

---

# Chemical-Thermodynamic Modelling of the MSR-Related Systems Under Normal and Accident Conditions

---

Submitted in Partial Fulfilment  
of the Requirements for the Degree of  
Master of Science  
by

Jonathan Dietz  
Department of Mechanical and Process Engineering,  
Swiss Federal Institute of Technology,  
CH-8092 Zürich

## Supervision

Dr. Sergii Nichenko  
Multiscale Materials Modelling Group,  
Paul Scherrer Institute,  
CH-5232 Villigen PSI

Dr. Jiří Křepel  
Advanced Nuclear Systems Group,  
Paul Scherrer Institute,  
CH-5232 Villigen PSI

Prof. Horst-Michael Prasser  
Department of Mechanical and Process Engineering,  
Swiss Federal Institute of Technology,  
CH-8092 Zürich

Zürich, September 30, 2020

# 1 Abstract

In order to be able to have a solid understanding of the species, phases and chemical interactions present in the materials of a molten salt reactor, the gap between the neutronic, isotope-based fuel evolution codes and the chemical-thermodynamic codes has to be bridged. To achieve this, the EQL0D code, which uses Serpent and MATLAB to simulate the evolution of materials in an MSR, was combined with the Gibbs energy minimization software GEMS in the scope of this thesis.

As a first step, basic cases were created and run for both GEMS as well as EQL0D, in order to validate the underlying chemical-thermodynamic database HERACLES and for familiarization purposes with EQL0D. For GEMS, these validation cases take the form of binary phase diagrams, wherein both the substance-specific, thermodynamic parameters as well as binary liquid phase interaction parameters were adjusted to match literature diagrams. Where necessary, new species were added to the HERACLES database. In EQL0D, many old cases were rerun, adjusted and analysed in order to facilitate the creation of four simplified, homogenized input structures which are to serve as the baseline for speciation work performed in GEMS.

Once the composition of the equilibrium salt mixture is obtained, it is put into GEMS, which will output the according speciation. In this work, the LiF-UF<sub>4</sub> system was selected. For the sake of simplicity as well as reducing the amount of required database work of the simulations, only the ten most prevalent elements are included. Then, some analysis in terms of the influence of redox conditions as well as temperature on the outcome of the speciation is performed.

In addition to the above chain of simulations, some tasks not falling within the chain of analyses outline above were completed. Using the updated thermodynamic database, cGEMS, a tool combining the release analysis of MELCOR with the powerful thermodynamics of GEMS was run in order to simulate the release from a given salt mixture. The results are compared with literature results obtained from the same input prior to the modifications made to the database. EQL0D is used extensively for calculations as part of the SAMOSAFAFER benchmarking process. Multiple codes from different institutes are compared against each other using incremental benchmarking, with the goal being to start simple and ramp up the complexity of the system to the realistic case. Results from various different cases are presented in this work.

On top of the as of yet quite basic benchmarking runs, a full-scale MSFR case is run, analysed and presented.

## 2 Declaration of Originality



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

### Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

**Title of work** (in block letters):

Chemical-Thermodynamic Modelling of the MSR-Related Systems Under Normal and Accident Conditions

**Authored by** (in block letters):

*For papers written by groups the names of all authors are required.*

**Name(s):**

Dietz

**First name(s):**

Jonathan

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the ['Citation etiquette'](#) information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

**Place, date**

Volketswil, 30.09.2020

**Signature(s)**

*For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.*

### 3 Acknowledgements

First of all, I would like to thank Prof. Prasser for setting up the opportunity for me to combine my interests in chemistry and nuclear engineering, as well as his willingness to supervise this thesis.

I would like to thank my co-supervisors, Dr. Nichenko and Dr. Křepel, for their support, explanations, as well as patience with my frequent questions.

On top of the supervisors, Rodrigo was a great help when it comes to technical support, especially with respect to Serpent installations and nuclear databases.

Additionally, many thanks to Dario for always being available for me to bounce ideas and issues off of, as well as providing occasionally much needed distractions.

This thesis was also partly made possible by Roland and Priska, who gave me a room to use as my home office, as well as (crucially) access to their coffee machine during the final stretch.

Finally, thanks are definitely due to my girlfriend Sabrina, for her unwavering support, understanding and love.



# Contents

<b>1</b>	<b>Abstract</b>	<b>i</b>
<b>2</b>	<b>Declaration of Originality</b>	<b>ii</b>
<b>3</b>	<b>Acknowledgements</b>	<b>iii</b>
	<b>Contents</b>	<b>iv</b>
	<b>List of Figures</b>	<b>vi</b>
	<b>List of Tables</b>	<b>viii</b>
<b>4</b>	<b>Introduction</b>	<b>1</b>
4.1	Generation IV Reactors . . . . .	1
4.2	The Molten Salt Reactor . . . . .	2
4.2.1	The Molten Salt Fast Reactor . . . . .	2
4.3	Alternative Gen IV Designs . . . . .	2
4.4	Goals and Approach . . . . .	4
<b>5</b>	<b>Methods and Tools</b>	<b>5</b>
5.1	MATLAB . . . . .	5
5.2	Serpent2 . . . . .	5
5.3	GEMS . . . . .	5
5.4	EQL0D . . . . .	6
5.4.1	Reactivity Control . . . . .	7
5.4.2	Redox Control . . . . .	8
5.5	MELCOR . . . . .	8
5.6	cGEMS . . . . .	8
<b>6</b>	<b>Chemical-Thermodynamic Modelling of MSR Salts</b>	<b>9</b>
6.1	Collecting Data and Organizing the Database . . . . .	9
6.2	Methodology Example . . . . .	10
6.3	Specific Cases . . . . .	12
6.4	Chemical-Thermodynamics of Two- and Three-Component Systems . . . . .	12
6.4.1	Methodology . . . . .	12
6.4.2	Chlorides . . . . .	20
6.4.3	Fluorides . . . . .	24
6.4.4	Ternary Diagram . . . . .	27
6.5	Speciation and Chemical-Thermodynamic Analysis of Equilibrium Composition	29
6.5.1	Base Salt . . . . .	30
6.5.2	Results . . . . .	31
6.6	Release Analysis in cGEMS . . . . .	34
6.6.1	Results . . . . .	35

<b>7</b>	<b>Fuel Cycle Modelling for MSR Systems</b>	<b>37</b>
7.1	Learning and Adjusting EQL0D . . . . .	37
7.2	SAMOSAFER Simulation Benchmark . . . . .	37
7.2.1	General Purpose . . . . .	37
7.2.2	Procedure . . . . .	38
7.2.3	Incremental Benchmarking . . . . .	38
7.2.4	Implementation . . . . .	39
7.2.5	Results . . . . .	41
7.3	Generating Representative Salt Compositions . . . . .	46
7.3.1	Previous Cases . . . . .	46
7.3.2	Adjustments for the Homogenized Cases . . . . .	46
7.3.3	Case Definitions . . . . .	46
7.3.4	Results . . . . .	47
7.4	Sensitivity Analysis . . . . .	48
7.4.1	Results . . . . .	49
7.5	Full-Scale MSFR Simulation . . . . .	53
7.5.1	Simulation Setup . . . . .	55
7.5.2	Results . . . . .	55
<b>8</b>	<b>Summary and Outlook</b>	<b>60</b>
8.1	Data Collection . . . . .	60
8.2	Phase Diagrams . . . . .	61
8.3	Speciation of an Equilibrium Composition . . . . .	62
8.4	Learning EQL0D and Early Cases . . . . .	62
8.5	SAMOSAFER Benchmark . . . . .	63
8.6	MSFR-Like EQL0D Cases and Sensitivity Analysis . . . . .	63
8.7	Complex Case . . . . .	64
8.8	Final Thoughts . . . . .	65

## List of Figures

1	RTparm optimization example . . . . .	11
2	The initial outcome of running the simulation for literature interaction parameters.	16
3	The outcome of the second iteration on the NaCl-ThCl <sub>4</sub> phase diagram. . . . .	17
4	Plot of the phase diagram of the NaCl-ThCl <sub>4</sub> system after the second adjustment.	18
5	The last phase diagram of NaCl-ThCl <sub>4</sub> before the final result. . . . .	19
6	Phase diagram of NaCl-ThCl <sub>4</sub> . . . . .	21
7	Phase diagram of NaCl-PuCl <sub>3</sub> . . . . .	22
8	Phase diagram of NaCl-UCl <sub>3</sub> . . . . .	23
9	Phase diagram of ThCl <sub>4</sub> -PuCl <sub>3</sub> . . . . .	24
10	Phase diagram of LiF-PuF <sub>3</sub> . . . . .	25
11	Phase diagram of ThF <sub>4</sub> -LiF . . . . .	27
12	Ternary phase diagram of the CsF-LiF-ThF <sub>4</sub> system for 500 °C. . . . .	28
13	Evolution of the base salt with changing fluorine content . . . . .	32
14	Evolution of the base salt in logarithmic scale . . . . .	32
15	Evolution of the base salt with temperature change . . . . .	33
16	Cumulative release of various substances as a function of time. . . . .	35
17	Fission product share evolution during benchmark . . . . .	41
18	Elemental distribution of final fuel composition . . . . .	42
19	Plot of the neutron capture rate change during a run of the benchmarking process.	44
20	Composition of various fuel-burning benchmark results as a function of atomic mass. . . . .	45
21	Atomic density plot for the four MSFR-likes . . . . .	50
22	Fission product composition for volatile removal variation . . . . .	51
23	Fission product composition for metallic removal variation . . . . .	52
24	Plot for combined metallic and volatile variation . . . . .	53
25	SAMOFAR reprocessing scheme . . . . .	54
26	Fission product evolution for the complex case . . . . .	55
27	Fission product evolution for the complex case in mass units . . . . .	55
28	Evolution of various elements with operating time . . . . .	57
29	Evolution of additional elements with operating time . . . . .	57
30	Plot of decay heat evolution over a three year timespan . . . . .	58
31	Plot of radiotoxicity evolution over a three year timespan . . . . .	58
32	GEMS project settings for the initial NaCl-ThCl <sub>4</sub> phase diagram . . . . .	77
33	GEMS project settings for the final NaCl-ThCl <sub>4</sub> phase diagram . . . . .	77
34	GEMS project settings used to construct the NaCl-PuCl <sub>3</sub> , NaCl-UCl <sub>3</sub> and ThCl <sub>4</sub> - PuCl <sub>3</sub> phase diagrams. . . . .	78
35	GEMS project settings used to construct the LiF-PuF <sub>3</sub> phase diagram. . . . .	78
36	GEMS project settings used to construct the LiF-ThF <sub>4</sub> phase diagram. . . . .	79
37	GEMS project settings used to construct the LiF-PuF <sub>3</sub> phase diagram. . . . .	79
38	GEMS project settings used to construct the CsF-LiF-ThF <sub>4</sub> ternary phase diagram.	80
39	Liquid phase composition for the nominal run of the representative salt for the fluoride/Uranium system. . . . .	80
40	GEMS project settings used for the plots depicting the influence of fluorine content and temperature on the base salt. . . . .	81

41	Serpent geometry definition for a fluoride-based Thorium-232 example of the four homogenized cases. . . . .	81
42	Plot of chloride-based Thorium cases with different volatile removal rates. . . . .	82
43	Plot of chloride-based Thorium cases with different metallic removal rates. . . . .	83
44	Plot of chloride-based Uranium cases with different volatile removal rates. . . . .	83
45	Plot of chloride-based Uranium cases with different metallic removal rates. . . . .	84
46	Plot of fluoride-based Uranium cases with different volatile removal rates. . . . .	84
47	Plot of fluoride-based Uranium cases with different metallic removal rates. . . . .	85
48	Example of a failed sensitivity study for the fluoride-based Uranium system. . . . .	85

## List of Tables

1	Table of the species added to the database . . . . .	12
2	Summary of binary interaction parameters in use for GEMS cases. . . . .	15
3	Redlich-Kister coefficient evolution during the adaptation of the NaCl-ThCl <sub>4</sub> diagram. . . . .	20
4	Changes to the LiF-ThF <sub>4</sub> mixed species . . . . .	26
5	This table shows the results of a standard run using the fluoride/Uranium composition. . . . .	31
6	Distribution of Caesium in the fluoride/Uranium base salt. . . . .	34
7	Representative fuel composition for release analysis. . . . .	35
8	Boundary conditions of SAMOSAFER benchmark cases . . . . .	40
9	Differences between the MSFR-like cases . . . . .	47
10	Base salt composition for all four cases . . . . .	48
11	Summary of the MATLAB input variables contained in the OPT structure . . .	70
12	Additional OPT variables contained in substruct REA. . . . .	71
13	Additional OPT variables contained in substructs CONV and REDOX. . . . .	72
14	Summary of files that EQL0D is comprised of . . . . .	72
15	Extension of table 14. . . . .	73
16	Variables used to construct a material object. . . . .	74
17	Variables used to construct a reprocessing object. . . . .	75
18	Reprocessing setup for the benchmark . . . . .	81
19	Reprocessing setup for the MSFR-likes . . . . .	82

## 4 Introduction

### 4.1 Generation IV Reactors

Often, the evolution of the nuclear reactor technology is divided into different developmental stages called “generations”. Here, generation I represents the earliest reactors as well as some of the prototypes of currently operating reactors. Most currently operating reactors fall into generation II, which includes predominantly light water reactors (LWR). Generations III as well as III+ are and were developed as an improvement upon the above-mentioned traditional LWRs, especially in terms of efficiency and safety. In 2001, the Generation IV International Forum (GIF) was founded and decided on a set of reactor design that would be the future of nuclear power.

The choice of which reactors to include in generation IV is based on four main criteria which are heralded as the proverbial next step in making nuclear power a viable option for the world. These four criteria are designed such that GenIV reactors can address a lot of the concerns that arise today both with nuclear power, but also with power generation and its impact on society and the planet as a whole. Two of the four keywords for the next step in nuclear power evolution are quite specific to nuclear, being safety and proliferation resistance.

While safety is a global concept when it comes to industrial practice, it is an essential feature of future reactors in order to both reduce their likelihood of core damage as well as its severity. Conversely, proliferation resistance is an issue exclusive to nuclear installations. Making a nuclear power setup unattractive for the generation of weapons-grade material is essential not only as a tool for public opinion, but also from the perspective of a legality issue in terms of international relations. On top of proliferation resistance, there is another concept sometimes mentioned in the same breath, which is the safety of nuclear power installations from external influence such as terrorist attacks. Currently, this version of safety is hugely dependent on the lawmakers of each respective country, rather than being an intrinsic design feature of certain reactor types.

Outside of the nuclear-specific topics, both sustainability as well as economics are also addressed in the scope of the GIF. Economics being an important point is a natural consequence of having power production be a privatized industry. Without competitiveness with other power generation options, nuclear power will by definition fall out of favour. Sustainability, however, is a complex convolution of multiple different improvements, all of which essentially are based on a more efficient, less wasteful usage of natural resources. While nuclear power has the distinct advantage over fossil fuel-based power generation of not emitting greenhouse gases as a direct consequence of power production, it does have to make considerations in terms of exhausting natural resources.

In total, there are six designs which were selected for generation IV. Those are, in no particular order: Very High Temperature Reactor (VHTR), Sodium-cooled Fast Reactor (SFR), Supercritical Water Reactor (SCWR), Gas-cooled Fast Reactor (GFR), Lead-cooled Fast Reactor (LFR) and last but not least, the Molten Salt Reactor (MSR).<sup>[1]</sup>

## 4.2 The Molten Salt Reactor

Generally, the term Molten Salt Reactor (MSR) refers to any reactor that uses molten salts for cooling purposes. While it is often used when describing reactors using fuel in salt form, this is not necessarily the case. In this work, however, only examples using molten salts as fuel as well as coolant will be considered.

The first large excursion into molten salt reactor research was the Aircraft Reactor Experiment (ARE), which aimed to use nuclear power to propel an aircraft. It was developed at Oak Ridge National Laboratory (ORNL) and operated successfully for some time. After this initial demonstration that molten salts can potentially have a place in nuclear power, ORNL continued to research the MSR's use as a power reactor. Both single- as well as two-fluid designs were considered, however in the end the single-fluid Molten Salt Reactor Experiment (MSRE) setup was constructed and operated starting in 1965. The decision in favour of a single fluid design was motivated by the higher complexity required for a two-fluid setup. This complexity and the issues that go along with it are amplified by material issues, which naturally arise when dealing with molten salts at elevated temperatures.

The MSRE operated for 4.5 years on a moderated neutron spectrum, using Li-Be-Zr fluorides as the carrier salt, while having access to three different fissile materials:  $^{233}\text{U}$ ,  $^{235}\text{U}$  and  $^{239}\text{Pu}$ . To address the material issues mentioned above, Inconel 600, which was used during the ARE, was replaced with Hastelloy-N. While this greatly improved the compatibility with fluoride salts in terms of corrosion, some embrittlement due to  $(n, \alpha)$  reactions in the materials still posed an issue that was addressed by altering the composition of Hastelloy-N.<sup>[2]</sup>

### 4.2.1 The Molten Salt Fast Reactor

The Molten Salt Fast Reactor (MSFR) is a concept developed by the Centre National de la Recherche Scientifique (CNRS) in France. In essence, it is a fast-spectrum, fluoride-based system that relies on Thorium breeding to produce the fissile Uranium-233.<sup>[3]</sup>

From a structural point of view, the MSFR contains  $18\text{ m}^3$  of fuel, where 9 are in the core at any given time. In addition to the main fuel vessel, there is a secondary loop to transport the heat from the reactor to the power-generating system. As the intermediary material, both a molten salt as well as liquid lead have been considered.

As a fuel, the MSFR concept assumes a Lithium/Thorium Fluoride based setup, with an actinide content of 22.5mol%. Because of the assumed use of the Thorium/Uranium system, the initial core needs to be designed separately due to the lack of Uranium-233 reserves. One attractive option for startup is the use of the transuranic elements (TRU) in LWR waste, because it can make use of otherwise problematic waste due to the fast spectrum of the MSFR.

## 4.3 Alternative Gen IV Designs

Even though this work's sole application is within the realm of MSRs, a brief overview of the other five Gen IV designs will be given here.<sup>[1]</sup>

The Very High Temperature Reactor (VHTR) is a carbon-moderated, helium-cooled concept that generally follows the approach of designing a reactor with the highest possible core outlet



temperature. Doing so enables operation at a higher electrical efficiency than current commercial steam-based reactors, while making a direct helium cycle possible which eliminates the need for an intermediate heat exchanger. On top of benefits for electricity generation, there are alternative uses for the heat generated from the nuclear process. Many different possibilities are available for the cogenerated heat, which include but are not limited to: Hydrogen production via various paths including steam methane reforming or the sulfur iodine process,<sup>[4]</sup> coal liquefaction and the production of fuel from biomass. Note that while the mentioned examples all relate to the power industry in some way, there are also plenty of possibilities in other areas, especially the chemical industry.

As the VHTR is a thermal design, it can use TRISO-particles as fuel. Generally, they are combined in either pebbles or prismatic blocks, which are the parallel to the fuel element in a classical LWR. While the pyrolytic carbon and silicon carbide shell employed in producing the TRISO particles as well as the blocks and pebbles are a great addition in terms of fission product release, they do account for a large amount of  $^{14}\text{C}$  in radioactive waste.

While the above design attempts to push its temperature, the Sodium Fast Reactor (SFR) has a hard maximum of core outlet temperature in the boiling point of the sodium coolant. The SFR is a fast reactor which has received a large amount of research compared to other Gen IV reactors. There are two main designs available, the pool and the loop layout. The pool layout is designed such that the reactor vessel is essentially a pool of molten sodium which contains the necessary pumps and travel paths for the coolant. It is relatively basic in terms of construction, however the fact that everything is covered in the coolant at all times can cause some difficulties. To counteract the large size, cost and maintenance issues, the loop layout abandons the large amount of coolant and the thermal inertia it provides in favour of a more compact setup, which also reduces the costs.

Contrary to the previous two concepts, the Super-Critical Water cooled Reactor (SCWR) does not stray from the classical reactor coolant, which is water. Instead, it aims to use it in its super-critical form, which offers the advantage of not having a phase transition. This allows for a reduction in plant complexity, because crucial components such as the steam dryers and recirculation systems can be cut when compared to a Boiling Water Reactor (BWR). However, the super-criticality of the water also comes with the biggest design issue of the SCWR: At the pressure and temperature required, the super-critical coolant is very corrosive. So far, no material has been found which can reliably withstand the unique combination of corrosion and irradiation in such a system.

The Gas cooled Fast Reactor (GFR) makes use of the fast spectrum, and therefore omits the use of any moderating substances by necessity. As such, previously mentioned fuel designs such as the TRISO particle are no longer applicable. Instead, the GFR makes use of fuels specifically tailored to high temperatures, such as carbide or nitride pellets. In addition to pellets being more heat-resistant, the cladding can be made from silicon carbide for the same reason. Because of the lack of thermal inertia due to the absence of graphite or other emergency heat storage in the core as well as comparatively high power density, the GFR requires extensive security systems to make up for these destabilizing factors in case of an accident scenario.

Last on the list of Gen IV designs is the Lead cooled Fast Reactor (LFR). Similarly to the SFR, the LFR is a liquid metal cooled concept. This time, however, as the name suggests, the chosen coolant is either liquid lead or a lead-bismuth eutectic (LBE). The decision between the two is based on two main factors. On the one hand, LBE comes with a lower melting point, which is beneficial because it helps avoid freezing of the coolant. On the other hand, however, having a large amount of Bismuth in the core is problematic due to the formation of  $^{210}\text{Po}$ .

Contrary to Sodium, neither Lead nor LBE react with air or water, removing a considerable hazard from the equation. Additionally, it has a much higher boiling point, preventing the core outlet temperature from limiting the thermodynamic efficiency of the cycle. Though both of these are advantages, lead also comes with stronger corrosion, which puts an additional strain on the vessel material as well as the pumps, which have to be able to reliably move the much denser lead.

## 4.4 Goals and Approach

The main goal of this thesis is to capitalise on the combination of multiple previously existing tools. Below, the main programs and routines used in the scope of this work are laid out. While most of them are either commonly known or have previously been used in research, their combination is able to give an insight into the chemical nature of the salts in a molten salt reactor.

Often, the isotopic composition of the salts in an MSR are the prime target of simulations, as they define the neutronics and therefore the economic design of the reactor. Chemical speciation is generally not of much relevance during normal operation of a reactor; however, it can play a crucial role specifically for an MSR or during accident sequences.

For an MSR, the chemical speciation can play a crucial role in understanding what phases the elements are present in. This especially applies for fission products and members of their respective decay chain, as the initial fuel salt does not contain them. Knowing the physical state of the elements and their chemical compounds is necessary in order to design a reprocessing system, which relies heavily on differentiating between the removal of gasses, soluble substances and those which precipitate.

Additionally, during an accident sequence, the source term is crucially influenced by the volatility of problem isotopes. This volatility is in no way attributed to the isotopic composition, but instead a product of their chemical state. Depending on the element in question, the volatility can vary wildly between different chemical species.

By using the combination of EQL0D and GEMS, most aspects of a molten salt can be determined at any desired moment in an MSR's life cycle. EQL0D will provide outputs including the composition of any desired material in the system as well as associated values such as radiotoxicity, decay heat and reaction rates for certain nuclear reaction. This information is given both on an isotope- as well as element-wise basis in only one comprehensive output file. In addition, the material composition with all its added information can be printed on the completion of each cycle, if desired. To complete the analysis of MSR salts, GEMS provides speciation of the material using the elemental composition. On top of that, a lot of supporting information can be extracted if required, including solidus and liquidus temperatures as well as vapour pressures, which are especially crucial for release analyses.

The interaction between the two codes in terms of providing input parameters for each other is a crucial part of this thesis. While a lot of this kind of interaction goes from EQL0D to GEMS in the form of compositions, it is easily imaginable that GEMS would be incredibly helpful in designing a new starting fuel for some reactor design.

## 5 Methods and Tools

### 5.1 MATLAB

MATLAB is a commercial computing software developed by MathWorks.<sup>[5]</sup> It has its own programming language as well as the option to expand its capabilities by using toolboxes. While no toolboxes are required for its use in the scope of this thesis, they can be incredibly powerful tools when trying to go into more specific use cases of the program.

In this thesis, MATLAB's main application is EQL0D (see below). In this function, it is run on the Linux Cluster Merlin6 at PSI, where the GUI is forgone in favour of the ability to use the slurm queueing system. As part of EQL0D, MATLAB interfaces heavily with Serpent2, which will be introduced below.

MathWorks has released many versions over the years, with the syntax changing slightly between them. Here, the release 2019a will be employed, as it is the newest available on the cluster.

### 5.2 Serpent2

Developed at VTT Technical Research Center in Finland, Serpent2<sup>[6]</sup> is a Monte Carlo-based particle transport code made to be an alternative to the ubiquitous Monte Carlo N-Particle transport code MCNP. In the scope of this work, it is primarily used as part of the EQL0D procedure, wherein it is used for criticality analysis as well as to provide reaction rates for material evolution within the EQL0D routine.

Additionally, some standalone cases are run within the scope of the SAMOSAFER Benchmark work introduced below in order to confirm the congruency of the results with those produced by EQL0D.

### 5.3 GEMS

The Gibbs Energy Minimization Software GEMS<sup>[7][8][9]</sup> is an open-source tool developed at Paul Scherrer Institute (PSI) since 2003. While originally intended for use with geochemical data, it can be very powerful when applied to molten salts.

By supplying thermodynamic data from the HERACLES database,<sup>[10]</sup> which has been developed specifically for use with nuclear systems and therefore includes many of the necessary fission products as well as actinides, GEMS can compute the chemical speciation of a system based on input parameters such as elemental composition and temperature. This is especially relevant to the investigation of Molten Salt Reactor (MSR) fuel salts, as both their equilibrium composition as well as the releases from accident scenarios are still a matter of research and discussion.

The abovementioned HERACLES database has been established at PSI and expanded from light water reactors (LWR) to also include a lot of MSR-relevant species. However, as with a lot of other MSR research, it is principally designed to work with fluoride-based molten salt

compositions. In order to investigate the slightly less-favoured chlorides in the scope of MSR salts, HERACLES was expanded with some of the appropriate chloride species. In the present work, GEMS finds use both in establishing an initial understanding of the compositions to be investigated, as well as a transition step between the elemental output composition of EQL0D and the chemical speciation required for MELCOR.

Before speciation is possible, the Gibbs energy of each component at a specified temperature has to be computed. Because the entropy as well as formation enthalpy are only specified at standard conditions, their temperature-specific value has to be computed using the  $C_p$  curve as follows:

$$\Delta S = \int C_p/T dT$$

$$\Delta H = \int C_p dT$$

Where  $\Delta S$  and  $\Delta H$  are the changes in entropy and enthalpy respectively. Using these quantities along with their counterparts at standard conditions, the Gibbs energy of a species at a certain temperature can be obtained from

$$G = H - TS$$

When using GEMS, it is not only the thermodynamic database that plays a major role in determining the correct result, but also the mixing parameters employed for the liquids and solid solutions present in the system. This is especially relevant for MSRs as compared to LWRs, where liquid is the principal state of matter within the system. To do this, the Redlich-Kister model<sup>[11]</sup> was employed. However, it later turned out that it is rather unstable for some of the cases investigated herein. After investigating other options such as UNIQUAC and NRTL, it was settled to use Redlich-Kister after all, however with implementation of a temperature dependency.

The Redlich-Kister model uses a polynomial expansion to approximate the excess molar Gibbs energy.

$$G_m^E = x_1 x_2 \sum_k L_{12} (x_1 - x_2)^k$$

Where  $x_1$  and  $x_2$  are the molar fraction of the two components being mixed,  $L_{12}$  is one of the coefficients of the Redlich-Kister model and  $G_m^E$  is the molar excess Gibbs energy. The excess Gibbs energy denotes the difference in Gibbs energy of a solution to the ideal case and is therefore the crucial quantity when departing from ideal solutions.

The alternatives to Redlich-Kister mentioned above are both based on the local concentration around a molecule, rather than the bulk concentration. This is based on the assumption that the local concentration will differ from the bulk concentration due to molecular level interactions between the different species.

## 5.4 EQL0D

Originally written by Boris Hombourger, EQL0D is a MATLAB and Serpent-based routine which computes the evolution of materials as they are used in an MSR.<sup>[12]</sup> It uses Serpent to

establish the geometry as well as to extract reaction rates for use with the burn-up calculations. Conveniently, most Serpent output files are easily compatible with MATLAB. For the purposes of calculations in this thesis, the ENDF/B-VII.0 nuclear data library was used. While there are newer versions available, there were modifications made by the EQL0D author whose replication in a newer database are beyond the scope of this work.

The burn-up/material evolution itself is computed within MATLAB, using the CRAM-method.<sup>[13]</sup> EQL0D allows for the addition of reprocessing steps to the otherwise occurring material evolution due to phenomena such as fission or capture. The reprocessing steps utilized herein depend on the type of MSR that is being modelled; For fluoride systems, a reprocessing scheme was put forward in the scope of the SAMOFAR project and adopted for calculations. Chlorides require a slightly different treatment as there has been less work done for them.

EQL0D supports both an equilibrium search as well as a step-based approach. In equilibrium mode, a cycle length is defined and repeated until user-specified criteria for equilibrium are observed on subsequent cycles. In the step-based mode, time-steps of specified length are provided to EQL0D from the MATLAB input file.

### 5.4.1 Reactivity Control

Because the reprocessing scheme defined within an EQL0D procedure only allows for a static removal amount which is then refilled with a defined mixture, it can happen that the reactivity of a simulated system runs out of control. In order to obtain a more realistic composition that is feasible to have in an MSR, a routine is implemented which adapts the composition such that the reactivity in the core stays somewhat constant. Note that this option needs to be enabled in the MATLAB input file, should it be necessary.

The reactivity control routine has a check implemented to decide whether to activate the control or not. This is based on the reactivity difference from the target as well as a tolerance that can be set in the input file. Should the difference exceed the tolerance limit, the algorithm is started depending on whether 'addMass', 'replace' or 'addVolume' is selected as the reactivity control mode by the user. The actual adjustment of the composition in order to control reactivity is based on an iterative process. On each iteration, the difference to the target in terms of reactivity is computed and if it falls below the tolerance limit, the loop is stopped.

During each execution of the loop, a scaling factor is determined based on the previous iterations. This scaling factor is applied to the affected isotopes of the controlled material, depending on which algorithm mode is selected. While the 'addMass' and 'addVolume' are very similar from a logical point of view, they use a different aspect of the material to base the calculation around. Note that the 'addMass' option does not directly change the mass of the system; Instead, it uses the atomic mass of the affected isotopes in order to compute and apply changes to the material using the atomic composition.

The 'replace' option is tied in closely with the 'addMass' option. In fact, choosing 'replace' will still execute the computations based on atomic mass, however an additional layer is added. Instead of simply adjusting the target materials by adding or removing some amount of the selected isotopes, there is a vector of replacement isotopes which are adjusted in order to keep a constant mass in the system.

### 5.4.2 Redox Control

Redox Control is an option that can be enabled within EQL0D in order to automatically adjust the salt composition based on redox considerations. Generally, it is disabled in simulations that are performed as part of this thesis, however for completion's sake it will be briefly discussed here.

While redox control is not essential for keeping the reactor working from a neutronic perspective, there are a slew of issues that can arise in the salt if the redox conditions are off. In general, having chemical species appear in your salt that are not intended is problematic in and of itself due to potential precipitation or insolubility. However, in more extreme deviations from design conditions, it is possible to have issues with the oxidation of structural material. Most notably this can be an issue for chromium, as it will be the first element to experience oxidation.

The basis of adjusting the redox conditions is a dataset that contains the assumed valency of each element within the system. This makes it a quite simple algorithm, however it also limits the usability somewhat, because the valencies can be very different between fluoride- and chloride-based systems. In the valency dataset present in EQL0D, a valency of 4 is assumed for Uranium, which suggests it was implemented for use with fluorides.

By summing over the product of the amount of atoms of an element with their respective assumed valency, the excess of positive or negative valency can be determined. If there is an excess in either direction, which there most likely always will be, the composition is corrected depending on a setting within the redox control environment. There are three options for this: Replacing by mass, replacing my atoms or simply removing the excess.

Then, each element which is assumed to be present as a halide species gets assigned an  $\alpha$ -value, which is obtained by dividing the atomic amount of that element by the sum of all halide amounts. Each species is corrected based on the overall excess valency, their respective oxidation state as well as the amount that was originally present.

## 5.5 MELCOR

In order to simulate a severe accident scenario with the materials compositions generated using the programs above, the MELCOR code is used. It is a commercial code developed at the Sandia National Laboratories as a plant risk assessment tool.<sup>[14]</sup> Using MELCOR, the source term during a user-defined accident scenario can be determined. In addition to the core code, MELCOR features many modules which can expand on the functions of the code. Examples of such modules include simulation of core-concrete interaction, behaviour of radioactive aerosols as well as classical core-related phenomena such as cladding oxidation and core relocation.

As can be assumed from the type of modules outlined above, MELCOR was originally developed for use with typical LWRs. Because of the flexibility of the code, specific phenomena which may occur can be cherry picked and simulated for any kind of reactor setup. Therefore, it can also be adapted to work with MSRs, especially if additional supporting functionality for the chemical side of the simulation is given.

## 5.6 cGEMS

Developed to address the rather simplistic treatment of chemistry in MELCOR, cGEMS<sup>[15]</sup> combines the ability of MELCOR to simulate severe accident scenarios with the powerful

chemical speciation ability of GEMS. This allows for more accurate treatment of the source term in an accident scenario created in MELCOR by coupling it to GEMS between time steps of specified length. After each step of MELCOR analysing the releases from the system, the remaining elemental composition and temperature is passed back to GEMS, which will re-evaluate the chemical composition. The thusly determined materials are passed back to MELCOR for another time step of releases, and so forth. Specifically, MELCOR wants the vapor pressure computed by GEMS for a specific composition and temperature in order to continue its release calculations.

cGEMS is developed at PSI and serves as an interface between the two codes. By accessing and passing data to and from both codes, it allows for the desired coupling. cGEMS is programmed in C++.

## 6 Chemical-Thermodynamic Modelling of MSR Salts

### 6.1 Collecting Data and Organizing the Database

The HERACLES Database<sup>[10]</sup> is used as the baseline for any calculations run in the scope of this work. While it already contains a lot of important species for the nuclear field, it is mainly targeted towards solid-fuelled systems. Therefore, data is often given for a substance's solid and gaseous phase respectively, as they are the most important for fuel and accident release simulations. This means that even if an MSR-relevant species is present in HERACLES, it is rather likely that the liquid phase is missing. Additionally, solid-fuelled reactors don't care very much about halides, which are the main components in an MSR chemical system. Even for the halides that are present, the amount is very skewed towards the fluorides rather than chlorides, as they are very much the favoured option for the salt in research.

Generally, new species are not added directly to HERACLES. Instead, the necessary data is copied from HERACLES into a separate project database, where any new species will be introduced. This makes it so that potential issues with the data or GEMS because of the new entries cannot corrupt the local copy of HERACLES.

In order to add a new record, there are certain parameters that need to be known at a minimum. These parameters are the  $C_p$  evolution with temperature, the enthalpy of formation and the standard entropy. Often, the liquid is not given with these quantities, but rather using transition enthalpy and entropy from the solid phase. The  $C_p$  curve is usually given in one of three ways. Either, an actual graph of the curve is given, or a fit it already made by the author of the source and the parameters provided, or a constant value is given. The constant value is usually the case for liquids. In the case of a non-constant value, the data needs to be transformed into the GEMS format of reporting a  $C_p$  curve, which uses the following formula:

$$C_p = A + BT + CT^{-2} + DT^{-0.5} + ET^2 + FT^{-1}$$

where A through F are the fitting coefficients for literature  $C_p$  curves and T is the temperature. While there are a maximum of 10 fitting parameters available in GEMS, only the above-mentioned were used in the data added within this work.

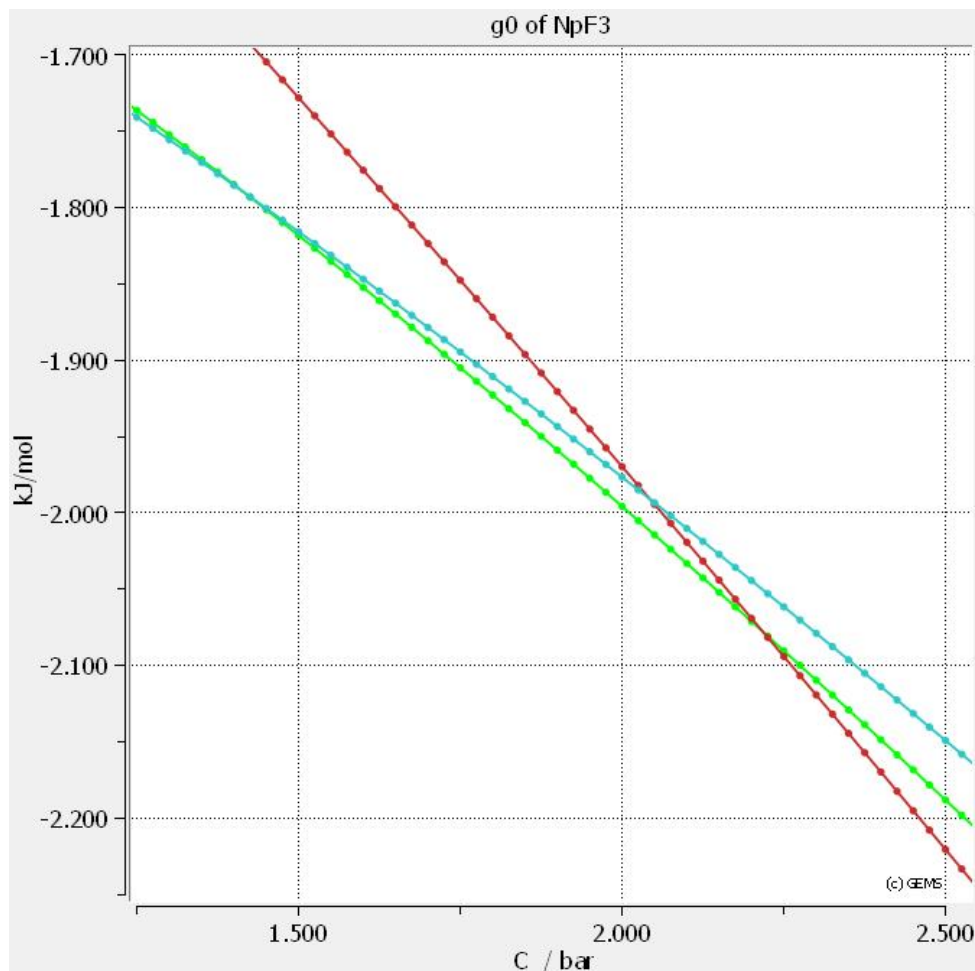


## 6.2 Methodology Example

GEMS contains a very powerful tool for checking thermodynamic data in its RTparm functionality. It allows for the user to calculate and plot various parameters as a function of temperature from the provided data, while adding additional data points to the plot to compare against. This is incredibly useful especially for adjusting a single phase of a compound in order to match melting and boiling points. In the case of the current work, the adjusted phase was usually the liquid, as it is the most difficult one to find data for. Therefore, the gaseous and solid Gibbs' energy curves were calculated separately and added as static data to the plot of the liquid phase. Then, when adjusting the parameters in the database, the calculated liquid phase diagram can be easily recalculated with the changed values.

The methodology will be illustrated in the following using the case of  $\text{NpF}_3$ , which is a species with a distinct lack of data for the liquid phase. Both the solid and gaseous phases were previously available in HERACLES, however they tend to be much easier to find as they are of elevated interest for solid-fuelled reactors.

While it is not always possible to make reasonable assumptions about missing data, in some cases there is a similar species already present in HERACLES that the generated data can lean on. In this case, that species is  $\text{UF}_3$ . To start out, the record of liquid  $\text{UF}_3$  is simply copied and the resulting entry is used to compute a temperature-dependent Gibbs' energy curve for the liquid phase. After this is done, the solid and gaseous phases are added to the RTparm entry, and a plot very similar to figure 1 is obtained. Now, the last step is to adjust the thermodynamic values in the liquid phase record such that the melting and boiling point agree with literature. In this case, literature values are a little bit suspect, however due to the lack of other options they were assumed to be correct.<sup>[16]</sup>



**Figure 1:** Plot showing the RTparm optimization of the  $\text{NpF}_3$  species. The green curve represents the liquid phase, the blue represents the solid phase and the red represents the gaseous phase.

In figure 1, one can tell where the phases transition by which Gibbs curve is the lowest at a certain temperature as they are the most thermodynamically favoured. Therefore, the intersections of neighbouring phases represent the melting and boiling points respectively.

After the initial run to produce a Gibbs' energy plot, the data has to be adjusted. There are a variety of things to adjust, so one must be aware of how each parameter affects the final shape of the Gibbs' energy curve. Because the target shape of the curve is unknown in this method, it is often reasonable to try to do as much work as possible simply changing the formation enthalpy, as this will only shift the Gibbs' energy curve up and down. Then, should it not be possible to match both melting and boiling point at the same time, the shape of the curve can be adjusted using the standard entropy as well as the  $C_p$  curve itself.

## 6.3 Specific Cases

In the scope of this work, the following species were either added to the HERACLES database or modified from the previously present data:

**Table 1:** Table of the species added to the database

Species	Notes
$\text{PuCl}_3$	Previously added during Semester Project, but adjusted to match the literature melting point. <sup>[17]</sup>
$\text{ThCl}_4$	Imported as is from Landolt-Börnstein. <sup>[18]</sup>
$\text{AmF}_3$	The gaseous and solid states were already present, however there is no reference to be found for the liquid. Because Americium is quite an important element for the equilibrium composition, it was adapted based on the literature melting point and a similarity to $\text{UF}_3$ . <sup>[19]</sup> Additionally, the solid phase heat capacity data was adjusted based on van der Meer et al. <sup>[20]</sup>
$\text{UCl}_3$	No liquid phase data found; matched manually to melting and boiling point data starting from assumed data in literature. <sup>[21]</sup>
$\text{UCl}_4$	Imported as is from Landolt-Börnstein. <sup>[18]</sup>
$\text{ZrF}_4$	Imported as is from Landolt-Börnstein. <sup>[22]</sup>
$\text{NdCl}_3$	Imported as is from Landolt-Börnstein. <sup>[22]</sup>
$\text{Np}$	Imported as is from Landolt-Börnstein. <sup>[23]</sup>
$\text{NpF}_3$	Both solid and gaseous phase were already present. Liquid is constructed from a similarity with $\text{UF}_3$ and some slightly questionable literature values. <sup>[16]</sup>
$\text{Pr}$	Imported as is from Landolt-Börnstein. <sup>[23]</sup>
$\text{PrCl}_3$	Imported as is from Landolt-Börnstein. <sup>[22]</sup>
$\text{PrF}_3$	Imported as is from Landolt-Börnstein. <sup>[22]</sup>
$\text{Na}_2\text{ThCl}_6$	Created in GEMS function ReacDC

## 6.4 Chemical-Thermodynamics of Two- and Three-Component Systems

### 6.4.1 Methodology

The above-mentioned program GEMS is a powerful tool for thermodynamic computations. It does not, however, have the innate capability of generating phase diagrams given a material composition and temperature range. Therefore, it was necessary to develop a small script within the "Process" section of the program to address this lack. It works by scanning through both the temperature as well as composition scale in defined intervals and checking at each point what phases the substances are present in. This affords a phase diagram not of the classical variety showing lines denoting phase transitions, instead one that gives for each point on the temperature x composition space an indication of what phases are present. While the resulting plot is somewhat unusual looking and rather dependent on the coarseness (or lack thereof) of the chosen scanning grid, it is a good way of making GEMS work in a way that it is not originally intended for.

In the following, scripts used by GEMS in order to make phase diagrams are shown with some small explanations. They are done separately for binary and ternary diagrams, as there are a considerable amount of differences between the two.

First of all, the control script of a binary phase diagram will be discussed. It is important to understand that this script will be executed a large number of times, rather than just once. In fact, it is run for each point on the phase diagram, meaning the number of computational steps equals  $(n + 1) * ((T_{final} - T_{initial})/T_{step} + 1)$ , where n is the number of steps to be made in terms of molar fraction of the added material.

```
$ Calculation of number of TC points
I =: floor( ( iTC[1]-iTc[0] ) / iTC[2] ) + 1;
$ Calculation of current x value from J index
x =: ( J > 1 ? floor( ( J - 1 ) / I ) * 0.02 : 0 );
xd_{ThF4} =: x + 1.0e-9;
xd_{LiF} =: ( 1 - x ) + 1.0e-9;
$ Calculation of current T value from J index
cTC =: iTC[0] + mod( J , I ) * iTC[2];
modC =: x;
$ End of script
```

Here, J is the iteration number, starting at one and counting up for each execution of the script. iTC is the temperature variable and is used to calculate the maximum number of temperature steps. To alter the composition, the xd\_ variables are used with the appropriate identifier for the species in order to overwrite the previous composition. The main thing to take away from this script is the fact that any given content of a certain species is never zero. A minimum amount of each species is added, 1.0E-9 moles in this case, in order to improve the numerical stability.

Next, the sampling script for a binary system will be discussed. While the control script essentially adjusts the relevant parameters, i.e. temperature and composition, as the program loops over it, the sampling script is what extracts the relevant data points from each iteration. These data points are what will be plotted to produce the phase diagram.

```
xp[ J ] =: modC;
yp[ J ][ 0 ] =: Xa[ { gas_gen } ];
yp[ J ][ 1 ] =: ( Xa[ { Liquid } ] > x ? cT : empty() );
yp[ J ][ 2 ] =: ( Xa[ { Li } ] > x ? cT : empty() );
yp[ J ][ 3 ] =: ( Xa[ { LiF } ] > x ? cT : empty() );
yp[ J ][ 4 ] =: ( Xa[ { Li2O } ] > x ? cT : empty() );
yp[ J ][ 5 ] =: ( Xa[ { Li3ThF7 } ] > x ? cT : empty() );
yp[ J ][ 6 ] =: ( Xa[ { Li7Th6F31 } ] > x ? cT : empty() );
yp[ J ][ 7 ] =: ( Xa[ { LiTh2F9 } ] > x ? cT : empty() );
yp[ J ][ 8 ] =: ( Xa[ { LiTh4F17 } ] > x ? cT : empty() );
yp[ J ][ 9 ] =: ( Xa[ { LiThF5 } ] > x ? cT : empty() );
yp[ J ][ 10 ] =: ( Xa[ { Th } ] > x ? cT : empty() );
yp[ J ][ 11 ] =: ( Xa[ { ThF4 } ] > x ? cT : empty() );
```

First and foremost, note that in the context of GEMS coding, Xa refers to the molar amount of the associated species. Note as well, that it is not the molar amount present in the system

being sampled; Instead, the temperature, being the y-value of the phase diagram, is saved on the condition of the molar amount exceeding a cut-off value  $x$ .

With respect to the ternary diagram, the main thing to note is that while GEMS is not especially made to construct binary diagrams, it is even less so prepared for having a ternary diagram with its three non-perpendicular axes. The only possibility coded into GEMS is that of a standard two-axis plot. In order to overcome this limitation, the ternary diagram needs to be translated into coordinates compatible with a standard plot.

```
$ Calculation of number of TC points
I =: floor((sqrt(8*(J+1))-1)/2)+1;
modC =: J - floor((I-1)*((I-1)+1)/2)+1;
$ Calculation of current x value from J index
xd_[{CsF}] =: 1.0-(I-1)*0.02 + 1.0e-8;
xd_[{LiF}] =: (modC-1) *0.02 + 1.0e-8;
xd_[{ThF4}] =: 1.0 - xd_[{CsF}] - xd_[{LiF}] + 3.0e-8;
$ Calculation of current T value from J index
cTC =: 500;
xp[J][0] =: xd_[{CsF}] + 0.5*xd_[{LiF}];
xp[J][1] =: xd_[{LiF}] * sqrt(0.75);
xp[J][2] =: xd_[{ThF4}];
$ End of script
```

Similarly to the binary version above, the ternary diagram relies on the control script to modify the composition on each execution of the script. It does so by using triangular numbers in order to translate the current iteration number  $J$  into variables along which the evolution of the composition can run, namely  $\text{modC}$  and  $I$ . Based on these variables, the composition of the system is determined, where the sum of all molar contents always amounts to one mole. Lastly, three parameters are recorded as part of the  $xp$  variable. This variable accounts for the placement of each phase indicator on the final triangular diagram, where the x-axis is controlled by the molar content of  $\text{ThF}_4$  (in this example). In order to place the calculated points along the y-axis, the molar content of  $\text{LiF}$  is saved in the sampling script exactly as is done for the temperature in the binary diagram above. Following this procedure, the regular GEMS plotter can be used without any point appearing outside of the wanted triangular plot.

Even though the obtained phase diagrams are not much of a help in analysing actual salt compositions, they are an important tool when it comes to verifying the correctness of the employed data. This applies to both the substance-specific thermodynamic data as well as the mixing parameters for the system in question.

While there are numerous elements and species present during the operation of any nuclear power reactors, the above data check is only performed for some species which have at least a somewhat reliable dataset along with being important enough to warrant optimizing interaction parameters and checking the data for accuracy.

For the easier cases, it is often enough to take literature values for the interaction parameters as well as thermodynamic values and run the simulation using the Process tab in GEMS. In the case of more complex interactions (or worse literature values), it is often necessary to do

some additional work with respect to optimizing the data such that a literature phase diagram can be accurately represented. In the following, a short insight into how such an optimization procedure might occur will be given.

Often, literature values are not given as Redlich-Kister parameters. Instead, other models such as the quasi-chemical Pelton model are often preferred. While there are options in GEMS other than the Redlich-Kister model, it is the only one used in the simulations presented in the scope of this work. This is, first and foremost, due to its considerable ease of use both in terms of theory as well as in keeping the same model across many different binary interactions.

The conversion from literature values to the Redlich-Kister variety required for the GEMS project is done simply by fitting parameters to the curve generated from the literature interaction parameters.

**Table 2:** Summary of binary interaction parameters in use for GEMS cases.

System	Redlich Kister Parameters					
LiF-ThF <sub>4</sub>	-82613.3	66.344	46951.5	-38.7547	-16323.3	12.6042
	-18361.1	18.0845				
LiF-PuF <sub>3</sub>	-15555.7	4.46043	2920.75	-0.837492	-244.518	0.0701124
	-27.3672	0.00784728				
NaCl-ThCl <sub>4</sub>	-67000	0	-29000	0	-15000	0
	10000	0				
NaCl-PuCl <sub>3</sub>	-34484.6	0	-6710.58	0	-4232.57	0
NaCl-UCl <sub>3</sub>	-40000					
ThCl <sub>4</sub> -PuCl <sub>3</sub>	-8000	0	-12000	0	-4800	0
LiF-UF <sub>4</sub>	-57230.4	20.4394	26585.5	-9.49483	7320.79	-2.61457
	-2563.14	0.915408				

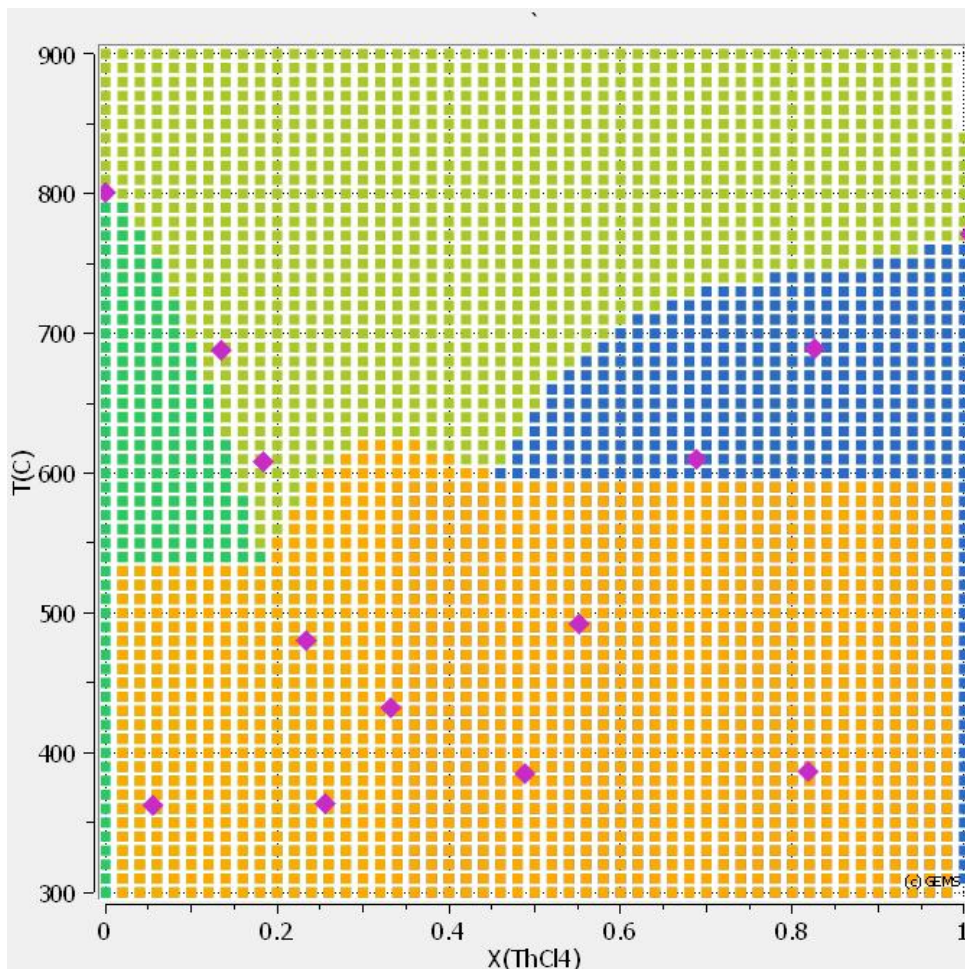
In table 2, an overview of the binary interaction parameters is given. The fitting of interaction parameters for all the described systems has been done in the current work, with an attempt to provide a detailed thermodynamic description of the MSR-related systems. The sheer amount of work that went into this task represents a significant share of the project time and would not have been possible to complete without the advice and supporting work of the supervisors. Note that every second parameter refers to a temperature-dependency, where the first uses a linear dependency. Every temperature-related coefficient thereafter increases the temperature exponent by one. Because fluoride diagrams tend to be somewhat more complicated, these coefficients are only in use for the two fluoride diagrams, however they could in theory be applied to any diagram.

The other parameters are simple Redlich-Kister values. While all binary systems show some form of interaction, provided the species mix, the complexity of the shape of the excess Gibbs' energy curve can vary wildly. As such, the number of parameters required can be vastly different; Note especially that the NaCl-UCl<sub>3</sub> system can be reliably reconstructed using only a single Redlich-Kister value, which basically tells us that the system demonstrates a regular solution behaviour.

The numbers given in table 2 are quite different in nature, being that some are rounded to thousands and others that have digits extending to after the decimal point. This difference

is to be attributed to a difference in choosing the parameters, where the easier systems can simply be made using a plotting software while varying the parameters to produce a reasonable curve. Instead, a Python<sup>[24]</sup> script can be used in order to automatically generate a fit through points chosen by the user, which generates values with rather extensive decimal digits.

One of the cases whose results will be provided below is the binary phase diagram of NaCl and ThCl<sub>4</sub>. However, when only basing the analysis on literature values, the result is not quite so attractive. In figure 2, this result is displayed. The pink dots denote lines on the reference spectrum, and it is quite clear that they do not match at all with the result shown here.

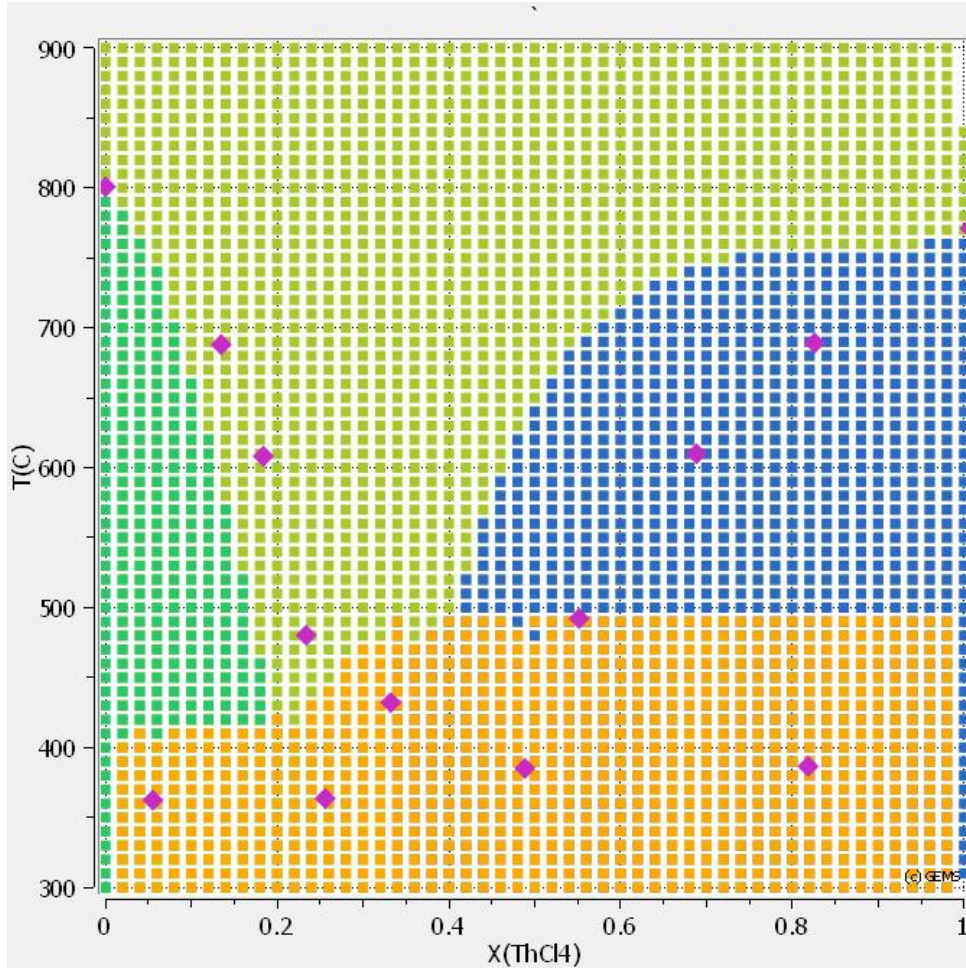


**Figure 2:** The initial outcome of running the simulation for literature interaction parameters.

From the way the graph looks, it seems that the solid solution (orange) is too stable for the expected temperature of the eutectic. Additionally, the range in which ThCl<sub>4</sub> is a pure solid extends a good bit too far into the diagram. As such, a reasonable first approach would be to modify the interaction parameters of the liquid ThCl<sub>4</sub>-NaCl system such that they are stronger as well as more shifted towards the right edge of figure 2. In order to do this, it is a good idea to have a continuous plot of the excess Gibbs' energy according to the Redlich-Kister equation introduced above. By simply applying a multiplicative modifier greater than one to each parameter, the overall shape of the excess curve remains the same, however the interaction becomes stronger.



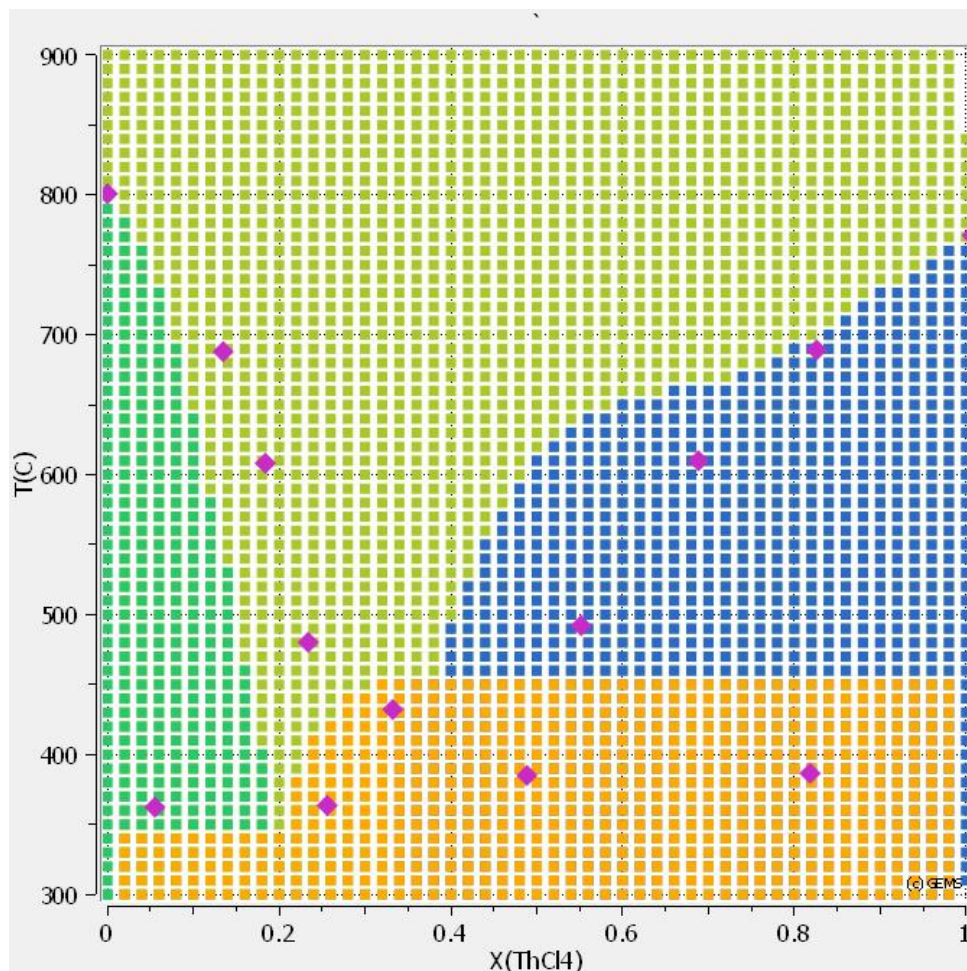
After scaling up all interaction parameters by a factor of 1.2, the temperature level of the transition out of the solid solution has already improved considerably. The  $\text{ThCl}_4$  curve has not yet improved, which was however not the purpose of this step.



**Figure 3:** The outcome of the second iteration on the  $\text{NaCl-ThCl}_4$  phase diagram.

Note that there are some issues with the boundaries of the phases in terms of their smoothness; this is a product of the fact that the project settings were not adjusted even though the system is no longer the same. Unless one is either quite close to the expected target or a lot of unusual artifacts are occurring in the diagram, it is probably not yet worth it to invest a lot of time into this kind of adjustment.

Now that the eutectic is in somewhat the correct realm in terms of temperature, it is time for the fine tuning. First of all, making the incline of the  $\text{ThCl}_4$  steeper so that it matches the entered literature values is a good approach. To do this, it is advisable to use some form of plotting software to have a continuous representation of the excess Gibbs' energy as a function of the Redlich-Kister parameters. By tuning them one by one, it becomes much easier to assess their individual impact on the shape of the curve, and therefore make more informed adjustments to the simulation.

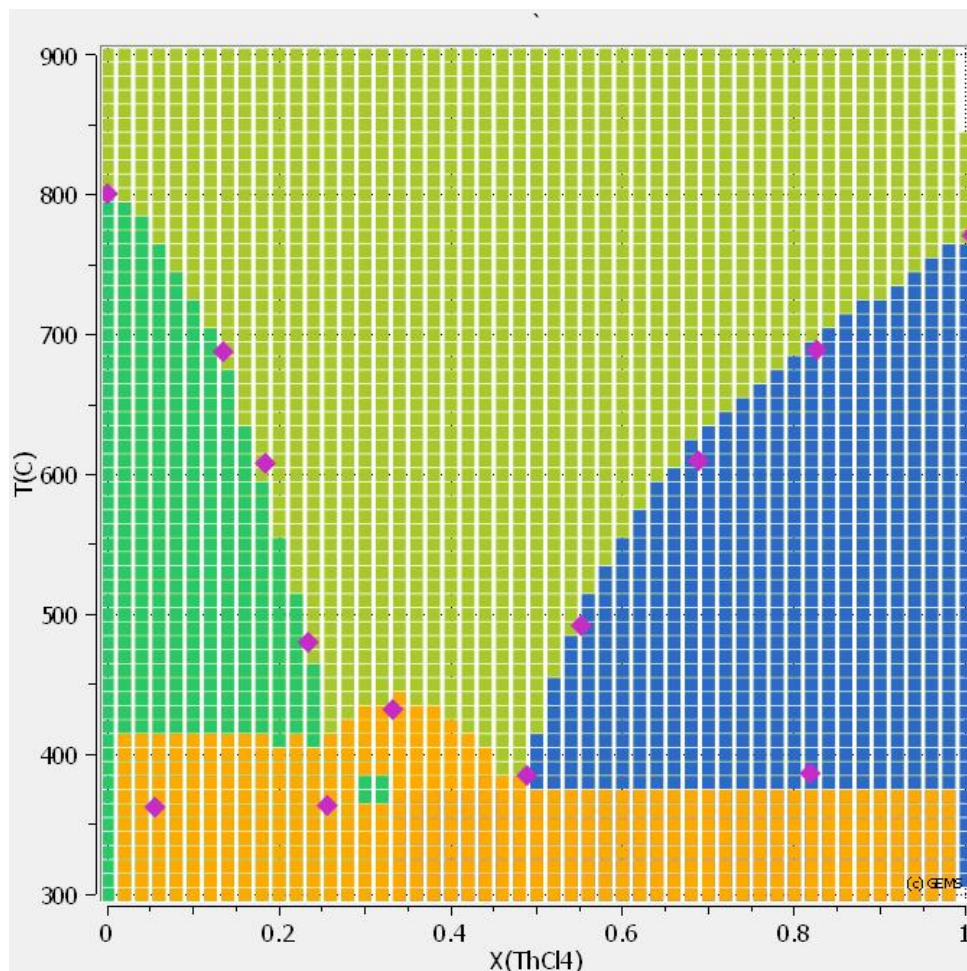


**Figure 4:** Plot of the phase diagram of the NaCl-ThCl<sub>4</sub> system after the second adjustment.

As a third step, the smoothness of the curve as well as a small step towards the desired solution in terms of temperature is taken. Note, however, the odd shape of the curve of the ThCl<sub>4</sub> compound, which is likely a result of a misplaced minimum of the excess Gibbs' energy. This will need to be one of the changes addressed in the next step.

Some significant changes were necessary in order to achieve better results than those displayed above. Both relocation of the minimum of the excess Gibbs' energy curve as well as significant adjustment of the edges to fix the pure compound curves needed doing.





**Figure 5:** The last phase diagram of NaCl-ThCl<sub>4</sub> before the final result.

In comparison to the previous iterations, the data presented in figure 5 is quite close to the literature values. The liquidus is very accurate, due to the fact that the parameters were designed in such a way that it was matched first. From there, however, it is very difficult to push the solidus down without significantly adversely impacting other parts of the diagram. Note the small NaCl square near the Na<sub>2</sub>ThCl<sub>6</sub> liquidus. This is another artifact created by numerical instability of the system - it is however a decent run overall, with very smooth liquidus lines across the spectrum.

While the resulting data presented in figure 5 is quite decent, a point can be made for taking an intermediate road of trying to match the liquidus as well as possible while investing a little more into adjusting the interaction parameters in favour of accuracy with respect to the solidus. This approach was selected for the final NaCl-ThCl<sub>4</sub> diagram presented in section 6.4.2 below.

**Table 3:** Redlich-Kister coefficient evolution during the adaptation of the NaCl-ThCl<sub>4</sub> diagram.

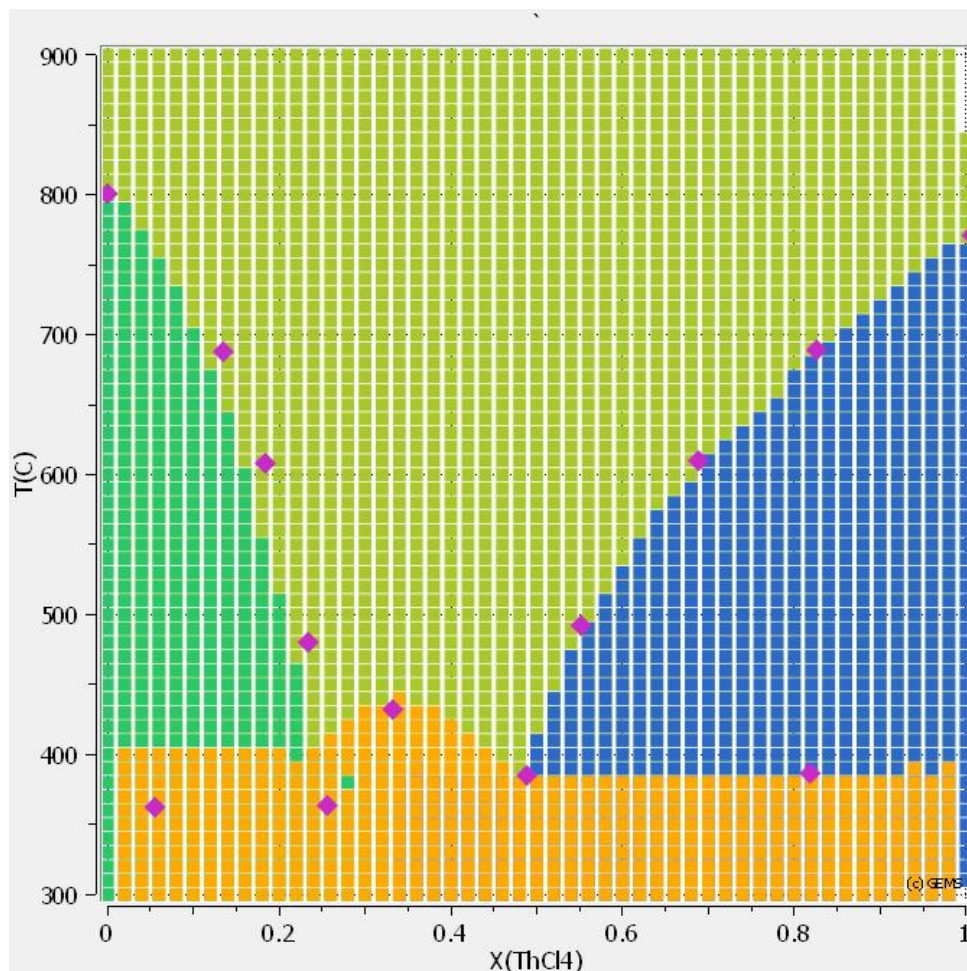
Redlich-Kister Coefficients				Reference	Reason for Use
-44120	-45902	-14244	0	Figure 2	The original coefficients, generated from a Python script based on placement of literature eutectic.
-53052	-55082.4	-18092.8	0	Figure 3	Obtained by scaling up the initial parameters by 1.2 to push down the solidus.
-60000	-30000	-40000	0	Figure 4	Adjusted values to widen the excess Gibbs' energy curve.
-68000	-28000	-15000	0	Figure 5	Adjusted from previous step in order to more closely match the solidus and eutectic area.
-67000	-29000	-15000	10000	Figure 6	Slight change to previous step to account for the introduction of the fourth Redlich-Kister parameter, with the goal of bringing the pure NaCl and ThCl <sub>4</sub> lines into line.

### 6.4.2 Chlorides

For the chloride data, the following species are considered for phase diagrams: NaCl, PuCl<sub>3</sub>, UCl<sub>3</sub>, ThCl<sub>4</sub>. Contrary to the fluorides, obtaining literature value for interaction parameters as well as phase diagrams to compare against can be a tricky endeavour. As such, some phase diagrams initial set of Redlich-Kister parameters is already an educated guess, rather than being purely literature-based.

#### NaCl - ThCl<sub>4</sub>

Being the diagram that is the final product of the approach outlined in section 6.4.1 above, the general shape is quite close to literature values. However, the solidus line underneath the NaCl is too high by quite a margin. In addition, the NaCl curve is slightly below the temperature that is expected in literature.



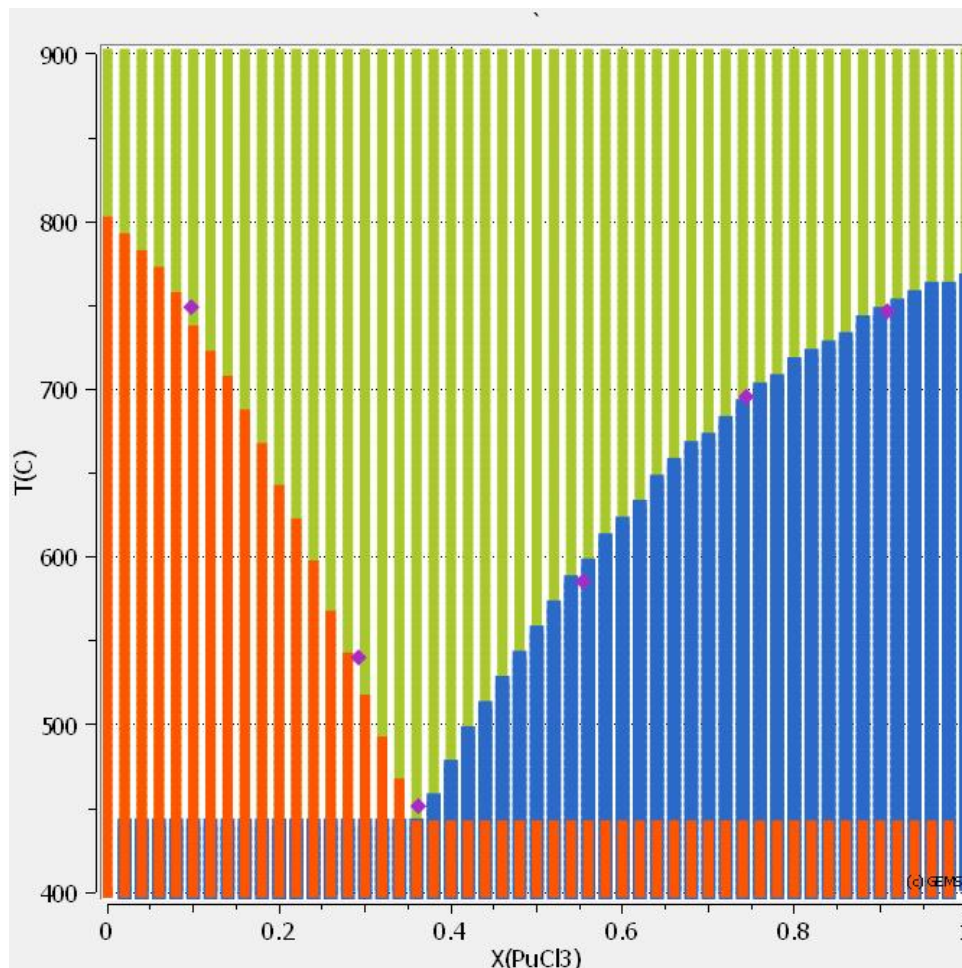
**Figure 6:** Phase diagram for the NaCl-ThCl<sub>4</sub> system, with NaCl in dark green, ThCl<sub>4</sub> in blue, the Na<sub>2</sub>ThCl<sub>6</sub> compound in orange and the liquid in light green. Overlaid in pink are literature values<sup>[25]</sup> to show where the diagram is accurate.

There are a variety of possible and probable things which may cause these issues in the diagram. First and foremost, it is notable that the Na<sub>2</sub>ThCl<sub>6</sub> compound takes its thermodynamic data from the ReacDC functionality of GEMS, which may introduce some errors that are hard to fix. While the general shape of the solidus line is quite decent, there are a lot of unwanted changes that occur when trying to push down the solidus to the left of the eutectics. Part of this issue may also be attributed to numeric issues within GEMS, as pushing down the minimum of the excess Gibbs' energy can destabilize the Na-Th compound massively.

It may be possible to address some of the problems encountered in the making of this diagram by introducing a temperature dependency in the interaction parameters, i.e. the excess Gibbs' energy curve. To do so, it would make the most sense to try to push the solidus line into position across the whole spectrum without regard for changes in the liquidus line to either side of the eutectic. Then, by introducing a temperature dependency, the liquidus may be fixed without displacing the matched solidus by adjusting the interaction parameters to be the same at the solidus temperature.

## NaCl - PuCl<sub>3</sub>

The NaCl - PuCl<sub>3</sub> phase diagram is relatively simple, with a single eutectic point at a molar fraction of PuCl<sub>3</sub> of around 0.36. The interaction parameters for the liquid phase were adapted from Beneš and Konings.<sup>[21]</sup>



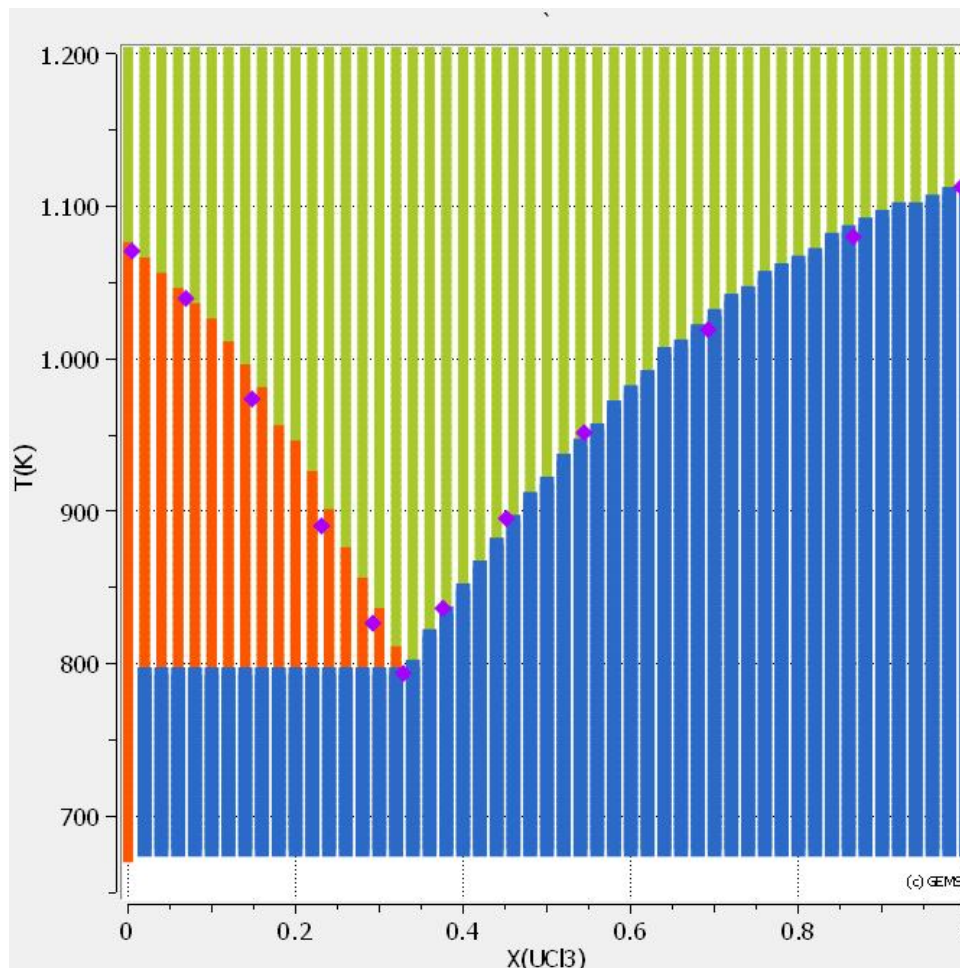
**Figure 7:** Plot of a phase diagram of the NaCl-PuCl<sub>3</sub> system, where NaCl is represented in orange, PuCl<sub>3</sub> is represented in blue and the liquid is given in green. Note the pink points on the diagram, which are overlaid literature values.<sup>[26]</sup>

The NaCl-PuCl<sub>3</sub> diagram is rather unique in the fact that it required little to no adjustment to the literature interaction parameters. Conversely, even other simple diagrams like that of NaCl-PuCl<sub>3</sub> above take some effort to accurately match, even though the initial fit tends to be better for simpler diagrams. Crucially, relying on literature interaction data also requires the thermodynamic data for the compounds themselves to be the same to even have a chance of getting a good result immediately. This is a rather unlikely thing, as a lot of data for both reference diagrams as well as the HERACLES database used in the scope of this work are not necessarily based on scientific data. Instead, a considerable portion of the data, especially for compounds only relevant to nuclear research, lacks a solid experimentative foundation and is therefore in need of adjustment or an educated guess in order to complete the dataset.



### NaCl - UCl<sub>3</sub>

The UCl<sub>3</sub> version of the binary diagram with NaCl is quite similar to those above. The eutectic point is located at a very similar molar fraction, however at a much higher temperature. Once again, the initial interaction parameters were taken from Beneš and Konings.<sup>[21]</sup>



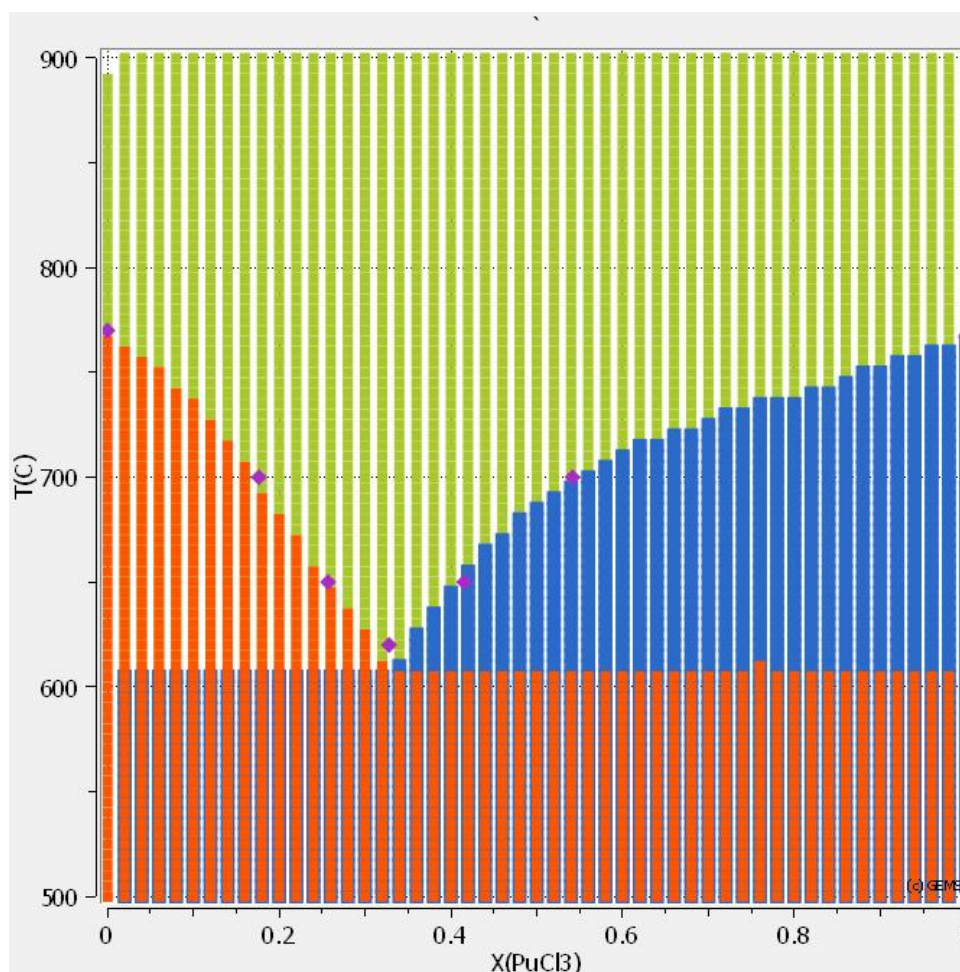
**Figure 8:** Shown here is the phase diagram of the NaCl-UCl<sub>3</sub> system, with NaCl in orange, UCl<sub>3</sub> in blue and the combined liquid phase in green. Additionally, literature liquidus values are given in pink.<sup>[21]</sup>

The reference point for the eutectic is located somewhat left and at a lower temperature than what the phase diagram shows, however it is worth remembering that the resolution of the phase diagram is limited by choice to 0.02 mol fraction per step as well as 10 degrees vertically, meaning it may well be that the eutectic is placed better than it seems. There is, however, also a small discrepancy in the slope of the liquidus line in the NaCl-rich region, so some small adjustments to the interaction parameters may be able to improve the diagram slightly.

### ThCl<sub>4</sub> - PuCl<sub>3</sub>

Once again, the diagram is rather simple. Compared to the previously described chloride diagrams, the ThCl<sub>4</sub>-PuCl<sub>3</sub> version is a lot more flat. Both pure substances have a relatively

low melting point as is, however their interaction parameters are simply not as strong as for some other binary combinations.



**Figure 9:** Plot of the  $\text{ThCl}_4$ - $\text{PuCl}_3$  phase diagram, with  $\text{ThCl}_4$  in orange,  $\text{PuCl}_3$  in blue and the liquid phase in green. The pink rhombuses represent literature liquidus values.<sup>[27]</sup>

Because of a lack of reference for interaction parameters, even the initial guess for the values is not based on any literature. Instead, it is based on an educated guess by the author as well as the knowledge where the eutectic is supposed to lie.

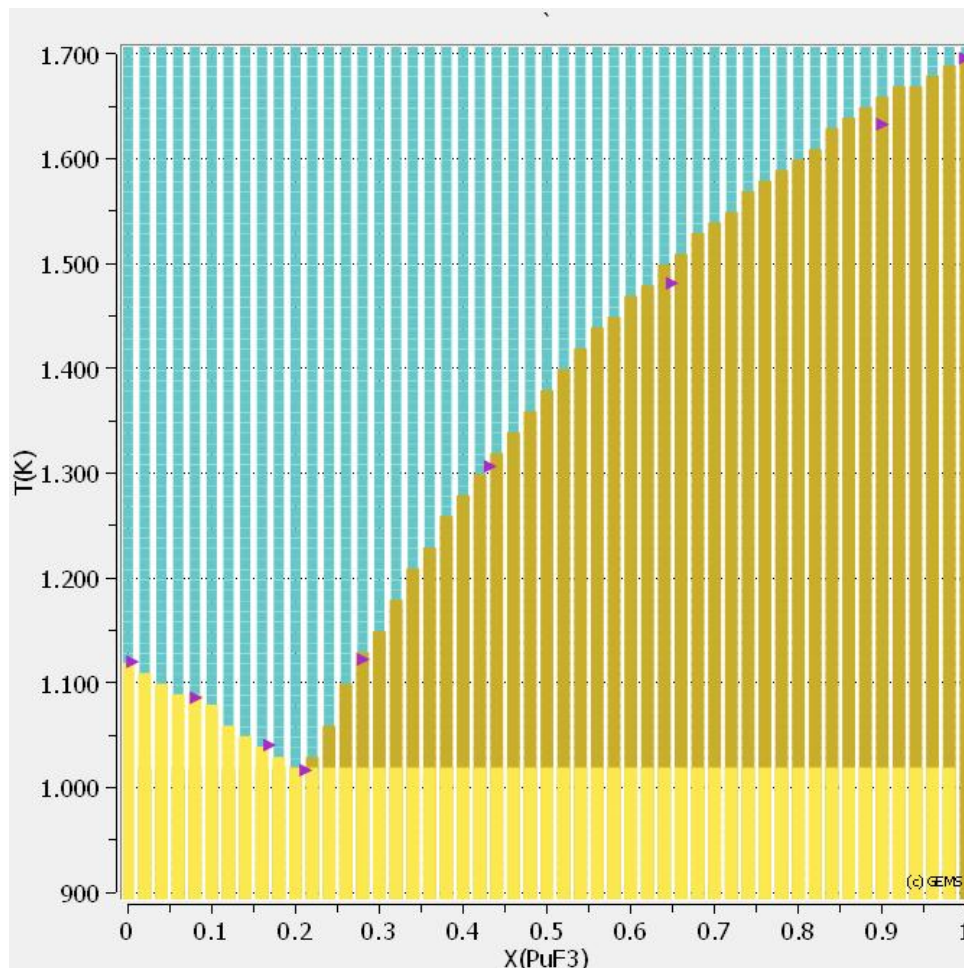
### 6.4.3 Fluorides

As opposed to the above phase diagrams of the chlorides, which tend to be on the simpler end of the spectrum, fluoride-species diagrams are not that straightforward. Because of the large amount of compounds which can be created from two fluoride salts, there is often a strong dependence on the project settings, as any instability can significantly move the line between two compounds. The sheer number of compounds can also introduce an issue where the region in which a species is expected to occur is so small that it is very difficult to find parameters which allow it to show up at all.



### LiF - PuF<sub>3</sub>

For this diagram, the initial interaction parameters were adapted from Capelli et al.,<sup>[28]</sup> who use the modified quasichemical model not available in GEMS at this time, however there were significant adjustments needed in order to make the final diagram look at all like its literature counterparts.



**Figure 10:** Shown here is the phase diagram of LiF-PuF<sub>3</sub>, where LiF is given in yellow, PuF<sub>3</sub> in brown and the liquid in blue. The purple triangles display literature values to compare against.<sup>[29]</sup>

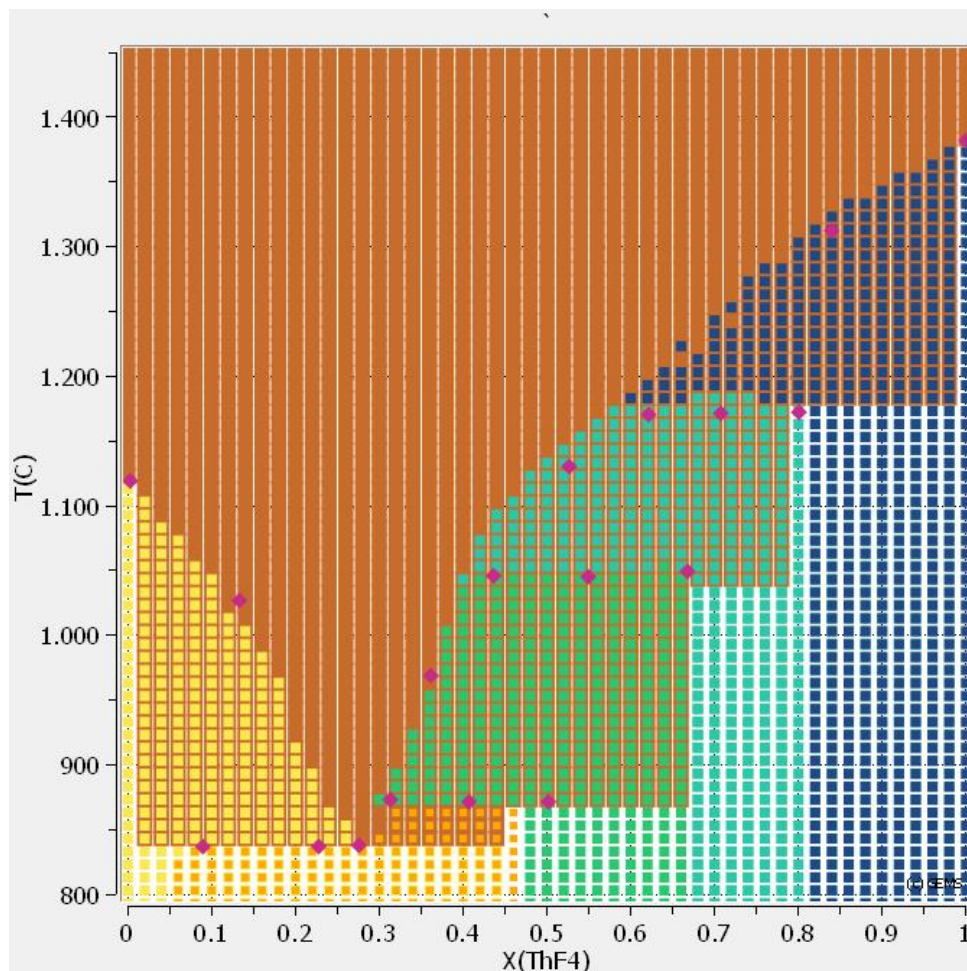
### LiF - ThF<sub>4</sub>

The LiF-ThF<sub>4</sub> is a very complex diagram due to the large amount of compounds other than the pure elemental fluorides. In addition to adjusting the interaction parameters to make the diagram, there are temperature dependencies as well as the changing of the thermodynamic parameters of compounds. The initial set of interaction parameters was taken from Capelli et al.,<sup>[28]</sup> however there is little resemblance left with what the system is based on now.

**Table 4:** This table outlines the changes to the underlying thermodynamic data of species relevant to the LiF-ThF<sub>4</sub> system. Species not mentioned are left at literature values.

Species	Initial Formation Enthalpy [J/mol]	Change[J/mol]
Li <sub>7</sub> Th <sub>6</sub> F <sub>31</sub>	-16974917	+5300
LiTh <sub>4</sub> F <sub>17</sub>	-9021140	-16700
Li <sub>3</sub> ThF <sub>7</sub>	-3960259	+2000
LiTh <sub>2</sub> F <sub>9</sub>	-4829731	-2200

In addition to the expected adjustments to the interaction parameters, some thermodynamic parameters were also changed in order to get this diagram to come close to the literature version. It is a little unclear whether or not it may have been possible to achieve more accuracy without altering the thermodynamic values of the underlying species, however the species are mostly relevant to direct LiF-ThF<sub>4</sub> interaction and the changes were not especially big. All in all, the benefit seems to be greater than the cost for the moment, however there may be future implications for ternary diagrams or other multi-species uses of the data.

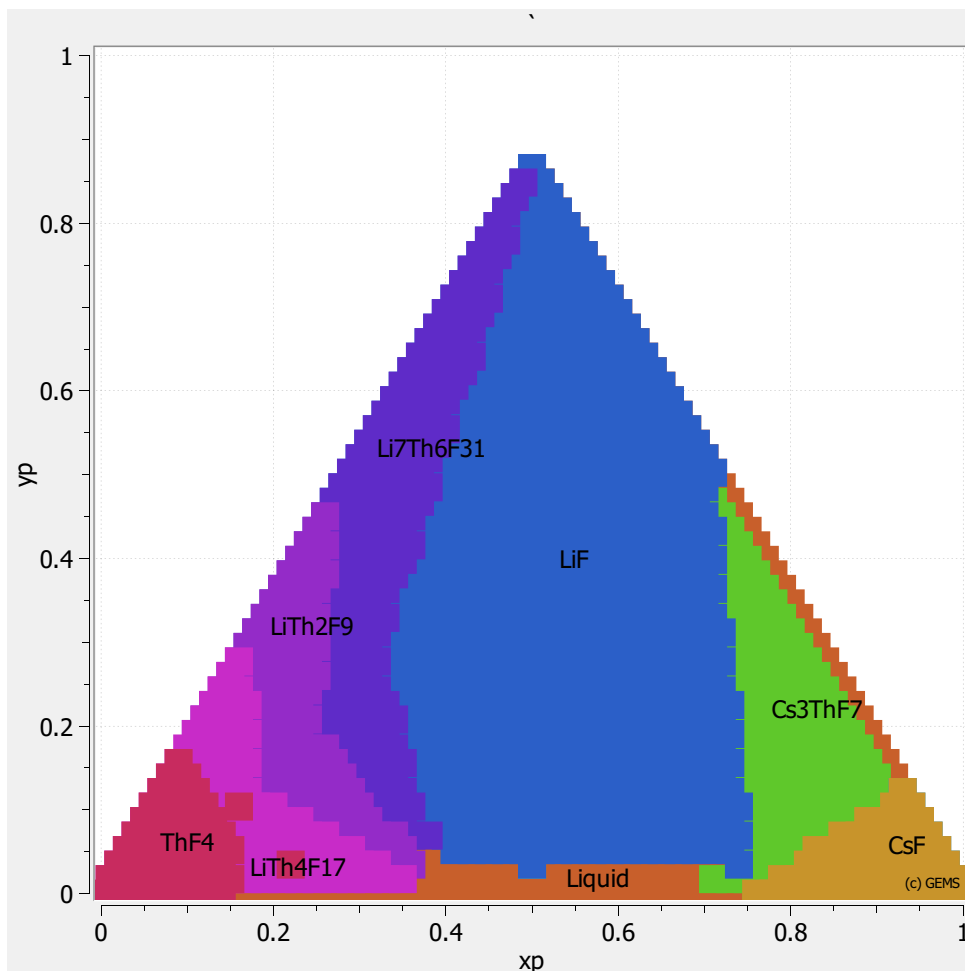


**Figure 11:** Phase diagram of the  $\text{ThF}_4$ - $\text{LiF}$  system, with the pure compounds in yellow and dark blue respectively. Additionally, there are three mixed compounds present on the diagram, namely  $\text{Li}_7\text{Th}_6\text{F}_{31}$  in orange,  $\text{LiTh}_2\text{F}_9$  in green and  $\text{LiTh}_4\text{F}_{17}$  in light blue. Literature values are given as a pink rhombus.<sup>[30]</sup>

While the general shape of the phase diagram in figure 11 is rather accurate, there are still a number of inaccuracies. Especially the liquidus line to the right of the eutectic is neither perfectly aligned with literature parameters nor especially stable. On top of that, literature diagrams suggest that there should be two separate eutectics with a small liquidus bump in between. In figure 11, the two literature eutectics are displayed as the two pink rhombi at the same temperature, around 0.25 on the molar fraction scale.

#### 6.4.4 Ternary Diagram

There is only one ternary diagram produced in the scope of this work, which is displayed in figure 12. This is mostly due to a lack of time; Once the control script for GEMS is made, it is simply a matter of parameter optimization similar to the usual routine for binary phase diagrams. However, the ternary diagrams tend to be a good bit more unstable, which makes any optimization in terms of parameters more difficult.



**Figure 12:** Ternary phase diagram of the CsF-LiF-ThF<sub>4</sub> system for 500 °C.

In addition to being more difficult to make, it is also quite a challenge to properly understand when a ternary diagram made in GEMS comes close to a literature reference.<sup>[31]</sup> This is due to GEMS ternary diagrams being made for a specific temperature, rather than having the isotherms and crystallization fields commonly found in literature.

Generally, the main approach to verify this phase diagram using a reference, is to look at which phases exist and in which region of the diagram. Crystallization fields essentially describe the main solid phase that is found if a liquid of a certain composition is cooled. This also means, however, that it is not necessarily the only solid that is found, and overlap is not at all uncommon. Note as well that these fields can change with temperature, especially so in a simulation as prone to instability - and therefore project parameter changes - as this.

The phase diagram presented above is not particularly useful to predict the melting point of a given mixture, seeing as it is fixed to be for one temperature only. Instead, it is likely of more use for a solid overview over a system which is designed for a certain temperature, especially as there are easier ways for a program like GEMS to scan for the melting point of a given mixture.

Concerning the actual speciation of the regions given on the ternary diagram, there are a lot of compounds missing which are in the HERACLES database as well as the reference diagram. While it is absolutely possible for their data to be incorrect to a point where they are too

unstable to appear, it is also an option for the temperature chosen to be outside of their stability range. In the entire composition range, no binary compounds of Lithium and Caesium fluoride appear, however there is only one possible species which occupies an extremely small space in the reference. Considering the difference in procedure of making the diagram as well as the problematic stability of the system, it is unsurprising to lose such a species.

A bigger problem arises when considering the Caesium - Thorium binaries, which seem to make up a much larger part of the diagram in the reference. While  $\text{Cs}_3\text{ThF}_7$  is present in figure 12,  $\text{CsTh}_6\text{F}_{25}$  is missing entirely. It is likely that the shape, size and placement of the phase would not be the same as in literature, however one would expect it to at least be present in some form. The same is true for some of the smaller patches of Cs-Th binaries, which should occupy a much smaller, but very nearby, region of the diagram. The fact that the region in question shows no other phase containing any Caesium suggests that these compounds have already melted at the  $500^\circ\text{C}$  used for this diagram. This is rather contradictory to the isotherms which populate these crystallization fields in Capelli et al.<sup>[31]</sup>

## 6.5 Speciation and Chemical-Thermodynamic Analysis of Equilibrium Composition

From the procedure outlined above, the isotopic composition of a molten salt reactor's fuel is obtained. While the isotopic composition is crucial for neutronic analysis, it does not give any indication of the materials behaviour as a response to temperature excursion. Only the combination of isotopic knowledge with chemical analysis can give an understanding of the release of substances and the potential danger during an accident scenario. In order to be able to use the isotopic composition in a chemical thermodynamics simulation such as one performed in GEMS, the composition needs to be translated into an elemental one. This is conveniently done automatically by EQL0D.

After the elemental composition is established, some concessions need to be made with respect to the realizability of the project. There are a large number of elements present in the final composition, which is owed to the fact that the fission product distribution is effectively a matter of probability. Every element in the periodic table will eventually be added to the salt, be it directly from fission or via the decay of some other isotope. As such, manually collecting enough data and making the case run with all elements involved is a near impossible task. Therefore, it is necessary to reduce the case's complexity.

By only using a set of elements which make up the majority of the salt, its major characteristics can be approximated. Additionally, many elements (especially those of the metallic variety) have a multitude of options when it comes to valency. Often, only a small subset of the possibilities is accounted for in the final product. This is mostly due to a lack of access to the necessary data. Some special techniques can be applied in order to approximate or adjust data such that certain criteria of the project are fulfilled. Those will be discussed individually where necessary. Often, groups of elements (such as the Lanthanides) behave very similarly when it comes to valency as well as Gibbs' energy curves. This makes it easier to approximate the necessary data in case it is not available in literature.

After the base salt composition has been established, the data found and/or made and the computation of chemical speciation performed, additional trace elements can be added. The amount of elements of interest added is, however, limited to those which have reliable data available in the underlying database. Instead, elements of interest with respect to speciation can be added to the base salt and removed after their thermodynamic behaviour has been sufficiently analysed. As mentioned above, these trace elements are very unlikely to have a major impact on the salt and interactions between different ones can safely be disregarded.

In order to have an elemental composition to base the GEMS runs on, the above-mentioned MSFR-like cases are used. Therefore, a total of four compositions are available which cover both fluoride and chloride salts, as well as Thorium and Uranium fertile options.

### 6.5.1 Base Salt

The base salt is meant to be consisting of only the most prevalent elements, as outlined above. There will obviously be differences between the four cases of interest, as their carrier salt as well as fertile species differ. Additionally, there are limitations as to what data is available, so some elements are skipped or are missing certain species. Missing species especially include all forms of Curium, for which it is nigh impossible to find proper thermodynamic data. Technically, it may be possible to assume certain characteristics about a missing species, however information about Curium halides are so scarce that there is nothing to base such assumptions on.

It needs to be noted that the halogen content of the elemental mixture needs to be changed from the resulting values of EQL0D. This is because when cutting off the elements which are taken into consideration, a lot of possible cations for halide salts are removed as well. Therefore, there is no exact right answer to how much of the respective halogen should be taken into consideration for the GEMS run, and an educated guess or some form of analysis needs to be done. Alternatively, it is possible to use the "Process" feature of GEMS in order to take snapshots of various halogen amounts. This would enable the user to make an attempt at finding an optimum, while also investigating possible consequences of deviating from the ideal redox conditions. This does, however, require additional data to be collected such that elements have multiple options with respect to halide valency.

The basis of the GEMS simulations is the fluoride-based Uranium equilibrium salt presented in table 10 below. Note that while there are a total of four cases presented, any further computation done on this topic will be performed with only one fuel salt. While the MSFR is traditionally considered to operate using a Thorium-based salt, a trade-off has to be made in the scope of the simulations presented below. This trade-off originates from the fact that more work has been put into the LiF-ThF<sub>4</sub> diagram than the other carrier salts, however the required thermodynamic data for some of the elements present in its resulting composition are not available for use with GEMS. As such, the Uranium-based system is chosen. For the purposes of these calculations, a standard temperature of 750 °C was chosen.<sup>[32]</sup>

Because of the fact that only the top 10 elements will be considered in the following, some adjustments have to be made with respect to the fluorine content. In this case, fluorine is the only element that will commonly take the role of the anion in the chemical context of salt. However, a considerable amount of cations have been cut, which suggests that the amount of

fluorine in the system may need to be adjusted to account for this. Reasonably, one option to go for here is to assume perfect stoichiometry for the cations which are included in the system. Such an assumption is taken here, and will mean that the amount of fluorine employed is  $4.77\text{E-}02$  moles. While the data of table 10 is in fact given in atomic density, it is more straightforward to use moles in the context of GEMS. As long as the ratio between elements remains the same, there should be no impact on the final results, so the numbers for atomic density are simply taken for moles as well. For stability, some gaseous Argon was also added to the system, which should not affect the outcome in any way.

### 6.5.2 Results

The first step in making an analysis of the chemical composition of the system in question is to run a simple equilibrium calculation using pure elements as input. Such a run was performed, with the results being summarized in table 5. This table only shows the content of the phases, rather than individual species; A more detailed breakdown of the liquid phase composition can be found in the appendix. Because of the fluorine content being tuned to match the expected valency of the cations, most of the results behave about as expected. Note that while it may appear that there is a good amount of gas being formed, 0.01 moles of this is Argon introduced to the system for stability as outline above.

**Table 5:** This table shows the results of a standard run using the fluoride/Uranium composition.

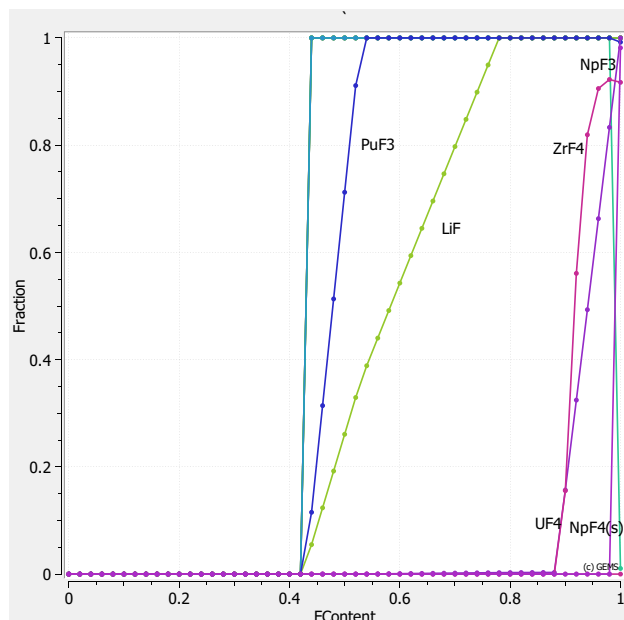
Phase	Molar Content
Gas	0.010003045
Liquid	0.02744012
$\text{NpF}_4$	$6.5268866\text{e-}006$

As expected, the system at its intended, stoichiometric setup is almost entirely in the liquid phase. The only outlier is the presence of  $\text{NpF}_4$  in its solid state. This is unexpected in two ways, as Neptunium was assumed to naturally take its trivalent form. This means that not only does there seem to be a small amount of solid forming, but also that there will necessarily be another deviation from expected valency elsewhere to make up for the formation of  $\text{NpF}_4$ . Because of the fact that, in chemical thermodynamics, everything is in equilibrium, there are a lot of elements which have a small part of their speciation be in a lower-valency state. This means that it is hard to pinpoint where exactly the reason for the generation of tetravalent Neptunium lies.

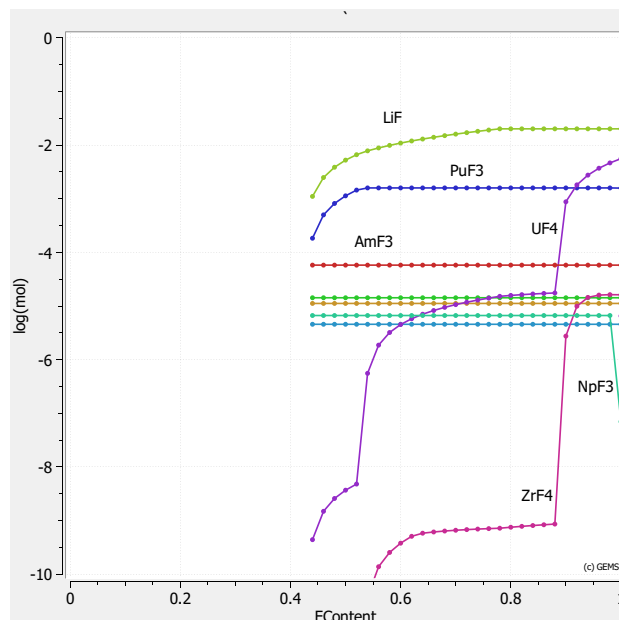
With respect to the solidification of the Neptunium compound it needs to be noted that there is no equivalent tetravalent compound available in the liquid phase of the GEMS model. Seeing as liquid phase data tends to be much more limited in literature, it is not surprising that the liquid compound options are limited to tri- or hexavalent options. In this case, where the redox conditions enter a space where  $\text{NpF}_4$  seems to be the favoured option, it may be that the disproportionation to  $\text{NpF}_3$  and  $\text{NpF}_6$  is less favourable than staying in the solid phase at high temperatures. In figure 13, a lot of components which are part of the system are not explicitly named. This is because they perfectly overlap and are all found under the leftmost peak on the plot. These compounds include the fluorides of Cerium, Neodymium, Neptunium and Praseodymium, however during the last step to stoichiometric fluorine, the content of liquid



$\text{NpF}_3$  drops to make way for solid  $\text{NpF}_4$ . This higher fluoride is solid at the conditions of the simulations.  $\text{ZrF}_4$  follows a quite rounded peak which stays below a liquid fraction of one, however no solid is created to make up for this deficit. Instead, the species becomes gaseous. The same is true for Plutonium species, however  $\text{PuF}_3$  makes up essentially all of the Plutonium content for a large range of redox conditions and is a liquid. Only at high fluorine content, some parts of the composition convert to  $\text{PuF}_4$ , which tends to be more volatile.



**Figure 13:** Evolution of the fraction of an element which is present in its fluorinated form as a function of the fluorine content. Both axes are scaled; The y-axis is scaled to the total amount of each pure element present and the x-axis is scaled to the stoichiometric amount of fluorine.



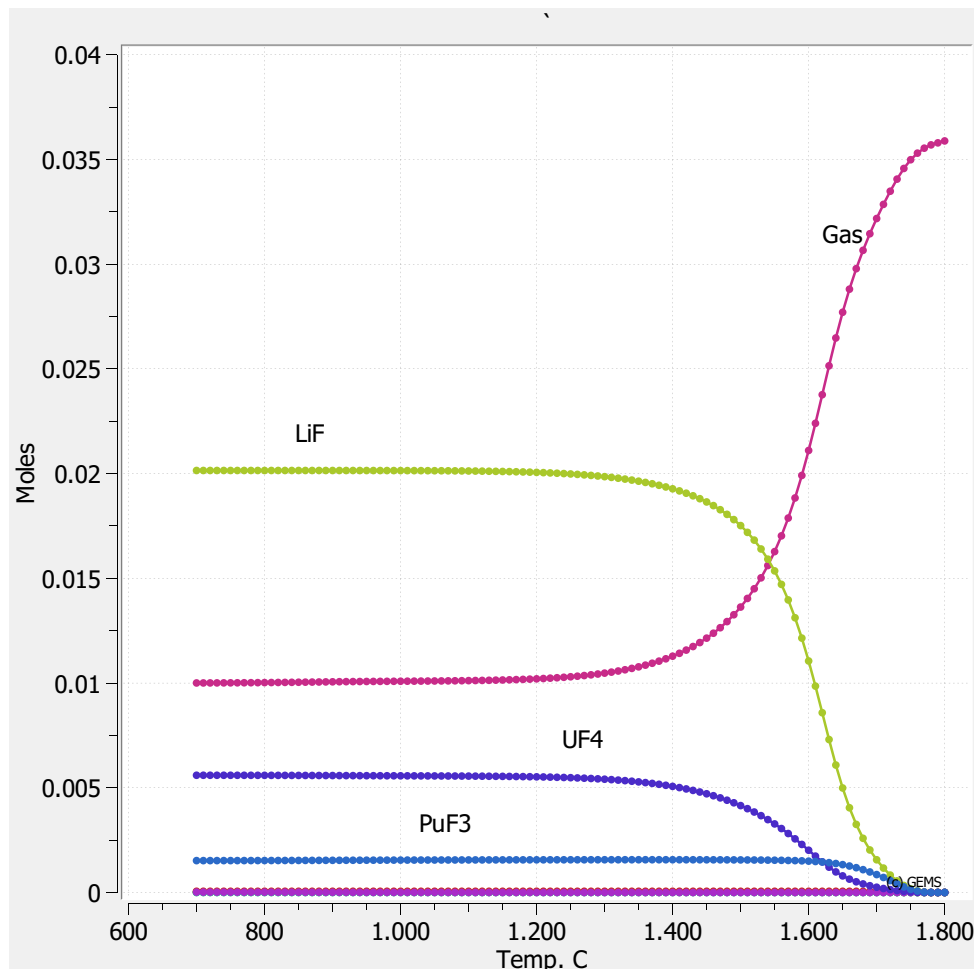
**Figure 14:** This shows the same information as figure 13, however plotting in logarithmic scale to resolve some of the overlaps.

Obviously, the chosen value of using stoichiometric fluorine comes with its share of issues. After about 40% of the stoichiometric fluorine amount is added, a lot of species start entering the liquid phase. Crucially, this does not include  $\text{UF}_4$ , which only starts to become soluble at much stronger redox conditions. This would make a difficult decision between the solubility of  $\text{UF}_4$  and the generation of solid materials like  $\text{NpF}_4$  necessary, assuming all results obtained in the course of this work are realistic.

While the picture may change with the addition of more species or adjustment of base data, the results put forward here are a strong example of what sort of analysis the addition of chemical thermodynamics can bring to the table. Rather than making assumptions on what species are present as well as what values parameters such as vapor pressure or activity may take, they can simply be simulated. Clearly, there are additional parameters which can be tuned to fit the requirements of the reactor design and can definitely change the outcome of a simulation like this.



On top of the investigation on the influence of redox conditions on the salt composition, a small investigation into the effect of temperature was performed. To do this, the standard fuel mixture presented in table 5 is investigated over a temperature range between 700 °C and 1800 °C in steps of 10 degrees.



**Figure 15:** Evolution of the most prevalent compounds of the fluoride/Uranium system with temperature increase.

The generation of gaseous products begins at around 1300 °C with the evaporation of LiF and UF<sub>4</sub>. While there are surely more compounds becoming gaseous, it is difficult to tell due to the extremely low amount present in the system. The only other major component which is visible in figure 15 is PuF<sub>3</sub>. It remains in the liquid phase for much longer than the previous two compounds.

As for the minor species, most of them evaporate at very high temperatures. AmF<sub>3</sub>, NdF<sub>3</sub> and PrF<sub>3</sub> all start evaporating around 1650 °C, whereas ZrF<sub>4</sub> already starts to disappear from the liquid at 800 °C. Even below that temperature, the content in the liquid is slowly declining. NpF<sub>3</sub> does not seem to follow any normal curve; the content as such is extremely low and the curve shows a lot of changes in slope for seemingly no reason. There is even an edge in the evolution of NpF<sub>3</sub> content at around 900 °C, which may be a simple artifact of the simulation due to the comparatively tiny amount of substance in the system.

Part of the original plan with respect to using GEMS as an analysis tool was to be able to make statements about the physical and chemical state of problematic isotopes during an accident scenario. This is also the main appeal of having a relatively simple representative salt composition which GEMS can handle rather easily; One can replicate the general chemical environment of an equilibrium fuel salt without going overboard in terms of numerical complexity as well as data collection effort. Clearly, removing most of the elements from the system will also remove certain compounds that elements can potentially form, however the extremely low content of most elements as their rank in terms of prevalence drops can justify these exclusions. In the scope of this work, a single example was run in order to demonstrate the feasibility of this work. It is quite simple to run more cases such as this one, however more time will need to be invested into identifying elements of interest in the first place.

For the purposes of this example, the element Caesium was chosen. It is of high importance during an accident scenario because of its problematic  $^{137}\text{Cs}$  isotope. In order to accommodate the addition of another cation, the fluorine amount was increased proportionally, assuming a valency of one for Caesium.

**Table 6:** Distribution of Caesium in the fluoride/Uranium base salt.

Species	Molar Content
Cs(g)	1.9124762e-020
Cs <sub>2</sub> F <sub>2</sub> (g)	1.7874526e-015
CsF(g)	3.2386034e-010
Cs(l)	1.2275183e-020
CsF(l)	9.7874914e-007

Generally, the distribution is pretty much what was expected. The most prevalent species in both the gaseous as well as the liquid phase is the common CsF salt. Even though CsF is part of the group of species typically under investigation for volatility during an accident scenario, there is not much of this behaviour observed at the 800 °C of this simulation.

With the exclusion of less prevalent elements, the availability of cations is reduced. However, this reduction also applies to anions, which may be a crucial missing piece in the case of Caesium. In previous analyses,<sup>[15]</sup> the iodide of Caesium is often considered as well, which is obviously omitted in this case. Therefore, the statement on Caesium volatility needs to be limited to the context of the simulation, and additional simulations need to be performed for a more comprehensive analysis.

## 6.6 Release Analysis in cGEMS

In order to illustrate the impact of the changes made to the HERACLES database in the scope of this thesis, a previously published cGEMS simulation<sup>[15]</sup> was repeated using the updated values.

The simulation uses a very simplified salt as well as geometry. The geometry considered is based on a molten-through cylindrical fuel casing in a cylindrical container, where the molten salt is distributed across the floor of the container. There are two heat exchangers located at the top of the container, 15 metres above the floor on which the salt is distributed. Initially, the container

is filled with nitrogen gas at 1 bar. In this setup, the temperature range covered is 300K-1500K using a heating rate of 6K/min. The heat exchangers are limited to a maximum temperature of 500K throughout the simulation. 300K is extremely low for a molten salt, however it can be useful to investigate other thermodynamic phenomena outside of releases. Such analyses exceed the scope of this thesis and will not be discussed.

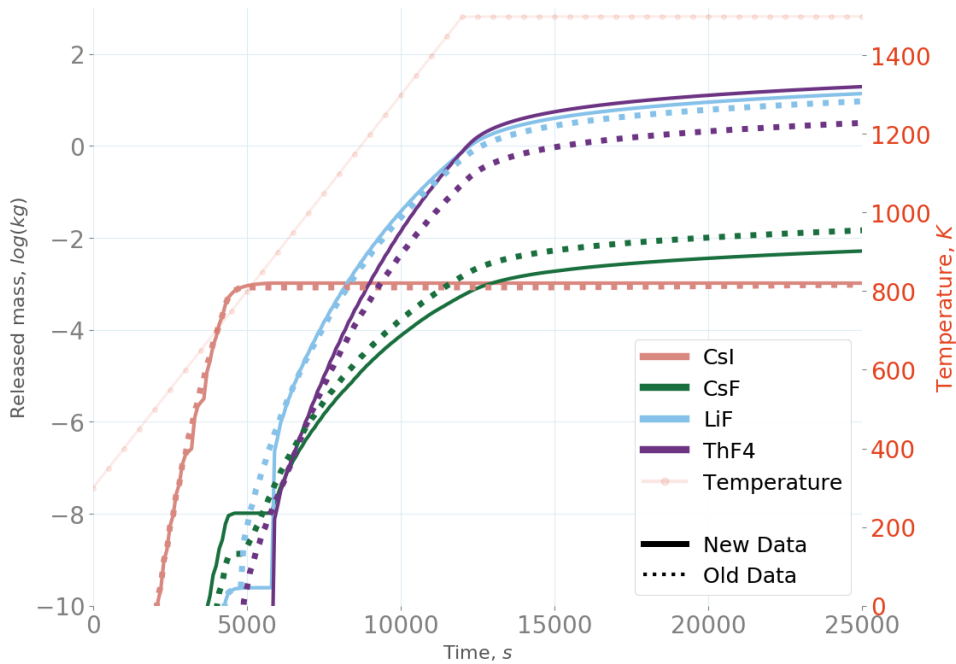
**Table 7:** Representative fuel composition for release analysis.

Species	Content [mol%]
LiF	78.8
ThF <sub>4</sub>	16.9
UF <sub>4</sub>	4.2
Cs	9.8E-3
I	5.8E-5

Table 7 outlines the composition used as the initial fuel. Note that it is quite different from what is generally used in other applications within this thesis, however being able to directly compare to the previously published results is very valuable. For the sake of the cGEMS runs, the time interval for which MELCOR is run between updating the vapor pressures using GEMS is set to 200 seconds.

### 6.6.1 Results

The only result presented here will be the integral released mass as a function of time. Being a function of time, it is also intrinsically a function of temperature.



**Figure 16:** Cumulative release of various substances as a function of time.

First of all, it is important to note that the change in database parameters only clearly affects some of the species here. Most notably, CsI is essentially unaffected; This is due to the fact that no interaction or database parameters that directly involve CsI at all. However, the same can be said for its fluoride counterpart, which does experience a considerable change in release. Upon investigating which species interact strongly with which, it becomes clear that CsI has very little interaction with any of the other components that are part of the system. Without any direct interaction and only very minor indirect interaction, it is no surprise that there is little change in the resulting numbers.

The retention of CsF in the salt has improved with the new dataset, suggesting that the changes made to the main carrier salt LiF-ThF<sub>4</sub> have made the mixing effects stronger. It is difficult to say what change exactly causes this, as the binary system in question essentially received a complete overhaul. The old parameters did not make use of temperature-dependency, so the difference is also a function of temperature. In return for the reduction in release of CsF, a larger amount of carrier salt is volatilized. Ignoring the impact of the other compounds, the binary LiF-ThF<sub>4</sub> diagram presented above has its eutectic point quite near the initial salt mixture. Before the adjustment made in the scope of this thesis, the eutectic point was a lot less well resolved, so the effective melting point of the system has likely been lowered. Just before the ThF<sub>4</sub> graph goes below the bottom cut-off of the plot, it becomes near vertical. This suggests that its melting point would occur just outside of the frame. Both LiF as well as CsF have their melting points at very similar temperatures, as indicated by the kink in their evolution at around 5000 seconds. Interestingly, CsI does not show any melting behaviour; In fact, it is entirely released before it gets the chance to melt. The plateau starting at 5000 seconds is the point at which the Iodine resources of the starting salt are exhausted, and therefore no more CsI exists in the system.

As mentioned above, the changes to the database affect the release of both LiF as well as ThF<sub>4</sub>, however it affects ThF<sub>4</sub> to a much larger extent. This is due to the fact that LiF is the main compound present in the system, starting out at 78.8mol%. Therefore, any given interaction an average Thorium atom experiences is likely to be involving Lithium. Compared to LiF, ThF<sub>4</sub> will experience a proportionately larger effect from changes to the LiF-ThF<sub>4</sub> interaction in a system like this.

## 7 Fuel Cycle Modelling for MSR Systems

### 7.1 Learning and Adjusting EQL0D

EQL0D is an incredibly complex tool, which was in no way made to be accessible to a future user. While there is a small amount of commenting within the code, the overall documentation is incredibly limited and also out of date. In addition, the tool was made to run on a specific cluster at PSI, with the according MATLAB installation as well as the according scripts that are needed to gain and keep connection to the cluster. In order to convert it over to the merlin6 cluster, which all cases in the scope of this thesis were run on, some modifications were necessary. Most of these are very basic operations, such as converting to MATLAB 2019a or removing the ticketing part from the launch script of the program. While the ticketing was previously handled using Kerberos, merlin6 works with a slurm-based queueing system, which needs its own launch file. In order to incorporate this, an additional file is needed, which is submitted to the cluster as an allocation request. Once an allocation of computational resources is granted, this script adds the directory of the run to the environment as well as to the MATLAB path variable. Then, it calls the actual EQL0D launch script.

More detailed information on EQL0D, the functions and directory setup involved as well as a small tutorial on how to run a case is attached in the description. This is not meant as a full replacement of documentation of the code, as it is much too complex to fully go into in a couple of pages. Instead, the section aims to give a stepping stone into working with the code, however additional work will be necessary to fully understand or even change the algorithms.

In order to become familiar with EQL0D, a couple of previously existing inputs were run. Starting out, due to the necessity of transferring the whole setup to a new cluster, the runs were performed using the 'salloc' feature of slurm. This allows the user to have an interactive window with their allocation on the cluster, rather than relying on the queueing system. On top of removing the need for a slurm submission file, it is a lot easier to work on bug- or compatibility-fixes when the running process can be followed using the 'top' command on the Linux command line. While it is good practice to become familiar with necessary file structures, it is also essential to gain an understanding of usual run times as well as the optimal allocation requests. This is especially important considering that any reduction in the requirements, both in terms of time as well as processing power, improves the priority in the merlin queue.

### 7.2 SAMOSAFER Simulation Benchmark

#### 7.2.1 General Purpose

In the scope of the international project SAMOSAFER, a group of people from a multitude of different institutes (such as POLIMI or SUBATECH) work together on the development on their respective tools. These tools' purpose is to simulate the evolution as well as equilibrium composition of the different materials within a molten salt reactor during its use. Because this goal can be approached from a multitude of angles, the functionality and mathematical foundation of the tools developed by the individual research centres can differ. This makes a benchmarking process necessary, which aims to identify differences in results when the same setup is used as well as hopefully tracing those differences back to an identifiable source.

In the end, there is quite little experimental data to compare against, as only a select few MSRs have ever been run. As such, it is also important to not only identify differences and their source, but have a healthy discussion about what the most accurate computational methods and algorithm choices are.

In many ways, being part of the SAMOSAFER benchmark follows a similar purpose to the creation of the phase diagrams and general work on the database performed for the use of GEMS. Without an accurate database, a chemical-thermodynamic tool becomes unreliable. Similarly, without validation processes, an as of yet unproven code cannot be assumed to produce correct results. Therefore, on top of getting the chance to contribute to the SAMOSAFER project, benchmarking EQL0D will be a great long-term asset to the quality of any results it produces.

### 7.2.2 Procedure

In order to be able to pinpoint the reason for any differences, the initial stage of the benchmarking process needs to be as basic as possible, with a ramp-up of complexity with iterations. Because of the huge amount of both variables as well as resulting values, it is already very difficult to find and understand deviations in the most basic case. Part of the reason for this difficulty is the fact that the programs being compared have very different options when it comes to the setting up of the case. On top of that, not all programs use the exact same algorithms with the same supporting parameters, which causes even slight deviations in the initial setup to propagate and give rise to considerable differences in the outcome. Even after the calculation has concluded, portraying a resulting composition of over 1600 isotopes in an easily analyzable and comparable way is a challenge in and of itself.

One of the aspects of an MSR simulation that can be reduced to the minimum for the sake of simplicity is the geometry. Rather than the complex, nested cylinder-like structures that make up an MSFR for example, the geometry of the benchmarking process is reduced to a cube of one cubic metre size, which is reflective at all sides. As a consequence, the case loses a lot of materials which complicate the case, as well as reprocessing streams necessary to exchange material with a possible blanket. It also removes the need to account for leakage.

On top of removing materials invalidating some of the reprocessing streams present in fully-fledged cases, one can remove all reprocessing streams for the first step of the benchmark in order to further reduce the complexity. Note that it is not possible to define zero reprocessing streams in EQL0D due to the way the program is coded, however it is possible to add a single stream for an element that is not present in the composition. A good choice for this is for example Roentgenium, which is an element of very high atomic number that has no generation mechanism within the EQL0D database.

### 7.2.3 Incremental Benchmarking

After having gone over some of the baseline choices made within the scope of the SAMOSAFER benchmark, the concept of incremental benchmarking will be discussed in the following. The idea of incremental benchmarking as pertaining to its application to this benchmark was put forward by the PSI contingent of the working package, who has taken somewhat of a leadership role in the working package. Not only is a considerable portion of the parameters defining

the cases put forward by the institute, the cases are also usually the first to complete and be submitted for comparison.

The entire concept of incremental benchmarking relies on the reduction of complexity of the system initially. This is very valuable in the case of the modelling of a nuclear reactor, due to the inherent complexity that is present in the basic components such as the nuclear database or the underlying algorithms. While reducing the complexity of the framework that the benchmark exists in is very feasible, the underlying systems still make even the most basic of setups difficult to analyze from a comparison or error tracing point of view.

In some sense, the natural intricacy of the underlying data and systems removes some of the intended ease of entry to benchmarking in the nuclear context, however one could argue that it also makes incremental benchmarking more necessary. Even if it is not possible to reduce the initial step into understanding discrepancies in codes and results to a point where it becomes easy, incremental benchmarking will certainly make the small steps along the way towards a full scale case more palatable.

This approach of subdividing the leap to a full-scale simulation into smaller steps along the way enables the comparison between the codes to be tackled one system at a time. Because the initial results are based exclusively on the base implementation of the algorithms combined with the database, statements can be made associating differences in the results with specific parts of the codes.

Then, adding or complexifying parts of the input as well as supporting structures one by one will allow for the analysis of the changes to the system based on the knowledge of the basic differences between the codes. It may even be worth the effort to go through the additions one by one to gain some insight on their individual impact on the base system, rather than only having incrementing complexity. Often, in cases such as an MSR simulation, routines strongly affect each other and therefore an effect of compounding changes to the system occurs.

Overall, the value and extent of incremental benchmarking will be a trade-off between the complexity of the analysis required and the effort in terms of computations and sheer number of results that need to be compared. In the context of the SAMOSAFER case explored here, it is essentially a necessity in order to even be able to make qualified statements about the discrepancies between algorithms and programs as well as making decisions as to how and why to implement the final product which may experience extended usage when it comes to finally designing a viable reactor setup.

#### **7.2.4 Implementation**

The basic idea of the benchmark is to use a cube of  $1\text{ m}^3$  with reflective boundary conditions as the geometry. This cube is entirely filled with fuel. While there is obviously the option of introducing a blanket to the system, it is omitted in this first cycle of cases. As a baseline, there is no reprocessing added other than the negligible one for the reasons above. As with all other EQL0D runs, the temperature is set to 900 K.

As the starting material for the fuel, a search is performed in order to approximate an equilibrium of Thorium-232 capture to Uranium-233 fission. The sum of the fertile and fissile materials is assumed to make up about 22.5% of the molar cation-composition of the fuel, with Lithium taking over the rest. For the Lithium, an enrichment in Lithium-7 of 99.9% is assumed. Fluorine is added proportionally, using assumed valences of four for both the Thorium as well as Uranium fluorides. A value of  $4.1249\text{ g mol}^{-1}$  is taken for the initial density of the fuel salt.

In the benchmark, the different cases can be separated into two categories: burning and decaying. While there are multiple different cases covering, for example, the burn phase, which all start from the same initial fuel, they differ in complexity. In table 8, the differences are explained. Note, however, that all cases share the same time-stepping, which subdivides 1800 days (five years following the 360-day calendar) into a total of 16 steps, excluding the initial run at zero days. The 1800 days are divided into the following 17 steps:

0 0.0003 0.0007 0.003 0.011 0.045 0.19 0.75 3 10 42 169 225 225 225 450 450

Note that these values are given in days. The heavy skew in allocation of a lot of cycles towards the first couple of days is because of the rapid changes in the material evolution matrix, which can cause large deviations from realistic behaviour due to extrapolation errors.

It is of note that the cases presented in table 8 are run twice, once during burning and once during pure decay. The difference between the cases is mainly notable in the power setting within the Serpent input, as well as the starting material. Because there are multiple different cases for the system without power, it is not reasonable to simply use the output of a burnup case as the input for a decay case. By having a standard input fuel composition for the decay runs, the comparison between different codes in the scope of the SAMOSAFER project becomes significantly easier as it excludes errors in the decay-case results from being consequences of differences in the input fuel composition.

**Table 8:** Cases run in the scope of the SAMOSAFER benchmarking process, denoting the respective changes from the previous step.

Index	Case description
1	The most basic case, following the setup outlined in section 7.2.2.
2	In addition to case 1, soluble removal is added. It runs at a cycle time of 450 days and affects elements was presented in table 18 in the appendix.
3	Rather than treat volatile removal addition separately in the scope of the benchmark, they are added on top of the soluble removal added in step 2. For this benchmark, metallics are also included in the same removal stream, as they are also removed using helium bubbling. Both streams are subject to a cycle time of 30 seconds.

In the initial cases run for familiarization with EQLOD, only very few elements were considered for reprocessing. As a consequence, some elements ended up accumulating heavily, due to being generated without any path for removal. This is especially problematic for cases that run longer than the five years employed for this benchmark. An example of such an isotope is the Sodium-23 used in chloride-based MSR designs, which can absorb a neutron to produce Magnesium-24. Magnesium-24 is a stable isotope, meaning that if it is not included in any of the reprocessing streams out of the fuel, it will accumulate. To combat issues like this, the benchmark includes all non-actinide elements which are not part of the carrier salt in the soluble removal stream, even if they are also considered soluble. Due to the volatile removal rate being order of magnitude faster, it will not have any real impact on the result.

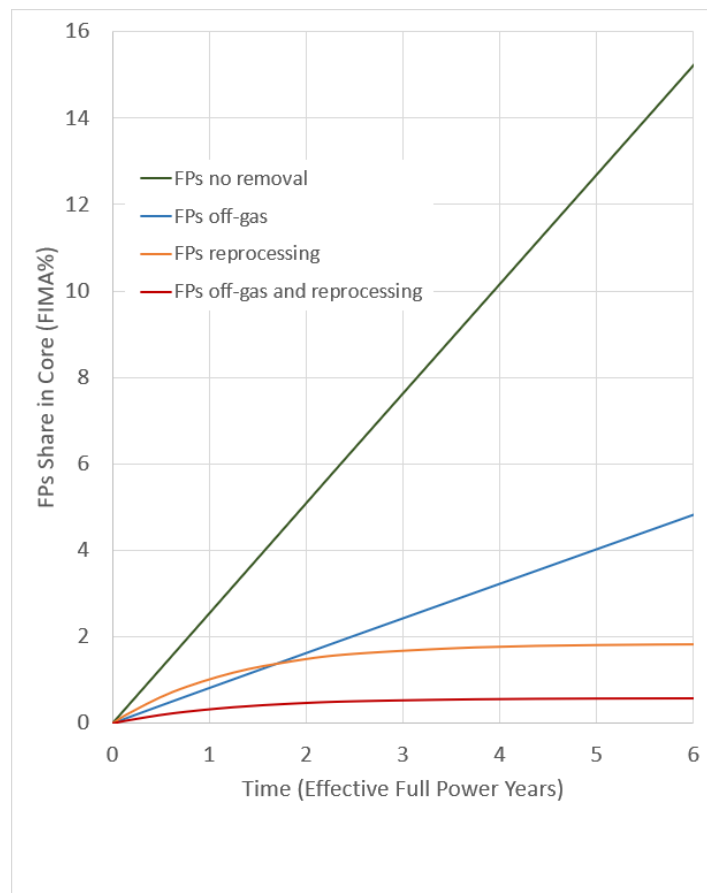


## 7.2.5 Results

In this section, some results of the benchmarking runs of EQL0D will be presented. While the benchmarking as such loses a lot of its meaning without having direct comparisons to the other codes, there are still some good statements and conclusions to be drawn both about EQL0D as well as the system in question.

First and foremost, it is important to remember that the framework most of the benchmarking runs executed and presented as part of this thesis are in is one of extreme simplification. As an example, while some trends and results from analysing fuel evolution can certainly be extended to a real reactor setup, a large section of what influences the fuel is left out of the system. At the same time, however, the entire concept of incremental benchmarking aims to slowly scale up the complexity towards an assumed real system, which automatically increases the value of any previous calculation by being able to see the impact that the addition of certain materials and structures has on the system.

The graphic of figure 17 displays the fission product evolution in the fuel of the benchmark case. In black, the fission product content is displayed without being affected by any stream out of the core. It evolves essentially linearly, as there is no removal path to form an equilibrium with.



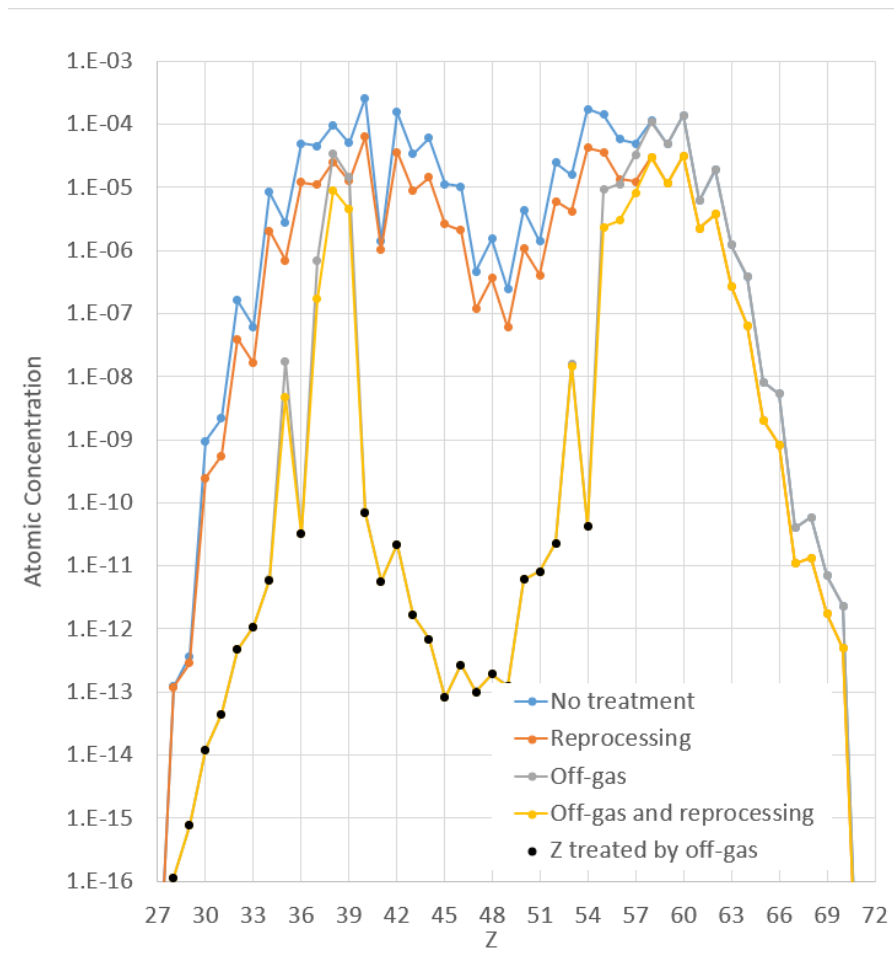
**Figure 17:** This plot shows the evolution of the fission product share in the core over the duration of the benchmark. The fission product share is given in %FIMA.

When the off-gas system is added, the fission product content in the core decreases considerably, however there is little equilibration due to the extremely fast cycle time of the off-gas system. In addition, the fission products to be removed by reprocessing are still accumulating, making any dip in the fission product share much less noticeable.

Comparatively, the reprocessing actually takes out a considerable amount more fission products than the off-gas system. This is, however, not true right at the start of the simulation, due to the much higher cycle time. It takes some time to catch up with the initial generation rate, however the equilibration is quite strong with the increase in fission product share per time becoming almost flat at the end of the simulation. There will still be some minor increase due to the difference in elements covered by the two removal streams.

The last case to cover is the combination of the two removal streams, which behaves essentially as expected. With both streams enabled, the equilibration can take full effect, and the share stays constant after about three years of operation. Obviously, the total share is also much lower than it is for the individual cases.

Rather than looking at the sum of all fission products, a more detailed look at an element-wise distribution will be given in the following. More specifically, the difference in elemental composition between benchmarking runs of different setups will be investigated.



**Figure 18:** In this plot, the elemental distribution of the resulting composition after fuel burning with various case setups is displayed.

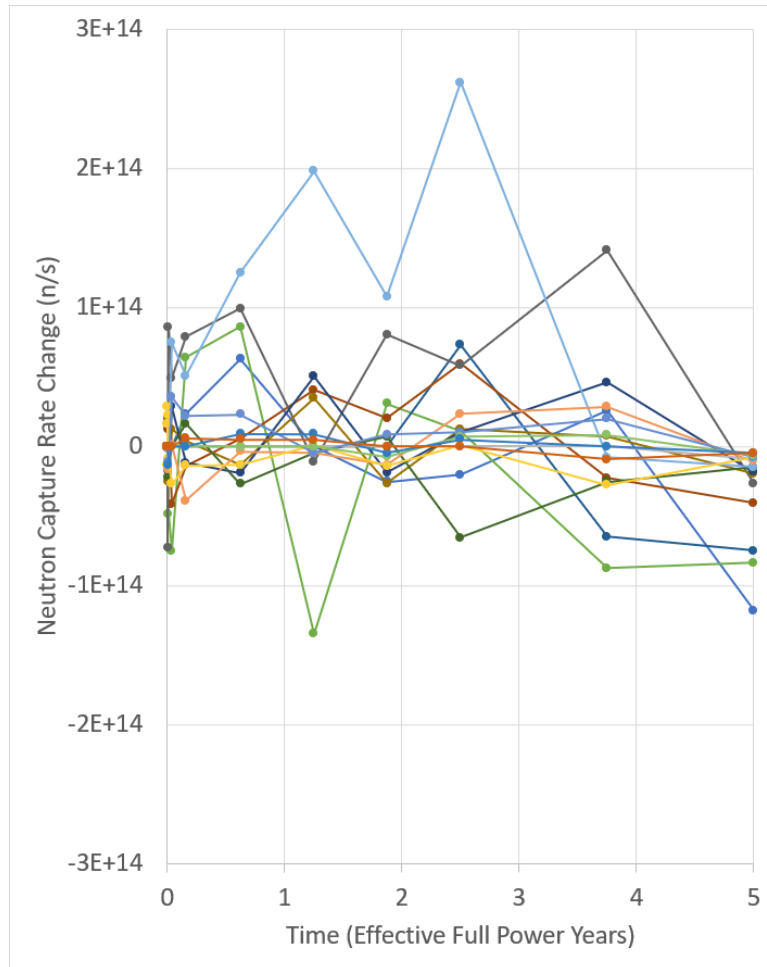
In general, the elemental distribution follows the expected double hump of fission product distribution quite well. There are some elements which are somewhat higher or lower in concentration than expected, however one needs to keep in mind that this is the final composition after burning the fuel, so decays have to be taken into account as well.

The elements affected by reprocessing and the off-gas system respectively can be found in the appendix. The majority of them, however, is also quite well represented on figure 18. While the elements treated by the off-gas system are specifically marked with black dots, reprocessing in the case of the benchmark affects, among other, all elements between the atomic numbers 10 and 88. This includes all elements represented on the graph. Note that this also means that many elements are affected by both; however the impact of reprocessing is mostly negligible in such a case due to the orders of magnitude difference in cycle time. This is especially visible for the elements marked with a black dot, as there is no visible distinction between the grey and yellow lines, representing cases with off-gas but with and without reprocessing respectively.

It is also worth noting that while reprocessing affects all elements on the figure at the same rate, the relative difference between the concentrations of cases with and without reprocessing is not consistent. This is likely another product of the varying stability between the isotopes of different elements. Additionally, however, there are also variations in the impact that reprocessing has on elements. Take Niobium, element 41, as an example. Compared to its surrounding elements, the impact of reprocessing is much weaker. This may suggest that it is regenerated quite quickly after being removed, however it is difficult to make precise statements by simply looking at the final composition.

When the off-gas system is enabled, it is quite clear to see that the affected elements experience a large reduction in their final concentration, which is to be expected. There are, however, also elements affected quite heavily which are not directly removed from the system. This phenomenon is a product of the nuclear state in which fission products are left in after a fission occurs; They are very neutron-rich, which means they will almost always decay into the next element of higher atomic number by  $\beta$ -decay.

Lastly, it is worth shining a little bit of light on one of the underlying issues of EQL0D. In figure 19, the evolution of the neutron capture rate with time is displayed. While no direct identifiers are given for the isotopes, the important point to be made is the general disorder at play here. There are no discernible tendencies in the evolution of the neutron capture rate of any of the isotopes displayed, which affects the result in a variety of ways. Most notably, it makes any effort towards having an analysis of the reactivity extremely difficult, as the multiplication factor is computed directly from reaction rates given by Serpent.

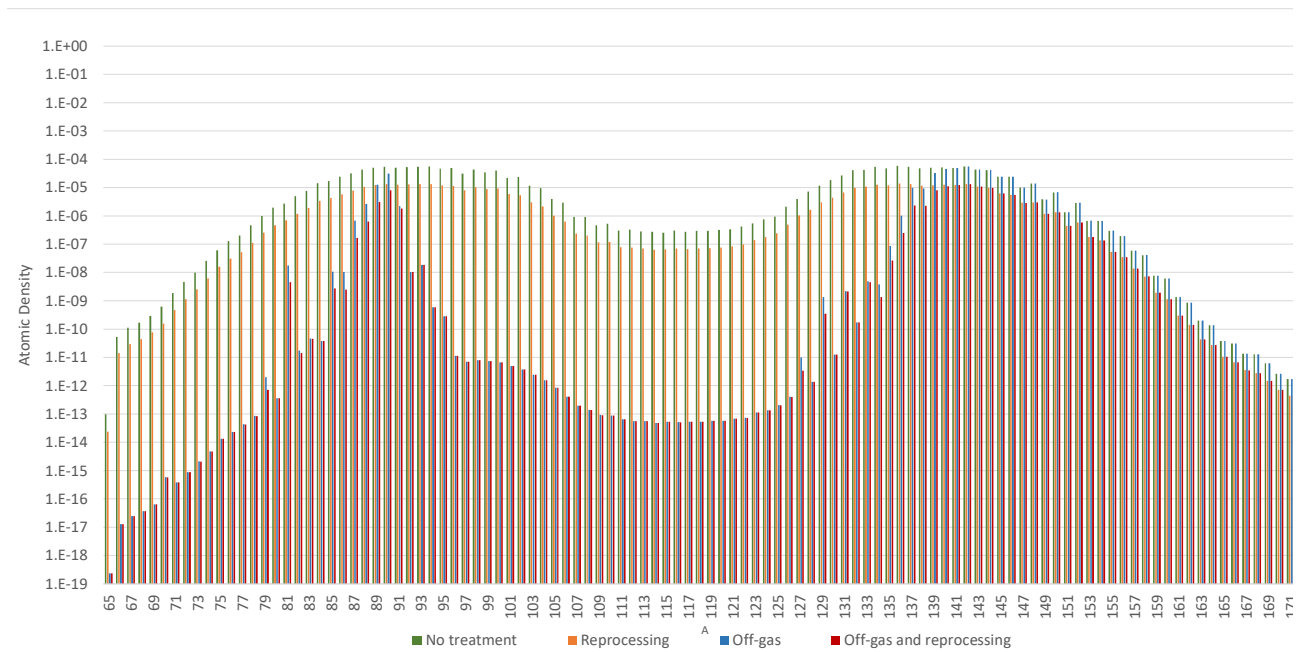


**Figure 19:** Plot of the neutron capture rate change during a run of the benchmarking process.

Even beyond the application to reactivity, having a seemingly random change in reaction rates over time will introduce inconsistencies in the evolution of materials. It is as of yet unclear why exactly this occurs; While there is certainly a somewhat random component to the Monte Carlo nature of EQL0D, the same seed is reused for each calculation that is performed within the scope of the same case. In theory, this should remove a lot of the blame from the randomness of the process due to repeating the same pseudo-random numbers generated.

There may be another explanation coming from the fact that Serpent is only rerun on each cycle, rather than each step. This means that the material can evolve considerably before any reaction rates can be updated. In the scope of this work, much shorter cycles with fewer steps within them are chosen for the start of a case, where the change in reaction rates is assumed to be the greatest. However, this does not exclude at least some impact of such a practice on the irregular evolution of the neutron capture rate in figure 19.

While the above diagrams are all either referring to the elemental make-up or the sum of all isotopes of a certain kind, the full complexity of the cases and results obtained as part of each execution of EQL0D is much greater. As soon as the composition is given based on atomic mass, such as is the case in figure 20, it becomes clear how many variables, and therefore possible differences to other codes, there truly are.



**Figure 20:** Composition of various fuel-burning benchmark results as a function of atomic mass.

Figure 20 shows very nicely the broad effect of having applied the reprocessing to every element, with a few exceptions. Enabling reprocessing affects every presented atomic mass, with no significant deviation from the typical double hump observed without any treatment. The addition of the off-gas system, similarly to the results obtained for the element-wise presentation of figure 18, is much more spotty. Still, the large singular spikes that were commonplace for the specific elements treated by the off-gas have largely disappeared, due to the removal streams of EQL0D being applied to elements rather than isotopes. It is also quite visible that the off-gas system stops having an effect for higher-mass isotopes, as indicated by the closing gap and eventual unification of the off-gas as well as no treatment cases, starting around the 125 mass. Additionally, the off-gas system has a region around atomic mass 87 where its effect is much less pronounced. This is likely a product of the fact that some of the highest yield isotopes fall in this region, combined with the fact that the coverage of the off-gas system for elements which are expected to be produced at such masses is spotty at best.

## 7.3 Generating Representative Salt Compositions

### 7.3.1 Previous Cases

EQL0D has been previously used by its creator on a large variety of cases, most of which aim to recreate a realistic concept put forward in the past. While these cases are representative of the systems they are built to simulate, they are extremely complex cases which lose a lot of similarity between each other. This makes it tough to compare simulations, as the differing results can be associated with many variations in the initial setup of the run. They are also much faster to run, which saves precious time during the thesis.

### 7.3.2 Adjustments for the Homogenized Cases

In order to combat the lack of comparability, the complexity of the inputs was reduced to four MSFR-like cases. To achieve this, a MATLAB script was used which produces Serpent geometry definitions that resemble an MSFR setup in relative proportions while using the core volume as an input. An example of such a geometry can be found in the appendix.

To further reduce the complexity of the system, no blanket was employed. Instead, the space originally designated for a blanket in the MSFR design was replaced with more Hastelloy. This not only removes an additional material; it also simplifies the reprocessing scheme significantly. The reprocessing scheme<sup>[3]</sup> employed in all of these cases is virtually the same, barring the difference in carrier salt. In general, elements are divided into volatiles, metallics and solubles, which are removed at different rates. Both volatiles and metallics assume a 30-second removal period, which is extremely fast. It is a number commonly used and referred to in MSR research, however it is somewhat unclear where and why it originated. This makes it a potential target for future analysis and possibly change. Both volatiles as well as metallics are planned to be treated using a helium bubbling method, where the noble gas is bubbled through the liquid fuel salt in order to collect gaseous fission products and metallic particles.

Because solubles are obviously not affected by this fast process, they are removed separately using a much more complex process. Because of the need for a chemical process due to the mixing of the liquids, there are many stages that need to be performed, including fluorination to remove Uranium, Neptunium and Plutonium followed by varying stages of reductive extraction and anodic oxidation to split the stream of salt into wanted and unwanted components. More specifically, actinides as well as the carrier salt can be recycled to the core for higher burnup and general efficiency.

### 7.3.3 Case Definitions

In this section, four total cases are considered. This includes a Thorium-232 and a Uranium-238 based chloride case, as well as both of these options using fluoride salts. For the chloride cases, NaCl was used as the carrier salt, with LiF being the fluoride option. It is important to appreciate the importance of the neutronic impact of the carrier salt; The isotopic composition here is crucial. One good example of this is the Lithium used in fluoride salts, which naturally consists of Li-6 and Li-7. If the material was used for nuclear purposes in its unenriched form, a considerable amount of neutrons would be absorbed by the lighter isotope to produce the heavier.

In the four cases presented here, due to the very simple carrier salt, only Lithium and Chlorine have these issues. For the purposes of the calculations, Lithium is assumed to be 99.9% enriched, whereas Chlorine is assumed to be entirely in its  $^{37}$  form.

The recycling of some of the salt after it has been treated by the reprocessing plant, as it is mentioned above, is forgone for these simplified cases. Instead, there are somewhat artificial feed streams in place in order to prevent all of the salt from draining out of the core. In terms of fertile material addition to the system, an inlet is coded such that any mass of fission products or actinides that is removed from the core is replaced by Thorium-232. This ensures that the breeding of new fissile material in the core is plenty. As an additional safekeeping mechanism against the loss of mass, a similar feed is added for Lithium-7. This feed is however based on the total mass of the system, and is not expected to contribute much, if at all, to the end result of the computation.

Which elements exactly are removed in the course of helium bubbling or reprocessing respectively can be checked in table 19 in the appendix.

For all four cases, both reactivity as well as redox control are disabled. The temperature is assumed to be 900 K.

**Table 9:** Case differences for the four MSFR-like inputs, with values taken from previous calculations performed in the scope of the creation of EQL0D.<sup>[12]</sup>

System	Fuel Density	Fuel Volume
Th-F	4.5941 g cm <sup>-3</sup>	28 m <sup>3</sup>
Th-Cl	3.32 g cm <sup>-3</sup>	131.7 m <sup>3</sup>
U-F	4.6693 g cm <sup>-3</sup>	36.3 m <sup>3</sup>
U-Cl	3.32 g cm <sup>-3</sup>	41.6 m <sup>3</sup>

Instead of having a specific time frame for the simulation, as is done for the benchmarking runs, the MSFR-like cases are all run to equilibrium. This is comparatively quite computationally expensive, which makes the use of larger time steps basically mandatory. The equilibrium was reached for all cases after 12600 Effective Full Power Days (EFPD), or around 34.5 years.

### 7.3.4 Results

Due to the fact that this is a simulation with the aim of generating a fuel composition to be used as the input for GEMS, it is the elemental rather than isotopic composition that is of interest. In table 10, the 10 highest elements in terms of atomic density are given.

**Table 10:** Base salt composition of the four homogenized cases using only the top 10 elements in order, given as atomic density in atoms per barn centimetre.

Chlorides/Thorium	Chlorides/Uranium	Fluorides/Thorium	Fluorides/Uranium
Cl: 2.21E-02	Cl: 2.15E-02	F: 5.15E-02	F: 4.87E-02
Na: 4.62E-03	Na: 8.88E-03	Li: 2.43E-02	Li: 2.01E-02
Th: 3.77E-03	U: 3.46E-03	Th: 5.51E-03	U: 5.60E-03
Pu: 3.92E-04	Pu: 6.74E-04	U: 1.05E-03	Pu: 1.59E-03
U: 3.49E-04	Am: 1.74E-05	Pu: 2.64E-04	Am: 5.79E-05
Am: 6.50E-05	Zr: 1.57E-05	Zr: 3.90E-05	Cm: 2.88E-05
Np: 2.54E-05	Nd: 1.24E-05	Am: 3.14E-05	Zr: 1.76E-05
Zr: 7.28E-06	Cm: 1.05E-05	Cm: 2.53E-05	Nd: 1.43E-05
Cm: 6.74E-06	Ce: 9.66E-06	Nd: 2.07E-05	Ce: 1.12E-05
Nd: 4.19E-06	Pr: 3.99E-06	Ce: 1.88E-05	Np: 6.65E-06
Ce: 3.69E-06	Np: 3.57E-06	Pa: 1.30E-05	Pr: 4.56E-06

In general, the elements which are prevalent in the respective systems are quite similar between the cases. Obviously, the carrier salt varies between fluoride- and chloride-based cases, however the gist tends to be the same: The highest elements are made up of the carrier salt, the appropriate fertile and fissile species as well as a mixture of prevalent fission products and some minor actinides. There are some additional differences between the systems, however they are mostly in terms of how high an element will place on the list. In the bottommost elements, there are some unique elements which come up, such as Protactinium for the fluoride-based Thorium case. Overall, the outcomes are about what was expected.

## 7.4 Sensitivity Analysis

The rate of removal for elements in an MSR is not a hard design quantity. As such, exploring the costs and benefits of higher and lower removal rates respectively is a necessary task for the final design of any MSR system. As part of the work performed during this thesis, many attempts at a sensitivity analysis of the reactivity of the simplified cases to removal rates were made. In the end, it turns out that EQL0D is not consistent enough to get a reasonable image in terms of the evolution of reactivity with the adjustment of removal rates. Most likely, the stochastic nature of Serpent plays at least some part in this, however a lot of parameters can have a considerable impact on the results. As part of the many attempts that were made in the scope of this study, most of the seemingly most likely offenders in terms of negatively affecting the results have been tested and found not to improve them a whole lot.

Rather than sticking to the idea of investigating the impact on reactivity, an effort is made to understand and properly display the impact of the various removal rates on the composition. In that scope, special focus was put on the fission product distribution as a function of removal rate. For this purpose, the off-gas system usually implemented in most MSR simulations has been split apart to treat gaseous and metallic fission products separately. In the nominal case, both cycle times remain at 30 seconds, however they are altered separately in the scope of the sensitivity analysis itself. Having the same cycle time for both gasses as well as metallic particles is most likely a consequence of having them treated by the same helium bubbling system, rather



than an actual equality of the removal rates. Therefore, them being potentially different, while obviously still in some way dependent on each other, is a likely future development in MSR design. In the scope of this thesis, they were treated as entirely separate, with no scaling of one with a change in the other.

### 7.4.1 Results

Before having a look at the results, it is important to remember that while they are homogenized in terms of removal streams, there are still considerable differences between the four MSFR-like cases. This is true for the fuel volume, the initial fuel material as well as the density. Density is something to pay attention to in particular, as the results presented in the following will be given in terms of atomic density.

Table 9 gives a small overview of some of the foundational parameters that the four cases are based on. One of the main things to note here is the significant difference between fluoride and chloride cases in terms of fuel density. This will implicitly influence the atomic density presented in the figures discussed below, especially figure 21 because of its comparative nature. While the fluoride densities are only marginally different, the chloride salts are assumed to have equal density. While this is definitely inaccurate, there simply aren't good enough numbers when it comes to literature. On top of that, the difference between Thorium- and Uranium-based salts seems to be quite small as suggested by the fluoride systems.

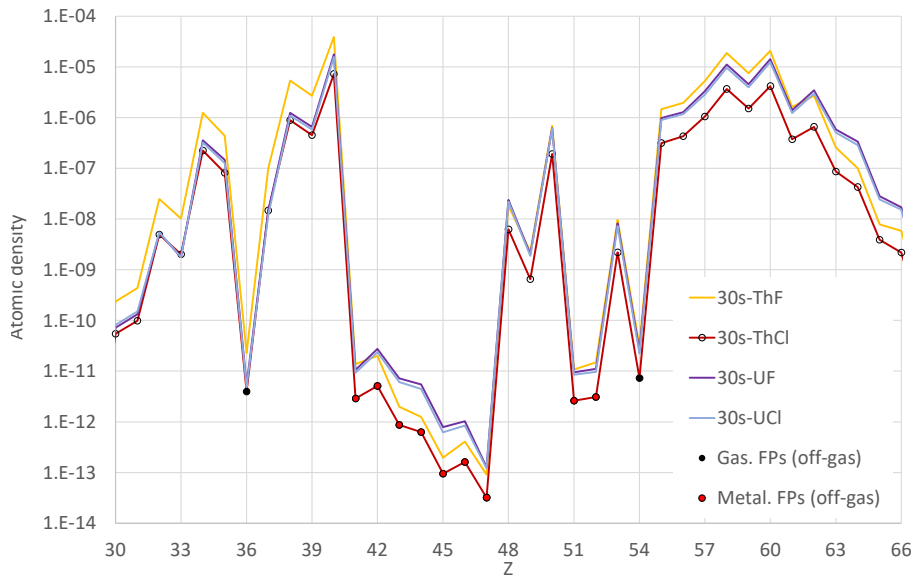
In the following, diagrams summarizing the output composition in terms of fission products as a function of removal rates will be shown and discussed. First, a comparison between the different fuel salt setups is made using the standard cycle times of 30 seconds and 450 days for off-gas and reprocessing respectively. Note that the reprocessing cycle time will not be altered in the course of this analysis.

For the results beyond the initial comparison presented here, the fluoride-based Thorium-232/Uranium-233 case will be used. The figures have however also been produced for the other three homogenized cases and can be found in the appendix.

Comparing the different setup types on figure 21, there are some crucial differences between the cases that are worth noting. In different areas of the plot, i.e. before, during and after the section considered metallic according to the legend displayed, the order of the densities will change. Comparing with table 9 above, note that fluoride salts tend to be much more dense than their chloride counterparts. In the figure, however, there are plenty of areas where this general rule is overshadowed by other effects in play. One such example is the metallic section in the forties of the atomic numbers, where the fluorides using Thorium-232 as fertile material fall to be the second lowest in terms of atomic density. At the same time, the other Thorium-based case also drops in density considerably, suggesting that the difference in fertile material is the cause for the switch-up in terms of order. One possible reason is the shift in fission yield curve due to the transition from Plutonium-239 to Uranium-233 as the main fissile material of the system.  $^{239}\text{Pu}$  has its left peak of the double hump fission yield curve shifted to the right, which could make it overlap more with the region of metallic removal. It is hard to say if and how big of an impact such a shift has, given that fission yield curves are generally given in terms of atomic mass rather than atomic number.

Another fact to consider is that the total power of the system remains the same for all four

cases, however the volume changes considerably. Therefore, a larger system will have a lower specific power. The biggest outlier by far is the Thorium/chlorides case, which requires more than triple the volume of the second largest system. Interestingly, this difference seems to not be represented much in the section left of metallic removal. When checking the rightmost area, which does not contain any elements affected by the off-gas removal, the system drops considerably below the other three. The difference is especially large, considering the scale of the diagram is logarithmic.



**Figure 21:** Plot of the atomic density for fission products on an elemental basis for the four standard MSFR-like cases.

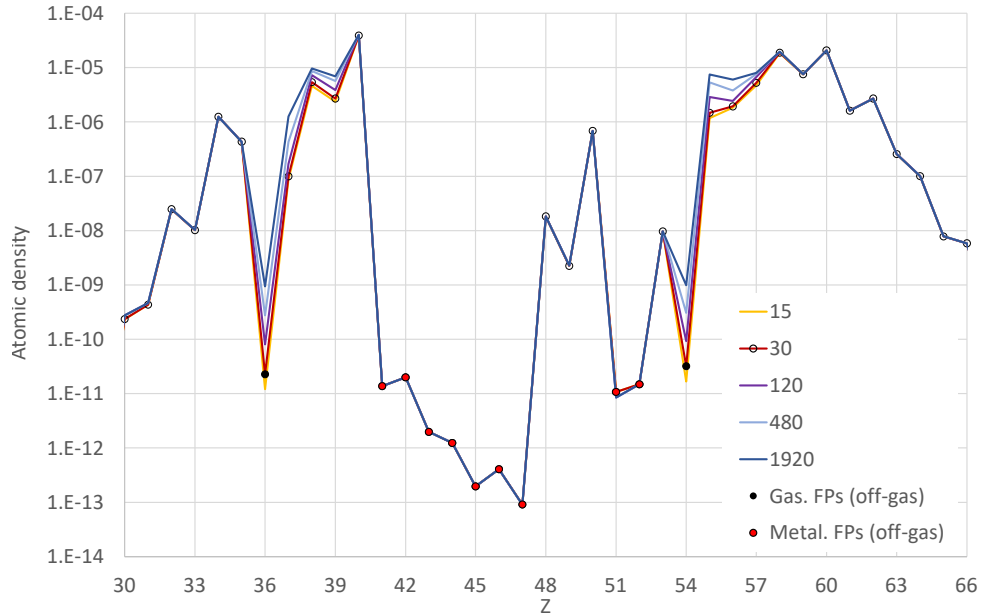
All in all, while there certainly are differences in terms of elemental composition between the four cases, the general shape of the curve remains very similar. Especially the off-gas removal can be clearly seen in the plot, even if one were to remove the denotation from the legend.

As mentioned above, the following diagrams pertain exclusively to the Thorium/fluorides system, as it is the classical MSFR setup.

The first diagram, figure 22 displays the impact of altering the cycle time associated with the volatiles elements of the off-gas system. Notably, there are only two elements which fall within the volatile category in the spectrum of elements displayed in the plots below. They are easily identifiable, namely elements 36 and 54, or Krypton and Xenon.

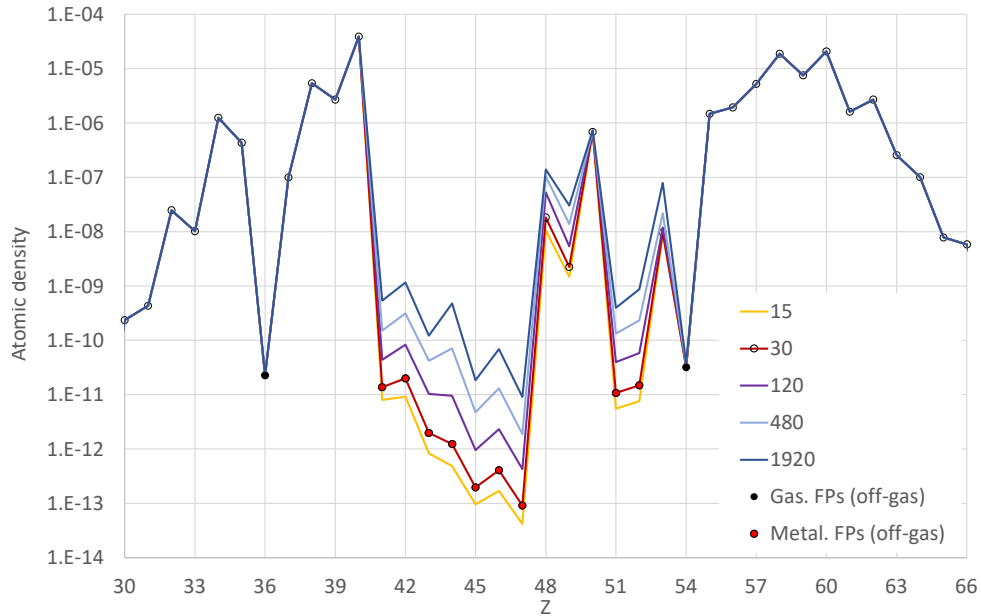
As expected, the two affected elements show a clear decrease in atomic density with a reduction in cycle time. Naturally, the isotopes belonging to the metallic removal category are unaffected. There are, however, elements which are influenced by the rate of volatile removal without being considered volatiles themselves. This is because the isotopes created by fission are very neutron-rich, and therefore tend to decay into elements of higher atomic number by  $\beta$ -decay. If, then, a fission product is removed from the system, it will also reduce the prevalence of elements

which originate as a decay product from that removed element.



**Figure 22:** Figure showing the fission product composition of the fluoride-based Thorium case for various volatile removal cycle times.

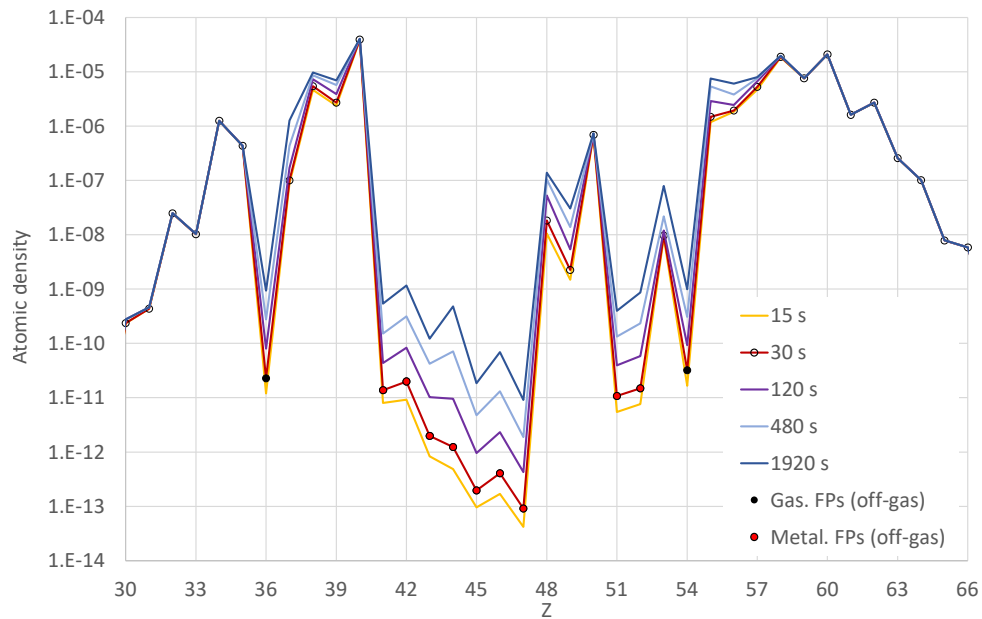
Similarly to the volatiles version, the metallic removal variations in figure 23 cause a propagating effect in those elements directly above them. It is a little less pronounced, as the metallic fission products considered for the removal streams in the scope of this work tend to be immediately next to each other. Most clearly, the impact can be seen for the two elements of atomic numbers 48 and 49, possibly because the effects of the two previous metallic fission products overlap.



**Figure 23:** Figure showing the fission product composition of the fluoride-based Thorium case for various metallic removal cycle times.

The propagation of the depression in the atomic density after a removed element seems to make a relevant impact for approximately the next two elements. This is a little bit less than would appear for the volatiles components above, where an effect can still be observed for the third element down the line. This may be affected by the specific element in question due to different decay chains, however the fact that the dip in density in comparison to the neighbours is a little larger for volatiles may also play a role here.

Lastly, some attention will be paid to the combined case. Here, both the volatile as well as the metallic part of the off-gas system will be altered at the same time. Mostly, the effects of the two categories of removal seem to be additive. Therefore, even though the affected species do not at all cover the entire range of fission products depicted in these diagrams, most of the elements are still exposed to the secondary effects outlined above.



**Figure 24:** Combined plot of metallic and volatile cycle time variation impact for the fluoride-based Thorium system.

In the end, decisions for the design of an MSR can be simplified using sensitivity analyses such as this one. For elements which are problematic for a reactor, such as an element which forms an insoluble halide or is an issue for proliferation, the content to be expected in the reactor at any given time can be calculated and scaled using a tool such as EQL0D.

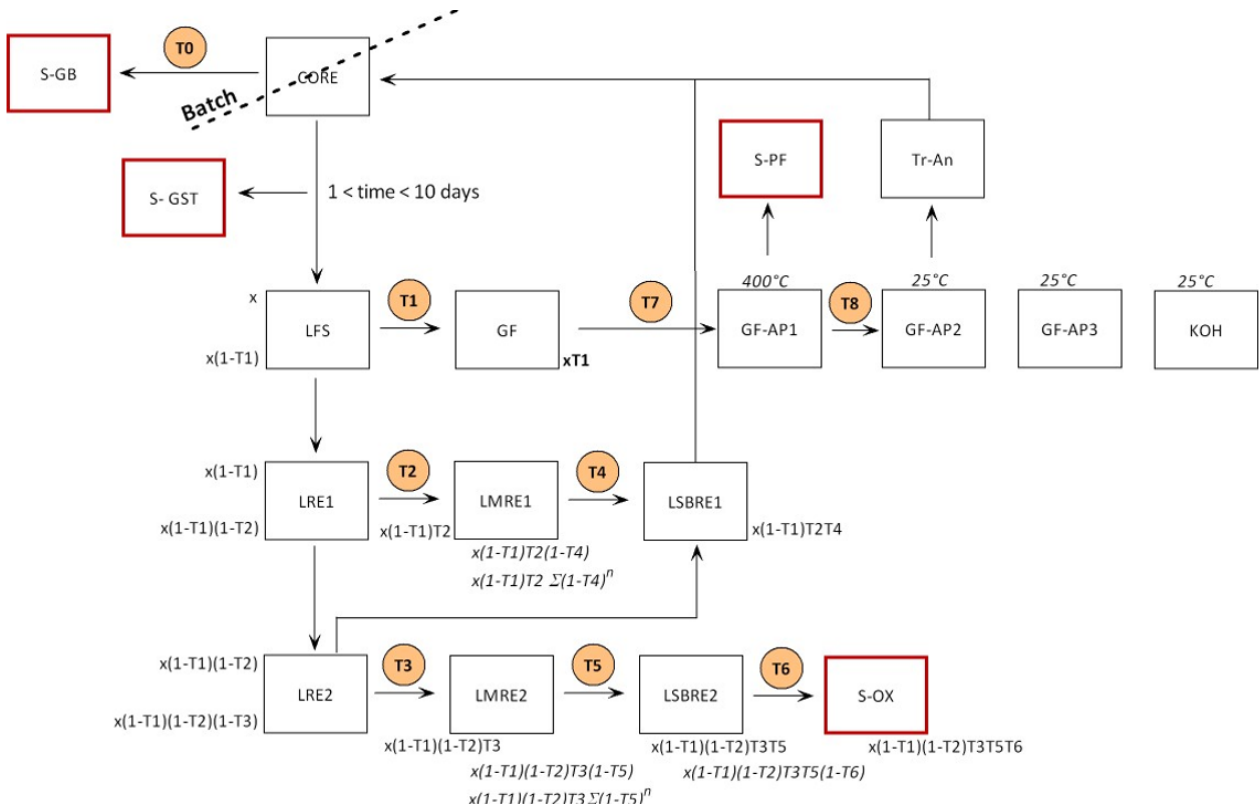
## 7.5 Full-Scale MSFR Simulation

This is the most complete as well as complex case run in the scope of this thesis. In addition to using a full MSFR geometry with a blanket, the entire reprocessing scheme is implemented in MATLAB. Note that the input file is a leftover from the creator of EQL0D,<sup>[12]</sup> however the results were never analysed or published. To add to the case, multiple different versions were created and run in order to investigate an issue, which will be discussed below.

On top of being an interesting case as is, this complex case is a good opportunity to practice slightly different aspects of setting up an EQL0D case compared to what is gained from the benchmark cases. While the benchmark cases are, as of yet, very basic in nature, the full-scale setup involves a lot of input parameters which are not used elsewhere. This includes an extensive reprocessing system with external storage locations, the use of removal coefficients applied to the removal streams from the core as well as automatic control systems. The complex case presented here is also in many ways likely to be similar to the final stage on the benchmarking process.

Note that with the complexity introduced in the system, the number of output parameters scales up incredibly strongly. Each different material prints its composition at completion of each cycle, which enables the tracking of materials and where they end up in terms of storage.

This also allows for a prediction of where the radiotoxicity and decay heat will be allocated.



**Figure 25:** Shown is a diagram of the reprocessing setup as defined within the scope of the SAMOFAR project.<sup>[33]</sup>

In the diagram of figure 25, the reprocessing streams as defined as part of the SAMOFAR project are displayed, where each box outlined in red is a storage location and every other box, other than the core, is a reprocessing treatment location. This means that there are five total final destinations of where material can end up after leaving the core, as a part will be recycled to the active fuel.

The first option for removal of some species is helium bubbling, which is displayed in figure 25 in the top left. Most volatile species as well as some metallics will partition to the helium bubbles, and immediately go to the gas bubbling storage.

The other stream out of the core is much slower than the helium bubbling, and takes actual molten salt out of the core. After the molten salt is taken out, it is given some cooling time, during which additional gaseous fission products may escape. These are stored in the S-GST location. After cooling, the salt is transferred to LFS, which is a fluorination step in order to extract, among a lot of other elements, uranium and neptunium. The fluorinated elements are subsequently transferred to multiple reductive extractions, where actinides and the carrier salt are removed from the fluorinated stream and returned to the core. Fission products contained in these streams are separated from the useful components and discarded. Discarded material is stored in S-OX, standing for oxidized.

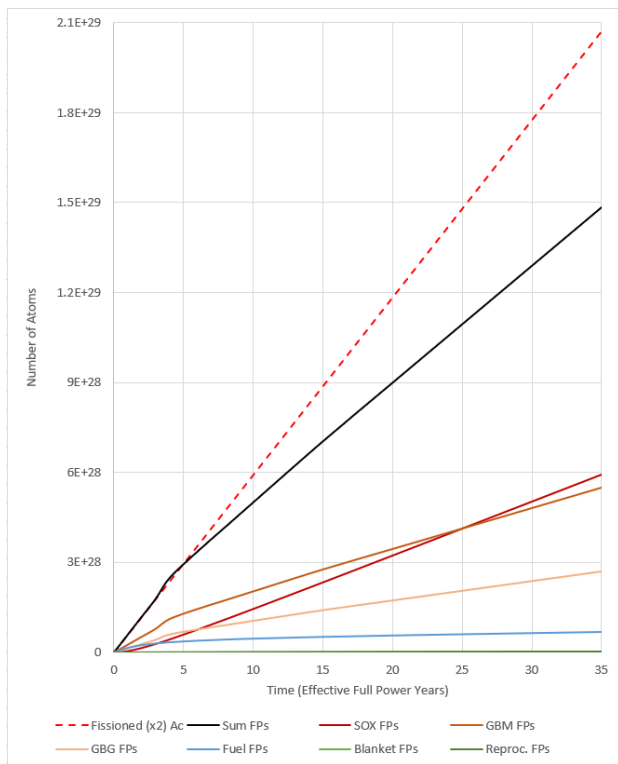
Those elements left over after the separation by fluorination are passed through a series of filter to remove fission products. Any actinides that were missed in the fluorination and passed through all of the filters are returned to the core. The storage unit here is S-PF.

### 7.5.1 Simulation Setup

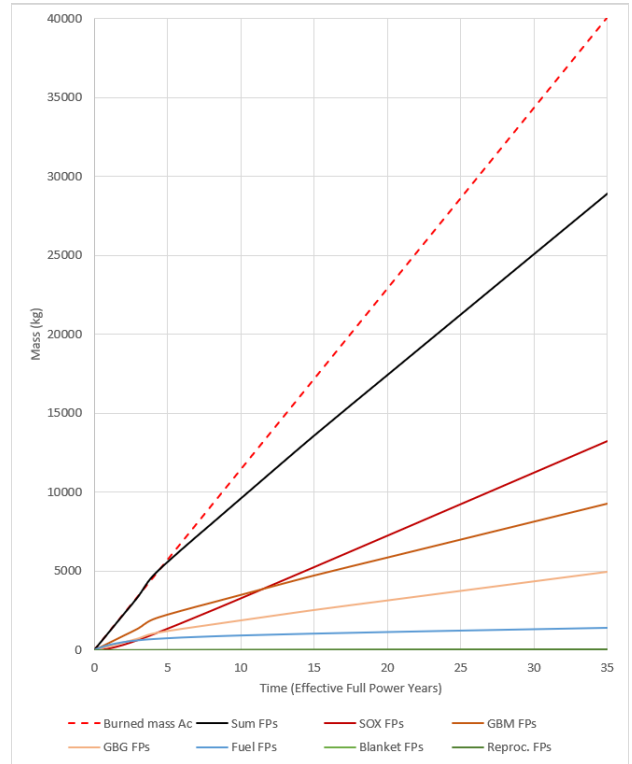
In this section, the application of EQL0D to the complex case is discussed. There are macroscopic options which can be enabled in EQL0D, such as reactivity control. For this case, reactivity control is enabled, whereas redox control is not. Both of these options are described more extensively in section 5.4. In addition, there are reprocessing streams outside of what is described in the general case outline above. These streams are meant to prevent the fuel as well as blanket material from diverging as the case simulates a considerable time-frame. To do this, a Thorium-232 control is put in place, which checks at each step whether the actinide mass in the system has changed. If it has, the changed mass is compensated for by adjusting the Thorium-232 amount present in the affected material. Note that this control element is put in place for both the fuel as well as the blanket. A temperature of 900 K is assumed.

### 7.5.2 Results

In the following, some results of the complex case will be discussed. This will focus mostly on the evolution of elemental contents of the salts as a function of operation time. Note that the contents are presented for all possible locations, which includes the storage units containing the discarded streams from the reprocessing systems.



**Figure 26:** This figure shows the evolution of fission product atoms over 35 years of operation in the reactor. In addition to the calculated values, the dashed line shows the expected total, based on number of atoms fissioned.



**Figure 27:** Similarly to figure 26, this figure shows the evolution of the fission products, however in units of mass.

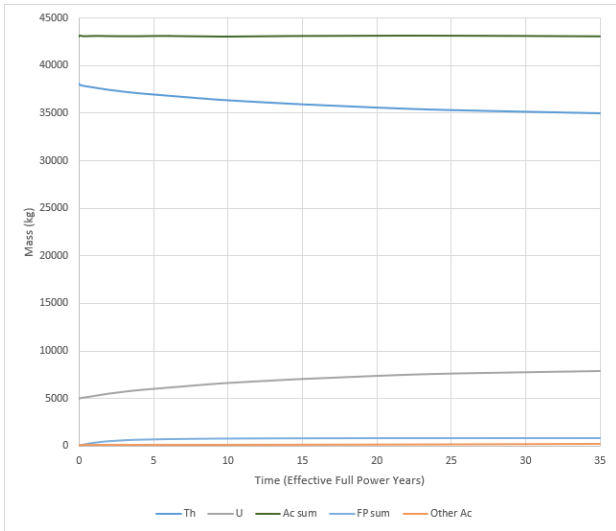
Figures 26 and 27 show quite clearly the main issue that occurs when running the system: After about four years, the sum of all fission products in the reactor diminishes considerably compared to the amount that would be expected according to the amount of actinides fissioned. The figures suggest that the main reason for such a discrepancy originates from the gas bubbling storages for metallic (GBM) and gaseous (GBG) materials, as they exhibit a kink in their evolution at exactly the same time.

While the problematic location is easily identified, the reason for this issue is much more difficult to determine. As presented in figure 25, there is no intended path for material to leave the gas bubbling storage. Even though the isotopes can no longer leave, they are still processed in the scope of material evolution on every step, meaning that any decays do still occur. This makes it possible for the decay as such to be the reason for the loss of material; If an isotopes decays into another which is not tracked by EQL0D, this material may be lost. Even though this may describe one option of how material could in theory be lost, it does not address the fact that this phenomenon only starts after about four years, and seems to remain at a constant discrepancy after that.

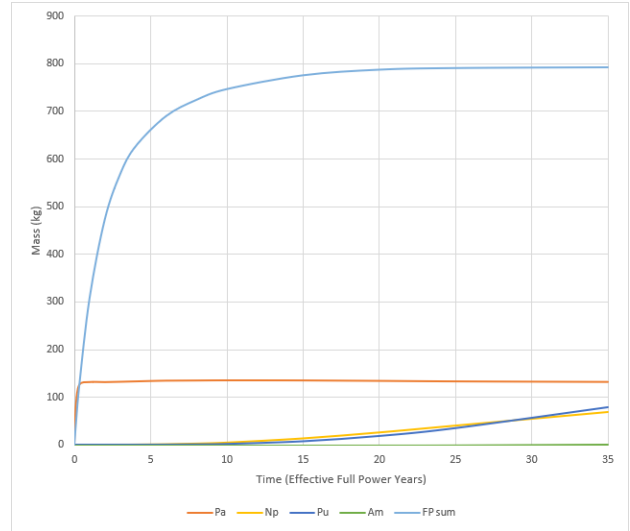
Another possible idea would be that something goes wrong at a specific step of the simulation, however even after rerunning the entire case with altered step sizes, the issue arises after the same amount of time. Still, there is always one step where a lot of the isotopes present in the gas bubbling storage are set to zero, from which point forward the slope of the stored fission products is incorrect. For the case presented here, the associated time of the error is 1460 days. The error not being based on an overall lack in the amount of material, but rather being limited by the complete vanishing of some of the isotopes suggests that the underlying issue is not one caused by a design error of the code. It may be that there is some numeric issue which make some of the isotopes go into negatives in terms of composition, which is caught by the code and automatically set to zero. In the end, it is as of yet unknown how this error occurs as well as why it only affects two materials, which are very similar.

In addition to looking at the fission product evolution, the evolution of the actinides is presented in a separate graphic.





**Figure 28:** A plot describing the evolution of various elements or groups of elements over 35 years of operation.

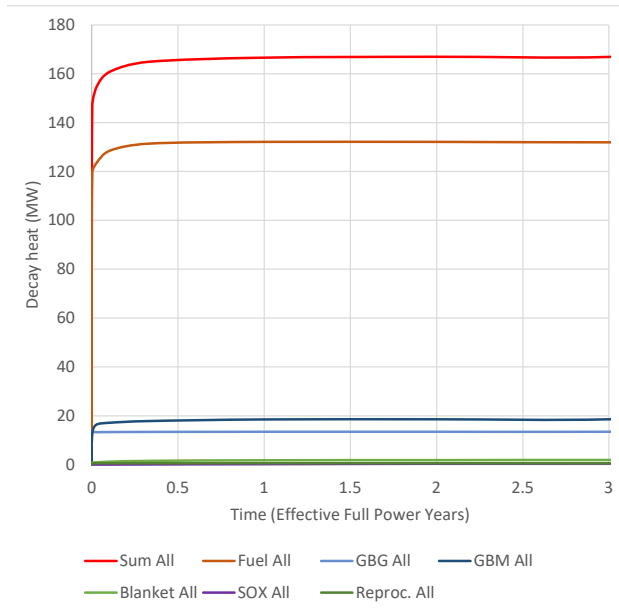


**Figure 29:** This figure is complementary to figure 28 in that it shows a lot of the actinides separately which are lumped under the tag "other actinides".

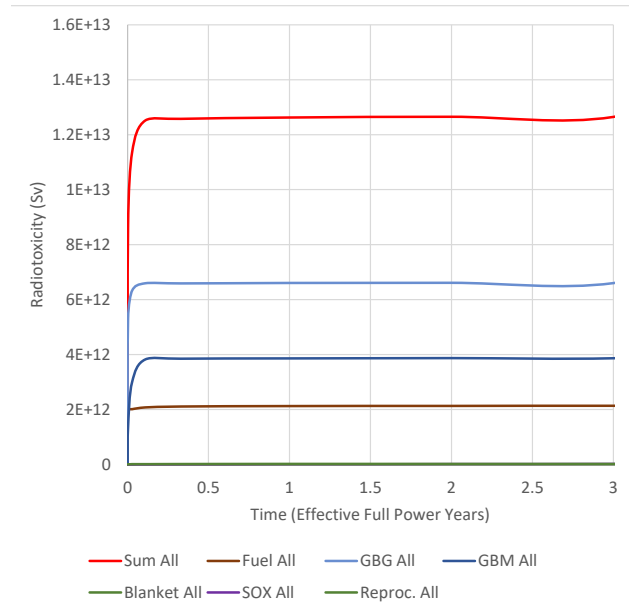
The first thing to note about figure 28 is the extremely straight line at the top, indicating the sum of all actinides (dark green). It would not behave like this naturally, however the implementation of Thorium control, as introduced in section 7.5.1 above, entirely eliminates any variation. When looking at the line for Thorium (blue) just below, note that it quite precisely mirrors the evolution of Uranium (grey), therefore indicating that the Thorium control reprocessing stream is working as intended.

Rather than the broad strokes of information in terms of composition that can be gained from figure 28, figure 29 deals with more individual isotopes. The first thing to notice here is the very fast equilibration of Protactinium, which is a good indication that the system is operating as an iso-breeder. Protactinium is relevant to a Thorium-based MSR, in that its 233 isotope is the predecessor to Uranium-233, which is obviously the fissile isotope used here.

As mentioned in the introduction to the complex case above, the division of materials into multiple storage locations according to the reprocessing scheme defines where the majority of decay heat and radiotoxicity end up.



**Figure 30:** This plot displays the evolution of decay heat for all isotopes, divided between the different locations.



**Figure 31:** This plot is a version of figure 30 treating radiotoxicity instead.

Clearly, figure 30 shows that the principal component of the decay heat remains in the fuel. This is pretty much expected, as it is the main location for fission to occur. Every other material, except for the blanket, does not experience a neutron flux, which means the only influx in terms of decay heat is from the reprocessing of decaying material. Notably, the blanket itself emits virtually no decay heat.

From the plot, the only other slightly stronger cases of decay heat happen in the two gas bubbling storage locations. Due to the fact that they are storage locations, however, the produced heat reduces significantly.

The picture looks quite different for the radiotoxicity, where the fuel loses its dominant position in terms of contribution. Instead, the gas bubbling storage locations take over, with the gaseous elements exceeding the metallic ones. Looking at the element-wise data obtained from EQL0D, the radiotoxicity for fluoride/Thorium cases originates mainly from the Zinc through Europium region of the periodic table, with two rather distinct peaks located around approximately around Krypton and Xenon. Since these values are absolute, the amount of substance present also contributes to the overall radiotoxicity assigned to an element. Clearly, the two peaks represent the typical fission yield curve, however somewhat bumpy due to being based on elements rather than atomic mass.

There is one element which is consistently highest on the charts of absolute radiotoxicity, which is Tin. Tin is not especially famous for its high radiotoxicity, however it does have a large amount of radioactive isotopes with extremely short half-lives. Remember that the fuel is analysed at the exact time it stops being irradiated. Should some marginal decay time be added, Tin would likely drop down the chart quickly.

Note that the above figures 30 and 31 cut off after only three years. This is due to the anomaly described above, which has a large impact on both the decay heat as well as the radiotoxicity of the system. After cutting out the clearly faulty sections, it seems that both the radiotoxicity as well as decay heat total equilibrates quite quickly, and remains nicely constant thereafter.

## 8 Summary and Outlook

First and foremost, it is necessary to mention that there is a considerable part of this thesis which was omitted due to time constraints. In this section, some previous sections will be gone through one by one and any notable conclusions as well as possible future development or completion will be discussed.

### 8.1 Data Collection

In terms of data collection, a considerable part of the issue comes with respect to the availability of the data. As is already mentioned within the main text of the thesis, research relating to MSRs often prefers fluorides over chlorides due to them being the seemingly easier version of the concept to realize. As a consequence, nuclear-relevant fluorides are generally more well-characterized and investigated. This fact also extends to more involved data, such as interaction parameters for mixtures or general information about possible designs for reactors.

It is also worth noting that in a lot of cases, alternate uses of a compound can massively influence the availability of data. Nuclear research seems to make up a comparatively small section of material testing, which means that strong use cases outside of a nuclear reactor can significantly increase the likelihood of solid foundational data in literature. This realization is especially solidified in the context of chemical analysis of burned fuel, as a considerable portion will have been converted to minor actinides. With respect to the components of starting fuel, such as Uranium or Plutonium, the later actinides have even bigger issues with data collection. While compounds such as  $\text{UCl}_3$  and  $\text{PuCl}_3$  already have a lot of issues, due to the compounding disadvantages of being almost exclusively of nuclear interest as well as being chlorides, halides of Curium or Americium are often either very difficult or impossible to find. While Americium compounds have been used in the scope of this work, most species are limited to the solid phase. Curium has not a single liquid phase species in the HERACLES database other than the pure element, and there is very little information even regarding the existence of possible fluorides or chlorides. This fact quite significantly hampers the chemical-thermodynamic simulation of equilibrium fuel, as Curium is in the top ten most prevalent elements of all four of the MSFR-like cases discussed in section 6.5. While this is most likely a direct consequence of the choices made in setting up the simulation, the fact that Curium is likely of considerable importance for a burned MSR salt remains.

In general, it is safe to say that any thermodynamic database for MSR systems is far from finished. On top of lacking a lot of base data for certain species, there is always the additional aspect of interaction parameters. While a decent amount of improvement was certainly made in the course of this work, there are simply so many elements in the core of a nuclear reactor that interaction parameters are a very difficult topic to get right. As will be elaborated on in the appropriate conclusion section, it is quite rare to be able to simply take and use literature values of interaction coefficients, making the whole process extremely work intensive. It is unlikely for a database to ever be able to cover all aspects of a chemical-thermodynamic simulation, which makes the entire aspect of tailoring the database to the problem at hand pretty much a necessity.

## 8.2 Phase Diagrams

The phase diagrams produced in the scope of this work generally fit their respective literature values quite well, however there are many more possible diagrams to be made from the main components of the fluorides' and chlorides' respective starting fuel. Making these phase diagrams is a valuable tool in terms of data validation, while at the same time requiring considerable time and effort. As such, they can be a big time sink due to the time-limited nature of a master's thesis.

While there were many more diagrams attempted than have ended up making it into the thesis, a lot of them suffer from similar problems to the LiF-ThF<sub>4</sub> system above. More than any other phase diagram included here, this diagram took an extreme amount of time and work to get it from essentially a random assortment of phases to something quite closely resembling its literature counterpart. While generalizing the difficulty of this system to the task as such is incorrect, it does apply relatively well to a broad spectrum of fluoride-based binary systems.

One of the crucial difficulties of working with GEMS when it comes to phase diagrams is the volatility of the results with the project parameters. There are a variety of parameters pertaining to the numerical side of the simulation, rather than the thermodynamic values of the database. Especially more complex systems with some instability issues are affected hugely by changes in the project settings, which makes it almost a full second set of parameters to be optimized at the same time. On top of the added difficulty that the mere number of parameters provides, it is extremely difficult to predict the behaviour of the system based on changes to, for example, the smoothing parameter. Other than getting a little bit of a feeling for the setup after hundreds of iterations, there is not much one is able to do to circumvent the effort of simple trial and error.

A specific optimization parameter that can be quite intriguing is the error tolerance. Generally, the more unstable a diagram is, the less stringent the requirements on the error tolerance can be. However, the closer to the desired system parameters one gets, the tighter the tolerance will need to be in order to be able to resolve phase boundaries neatly. This is especially tricky for unstable diagrams with a large amount of different phases, such as the LiF-ThF<sub>4</sub> system presented above, as it is difficult to tell which of the optimization settings to adjust at any given time.

In addition to what is outlined above, ternary diagrams come with the problem of being unable to identify whether or not the current iteration is in line with literature or not. Because GEMS is relatively limited in its application, it is extremely difficult to make a diagram which would work in the same way a literature diagram tends to. Technically, it may be possible to develop a script that can scan through a ternary composition as well as a temperature range at the same time. This comes with a number of issues, not least of which is the hard limitation of a GEMS process to be limited to 9999 simulations per run of a process. To include a temperature range at each possible composition would mean a huge sacrifice in the possible step sizes, intrinsically removing some values for more complex systems.

Should time be committed to developing more ternary diagrams in the future, the option of adjusting the stability of individual compounds in the thermodynamic database itself may be considered. It is important to keep in mind, however, that a lot of liquid phase interaction parameters as well as some base values such as formation enthalpy or standard entropy have been optimized for their respective binary systems. Changing them again to adapt to a different diagram with different stability criteria from a numerical point of view can have an unpredictable impact on previously decent and accurate outputs.

### 8.3 Speciation of an Equilibrium Composition

A large part of the previously available HERACLES database is dedicated to elements and compounds thereof that are commonplace for initial fuel or severe accident analyses. The entire concept of having a 'base salt' goes somewhat against the previous database priorities, as elements which are neither volatile and radiotoxic nor part of the starting fuel become relevant to the simulation. Similarly to a lot of other tasks pertaining to chemical-thermodynamic simulations with GEMS, every effort on the front of scripting or setting up viable cases within the program are accompanied by even greater necessary efforts with regards to database extension. As touched on previously in the appropriate sections, some elements and their compounds are extremely hard to find in literature. There are ways of getting around such limitations, but their application vastly exceeds the scope of this thesis both in terms of complexity and necessary time commitment. Therefore, the idea of having a representative salt composition upon which to build an analysis of more scarce elements' behaviour can become a struggle.

On top of the issue of entire elements being difficult to obtain information for, there are plenty of less represented valencies of halides of more common elements, which can be an issue for simulations addressing the effects of redox conditions. Should an element only have one option for a given halogen anion, the side-effects can be very influential on the overall composition. Without the ability to transition to a lower halide as a response to lowering the halogen content of a system, a metallic phase can form, which is not in line with real expectations. In addition, the change in anion availability in the entire system as a response to a phase- or compound change of any element as a consequence of, for example, temperature can be entirely unrealistic.

### 8.4 Learning EQL0D and Early Cases

EQL0D is incredibly difficult to get an initial understanding of. Part of this is the fact that there is little to no commenting in the code or documentation, however it doesn't help in any way that there is no clear final version of the code. On top of that there is the problem of versioning, in that it was written for an old MATLAB version as well as made to be compatible with a quite out-of-date cluster. A lot of work was necessary to get the whole routine to run at all, which is an issue that was exacerbated by the lack of access to information about the workings of the code without the ability to run it.

After the initial hump of making the code run all the way through without throwing errors, the leftover inputs as well as supporting files from previous users was invaluable as a source of practice. Over the course of many runs of multiple different types of inputs, one of the crucial takeaways was some understanding of what kind of allocation benefited runtimes the most. It was discovered quite early on that spreading the computational resources across multiple

nodes is rather inefficient, which suggests that the 44 cores on one node of merlin6 is a good upper limit for requests. This becomes a bit of an issue, however, when very complex cases are considered. These will struggle to finish within the seven day limit for computations, a problem that could be addressed with the introduction of a checkpoint system. Such a large change to the code was sadly not feasible in the scope of this thesis.

In order not to lose access to information on how EQL0D works and how to get into it, a tutorial of sorts was created. It can be found in the appendix. Note that it is not exactly made to replace regular documentation, but rather make it possible for people to use the routine without going through a multiple-month learning period. This should be possible, as the results obtained from EQL0D are very well formatted and understandable.

There are a couple of subroutines and options which are planned to be introduced to EQL0D in the future. Currently, the highest priority is most likely the conversion of the debugging mode to a version of EQL0D that can run all the way through a simulation without ever calling Serpent. This may be valuable for the sensitivity analysis and will be discussed in more detail in the appropriate section below. Also, the above-mentioned checkpointing of the code may be a worthwhile goal, however it is likely a larger scale project.

## 8.5 SAMOSAFER Benchmark

Because of the time-limited nature of this thesis as well as the reliance on comparing results with other institutes, the benchmark is not yet finished. This is simply a matter of time, as the runs are relatively easy, at least for the moment. Once the complexity ramps up due to the incremental benchmarking, the issue with the complex case has hopefully been solved.

The results of the benchmark are mostly comparative in nature, and the first few comparisons with the other participants have looked alright. EQL0D is generally performing decently, however there is always a small difference between programs. Part of this may be traced back to the different nuclear databases in use, however the entire goal of such a benchmark is to be able to hone in on the optimal, and ideally correct, setup for codes to use in order to design future MSRs. As such, some changes to EQL0D may be necessary as a consequence of the results of the benchmark. This is, however, very much dependent on what future iterations of the benchmark will show, and can virtually not be predicted.

## 8.6 MSFR-Like EQL0D Cases and Sensitivity Analysis

The MSFR-like cases were created as a way to compare cases using different initial salts, which required a lot of homogenization in terms of geometry as well as reprocessing streams. As a consequence of the massive reduction in complexity with respect to simulations aiming to represent a realistic design, the cases run very quickly. It seems that the runtime scales especially highly with the number of materials to be tracked and computed. The much faster completion is part of why they were chosen as a baseline for any attempted sensitivity analysis, as such a goal would require a large number of runs. Cases run using the MSFR-like setup can be completed in just under an hour, assuming the usual 44 core allocation.

Overall, the MSFR-like cases were some of the easiest to complete, especially once the initial fuel composition was established. This applies both in terms of difficulty to set up the case, as

well as time commitment for actually completing the computations.

On a complete opposite note to the baseline cases, the sensitivity analysis was the biggest issue among any task performed for this thesis. A very large number of cases was run with many different adjustments to the input, however with little to no success in terms of the sensitivity of reactivity to changes in reprocessing cycle times.

Running just the normal case for multiple different cycles times yielded seemingly nonsensical results. This is the case for all four systems as well as including all modifications that were tried in the scope of this work.

One example of such a failed case can be found in the description, where a fluoride system based on  $^{238}\text{U}$  as the fertile material is described. For the run presented, the algorithm keeping the total mass of the system constant using Lithium was disabled, as an option to possibly influence the consistency of the results. This, and every other change made as part of the effort to fix the reactivity, does have an effect on the results. Even though changes are there, there is no improvement to the tendencies of the reactivity.

Considering the demonstrated randomness of some of the foundational parameters of EQL0D, such as the neutron capture rate displayed in figure 19, another possible option is to try and fix them in some fashion. This would allow for the entire effect on reactivity to originate from the removal rate of impactful isotopes. one option to do this is to implement the above-mentioned change to the debugging mode of EQL0D. This would allow for the same Serpent output to be re-used for all runs of the same sensitivity analysis. Note, however, that any advantage gained by removing the impact of unexplainable differences between the cases may be lost in terms of accuracy due to the differences in material input for Serpent that cases with different removal rates should naturally have.

It is possible that the perceived failures to obtain a sensitivity analysis for the MSFR-like cases is merely the inability to predict certain reactivity effects, however this seems unlikely seeing as the cases are extremely basic.

## 8.7 Complex Case

The results obtained for the complex case are very interesting, however they come with a severe issue for gas bubbling storage which is outlined in the according results section. One of the main problems in understanding where this issue comes from is the complete lack of reproducibility with less complex systems. A lot of different approaches to try and locate the issue were made, including redistributing the operation time into smaller steps as well as separating the storage into four separate locations split between metallic/volatile as well as based on the origin in terms of blanket or fuel. Both cases experience the exact same issue as the initial setup. In the future, some time will need to be invested in having a closer look at the code itself, in order to try and pinpoint the issue. It seems unlikely that the issue originates from a mistake in the code itself, as no other material is affected. Instead, it may be that the context with the complex case as such is what causes the loss of material. An option which would be promising is to try and reproduce the problem in another complex case, such as one made for a chloride salt. It should be entirely possible to rebuild the inputs for another salt, even if the literature for a reprocessing system may be problematic. In the end, the goal of such a calculation would be to reproduce the issue for a different case, rather than make the most accurate possible description of a real system.



Even with the option of describing another system with the express purpose of trying to find the issue with EQL0D, there is definitely a case to be made for designing an input in order to try and be as accurate as possible with the results. While literature information on the design of a reprocessing system for chloride salts is much more scarce than for the classical fluoride/Thorium setup, a lot of similarities to the fluoride can probably be exploited due to a relatively strong similarity in terms of chemical behaviour for the two halides. In case the results have some issue with an element not being sufficiently treated by the reprocessing system, there is always the option of having a more iterative approach once the time constraints of a master's thesis are removed.

On top of altering the system to include different salt mixtures, another option is to extend the work to other types of reactors, such as breed and burn.

## 8.8 Final Thoughts

During the work performed in the scope of this thesis, a lot of issues relating to the codes used as well as the time available arose. A considerable part of the time was committed to very foundational work, such as rerunning cases and attempting to optimize parameters. While the results thereof are relatively basic in nature, the sheer amount of effort is made worth by the fact that a solid thermodynamic database is absolutely necessary in order to obtain reliable results from future projects.

A similar kind of work was done with respect to EQL0D; Some of the biggest advantages gained in the scope of this work is the expertise, which can and will be applied to future cases. In addition, the tutorial created as part of this work may benefit future users. There are still issues left to resolve, however considerable strides have been made both in terms of usage as well as understanding of the outputs.

A lot of the aspects that were omitted from this thesis due to time constraints will be continued after the deadline as part of an after-master internship at PSI. This includes continuing the work on updating the HERACLES database, especially focussed around finishing up the last binary interactions as well as improving the tertiary systems. On the EQL0D front, it will continue to be used for the SAMOSAFER benchmark, which is due to increase in complexity and challenge for the code. In addition, with the removal of the time restriction, changes which pertain to the code itself become viable as a means to address some of the issues discovered and discussed.

## References

- [1] G. Locatelli, M. Mancini, N. Todeschini, “Generation IV nuclear reactors: Current status and future prospects”, *Energy Policy* **2013**, *61*, 1503–1520.
- [2] J. Serp, M. Allibert, O. Beneš, S. Delpech, O. Feynberg, V. Ghetta, D. Heuer, D. Holcomb, V. Ignatiev, J. L. Kloosterman, L. Luzzi, E. Merle-Lucotte, J. Uhlíř, R. Yoshioka, D. Zhimin, “The molten salt reactor (MSR) in generation IV: Overview and perspectives”, *Progress in Nuclear Energy* **2014**, *77*, 308–319.
- [3] P. Rubiolo, D. Heuer, E. Merle, M. Brovchenko, V. Ghetta, M. Allibert, A. Laureau, “OVERVIEW AND PERSPECTIVES OF THE MOLTEN SALT FAST REACTOR (MSFR) CONCEPT”, **2013**.
- [4] R. Liberatore, M. Lanchi, G. Caputo, C. Felici, A. Giaconia, S. Sau, P. Tarquini, “Hydrogen production by flue gas through sulfur–iodine thermochemical process: Economic and energy evaluation”, *International Journal of Hydrogen Energy* **2012**, *37*, 8939–8953.
- [5] MATLAB version 9.6.0.1072779 (R2019a), The Mathworks, Inc., Natick, Massachusetts, **2019**.
- [6] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta, T. Kaltiaisenaho, “The Serpent Monte Carlo code: Status, development and applications in 2013”, *Annals of Nuclear Energy*, Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013, SNA + MC 2013. Pluri- and Trans-disciplinarity, Towards New Modeling and Numerical Simulation Paradigms **2015**, *82*, 142–150.
- [7] GEM Software (GEMS) Home, <http://gems.web.psi.ch/> (visited on Sept. 20, 2020).
- [8] D. A. Kulik, T. Wagner, S. V. Dmytrieva, G. Kosakowski, F. F. Hingerl, K. V. Chudnenko, U. R. Berner, “GEM-Selektor geochemical modeling package: revised algorithm and GEMS3K numerical kernel for coupled simulation codes”, *Computational Geosciences* **2013**, *17*, 1–24.
- [9] T. Wagner, D. Kulik, F. Hingerl, S. Dmytrieva, “GEM-Selektor Geochemical Modeling Package: TSolMod Library and Data Interface for Multicomponent Phase Models”, *The Canadian Mineralogist* **2012**, *50*, 1173–1195.
- [10] GEMS specific HERACLES v.0.2. database for U, TRU and FP speciation, Paul Scherrer Institut (PSI), <https://www.psi.ch/en/heracles/gems-specific-heracles-database> (visited on Sept. 20, 2020).
- [11] J. D. G. de Oliveira, J. C. R. Reis, “The two faces of the Redlich–Kister equation and the limiting partial molar volume of water in 1-aminopropan-2-ol”, *Thermochimica Acta* **2008**, *468*, 119–123.
- [12] B. A. Hombourger, PhD thesis, École Polytechnique Fédérale de Lausanne, Lausanne, **2018**, 169 pp.
- [13] A. Isotalo, M. Pusa, “Improving the Accuracy of the Chebyshev Rational Approximation Method Using Substeps”, *Nuclear Science and Engineering* **2016**, *183*, Publisher: Taylor & Francis \_eprint: <https://doi.org/10.13182/NSE15-67>, 65–77.
- [14] L. Humphries, B. Beeny, F. Gelbard, D. Louie, J. Phillips, MELCOR Computer Code Manuals Vol. 1: Primer and Users’ Guide, **2017**.

- [15] J. Kalilainen, S. Nichenko, J. Krepel, “Evaporation of materials from the molten salt reactor fuel under elevated temperatures”, *Journal of Nuclear Materials* **2020**, *533*, 152134.
- [16] В Википедия, Фторид нептуния(III), Page Version ID: 87622621, 2017.
- [17] D. M. Gruen, C. W. DeKock, “Electronic absorption spectra of gaseous PuCl<sub>3</sub>, PuBr<sub>3</sub> and PuCl<sub>4</sub>: The thermodynamics of PuCl<sub>4</sub>”, *Journal of Inorganic and Nuclear Chemistry* **1967**, *29*, 2569–2575.
- [18] Collaboration: Scientific Group Thermodata Europe (SGTE) in *Pure Substances Part 3: Compounds from CoCl<sub>3</sub><g> to Ge<sub>3</sub>N<sub>4</sub>*, Vol. 19 A3, (Eds.: Lehrstuhl für Theoretische Hüttenkunde, Rheinisch-Westfälische Technische Hochschule Aachen), Series Title: Landolt-Börnstein - Group IV Physical Chemistry, Springer-Verlag, Berlin/Heidelberg, **1999**, pp. 325–350.
- [19] J. L. Burnett, “Melting points of CmF<sub>3</sub> and AmF<sub>3</sub>”, *Journal of Inorganic and Nuclear Chemistry* **1966**, *28*, 2454–2456.
- [20] J. P. M. van der Meer, R. J. M. Konings, M. H. G. Jacobs, H. A. J. Oonk, “Thermodynamic modelling of LiF–LnF<sub>3</sub> and LiF–AnF<sub>3</sub> phase diagrams”, *Journal of Nuclear Materials* **2004**, *335*, 345–352.
- [21] O. Beneš, R. J. M. Konings, “Thermodynamic evaluation of the NaCl – MgCl<sub>2</sub> – UCl<sub>3</sub> – PuCl<sub>3</sub> system”, *Journal of Nuclear Materials* **2008**, *375*, 202–208.
- [22] Collaboration: Scientific Group Thermodata Europe (SGTE) in *Pure Substances Part 2: Compounds from BeBr<g> to ZrCl<sub>2</sub><g>*, Vol. 19 A3, (Eds.: Lehrstuhl für Theoretische Hüttenkunde, Rheinisch-Westfälische Technische Hochschule Aachen), Series Title: Landolt-Börnstein - Group IV Physical Chemistry, Springer-Verlag, Berlin/Heidelberg, **1999**, pp. 325–350.
- [23] *Pure Substances. Part 1 \_\_ Elements and Compounds from AgBr to Ba<sub>3</sub>N<sub>2</sub>*, (Eds.: Lehrstuhl für Theoretische Hüttenkunde, Rheinisch-Westfälische Technische Hochschule Aachen), Springer-Verlag, Berlin/Heidelberg, **1999**.
- [24] G. Van Rossum, F. L. Drake, *Python 3 Reference Manual*, CreateSpace, Scotts Valley, CA, **2009**.
- [25] R. Oyamada, “Phase Diagrams of ThCl<sub>4</sub> Systems Containing NaCl, KCl and LiCl and the Limitation of the Congruently Melting Compounds Formation”, *Denki Kagaku oyobi Kogyo Butsuri Kagaku* **1971**, *39*, 2–5.
- [26] E. Sooby, M. Adams, A. Baty, J. Gerity, P. McIntyre, K. Badgley, S. Phongikaroon, N. Pogue, A. Sattarov, M. Simpson, P. Tripathy, P. Tsvetkov, “Molten Salt Considerations For Accelerator-Driven Subcritical Fission To Close The Nuclear Fuel Cycle”, *AIP Conference Proceedings* **2013**, *1525*, 230–235.
- [27] В. Н. Десятник, М. П. Воробей, Н. Н. Курбатов, И. С. Калащников, О. В. Скиба, «Диаграммы плавкости тройных систем содержащих хлориды натрия, калия, тетрахлорид тория и трихлорид плутония», *Атомная энергия* 1975, 38.
- [28] E. Capelli, O. Beneš, R. J. M. Konings, “Thermodynamic assessment of the LiF–ThF<sub>4</sub>–PuF<sub>3</sub>–UF<sub>4</sub> system”, *Journal of Nuclear Materials* **2015**, *462*, 43–53.
- [29] R. Mulford, Calculation of the LiF–CeF<sub>3</sub>–BeF<sub>2</sub> and LiF–PuF<sub>3</sub>–BeF<sub>2</sub> Ternary Phase Diagrams, **1993**.

- [30] O. Beneš, R. J. M. Konings, “Thermodynamic properties and phase diagrams of fluoride salts for nuclear applications”, *Journal of Fluorine Chemistry*, Fluorine & Nuclear Energy **2009**, *130*, 22–29.
- [31] E. Capelli, O. Beneš, R. J. M. Konings, “Thermodynamics of soluble fission products cesium and iodine in the Molten Salt Reactor”, *Journal of Nuclear Materials* **2018**, *501*, 238–252.
- [32] IAEA, Advanced Reactors Information System (ARIS), <https://aris.iaea.org/PDF/MSFR.pdf> (visited on Sept. 26, 2020).
- [33] S. Delpech et al., “Description of the chemical plant required for the MSFR fuel reprocessing”, *SAMOFAR Horizon 2020 Euratom Research Project Contract Number:661891* **2018**.

## Appendix

### EQL0D Manual

EQL0D generally needs at least two input files; One for MATLAB and one for Serpent. As outlined in the main document, MATLAB treats the evolution of materials as well as reprocessing schemes and the like, while Serpent is responsible for handling only the neutronic side of the problem. Therefore, it is in the MATLAB file that a lot of options with respect to reprocessing and algorithm control need to be placed. In turn, the Serpent input file is the location for the geometric setup of the system in question. Materials are generally defined in both.

#### Input Variables for MATLAB

In MATLAB, input variables are generally summarized in structs, which are separated into categories of options. Materials are defined by using the Mat class constructor. An additional possibility is to define reprocessing streams, which use the Rep class. Each additional stream adds a new object to the REP vector, which is used for reprocessing.

#### Variables in MATLAB

EQL0D is quite heavily based on structs and classes. While both of these are extremely powerful tools when coding in MATLAB, they can be somewhat unintuitive if not familiar with the language. While it is not necessary to manipulate the code when using it, it can be useful to have an understanding of the basics in case there is a need to gain a deeper understanding of the algorithm or even make some small changes should the need arise.

In general, the purpose of a class in MATLAB is to have a way of ordering objects of similar type into a group, all of which have the same data points. Classes are defined in a separate MATLAB file, and have a so-called "constructor" function. This function defines what properties are necessary in order to create an object of the wanted kind. Therefore, when trying to create a variable containing an object of said class, all of the values for these properties need to be provided. Usually, there are additional functions provided within a class, which will return values or alter the created variable itself. To call these functions, one can use the dot notation. If you have created an object called 'bike' of the 'vehicle' class, you can access the bike's velocity by typing 'bike.velocity' (assuming there is a function in place to calculate this velocity).

Structs are variables which contain multiple so-called fields. The powerful thing about structs is that the fields can be named, and can contain any type of variable, including another struct. It is therefore theoretically possible to save all data related to some MATLAB code in just one variable. While that would be somewhat over the top, it can be very convenient to group a lot of the used variables and data that are related to each other in a single struct. This is done in EQL0D a considerable amount, a good example being the above-mentioned OPT struct containing options for the running of the program. Similarly to the class above, information within a struct can be accessed using the dot notation: Say you've created a struct called 'Me' containing a field called 'phonenumber', you can make MATLAB return your phone number by typing 'Me.phonenumber'.

This notation can be extended for struct within structs, meaning you can chain multiple dot notations together to obtain data stored multiple levels within the struct.

**Table 11:** Summary of the MATLAB input variables contained in the OPT structure for general options.

Variable	Purpose	Type
nCycles	Gives the number of cycles for the current run, which is the number of outer loop iterations of the algorithm. Serpent is called once per cycle.	Positive integer
cycleLength	Gives the length of each cycle in days.	Vector of positive integers
nSteps	Gives the number of inner loop iterations per outer loop iteration.	Vector of positive integers
iterMode	Used to switch between step- and equilibrium mode.	'steps' / 'equilibrium'
reactControl	Switch reactivity control on or off.	Boolean
redoxControl	Switch redox control on or off.	Boolean
PCC	Switch predictor/corrector functionality on or off.	Boolean
renormalize	Renormalizes the burn matrices based on the ratio of current and target fission rates.	Boolean
serpentPath	Used to call Serpent in the outer loop.	String
nSubSteps	Subdivides the solving of the burn-matrix into n individual steps.	Positive integer
defaultDataLibrary	Set the default library for nuclear cross-sections.	'endfb7' / 'jeff311'
writeMail	Writes a mail to the user on completion or crash.	Boolean
matWriteStyle	Alters the style of the created Serpent files.	'dec' / 'fix' / 'nofix'
keepFiles	If enabled, saves the results of each cycle in an appropriately named folder.	Boolean

There are additional values stored within the OPT struct, however they are part of a struct within OPT, and will be separately listed below.

The REA struct is contained within the option storage struct OPT. Below, the use of REA and the parameters within it is outlined. There are additional options not mentioned in the table below because there are a lot of them that work very similarly to each other. These will be discussed in text below.

**Table 12:** Additional OPT variables contained in substruct REA.

Variable	Purpose	Type
allowRemoval	Changes the reprocessing such that removing material from the core is allowed.	Boolean
allowAddition	Changes the reprocessing such that adding material to the core is allowed.	Boolean
targetKeff	Used by reactivity control to check how far from target the reactivity is.	Double
tol	This is the tolerance used to judge if the distance to target is large enough to warrant action.	Double
maxIter	The maximum number of iterations allowed for the loop to adjust the reactivity.	Positive Integer

In addition to the above variables, there are multiple variables assigned to defining what isotopes and materials are used and affected by reactivity control. Generally, the variables `feedMat` and `targetMat` are assigned to the feed and fuel materials defined within the MATLAB input. They define which material is to be optimized with respect to reactivity and where the replacement material comes from.

Additionally, there are vectors defining which nuclides as well as their distribution are to be used in order to adjust the reactivity, both in terms of being removed from the fuel, added to the fuel as well as what isotopes are to be used as replacement to keep the mass balance.

There are two more sub-structs within the OPT framework, which are `CONV` for convergence criteria and `REDOX` to adjust the redox control algorithm. Both have only relatively few options, and will be presented together in the below table. Note that each variable relating to convergence has an inner as well as outer version, for the respective loops in the main `EQL0D` algorithm.

**Table 13:** Additional OPT variables contained in substructs CONV and REDOX.

Category	Variable	Purpose	Type
CONV	criterion	Defines how the difference between current and previous result is calculated.	'maxRelDiff' / 'maxNucDatRelDiff' / 'maxActRelDiff'
CONV	value	Sets the cutoff value for when a case is considered converged.	Double
CONV	cutoff	Sets the cutoff value for the minimum amount of substance that needs to be present to be considered for convergence.	Double
REDOX	materials	Defines which materials should be affected by redox control.	String of material name
REDOX	replaceMode	Switched between different algorithms for redox control.	'replaceMass' / 'replaceAtom' / 'remove'
REDOX	replaceWith	Denotes the isotope mixture used to replace the changes made during redox control.	Positive integer / ZAI

### EQL0D Files and Their Purpose

EQL0D is comprised of many separate functions. In the following, a quick insight into their purpose is given, without going into too much mathematical detail.

In EQL0D, most functions are separated into their own file. Even the main launch-file is a function, due to it being called from the command line.

**Table 14:** Summary of files that EQL0D is comprised of. Note that the MATLAB file extension .m is omitted here.

File	Purpose
EQL0Dlauncher	Sets some general options for MATLAB and calls the main file EQL0D.
EQL0D	Contains the main loops that EQL0D is based on and calls the according functions to make the changes to the materials between cycles.
initialize	Initializes variables, loads files for future reading and writing, loads default configuration of options and does general housekeeping.
defaultConfig	Stores default values and settings, which are overwritten if the user provides a different value.
createRepMatrices	Creates the matrices for burning, reprocessing and so forth from material and reprocessing objects.
buildSystemMatrix	Combines matrices for decay, burn and reprocessing to form the overall system matrix.
parseInput	Called by initialize to parse the input data.
runSerpent	Called by EQL0D once per cycle to run Serpent using the current material definitions.



**Table 15:** Extension of table 14.

File	Purpose
loadSerpentData	Gets necessary data from the Serpent outputs, such as reaction rates, the burnup matrices and k-effective/infinity.
modifyInput	Can automatically adapt Serpent input files to values for the next cycle.
neutronBalance	Computes and prints the neutron balance for the system.
printK	Prints k-eff and k-inf to an output file.
renormalizeBurnMatrices	If the renormalize option is enabled, this function will adjust the burnmatrix to the new fission rate.
saveFiles	Moves files generated in each cycle to a separate folder if the option is enabled. Otherwise, they are overwritten.
testConvergence	Checks if convergence has been reached for equilibrium mode based on convergence criteria set in input.
updateRates	If PCC is enabled, this functions adjusts the reaction rates to be the interpolated values.
burnCycle	Evolves the materials to the next step, including actual burning, reprocessing as well as redox and reactivity control.
CRAMsolve	Called by burnCycle to use CRAM to solve the burning of materials.
computeK	Calculates k-eff and k-inf from reaction rates.
reactivityControl	Adapts the material composition to match the target reactivity based on user options given in REA.
isActinide / isFP / isProduced / isElement	These functions check is the passed ZAI is part of: Actinides, Fission Products, the chosen nuclear data library or if it is an element rather than an isotope.
hasSymbol	Used by the Mat class for printing material compositions.
hasXSData	When given a vector of ZAI, returns only those that have a cross-section entry in the data library.
ingAndInhTox	Computes inhalation and ingestion toxicity for the end-of-cycle material summary.
printStatus	Prints a summary of the burned material to the log once per cycle.
valenceStates	Returns the expected valency for a vector of ZAI that is obtained.

In order to use EQL0D, it is likely not necessary to understand every single function precisely, however any issue or modification that comes up can only viably be tackled with some amount of knowledge of the workings of the code.

### The Mat Class

In EQL0D, materials are defined by the user in the input file. To do this, one makes use of the constructor function of the Mat class. On top of storing material information, there are a lot of functions available and used during the procedure. These include, but are not limited to: Redox Control, extracting any and all data about the saved material or printing the material to an output file.

The constructor requires the inputs given in table 16, in the order that they appear. They need to be comma-separated.

**Table 16:** Variables used to construct a material object.

Variable	Use	Type
Name	Names the material.	String
Type	Defines whether a material is in flux, burned, or unaffected.	0 / 1 / 2
Density	Defines the density of your material. If it is negative, mass density is assumed. If positive, atomic density is assumed. In the case of density=0, MATLAB will calculate the density automatically.	Double
Volume	Used to state the volume of the material, which is used in the computation of the atomic composition.	Double
Temperature	Defines the temperature.	Double
ZAI	Used to specify which isotopes are present in the system. Defined as a Vector.	Vector of positive integers
Composition	Vector of the same length as ZAI which specifies the prevalence of each isotope.	Vector of doubles

### The Rep Class

Similarly to the material definition above, reprocessing streams are defined in the MATLAB input file using the constructor function of the Rep class. There are additional functions contained within Rep, the most important one being batchProcessing. That function is called by the burnCycle function to apply the batch reprocessing steps to the materials.

**Table 17:** Variables used to construct a reprocessing object.

Variable	Use	Type
Name	Names the material.	String
Source Material	Defines which material the stream is taken from.	String that's a name of a material
Target Material	Defines which material the stream is going to	String that's a name of a material
Elements	Provides a list of elements which are taken by the reprocessing stream.	Vector of positive integers
Coefficient	Adjusts the efficiency of the removal from the source material.	Vector of doubles
Rate	Gives the rate at which the material is removed.	Double
Type	Specifies the type of reprocessing that is performed.	'batch' or 'cont'

## Tutorial

In the following, a small, simplified tutorial will be given as to how to effectively run a case. Note that some parts are specific to the PSI cluster setup, which involves the use of the slurm queueing system.

There are a couple of files that are necessary to have in order to run EQL0D. First of all, the general input files for MATLAB and Serpent as well as the EQL0D installation are obviously required. Additionally, the slurm system requires a launch file. On top of that, each material that is expected to change during the run will need its material definition to be located in an auxiliary file to the main Serpent input. Those files will need to be named after the material they represent, with the extension ".serp".

It is often convenient to have the geometry as well as unburned materials in a separate file as they are shared between cases, however this is not necessary.

While the two main input files have been mostly described using the above tables, some attention will be given here to the auxiliary files.

The slurm launch file is necessary as a bridge between the UNIX-based launch script of EQL0D and the command line. In theory, it would be possible to launch the routine directly from the command line, however with the slurm system this is very inefficient. Therefore, the launch file is used to enter the queue and pass on the required launch parameters to EQL0D once the resources have been obtained from the cluster.

For this, the following information is required:

- Specifications for slurm, which includes information on runtime, required number of cpus as well as memory per cpu
- Loading the MATLAB module (if modules are used on the cluster this is run on)
- Add the current path to the MATLAB directory, if it is not already added
- Change directory to where the files are located
- Call the UNIX-based EQL0D launch file while specifying options.
  - -n 44 changes the number of cpus to 44 (or whatever your specifications are)
  - -v enables verbose mode, which prints more information to the log file.
  - -b enables batch mode
  - -r and -R denote a restart / reset mode respectively. Restart implies restarting an interrupted calculation, whereas reset starts from the beginning.
  - State the name of the case, which needs to be the name of both input files (excluding the .m extension)

In order to submit the above UNIX file to the slurm queue, use the 'sbatch' command followed by the name of the launch script on the command line of the cluster. This will submit the job. Once resources have been allocated to the job, the EQL0D launch file will be called, which will subsequently start the routine.

Once the run completes, there will be numerous files and folders created in the directory specified within the slurm launch file. Assuming that the keepFiles option is enabled in EQL0D, there will be a folder for each cycle that contains Serpent inputs and outputs from that cycle. In addition, each material will produce an output for each cycle, specifying both the isotopic as well as elemental composition. This is also true for materials which are not explicitly defined, but produced as a consequence of a reprocessing stream.

There are also two main log files created during the run of EQL0D, one by the routine itself and one by Serpent. The Serpent version is named 'sss2.log', whereas the overall EQL0D log file is named after the case that was run. Note that outputs which would normally be directed to the command line are not naturally available when using slurm. To remedy this, the output and error files can be allocated to filenames in the options, which will collect and store anything directed to the command line. This is especially useful before one is sure that an EQL0D attempt will work without issue.

## GEMS Project Parameters

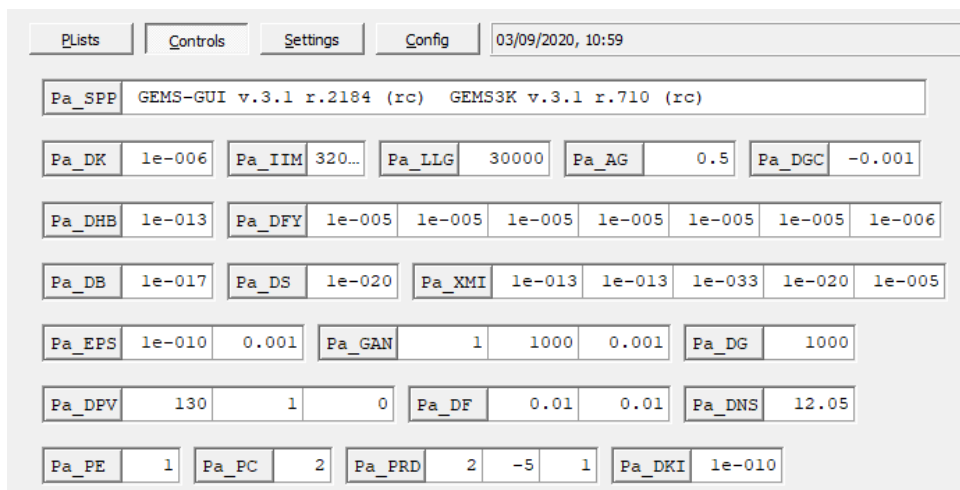
### Chlorides

Pa_SPP	GEMS-GUI v.3.1 r.2184 (rc) GEMS3K v.3.1 r.710 (rc)										
Pa_DK	1e-007	Pa_IIM	320...	Pa_LLG	30000	Pa_AG	0.5	Pa_DGC	0.01		
Pa_DHB	1e-013	Pa_DFY	1e-005	1e-005	1e-005	1e-005	1e-005	1e-005	1e-005	1e-006	
Pa_DB	1e-017	Pa_DS	1e-020	Pa_XMI	1e-013	1e-013	1e-033	1e-020	1e-005		
Pa_EPS	1e-010	0.001	Pa_GAN	1	1000	0.001	Pa_DG	1000			
Pa_DPV	130	1	0	Pa_DF	0.01	0.01	Pa_DNS	12.05			
Pa_PE	1	Pa_PC	2	Pa_PRD	2	-5	1	Pa_DKI	1e-010		

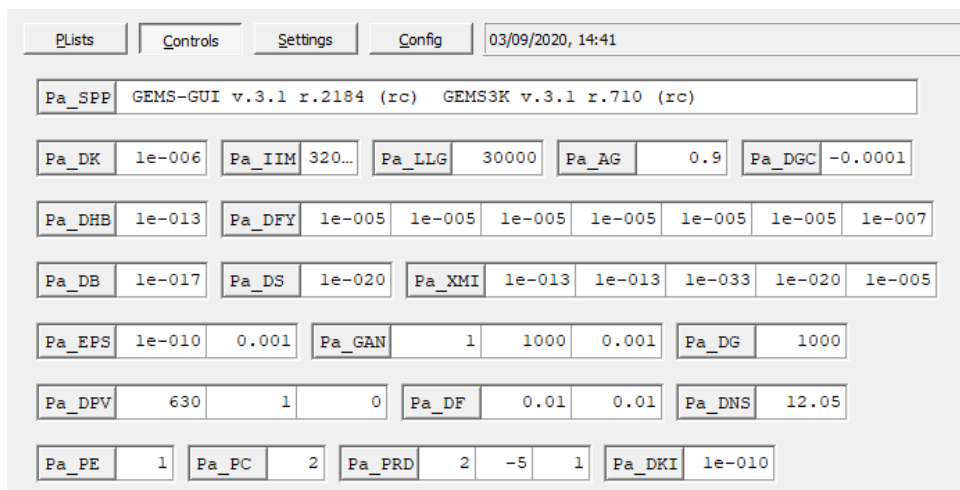
**Figure 32:** GEMS project settings for the initial NaCl-ThCl<sub>4</sub> phase diagram in section 6.4.1. For some intermediate steps that followed, the tolerance threshold (Pa\_DK) was lowered to 1e-006 depending on necessity.

Pa_SPP	GEMS-GUI v.3.1 r.2184 (rc) GEMS3K v.3.1 r.710 (rc)										
Pa_DK	1e-006	Pa_IIM	320...	Pa_LLG	30000	Pa_AG	0.6	Pa_DGC	0.01		
Pa_DHB	1e-013	Pa_DFY	1e-005	1e-005	1e-005	1e-005	1e-005	1e-005	1e-005	1e-006	
Pa_DB	1e-017	Pa_DS	1e-020	Pa_XMI	1e-013	1e-013	1e-033	1e-020	1e-005		
Pa_EPS	1e-010	0.001	Pa_GAN	1	1000	0.001	Pa_DG	1000			
Pa_DPV	130	1	0	Pa_DF	0.01	0.01	Pa_DNS	12.05			
Pa_PE	1	Pa_PC	2	Pa_PRD	2	-5	1	Pa_DKI	1e-010		

**Figure 33:** GEMS project settings for the final NaCl-ThCl<sub>4</sub> phase diagram in section 6.4.2. There is only a small adaptation required with respect to the initial diagram.



**Figure 34:** GEMS project settings used to construct the NaCl-PuCl<sub>3</sub>, NaCl-UCl<sub>3</sub> and ThCl<sub>4</sub>-PuCl<sub>3</sub> phase diagrams.



**Figure 35:** GEMS project settings used to construct the LiF-PuF<sub>3</sub> phase diagram.

## Fluorides

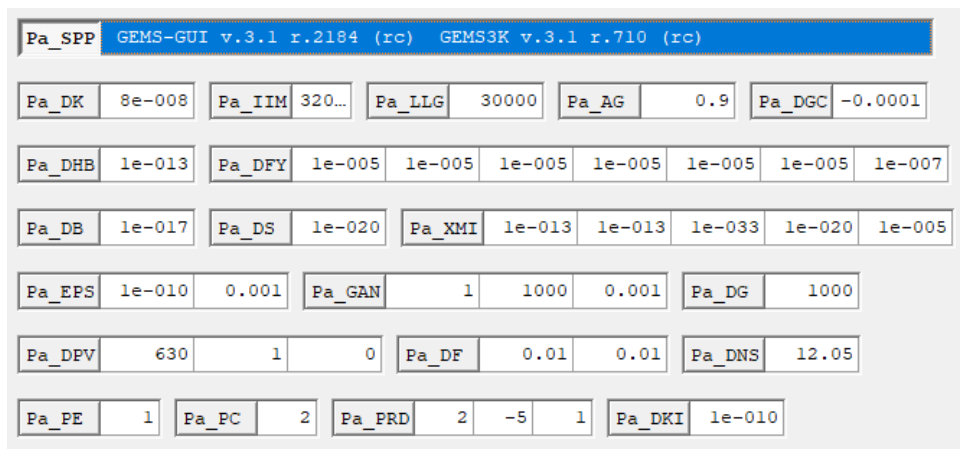


Figure 36: GEMS project settings used to construct the LiF-ThF<sub>4</sub> phase diagram.

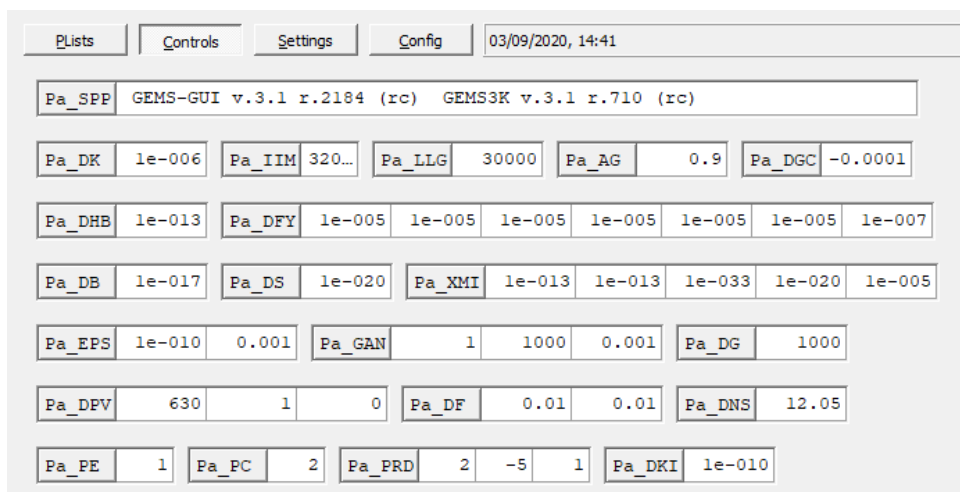


Figure 37: GEMS project settings used to construct the LiF-PuF<sub>3</sub> phase diagram.

## Ternary Diagram

The screenshot shows the 'Config' tab of the GEMS software interface. The title bar indicates the date and time as 24/09/2020, 16:20. The main window contains several rows of input fields for project settings. The first row shows 'Pa\_SPP' with values 'GEMS-GUI v.3.1 r.2184 (rc)' and 'GEMS3K v.3.1 r.710 (rc)'. Subsequent rows contain various parameters such as 'Pa\_DK', 'Pa\_IIM', 'Pa\_LLG', 'Pa\_AG', 'Pa\_DGC', 'Pa\_DHB', 'Pa\_DFY', 'Pa\_DB', 'Pa\_DS', 'Pa\_XMI', 'Pa\_EPS', 'Pa\_GAN', 'Pa\_DPV', 'Pa\_DF', 'Pa\_DNS', 'Pa\_PE', 'Pa\_PC', 'Pa\_PRD', and 'Pa\_DKI', each with numerical or string values.

**Figure 38:** GEMS project settings used to construct the CsF-LiF-ThF<sub>4</sub> ternary phase diagram.

## Representative Salt Analysis

1	Liquid	14	1	0.02744012	-1.134e-05			
	AmF3 (1)	M	5.78739e-005	0.00553403	0.0021090979	2.6236791		
	CeF3 (1)	M	1.1166e-005	0.00106772	0.00040692242	2.6236791		
	LiF (1)	M	0.020147278	0.509351	0.73422706	0.69394101		
	NdF3 (1)	M	1.42509e-005	0.0013627	0.0005193454	2.6236791		
	NpF3 (1)	M	6.7234601e-008	6.42911e-06	2.4502299e-006	2.6236791		
	NpF6 (1)	M	3.4596811e-023	3.30829e-21	1.2608112e-021	2.6236791		
	PrF3 (1)	M	4.56187e-006	0.000436216	0.00016624818	2.6236791		
	PuF3 (1)	M	0.0015795458	0.0312913	0.057563372	0.54348483		
	PuF4 (1)	M	1.1856352e-005	0.00113373	0.00043208091	2.6236791		
	UF3 (1)	M	1.6387286e-012	1.56691e-10	5.972017e-011	2.6236791		
	UF4 (1)	M	0.0055973341	0.0130583	0.20398359	0.063956634		
	ZrF2 (1)	M	1.3396809e-025	1.28101e-23	4.8821976e-024	2.6236791		
	ZrF3 (1)	M	2.7960994e-017	2.67368e-15	1.0189822e-015	2.6236791		
	ZrF4 (1)	M	1.6185029e-005	0.00154765	0.00058983084	2.6236791		

**Figure 39:** Liquid phase composition for the nominal run of the representative salt for the fluoride/Uranium system.



Pa_SPP	GEMS-GUI v.3.1 r.2184 (rc) GEMS3K v.3.1 r.710 (rc)										
Pa_DK	1e-006	Pa_IIM	320...	Pa_LLG	30000	Pa_AG	0.9	Pa_DGC	0.001		
Pa_DHB	1e-013	Pa_DFY	1e-005	1e-005	1e-005	1e-005	1e-005	1e-005	1e-005	1e-007	
Pa_DB	1e-017	Pa_DS	1e-020	Pa_XMI	1e-013	1e-013	1e-033	1e-020	1e-005		
Pa_EPS	1e-010	0.001	Pa_GAN	1	1000	0.001	Pa_DG	1000			
Pa_DPV	630	1	0	Pa_DF	0.01	0.01	Pa_DNS	12.05			
Pa_PE	1	Pa_PC	2	Pa_PRD	2	-5	1	Pa_DKI	1e-010		

**Figure 40:** GEMS project settings used for the plots depicting the influence of fluorine content and temperature on the base salt.

## Additional Information and Data for the SAMOSAFER Benchmark

**Table 18:** Elemental assignment to the different reprocessing avenues out of the MSR fuel employed in the benchmark, using the atomic numbers to denote the different elements.

Category	Elements
Volatiles	5:7, 10, 14:16, 18, 27:34, 36, 40:52, 54, 73:86
Solubles	1, 2, 4:8, 10:88

## Additional Information and Data for EQL0D Settings

### Example Geometry Input

```
% MSFR-like geometry for core volume of 14.000000 m^3.
surf refl_outer cyl 0.0 0.0 259.16 -230.61 230.61
surf refl_inner cyl 0.0 0.0 239.16 -130.61 130.61
surf prot_outer cyl 0.0 0.0 211.52 -108.89 108.89
surf blkt_outer cyl 0.0 0.0 191.52
surf blkt_inner cyl 0.0 0.0 189.52 -106.89 106.89
surf core_outer cyl 0.0 0.0 132.61
surf core_inner cyl 0.0 0.0 130.61
```

**Figure 41:** Serpent geometry definition for a fluoride-based Thorium-232 example of the four homogenized cases.

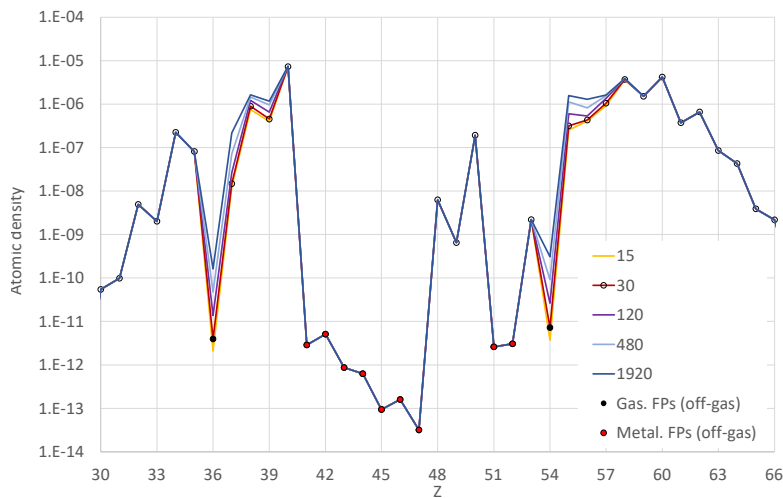
**Table 19:** Elemental assignment to the different reprocessing avenues out of the MSR fuel for the homogenized cases, using the atomic numbers to denote the different elements.

Removal Path	Category	Elements
Helium Bubbling	Volatiles	1 2 7 8 10 18 36 54 86
	Metallics	41:1:47 51 52
Reprocessing	Solubles	30:35 37:40 48:50 53 55:70
	Others	3 4:6 9 12:16 19 20:29 71:89 97:110

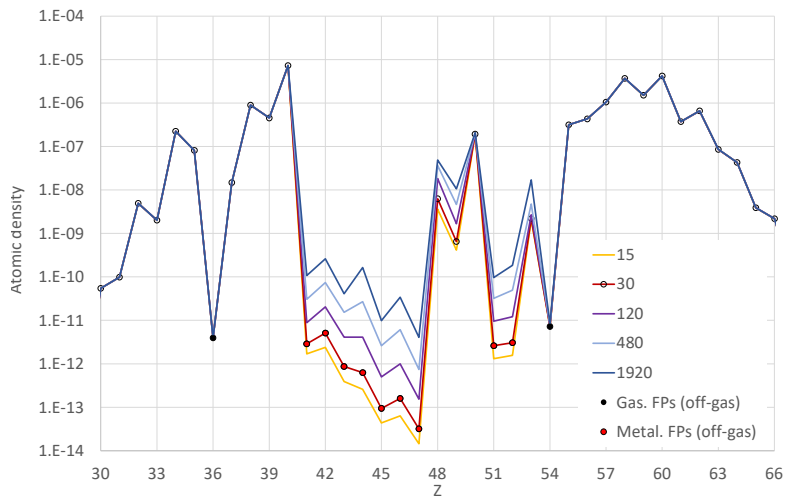
Note that the elements removed by reprocessing labelled as "others" are simply all elements that are not otherwise removed, part of the carrier salt or an actinide. While elements higher in atomic number than Curium are also removed, they do generally not occur in the salt due to the lack of production mechanisms and are therefore not influential for the result.

## Additional Sensitivity Plots

### Chloride-Thorium

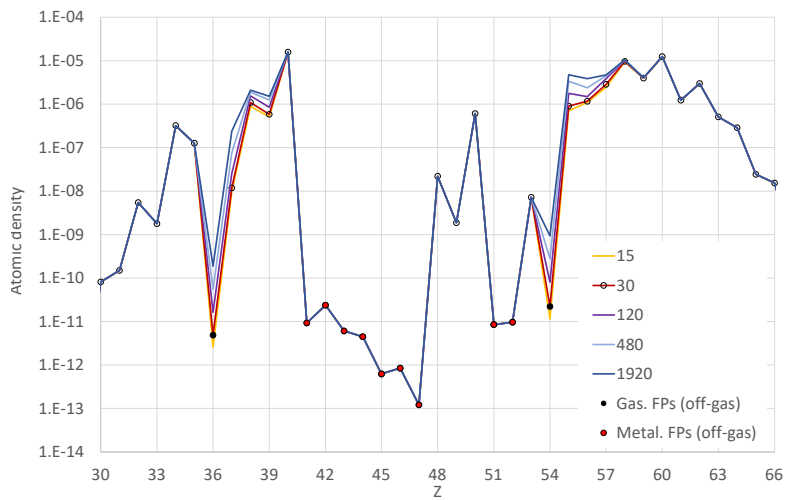


**Figure 42:** Plot of chloride-based Thorium cases with different volatile removal rates.



**Figure 43:** Plot of chloride-based Thorium cases with different metallic removal rates.

### Chloride-Uranium



**Figure 44:** Plot of chloride-based Uranium cases with different volatile removal rates.

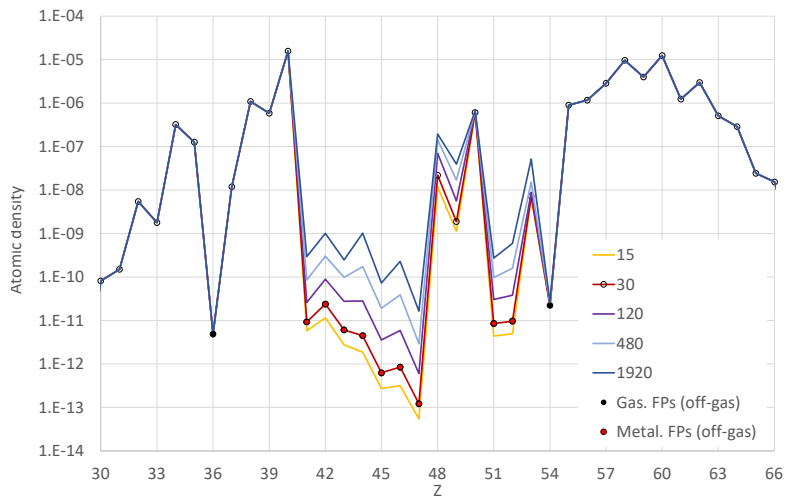


Figure 45: Plot of chloride-based Uranium cases with different metallic removal rates.

### Fluoride-Uranium

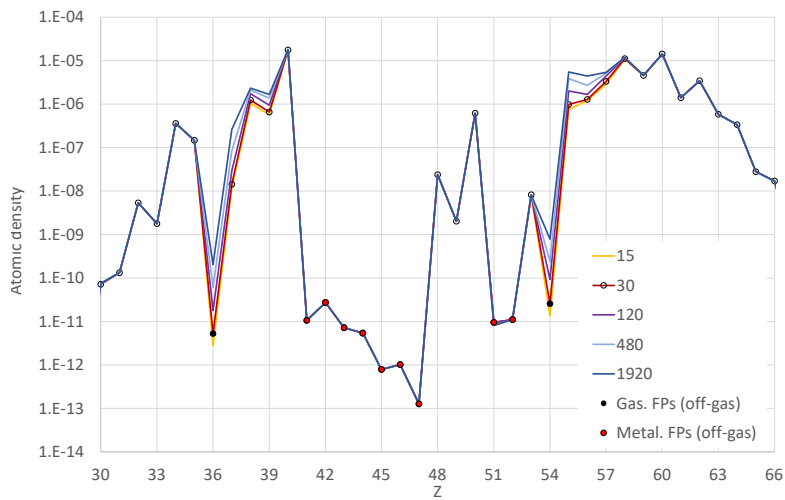


Figure 46: Plot of fluoride-based Uranium cases with different volatile removal rates.

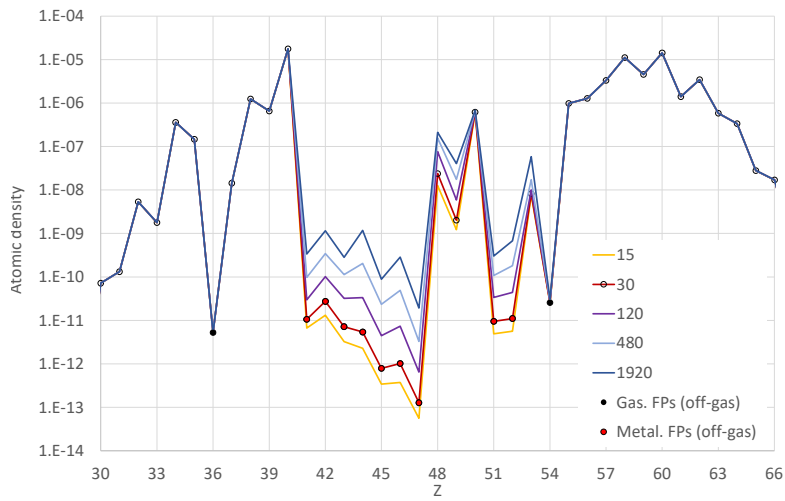


Figure 47: Plot of fluoride-based Uranium cases with different metallic removal rates.

### Failed Sensitivity Study Example

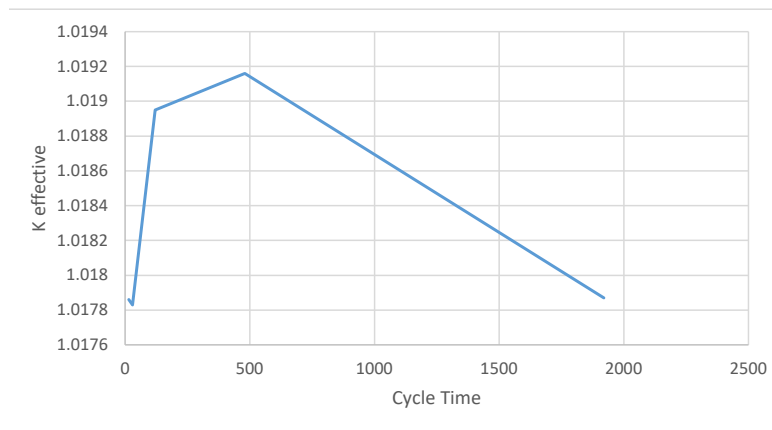


Figure 48: Example of a failed sensitivity study for the fluoride-based Uranium system.