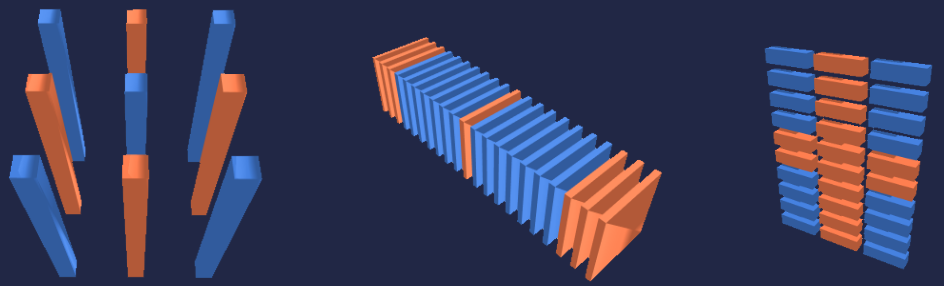


Interface to Accelerate MCNP Reactor Simulations

Duncan Beauch, CORE Student

August 3rd, 2023

Disclaimer

Certain commercial equipment, instruments, or materials are identified in this study in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

Physics

CS

VASST

About me

Why simulate a reactor?

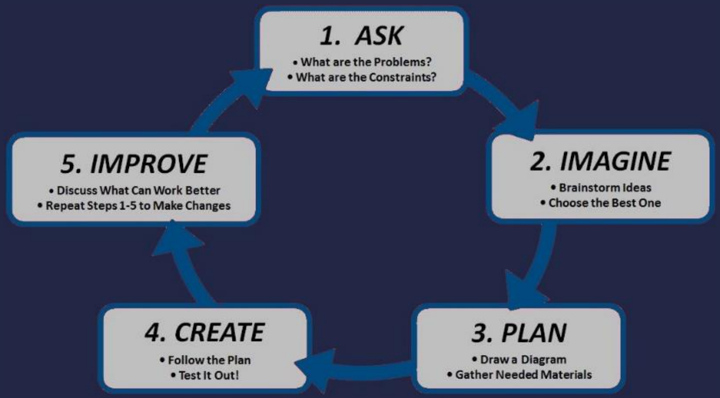
Safety

- Predictive Capabilities
- Stress Testing
- Radiation Detection
- Accident Prevention
- Regulatory Requirements

U.S.NRC
United States Nuclear Regulatory Commission
Protecting People and the Environment

Design

- Virtual Prototyping
- Design Optimization
- Burnup Analysis
- Low cost
- Scalability



MCNP Simulation

Monte **C**arlo **N**-**P**article Transport Code

- Developed at Los Alamos National Lab
- Standard for most nuclear facilities

Process:

1. Define geometry
   - Basic example: Book
2. Create particle sources
3. Simulate interactions
   - Based on experimental measurements
4. Particle tallying
   - Neutron flux and criticality ***k***

<u>Try building a library full of books: very tedious!</u>

Cell boundaries

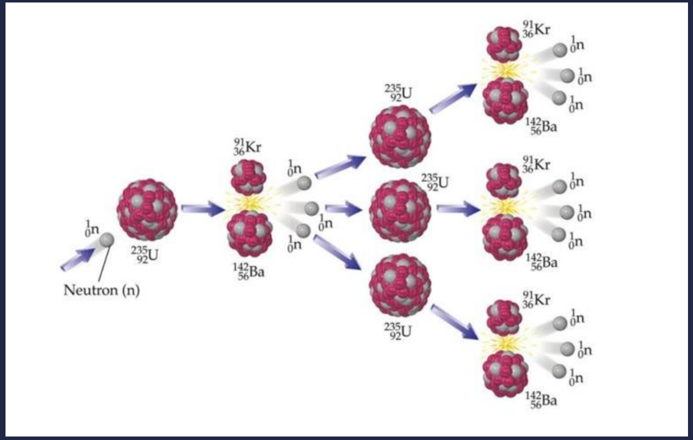
Cell number

Cell material

```
1 1 -4 1 -2 3 -4 5 -6  $ Cell 1 - Book

1 PX -1   $ Left Face
2 PX 1    $ Right Face
3 PY -1   $ Front Face
4 PY 1    $ Back Face
5 PZ -1   $ Bottom Face
6 PZ 1    $ Top Face

m1 6012 -6.0      $ Carbon
   1001 -10.0     $ Hydrogen
   8016 -5.0      $ Oxygen
```



NIST Neutron Source (NNS)

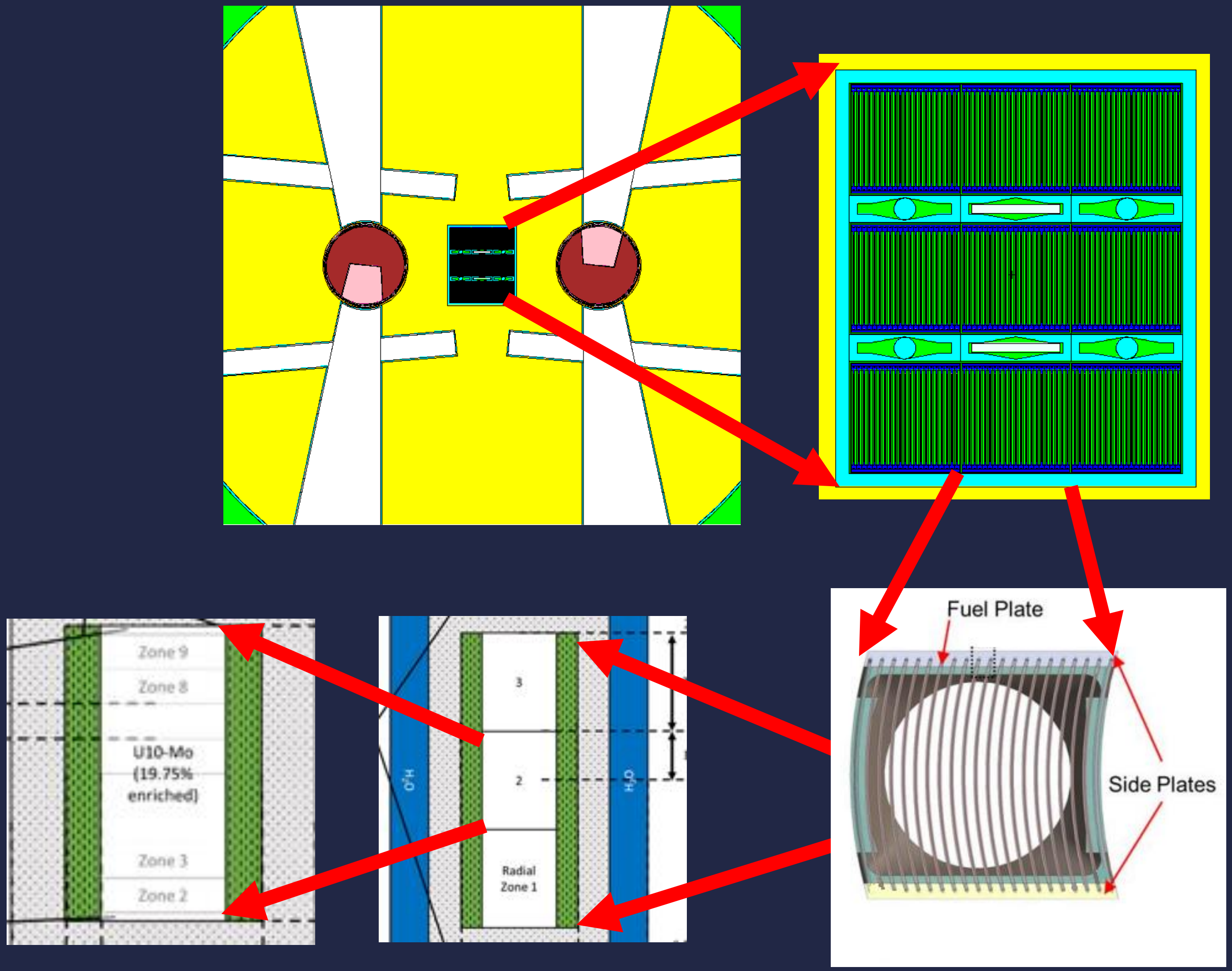
- Core design to replace current reactor (NBSR)

MCNP Input File Totals

- **2000+ Cells**
- **500+ Surfaces**
- **1500+ Materials**
- **150,000+ lines of code**

Problem: Change the material composition of a fuel plate

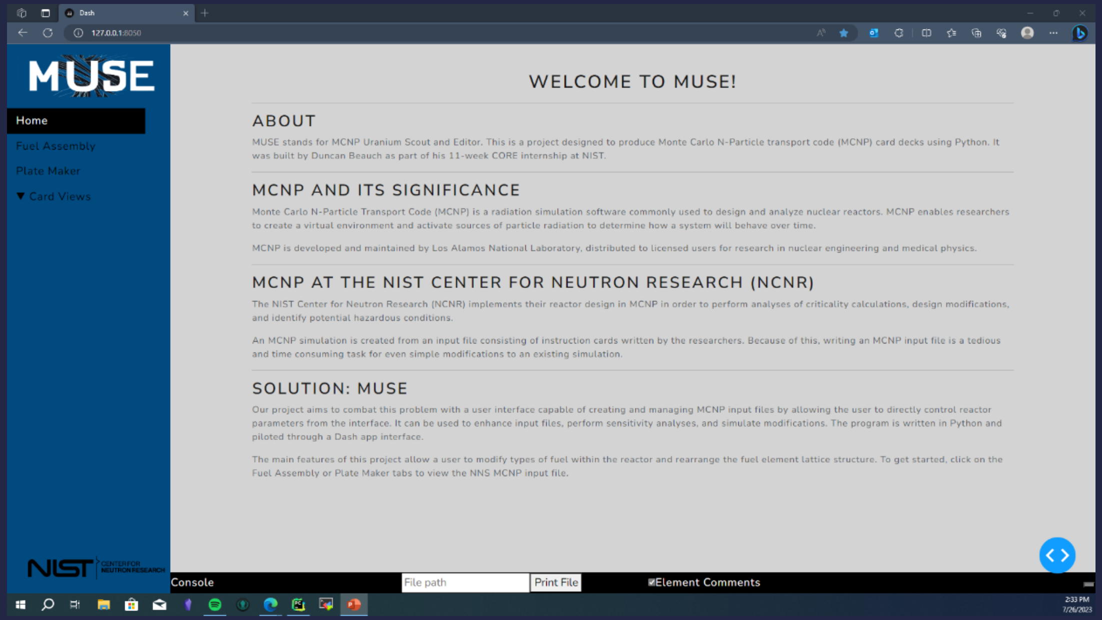
- **9 Fuel Assemblies**
- **21 Plates per Assembly**
- **30 Sections per Plate**



Project: MUSE

MCNP Uranium Scout and Editor

- Python backend
- Dash front end
  - Web application framework
  - Implements React JavaScript for UI and Flask for web server operations
- Locally hosted at runtime



Features

- Select and edit fuel assemblies or fuel plates
- Create new fuel plates from known materials
- Print new input file

Demo: Helium Core

Dash

127.0.0.1:8050

MUSE

Home

Fuel Assembly

Plate Maker

Legacy Assembly

▼ Card Views

Cell Cards

Surface Cards

Material Cards

Option Cards

Universes

WELCOME TO MUSE!

ABOUT

MUSE stands for MCNP Uranium Scout and Editor. This is a project designed to produce Monte Carlo N-Particle transport code (MCNP) card decks using Python. It was built by Duncan Beauch as part of his 11-week CORE internship at NIST.

MCNP AND ITS SIGNIFICANCE

Monte Carlo N-Particle Transport Code (MCNP) is a radiation simulation software commonly used to design and analyze nuclear reactors. MCNP enables researchers to create a virtual environment and activate sources of particle radiation to determine how a system will behave over time.

MCNP is developed and maintained by Los Alamos National Laboratory, distributed to licensed users for research in nuclear engineering and medical physics.

MCNP AT THE NIST CENTER FOR NEUTRON RESEARCH (NCNR)

The NIST Center for Neutron Research (NCNR) implements their reactor design in MCNP in order to perform analyses of criticality calculations, design modifications, and identify potential hazardous conditions.

An MCNP simulation is created from an input file consisting of instruction cards written by the researchers. Because of this, writing an MCNP input file is a tedious and time consuming task for even simple modifications to an existing simulation.

SOLUTION: MUSE

Our project aims to combat this problem with a user interface capable of creating and managing MCNP input files by allowing the user to directly control reactor parameters from the interface. It can be used to enhance input files, perform sensitivity analyses, and simulate modifications. The program is written in Python and piloted through a Dash app interface.

The main features of this project allow a user to modify types of fuel within the reactor and rearrange the fuel element lattice structure. To get started, click on the Fuel Assembly or Plate Maker tabs to view the NNS MCNP input file.

NIST CENTER FOR NEUTRON RESEARCH

Console | File path | Print File | ☑ Element Comments

11:03 AM
7/28/2023

How MUSE Works

Code Flow

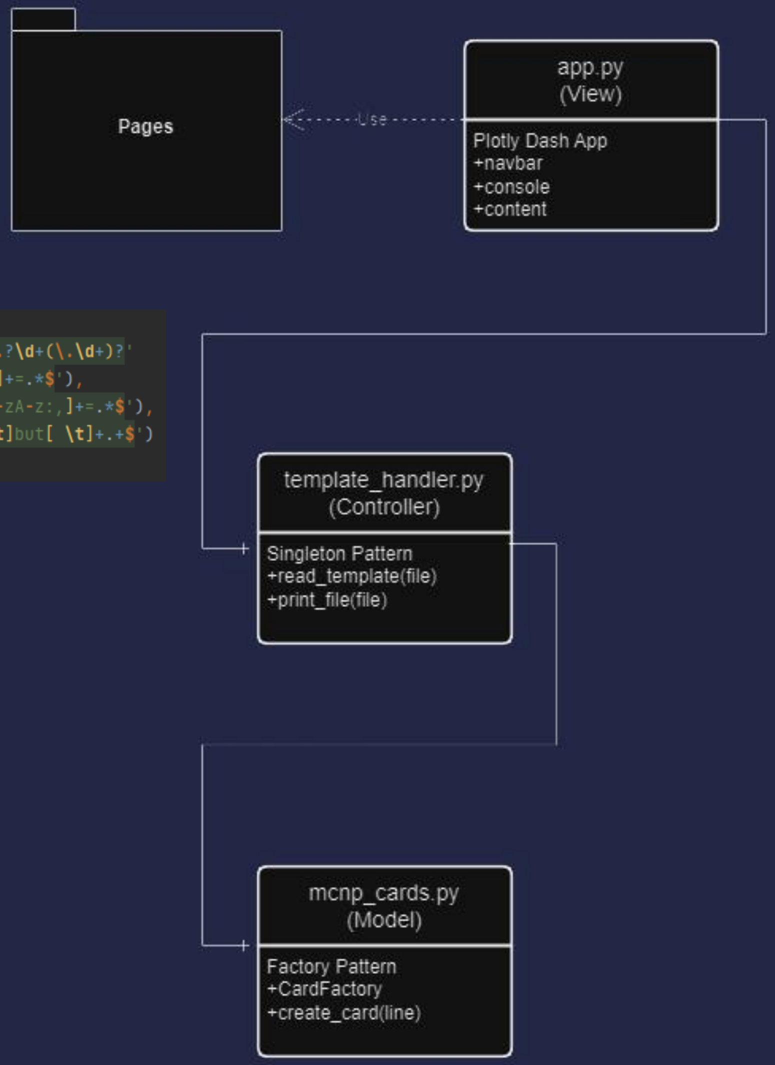
1. Parse input file (template)
2. Translate all MCNP lines into data structures with Regular Expressions
3. Create Fuel Assemblies
4. Display app to user

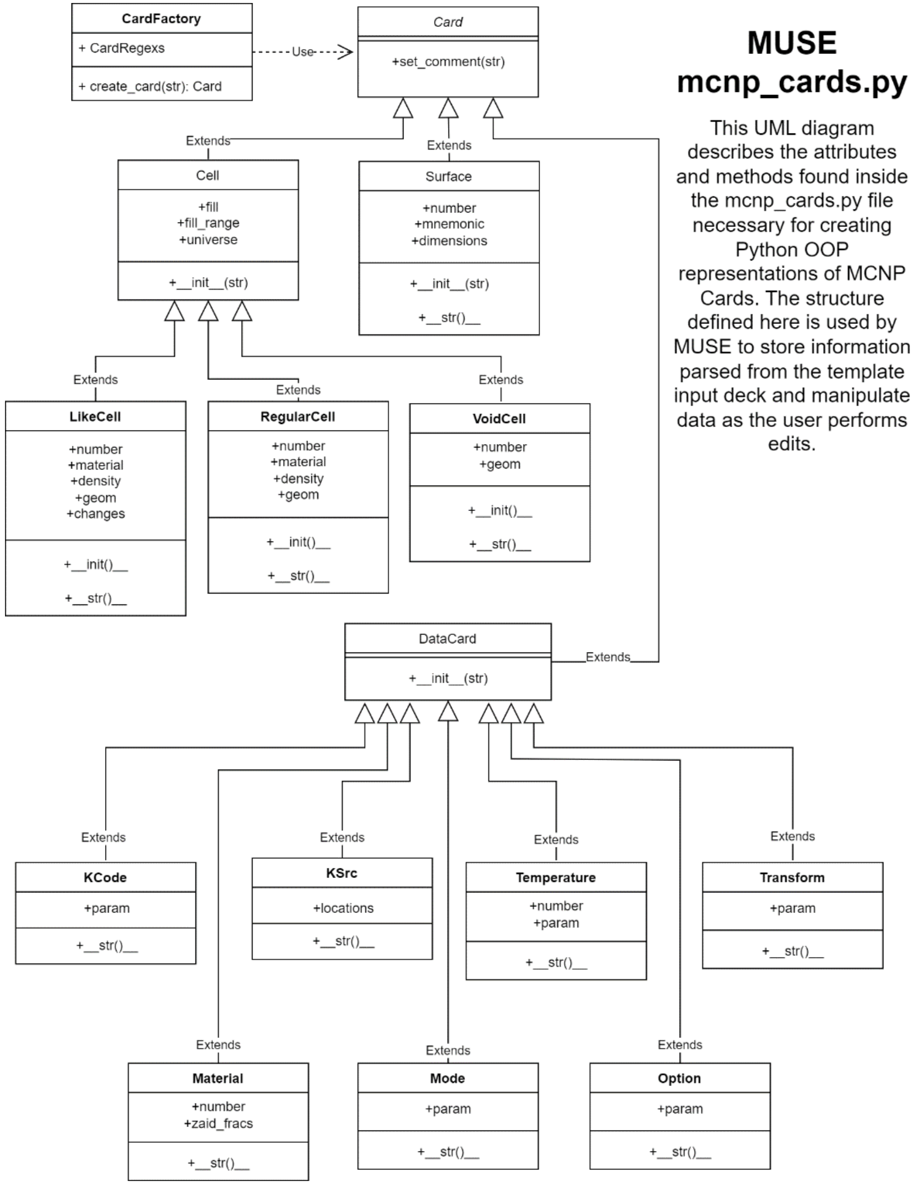
```
CELLS_REGEX = {
    'regular': re.compile(r' {0,6}^\d{1,6}[ \t]+[1-9]\d{0,6}[ \t]+-?\.?\d+(\.\d+)?'
                          r'[eE]?-?\d*[ \t]+[^a-zA-z]+[ \t]+[a-zA-z:,]+=.*$'),
    'void': re.compile(r'^ {0,6}\d{1,6}[ \t]+0[ \t][^a-zA-z]+[ \t]+[a-zA-z:,]+=.*$'),
    'like_but': re.compile(r'^ {0,6}\d{1,6}[ \t]+like[ \t]+\d{1,6}[ \t]but[ \t]+.+$')
}
```

MVC Structural Design

- Model
  - Object definitions for storing and printing data
- View
  - GUIs displayed to user
- Controller
  - Interface between model and controller
  - Handles user requests

MUSE:
Model-View-Controller (MVC)





MUSE mcnp_cards.py

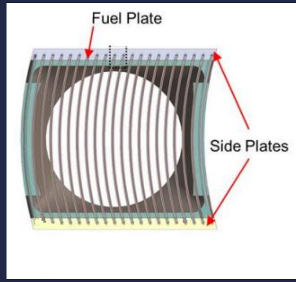
This UML diagram describes the attributes and methods found inside the mcnp_cards.py file necessary for creating Python OOP representations of MCNP Cards. The structure defined here is used by MUSE to store information parsed from the template input deck and manipulate data as the user performs edits.

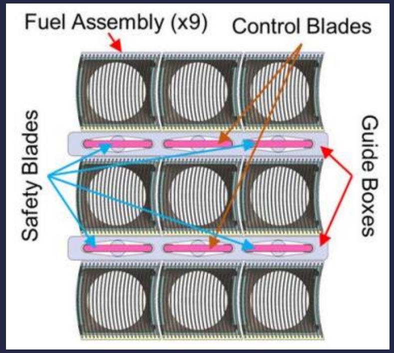
Models UML Diagram

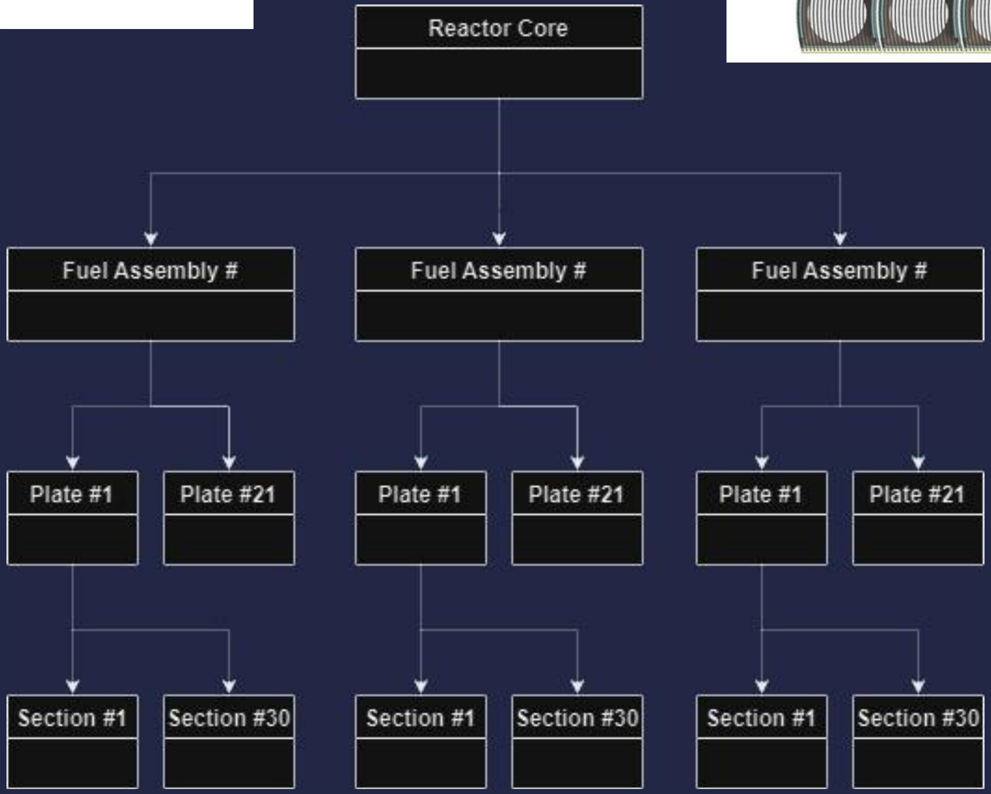
How Fuel Assembly Works

Creating Assemblies

1. Finds any cell with Uranium
2. Add cell to dictionary keyed by universe (Plate)
3. Find where each plate is used to identify a Fuel Lattice
4. Find where the lattice is used as a fill to identify Fuel Section
5. Find universe of Fuel Section contained in a Fuel Assembly
6. Append Fuel Section to Fuel Assembly Object containing reference to Fuel Lattice



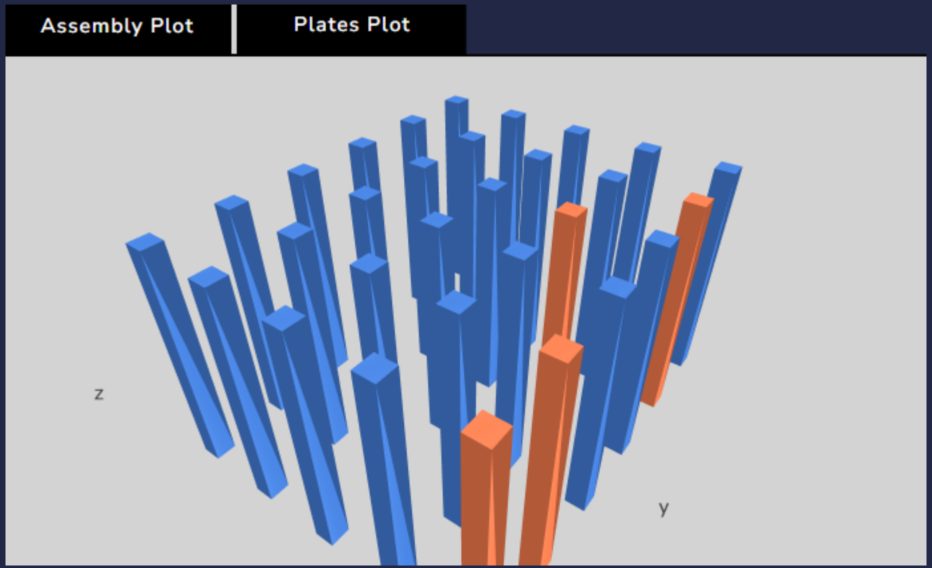




Limitations and Future Development

Generality

- Codebase and data storage is dynamic
- 3D plot selection is static
  - NNS design prioritized
- UI cannot create new objects



Improvements

- NCNR hosted server
- Dynamic 3D plot selection
- File preset selection
- Core design scaling

References and Acknowledgement

NNS Design:
[NURETH12 Paper Template (nist.gov)]

LANL MCNP 6.2 Manual
CoverSheet (lanl.gov)

LBL MCNP Primer
MCNPprmr\MCNPprmr.DVI (lbl.gov)

Dash Plotly:
Dash Documentation & User Guide | Plotly

Special Thanks to Abdullah Weiss and Osman Celikten

NCNR SURF Directors:
Julie Borchers, Susana Teixeira, Leland Harriger

Questions?

NIST CENTER FOR NEUTRON RESEARCH

CHRNS
CENTER FOR HIGH RESOLUTION NEUTRON SCATTERING
NSF
NIST