Max Schalz

Fuel Cycle Simulations To Reconstruct Uranium Enrichment Programmes

A First Implementation

November 2020

Master's Thesis in Physics

Supervisor:
Prof. Dr. Malte Göttsche
III. Physikalisches Institut B
RWTH Aachen University

Examiner:
Prof. Dr. Malte Göttsche
III. Physikalisches Institut B
RWTH Aachen University

Second examiner:
Prof. Dr. Christopher Wiebusch
III. Physikalisches Institut B
RWTH Aachen University

Abstract

The field of nuclear archaeology aims to reconstruct the fissile material production of a state in order to check the consistency of official records or to detect undeclared production. While previous research has mostly focused on specific facilities, such as a nuclear reactor, this thesis shifts the focus towards the fuel cycle as a whole. This so-called integrated approach has the advantage that interactions between facilities are taken into account, as well. However, the reconstruction of material flows in a fuel cycle becomes increasingly difficult with its size.

To counter this difficulty, Cyclus, a nuclear fuel cycle simulator, is used. In order to make it suitable for the nuclear archaeology context, a new module, MIsoEnrichment, is developed in this thesis. It contains an implementation of the matched abundance ratio cascade theory (MARC) by von Halle, which describes the enrichment of a multi-isotope mixture. Thus, using this module in a simulation, one can obtain expectancy values for the concentrations of minor isotopes such as U-234 or U-236 in the enriched uranium product and in the enrichment tails. These isotopes are typically used in nuclear archaeology to reconstruct uranium enrichment programmes.

Finally, a case study is defined and a nuclear fuel cycle is simulated using Cyclus and the new module. By taking measurements from the tails and calculating expectancy values with the matched abundance ratio cascade (MARC), the enrichment programme in the simulated scenario is successfully reconstructed.

Contents

Figures

Tables

Abbreviations

DRE	dynamic resource exchange
HEU	highly enriched uranium
IAEA	International Atomic Energy Agency
ICP-MS	inductively coupled plasma mass spectrometry
INF	Intermediate-Range Nuclear Forces Treaty
JCPOA	Joint Comprehensive Plan of Action *or* Iran nuclear deal
LEU	low enriched uranium
MARC	matched abundance ratio cascade
MOX	mixed oxide
New START	New Strategic Arms Reduction Treaty
NFC	nuclear fuel cycle
NPT	Treaty on the Non-Proliferation of Nuclear Weapons *or* non-proliferation treaty
SEU	slightly enriched uranium
SWU	separative work unit
TIMS	thermal ionisation mass spectrometry
TPNW	Treaty on the Prohibition of Nuclear Weapons *or* nuclear weapon ban treaty
UOC	uranium oxide concentrate
UOX	uranium dioxide

Acknowledgements

Research is not a single-player's game nor is it a 100 metre sprint. It is the hard and sometimes tedious work of a many people collaborating and developing new ideas together. I have been fortunate enough to meet and work with brilliant people throughout my master's studies for which I am incredibly grateful. Most notably, I am glad to have found such amazing colleagues in the Nuclear Verification and Disarmement group.

I would like to thank Malte Göttsche for giving me the opportunity to work on this thesis and to discover this vast field of research of uttermost importance. Thank you for sharing your knowledge on nuclear weapons, verification and disarmament with me.

I am much obliged to Madalina Wittel and Antonio Figueroa. Madalina, you have initiated me to the—sometimes confusing—world of Cyclus and enrichment. Without your guidance and advice, this thesis would not be what it is now. Antonio, thanks for our discussions, which helped me out of a lot of dead ends and thank you for kindly running reactor simulations for me.

I am thankful to have shared the office with Benjamin Jung. Without your jokes, support and our common love for cookies and coffee, this year would not have been as great as it was.

I want to thank Baptiste Mouginot and Jordan Stomps from University of Wisconsin, Madison, for their help and for patiently explaining Cyclus and Mbmore.

I am indebted to Madalina, Benjamin and Philipp Fürst. Thanks for working through some $\mathsf{c}^{\mathrm{h}^{\mathrm{a}}}{}_{o}t\mathcal{IC}$ drafts, thanks for giving critical and constructive feedback, thanks for all the time that you have invested in proofreading this work.

Thank you, Patricia, for composing the informative fuel cycle graphic.

Last, but most definitely not least, I want to thank my beloved parents, my grandmother and my family and friends for having my back and for being by my side. Finally, thanks to Leo for being the coolest and laziest dog on Earth and for being the namesake of my case study.

Chapter 1

Introduction

> *It is now 100 seconds to midnight, the most dangerous situation that humanity has ever faced.*
>
> — 2020 Doomsday Clock Statement

In 2020, the science and security board of the Bulletin of the Atomic Scientists set the Doomsday Clock closer to midnight than ever before. The board members identified three factors threatening mankind: cyber-based information warfare, climate change and nuclear weapons. In the nuclear weapons context, they argued that international cooperation in arms control has come to a halt [5]. Indeed, the list of treaties in force is dwindling with agreements expiring (New START expires in 2021 and no extension has been agreed on as of October 2020), agreements becoming inactive because of non-compliance and withdrawal (INF) and agreements standing on the brink of failure (JCPOA) [3, 8, 48].[1] At the same time, conflicts between nuclear weapon states continue to pose a significant threat, be it on a political level such as the North Korean-American dispute or involving military clashes such as the Sino-Indian border dispute or the Kashmir conflict [19, 38, 64]. Additionally, all nuclear weapon states are modernising their nuclear arsenals or have declared to do so in the near future [34].

Despite these dark circumstances, there are positive developments, as well. On 24 October 2020, Honduras became the fiftieth country to ratify the Treaty on the Prohibition of Nuclear Weapons (TPNW), thereby enabling it to enter into force on 22 January 2021 [29]. This treaty is the first to ban nuclear weapons in their entirety, meaning their development and testing, their possession and use etc. will become forbidden to the adherent parties. Needless to say that nuclear weapon states oppose the TPNW, but growing support for the treaty increases pressure on them to disarm their nuclear weapons. [30]

Looking into the—possibly distant—future, what if a nuclear-weapon state decides to disarm and to accede to the TPNW? One requirement the acceding state must fulfill is to submit a fissile material *baseline declaration*, i.e., an initial declaration on their possession of weapon-grade uranium and plutonium. In the light of this, but also for other verification and disarmament treaties, a precise reconstruction of a country's fissile material production is necessary.

[1] The treaties' full names are listed in the list of abbreviations on page ix.

By assessing the amount of produced fissile material speculations about secret and hidden fissile material or nuclear weapons stockpiles could be prevented. In turn, this would strengthen the credibility and confidence in the verification regime. [18]

The reconstruction of the nuclear past is the subject of *nuclear archaeology*, a comparably new field of research in nuclear disarmament and verification. Nuclear archaeology methods have been developed for various cases and facilities. These include investigating the plutonium production in reactors using long-lived radionuclides [11], determining the U-235 content of enriched uranium using the U-234 content of an enrichment plant's waste [11, 51] and taking measurements in spent nuclear fuel [13]. All of these contributions have one point in common: each one considers only a specific and isolated facility. While this is understandable and the research results are invaluable, putting things together to create the broader picture is of similar importance. This so-called *integrated approach* analyses the interaction between facilities part of the fissile material production in order to find inconsistencies between declarations and actual stockpiles and possibly trace them back to their origins. Specifically, the use of simulation tools becomes vital when looking at countries with a large and complex nuclear programme [21].

In this work, I take a two-pronged approach to nuclear archaeology. First, an overview is given by introducing the nuclear fuel cycle (NFC) and its constituents and the appropriate simulation tools are explained. This is followed by a presentation of the nuclear fuel cycle simulator Cyclus. Then, the enrichment of uranium is described mathematically using the MARC model, leading to the second part of this thesis which is the reconstruction of the uranium enrichment process. A first evaluation of possible nuclear archaeology is done by focussing on using the uranium tails—the depleted side-product of enrichment plants—and the information stored in the form of concentrations and isotopic ratios. From there, conclusions on the enrichment of the product are drawn.

Both aspects, i.e., the NFC and Cyclus on the one hand and enrichment on the other hand, are brought together with the implementation of MIsoEnrichment. The development of this module is one of the key points of this thesis and it performs multi-isotope enrichment based on the MARC theory. Using this module in Cyclus allows applying nuclear archaeology methods to the nuclear fuel cycle as a whole to address the above-mentioned creation of a broader picture. Finally, a case study in a fictious country is defined to analyse the use of MIsoEnrichment and Cyclus.

Chapter 2

Nuclear Fuel Cycle

In this chapter, the *nuclear fuel cycle (NFC)*, the collection of all processes involved in the production of nuclear energy, is presented [27]. It is split into two categories, the civilian and the military NFC. First, the constituents of the civilian fuel cycle are explained to illustrate the broader picture and to give an overview. Then, the misuse of a peaceful nuclear energy project for military purposes is laid out followed by a presentation of nuclear fuel cycle simulators and their benefits.

2.1 Civilian Nuclear Fuel Cycle

In the majority of cases, the civilian NFC aims to produce electrical energy in nuclear power plants.[1] The different stations of the civilian NFC are exemplified in Fig. 2.1. Variations of this fuel cycle are possible and are indicated in the text. The description is based on [61] unless noted otherwise.

Uranium ore mining The NFC starts with the extraction of uranium from ores using open-pit mining, underground mining or (predominantly) in-situ leaching, where the ore is dissolved with chemicals and extracted through a borehole [43]. Then, the product is treated mechanically and chemically to extract the uranium from the ore. It leaves the facility as uranium oxide concentrate (UOC), also known as *yellowcake*, a powder containing uranates[2].

Processing the yellowcake A conversion facility further purifies the UOC and either directly produces nuclear fuel or converts it to uranium hexafluoride, UF_6, which is sent to the enrichment facility. The former is only done for a minority of nuclear reactor types while the latter is more common. The latter also corresponds to the path shown in the scheme, see Fig. 2.1. The conversion to UF_6 serves two purposes: First, it creates a more volatile compound. While uranium hexafluoride is solid at room temperature, it sublimes at 56.5 °C and

[1] Exceptions to this are research reactors and the production of medical isotopes, but these applications are rare compared to energy production.

[2] Uranates are compounds of some uranium oxide (e.g., UO_2, UO_3 or U_3O_8) and a third compound. The exact form in which uranium is present in yellowcake varies. One typical compound would be ammonium diuranate, $(NH_4)_2U_2O_7$.

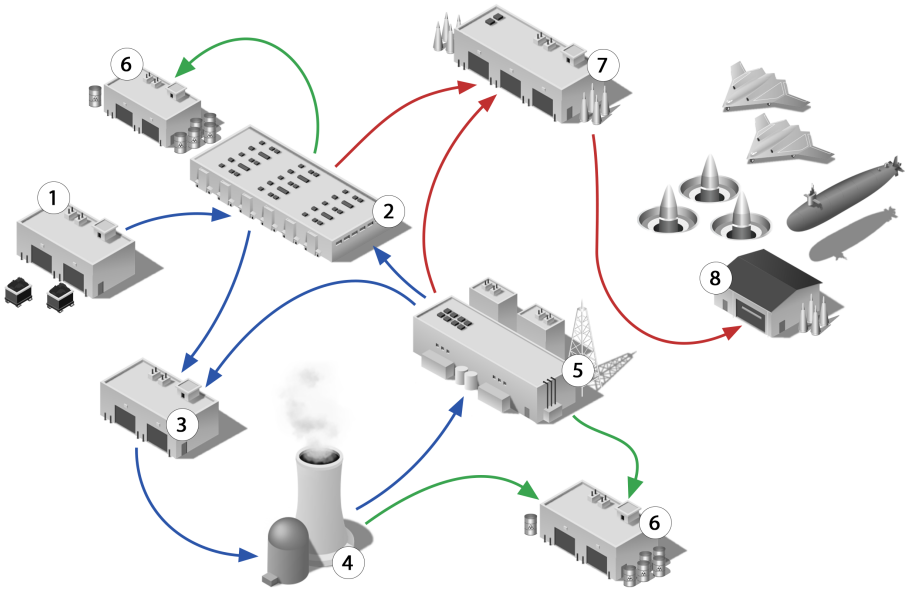


Figure 2.1: Scheme of a nuclear fuel cycle including possible military actors. Nuclear material used or reused in the cycle is shown in blue, nuclear material to be disposed is shown in green and fissile material for military use is shown in red. The different facilities are: (1) uranium mine and conversion plant, (2) enrichment facility, (3) fuel fabrication plant, (4) nuclear reactor, (5) reprocessing plant, (6) nuclear waste storage or repository, (7) nuclear warhead assembly, (8) warhead storage and deployment. Icons adapted from [46].

at 1 atm [44]. This prevents the need for extreme conditions during enrichment such as high heat and/or high pressure. Second, fluorine has only one stable isotope, F-19, meaning that enriching UF_6 effectively only enriches the uranium component.

Enriching uranium Most nuclear reactors do not take natural uranium as fuel but rather need low enriched uranium (LEU), which is produced in the enrichment facility. Uranium is mainly composed of two isotopes: U-235 and U-238. During the *enrichment process*, the U-235 concentration of the feed uranium, i.e., the input, gets raised. Historically, enrichment was performed using gaseous diffusion, but this method has been replaced completely by the more efficient gas centrifuges [7, 28]. For the use as reactor fuel, the concentration gets increased from 0.72% to 3% to 5%. In few cases, such as using the enriched uranium as fuel for a *research* reactor or for medical isotope production, it gets enriched to 20% or even up to 90%. Uranium enrichment is covered in more detail in Chapter 4.

Fuel fabrication The fuel fabrication plant converts the enriched uranium hexafluoride into uranium oxide or metallic uranium using various chemical processes. Then, the uranium is either cast into uranium fuel rods or pressed

to uranium pellets, both of which serve as nuclear fuel. The exact process used depends on the nuclear reactor type the fuel is produced for.

Nuclear reactor As there are a multitude of different nuclear reactor types, an explanation of their functioning would vastly exceed the purpose of this section. A well-arranged introduction can be found for example in [24]. In short, nuclear reactors use a nuclear fission chain reaction to heat water directly or indirectly. This, in turn, generates steam that powers turbines and generators, thereby creating electrical energy. During operation, the nuclear fuel changes its composition. Most importantly, the U-235 concentration diminishes and other uranium isotopes and elements (including plutonium) are formed. As U-235 is the main "force" behind the chain reaction, at some point its concentration becomes too low for maintaining the chain reaction. At this moment, the nuclear fuel is discharged and two options are possible: in a *once-through NFC*[3], the spent fuel is sent directly to a storage unit or repository while in a *closed NFC*, it is sent to a reprocessing facility (as shown in the figure).

Recycling spent fuel In the reprocessing facility, spent fuel is chemically split into three components: uranium, plutonium and nuclear waste, this being fission products of U-235 and Pu-239 or decay products thereof. It is at this station where the NFC becomes an actual *cycle*: the uranium and plutonium can be mixed to produce fresh nuclear fuel and to power a reactor[4]. The nuclear waste is sent to a repository.

2.2 Military Nuclear Fuel Cycle and Proliferation Potential

After having explained the civilian NFC, the military NFC is now discussed. Despite a similar structure and constituents, their goals are in opposition to each other: in the military cycle, the focus lies on the production of fissile materials, i.e., weapon-grade uranium and plutonium, which can be used to build nuclear weapons. In contrast, the reactor does not exist to produce electrical energy, it is—at most—a byproduct; the reactor runs in a different operating mode favouring the production of plutonium over the production of energy.

In a military NFC, there are two key facilities: the enrichment plant, enriching uranium to 90% and higher, and the reprocessing plant, allowing the recovery of plutonium from the spent reactor fuel. Referring back to Fig. 2.1, the military NFC corresponds to the civilian cycle shown in green and blue arrows *and* additionally the red arrows representing the military use of plutonium and weapon-grade uranium.

The general consensus is that the main difficulty in the acquisition of nuclear weapons is not the construction of the weapon itself, but rather the production of sufficient quantities of fissile material [31]. While nuclear-weapon states can

[3]Do not let yourself be fooled by the contradicting use of *cycle*, the only cyclic element in a once-through cycle is its name.

[4]The plutonium could also be used for nuclear weapons production, but remember—this is a presentation of the *civilian* NFC.

have designated military fuel cycles and fissile material, the Treaty on the Non-Proliferation of Nuclear Weapons (NPT) forbids non-nuclear-weapon states to acquire such material with the intent of a military use. However, referring to its 'inalienable right [...] [to use] nuclear energy for peaceful purposes' [55] every state may construct NFCs including enrichment facilities and reprocessing plants. Therefore, a (non-nuclear-weapon) state can legally acquire the key technologies needed to produce fissile material. In such a scenario, only few additional requirements are needed for the construction of a nuclear weapon. Thus, preventing a *proliferation*, the 'unlawful diversion of fissile material for military purposes' [45], is crucial. Under the NPT, the International Atomic Energy Agency (IAEA) inspects the nuclear facilities of the non-nuclear weapon states to ensure that no material is diverted [55].

2.3 Simulating the Nuclear Fuel Cycle

In order to analyse or to optimise an NFC or to look at the impact of an energy transition, material flow rates between the facilities need to be calculated first. Additionally, the operation of the facilities and their outputs need to be considered. For example, the composition of spent reactor fuel must be calculated with simulation tools. Following this, the metrics and NFC characteristics can be determined. Possible quantities of interest could be the natural uranium consumption, the energy production or plutonium production or the amount of waste produced.

For a small fuel cycle with a limited number of actors, such as the one shown previously in Fig. 2.1, these calculations may be done in a spreadsheet or even using pen and paper in a reasonable amount of time. But what if one wants to use high-fidelity reactor simulation models or compare the influence of different operating parameters on the NFCs? What if one wants to recreate the cycle of an entire country, say, the United States or France? It is obvious that with an increasing number of fuel cycle actors its complexity increases and non-automatised approaches such as pen and paper calculations, simple mass balances or spreadsheets quickly reach their limits.

It is at this point that NFC simulators start showing their strengths. Contrary to the above-mentioned methods, the size of a cycle does not render simulations *as such* less feasible or even infeasible. In the worst case, it solely increases the input file's size and the simulation runtime. Ideally, simulators have additional advantages: they can track individual material batches (as opposed to continuous material flows) and changes to material such as decay or irradiation, facilities operations and processing costs and all of it while allowing a flexible deployment of individual fuel cycle facilities [32, 33]. Hence, a nuclear fuel cycle simulator, Cyclus [58], is used in this work and it is presented in Chapter 3.

Chapter 3

Cyclus

In the previous chapter, the nuclear fuel cycle and the use-cases of fuel cycle simulators have been explained. In this chapter, the NFC simulator used in this thesis, Cyclus, is presented. The following sections are based on [25, 58] and on the source code [57].

3.1 Concepts of Cyclus

Cyclus is an open-source, modular fuel cycle simulator written in C++. These properties are important for two reasons. Cyclus being open-source allows to understand *exactly* what the simulation code is doing and to get insights on its working principles. Combined with the modularity, it becomes flexible, can be used in different scenarios and one can write tailor-made modules for specific problems.

Cyclus is built on two pillars, the *agents* and the *dynamic resource exchange (DRE)*. The agents are autonomous and independent actors in the simulation. They represent the various participants of the fuel cycle, such as nuclear facilities, companies or governmental agencies and their behaviour is implemented on an agent-based level, meaning that they are defined independently from each other and they solely interact through the framework set by Cyclus.

The DRE serves as a liaison between the agents by coordinating resource transfers[1] and agent-to-agent interactions. It acts as Cyclus' 'brain' in the sense that it optimises the material flows between the facilities. However, the agents themselves are black boxes to the DRE and the interaction is limited to well-defined queries. It should be noted that Cyclus and its main additional module, Cycamore, do not consider any monetary transactions or the like, meaning that material transactions are done for free. However, it is possible to assign preferences to certain transaction types. Section 3.3 covers this in more detail.

The actual DRE and agent interaction is explained best when looking at an actual simulation time step. Fig. 3.1 shows the different phases of a such

[1] Resources are either *materials* or *products*, where materials are characterised by a quantity and an isotopic composition. In contrast, products are 'the rest': they are characterised by a quantity and a qualitative name. Examples of products could be 'workforce' or 'electricity'.

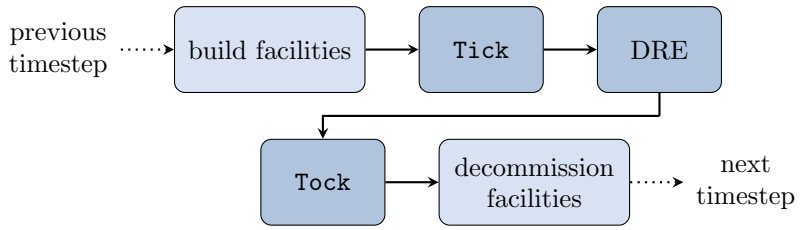


Figure 3.1: The different phases of a CYCLUS simulation time step. The first and the last phases, shown in light blue, are optional: if no facility needs to be built or be deleted, nothing changes. The in-between phases, in dark blue, always take place. In `Tick` and `Tock`, the agents take decisions and perform actions based on their current states. In DRE, resources between agents are exchanged.

a time step. At the beginning of each step, new facilities might be entering the simulation. Then, the simulation back end informs the agents that the `Tick` phase has started. Here, each agent takes individual decisions solely based on its current state, i.e., independent of all the other agents. Once this is completed, the trading phase takes place where resources are exchanged between the agents using the DRE mechanism. Following this, the simulation enters the `Tock` phase, where the agents can again perform actions based on their individual state. Finally, if applicable, facilities might be leaving the simulation.

Two aspects should be noted. First, the `Tick` and `Tock` phases seem to be identical. However, there is a subtle difference: the facilities' state in `Tock` might differ from their state in `Tick` because of material trades during the DRE. The `Tick-Tock` mechanism is generic and leaves it to the developers to decide which actions of a facility can take place in `Tick` and which ones take place in `Tock`.

Second, the DRE is the time step during which resources are exchanged. However, agents may perform actions *during* the DRE if they are directly related to the trade requests or offers. This is explained in more detail in the following section.

3.2 Agents

The agents are all of the actors taking part in a simulation, meaning they are either *facilities* (e.g., a nuclear reactor or a uranium mine), *institutions* (e.g., a company) or *regions* (e.g., a country or a governmental agency).[2] In the following, I focus on facilities as they are the only agent-type of the three that are important to this work.[3]

Facilities do not take part *as such* in the building and decommissioning phase. They cannot influence if they get deployed or destroyed as these actions are done by regions or by the simulation itself. In contrast, they can perform actions during the `Tick` and `Tock` phases as explained previously. Finally, they are the only agent type that participates in the DRE. Here, they can offer and

[2]Note that the DRE itself is not an agent, it *organises* the interaction between agents.

[3]Regions and institutions are higher-ranking agents that allow organising facilities. For example, some facilities could be part of an institution which, in turn, is part of a region. Institutions and regions can then influence the behaviour of their subordinate facilities.

request materials and they react to the incoming requests. In order to illustrate this, I present two facilities from the CYCAMORE module which contains a set of basic agents useful for the simulation of a complete fuel cycle [25, 56].

Reactor The reactor facility takes advantage of the aforementioned difference between `Tick` and `Tock`. Consider a reactor without nuclear fuel: during `Tick`, it does nothing. Then, during the DRE it successfully requests the fuel. In `Tock`, it loads the freshly acquired fuel in the core and starts operation. After a predefined number of time steps, the reactor stops operating in the `Tick` phase, unloads the spent fuel and offers it in the DRE of the same time step. If the `Tick-Tock` mechanism did not exist, then this behaviour would not be feasible and each reactor cycle would take one time step longer than it should.

Enrichment In contrast to the reactor, CYCAMORE's enrichment facility performs most of its actions during the DRE. It does not enrich uranium per se, rather it only enriches uranium *on request*. Consider another facility requesting enriched uranium via the DRE. In response to this, the enrichment facility enriches uranium instantaneously (i.e., during the DRE) and then trades it to the requester. While this instantaneous or 'retroactive' enrichment might seem counterintuitive, it allows CYCLUS to optimise the trades. The alternative (that is, obtain an enrichment request in time step t_1, then start enrichment and send the product to the requester in $t_1 + 1$) would not work as requests are deleted at the end of each DRE and they cannot span over multiple time steps.

3.3 Dynamic Resource Exchange

The DRE is subdivided in three steps starting with the gathering of material offers and demands. Then, these are organised using algorithms to find possible trades and in the last step, the trades are executed. In the following, these steps are explained in more detail. Throughout this explanation, I will use an enrichment facility as an example. It has some uranium as feed material at its disposal and can enrich it both to LEU and highly enriched uranium (HEU). Additionally, it needs more feed material in the form of natural uranium or slightly enriched uranium (SEU).

Gathering offers and demands The gathering starts with the DRE collecting the *requests* of the facilities. In these requests, each facility indicates all of the material it needs along with the respective quantities needed. Facilities can submit zero, one or multiple requests which can also be linked to each other in different ways. The enrichment facility can for example do two *mutual* requests, one for natural uranium and one for slightly enriched uranium, where one *or* the other material will satisfy the requests.

Following this, the collection of requests is passed on to the facilities. In turn, they reply to requests with *bids* in which they specify the quantities and compositions of the materials they are able to trade. Additionally, they might place constraints on bids. Here, the enrichment facility responds to requests for LEU and HEU while taking into account how much feed material it has at its disposal. Requests for material other than enriched uranium (e.g., a request for plutonium) are ignored.

In the last step of the gathering phase, requesters can view the request-bid pairs and they can set preferences on these bids. These preferences can depend on many different aspects, e.g., the enrichment facility prefers the SEU bid over the natural uranium bid because of the higher U-235 content.

Calculating possible trades In the second phase, the bids and requests for different material and quantities are translated into *nodes* that contain all the relevant information. A solver greedily matches nodes, meaning that requests get connected to as many matching bids as possible. Then, connected request-bid pairs get sorted according to preferences and the resulting graph gets solved using a linear program or a mixed-integer linear program model, i.e., the material flows along the connections are calculated. An extensive discussion of solving algorithms and of the trade calculations is found in [15].

Coming back to the example of the enrichment facility means that in this step, its bids and requests are organised together with the bids and requests of the other facilities. Request-bid pairs are formed: the natural uranium request is matched with the bid of a uranium mine, the SEU request is matched with the bid of a reprocessing plant and the reactor's LEU request is matched with the enrichment facility's LEU bid. Then, all the resulting pairs are ranked. Here, the SEU bid gets ranked higher than the natural uranium bid because of the preference set by the enrichment facility. Finally, the solver calculates possible trades. In this case, one trade is that of x units of SEU to the enrichment facility and the other is the trade of y units of LEU to the reactor. The actual transfers take place in the next step.

Executing the trades In the last phase, the trades are executed: Suppliers and requesters are notified by the DRE and in turn, they remove or add the materials from or to their inventories and the executed trades are tracked in a log. Here, the enrichment facility produces the LEU from the feed at its disposal and sends it to the reactor. Then, it receives the SEU from the reprocessing plant. Finally, all of these steps are recorded in the facility's log.

It should be noted that the DRE is performed at each time step of the simulation and it is independent of the preceding ones: a facility can re-submit previously unmatched requests or it can also submit a completely different request.

Chapter 4

Theory of Enrichment

In Chapter 2, the NFC was introduced and it was mentioned that typical nuclear reactors require uranium containing around 3% to 5% U-235. This can be generalised by saying that different uranium applications require different U-235 concentrations. Common uranium enrichment levels are indicated in Table 4.1.[1] These concentration levels are obtained using *enrichment*, a process during which U-238 gets separated from the isotopic mixture, thus increasing the mixture's concentration in U-235. However, in practice uranium is always composed of *at least* three isotopes, the third one being U-234. As will be seen in this chapter, the enrichment process acts differently on each isotope. This so-called *multi-isotope enrichment* can be quantified using the matched abundance ratio cascade (MARC) theory presented in this chapter. Furthermore, the relevant uranium isotopes are introduced, today's prevalent enrichment technology, the gas centrifuge, is explained and an example reconstruction is presented.

Table 4.1: The different categories of uranium defined by the enrichment grade. Note that these are commonly used indications rather than strict boundaries. [1, 39]

category	U-235 concentration [%at]
depleted uranium	< 0.72
natural uranium	0.72
low enriched uranium	0.72 – 20
slightly enriched uranium	0.9 – 2
reactor-grade uranium	3 – 5
highly enriched uranium	> 20
weapon-grade uranium	> 90

[1] In this thesis and in enrichment theory in general, 'U-235 content' is synonymous with *enrichment grade*, the *enrichment level* or the *assay*.

Table 4.2: The composition of natural uranium. Note that the U-234 concentration can vary between 0.0051%at and 0.0054 %at. [49]

isotope	natural U [%at]
U-234	0.0054
U-235	0.7204
U-238	99.2742

4.1 Uranium Isotopes

In this section, the origins of U-234, U-235 and U-238 are presented briefly as they are used in this thesis. Additionally, U-236 is introduced because of its importance to nuclear archaeology. U-234 and U-236 are known as *minor isotopes* because of their small concentrations in uranium and they can be used for reconstructing uranium enrichment [21, 51, 62]. Unless stated otherwise, nuclear data in this section is taken from [37], the times indicated are the isotopes' half-lives and information on natural and reprocessed uranium compositions is taken from [39].

Natural uranium is composed of three isotopes: U-234, U-235 and U-238, see Table 4.2.[2] U-235 and U-238 are remnants of the stellar nucleosynthesis and their concentration fluctuations in natural uranium are low, less than a per mil. In contrast, U-234 is a product in U-238's decay chain, see Eq. (4.1), and its concentration can vary significantly. Typically, the U-234 concentration lies between 0.0051 %at and 0.0054 %at, corresponding to a variation of about 5% [49]. These fluctuations depend on the uranium's origin: natural uranium from one geographical location is similar in isotopic composition but might be different to uranium from another location.

$$^{238}\mathrm{U} \xrightarrow{\alpha,\, 4.5\times10^{9}\,\mathrm{a}} {}^{234}\mathrm{Th} \xrightarrow{\beta^-,\, 24.1\,\mathrm{d}} {}^{234}\mathrm{Pa} \xrightarrow{\beta^-,\, 6.8\,\mathrm{h}} {}^{234}\mathrm{U} \tag{4.1}$$

U-236 is found in significant quantities in reprocessed uranium, i.e., uranium that has been irradiated in a nuclear reactor and then separated from the rest of the spent fuel. Its concentration can amount to as much as 0.3%, hence its presence directly indicates that uranium has been used as nuclear fuel. It is formed in the reactor by the neutron capture of U-235:

$$^{235}\mathrm{U} + \mathrm{n} \longrightarrow \begin{cases} \text{fission products}\,(82\%) \\ {}^{236}\mathrm{U}\,(18\%) \end{cases} .$$

Additionally, reprocessed uranium can contain U-232 and U-233, as well. However, they are omitted here as their concentrations are typically lower than 10^{-6}%. More information can be found for example in [53].

4.2 Separating Element

The smallest, most fundamental unit found in the enrichment process is the *separating element* presented in this section. In the following, I will consider the

[2] Recent studies have discovered traces of U-236 in natural uranium. However, its concentration is minuscule—close to the detection threshold—and can therefore be ignored.

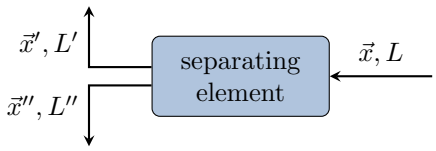


Figure 4.1: A separating element is a generic element that separates the inflowing *feed* stream into an enriched outflowing *product* stream and a depleted outflowing *tails* stream. $\vec{x}$ denotes the isotopic composition and L the material flow rate, with no, single and double prime denoting feed, product and tails stream, respectively.

separating element as a black box, meaning that the physical processes taking place inside are irrelevant. Thus, I can give a generic definition of the element, whereas one of its real-life counterparts, the gas centrifuge, is introduced in Section 4.3. The section is based on [4, 35].

A generic separating element, as shown in Fig. 4.1, is an element that separates an isotopic mixture. It takes gaseous UF_6 with isotopic composition $\vec{x}$ and at a rate L as input material, called the *feed*. Each component x_i of $\vec{x}$ is the mole fraction of the uranium isotope i in the mixture, thus $\sum_i x_i = 1$. Inside the element, the feed then gets separated into two streams. The *upstream* fraction (or *product*), of composition $\vec{x}'$ and at rate L', is enriched in U-235 while the *downstream* fraction (or *tails*), of composition $\vec{x}''$ and at rate L'', is depleted in U-235.

In enrichment theory, one typically does not consider concentrations but rather isotopic ratios called the *abundance ratios* R. They are the concentration of each element relative to the concentration of the *key isotope* or *key component* U-238, denoted x_k. Thus, the abundance ratio R_i of uranium isotope i is defined as

$$R_i = \frac{x_i}{x_k}. \tag{4.2}$$

As the concentrations add up to unity, it follows that

$$\sum_{i=1}^{J} R_i = \sum_{i=1}^{J} \frac{x_i}{x_k} = \frac{1}{x_k}, \tag{4.3}$$

with J being the total number of isotopes present in the mixture.

Among others, the abundance ratios are necessary to quantify the enrichment and depletion of product and tails, respectively. For this, the *overall separation factor* γ is introduced as

$$\gamma_i = \frac{R_i'}{R_i''}. \tag{4.4}$$

Note that the prime and double prime notation for abundance ratios is used analogously to the notation for concentrations, e.g., R_i'' is the tails abundance ratio of isotope i. The components of $\vec{\gamma}$ could be determined experimentally by measuring the abundance ratios and using the definition above.

There is a second method to determine the separation factor, though. If one component of it is known—typically this is γ_{235}—then the other components can

be calculated. The formulae used depend on the enrichment method in question. For enrichment by gaseous diffusion, the separation factors are calculated using

$$\gamma_i = \sqrt{M_{238}/M_i}. \tag{4.5}$$

Here it should be noted that gaseous diffusion enriches uranium in the form of UF_6. Because of this, the molecular mass of uranium hexafluoride has to be used [62].

In the case of enrichment by gas centrifuges, the factors are given by

$$\gamma_i = 1 + \frac{M_{238} - M_i}{M_{238} - M_{235}} \left(\gamma_{235} - 1\right), \tag{4.6}$$

with M_i being the mass of isotope i and γ_{235} typically ranging between 1.2 to 1.6 [16, 41, 62].

4.3 Separating Isotopes: the Gas Centrifuge

The *gas centrifuge* is the real-life counterpart to the separating element and currently is the standard method to enrich uranium. The section is based on [2] and extended discussions can be found in [2, 16, 52].

While the idea of using an artificial gravitational field for isotope separation traces back to experiments as early as 1895 and 1919, it was only in the early 1960s that gas centrifuge became efficient and started superseding other enrichment methods such as gaseous diffusion [2, 61].

The *countercurrent gas centrifuge*, depicted in Fig. 4.2, consists of a fixed cylinder encasing a spinning rotor in a vacuum. Typically, the rotor is mounted using a magnetic bearing at the top and a needle bearing at the bottom to reduce friction and it is powered by an electromagnetic motor. Inside the rotor is an entry for the feed material, which is gaseous uranium hexafluoride. Additionally, scoops are located at both rotor end caps and are in essence open-end pipes allowing the retrieval of enriched and depleted material, respectively.

Upon entering the rotor chamber, the feed material becomes subject to two forces. Because of the rotation and the resulting centripetal force, a radial pressure gradient builds up and balances with the diffusion of the particles. Because of their higher diffusion rate, lighter particles are slightly more concentrated closer to the axis than farther away, resulting in a small separation. In order to magnify it and effectively separate the material multiple times inside one centrifuge, a countercurrent flow is generated. This flow, represented in Fig. 4.2 by the black U-shaped arrows, is established along the rotational axis and perturbs the stable radial gradient. In the centrifuge shown in the scheme, the upflowing stream is located close to the rotational axis. As isotopes get transported between up- and downflowing streams, the former becomes increasingly enriched in U-235 (and in lighter isotopes in general) while flowing upwards. In contrast to this, the downflowing part becomes more and more depleted in U-235. Creating this flow can be achieved by applying a thermal gradient, using a mechanical pump or by designing components to create drag. All things considered, the enriched material tends to 'gather' in the upper end of the centrifuge close to its rotational axis while the depleted material 'accumulates' in the lower outer part of the rotor. At both ends, scoops extract the enriched and depleted uranium hexafluoride from the rotor.

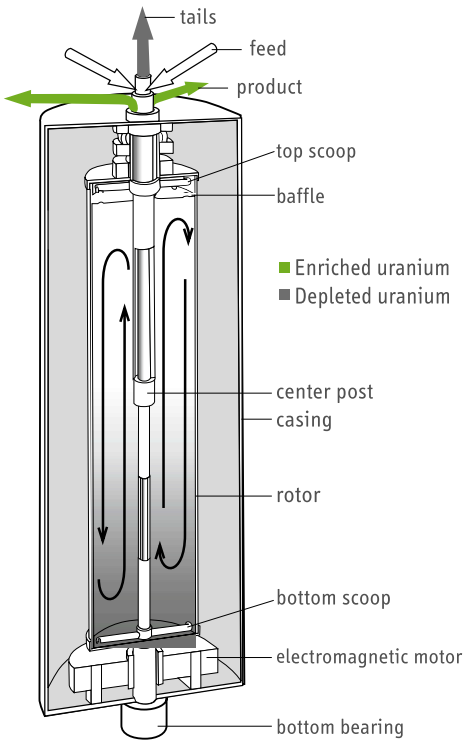


Figure 4.2: Cross-section of a countercurrent gas centrifuge. The curved black arrows inside the rotor indicate the countercurrent flow. In the centrifuge shown here, the *upflowing* stream gets enriched in U-235, but this is model-dependent. Not indicated on this scheme is the vacuum between the casing and rotor. Adapted from [31].

The separation performance of a centrifuge depends on a multitude of factors. Among these are centrifuge-type specific factors such as the peripheral speed, its geometry and the relative positions of feed, product and tails scoops. Technological progress has been immense, compare for example a P-1 type centrifuge from the 1960s of 10 cm in diameter, 2 m in height and operating at 350 m/s to the recently developed AC100 of 60 cm in diameter, 12 m in height and running at 900 m/s, resulting in a performance increase of a factor 100 [16].

Additionally, the operating conditions themselves change the performance. Here, the most prominent variables are the centrifuge feed flow, the countercurrent flow and their ratio. The influence of the feed flow on the separative power and on the separation factor are shown in Fig. 4.3 for a P-1 type centrifuge. The separation factor has been introduced in Eq. (4.4) and the separative power and its unit, SWU, is introduced later in this chapter in Section 4.6.4. For the moment, it is sufficient to view it as a performance indicator—the higher its value is, the more material can be enriched to a higher degree by one centrifuge. One clearly sees that for a given value of the countercurrent-to-feed ratio k there is one feed flow maximising the separative power. Deviations from this optimal point can influence the performance significantly and it is for this reason that in practice, it is often not possible to run centrifuges in an optimal state.

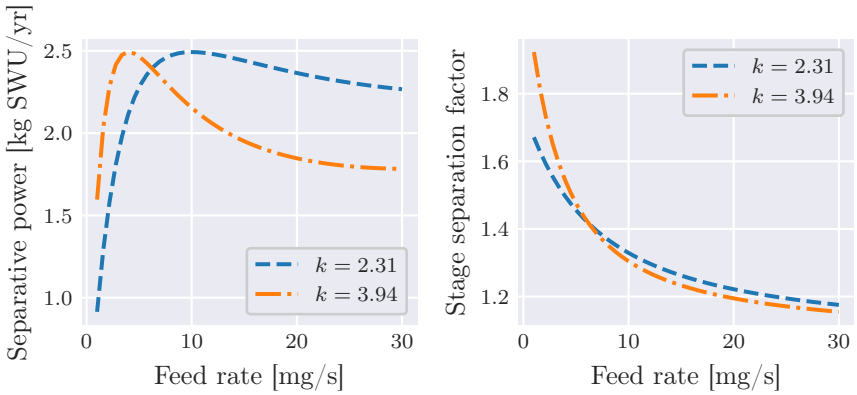


Figure 4.3: The separative power and the overall separation factor of a gas centrifuge strongly depend on the feed flow rate as shown here. Other operational parameters influencing its performance are for example the countercurrent flow rate and k, the countercurrent-to-feed flow ratio. Shown here are the separative power and separation factor of a P-1 type centrifuge. Recreated from [16].

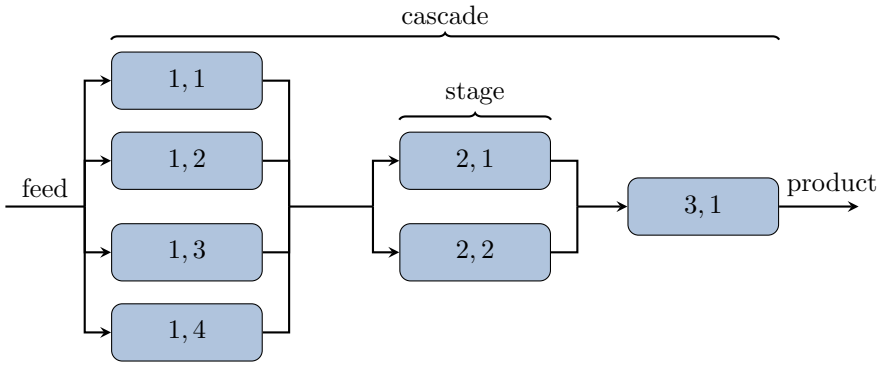
4.4 Interconnecting Centrifuges: the Enrichment Cascade

In the previous sections, the separating element and the gas centrifuge have been introduced and I have explained how uranium is enriched. However, one centrifuge is limited in its enrichment capacity both in terms of quantity it can produce and in terms of product enrichment grade. To counter this limitation, centrifuges are interconnected in two ways as explained in the following. The section is based on [4, 35].

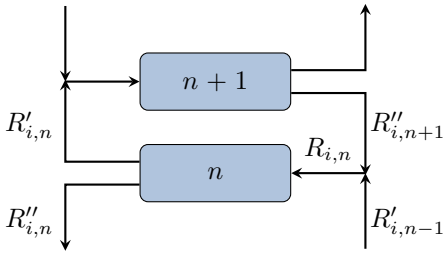
A parallel interconnection of centrifuges is called a *stage*, corresponding to one column in Fig. 4.4a. Inside of one stage, centrifuges behave exactly the same: their separation factors are equal, they all have the same feed rate and the identical feed composition. Equally, they all produce the same amount of product of identical composition and the same amount of tails of identical composition. Thus, the construction of a stage increases the production rate and of course also the feed consumption.

In order to allow for higher product enrichments, centrifuges or stages have to be connected in series, forming a *cascade*, see Fig. 4.4a. In practice, this means that the product of stage 1 gets fed into stage 2, where it gets enriched further. The resulting product gets enriched further in stage 3 and so on. The final enrichment product of the cascade is extracted at the product stream of the last stage. Meanwhile, the tails of each stage *could* get discarded but this would be highly inefficient considering that they still contain U-235, especially in the upper stages.[3] To prevent this, the tails are recycled. More specifically, while

[3] Typically, in 'enrichment slang' height indications (e.g., *upper*, *lower*, *above*, *below*) refer to the position of a stage in the cascade. Here, the top would be the last stage, where the cascade product is withdrawn, and the bottom would the stage where the cascade tails are withdrawn. In contrast, the *width* of the cascade refers to the number of separating elements in the stages.



(a) Separating elements arranged in a three-stage cascade. Each box corresponds to one separating element and each column to a stage. The first number in each box indicates the stage number, the second one the number of the separating element in the stage. Note that the tails flows are not shown here.



(b) Linking in a countercurrent symmetric cascade with the abundance ratios of isotope i indicated. Contrary to the figure above, each box in this figure represents a stage. Note how the tails of stage $n+1$ are reused in stage n.

Figure 4.4: Parallel and serial interconnection of separating elements in a symmetric countercurrent enrichment cascade

the product is fed to the subsequent stage, the tails are reused in the feed of the previous stage, see Fig. 4.4b. This configuration is known as the *symmetric enrichment cascade* and this cascade type is the one considered throughout this thesis. At this point, six additional variables need to be introduced:

- $\vec{x}_F$ and F: the composition and material flow rate of the *cascade* feed,
- $\vec{x}_P$ and P: the composition and material flow rate of the *cascade* product,
- $\vec{x}_T$ and T: the composition and material flow rate of the *cascade* tails.

It follows that in a cascade composed of N stages:

$$\vec{x}_P = \vec{x}'_N, \quad P = L'_N, \tag{4.7}$$

with the subscript denoting the stage number (in this case the last stage N). It may be tempting to conclude analogous relations such as $F = L_1$, but this is not true, as will be explained in the following paragraphs.

In order to enhance the tails recycling, stages below the cascade feed stage can be added similar to the stages above the feed stage: while the latter increasingly *enrich* the product, the former increasingly *deplete* the tails. Thus,

the cascade is divided in two parts: below and above the cascade feed. The section above the cascade feed is called the *enriching section* and its purpose is to produce the uranium with the desired U-235 content. In the following, its stages will be denoted using the variable n with $1 \leq n \leq N$. Here, stage number 1 is the stage directly above the cascade feed and stage N is the stage where the cascade product is withdrawn. The other section is the *stripping section*. In contrast, its purpose to ensure that as little U-235 as possible gets discarded into the cascade tails thereby increasing the cascade's product rate. Stripping section stages are denoted by m with $1 \leq m \leq M$, M being the stage directly below the cascade's feed[4] and stage $m = 1$ being the stage from which the cascade tails are extracted. Thus, the equalities in Eq. (4.7) can be completed as follows:

$$\begin{aligned} \vec{x}_T = \vec{x}''_{m=1}, \quad & T = L''_{m=1} \\ \text{but } \vec{x}_F \neq \vec{x}_{n=1}, \quad & F \neq L_{n=1}. \end{aligned} \tag{4.8}$$

The inequalities in Eq. (4.8) are a result of having a stripping section. Consider the feed of stage $n = 1$: it is composed of the tails of stage $n = 2$, of the product of stage $m = M$ and of the cascade feed. This can result in the mixing of streams of different compositions and clearly $L_{n=1}$ is larger than F. However, the inequalities become equalities in special cases: the former if a binary mixture is considered, i.e., a mixture composed exclusively of U-235 and U-238, and the latter if no stripping section exists.

4.5 Efficient Enrichment: the Ideal Cascade

The *ideal cascade* is a theoretic model and it is the most efficient enrichment cascade possible.

For a binary uranium mixture, a cascade can be constructed such that at each interconnection, material streams of identical concentrations merge. Specifically, considering the right-hand side node in Fig. 4.4b, the following equalities hold

$$R_{i,n} = R'_{i,n-1} = R''_{i,n+1}, \quad \forall\, i, \forall\, n. \tag{4.9}$$

This has a large impact. It means that at no point in the cascade, enrichment capacities get 'lost' or 'wasted' because they would be need to counter inefficiencies. Thus, in practice this means that the number of centrifuges and the power consumption get minimised.

However, there is one caveat to this model: it is only applicable to the enrichment of binary mixtures and in the multi-isotope case Eq. (4.9) breaks down. In this case, the MARC theory can be used and it is introduced in the next section.

4.6 Multi-Component Isotope Separation

The MARC theory is the multi-isotope equivalent of the ideal cascade. In this work, I present the model developed by von Halle [22]. Contrary to other

[4] Hence, the feed of stage M is composed of the product of $M - 1$ and of the tails of stage $n = 1$, its product flows in the feed of stage $n = 1$ and its tails in the feed of stage $M - 1$.

versions, this one is *not* limited to separation factors close to unity. Thus, it is applicable to both enrichment by gaseous diffusion *and* enrichment by gas centrifugation.

In addition to the definitions and notions introduced in the previous section, it is assumed that the overall separation factor is constant (i.e., identical at each stage) and that no material is lost (i.e., mass is conserved). Moreover, the U-235 component of the uranium mixture is called the *matched* component and it is designated by the subscript j. The matched component has the property that when two streams of the cascade are mixed, its abundance ratio R_j is identical in both streams.[5] This has several implications, consider for example the feed stream of stage n, composed of the tails of stage $n+1$ and of the product of stage $n-1$. For component j, the equalities $R''_{j,n+1} = R_{j,n} = R'_{j,n-1}$ hold. This corresponds to the bottom right node in Fig. 4.4b shown previously in Section 4.4. Taking into account Eq. (4.4), this implies:

$$R''_{j,n+1} = \gamma_j R''_{j,n-1} = \sqrt{\gamma_j} R''_{j,n}. \tag{4.10}$$

The last equality holds because the matched component gets separated symmetrically, meaning the product stream gets enriched by a factor $\sqrt{\gamma_j}$ and the tails stream gets depleted by a factor $\sqrt{\gamma_j}$. In mathematical terms, this implies

$$R'_j = \sqrt{\gamma_j} R_j \text{ and } R''_j = \frac{1}{\sqrt{\gamma_j}} R_j.$$

In the following, the definition $\beta_j = \sqrt{\gamma_j}$ is used. From here, one can determine a relation for the abundance ratios spanning over multiple stages

$$R''_{j,n} = \beta_j^{n-N-2} R_{j,P}, \tag{4.11}$$

where the -2 in the exponent arises from replacing $R''_{j,N}$ with $\gamma_j^{-1} R_{j,P}$.

4.6.1 Enriching Section

Consider the enriching section of a cascade with N stages such that stage $n = 1$ is the cascade feed stage and the upstream flow of stage $n = N$ is the cascade product stream. The material balance of isotope i at the last stage equals

$$L'_{N-1} x'_{i,N-1} = (L'_{N-1} - P) x''_{i,N} + P x_{i,P}.$$

This equation takes the same form for the material balance at the top of the cascade above stage n and can be written in terms of abundance ratios using Eqs. (4.2) and (4.4), yielding

$$L'_n x'_{k,n} \gamma_i R''_{i,n} = (L'_n - P) x''_{k,n+1} R''_{i,n+1} + P x_{i,P}. \tag{4.12}$$

If we consider the key component ($i = k$) in Eq. (4.12), both the abundance ratios and the separation factor equal unity and we obtain an expression for $x''_{k,n+1}$. This is inserted into Eq. (4.12) and the equation is rewritten for the j^{th} component. Furthermore, Eqs. (4.10) and (4.11) allow substituting $R''_{j,n}$ and $R''_{j,n+1}$, which yields

$$L'_n x'_{k,n} = \frac{\beta_j^{N+1-n} - 1}{\beta_j - 1} P x_{k,P}.$$

[5] Hence the name of the theory.

and

$$z_n = \frac{L'_n x'_{k,n}}{P x_{k,P}} = \frac{\beta_j^{N+1-n} - 1}{\beta_j - 1}. \tag{4.13}$$

It should be noted that z_n is introduced for convenience, it has no physical meaning. By performing multiple substitutions, as indicated in Eqs. (13) to (15) in [22], one obtains a first-order linear difference equation

$$z_{n+1} R''_{k,n+1} - \gamma_i^* z_n R''_{i,n} = -\frac{R_{i,P}}{\beta_j} \tag{4.14}$$

with $\gamma_i^* = \gamma_i / \beta_j$. Using the boundary condition $R_{i,P} = \gamma_i R''_{i,N}$, a solution to Eq. (4.14) in terms of abundance ratios is found:

$$R''_{i,n} = \frac{R_{i,P}}{(\gamma_i^* - 1)\,\beta_j z_n} \left(1 - \gamma_i^{*n-N-1}\right). \tag{4.15}$$

By writing $R''_{i,n}$ and $R_{i,P}$ as ratios of concentrations, see Eq. (4.2), and by substituting the resulting $1/x''_{k,n}$ with Eq. (4.3) one obtains the final result

$$x''_{i,n} = \frac{\frac{x_{i,P}}{\gamma_i^* - 1}\left(1 - \gamma_i^{*n-N-1}\right)}{\sum_{l=1}^{J} \frac{x_{l,P}}{\gamma_l^* - 1}\left(1 - \gamma_l^{*n-N-1}\right)}, \tag{4.16}$$

with J being the number of isotopes present in the uranium mixture. Given a product concentration, known separation factors and known number of stages, this formula allows to calculate the composition of the downflowing stream of the n^{th} stage. Furthermore, we can obtain similar formulae for $x_{i,n}$ and $x'_{i,n}$ by multiplying Eq. (4.15) with β_i and γ_i, respectively, and repeat the above-mentioned substitutions.

Having determined the concentrations, the next step is to determine the flow rates. This is achieved by solving Eq. (4.13) for L'_n and using the previous result:

$$\begin{aligned} L'_n &= P x_{k,P} z_n / x'_{k,n} \\ &\overset{(4.3)}{\overset{(4.4)}{=}} P x_{k,P} z_n \sum_{i=1}^{J} \gamma_i R''_{i,n} \\ &\overset{(4.15)}{=} P \sum_{i=1}^{J} \frac{\gamma_i^* x_{i,P}}{\gamma_i^* - 1}\left(1 - \gamma_i^{*n-N-1}\right). \end{aligned}$$

4.6.2 Stripping Section

As the development of the MARC in the stripping section is similar to the one in the enriching section, I omit the calculations and present only the assumptions and results. The whole derivation is found in [22], Section 2.3 pp. 9–11.

Consider a stripping section with M stages, where the downflowing stream of stage $m = 1$ is the cascade's tails stream and where stage M is the stage directly below the cascade feed stage. For this section of the cascade, the concentration in the downflowing stream of stage m equals

$$x''_{i,m} = \frac{\frac{x_{i,T}}{\gamma_i^* - 1}\left(\gamma_i^{*m} - 1\right)}{\sum_{l=1}^{J} \frac{x_{l,T}}{\gamma_l^* - 1}\left(\gamma_l^{*m} - 1\right)} \tag{4.17}$$

and the stage upflow rate is

$$L_m = T \sum_{i=1}^{J} \frac{\gamma_i^* x_{i,T}}{\gamma_i^* - 1} \left(\gamma_i^{*m} - 1\right).$$

4.6.3 Putting Things Together: The Cascade

In Sections 4.6.1 and 4.6.2, the formulae for the enriching and stripping sections have been developed. The next step is to combine them such that the product and tails compositions and flows can be calculated given a certain feed. Please note that from here on, the stage $n = 1$ (or, equivalently $m = M+1$) is denoted *feed stage*, with a subscript $_F$ in mathematical expressions.

First, two expressions for $x''_{i,F}$, the concentration of component i of the downflowing stream of the cascade feed stage, are determined. On the one hand, it can be obtained by setting n to 1 in Eq. (4.16),

$$x''_{i,F} = \frac{x_{i,P}}{E_i} \sum_{l=1}^{J} E_l x''_{l,F}, \quad E_i = \frac{\gamma_i^* - 1}{1 - \gamma_i^{*-N}},$$

and on the other hand, one can set m set to $M + 1$ in Eq. (4.17) which yields:

$$x''_{i,F} = \frac{x_{i,T}}{S_i} \sum_{l=1}^{J} S_l x''_{l,F}, \quad S_i = \frac{\gamma_i^* - 1}{\gamma_i^{*M+1} - 1}.$$

Then, both expressions are equated and using the overall material balance of component i,

$$P x_{i,P} + T x_{i,T} = F x_{i,F}, \tag{4.18}$$

$x_{i,T}$ is eliminated from the equation, resulting in

$$x_{i,P} \left(P + T \frac{S_i \sum_{l=1}^{J} E_l x''_{l,F}}{E_i \sum_{l=1}^{J} S_l x''_{l,F}} \right) = F x_{i,F}. \tag{4.19}$$

Having introduced the feed composition into the expression, the next objective is to eliminate the summation terms. This is done by rewriting Eq. (4.19) for component j and for component k, and dividing them to obtain the following expression:

$$R_{j,P} \left(P + W \frac{S_j \sum_{l=1}^{J} E_l x''_{l,F}}{E_j \sum_{l=1}^{J} S_l x''_{l,F}} \right) = R_{j,F} \left(P + W \frac{S_k \sum_{l=1}^{J} E_l x''_{l,F}}{E_k \sum_{l=1}^{J} S_l x''_{l,F}} \right).$$

In the following step, the explicit forms of E_j, E_k, S_j and S_k are used[6] and it is taken into account that for the matched component $R_{j,P} = \beta_j^N R_{j,F}$ holds. This results in

$$\frac{\sum_{l=1}^{J} E_l x''_{l,F}}{\sum_{l=1}^{J} S_l x''_{l,F}} = \frac{P}{T},$$

[6]For more details, please consult the original paper [22], Eqs. (44) to (45).

which is substituted back into Eq. (4.19):

$$Px_{i,P}\left(1+\frac{S_i}{E_i}\right)=Fx_{i,F}. \tag{4.20}$$

After rearranging the terms, summing over all components and using the fact $\sum_{i=1}^{J} x_{i,P}$ is unity, one obtains:

$$\frac{P}{F}=\sum_{i=1}^{J}\frac{E_i x_{i,F}}{E_i+S_i}. \tag{4.21}$$

This result is the first of the four desired expressions. Given a feed quantity and composition, the amount of enriched material can now be calculated. At first glance, the amount of produced material seems to be independent of its enrichment level $x_{j,P}$. However, this is not the case, since higher enrichment levels require a large number of stages in the enriching section, meaning that E_i will decrease.

The second desired expression, the composition of the product, then follows directly by resubstituting Eq. (4.21) in Eq. (4.20):

$$x_{i,P}=\frac{E_i x_{i,F}}{E_i+S_i}\Bigg/\sum_{i=1}^{J}\frac{E_i x_{i,F}}{E_i+S_i}\ . \tag{4.22}$$

The third expression, the tails quantity, is obtained using Eq. (4.21) and the conservation of total mass:

$$\begin{aligned} &T=F-P\\ \Rightarrow\ &T=F\left(1-\sum_{i=1}^{J}\frac{E_i x_{i,F}}{E_i+S_i}\right)=F\sum_{i=1}^{J}\frac{S_i x_{i,F}}{E_i+S_i}. \end{aligned} \tag{4.23}$$

In the last step, $\sum_{i=1}^{J} x_{i,F}=1$ is used.

Taking into account the conservation of mass per component, see Eq. (4.18), and the above results, Eqs. (4.21) to (4.23), the tails composition is given by

$$x_{i,T}=\frac{S_i x_{i,F}}{E_i+S_i}\Bigg/\sum_{i=1}^{J}\frac{S_i x_{i,F}}{E_i+S_i}\ . \tag{4.24}$$

4.6.4 Separative Work

In Sections 4.6.1 to 4.6.3 the formulae needed for the calculation of an enrichment process have been developed. However, while knowing *how* to calculate an enrichment process, it is not yet clear *how much* uranium can be enriched. Currently, the only limit one could encounter is a constraint on the available feed material. It should be obvious that—additionally—a constraint based on the quality and quantity of available separating elements can exist.[7]

Hence, a criterion for the "difficulty" of an enrichment needs to be developed. Clearly, an enrichment process to higher enrichment levels or enriching larger

[7] In other words: a quantity is needed that allows the differentiation between a facility with 10 old centrifuges from 1970 and a facility with 1000 brand-new centrifuges from 2020.

amounts should be more "difficult". First steps towards such a criterion consisted of using an entropy-like quantity, the idea being that separating an isotopic mixture is equivalent to reducing its disorder [35].

Later, this got replaced with the *value* (or: the *separative work*) of an enrichment process ΔU. ΔU should have the property of being proportional to the number of separating elements needed for an enrichment, each element with separative power δU. By assigning each uranium mixture a value U proportional to its mass and enrichment level, the separative work can be calculated as a balance:

$$\Delta U = U_P + U_T - U_F = PV(x_P) + TV(x_T) - FV(x_F). \tag{4.25}$$

Here, the proportionality of U to the enrichment level x is reflected in $V(x)$, the *elementary value function* [2, 35].

For the purpose of illustration, consider the elementary value function for a binary uranium mixture in an ideal cascade,

$$V(x) = (2x - 1) \ln \frac{x}{1 - x}. \tag{4.26}$$

In this case, the enrichment of 1 kg of natural uranium to 3% LEU uses a separative work of around 0.52 kg SWU (using Eqs. (4.25) and (4.26), and 0.3% tails enrichment). However, the enrichment of the same feed to 20% HEU would require 0.80 kg SWU and yield only about a seventh of the product quantity. Here 'kg SWU' is the unit of the separative work (and of the separative power, as well). While both have the dimension of mass, they represent the amount of separative work performed, as indicated by the name 'separative work unit (SWU)'. Note that Eq. (4.25) holds for every enrichment, be it the enrichment of a binary or of a multi-component mixture. However, the definition of V might change—if a definition exists, at all!

Currently, no agreed-upon elementary value function exists for the case of multi-isotope mixtures separation because the no-mixing condition cannot be fulfilled [63]. One approach is to use a multi-component extension of Eqs. (4.25) and (4.26) such that

$$V = \sum_{i=1}^{J} \left(\frac{x_i}{2a_i - 1} \right) \ln R_j \quad \text{with } a_i = \frac{\gamma_i - 1}{\gamma_j - 1} \neq \frac{1}{2}. \tag{4.27}$$

In the case of $k_i = \frac{1}{2}$, this summation term gets substituted by $\ln(R_i)$ [9, 10]. An advantage of this method is that it remains consistent with the binary expression[8], however its (possible) drawback is that it was developed in a theory limited to small enrichment factors, $\gamma \gtrsim 1$. It is unclear if this limitation is only relevant to the development of the enrichment formulae (product and tails composition and quantities) or if it is also used during the derivation of the elementary value function. However, it has been used previously see, e.g., [9, 10, 23] an it will also be used in this thesis.

4.7 The MARC in Practice

The MARC presented previously needs multiple input variables which can be loosely grouped into two categories. The first set of variables are those fixed by

[8] To prove this, set $J = 2$, use $a_j = 0, a_k = 1$ and the normalisation of concentrations.

the enrichment circumstances and this set contains:

- the overall separation factor $\vec{\gamma}$,
- the cascade feed composition $\vec{x}_F$,
- the feed amount F that one has at their disposal, and
- the SWU capacity that is available.

The second group of variables all get set by the user and they are:

- the desired product enrichment level $\tilde{x}_{235,P}$,
- the desired amount of product P and
- the desired tails enrichment level $\tilde{x}_{235,T}$.

Additionally, given the number of stages in the enriching section, N, and the number of stages in the stripping section, M, one can then calculate the enrichments parameters using MARC. More precisely, these parameters are the product composition and quantity using Eqs. (4.21) and (4.22) and the tails composition and quantity using Eqs. (4.23) and (4.24). However, MARC does not include a method on how N and M can be calculated.

To this end, I have developed an algorithm that takes the aforementioned two categories of variables as input, then determines N and M through minimisation and finally calculates the enrichment parameters. The algorithm starts with random initial values for N and M (and the input variables) and calculates the resulting enrichment parameters. Then, the minimisation procedure begins: the algorithm determines N and M such that $f(N, M)$ gets minimised. f is defined as the quadratic sum of the relative differences between desired U-235 concentration and actual U-235 concentration in both product and tails:

$$f(N,M) = \sqrt{\left(\frac{x_{235,P} - \tilde{x}_{235,P}}{\tilde{x}_{235,P}}\right)^2 + \left(\frac{x_{235,T} - \tilde{x}_{235,T}}{\tilde{x}_{235,T}}\right)^2}. \tag{4.28}$$

In practice, a two-dimensional minimisation is used, meaning that N and M are determined simultaneously. More precisily, the minimisation is done in a Python script using SciPy's implementation of the L-BFGS-B algorithm[9] [6, 50, 60, 65].

Unfortunately, as enrichment technologies are mostly classified, there was no experimental data to compare or validate the algorithm. Because of this, the algorithm results have been compared to the few publicly available research results. Note that in the following, I do not look at the data to gain knowledge on enrichment or nuclear archaeology. Rather, I exclusively compare my results with literature.

The first publication used is by Matthew Sharp, where Figs. 1 and 5 (top left) have been reproduced [51]. As no data sets have been available, I have digitised the figures. Then, the data has been reproduced with the MARC algorithm by calculating the value corresponding to each abscissa from the data set.

[9] Short for *Limited-memory Broyden–Fletcher–Goldfarb–Shanno* B algorithm

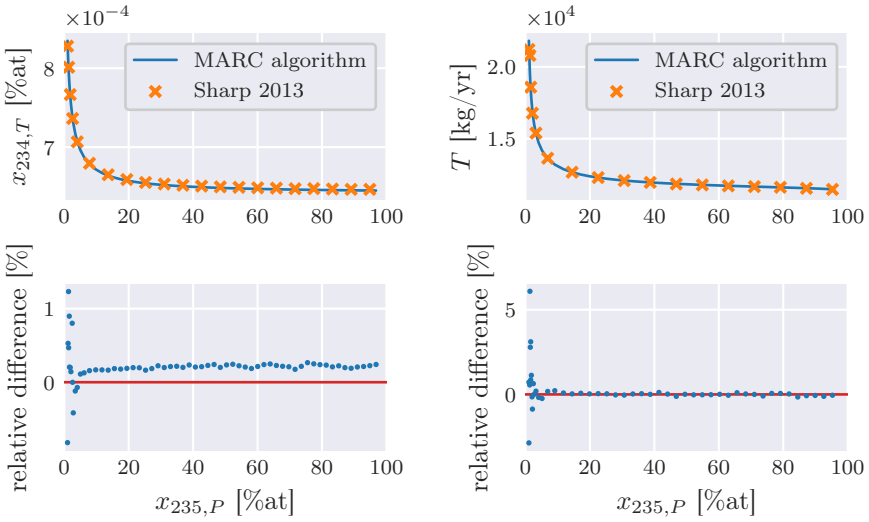


Figure 4.5: Comparison of the MARC algorithm developed in this thesis to [51]. Overall, the calculated values are in good agreement with Sharp's data. However, in both cases fluctuations resulting from the digitisation process of Sharp's data are observed for product enrichments smaller than 5%at. Additionally, the calculated MARC $x_{234,T}$ values are systematically smaller by about 0.2%, the reason for which is unknown. Note that to improve visibility, only every third data point of Sharp has been plotted in the upper graphs. However, the lower graphs shows all data points.

The result of the first reproduction is shown the upper left panel in Fig. 4.5. It shows a good overall accordance of the calculated MARC data and the data from Sharp. The lower left panel shows the relative difference between both datasets:

$$\text{relative difference } [\%] = \frac{x_{234,T,\text{Sharp}} - x_{234,T,\text{MARC}}}{x_{234,T,\text{Sharp}}} \times 100. \tag{4.29}$$

Here, two interesting effects are observed. First, in the range of $x_{235,P}$ smaller than 5%at, there are strong fluctuations. This effect is most probably due to the digitisation process failing in this steep region of the graph. More precisely, the (digitised) x values that I use as MARC input all bear a statistical error that propagates onto the MARC reconstructed MARC values. The second effect is that for higher product enrichment ranges, above 10%, an offset of about 0.25% is observed. The reason for this shift is unknown. One possible explanation could be that Sharp does not use the general MARC theory as introduced in Section 4.6. Instead he uses a MARC approximation only valid for small separation factors as developed by de la Garza, see [9, 10].

The comparison done in the right panels in Fig. 4.5 show the same digitisation effect in the lower ranges. However, no systematic offset is observed in the higher ranges and the values are distributed evenly around 0. The relative difference has been calculated analogous to Eq. (4.29).

Table 4.3: Comparison of the calculated product concentrations to literature values for the enrichment of reprocessed uranium to LEU and HEU using gas centrifuges [62].

product	isotope	MARC algorithm [%]	Wood 2008 [%]	rel. diff. [%]
LEU	U-232	7.15×10^{-9}	7.26×10^{-9}	1.57
LEU	U-234	0.128	0.129	0.58
LEU	U-235	5	5	0
LEU	U-236	1.68	1.67	−0.14
HEU	U-232	1.13×10^{-9}	1.15×10^{-9}	2.02
HEU	U-234	2.00	2.02	0.95
HEU	U-235	75	75	0
HEU	U-236	19.69	19.34	−1.81

The next comparison is done using numerical results from Houston Wood, [62]. In this publication, Wood calculates the product compositions for the enrichment of natural uranium or reprocessed uranium to LEU or HEU. Additionally, these four cases are calculated twice, once using a small separation factor (corresponding to enrichment by gaseous diffusion) and once using a large separation factor (corresponding to enrichment with gas centrifuges).

As an example, the results of the enrichment of reprocessed uranium with gas centrifuges are shown in Table 4.3. The results of the remaining cases are listed in Appendix B, Tables B.1 to B.3. All values calculated with the MARC are within a 2% range of the original values and no systematic shift or similar can be found. The largest deviations are found in the enrichment of reprocessed uranium using gas centrifuges, whereas in the gaseous diffusion case, the deviations are smaller than a per mil. The origin of these deviations is unknown. Contrary to Sharp, Wood has used the general MARC theory used in this thesis, as well. However, no details are given on how the computer program calculates the enrichment parameters.

4.8 Using Enrichment for Nuclear Archaeology

Having presented the different uranium isotopes, enrichment and the MARC theory, I now give an example for reconstructing an enrichment process with nuclear archaeology methods. Here, an enrichment facility using a known feed composition and a known separation factor is considered and the objective is to reconstruct the produced material, i.e., its enrichment level and quantity.

A first step is to gain information on the product's enrichment level. Using Eq. (4.24) allows calculating the tails composition given a product enrichment level and a known feed, see Fig. 4.6. By measuring the U-234 content in the tails, one can readily determine if the product has been enriched to LEU or HEU due to the distinct levelling out of the curve in the region of $x_{235,P}$ equal to 10 to 20%. Moreover, commonly used methods for measuring isotopic compositions, such as thermal ionisation mass spectrometry (TIMS) and inductively coupled plasma mass spectrometry (ICP-MS), have a mass resolution of a per mil or better [40]. Hence, given feed and tails compositions, determining the product assay remains precise even in the HEU regime where $x_{234,T}$ changes only slightly

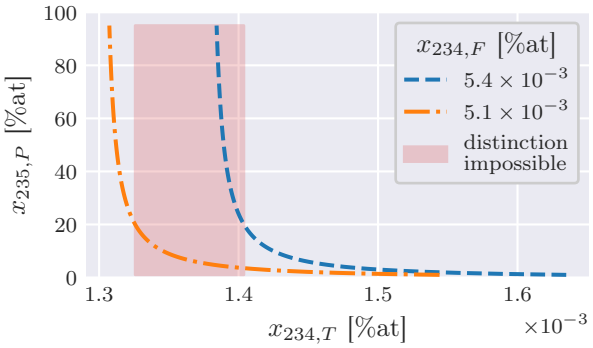


Figure 4.6: Reconstruction of the product enrichment from the tails' U-234 content for different feed compositions. The feed composition strongly influences the reconstruction to the point that the distinction between LEU and HEU can be become impossible for an unknown natural feed composition, as shown by the red box. The feed and tails assays are fixed to 0.72% and 0.3%, respectively.

with $x_{235,P}$.

However, one major deficiency becomes apparent which is the importance of knowing the feed composition *precisely*. If one takes into account concentration fluctuations of U-234 in natural uranium as indicated previously in Table 4.2, the distinction between LEU and HEU is not possible with this method, as shown by the orange dash-dotted curve and the region shaded in red in Fig. 4.6. For example, a tails measurement of $x_{234,T} = 1.385 \times 10^{-3}\%$ could correspond to a product enrichment anywhere between 4.8% to 90% in case of an unknown (natural uranium) feed composition. This can be countered by taking measurements of available material of similar origin, such as uranium from the same mine, as has been explained in Section 4.1

Assuming that the product assay has been successfully determined, the next reconstruction step is to investigate the amount of enriched uranium. Here, Eqs. (4.21) and (4.23) are of help: both the tails quantity and the feed composition are known or at least they can be estimated and thus the amount of product can be calculated. Importantly, these calculations are completely independent of the amount of feed that was used. The result of applying these steps is shown in Fig. 4.7. The amount of enrichment product is proportional to the tails quantity with the proportionality constant depending on the product enrichment assay. Among others, it can also be determined if one or more significant quantities of uranium have been produced, which is a crucial threshold. One significant quantity of fissile material is the quantity 'for which the possibility of manufacturing a nuclear explosive device cannot be excluded' [26].

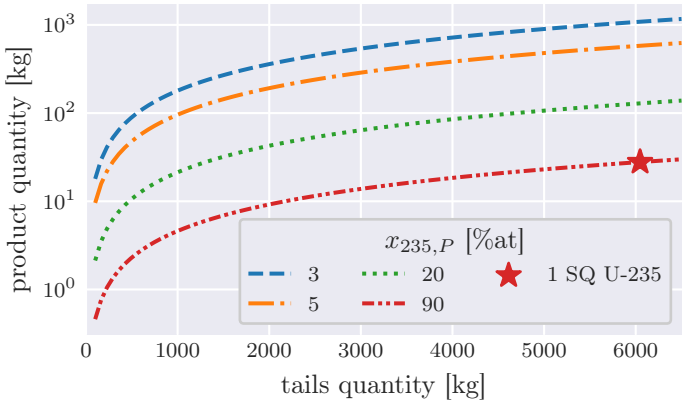


Figure 4.7: Amount of enriched product given the tails quantity for different product assays. Being able to differ between LEU and HEU allows a first estimate of the produced amounts while a precise knowledge of the assay enables an exact determination of the product quantity. The red star corresponds to one significant quantity of uranium. The tails assay is fixed to 0.3% and the natural uranium composition used in these calculations is indicated in Table 4.2.

Chapter 5

Misoenrichment

In Chapter 2, the importance of the enrichment facility to both the civilian and military NFC has been presented. Because of this, a precise modelling of the enrichment process in Cyclus is elementary. However, currently existing enrichment modules all have some shortcomings making them less suitable for the use in a nuclear archaeology context. The enrichment facilities of the modules Cycamore and Mbmore do not support multi-isotope enrichment [56, 59]. In contrast, Cycvt allows multi-component isotope enrichment. However, it does so by approximating the minor isotopes enrichment with user-defined enrichment efficiencies relative to the U-235 enrichment. While the motivation behind this module is identical to ours (i.e., doing nuclear archaeology), the calculations used are estimates yielding only limited accuracy [21, 42].

To fill this void, a new Cyclus module called MIsoEnrichment has been developed as part of this thesis. In the following, the module MIsoEnrichment, short for multi-isotope enrichment, and its Cyclus facility `MIsoEnrich` are presented.[1] Section 5.1 introduces the goals of the new module, followed by the actual implementation described in Section 5.2. Section 5.3 explains how the code is verified and tested and Section 5.4 gives a brief evaluation of the module.

5.1 Objectives

The main goal of MIsoEnrichment is to accurately model multi-component isotope enrichment in Cyclus. To this end, the MARC theory by von Halle is chosen and implemented based on the MARC algorithm introduced in Section 4.7. This allows to calculate the enrichment parameters for both enrichment by gas centrifuge and enrichment by gaseous diffusion.

In the MARC theory, the number of stages in an enrichment cascade can be decimals. Clearly, this is not possible in reality. To account for this, a downblending mechanism is added to the facility. Here, the enrichment product gets mixed with parts of the feed to obtain exactly the desired product enrichment level.

[1] Please note the difference between the *module* MIsoEnrichment and the *facility* `MIsoEnrich`. Whereas the facility is actually taking part in the simulations, the module is the library containing the facility and it could potentially also contain other facilities.

Another important aspect is that the facility must be able to dynamically change its configuration during the simulation. More precisely, when the user defines the simulation setup in the CYCLUS input file, then `MIsoEnrich`'s parameters should be time-dependent if desired. This allows the user to account for changes in an enrichment's facility operation mode. For example, the facility's SWU capacity may increase during the simulation or the user wants that, from a certain time step on, the facility only produces LEU.

Finally, the facility should flexibly choose its feed material, meaning that the feed need not have exactly one precise composition. Of course the facility should handle the changing feed material correctly, i.e., not mix it with leftover feed of a different composition.

5.2 Implementation

Currently, the MISOENRICHMENT module contains only one agent, `MIsoEnrich` , and related routines and subclasses. Roughly speaking, `MIsoEnrich` is subdivided in two parts. The first part can be viewed as the agent's front end which handles everything related to CYCLUS and the simulation itself. It is based on CYCAMORE's enrichment facility as the latter is well-tested and has gone through a lot of debugging. The front end is responsible for setting up the facility and for storing the variables defined in the simulation input file, it performs the communication and trading with the DRE and it stores, hands out and enriches the uranium.

In contrast to this is the facility's back end which contains the enrichment calculator. The whole MARC theory related calculations are outsourced to this calculator. This structure has the advantage that enrichment theory and CYCLUS simulation are clearly separated from each other. Front and back end interact with each through two functions: one passing the MARC input parameters to the calculator and one returning the calculated enrichment parameters to the agent's front end.

In the following, the different classes and their interactions are presented in detail. The nomenclature of mathematical variables follows Chapter 4.

MIsoEnrich To begin with, `MIsoEnrich`, the facility's main class and the front end, is presented. When it enters the simulation, the variables defined in the input file are set. One category of particularly important variables are the overall stage separation factors. Here, the user indicates γ_{235} and the γ_i then get calculated using Eq. (4.6). In the same step, the facility makes use of the `FlexibleInput` feature which is currently implemented for the SWU capacity. This allows the user to define a list of SWU capacities together with the corresponding number of time steps (as opposed to setting only one SWU capacity that remains constant throughout the simulation). At these time steps, the SWU capacity will be updated accordingly.

Following the simulation entry, the facility start the regular cycle of `Tick`-DRE-`Tock` until it gets deleted. In the `Tick` phase, the only action done is that `FlexibleInput` updates the SWU capacity if needed.

Subsequently, the DRE starts. The facility requests as much feed material as it can accommodate in its inventory. Then, if it obtains a request for enriched material, it extracts the relevant information, this being the desired

enrichment grade and the amount of material requested. In combination with the available feed material and composition, this is passed to the agent's back end, the `EnrichmentCalculator`. The calculator determines the MARC enrichment parameters such as feed used, product composition etc. and returns theses variables to the facility. The exact functioning of the calculator is explained later. Using these information, the facility creates a bid for the request.

If one or more of the facility's bids get accepted, the facility removes the corresponding amount of material from the feed, enriches it (instantaneously during the DRE) and sends it to the requesters. Then, upon obtaining new feed material, the facility checks its composition and stores the material in the inventory containing the same material. If the composition is unknown, meaning that no matching inventory exists, then a new one is created.

Last, the facility enters the `Tock` phase. There, the performed enrichments are summed up and stored in the log of the facility.

EnrichmentCalculator This class is the facility's brain. When a `MIsoEnrich` facility performs an enrichment or offers enriched material, it passes the relevant enrichment parameters (e.g., desired enrichment level) to the calculator (i.e., `EnrichmentCalculator`) which then returns the calculated values.

After having set the parameters, the class first calculates the number of enriching and stripping stages N and M, respectively. In von Halle's MARC theory, N and M are supposed to be known, which is not the case here. N gets determined iteratively by incrementing it by one per step, then calculating the product and tails concentrations using Eqs. (4.22) and (4.24) until the desired enrichment level is reached or exceeded. The number of stripping stages M is determined analogously.

After having determined the number of stages as well as the concentrations, the material flows are calculated. Here, it is taken into account that either F, P or the SWU capacity can be limiting factors. First, the feed needed to produce the desired product quantity is calculated. Independently of this, the product that could be produced with the available feed quantity is calculated as well. Both steps use Eq. (4.21). From there, it is checked whether the available feed is the limiting quantity and both F and P are calculated accordingly. Then, the tails quantity is calculated following Eq. (4.23).

At this point, a possible SWU constraint is considered by calculating the SWUs of the enrichment process using Eqs. (4.25) and (4.27). Then, if it is indeed a constraint, Eq. (4.25) is solved for F and the feed quantity is reduced such that all of the available SWU gets used. Subsequently, the product and tails quantities are recalculated using the updated value of F.

The last optional step in the calculations is the *downblending* of the product. Because enrichment is done in discrete stages, the achievable enrichment levels are discrete, as well. To allow reaching other enrichment levels, a downblending method has been implemented which mixes feed and the (over-enriched) product to lower the enrichment to the desired level. It should be noted that it is not known if this method is used in practice[2]. In the following, the method and formulae used for the downblending are presented.

[2] Another possibility (used in practice, not in MIsoEnrichment) could be to run centrifuges in a non-optimal manner, thereby lowering the separation factor such the desired enrichment gets reached. However, MIsoEnrichment does not do a physical simulation of each centrifuge thus this procedure cannot be modelled.

Consider an enrichment with P units of product of actual enrichment level x_P and with a desired enrichment level $\tilde{x}_P < x_P$. To reach $\tilde{x}_P$, the product is downblended (*or* diluted) using F_{db} units of feed with enrichment level x_F. Using the conservation of U-235 mass during the mixing,

$$F_{\mathrm{db}} x_F + P x_P = \tilde{P} \tilde{x}_P,$$

with $\tilde{P}$ being the final downblended product quantity, and the conservation of total mass,

$$F_{\mathrm{db}} + P = \tilde{P}, \tag{5.1}$$

one obtains the amount of blending feed per product Δx:

$$\Delta x = \frac{F_{\mathrm{db}}}{P} = \frac{x_P - \tilde{x}_P}{\tilde{x}_P - x_F}. \tag{5.2}$$

One needs to bear in mind that constraints on feed or product can exist[3]. It needs to be ensured that the sum of enrichment feed material and downblending feed must not exceed the available feed quantity (the *feed constraint*). Analogously, the sum of downblending feed and enrichment product must not exceed the requested product quantity, else the simulation fails (the *product constraint*). These two constraints are handled separately. In case of a product constraint, P needs to be reduced such that after downblending x_P equals $\tilde{x}_P$ and such that Eq. (5.1) holds. This is achieved by substituting F_{db} from Eq. (5.2) in Eq. (5.1), yielding

$$P = \frac{\tilde{P}}{1 + \Delta x}.$$

In case of a feed constraint, the available feed $\tilde{F}$ needs be split into an enrichment feed F and a downblending feed F_{db}. Substituting P in Eq. (5.2) with Eq. (4.21) and using the resulting expression for F_{db} in

$$F + F_{\mathrm{db}} = \tilde{F} \tag{5.3}$$

gives the final result:

$$F = \tilde{F} \Bigg/ \left(1 + \Delta x \sum_{i=1}^{J} \frac{E_i x_{i,F}}{E_i + S_i} \right).$$

Having calculated the required amount of downblending feed, the last step in the code is the actual downblending. Here, the final product composition is calculated

$$\tilde{x}_{i,P} = \frac{F_{\mathrm{db}} x_F + P x_P}{F_{\mathrm{db}} + P}$$

as well as the final feed and product quantities $\tilde{F}$ and $\tilde{P}$ using Eqs. (5.1) and (5.3).

[3] A SWU constraint is not needed as downblending does not require separative work to be performed.

FlexibleInput Previously, I have explained that the facility's SWU capacity can be defined as a time-dependent variable. This is done using the `FlexibleInput` class. It stores and updates the parameters as needed and, upon startup, performs input checks. The advantage of this class is that it is conceptually completely independent of `MIsoEnrich`. Thereby, it can potentially be included and implemented in any other facility as well, or it can be used in `MIsoEnrich` for input parameters other than the SWU capacity.

5.3 Verifying and Testing the Code's Behaviour

Unit tests allow checking the correct and expected behaviour of code snippets—be this a class or a single function. Following Cyclus, the unit tests in MIsoEnrichment are embedded in the Google Test framework [20].

Currently, MIsoEnrichment contains 40 unit tests of which 15 are inherited from Cyclus classes and the rest being specific tests to this implementation. While `MIsoEnrich` unit tests mainly check the correct ordering and bidding of resources and special cases where errors are thrown, the `EnrichmentCalculator` tests mostly compare calculated enrichment parameters with values calculated using the MARC algorithm introduced in Section 4.7. This is sufficient as the algorithm used here is *as such* analogous to the one from Section 4.7. Thus, the unit tests only serve to verify the integration in C++ and Cyclus. The `FlexibleInput` tests assure the correct behaviour upon instantiation, meaning they check that errors are only thrown in the expected situations (i.e., when the input is corrupted).

5.4 Evaluation

In this section, MIsoEnrichment's strengths, but also shortcomings and future work, are discussed.

Using von Halle's MARC theory has both advantages and disadvantages. Its major strength is the quick and precise calculation of enrichment parameters without resorting to numerical methods of limited precision (integration, differentiation) or Monte-Carlo simulations. Additionally, it is suited for calculating enrichment processes with higher separation factors such as gas centrifugation.

The downside to this method is that it does not include SWU calculations, meaning that other methods have to be adapted. Possibly, an alternative is given in [54, 63] and this should be investigated in the future. However, these advantages and disadvantages are inherent of von Halle's MARC method and are *not* implementation-specific.

A MIsoEnrichment-specific feature that is still in development is the existence of multiple feed inventories which have been introduced in Section 5.2. Currently, when a feed material is accepted and it has a feed composition different from previous ones, it is added to a new inventory. This inventory then becomes the "active" one and remains the active one until a feed material with different composition is accepted. Effectively, this means that the inventory is not dynamically chosen and if the active one is empty, then no enrichment is performed despite other feed inventories possibly being filled. This behaviour does not influence the tests or case study shown in this work, but it is non-

etheless counterintuitive and if one is unaware of this, it can lead to erroneous results. Improvements of this are to be expected in later versions.

The current version calculates the separation factors using Eq. (4.6), which is developed for enrichment by gas centrifuges. Future versions of MIsoEnrichment will enable the user to choose between enrichment by gas centrifuge or by gaseous diffusion. In the latter case, the separation factors will be calculated using the appropriate formula, see Eq. (4.5).

Chapter 6

Case Study: Republic of Leonia

In Chapter 5, the development, implementation and testing of MIsoEnrichment has been laid out. While the unit tests are sufficient to verify the correct behaviour of the module, they do not show the benefits or the usage of this module. For this reason, a case study in the fictitious Republic of Leonia is constructed and two cases are simulated using Cyclus and MIsoEnrichment.

Section 6.1 introduces the hypothetical country, its military nuclear programme and the fuel cycle it uses. Following this is Section 6.2 which explains how this NFC is implemented in Cyclus and which assumptions are used. In order to check the simulation's behaviour and the Cyclus output files, cross-check calculations are performed in Section 6.3. Finally, a reconstruction of the fissile material production using nuclear archaeology methods is performed in Section 6.4.

6.1 Overview

In this case study, the following scenario has been considered: the Republic of Leonia possesses a military nuclear programme in which it produces both plutonium and weapon-grade uranium. The complete fuel cycle is shown in Fig. 6.1. It starts with the mining of natural uranium. Then, the uranium gets enriched, serving two purposes. On the one hand, it is used as nuclear fuel for plutonium production reactor and on the other hand, it is enriched further to produce weapon-grade uranium. The reactor's spent fuel is separated into plutonium, uranium and nuclear waste (i.e., the fission products of U-235 and Pu-239). In order to save natural uranium, Leonia enriches reprocessed uranium once again and reuses it as reactor fuel. It is not used for the production of nuclear weapons and it is only reprocessed once. After the second irradiation cycle and the removal of plutonium, it is sent to a storage site. Leonia focuses on the production of plutonium, meaning that cuts in the HEU production can be made in order to keep the reactor running.

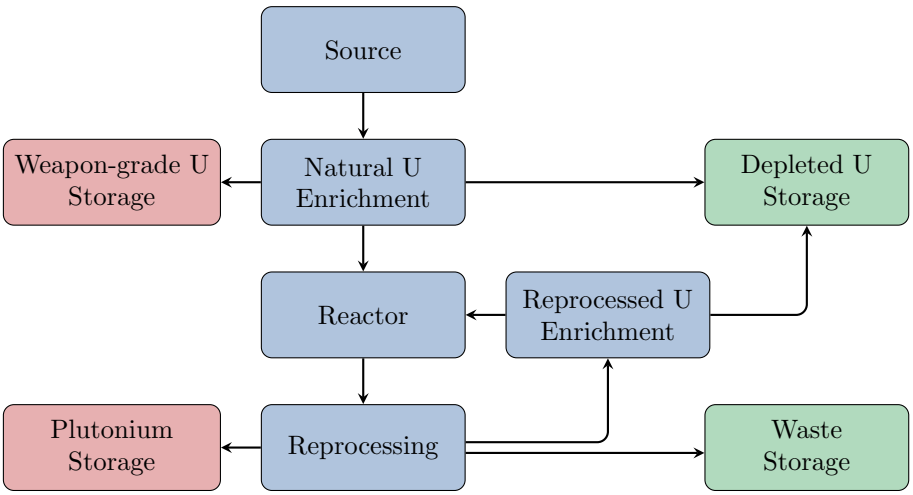


Figure 6.1: Scheme of Leonia's NFC. Shown in the left column in red are the fissile material storages, while the right column, in green, contains the waste and tails repositories. Spent fuel from the reactor is reprocessed and is subsequently reused *at most once* as reactor fuel. After this second irradiation, plutonium is again separated and the rest is sent to waste storages. Not shown here are two storages that each hold one spare reactor fuel assembly. In one case, it is obtained from enriching natural uranium, in the other case from the enrichment of reprocessed uranium.

6.2 Cyclus Simulation Setup

In this section, details and numbers about the implementation of the fuel cycle in Cyclus are given. All facilities are taken from the Cycamore library [56], except for both enrichment facilities which use the MIsoEnrichment module. Also, the difference between both simulation cases is explained in the paragraph *Reactor*.

Source The uranium source produces 10 t of natural uranium per day. Its composition is indicated in Table 4.2.

Enrichment Two enrichment facilities are used, one taking natural uranium as feed and the other one enriching reprocessed uranium. The natural uranium is enriched either to reactor-grade, 1.1%at, or to weapon-grade, 90%at. In contrast, the reprocessed uranium is only enriched to reactor-grade level, 1.1%at. Both facilities have no SWU limitations, they are only limited by the feed available.

Storages Two storage facilities are deployed, one associated to each enrichment facility. Both take the respective reactor-grade uranium as input and store it. They each have the capacity to hold the uranium needed for one complete reactor core. Their deployment is necessary because of the way Cycamore's reactor works but they do not have a particular role but to serve as reactor fuel storage.

Reactor The nuclear reactor is a heavy water reactor of the same type as the American Savannah River Site reactors. Information on these reactors can be found for example in [47]. It is run at a constant power level of 2496 $\mathrm{MW_t}$. Two simulations are performed, each with a different fuel burnup: 0.5 MWd/kg and 2 MWd/kg.[1] As the burnup—the total released energy—is the product of the power and the irradiation time, the latter has to be adapted as well. The times are 22 days and 88 days, respectively. In both cases, the refuelling takes 6 days and during refuelling the whole core is exchanged [47]. The reactor output, i.e., the spent fuel composition, has been calculated using the burnup calculation code Serpent 2 following the methodology presented in [12, 13, 36]. The uranium and plutonium components of the spent fuel are indicated in the appendix in Table A.1.

Reprocessing The separation of spent fuel into uranium, plutonium and nuclear waste assumes a constant separation efficiency of 97% for all elements and isotopes, which is a standard literature value [45].

Sink The sinks are material 'containers', facilities that accept material but do not offer it. Real-life counterparts would be, e.g., deep geological repositories. They are unlimited in capacity and unlimited in throughput.

It should be noted that, for simplicity, in these simulations uranium is used in its pure form, meaning that no conversions to UF_6 or to a specific reactor fuel compound (e.g., uranium dioxide (UOX) or mixed oxide (MOX)) is taken into account.

6.3 Consistency Checks

In order to increase trust in the simulation, some expected and simulated metrics should be compared. These expected metrics should be calculated relying *exclusively* on values taken from the input files and should not use any information from the output file. As Leonia's fuel cycle is limited in size, such calculations can be performed.[2] In the following, I will first explain how these expected values are determined, which information is used, then present all the results and compare them to the simulated values.

Completed reactor cycles First, the number of completed reactor cycles in the simulation, n_{cycles}, is calculated.

For this, one needs to know the time at which the first reactor cycle starts. It is different from the beginning of the simulation because initially, enough uranium needs to be enriched to supply fuel for a complete reactor core. This time, denoted t_{enrich}, is obtained as follows. Using Eq. (4.21) from the MARC theory, the feed rate and feed composition from the input file gives a production rate P of about 5290 kg of SEU per day. Together with the known core mass, 110.82 t, this yields

$$t_{\mathrm{enrich}} = \frac{m_{\mathrm{core}}}{P} = 20.95\,\mathrm{d} \tag{6.1}$$

[1] A low burnup is typical of plutonium production in a reactor. At low burnup, it does produce less plutonium but the plutonium contains mostly Pu-239 and it is weapon-grade.

[2] In contrast, for large fuel cycles, this might well become a gargantuan task.

As the simulation runs in discrete time steps, t_{enrich} is rounded to 21.

The irradiation time, t_{irr}, and the refuelling time, t_{refuel}, are known from the input file. Thus all information necessary to calculate n_{cycles} is now avaible:

$$n_{\text{cycles}} = \left\lfloor \frac{t_{\text{sim}} - t_{\text{enrich}}}{t_{\text{irr}} + t_{\text{refuel}}} \right\rfloor,$$

where $\lfloor \cdot \rfloor$ denotes the floor function, meaning that the result is rounded to the next-lower integer. Thus, the result corresponds to the number of *complete* reactor cycles.

Fresh fuel and reprocessed fuel reactor cycles Subsequently, the number of reactor cycles is subdivided into the number of cycles using fuel stemming from natural uranium (*fresh fuel cycles*), n_{nat}, and those using fuel from reprocessed uranium (*reprocessed fuel cycles*), n_{rep}. These values are of particular interest because they are influenced by *every* actor in the simulation but the source and final repositories. Because of this, several quantities need to be determined before getting the final result.

In a first step, I calculate the amount of reprocessed fuel m_{rep} obtained from one batch of irradiated fresh fuel. m_{rep} depends on the mass of the fully loaded reactor core m_{core}, the separation efficiency ε_{sep} of the reprocessing plant and the uranium percentage in spent fuel x_{U}. Additionally, the reprocessed uranium needs to be re-enriched meaning that only a fraction $\varepsilon_{\text{enrich}}$ ends up as reprocessed fuel. $\varepsilon_{\text{enrich}}$ is calculated using Eq. (4.21) and the remaining variables are known from the input file, yielding

$$m_{\text{rep}} = m_{\text{core}} x_{\text{U}} \varepsilon_{\text{sep}} \varepsilon_{\text{enrich}}. \tag{6.2}$$

An iterative approach is used to determine both n_{nat} and n_{rep}. Consider a variable m_{storage} that indicates the amount of reprocessed fuel available. At the beginning of the iteration (corresponding to the start of the simulation) it equals 0. Hence, the reactor must use fresh fuel, meaning that n_{nat} is incremented by one and m_{storage} is incremented by m_{rep}. Then, it is checked if m_{rep} is larger than the core mass. If so, n_{rep} gets increased by one and m_{core} is subtracted from m_{storage}, else the algorithm falls back to the fresh fuel case. This iteration is repeated while $n_{\text{nat}} + n_{\text{rep}}$ is smaller than n_{cycles} and then the results are returned. Additionally, the program indicates if the last incomplete cycle would have used fresh or reprocessed fuel. This last output is needed in later consistency-checks.

Plutonium production Next, the amount of produced plutonium is checked. Here, the number of reactor cycles, the separation efficiency and the spent fuel composition have been considered. In the spent fuel, the relevant elements are plutonium and Np-239 at concentrations x_{Pu} and $x_{\text{Np-239}}$. Np-239 needs to be considered because it is present in non-negligible concentration and it decays into Pu-239 with a half-life of 2.356 days. The time step between unloading the spent fuel from the reactor to separating the plutonium from the spent fuel is one day and it is taken into account in the calculations. The amount of plutonium is then calculated as follows:

$$m_{\text{Pu}} = m_{\text{core}} (x_{\text{Pu}} + x_{\text{Np-239}} \mathrm{e}^{-1/2.356}) \varepsilon_{\text{sep}}.$$

Weapon-grade uranium production The penultimate check is performed on the weapon-grade uranium production. Its total quantity m_{HEU} depends on the amount of HEU enriched per time step and on the time that uranium actually got enriched to weapon-grade level[3]:

$$m_{\mathrm{HEU}} = \frac{\Delta m_{\mathrm{HEU}}}{\Delta t} \Delta t_{\mathrm{HEU}}. \tag{6.3}$$

The first factor, the amount of HEU per time step, is directly obtained from the MARC theory, see Eq. (4.21), using the throughput of the source (which is 10 t/d) as feed quantity.

The second factor, the time available for HEU production, is obtained by subtracting the time needed to produce fresh reactor fuel from the total simulation time. For this, n_{nat} is needed among others. It has been determined in one of the previous consistency checks. Additionally, the aforementioned last, incomplete reactor cycle needs to be taken into account, as the amount of weapon-grade uranium is susceptible to it. In case said cycle runs on fresh fuel, additional enrichment capacities are used up thus unavailable for HEU production. This is indeed the case: it is found that in both burnup scenarios, the last reactor cycle uses natural uranium fuel. Finally, it needs to be considered that the one fresh fuel assembly is stored in the storage (see Section 6.2) 'between' the enrichment facility and the reactor. Clearly, this assembly has to be produced in the enrichment facility, as well. Using Eq. (6.1) together with these findings yields

$$\Delta t_{\mathrm{HEU}} = t_{\mathrm{sim}} - t_{\mathrm{enrich}}(n_{\mathrm{nat}} + 2). \tag{6.4}$$

Two things should be noted. First, the addition of 2 in the last factor stems from the explanations above, that is the incomplete reactor cycle and the spare fuel assembly. Second, $t_{\mathrm{enrich}} = 20.95\,\mathrm{d}$ is *not* rounded here. This is due to the fact the on the last enrichment day, only 95% of the enrichment facility's capacities are used for enriching to reactor-grade, meaning that 5% can be used for the enrichment to HEU. Using the calculated Δt_{HEU} together with the amount of HEU per time step in Eq. (6.3) yields the result.

Amount of spent uranium Last, the expected amount of spent uranium, $m_{\mathrm{spent\ U}}$, is calculated. More specifically, this is the uranium that is sent to a final repository as it got irradiated twice (once as fresh fuel and once as reprocessed fuel), then got separated from plutonium and other elements and got stored away.

Its calculation is straightforward when using the results from previous consistency checks. The number of reactor cycles using reprocessed fuel is known and the mass of uranium in one batch of spent reprocessed fuel, $m_{\mathrm{spent\ batch}}$, is calculated similar to Eq. (6.2):

$$m_{\mathrm{spent\ batch}} = m_{\mathrm{core}} x_{\mathrm{U}} \varepsilon_{\mathrm{sep}}.$$

Multiplying both values yields the result:

$$m_{\mathrm{spent\ U}} = n_{\mathrm{rep}} m_{\mathrm{spent\ batch}}.$$

[3]Please note that the terms *weapon-grade uranium* and *HEU* are used interchangeably here. In both cases, they mean 'uranium enriched to 90%at'.

Table 6.1: Comparison of the expected and simulated values of some quantities for the 0.5 MWd/kg and 2 MWd/kg simulations. All but the 0.5 MWd/kg HEU production are in agreement with each other. However, this deviation can be explained and is not an error in the simulation.

	0.5 MWd/kg		2 MWd/kg	
quantity	expected	simulated	expected	simulated
completed reactor cycles	25	25	7	7
fresh to reprocessed fuel	14:11	14:11	5:2	5:2
plutonium [kg]	1168	1168	1273	1274
HEU [kg]	18716	19094	27677	27677
spent uranium [t]	1181	1181	214	214

Using the above explained methodology to both simulation cases (i.e., with a burnup of 0.5 MWd/kg and of 2 MWd/kg, respectively) yields the results shown in Table 6.1. The expected number of completed reactor cycles as well as its subdivision into number of fresh fuel cycles and reprocessed fuel cycles all match the simulated values.

The expected and simulated values of produced plutonium differ by less than a per mil This can stem from the fact that I have only considered plutonium isotopes 239 to 241 and Np-239 in the calculations above. Additional contributions such as U-239 decaying twice into Pu-239 have been neglected as the plutonium and Np-239 dominate. Taking this into account, the values are in accordance to each other.

The results of the weapon-grade uranium production of the 2 MWd/kg case are in accordance while the 0.5 MWd/kg values differ by 2%. This difference can be explained by considering the last simulation time steps in more detail. There, a reactor cycle using fresh fuel is started, hence the fresh fuel storage has to be refilled and the enrichment facility has to enrich natural uranium to reactor-grade level. This is taken into account and has been explained before. However, the simulation ends 9 days *before* the facility has finished enriching sufficient uranium to reactor-grade level. In terms of Eq. (6.4), this means that the '+2' should rather be a number between 1 and 2. A confirmation of this explanation is obtained by considering the difference between expected and actual values which amounts to 378 kg HEU. Using Eq. (4.21), it can be calculated that the daily production rate of HEU is about 47 kg. In addition, I know that there are 8 full enrichment days missing plus the 'hybrid' day where only 2 kg of weapon-grade uranium are produced Taking this into account yields consistent results, $47\,\mathrm{kg/d} \times 8\,\mathrm{d} + 2\,\mathrm{kg} = 378\,\mathrm{kg}$, or in terms of the total produced HEU amounts:

$$18716\,\mathrm{kg} + 47\,\mathrm{kg/d} \times 8\,\mathrm{d} + 2\,\mathrm{kg} = 19094\,\mathrm{kg}.$$

The expected and simulated amounts of spent uranium are in accordance with the simulated values.

6.4 Reconstruction of the Nuclear Programme

In this section, one possible case demonstrating an integrated nuclear archaeology approach is given. It should be noted that throughout the example, I

assume that values are known exactly, i.e., measurements uncertainties and the like are not considered.

In the scenario, Leonia recognises the futility and potential dangers of nuclear weapons and it desires to join the TPNW. During the adherence process, it submits a comprehensive baseline declaration listing all of its nuclear activities from the past. In order to verify the declaration, authorities choose to reconstruct the weapon-grade uranium and plutonium production programme and compare these results to the declaration. For this purpose, I assume that measurements of the uranium enrichment tails and of the reactor's structural elements can be taken.

One method to check the plutonium production would be the application of the isotope ratio method. During reactor operation, impurities in the structural elements of the reactor change their isotopic composition. Measuring these isotopic ratios allows reconstructing the reactor operation parameters and inferring on the plutonium production. A detailed explanation of this can be found for example in [14]. Here, I assume that the reactor's specifics, such as its geometry, are known either from on-site inspections or from the declaration. Thus, one can additionally perform forward-simulations of the reactor to get consistency checks on the plutonium estimates.

Following this, the HEU production needs to be reconstructed. Leonia only has one source of uranium, hence there are only two enrichment cases possible: either natural uranium or reprocessed uranium is enriched. In both cases I assume that the feed compositions are known, which can be justified. First, the natural uranium composition can either be measured from leftover feed or using samples from the uranium source. Second, the reprocessed uranium composition can be calculated using reactor simulations with the known reactor operation parameters. An additional assumption is made: it is known that the product is enriched to either 1.1%at or 90%at. The former can be determined from the knowledge on the reactor, i.e., from simulations and inspections is is known that the reactor needs 1.1% enriched uranium for an ideal operation. The latter is an assumption. However, if it were wrong, this would show in the calculations and further investigation could be done.

In the following, the reconstruction method is described. As there are two different enrichment facilities, there are two different tails measurements, thus this method is applied to both cases separately. First, the tails, containing both tails stemming from enrichment to SEU and from enrichment to HEU, should be decomposed. To do so, the tails composition $\vec{x}_T$ for both enrichment to SEU and to HEU are calculated using Eq. (4.24). In combination with the measured tails composition $\vec{x}_{T,\mathrm{meas}}$, these results allow the decomposition by using the formula to calculate the mixing of two materials

$$x_{T,\mathrm{meas},i} = \left.\frac{T_{\mathrm{SEU}} x_{\mathrm{T,\ SEU}} + T_{\mathrm{HEU}} x_{\mathrm{T,\ HEU}}}{T_{\mathrm{SEU}} + T_{\mathrm{HEU}}}\right|_i ,$$

where T is the respective tails quantity, and by rearranging it to

$$\left.\frac{T_{\mathrm{SEU}}}{T_{\mathrm{HEU}}} = \frac{x_{T,\mathrm{HEU}} - x_{T,\mathrm{meas}}}{x_{T,\mathrm{meas}} - x_{T,\mathrm{SEU}}}\right|_i .$$

The absolute values for T_{SEU} and T_{HEU} are obtained using the measured total

Table 6.2: Reconstructed and simulated HEU production for both enrichment of natural and of reprocessed uranium. The values are in accordance to each other but for the 2 MWd/kg natural U case, where the deviation is an artefact of Cyclus and `MIsoEnrich`. It can be explained and is not a methodological error in the reconstruction.

burnup [MWd/kg]	enrichment feed	recon. HEU [kg]	actual HEU [kg]
0.5	natural U	19 094	19 094
0.5	reprocessed U	0	0
2	natural U	27 629	27 677
2	reprocessed U	0	0

tails mass T:

$$T_{\text{SEU}} = \frac{T \times \frac{T_{\text{SEU}}}{T_{\text{HEU}}}}{1 + \frac{T_{\text{SEU}}}{T_{\text{HEU}}}}, \quad T_{\text{HEU}} = \frac{T}{1 + \frac{T_{\text{SEU}}}{T_{\text{HEU}}}}. \tag{6.5}$$

Note that measuring the total tails mass can be a tedious and long process, but it is technically feasible and typically, depleted uranium tails are stored away and not processed further [17].

Having 'untangled' the tails, one can then calculate the amount of enrichment product per tails using Eqs. (4.21) and (4.23). The absolute values of the product quantities are obtained with the results from Eq. (6.5).

Applying this methodology to both enrichment facilities yields the reconstructed HEU values shown in Table 6.2. While three of the four reconstructed values match the simulated ones, the value for the 2 MWd/kg natural uranium case deviates by around 0.2%. This inconsistency is an artefact of Cyclus and `MIsoEnrich`. If an enrichment is performed at time step t_1, then the resulting product is traded away in the same time step. However, the tails from this enrichment are offered and possibly traded only one time step later, at $t_1 + 1$. Thus, in the last simulation time step, HEU is enriched and sent to the HEU storage but the corresponding tails are not yet sent to the tails storage.[4] This gets confirmed by the fact that the enrichment facility produces about 47 kg of weapon-grade uranium in a time step—exactly the amount missing in the reconstruction.

Using the above explained method allows to reconstruct the SEU production, as well. These results are shown in Table 6.3. Here, only one reconstruction is in direct accordance to the simulation, that is the 2 MWd/kg natural uranium case. Looking at the 0.5 MWd/kg natural uranium value, the same Cyclus and `MIsoEnrich` artefact as in the HEU reconstruction is observed. However, this time the absolute deviation is larger: this is expected as significantly more SEU is produced in one time step than HEU. The difference between reconstructed and simulated value, 5290 kg, coincides with the SEU production per time step.

[4] This behaviour has an undesirable side-effect here, nonetheless it remains implemented as is for a reason: the enrichment of *product* is done upon requests from other facilities while the tails are a *byproduct* of this process. Phrased differently: one would not enrich uranium in order to obtain depleted uranium. At the same time, because of how Cyclus' DRE works, the tails cannot be offered in the same time step as the product.

Table 6.3: Reconstructed and simulated SEU production for both enrichment of natural and of reprocessed uranium. The deviation in the 0.5 MWd/kg natural U case is an artefact of CYCLUS and `MIsoEnrich` and it is not a fault in the methodology. The small deviations in both reprocessed U reconstructions are probably due to imprecisions in the nuclear data used during the reconstruction.

burnup [MWd/kg]	enrichment feed	recon. SEU [kg]	actual SEU [kg]
0.5	natural U	1 725 779	1 731 069
0.5	reprocessed U	1 401 670	1 401 553
2	natural U	775 740	775 740
2	reprocessed U	405 986	405 985

The other two deviations are both observed for the enrichment of reprocessed uranium, but they are magnitudes smaller than the previous ones: around 8×10^{-3}% in the 0.5 MWd/kg simulation and 2×10^{-4}% in the 2 MWd/kg simulation. The reason for these variations is not unknown. Most probably, they are introduced when doing conversions between mass fractions and atom fractions in the reconstruction. These conversions are needed because CYCLUS and SERPENT 2 output files store compositions in mass fractions in contrast to the atom fractions used in the MARC calculator. It can be that small deviations are introduced during these conversions because of minor deviations in the nuclear data used. A possible fix to this could be to ensure that all three programs use the same nuclear data library.

Chapter 7

Conclusion and Outlook

In the course of this thesis, I have created an initial implementation of a nuclear archaeology module for Cyclus.

First, the nuclear fuel cycle and its constituents have been presented. In addition to this, the importance of enrichment in the civilian fuel cycle, but also in the production of fissile material, has been explained.

Then, von Halle's MARC theory on multi-isotope enrichment has been analysed and presented. Its implementation and the implementation of the whole facility in a Cyclus module has been done successfully. The MARC model offers a stable and precise method to calculate the enrichment and it relies on closed and analytical formulae. Thus, the enrichment parameters can be quickly calculated and no complicated numerical tools are needed. This makes it highly suitable for the use in NFC simulators where simulations can cover large periods of time correspondingly resulting in long simulation runtimes. Additionally, MARC is *not* restricted to separation processes with small separation factors; it works equally well for both small and large separation factors.

An inconvenience has been the lack of publicly accessible experimental data or validation data due to centrifugation being a highly classified technology. The few available research articles have been used to counter-check the functionality of the MARC algorithm and these checks have been successful.

Last, the case study has demonstrated the use of Cyclus and MIsoEnrichment in a nuclear archaeology context.

The next steps in the development of MIsoEnrichment could be to investigate and implement new methods for calculating the SWU in a multi-isotope enrichment process. Another idea is to implement a nuclear reactor facility in MIsoEnrichment using the Gaussian process regression method. Once configured, this method allows emulating the reactor operation for a multitude of parameters without resorting to computationally expensive simulations.

Finally, future work could be the simulation and reconstruction of a real-life case such as the North-Korean or the Pakistani nuclear weapons programme.

Appendix A

Spent Fuel Composition

In this chapter, the spent fuel composition used in the case study, Chapter 6, is indicated. It was calculated with SERPENT 2 following the methodology from [12, 13, 36]. The nuclear reactor is an American Savannah River Site heavy water reactor, see for example [47].

The spent fuel composition is indicated in Table A.1. Note that this table only lists the isotopes relevant to this work. The actual composition used in the simulation contains a plethora of other isotopes. As expected, the low burnup case produces less plutonium, but it has a higher concentration of Pu-239, more than 97%. The second case with a burnup of 2 MWd/kg produces four times more plutonium, but with a Pu-240 content of 7.0%. Thus, the plutonium is on the limit of not being weapon-grade[1] [39].

Table A.1: Spent fuel composition for both simulated burnup cases and for both fresh and reprocessed fuel. Note that here, only the isotopes relevant to this work are listed. All values are in %mass. [12]

Burnup	0.5 MWd/kg		2 MWd/kg	
Fuel	Natural U	Reprocessed U	Natural U	Reprocessed U
U-234	0.0085	0.0089	0.0081	0.0095
U-235	1.0299	1.0299	0.8854	0.8855
U-236	0.0098	0.0199	0.0347	0.0747
U-238	98.8509	98.8404	98.6897	98.6475
Np-239	0.0078	0.0078	0.0077	0.0077
Pu-239	0.0405	0.0405	0.1531	0.1530
Pu-240	0.0009	0.0009	0.0117	0.0117
Pu-241	0.0001	0.0001	0.0024	0.0024
uranium	99.8991	99.8990	99.6179	99.6172
plutonium	0.0415	0.0415	0.1672	0.1671
waste	0.0517	0.0517	0.2072	0.2080

[1] Weapon-grade plutonium is defined as plutonium having a Pu-240 content of 7% or lower. Because of Pu-240's comparably high spontaneous fission rate, the risk of a predetonation of the nuclear weapon increases.

Appendix B

Verification of the MARC Algorithm

This chapter shows the results of comparing enrichment product compositions, calculated with the MARC algorithm, to values from literature [62]. More details are found in Section 4.7.

Table B.1: Comparison of the calculated product concentrations to literature values for the enrichment of natural uranium to LEU and HEU using gas centrifuges [62].

product	isotope	MARC algorithm [%]	Wood 2008 [%]	rel. diff. [%]
LEU	U-234	4.55×10^{-2}	4.57×10^{-2}	0.57
LEU	U-235	5	5	0
HEU	U-234	0.881	0.890	0.96
HEU	U-235	93	93	0

Table B.2: Comparison of the calculated product concentrations to literature values for the enrichment of natural uranium to LEU and HEU using gaseous diffusion [62].

product	isotope	MARC algorithm [%]	Wood 2008 [%]	rel. diff. [%]
LEU	U-234	4.78×10^{-2}	4.78×10^{-2}	0.03
LEU	U-235	5	5	0
HEU	U-234	0.934	0.935	0.04
HEU	U-235	93	93	0

Table B.3: Comparison of the calculated product concentrations to literature values for the enrichment of reprocessed uranium to LEU and HEU using gaseous diffusion [62].

product	isotope	MARC algorithm [%]	Wood 2008 [%]	rel. diff. [%]
LEU	U-232	7.55×10^{-9}	7.56×10^{-9}	0.06
LEU	U-234	0.133	0.133	0.03
LEU	U-235	5	5	0
LEU	U-236	1.57	1.57	-0.03
HEU	U-232	1.20×10^{-7}	1.20×10^{-7}	0.01
HEU	U-234	2.10	2.10	0.01
HEU	U-235	75	75	0
HEU	U-236	16.51	16.35	-0.97

Appendix C

Reproducible Research

While this thesis aims to give both comprehensive and comprehensible explanations of the methods used in this work, the reader may want additional insights. To this end and to ensure the reproducible results, the code used during the development of the thesis is publicly available at `https://github.com/maxschalz/studious_potato/tree/main`.

In the following, a brief explanation of the structure of the code repository is given. First, please note that CYCLUS, CYCAMORE and MISOENRICHMENT are *not* included in the repository. The source codes thereof are found in [56, 57] and the MISOENRICHMENT module is found in `https://github.com/maxschalz/miso_enrichment`. The versions and the dependencies' versions used are indicated in the upper half of Table C.1.

Apart from that, the repository contains all the code used in this thesis. The `data` folder contains the digitised data of [51] used in the verification of the MARC algorithm (Section 4.7), the SERPENT 2 output files (i.e., the reactor simulation) and the CYCLUS simulations used in the case study (Chapter 6). Note that the simulation output files are not published because of their large file sizes. However, they can be recreated using the available input files.

The `analysis` folder contains all the Python scripts needed to reproduce the plots shown in Chapter 4, the MARC algorithm checks and the whole analysis of the case study. The Python and SciPy version used is indicated in the lower half of Table C.1.

Table C.1: The required software libraries and the versions used. The upper half are all CYCLUS related packages while the lower half is Python related and only needed in the analysis.

CYCLUS	1.5.5	MISOENRICHMENT	1.0
CYCAMORE	1.5.5	Boost	1.71
Coin-Cbc	2.8	Coin-Clp	1.15
Hdf5	1.10.5-	Sqlite3	3.30.1
xml2	2.9.9	xml++	2.40.1
Python	3.6.5	SciPy	1.1.0
Pandas	0.23.0	NumPy	1.14.3
Matplotlib	2.2.2		

Zentrales Prüfungsamt/Central Examination Office

RWTH AACHEN UNIVERSITY

Eidesstattliche Versicherung
Statutory Declaration in Lieu of an Oath

Schalz, Max

Name, Vorname/Last Name, First Name

Matrikelnummer (freiwillige Angabe)
Matriculation No. (optional)

Ich versichere hiermit an Eides Statt, dass ich die vorliegende ~~Arbeit/Bachelorarbeit/~~ Masterarbeit* mit dem Titel
I hereby declare in lieu of an oath that I have completed the present paper/Bachelor thesis/Master thesis* entitled

Fuel cycle simulations to reconstruct uranium enrichment programmes: A first implementation

selbstständig und ohne unzulässige fremde Hilfe (insbes. akademisches Ghostwriting) erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Für den Fall, dass die Arbeit zusätzlich auf einem Datenträger eingereicht wird, erkläre ich, dass die schriftliche und die elektronische Form vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.
independently and without illegitimate assistance from third parties (such as academic ghostwriters). I have used no other than the specified sources and aids. In case that the thesis is additionally submitted in an electronic format, I declare that the written and electronic versions are fully identical. The thesis has not been submitted to any examination body in this, or similar, form.

Ort, Datum/City, Date

Unterschrift/Signature

*Nichtzutreffendes bitte streichen
*Please delete as appropriate

Belehrung:
Official Notification:

§ 156 StGB: Falsche Versicherung an Eides Statt
Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.
Para. 156 StGB (German Criminal Code): False Statutory Declarations
Whoever before a public authority competent to administer statutory declarations falsely makes such a declaration or falsely testifies while referring to such a declaration shall be liable to imprisonment not exceeding three years or a fine.
§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt
(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.
(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.
Para. 161 StGB (German Criminal Code): False Statutory Declarations Due to Negligence
(1) If a person commits one of the offences listed in sections 154 through 156 negligently the penalty shall be imprisonment not exceeding one year or a fine.
(2) The offender shall be exempt from liability if he or she corrects their false testimony in time. The provisions of section 158 (2) and (3) shall apply accordingly.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:
I have read and understood the above official notification:

Ort, Datum/City, Date

Unterschrift/Signature

Bibliography

[1] David Albright, Frans Berkhout and William Walker. *Plutonium and Highly Enriched Uranium 1996*. Oxford University Press, 1997. ISBN: 0-19-828009-2.

[2] D. G. Avery and E. Davies. *Uranium Enrichment by Gas Centrifuge*. London, United Kingdom: Mills & Boon Ltd., 1973. ISBN: 0-263-05121-8.

[3] Shannon Bogus. 'U.S. Completes INF Treaty Withdrawal'. In: *Arms Control Today* 49 (Sept. 2019).

[4] B. Brigoli. 'Cascade Theory'. In: *Uranium Enrichment*. Ed. by S. Villani. Vol. 35. Topics in Applied Physics. Berlin, Heidelberg and New York: Springer-Verlag, 1979, pp. 13–54. ISBN: 3-540-09385-0.

[5] Bulletin of the Atomic Scientists. *It is 100 seconds to midnight. 2020 Doomsday Clock Statement*. Ed. by John Mecklin. 23rd Jan. 2020. URL: https://thebulletin.org/doomsday-clock/current-time/ (visited on 24/09/2020).

[6] Richard H. Byrd et al. 'A Limited Memory Algorithm for Bound Constrained Optimization'. In: *SIAM Journal on Scientific Computing* 16.5 (1995), pp. 1190–1208. DOI: 10.1137/0916069.

[7] United States Nuclear Regulatory Commission. *Uranium Enrichment*. 15th Apr. 2019. URL: https://www.nrc.gov/materials/fuel-cycle-fac/ur-enrichment.html (visited on 27/08/2020).

[8] Kelsey Davenport. 'Iran Newly Breaches Nuclear Deal'. In: *Arms Control Today* 49 (Dec. 2019).

[9] A. de la Garza. 'A Generalization of the Matched Abundance-Ratio Cascade for Multicomponent Isotope Separation'. In: *Chemical Engineering Science* 18 (1963), pp. 73–82. DOI: 10.1016/0009-2509(63)80016-0.

[10] A. de la Garza, G. A. Garrett and J. E. Murphy. 'Multicomponent Isotope Separation in Cascades'. In: *Chemical Engineering Science* 15.3,4 (1961), pp. 188–209.

[11] Steve Fetter. 'Nuclear Archaeology: Verifying Declarations of Fissile-Material Production'. In: *Science & Global Security* 3 (1993), pp. 237–259. DOI: 10.1080/08929889308426386.

[12] Antonio Figueroa. Personal communication. 25th Sept. 2020.

[13] Antonio Figueroa and Malte Göttsche. 'Nuclear archaeology: reconstructing reactor histories from reprocessing waste'. In: *ESARDA Bulletin* 59 (Dec. 2019), pp. 39–46. ISSN: 1977-5296.

[14] Alex Gasner and Alexander Glaser. 'Nuclear Archaeology for Heavy-Water-Moderated Plutonium Production Reactors'. In: *Science & Global Security* 19.3 (2011), pp. 223–233. ISSN: 0892-9882. DOI: 10.1080/08929882.2011.616124.

[15] Matthew J. Gidden. 'An Agent-Based Modeling Framework and Application for the Generic Nuclear Fuel Cycle'. Ph.D. dissertation. Madison, WI, United States: University of Wisconsin, 2013.

[16] Alexander Glaser. 'Characteristics of the Gas Centrifuge for Uranium Enrichment and Their Relevance for Nuclear Weapon Proliferation'. In: *Science & Global Security* 16 (2008), pp. 1–25. ISSN: 0892-9882. DOI: 10.1080/08929880802335998.

[17] Alexander Glaser. 'Facilitating Nuclear Disarmament. Verified Declarations of Fissile Material Stocks and Production'. In: *The Nonproliferation Review* 19.1 (2012), pp. 125–135. DOI: 10.1080/10736700.2012.655092.

[18] Alexander Glaser and Malte Göttsche. 'Fissile Material Stockpile Declarations and Cooperative Nuclear Archaeology'. In: *Verifiable Declarations of Fissile Material Stocks: Challenges and Solutions*. FM(C)T Meeting Series. 2017, pp. 25–38. URL: https://www.unidir.org/publication/fmct-meeting-series-verifiable-declarations-fissile-material-stocks-challenges-and (visited on 24/09/2020).

[19] Russell Goldman. *India-China Border Dispute: A Conflict Explained*. 17th June 2020. URL: https://www.nytimes.com/2020/06/17/world/asia/india-china-border-clashes.html (visited on 24/09/2020).

[20] Google Inc. *Googletest – Google Testing and Mocking Framework*. 2008. URL: https://github.com/google/googletest (visited on 22/09/2020).

[21] Malte Göttsche and Baptiste Mouginot. 'An Integrated Nuclear Archaeology Approach to Reconstructing Fissile Material Production Histories'. In: *ESARDA 39th Annual Meeting*. ESARDA. Publications Office of the European Union, 2017. ISBN: 978-92-79-73861-6. DOI: 10.2760/631364.

[22] E. von Halle. 'Multicomponent Isotope Separation in Matched Abundance Ratio Cascades Composed of Stages with Large Separation Factors'. In: *Proceedings of the 1st Workshop on Separation Phenomena in Liquids and Gases*. Ed. by K. G. Roesner. Darmstadt, Germany, July 1987.

[23] Richard D. Harvey. 'An Optimization Method for Matched Abundance-Ratio Cascades by Varying the Key Weight'. Ph.D. dissertation. Knoxville, TN, United States: University of Tennessee, 2017. URL: https://trace.tennessee.edu/utk_graddiss/4403/ (visited on 17/09/2020).

[24] K. W. Hesketh. 'Power reactors'. In: *The Nuclear Fuel Cycle*. Ed. by Peter D. Wilson. Oxford, New York and Tokyo: Oxford University Press, 1996, pp. 78–101. ISBN: 0-19-856540-2.

[25] Kathryn D. Huff et al. 'Fundamental concepts in the Cyclus nuclear fuel cycle simulation framework'. In: *Advances in Engineering Software* 94 (Apr. 2016), pp. 46–59. ISSN: 0965-9978. DOI: 10.1016/j.advengsoft.2016.01.014.

[26] International Atomic Energy Agency. *IAEA safeguards glossary. 2001 Edition*. International nuclear verification series 3. Vienna, Austria, 2002. ISBN: 92–0–111902–X.

[27] International Atomic Energy Agency. *IAEA Safety Glossary. 2018 Edition*. Non-serial Publications. Vienna, Austria, 2019. ISBN: 978-92-0-104718-2.

[28] International Atomic Energy Agency. *Nuclear Fuel Cycle Information System*. 2012. URL: https://infcis.iaea.org/NFCIS/Facilities/ (visited on 27/08/2020).

[29] International Campaign to Abolish Nuclear Weapons. *Historic milestone: UN Treaty on the Prohibition of Nuclear Weapons reaches 50 ratifications needed for entry into force*. URL: https://www.icanw.org/historic_milestone_un_treaty_on_the_prohibition_of_nuclear_weapons_reaches_50_ratifications_needed_for_entry_into_force (visited on 03/11/2020).

[30] International Campaign to Abolish Nuclear Weapons. *The Treaty*. URL: https://www.icanw.org/the_treaty (visited on 24/10/2020).

[31] International Panel on Fissile Materials. *Global Fissile Material Report 2015*. Annual report of the International Panel on Fissile Materials. Dec. 2015. URL: http://fissilematerials.org/library/gfmr15.pdf (visited on 06/10/2020).

[32] Jacob J. Jacobson et al. 'Verifiable Fuel Cycle Simulation Model (VISION): A Tool for Analyzing Nuclear Fuel Cycle Futures'. In: *Nuclear Technology* 172.2 (2010), pp. 157–178. DOI: 10.13182/NT172-157.

[33] Christopher A. Juchau, Mary Lou Dunzik-Gougar and Jacob J. Jacobson. 'Modeling the Nuclear Fuel Cycle'. In: *Nuclear Technology* 171.2 (2010), pp. 136–141. DOI: 10.13182/NT171-136.

[34] Shannon N. Kile and Hans M. Kristensen. 'World nuclear forces. Overview'. In: *SIPRI Yearbook 2018*. Oxford: Oxford University Press, 2018, pp. 235–236.

[35] Allan S. Krass et al. *Uranium Enrichment and Nuclear Weapon Proliferation*. London and New York: Taylor & Francis Ltd., 1983.

[36] Jaakko Leppänen et al. 'The Serpent Monte Carlo code: Status, development and applications in 2013'. In: *Annals of Nuclear Energy* 82 (2015), pp. 142–150. ISSN: 0306-4549. DOI: https://doi.org/10.1016/j.anucene.2014.08.024.

[37] J. Magill, R. Dreher and Zs. Sóti. *Karlsruher Nuklidkarte. Chart of the Nuclides*. Wall Chart. Version 10. Nucleonica GmbH, 2018.

[38] Julia Masterson. 'North Korea Sets Conditions for Diplomacy'. In: *Arms Control Today* 50 (Sept. 2020).

[39] Klaus Mayer, Maria Wallenius and Zsolt Varga. 'Sample characteristics and nuclear forensic signatures'. In: *The New Nuclear Forensics*. Ed. by Vitaly Fedchenko. SIPRI Monographs. Oxford University Press, 2015. ISBN: 978–0–19–873664–6.

[40] Klaus Mayer et al. 'Inorganic mass spectrometry as a tool of destructive nuclear forensic analysis'. In: *The New Nuclear Forensics*. Ed. by Vitaly Fedchenko. SIPRI Monographs. Oxford University Press, 2015. ISBN: 978–0–19–873664–6.

[41] Patrick J. Migliorini, William C. Witt and Houston G. Wood. 'Semi-Empirical Method for Developing a Centrifuge Performance Map'. In: *Separation Science and Technology* 48.15 (2013), pp. 2225–2233. DOI: 10.1080/01496395.2013.805229.

[42] Baptiste Mouginot. *cycvt*. Source Code. URL: https://github.com/bam241/cycvt (visited on 22/09/2020).

[43] Nuclear Energy Agency and International Atomic Energy Agency. *Uranium 2018*. 2019. ISBN: 9789264311541. DOI: 10.1787/20725310.

[44] George D. Oliver, H. T. Milton and J. W. Grisard. 'The Vapor Pressure and Critical Constants of Uranium Hexafluoride'. In: *Journal of the American Chemical Society* 75.12 (1953), pp. 2827–2829. DOI: 10.1021/ja01108a011.

[45] J. P. Patterson and P. Parkes. 'Recycling uranium and plutonium'. In: *The Nuclear Fuel Cycle*. Ed. by Peter D. Wilson. Oxford, New York and Tokyo: Oxford University Press, 1996, pp. 138–160. ISBN: 0-19-856540-2.

[46] Princeton Nuclear Futures Lab. *NU – Mapping Nuclear Verification*. URL: http://verification.nu (visited on 06/10/2020).

[47] Mary Beth Reed et al. *Savannah River Site at Fifty*. Ed. by Barbara Smith Strack. U.S. Government Printing Office, 2002. ISBN: 0-16-067182-5.

[48] Kingston Reif and Shannon Bugos. 'No Progress Toward Extending New START'. In: *Arms Control Today* 50 (July–Aug. 2020).

[49] S. Richter et al. 'Isotopic "fingerprints" for natural uranium ore samples'. In: *International Journal of Mass Spectrometry* 193.1 (1999), pp. 9–14. ISSN: 1387-3806. DOI: 10.1016/S1387-3806(99)00102-5.

[50] SciPy. *scipy.optimize.minimize*. 17th Oct. 2020. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html (visited on 30/10/2020).

[51] Matthew Sharp. 'Applications and Limitations of Nuclear Archaeology in Uranium Enrichment Plants'. In: *Science & Global Security* 21 (2013), pp. 70–92. ISSN: 0892-9882. DOI: 10.1080/08929882.2013.755028.

[52] Soubbaramayer. 'Centrifugation'. In: *Uranium Enrichment*. Ed. by S. Villani. Vol. 35. Topics in Applied Physics. Berlin, Heidelberg and New York: Springer-Verlag, 1979, pp. 183–244. ISBN: 3-540-09385-0.

[53] Carey Sublette. *Nuclear Weapons Frequently Asked Questions. Nuclear Materials*. 20th Feb. 1999. URL: http://nuclearweaponarchive.org/Nwfaq/Nfaq6.html (visited on 26/10/2020).

[54] PyNE Development Team. *PyNE. The Nuclear Engineering Toolkit*. 2019. URL: http://pyne.io (visited on 17/09/2020).

[55] *Treaty on the Non-Proliferation of Nuclear Weapons*. 140. Information Circular. International Atomic Energy Agency, 22nd Apr. 1970. URL: https://www.iaea.org/sites/default/files/publications/documents/infcircs/1970/infcirc140.pdf (visited on 22/09/2020).

[56] University of Wisconsin Computational Nuclear Engineering Research Group. *Cycamore : The CYClus Additional MOdules REpository*. Source Code. 2016. URL: https://github.com/cyclus/cycamore (visited on 22/09/2020).

[57] University of Wisconsin Computational Nuclear Engineering Research Group. *Cyclus*. Source Code. 2016. URL: https://github.com/cyclus/cyclus (visited on 22/09/2020).

[58] University of Wisconsin Computational Nuclear Engineering Research Group. *Cyclus*. 29th Mar. 2020. URL: http://fuelcycle.org (visited on 24/08/2020).

[59] University of Wisconsin Computational Nuclear Engineering Research Group. *Mbmore*. Source Code. 2016. URL: https://github.com/CNERG/mbmore (visited on 22/09/2020).

[60] Pauli Virtanen et al. 'SciPy 1.0: fundamental algorithms for scientific computing in Python'. In: *Nature Methods* 17.3 (3rd Feb. 2020), pp. 261–272. ISSN: 1548-7105. DOI: 10.1038/s41592-019-0686-2.

[61] Peter D. Wilson, ed. *The Nuclear Fuel Cycle*. Oxford, New York and Tokyo: Oxford University Press, 1996. ISBN: 0-19-856540-2.

[62] Houston G. Wood. 'Effects of Separation Processes on Minor Uranium Isotopes in Enrichment Cascades'. In: *Science & Global Security* 16.1–2 (2008), pp. 26–36. ISSN: 0892-9882. DOI: 10.1080/08929880802361796.

[63] Houston G. Wood, Valentin D. Borisevich and Georgiy A. Sulaberidze. 'On a Criterion Efficiency for Multi-Isotope Mixtures Separation'. In: *Separation Science and Technology* 34.3 (1999), pp. 343–357. DOI: 10.1081/SS-100100654.

[64] Moeed W. Yusuf. 'The Pulwama Crisis: Flirting With War in a Nuclear Environment'. In: *Arms Control Today* 49 (May 2019).

[65] Ciyou Zhu et al. 'Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-Scale Bound-Constrained Optimization'. In: *ACM Transactions on Mathematical Software* 23.4 (Dec. 1997), pp. 550–560. ISSN: 0098-3500. DOI: 10.1145/279232.279236.