Enhancing Research Workflows with a Python-based Automation for Data Management

Author: Kai Spicer **Mentors**: Matt Asmussen, Jacob Anderson

Background

Purpose of Research

- The radiological materials group is focused on understanding and developing treatment plans for the variety of wastes that are present and will be generated at the Hanford site.
- Over 30 years of plutonium production the Hanford Site generated vast quantities of radioactive waste, with more than 56 million gallons currently stored in underground tanks. The wastes are chemically complex and will require a combination of retrieval, pre-treatment and treatment processes for disposal.
- The team that I am a part of is focused on understanding the long-term chemical properties and interactions that take place within these tanks and how the eventual waste forms will behave upon disposal.

eSTOMP

- One of the main tools that the team uses to model these long-term interactions is eSTOMP (exascale-scale Subsurface Transport Over Multiple Phases).
- This software was developed at PNNL in the early 2000s and allows us to create complex and detailed simulations of how different chemical reactions and grout forms interact with the environment.
- While this software is very powerful it also requires excessive computational power to run. This means we must run it via High Performance Computing (HPC). The HPC tools are hosted by the Research Computing team here at PNNL.
- To run this complex simulation on a HPC, the user must create an input card that specifies everything from the physical condition of the waste form, to the physical properties of the waste form.

```
~Rock/Soil Zonation Card
formatted zonation file,grout.zon,
HD_BACKF,
LD_BACKF,
W1,
~Mechanical Properties Card
HD_BACKF,2.710,g/cm^3,0.35,0.35,,,Constant       ,1,
LD_BACKF,2.710,g/cm^3,0.37,0.37,,,Constant ,1,
W1,2.820,g/cm^3,0.557,0.557,,,Constant ,1,
~Hydraulic Properties Card
HD_BACKF,4.91e-03,hc cm/s,4.91e-03,hc cm/s,4.91e-03,hc cm/s,
LD_BACKF,1.86e-02,hc cm/s,1.86e-02,hc cm/s,1.86e-02,hc cm/s,
W1,1.54E-09,hc cm/s,1.54E-09,hc cm/s,1.54E-09,hc cm/s,
~Saturation Function Card
HD_BACKF,van Genuchten,6.50e-02,1/cm,1.7,0.086,,
LD_BACKF,van Genuchten,5.70e-02,1/cm,2.8,0.081,,
W1,van Genuchten,6.03E-06,1/cm,1.649,0.108,,
~Aqueous Relative Permeability Card
HD_BACKF,Mualem,,,
LD_BACKF,Mualem,,,
W1,Mualem,,,
~Solute/Porous Media Interactions Card
HD_BACKF,0.0,cm, 0.,cm,
LD_BACKF,0.0,cm, 0.,cm,
W1,0.0,cm, 0.,cm,
~Aqueous Species Card
1,Conventional,0.315,cm^2/yr,Constant,1.0,
CO2(aq),
~Solid Species Card
4,
calcite,2.710,g/cm^3,100.0873,g/mol,
solution,1.7,g/cm^3,18.01534,g/mol,
CO2_product,1.7,g/cm^3,44.009,g/mol,
solution_product,1.7,g/cm^3,44.009,g/mol,
~Conservation Equations Card
3,
Total_calcite,3,calcite,1,solution_product,1,CO2_product,1,
Total_CO2(aq),2,CO2(aq),1,CO2_product,1,
Total_solution,2,solution,1,solution_product,1,
~Kinetic Equations Card
2,
Kinetic_CO2_product,1,CO2_product,1,
   1,KnRc-1,   1.0000,
Kinetic_solution_product,1,solution_product,1,
   1,KnRc-2,   1.0000,
```

Figure 1: An example portion of an eSTOMP input card. Any line that starts with a ~ denotes the start of a new "sub-card". Each comma and new line represents a new piece of input data.

- Currently, to create these input cards users would copy a card they have previously created and change certain properties to simulate what they were interested in this time. This process is not only time consuming for the user but also requires a very strong understanding of the syntax required for these input cards.

Design

Development Goals

- To make this process more efficient for users, I set out to develop a script that would automate the process of creating new input cards, allowing the users to choose what values to change and create multiple new cards at once.
- I also worked with the NNLEMS team to connect their newly created Grout Database to my script allowing users to select grout properties from different research papers that have been saved within that database and have these properties automatically be placed at the correct location in the input card.

Data Flow Diagram

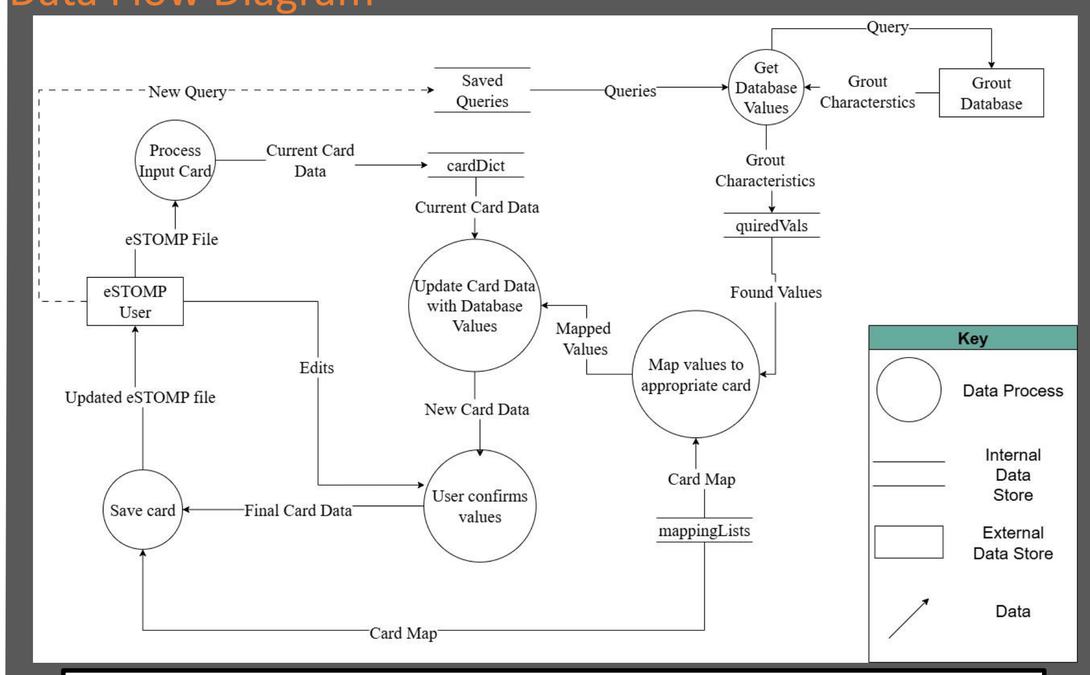


Figure 2: A high-level description of how data is moved and transported during my script. Below the image is a table that gives a brief description of the processes.

Table 1 : A short description of how each process shown in Figure 2 manipulates the data.

Process Name	Process Description
Process Input Card	• Reads in the users input card file and splits it into its sub-cards. • Saves these sub-cards in a dictionary with the card name as the key and the paragraph as the value
Get Database Values	• Queries the database for any relevant values and returns what it found as a dictionary.
Map Values to Appropriate Card	• Goes through all the possible variables that could be in each card which is saved in mappingLists and saves what card(s) each database value goes to.
Update Card Data with Database Values	• Turns each of the cards found above into a data structure that can be edited • Finds the correct value to replace in that data structure and updates that value
User Confirms Values	• User can "Zoom in" on any card they are interested in and will see what each value represents and can change those values
Save Card	• Any card that was edited is turned back from its data structure form into the syntax required for the eSTOMP card and all the cards are saved and recombined into a .txt file

User Experience

- The images below show what the GUI looks like for the user.

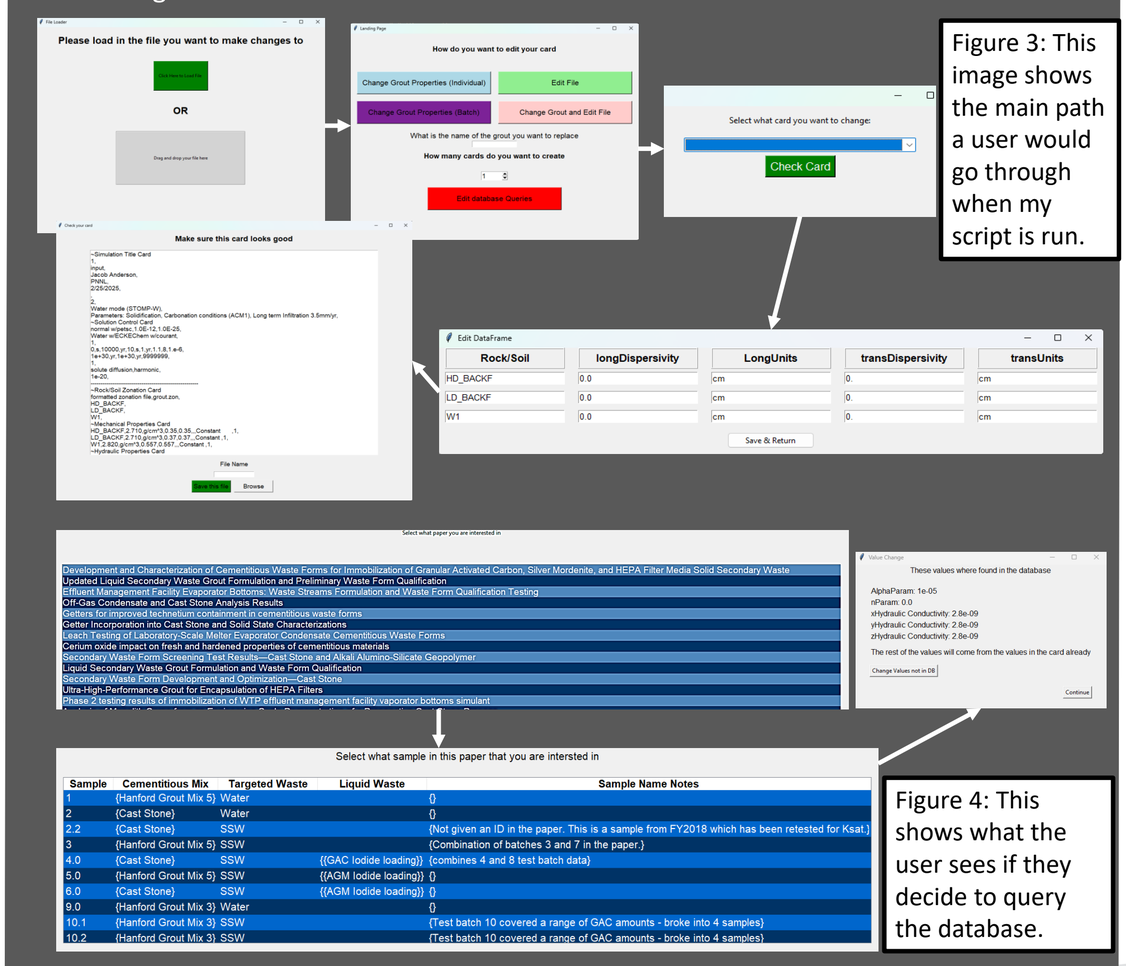


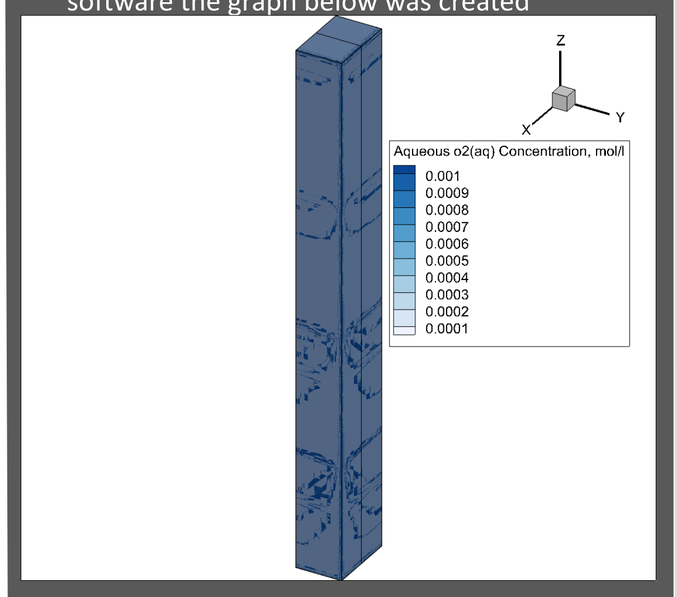
Figure 3: This image shows the main path a user would go through when my script is run.

Figure 4: This shows what the user sees if they decide to query the database.

Results

Validation

- To ensure that my script successfully transforms the data and recreates the cards according to the specification required by eSTOMP I had to validate the input cards that were returned when my script finished.
- To simulate a real use case of this script I decided to take a pre-made input card that had been created for a previous simulation that ran successfully and run it through my script changing the grout properties with values that were found within the database.
- After creating my input card I ran the eSTOMP simulation, the simulation returns a variety of output files that require further postprocessing to visualize.
- Utilizing postprocessing scripts and a plotting software the graph below was created



This image was generated using Tecplot

Figure 5: A 3-D model of the O_2 concentration inside of a simplified waste container

- This graph serves as validation that my script successfully executed the simulation, producing output that is both visualizable and ready for analysis by modelers, enabling the extraction of valuable insights.

Conclusions

- Overall, I was able to create a script that takes an eSTOMP input card, connects it to a database of grout properties and gives a user the ability to edit values through a GUI.
- The functionality of my script was validated by taking one of its generated cards and running it through an eSTOMP simulation run to generate a graph.
- This script allows researchers to run simulations much more efficiently saving projects both time and money.
- With the ability to run more scripts researchers are also able to run a wider variety of tests which creates models with greater variation and gives researchers a deeper understanding of their system.

Acknowledgments

- This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internship Program (SULI).
- My sincere thanks to Jacob Anderson and Matt Asmussen for their support and mentorship, which this project would not have been possible without. Lastly a special thanks to the NNLEMS database team for giving me access to the newly developed grout database and helping me find the data I needed.

For additional information, contact:

Kai Spicer

(509)-375-6117

Kailer.Spicer@PNL.gov

Pacific Northwest NATIONAL LABORATORY

www.pnnl.gov