

Technical University of Denmark

Initiation of an open-source molten salt data library.



Atli Freyr Magnsson - s172415

Binoy Hasmukh Shah - s171890

Supervisors:

Associate professor at DTU - Peter Szabo

Chief Technology Officer at Copenhagen Atomics - Aslak Stubsgaard

A report submitted in partial fulfillment of the requirements
for 5 ECTS as a Special Project

in the

Department of Chemical and Biochemical Engineering

June 28, 2019

Declaration

All sentences or passages quoted in this document from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure.

Name: Atli Freyr Magnússon

Signature: *Atli Freyr*

Date: 28.06.2019

Name: Binoy Shah

Signature: *Binoy Shah*

Date: 28.06.2019

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 6 |
| 1.1 | Problem Statement and Proposed Solution | 7 |
| 1.1.1 | Problem Statement | 7 |
| 1.1.2 | Proposed Solution | 8 |
| 1.1.3 | Report Structure | 8 |
| 2 | Data Gathering | 9 |
| 2.1 | Experimental data selection | 9 |
| 2.2 | Salts and properties considered | 10 |
| 2.3 | Findings during Literature review | 10 |
| 2.3.1 | The Original Paper Mystery | 11 |
| 3 | Technical Aspects and User guide | 12 |
| 3.1 | Introduction to Parser | 13 |
| 3.1.1 | Adding data to existing file | 14 |
| 3.1.2 | Creating new file | 15 |
| 3.2 | Introduction to API | 15 |
| 3.2.1 | Extracting raw viscosity data of $CaCl_2$ | 16 |
| 3.2.2 | Full regression and model building of $CaCl_2$ viscosity data | 16 |
| 3.2.3 | Analysis of a single $LiCl - KCl$ mixture with a molar ratio 20% molar percentage of $LiCl$ | 17 |
| 3.2.4 | Modelling density of $LiCl - KCl$ system for different mixing ratios . . | 17 |
| 4 | Conclusions and future work | 20 |
| | Appendices | 22 |

List of Figures

| | | |
|---|--|----|
| 1 | Example of an asdf file structure, with a) depicting a dataset structure and b) depicting the metadata | 13 |
| 2 | Example Excel file with correct structure and syntax | 14 |
| 3 | Showcasing where to put Excel and Reference Files | 15 |
| 4 | Snapshot of the program source code showcasing where to write new commands | 15 |
| 5 | Example Excel output when asking for viscosity data of molten $CaCl_2$ | 16 |
| 6 | Results of Regression and Model fitting of $CaCl_2$ viscosity data | 17 |
| 7 | Example output when asking for density analysis of 20-80 Mixture of $LiCl - KCl$ | 18 |
| 8 | Surface plot of $LiCl - KCl$ Regression Kriging model predicting Density . . | 18 |
| 9 | Example output when asking for density analysis of $LiF - BeF_2$ | 19 |

List of Tables

| | | |
|----|---|----|
| 1 | Excel Format for Data Extraction | 10 |
| 2 | Reduction in Paper over layered search | 11 |
| 3 | List of 3rd party Python dependencies | 12 |
| 4 | List of physical properties currently in the database with expected associated unit and basis function to regress Temperature dependent data to parameters a, b and c | 14 |
| 5 | List of ooASDF methods | 14 |
| 6 | List of API functions and methods | 16 |
| A1 | List of salts and salt mixtures currently in the datasets as well as the total number of experimental data | 22 |

Abstract

Molten salts have been used for nuclear applications since the first half of the 20th century. Recent interests garnered into Molten Salt Reactors and Concentrated Solar Power applications has resulted in renewed focus on Molten salt, justified by their properties. However, experimental data on important thermo-physical properties is sparse and scattered over 4 decades of publications. This creates difficulty in advancing the research related to applications of Molten Salts. The aim of this project was to initiate development of an open source molten salt database library. Experimental data of multiple salts is gathered via Literature sweep and matrix search. Obtained data is converted to a newly developed Advanced Scientific Data Format (ASDF) and published on <https://github.com/openmsr/msdl>. Two computational tools were developed to supplement the project, a Parser that can easily convert raw experimental data into the desired data structure to make external contribution easier. Also developed was an Application Programming Interface (API) to aid with extracting, filtering and data filtering for salt mixtures of interest. By the end of the timeline of this project 133 data sets containing values of 7 different properties were obtained, converted to ASDF and added to the data library.

1 Introduction

Molten salts generally describe a liquid obtained via fusion of one or more inorganic salts. Certain thermo-physical properties present in these salts make them attractive in scientific fields. Properties such as stability at high temperature (>200 .C for high temperature applications) , low vapor pressure, high heat capacity, easy solubility of multiple compounds ; are looked into when molten salts are considered.

In the second half of 20th century, molten salts were being considered for Nuclear Applications. In the same period socio-political landscape resulted in a fairly large amount of work in short time. Aircraft Nuclear Propulsion (ANP) and Nuclear Energy for the Propulsion of Aircraft (NEPA) program by US Air Force were oriented towards using Nuclear technology for sustained air surveillance and as a viable nuclear strategic deterrent to Soviet capabilities. Oak Ridge National Laboratory (ORNL) was involved in research and also made the first Molten Salt Reactor (MSR) as part of the Aircraft Reactor Experiment. A fluoride salt of composition, $\text{NaF-ZrF}_4\text{-UF}_4$ (53-41-6) was used with beryllium oxide as a moderator. In 1961, the program was cancelled owing to extremely high cost of the project, without any viable results. The report stated that ***15 years and about \$1 billion have been devoted to the attempted development of a nuclear-powered aircraft; but the possibility of achieving a militarily useful aircraft in the foreseeable future is still very remote.*** [3]

However, this project confirmed the possible use of MSR for nuclear, and possibly civilian use. During the Molten-Salt Reactor Experiment (MSRE) which ran from 1960-1969, a 7.4 MW operational reactor was built and rigorous tests were carried out. During its successful runtime of almost 18000 hours of critical runtime (operating at 80 % capacity) there were no major breakdowns, flaws or delays in form of downtime. Since the late 70s and towards the end of the 19th century. Nuclear technology was devoted to the conventional Uranium 235 based reactors. The events of Chernobyl and cold war arms race led to concerns amongst public and an outcry for discontinuation of commercial nuclear power plans. Latest studies have shown that out of currently used major sources of energy generation, nuclear is the safest. Under nominal circumstances, using nuclear energy for power generation would result in 0.07 death as compared with 24.62 via normal coal and 18.43 via oil [7]. The values are for production per TWh (Tera Watt hour). This clearly shows that the general perception on dangers of nuclear energy are short sighted and debatable. This is similar to the perception of climate change and global warming. One reason for this was discussed by George Marshall in his book [8]. Using the same argument, it can be said that dangers of nuclear technology are very real and detailed in peoples mind, while that related to Coal or other conventional are relatively abstract and distant. Hence on a relativistic scale, the fallacious argument that nuclear power is on the extreme end of danger is ingrained within a high percentage of general population.

This short term pervasive mentality and public perception of the nuclear energy as a hazard led to the nuclear power phase out through Europe. After the end of cold war, focus was kept more on renewable energy research and virtually no research as conducted on MSR technology. This brings us to the reasoning and motivation behind initiation of this project.

1.1 Problem Statement and Proposed Solution

Technologies for energy generation and energy storage are being evaluated in recent years. Molten salts are receiving special attention due to its aforementioned properties making it a suitable contender for heat transfer, coolant and heat storage substrate[9][11]. The important properties which are taken into consideration with renewed interest of molten salts are as follows.

- high volumetric heat capacity,
- high boiling point and low vapor pressure,
- no undesirable chemical exothermic reactions between different zones of energy plants and power cycle coolants (core, heatexchange loop),
- optical transparency during inspection operations,
- ability to dissolve actinides,
- great insensitivity to radiations

In recent years, there has been a renewed interest in developing MSR and its related technology. This push is partly because of the limitations seen in renewable energy and the need for a suitable sustainable alternative; and partly because of the perception change on nuclear energy and availability of funding to model and develop the technology. Technology like Concentrated Solar Plant (CSP) are have also gained traction, where integral storage is a component of the process. Molten salts are rigorously studied and their property such as volumetric absorption are deciding factors in the development of the same[12].

1.1.1 Problem Statement

Looking into these technologies, one thing gradually becomes clear. Molten salts will be used in a wide variety of applications within nuclear, solar and chemical industry. Thermo-physical and chemical property data for a wide variety of eutectic salt mixtures are necessary to study and evaluate further. However, a literature survey of experimental properties on molten salts found that the data was spread over 30 years in multiple papers. This gave a problem statement in the form, *while recently garnered interest in technologies have focus on study of molten salt properties, the dearth and difficulty in finding data in suitable format can create a possible hindrance for researchers intending to work on this subject*. This project is aimed at starting to solve this problem.

1.1.2 Proposed Solution

Solution Strategy was discussed with project supervisors who had relevant knowledge in the field and who were industrially established to give suitable advice. The proposed solution was *creation of an online data library which can act as a single major source for obtaining thermo-physical and chemical properties*. The data would on an open source platform in a suitable format for wide accessibility. The methodology which is used in creation of this library is in two fold step.

- Step-1 Literature sweep of papers in which experimental property of molten salts are given. The data from papers should be converted into an efficient sustainable data format with human readable meta-data where the source is listed.
- Step-2 Develop an Application Program Interface (API) that can filter and extract relevant data and perform rudimentary data processing for modelling purposes.

Final product was planned to have a robust programming interface to carry out analysis of various properties based on the library created as well as the possibility of predicting Molten Salt properties for modelling work. The main aim of this project is to help academic people who are starting to work on Molten Salts via single source to obtain and analyze their thermo-physical properties. Other advantages of this project are that due to It being open source, contribution to this project becomes easy and in a circular fashion, available to everyone.

1.1.3 Report Structure

The report structure is also kept in the two parts. First part , data gathering, as the names suggests describes the process through which experimental data relevant to molten salts was obtained and converted to excel files. An intermediate step of data extraction to excel files was used because a large proportion of papers with experimental data were quite old, making it difficult to convert them to API. The second part describes the technical aspects of the programming. It describes the features developed and the logic behind it. This section contains user guide as an explanation for new users on getting started.

2 Data Gathering

This section describes Step-1 of the library creation. It will describe the type of data gathered, properties and salts, which were initially considered. The reason behind the selection is outlined. Since experimental data is considered, the papers which show their results in their report are selected. Literature review of such papers revealed some interesting results, which are discussed. The amount of data gathered is also listed. Since this project can and should be continued in future, this section serves as guide on how to start the process.

2.1 Experimental data selection

During the high tide when suitable funding was available for research related to properties of molten salts, a period between 1950-1985, there were many experiments conducted and published by independent researchers as well as ORNL, which published sets of volumes containing both experimental as well as modelling values for molten salt properties. Some of the work was published in Idaho National Laboratory for U.S. Department of Energy and Office of Nuclear Energy. One of the main criterion for initial selection of data was availability of actual experimental values in tabular form. This significantly reduced the data available and gave a surprising inside about the work being carried out, the latter part will be discussed towards the end of this section.

To give an example [13], was published which contained the title as *Engineering Database of Liquid Salt Thermophysical and Thermochemical Properties*, they review properties of various salts and mention the results. However, experimental data is not mentioned here, instead correlations are mentioned, data for which is taken from literature. This material then is not as relevant for the library, because it does not have experimental values, but it does point in the right direction, via references to the source data.

Compared to that [1], is one of the first paper included in the library. This was the category of paper which were desirable and required for the library. It mentions the experimental process, characteristics of the material used, an most important for us, Experimental results of a property with its constituent composition and the Temperature range over which it is observed. Other information such as amount of experiments conducted at each Temp point, error propagation and uncertainty is also noted down, which is an advantage. Papers like this are hard to find but true to keep. This is shown by a simple search for citation in further studies. A single experimental paper is then used by multiple sources to model or analyze the property and work on further development of the model. After obtaining a few papers, a set format to extract the data into excel files was created. Tabula was used to extract data from the pdf files, since it would have been a tedious task manually. The standard format used in excel file is shown in table 1.

This streamlined the extraction process, then next step was selection of salts and properties which are readily available and which could be most helpful.

Table 1: *Excel Format for Data Extraction*

| Property | Description |
|-----------------|-----------------------------------|
| Components | Constituent Compounds |
| Composition | Molar Fraction of Salt in Mixture |
| Temperature | Taken in °C |
| Measurements | Value of physical property |
| Data Points | Number of experiments |
| Parameters | Regression Parameters Reported |
| Regression Type | Basis Function used in regression |
| Error | Deviation from Regression in % |
| Range (+ -) | |

2.2 Salts and properties considered

To start with, salts which were utilized frequently in MSR were analyzed. They were FLiBe, a mixture of lithium fluoride (LiF) and beryllium fluoride (BeF₂) and FLiNaK, a ternary eutectic alkaline metal fluoride salt mixture of LiF-NaF-KF. Fluoride salts were primarily researched for MSR, until it was discovered that Chlorides have more stability at high temperature. The salt considered there was LiClKCl mixture, which had data points over a range of temperature for multiple composition. Over the course of this project, 25 salts in various compositions were analyzed. The full list is found in Appendix A- TableA1.

Properties, which were vital for application of a molten salt and ones, which were experimented abundantly, were started with. These were Density, Viscosity and Thermal Conductivity. The full list of properties which were evaluated are shown in Table 4 in Section 3.

2.3 Findings during Literature review

At the start of the project a general sweep was done and the papers collected were analyzed. Three class of papers were obtained.

- Experimental results of a salt property in tabular form with each individual data set mentioned [Class 1] [20%]
- Mixture of Experimental and Fitted results in Graphical Format [Class-II] [40%]
- Results from correlations developed on either experimental or modelling work of other researchers. [Class- III] [40%]

Class I papers are ideally required for the library. They are also required to ensure that the program has enough actual data-points for regression and statistical modelling. However, only 20% of the total papers found met this criterion. All the papers currently in the library are Class-I papers.

In Class-II & Class-III papers, it was observed that experimental data was not available for extraction. When results are mentioned in graphical format, all the data sets are converted into a standard fit equation, back tracking them is not optimal as error propagation will increase. Papers wherein correlations are used to predict or improve a property also cannot be

used, directly. However, these papers cite other papers which can be back tracked to a source experimental paper. Looking through multiple Class-III papers, there was an interesting revelation.

2.3.1 The Original Paper Mystery

When searching in the vast matrix of review, experimental and correlation papers, review papers of basic properties are a good source for back tracking to original source. One such paper is [11]. It has over 200 reference papers, which are cited as source for the results in this paper. A natural back-tracking process was to single out the references cited multiple times , so that high volume of data can be obtained from one source file. 60 papers showed promise as a result which were then searched for via Science Direct and DTU find it. However, a surprising revelation was that most of the data in those papers, was modelled and the experimental values were taken from an older source which was cited. Another round of searching for experimental data led us in papers published in range 1960-1985. There was another issue here with the papers, a lot of them were missing on all searchable engines, some needed to be bought and some were restricted in access. A large portion of missing files were from ORNL quarterly publications, which are not available online because a virtual copy of them does not exist. This is being mitigated as people at ORNL are scanning the files and making them virtually available, however it is a slow process. With the availability of these Class-I files, a large amount of data should be gathered. This process and reduction in amount of papers is seen in Table 2.

Table 2: *Reduction in Paper over layered search*

| Papers | Search Layer | | | |
|----------------|------------------------------|-----------------------------|--|---------------------------|
| | 1 | 2 | 3 | 4 |
| Desired | 230- Files | 60-Usable Papers | 20-Obtained Files | 10-Extracted Files |
| | 150- Non-experimental Papers | 40- Non-Experimental Papers | 10- Not Available, Pay wall, Not found | |

This point needs to be kept in mind when the project is taken forward. Sweep search gives a large amount of quantitative results but did not yield relevant data. Instead doing a matrix back-through search with review papers is a technique recommended for future use. Towards the end of the report is a future work section, where other methods to further streamline the data gathering process is discussed. This concludes the data gathering section.

3 Technical Aspects and User guide

The data library is stored as a collection of Advanced Scientific Data Format (ASDF) files [4]. ASDF is a new interchange format for scientific data developed by the Space Telescope Science Institute primarily for astronomy and is still in active development. This format was chosen due to its hierarchical and human-readable metadata format and its native Python data types. This ensures the library remains usable despite version changes of computational tools used as well as easily storing the source of the data. The official data library and its associated tools are maintained by Copenhagen Atomics as an open source project on GitHub.

Following the data library is a Parser which allows users to add custom data into the data library while keeping the format correct which makes it easy for contributors to expand the data library. A secondary API script is also included which handles basic data processing of the library for researchers. Both of these tools are written in Python 3.6 and some basic knowledge of the language is required to utilize the tools. To aid with some specific functions some packages not included in the Python base library are used, shown in table 3. The most important feature of API is to easily extract all data that matches a requested physical property and component mixture and organizing the data into arrays that allow for further data analysis. A secondary feature of the API is to fit the data to a variety of basis function if applicable and build a mathematical model around the data.

The API currently uses Regression Kriging model as the surrogate model to fit and predict data. Kriging model has been used extensively in engineering to model and understand sparse data[2] which makes it a strong tool for the data library where we expect sparse datasets when it comes to different mixing ratios of the salts. The mathematical description of Kriging models is illustrated by Jones et al[6]. Ordinary Kriging is designed as an exact interpolation technique which may lead to high oscillating behaviour when dealing with datasets that are expected to be noisy. The problem is addressed by using the augmentation to Kriging where a regression component is added[5]. These models are created using the Python package pyKriging, which is an open source package maintained by Paulson and Ragkousis[10].

Table 3: *List of 3rd party Python dependencies*

| Name | Function |
|------------|--|
| asdf | Handles creation and manipulation of ASDF files |
| openpyxl | For extracting data from Excel Files |
| numpy | Numerical array object |
| matplotlib | Creating Figures |
| scipy | For the optimized functionality to fit regression parameters |
| pyKriging | For training of a Kriging Model |

To fully utilize all aspects of the program the dependencies listed in Table 3 must be installed, access to Microsoft Excel is required as well in the current version. Each file in the library contains the experimental results of a journal article, book etc. converted into a tree like structure which makes it easy for a python script to navigate and extract raw data for further data processing. The experimental data is saved as a data set,

where each data set contains temperature readings and the measurements of a single physical property of a salt mixture. It will also contain any optional information supplied by the authors such as regression, errors etc. The raw data itself can be extracted using Python's numpy package as asdf by default compresses the data for better efficiency. Each file also contains a metadata tree which contains creation and update times of the file itself as well as reference to the data written in a bibTeX format.

```

10 ~: /core/extension_metadata-1.0.0
11 extension_class: asdf.extension.BuiltinExtension
12 software: {name: asdf, version: 2.3.1}
13 dataSet1:
14   Components: [LiF, BeF2]
15   Composition: /core/ndarray-1.0.0
16   source: 0
17   datatype: float64
18   byteorder: little
19   shape: [2]
20   Data Points: [3, 3, 3, 3, 3, 3, 3, 4, 5, 5, 3, 3, 3, 3, 4, 3, 5, 3, 5, 3, 6, 3, 4,
21     3, 5, 3, 2]
22   Error Range: {+/-}: [0.9]
23   Measurements: /core/ndarray-1.0.0
24   source: 2
25   datatype: float64
26   byteorder: little
27   shape: [24]
28   Parameters: /core/ndarray-1.0.0
29   source: 3
30   datatype: float64
31   byteorder: little
32   shape: [2]
33   Property: [Viscosity]
34   Regression: [a*exp(b/(0.314*T))]
35   Temperature: /core/ndarray-1.0.0
36   source: 1
37   datatype: float64
38   byteorder: little
39   shape: [24]
40 dataSet2:
41   Components: [LiF, BeF2]
42   Composition: /core/ndarray-1.0.0
43   source: 0
44   datatype: float64
45   byteorder: little
46   shape: [2]
47   Data Points: [3, 3, 3, 3, 3, 3, 3, 4, 5, 5, 3, 3, 3, 3, 4, 3, 5, 3, 5, 3, 6, 3, 4,
48     3, 5, 3, 2]
49   Error Range: {+/-}: [0.9]
50   Measurements: /core/ndarray-1.0.0
51   source: 2
52   datatype: float64
53   byteorder: little
54   shape: [24]
55   Parameters: /core/ndarray-1.0.0
56   source: 3
57   datatype: float64
58   byteorder: little
59   shape: [2]
60   Property: [Viscosity]
61   Regression: [a*exp(b/(0.314*T))]
62   Temperature: /core/ndarray-1.0.0
63   source: 1
64   datatype: float64
65   byteorder: little
66   shape: [24]
67 dataSet3:
68   Components: [LiF, BeF2]
69   Composition: /core/ndarray-1.0.0
70   source: 0
71   datatype: float64
72   byteorder: little
73   shape: [2]
74   Data Points: [3, 3, 3, 3, 3, 3, 3, 4, 5, 5, 3, 3, 3, 3, 4, 3, 5, 3, 5, 3, 6, 3, 4,
75     3, 5, 3, 2]
76   Error Range: {+/-}: [0.9]
77   Measurements: /core/ndarray-1.0.0
78   source: 2
79   datatype: float64
80   byteorder: little
81   shape: [24]
82   Parameters: /core/ndarray-1.0.0
83   source: 3
84   datatype: float64
85   byteorder: little
86   shape: [2]
87   Property: [Viscosity]
88   Regression: [a*exp(b/(0.314*T))]
89   Temperature: /core/ndarray-1.0.0
90   source: 1
91   datatype: float64
92   byteorder: little
93   shape: [24]
94
95 metadata:
96   Created: '2019-06-25 17:54:56.410201', Updated: '2019-06-25 17:54:56.425276',
97   bibTex: "Bastille(Nissen1962a,\n title = (Thermophysical properties of the equimolar
98     mixture sodium nitrate-potassium nitrate from 300 to 600-degree-C),\n language
99     = (engl),\n journal = (Journal of Chemical and Engineering Data),\n volume =
100     (127),\n number = (3),\n pages = (248-273),\n year = (1962),\n isbn = (15205134,
101     00219469),\n doi = (10.1021/jo00038a012),\n author = (Nissen, Donald A.),\n",
102   fileName: Nissen1965.asdf, notes: '', numDatasets: 4)
103
104 ...

```

Figure 1: Example of an asdf file structure, with a) depicting a dataset structure and b) depicting the metadata

3.1 Introduction to Parser

The parser source code is contained in the file *ooASDF.py* located in the *Director* folder, it is written to scan excel file containing experimental data points as well as text files containing citation information and create asdf file with the proper tree structure and syntax as shown in figure 1. Before using the parser, it is necessary to prepare two files, an Excel file containing the raw data and optional information in a specific structure and a text file containing reference information in a bibTeX format. An example Excel file is shown in figure 2. Note that each asdf file is reserved for a single source, so if a single report or book contains multiple data sets for multiple salts or properties it is necessary separate them into new identical Excel sheets, the Parser will scan all Excel sheets and treat them all equally. It is important the data input for the excel files matches the units showcased in Table 4. The information with red lettering is a requirement to fill out, but anything in green is optional and any additional information the user wants to convert can be added by going down the rows.

The reference file can simply be a Text File that contains reference information in bib format as raw text. One of the primary features of the project is the ability to track all the experimental data, meaning this is a requirement when creating new files. If the reference information is missing the program will throw an error. These two Input files should then be placed in the *Input Files* folder as seen in figure 3 before running the program. Afterwards the Excel and reference text files will be stored in the *Converted Files* folder

Using the program can be done either by typing in the commands at the bottom of *ooASDF.py* as shown in figure 4 file or by importing the *ooASDF* class from *ooASDF.py*, It is important not to change the working directory when using the program and keep it the *Director* folder to ensure that the manipulated files end in *Library* folder. Below is a table

Figure 2: Example Excel file with correct structure and syntax

Table 4: List of physical properties currently in the database with expected associated unit and basis function to regress Temperature dependent data to parameters a, b and c .

| Property | Unit | Fitting Function |
|------------------------|---------------------------|------------------------|
| Temperature | $^{\circ}\text{C}$ | N/A |
| Density | g/cm^3 | $a + bT$ |
| Viscosity | $Pa\ s$ | $a * \exp(b/(8.314T))$ |
| Surface Tension | dyn/cm | $a + bT$ |
| Electrical Conductance | $1/(ohm\ cm)$ | $a + bT + cT^2$ |
| Heat Capacity | $cal/(K\ mol)$ | $a + bT + cT^{-2}$ |
| Vapor Pressure | $mmHg$ | $10^{a+b/T}$ |
| Thermal Conductivity | $10^4 * cal/(cm\ sec\ K)$ | $a + bT$ |

of the functions for the program and a couple of example usages.

Table 5: List of ooASDF methods

| Method | Usage |
|----------------------|--|
| ooASDF(fileName) | Constructor, opens up the specified file or creates one if it doesn't exist |
| .addData() | Scrapes the Excel File and collects the data |
| .importBib(fileName) | Imports the contents of fileName into a string to use as reference information |
| .finish() | Writes the collected data into an asdf file |

3.1.1 Adding data to existing file

If a file already exists in the library but data is missing and needs to be added, the reference bib file is not necessary. Filling up the Excel file akin to figure 2 and putting it in the *Input Files* should be done first. Then use the following three commands to add the data.

```
asdfFile = ooASDF('existingFile')
asdfFile.addData()
asdfFile.finish()
```

| | | | | | | |
|----------------|------------------|-----------|---------------|------------------|-----------------------|-------|
| API | 02-07-2019 12:44 | Filemappe | Router | 27-06-2019 16:47 | Microsoft Excel-re... | 10 KB |
| ConvertedFiles | 01-07-2019 20:08 | Filemappe | Reference.txt | 27-06-2019 14:12 | Tekstdokument | 2 KB |
| Director | 27-06-2019 12:46 | Filemappe | | | | |
| docs | 06-02-2019 16:44 | Filemappe | | | | |
| InputFiles | 01-07-2019 18:22 | Filemappe | | | | |
| Library | 01-07-2019 20:10 | Filemappe | | | | |
| LICENSE | 06-02-2019 16:44 | File | | | | 35 KB |
| README.md | 06-02-2019 16:44 | MD-File | | | | 1 KB |

Figure 3: Showcasing where to put Excel and Reference Files

```

88 def finish(self):
89     self.tree['metadata']['Updated'] = datetime.datetime.now()
90     os.chdir(self.libLoc)
91     af = asdf.AsdfFile(self.tree)
92
93     if self.tree['metadata']['bibTex'] == '':
94         raise Exception('bibTex Field is empty')
95
96     af.write_to(self.fileName)
97     print("ASDF File manipulation existed successfully")
98     return None
99
100
101
102
103 #Type Commands after this line
104
105 ''' Example Commands
106 newASDF = ooASDF('nsrds1979')
107 newASDF.addData() #Ensure that InputFiles folder has an excel file
108 newASDF.importBib('bibInfo.txt') #Ensure that there is a Reference file in InputFiles
109 newASDF.finish()
110 '''

```

Figure 4: Snapshot of the program source code showcasing where to write new commands

The first line opens up an existing datafile, the second line scans all the Excel sheets and adds the data to it and the third line updates the datafile with the scraped information and updates the metadata.

3.1.2 Creating new file

Creating a new datafile is very similar as updating an existing one, except now it is necessary to include the reference information along with the Excel file in *Input Files*

```

newFile = ooASDF('newFile')
newFile.addData()
newFile.importBib('bibInfo.txt')
newFile.finish()

```

This follows a very similar workflow as for adding data except now the first line creates the new datafile with the specified file name and the third line scans in the reference information and adds it to the metadata. Trying to build a new datafile without using `.importBib()` method will result in an error due to missing source.

3.2 Introduction to API

The API source file is *api.py* located in *API* folder. This part of the program handles all data extraction from the molten salt data library while also containing some data processing features such as regression, model fitting and visualization. Using the api to do dataprocessing is simply importing the *api* class or use the *api.py* source file similar to how to use the Parser and simply ask for a physical property and a salt mixture. Optionally it is possible to extract specific mixing ratios of the molten salt. After running the parser a new *output.xlsx* file is created which contains the filtered information found in the data library as well as regression information if applicable. Table

Table 6: *List of API functions and methods*

| Method | Usage |
|--|--|
| API(Property,Salt List, <i>Composition</i>) | Constructor to set up the object, the <i>Composition</i> argument is optinal |
| .scanLibrary() | Scans the entire library for matching salt mixture and physical property |
| .organizeData() | Organizes the data into X and Y arrays ready for data analysis |
| .regressData() | Regresses data of single mixtures to a basis function in table 4 |
| .buildModel() | Builds a mathematical model of the data, currently Regression Kriging model |
| .makePlot() | Visualizes the data in graphical format |
| .printExcelReport() | Creates the <i>output.xlsx</i> file with all the gathered data |
| .initializeData() | Combines scanLibrary(), organizeData() and printExcelReport() methods for pure data extraction |
| .intializeFull() | Combination of all above function for full data analysis of given salt mixture |

Below are some example usages on the program to extract relevant information from the library

3.2.1 Extracting raw viscosity data of $CaCl_2$

Simply extracting raw data from library containing only viscosity data of $CaCl_2$ can be done in two commands

```
CaCl2Dat = API('viscosity',['CaCl2'])
CaCl2Dat.initializeData()
```

Where first command creates the Python object and second command combines the API features of scanning, organizing and output into an excel file. This will result in an excel file depicted in

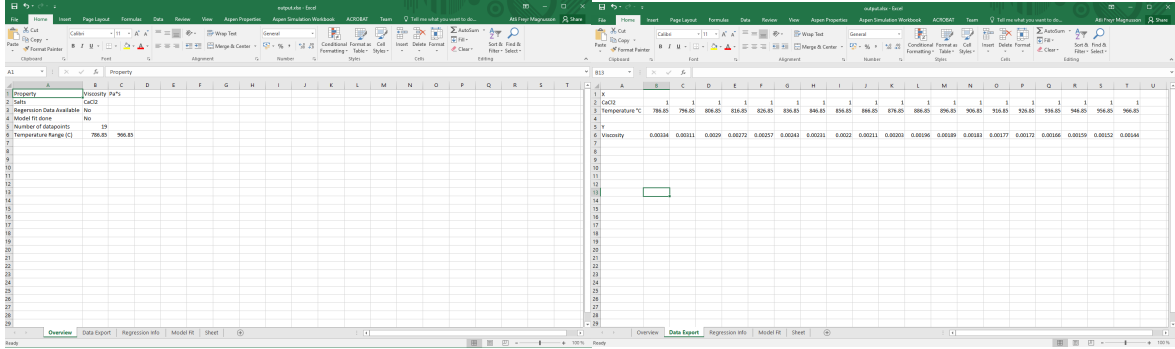


Figure 5: *Example Excel output when asking for viscosity data of molten $CaCl_2$.*

3.2.2 Full regression and model building of $CaCl_2$ viscosity data

To utilize all the data processing features implemented so far two commands are needed, similar to above commands except replace initializeData() with intializeFull().

```
CaCl20bj = API('viscosity',['CaCl2'])
CaCl20bj.intializeFull()
```

This also results in the Excel information depicted in figure 5, but now the Regression Info and Model Fit sheets also contain information shown in figure 6. The Regression Info contains

the basis function used, the fitted parameters and an illustration of the regression. The Model Fit only contains an illustration of the Kriging model due to it's complexity.

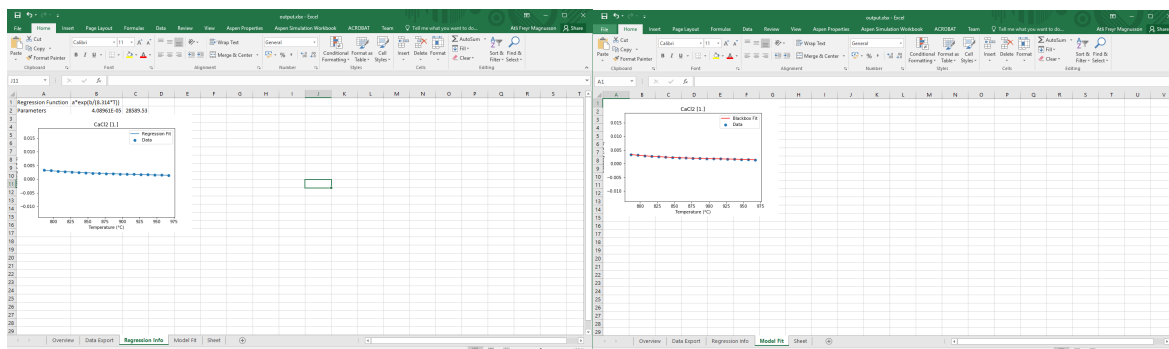


Figure 6: Results of Regression and Model fitting of CaCl_2 viscosity data

The Regression Kriging model can still be used to predict new data points. The regression model is stored in the variable *fitModel* in the Python object and has the *.predict()* method which accepts a Temperature coordinate and outputs a prediction. A full command example:

```
CaCl2Dat.fitModel.predict([862.52])
>>0.002147126842301393
```

3.2.3 Analysis of a single $\text{LiCl} - \text{KCl}$ mixture with a molar ratio 20% molar percentage of LiCl

When dealing with a salt mixture that in the library holds temperature dependent data for multiple mixing ratios like for the $\text{LiCl} - \text{KCl}$ system, it is possible to specify the mixing ratio as desired with an optional mixing ratio input to the API() command.

```
binaryMixture = API('density',['LiCl','KCl'],[20,80])
binaryMixture.initializeFull()
```

This will result in a similar *output.xlsx* as for a single salt system depicted in figures 5 and 6. The *fitModel* feature works under the same condition as above. Since only temperature dependence is analysed the visualization is a 2D plot exploring temperature dependence as shown in figure 7.

3.2.4 Modelling density of $\text{LiCl} - \text{KCl}$ system for different mixing ratios

Alternatively it is possible to extract the data and get a model fit for the entire system at all mixing ratios. In this case there is no regression done as now the program also analyses effect of mixing and currently we lack a basis function that accounts for both temperature and mixing dependencies. Thus only fitting done is the Regression Kriging model. To run this analysis for a salt mixture that contains data for multiple mixing ratios simply skip the mixing ratio parameter in the API() call.

```
binaryMixture = API('density',['LiCl','KCl'])
binaryMixture.initializeFull()
```

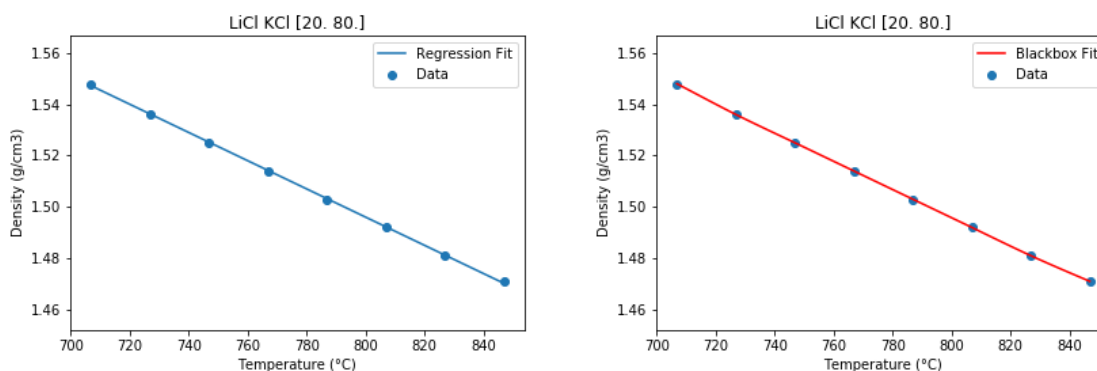


Figure 7: Example output when asking for density analysis of 20-80 Mixture of *LiCl* – *KCl*

Since this is a binary mixture we can depict the fit model as a 3D surface plot depicted in figure 8. This is done automatically for all binary mixtures, however no illustration is made for ternary or higher dimensional mixtures.

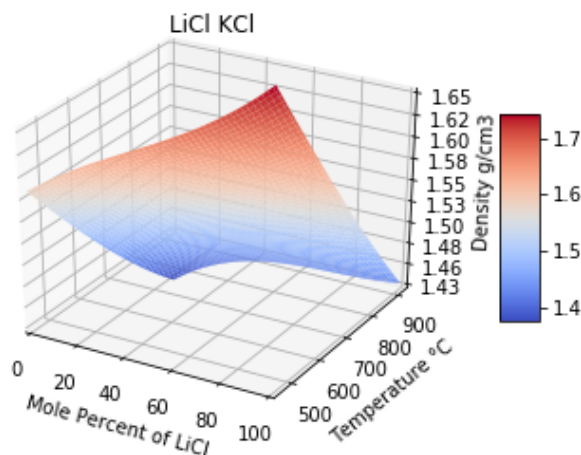


Figure 8: Surface plot of *LiCl*–*KCl* Regression Kriging model predicting Density

Using the fitted model is done with a similar syntax as for a single salt case but this case a mixing coordinate also has to be passed into the `.predict()` method. For a binary case simply put the mole percentage of the first case and then Temperature in Celsius.

```
binaryMixture.fitModel.predict([62.3,752.4])
>>1.5096203630922684
```

If the mixture contains more salts, then to use the prediction function it is necessary to pass the entire mixing coordinate and then supply the temperature

```
ternaryMixture = API('viscosity',['lif','bef2','thf4'])
ternaryMixture.initializeFull()
ternaryMixture.fitModel.predict([0.62,0.34,0.04,895.5])
>>0.8954644016700709
```

Note: Most salt systems currently in the library only contain a single mixture, in this case data analysis reverts to a single mixture as in 3.2.3 automatically, even if no mixing ratio was specified.

Example when asking for data on a $LiF - BeF_2$ mixture

```
binaryMixture = API('density',['Lif','BeF2'])
binaryMixture.initializeFull()
```

This will results in illustrations depicted in figure 9.

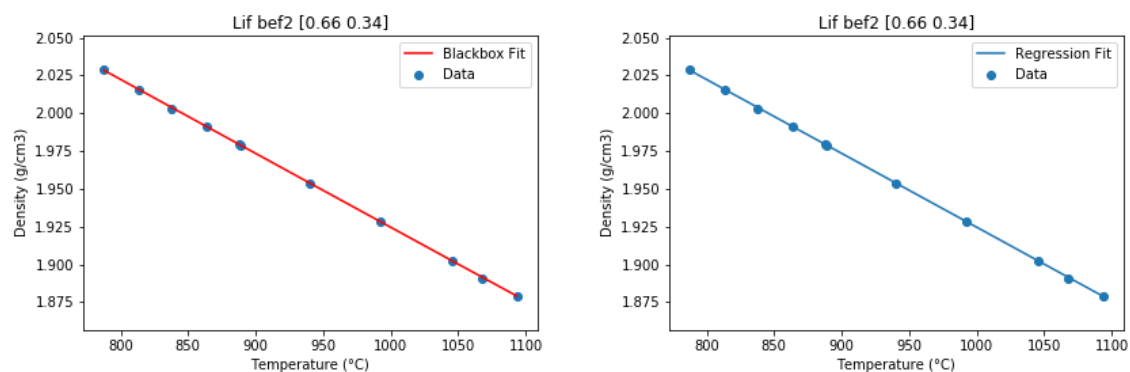


Figure 9: Example output when asking for density analysis of $LiF - BeF_2$

4 Conclusions and future work

In this project we started building the infrastructure for a molten salt data library. We have created a base standard for the data format and provided some tools to easily convert experimental results into the specified format as well as an easy to use tool for simple data analysis. This is however only the beginning and a lot of work is still needed before this project can be used in large scale projects in the likes of reactor design. For future work on the project we suggest to address the following issues:

1. **Expand the data library**

Over 100 experimental results have been converted to the proper data format and are ready to be analyzed by the tools, this only scratches the surface on how much data exists in literature. To represent a full fledged data library this needs to be expanded to encompass all important salt mixtures in the literature which could mean over a thousand sets. However the parser constructed for the project can easily add the data to the library thus expanding the data library is all about continuing the search through the scientific literature this project started. Obtaining the data from ORNL should be focused upon, as experimental data is very valuable. Another feature could be inclusion of Class-II and Class-III papers, when we have a robust Class-I data set available. The data library syntax will have to be updated on that.

2. **Eliminate dependencies**

The Parser only accepts Microsoft Excel files for the input and the API only prints the output information in a Microsoft Excel format. This was done for convenience when starting the data collection and processing. As this project grows in scope however it's important that other alternatives are presented. This is to follow the design philosophy that the data library is accessible by everyone on multiple platforms.

3. **Improve the mathematical modelling** Currently the program only supports one mathematical model when it comes to predicting physical properties of new salt mixtures which is the Regression Kriging model. This model is currently used while the dataset is still small and sparse but it will run into computational issues for larger data sets and is a blind blackbox model. The library of mathematical models to choose from should be expanded to allow for greater degree of data analysis. This could range from having alternative surrogate models such as Support Vector Regression or building a mechanistic model more grounded in the science of molten salts.

4. **Improve user friendliness**

Using the data library or any of the tools requires some basic knowledge of Python programming. However many of the fields experts may not be familiar with it or have time to learn the language. To make the library more accessible it may be necessary to develop a Graphical User Interface for the data library that retains all the features already implemented.

Of course while tackling the aforementioned issues, all bugs reported should be fixed as soon as possible and code should be readily maintained to ensure that the project remains in a usable state for the foreseeable future.

References

- [1] Y. Abe, O. Kosugiyama, and A. Nagashima. Viscosity of lif-bef//2 eutectic mixture (x//b//e//f//2 equals 0. 328) and lif single salt at elevated temperatures. *Journal of Nuclear Materials*, 99(2):173–183, 1981.
- [2] A. Forrester, A. Sobester, and A. Keane. *Engineering design via surrogate modelling* .: Wiley, 2008.
- [3] H. Frank. *Race to Oblivion*. Simon and Schuster, 1970.
- [4] P. Greenfield, M. Droettboom, and E. Bray. Asdf: A new data format for astronomy. *Astronomy and Computing*, 12:240–251, 2015.
- [5] T. Hengl, G. B. Heuvelink, and D. G. Rossiter. About regression-kriging: From equations to case studies. *Computers Geosciences*, 33(10):1301 – 1315, 2007. Spatial Analysis.
- [6] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, Dec 2001.
- [7] A. Markandya and P. Wilkinson. Electricity generation and health. *Lancet*, 370(9591):979–990, 2007.
- [8] G. Marshall. Don’t even think about it: Why our brains are wired to ignore climate change. 2014.
- [9] V. M. Nunes, C. S. Queirs, M. J. Loureno, F. J. Santos, and C. A. Nieto de Castro. Molten salts as engineering fluids a review: Part i. molten alkali nitrates. *Applied Energy*, 183:603–611, 2016.
- [10] C. Paulson and G. Ragkousis. pykriging: A python kriging toolkit, July 2015.
- [11] R. Serrano-pe3, J. Fradera, and S. Cuesta-Lpez. Molten salts database for energy applications. *Chemical Engineering and Processing - Process Intensification*, 73:87–102, 2013.
- [12] A. H. Slocum, D. S. Codd, J. Buongiorno, C. Forsberg, T. McKrell, J. C. Nave, C. N. Papanicolas, A. Ghobeity, C. J. Noone, S. Passerini, F. Rojas, and A. Mitsos. Concentrated solar power on demand. *Solar Energy*, 85(7):1519–1529, 2011.
- [13] M. S. Sohal, M. A. Ebner, P. Sabharwall, and P. Sharpe. Engineering database of liquid salt thermophysical and thermochemical properties.

Appendices

Appendix A

Table A1: List of salts and salt mixtures currently in the datasets as well as the total number of experimental data

| Salt Mixture | Number of Datasets |
|------------------------------|--------------------|
| BeF_2 | 1 |
| $NaNO_3 - KNO_3$ | 4 |
| KNO_3 | 7 |
| $MgCl_2$ | 6 |
| $CaCl_2$ | 7 |
| $LiCl - KCl$ | 37 |
| $AlCl_3$ | 1 |
| LiF | 4 |
| Li_2CO_3 | 6 |
| Na_2CO_3 | 6 |
| K_2CO_3 | 5 |
| $LiNO_3$ | 5 |
| $NaNO_3$ | 6 |
| Li_2SO_4 | 4 |
| Na_2SO_4 | 9 |
| K_2SO_4 | 1 |
| $LiBr$ | 5 |
| LiI | 5 |
| $LiF - BeF_2 - ThF_4$ | 5 |
| $LiF - BeF_2$ | 1 |
| $LiF - BeF_2 - ZrF_4$ | 1 |
| $LiF - BeF_2 - ZrF_4 - UF_4$ | 1 |
| $NaBF_4 - NaF$ | 2 |
| $KNO_3 - NaNO_2 - NaNO_3$ | 3 |
| $NaNO_2$ | 1 |
| Total | 133 |