Non-profit nuclear information website by nuclear engineers
http://www.nuclear-power.net/

World Nuclear Association
http://www.world-nuclear.org/

Non-profit nuclear information website by nuclear engineers
https://whatisnuclear.com/

Thorium information website partially endorsed by Kirk Sorensen
http://energyfromthorium.com/

Nuclear Research and Consultancy Group (NRG) website
https://www.nrg.eu/

Fast, J.D. (1980). *Energie uit Atoomkernen*. Maastricht, Netherlands: Natuur en Techniek. (H.1, H.2)

LFTR information website
http://liquidfluoridethoriumreactor.glerner.com

Dalen, Bart van. (2015). *Systematische Natuurkunde, basisboek, vwo 6*. Amersfoort, Netherlands: ThiemeMeulenhoff. (H.12)

Small sources (just pages):

Chapter 1

[1] Half-lives
http://arstechnica.com/science/2011/03/know-your-nukes-understanding-radiation-risks-in-japan/

[2] Radioactivity
http://hyperphysics.phy-astr.gsu.edu/hbase/Nuclear/radact2.html#c1

[3] Neutron Energy
http://www.nuclear-power.net/nuclear-power/reactor-physics/atomic-nuclear-physics/fundamental-particles/neutron/neutron-energy/

[4] Neutron cross section
http://www.nuclear-power.net/neutron-cross-section/

Chapter 2

[5] Control rods
http://www.nuclear-power.net/nuclear-power-plant/control-rods/

[6] History of thorium
http://education.jlab.org/itselemental/ele090.html

[7] Components nuclear reactor
https://whatisnuclear.com/articles/nucreactor.html

[8]
http://www.world-nuclear.org/information-library/nuclear-fuel-cycle/nuclear-power-reactors/nuclear-power-reactors.aspx

[9]
http://turbinegenerator.org/steam/steam-turbine-works/

[10] Types of reactors
http://www.nuclear-power.net/nuclear-power-plant/reactor-types/

[11]
http://www.world-nuclear.org/information-library/nuclear-fuel-cycle/nuclear-power-reactors/nuclear-power-reactors.aspx

Chapter 3

[12]
http://energyeducation.ca/encyclopedia/Thorium_fuel_cycle

Chapter 4

[13] Definition of weapons grade U-233
http://web.ornl.gov/info/reports/1998/3445606041609.pdf

[14] Molten Salt Coolants for High Temperature Reactors
https://www.iaea.org/INPRO/CPs/COOL/2nd_Meeting/Literature_Summary-IAEA.pdf

[15] Reactor designs
http://energyfromthorium.com/2007/08/23/the-modified-geometry-2-fluid-molten-salt-breeder/

[16]
http://nucleargreen.blogspot.nl/2011/05/molten-salt-reactor-family-one-fluid.html

[17]
http://liquidfluoridethoriumreactor.glerner.com/2012-solving-technical-challenges-in-building-lftrs/

[18]
http://liquidfluoridethoriumreactor.glerner.com/2012-liquid-fluoride-thorium-reactors-have-passive-and-inherent-safety/

[19] Protactinium separation:
http://www.theenergycollective.com/charlesbarton/64177/what-are-problems-lftr-technology

[20]
http://www.nature.com/nature/journal/v492/n7427/box/492031a_BX1.html

[21] Advantages LFTR
http://lftrsuk.blogspot.nl/p/benefits-of-lftrs.html

[22]
https://sites.google.com/a/illinois.edu/liquid-fluoride-thorium-reactor/lftr-advantages

[23] Disadvantages LFTR
https://sites.google.com/a/illinois.edu/liquid-fluoride-thorium-reactor/lftr-disadvantages

[24]
https://whatisnuclear.com/articles/thorium.html

Chapter 5

[25] List of nuclides with decay data:
https://en.wikipedia.org/wiki/List_of_nuclides

[26] Radioactivity data per isotope
http://nucleardata.nuclear.lu.se/toi/perchart.htm

[27]
http://www.scielo.br/img/fbpe/aabc/v73n1/0065t6.gif

[28] Formulae
http://www.nuclear-power.net/nuclear-power/reactor-physics/nuclear-engineering-fundamentals/neutron-nuclear-reactions/reaction-rate/

Chapter 6

[29] Long lived fission products
http://www.radioactivity.eu.com/site/pages/Long_Lived_Fission_Products.htm

[30]
http://energyfromthorium.com/2008/04/01/long-lived-fission-products/

[31]
http://www.liquisearch.com/medium-lived_fission_products/the_7_long-lived_fission_products

[32]
http://www.liquisearch.com/medium-lived_fission_products

[33]
https://en.wikipedia.org/wiki/Long-lived_fission_product

[34] Medium lived fission products
http://www.liquisearch.com/long-lived_fission_product/evolution_of_radioactivity_in_nuclear_waste/medium-lived_fission_products

[35] Strontium-90:
http://www.chemistrylearner.com/strontium-90.html#strontium-90-properties

[36] Iodine-129
http://enenews.com/forever-iodine-129-growing-radiological-risk-157-million-year-half-life-almost-undetectable-traveled-along-iodine-131-fukushima-concentrates-hotspots

[37] Iodine-131
https://www.cancer.gov/about-cancer/causes-prevention/risk/radiation/i-131

[38] Caesium-137:
http://healthvermont.gov/emerg/drill/dirtybomb/facts-about-cesium137.aspx

[39] Caesium-135
http://periodictable.com/Isotopes/055.135/index2.p.full.prod.html

[40] Krypton-85
http://www.chemistrylearner.com/krypton-85.html

[41] Nuclide information
http://www.world-nuclear.org/information-library/non-power-nuclear-applications/radioisotopes-research/radioisotopes-in-medicine.aspx

[42] Half-life data
https://www.nist.gov/pml/radionuclide-half-life-measurements-data

[43] Typical high neutron flux reactor
http://www.emtr.eu/br2.html

[44] Kirk Sorensen powerpoints
https://web.archive.org/web/20111212091230/http://home.engineering.iastate.edu/~pjscott/Sorensen_Google_LFTR.pdf

[45]
http://www.thoriumenergyalliance.com/downloads/TEAC3%20presentations/TEAC3_Sorensen_Kirk.pdf

[46]
http://nuclearsafety.gc.ca/eng/resources/fact-sheets/natural-background-radiation.cfm

Images

Freeze plug
http://image.slidesharecdn.com/hargraveslftraimhighbrc-100920101208-phpapp01/95/liquid-fluoride-thorium-reactor-blue-ribbon-commission-19-728.jpg?cb=1284981115

U-233 fission yield
http://wwwndc.jaea.go.jp/cgi-bin/FPYfig?iso=nU233

http://wwwndc.jaea.go.jp/cgi-bin/FPYfig?iso=nU233&typ=g3