LA-UR-16-26661
Approved for public release; distribution is unlimited.

Title: Summer 2016

Author(s): Mendoza, Paul Michael

Intended for: Internal Presentation

Issued: 2016-08-31

Summer 2016

Paul Mendoza
Mentors:
Jeremy Conlin
Morgan White

August 23, 2016

Outline

1. Project Goals
2. Context
3. Applications Developed
   * benchmark_runner
   * format_static_bench_info
   * compile_bench_data
   * plot_benchmarks
4. Results
5. Future Work
6. Acknowledgements

Project Goals

Develop applications to:

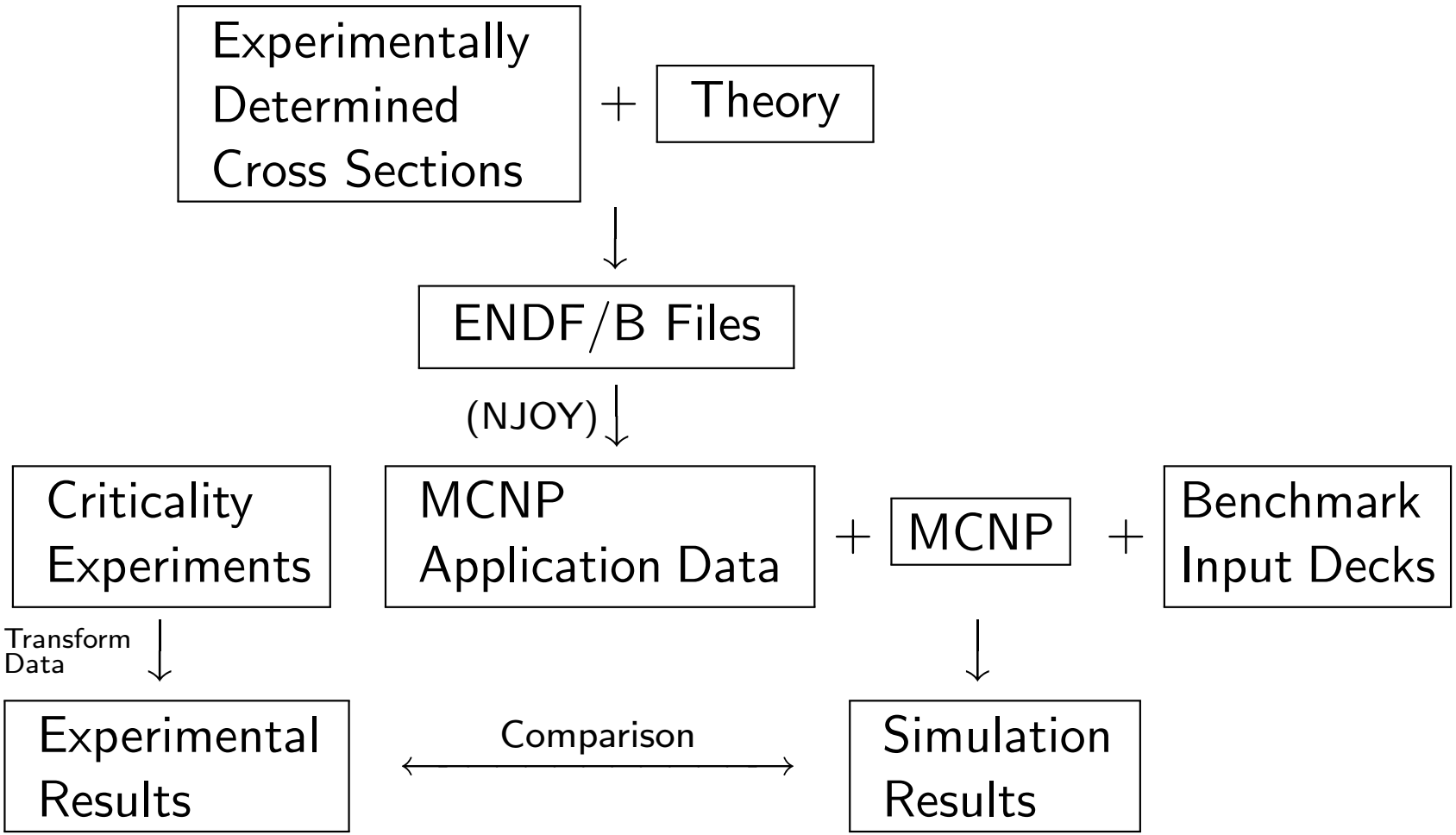
I. Automate MCNP criticality benchmark execution
   - With certain modifications

II. Create a dataset containing static benchmark information

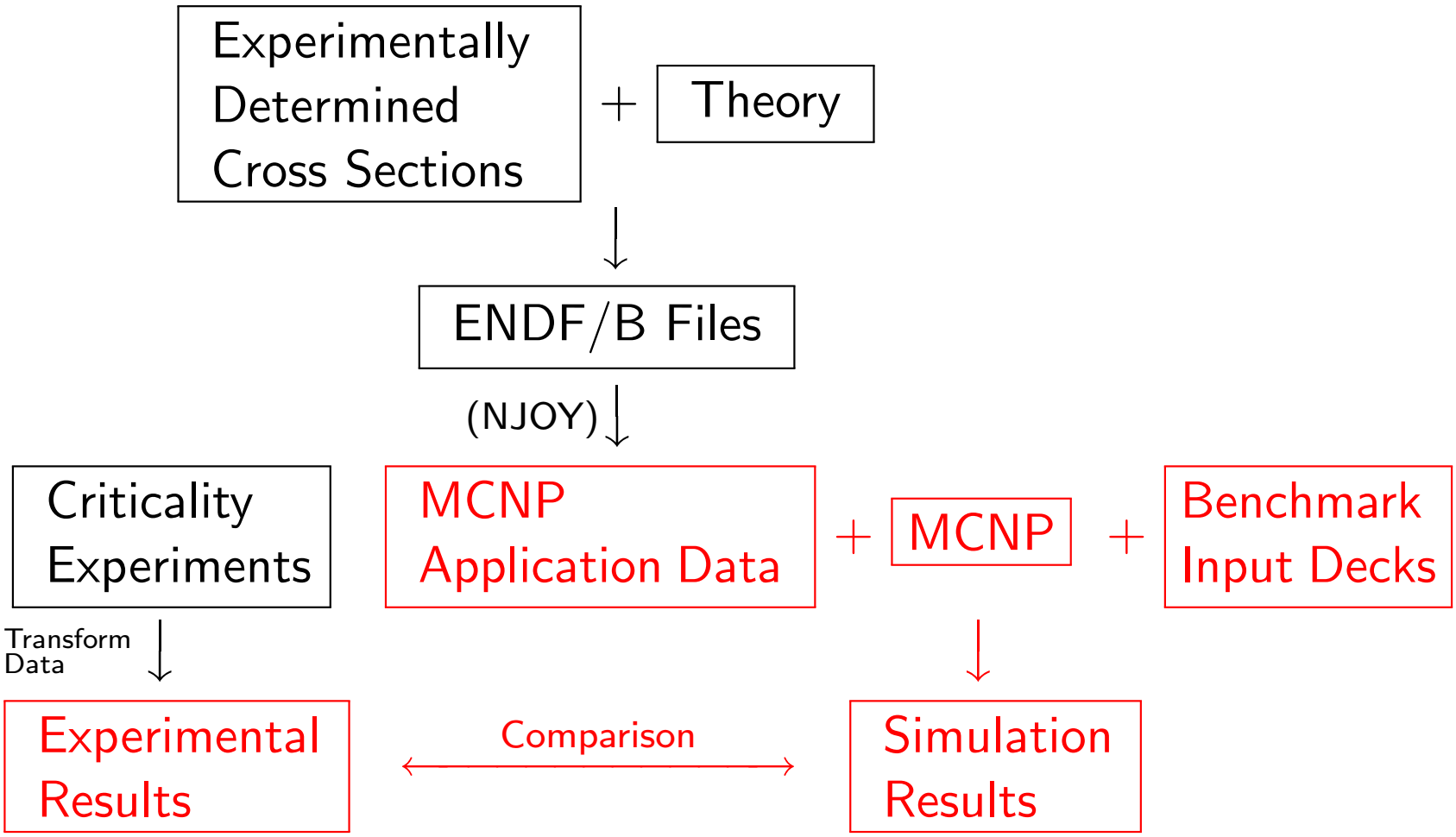
III. Combine MCNP output with benchmark information

IV. Fit and visually represent data

Context of Project



Context of Project



I: Automate MCNP benchmark execution

Algorithm flowchart for benchmark_runner

1. Process input information
   - Cross section dataset
   - User information: email and job email frequency
   - Job information
   - Input deck modifications (kcode and material card modifications)
2. Prepares to create moab scripts
3. Creates moab scripts and batch files for execution

I: Automate MCNP benchmark execution

Run benchmark_runner

1. Modify benchmark_runner inputs section to execute benchmarks for a selected configuration
2. Execute benchmark_runner in an empty directory
3. Execute RunBenchmarks

```
while [ $SubmitsAva -le "0" ];do
      sleep 1395
      JobsRunning="$(echo ljobs | sed -n '/paulmme/p' | wc -l)"
      SubmitsAva="$(echo "(30-$JobsRunning)" | bc)"
done
```

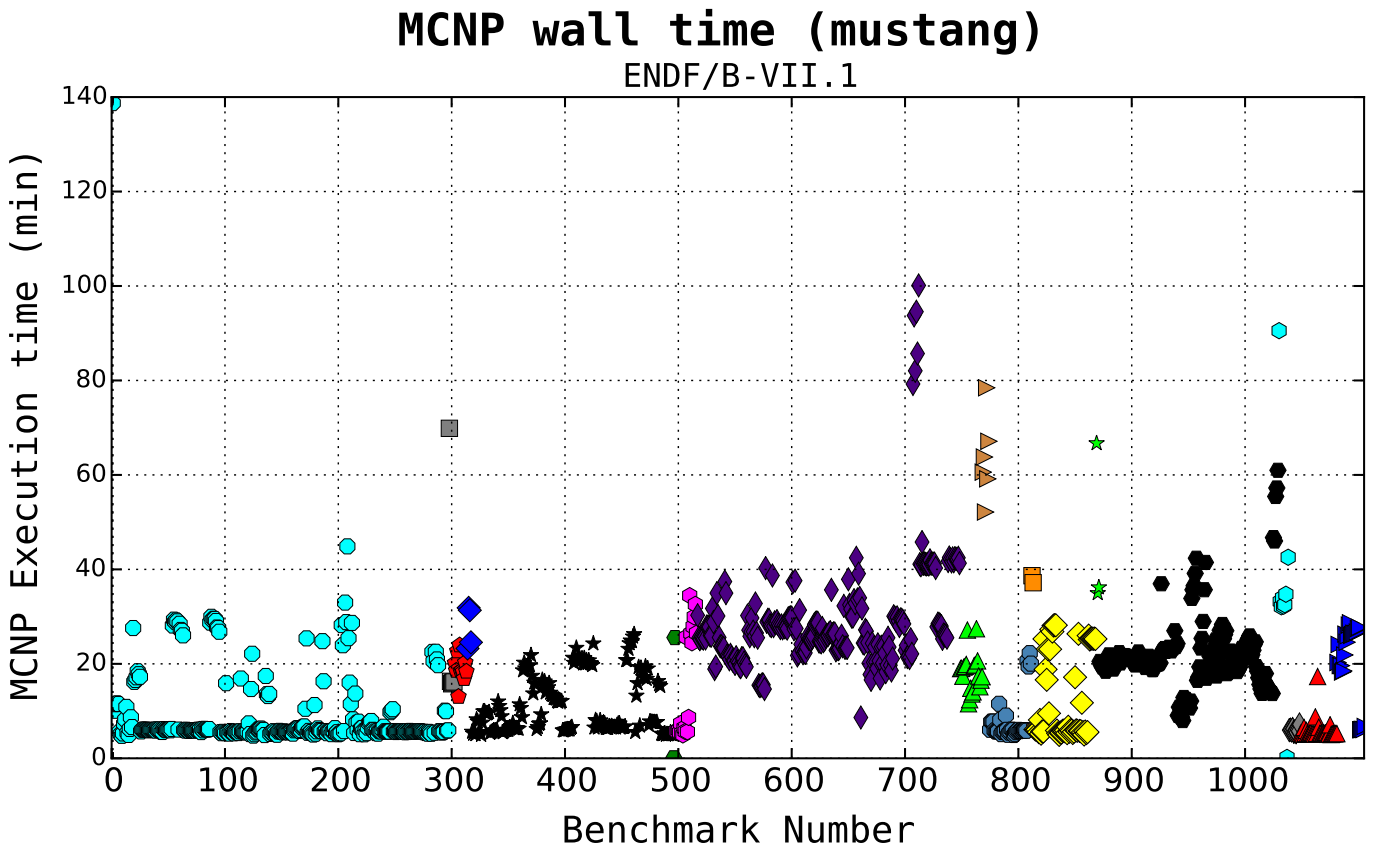
I: Automate MCNP benchmark execution

Check status of benchmark_runner

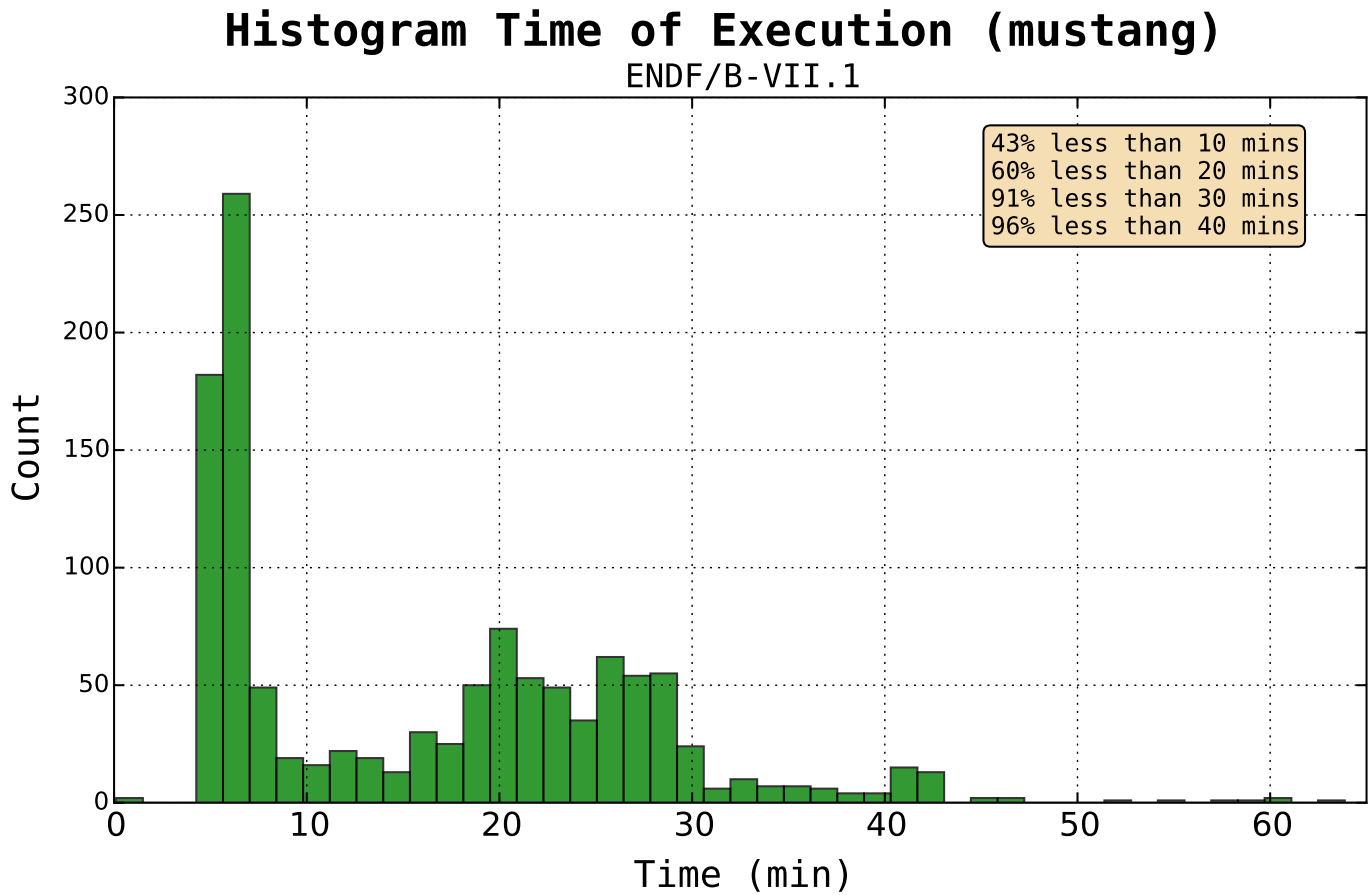
Use:
check_benchmarks
or
check_fatal
check_time
check_basic

```
The number of moab files in the run directory   = 1187

The number of .o outputs in output directory    = 1187
The number of .r outputs in output directory    = 1187
The number of .s outputs in output directory    = 1187
The number of .m outputs in output directory    = 1187

The number of slurm files in the run directory  = 1187
The number of runs that have completed          = 1187 (should = # .m outputs)

The number of runs that did not complete        = 0
The number of runs that have fatal errors       = 0
The number of runs that ran out of time         = 0
The number of runs that are still running       = 0

The first 10 slurm files names with fatal errors
The first 10 fatal error input file names
The first 10 unique fatal errors

The first 10 slurm file names for runs that ran out of time
The first 10 time sensitive input file names
The first 20 last cycle completed for runs that ran out of time

The first slurm file created was slurm-24877.out which ran HEU-SOL-THERM-025-015
The run started on 2016-07-06 around 13:17:08 and ran for 709.97 seconds

The last slurm file created was slurm-255741.out which ran HEU-MET-FAST-001
The run started on 2016-07-15 around 10:30:25 and ran for 7985.22 seconds
```

I: Automate MCNP benchmark execution



I: Automate MCNP benchmark execution



Project Goals

Develop applications to:

- ☒ Automate MCNP criticality benchmark execution
- ☐ Create a dataset containing static benchmark information
- ☐ Combine MCNP output with benchmark information
- ☐ Fit and visually represent data

II: Formatting Benchmark Information

The International Criticality Safety Benchmark Evaluation Project (ICSBEP)

- Compile critical and subcritical benchmark exerpiment data into a standardised format
- Used to validate calculation tools and cross section libraries
- Over 4000 critical, near-critical, or subcritical configuations

From Skip Kahler

- Provided over 1000 MCNP input decks from ICSBEP
- and an excel spreedsheet with k-eigenvalues, errors, and geometry configurations

II: Formatting Benchmark Information

Compile information from Skip's spreedsheet with additional other static information

- Gather and organize data from Skip's spreedsheet
  - expy.py
  - With complete information about: names, benchmark keff with error, geometry
  - With sparse information about: reflectors, moderators, ratio of volumes, geometric dimentions
- Merge with number density and volume data
  - format_static_bench_info.py

Project Goals

Develop applications to:

- ☒ Automate MCNP criticality benchmark execution
- ☒ Create a dataset containing static benchmark information
- ☐ Combine MCNP output with benchmark information
- ☐ Fit and visually represent data

Compile static and MCNP results benchmark data

compile_bench_data algorithm flowchart

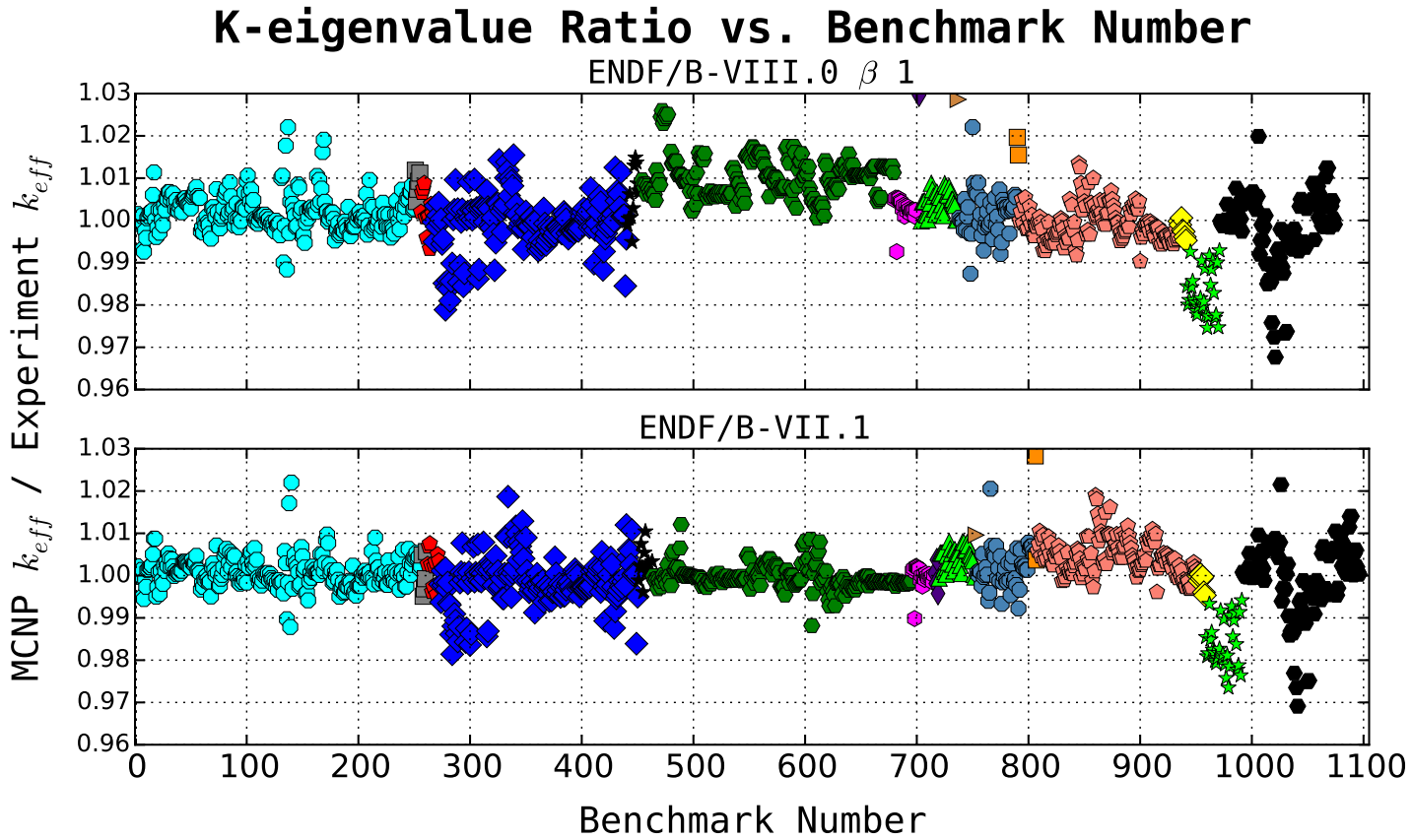
- Save in a matrix information from the static benchmark info dataset
- Loop through MCNP outputs and gather:
  - keff with error, percent fission (thermal, intermediate, fast), average energy causing fission
- Link data with static benchmark information
- Save information in a csv file

Plotting program

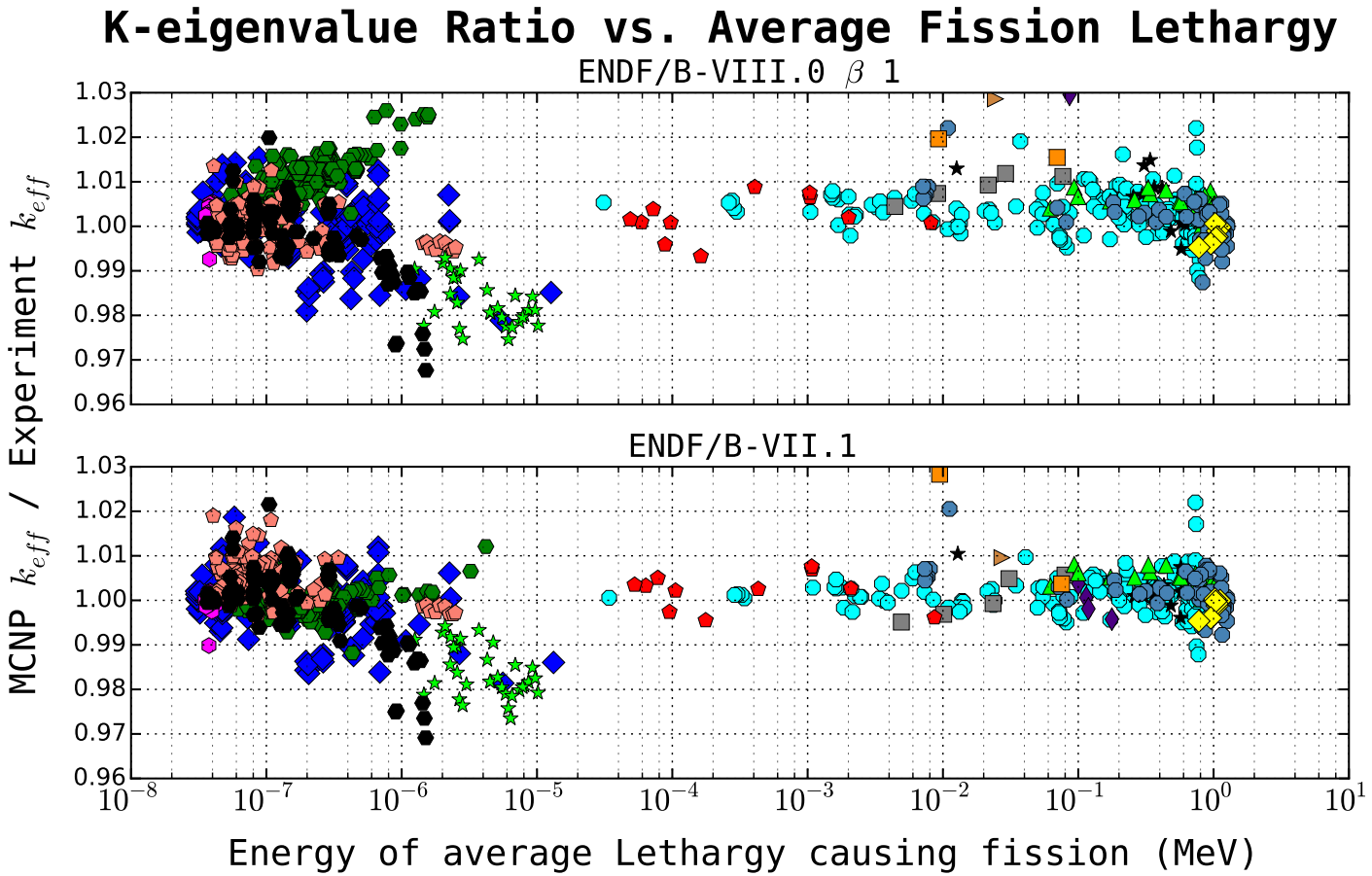
How to use:

- Create input deck for plot.py
  - Input deck should specify:
    - The data to be plotted
    - The type of plot
    - The geometry of the plots
- Execute with 'python plot.py input_deck.py'

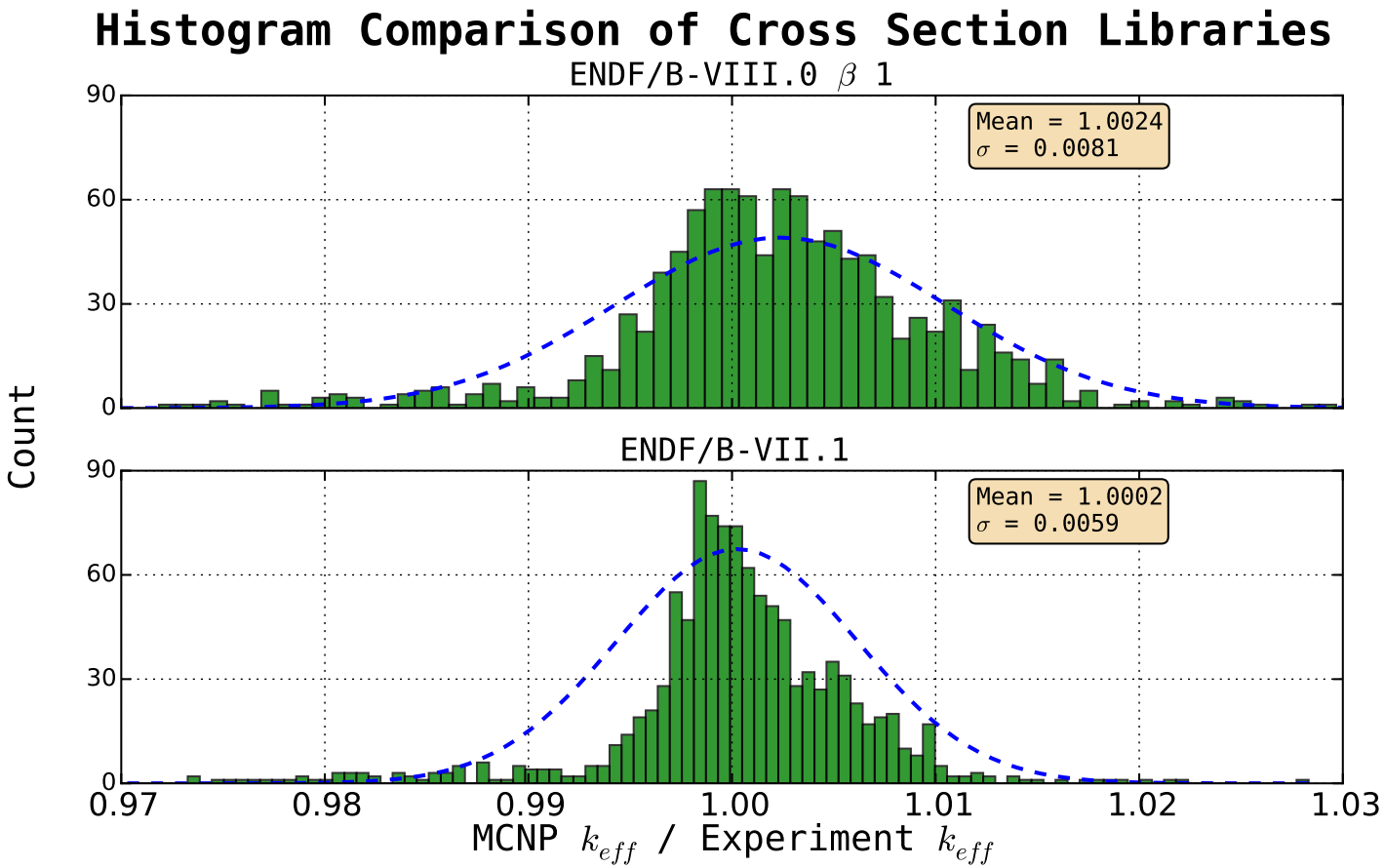
Plots


K-eigenvalue Ratio vs. Benchmark Number
ENDF/B-VIII.0 β 1
ENDF/B-VII.1
MCNP k_{eff} / Experiment k_{eff}
1.03
1.02
1.01
1.00
0.99
0.98
0.97
0.96
0
100
200
300
400
500
600
700
800
900
1000
1100
Benchmark Number

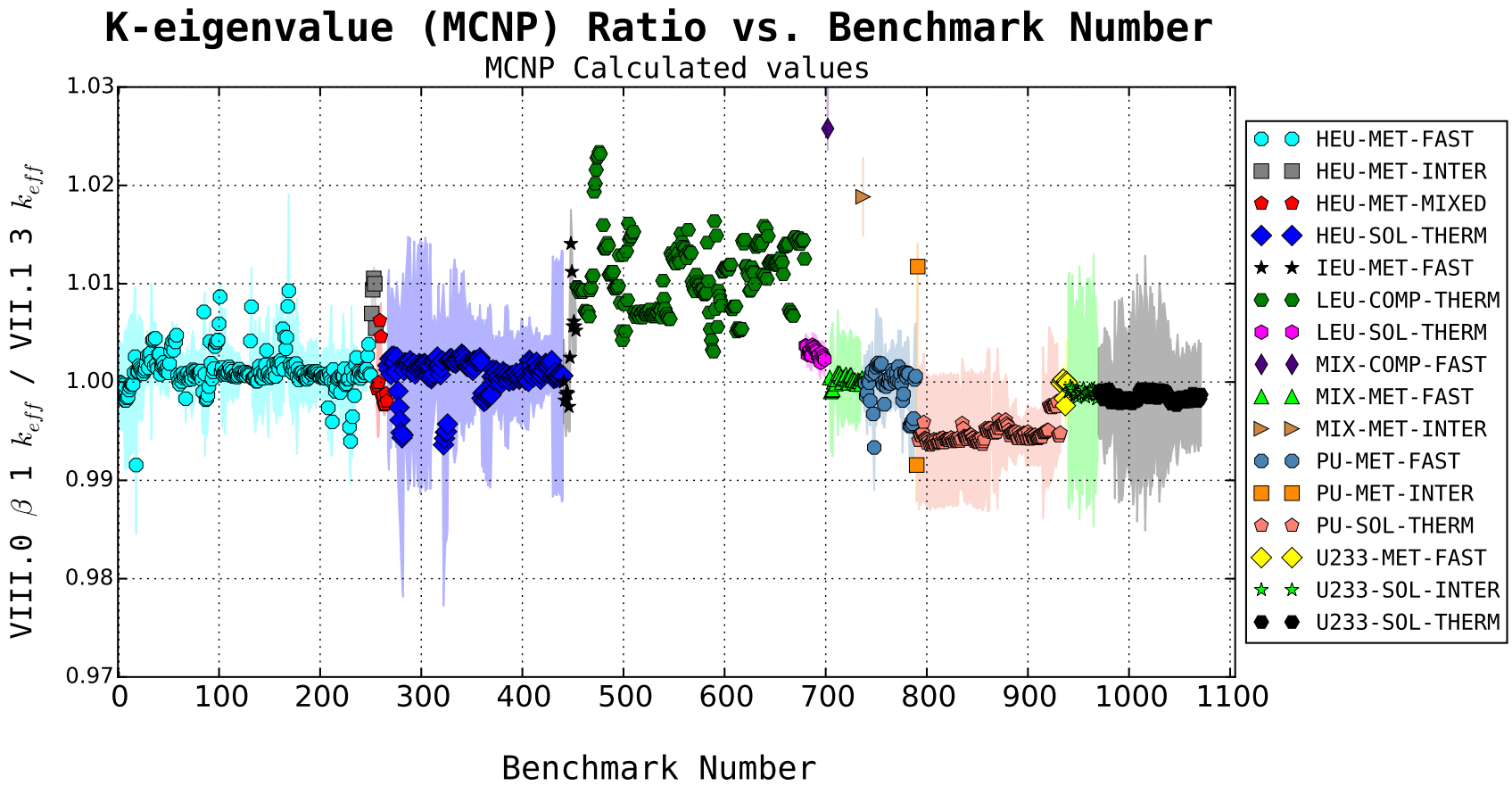
Plots


K-eigenvalue Ratio vs. Average Fission Lethargy
ENDF/B-VIII.0 β 1
ENDF/B-VII.1
MCNP k_{eff} / Experiment k_{eff}
1.03
1.02
1.01
1.00
0.99
0.98
0.97
0.96
10^{-8}
10^{-7}
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
Energy of average Lethargy causing fission (MeV)

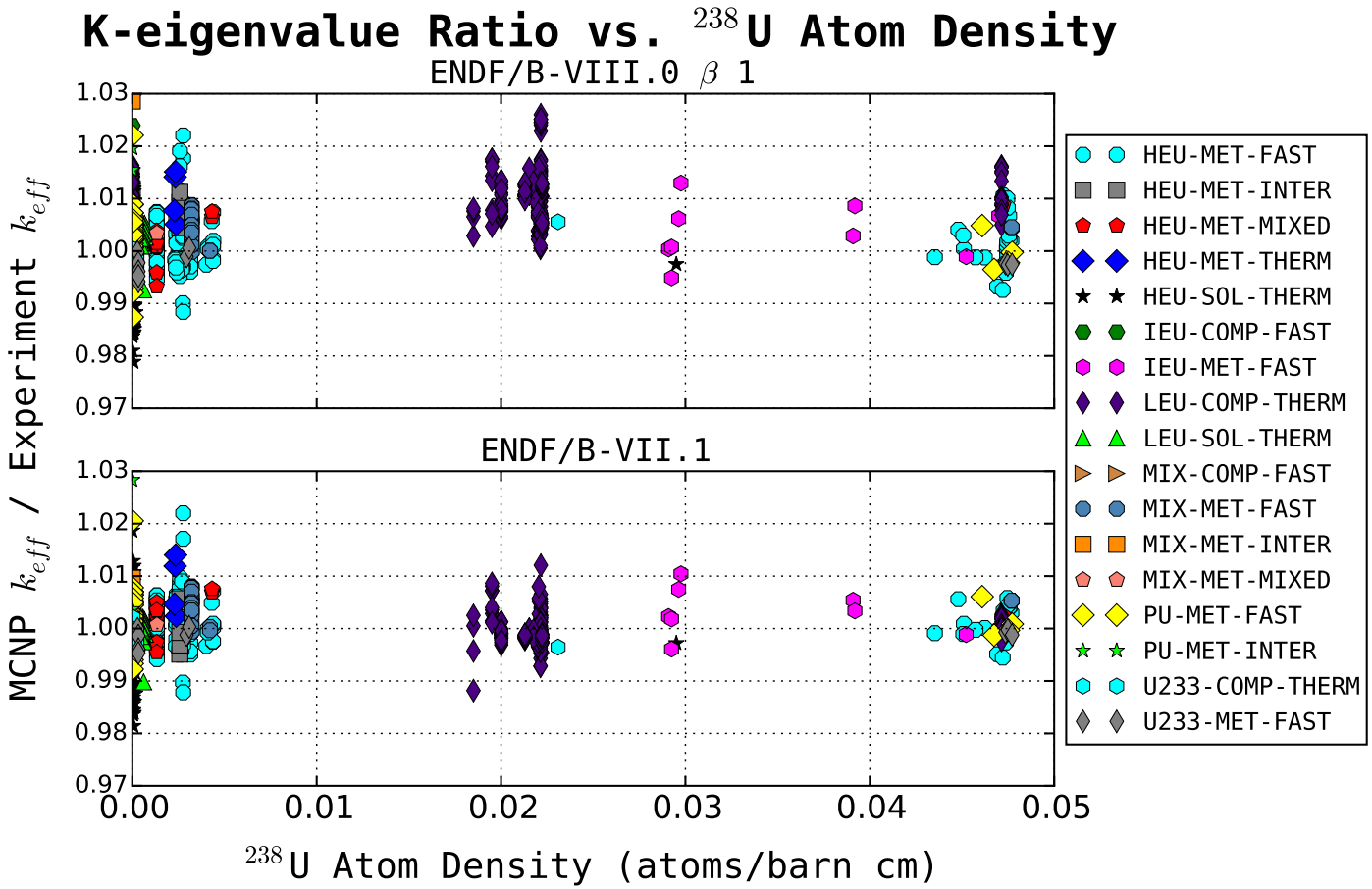
Plots


Histogram Comparison of Cross Section Libraries
ENDF/B-VIII.0 β 1
Mean = 1.0024
σ = 0.0081
ENDF/B-VII.1
Mean = 1.0002
σ = 0.0059
Count
0
30
60
90
0.97
0.98
0.99
1.00
1.01
1.02
1.03
MCNP k_{eff} / Experiment k_{eff}

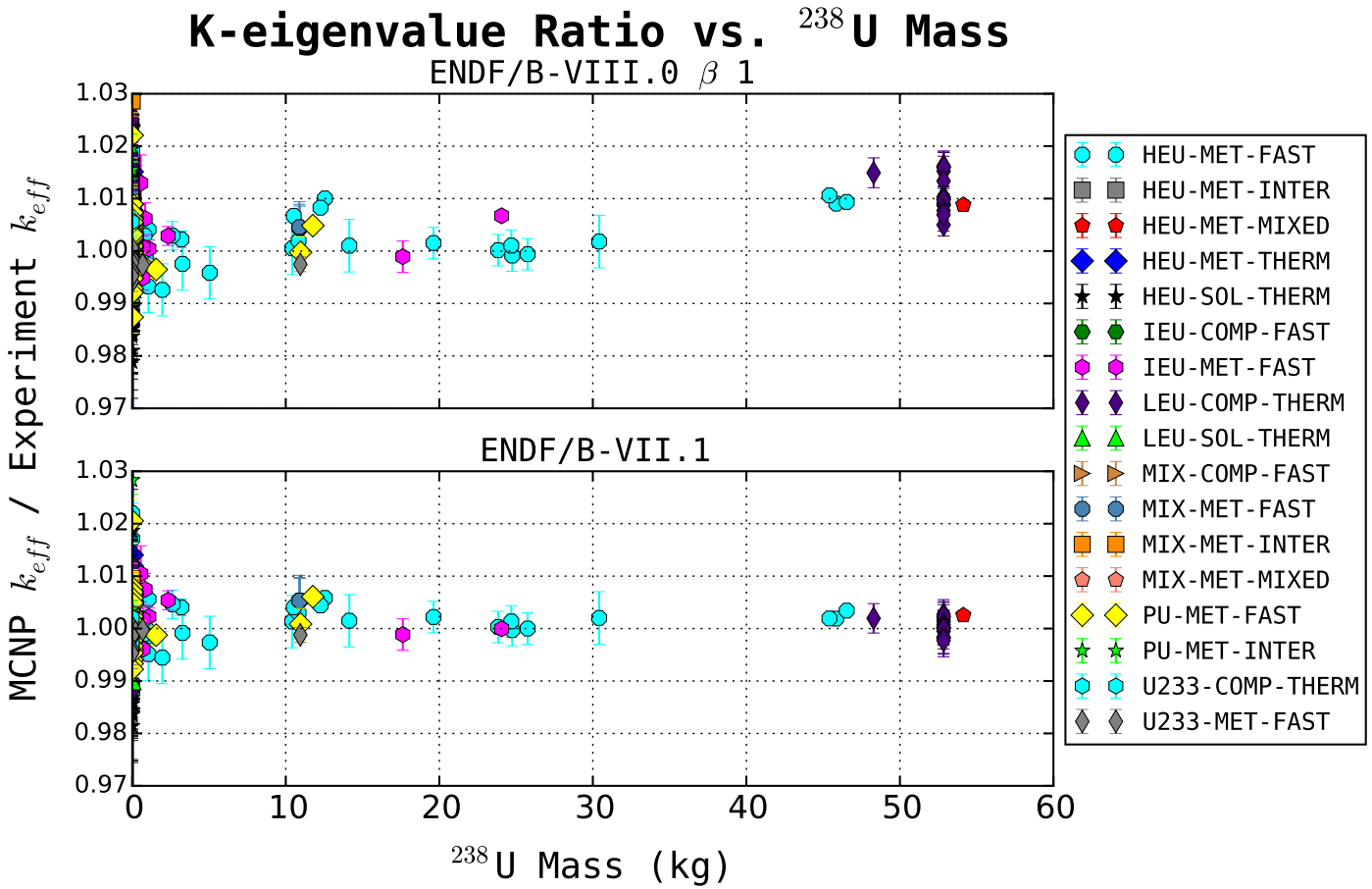
Plots


K-eigenvalue (MCNP) Ratio vs. Benchmark Number
MCNP Calculated values
VIII.0 β 1 k_{eff} / VII.1 3 k_{eff}
Benchmark Number
1.03
1.02
1.01
1.00
0.99
0.98
0.97
0
100
200
300
400
500
600
700
800
900
1000
1100
HEU-MET-FAST
HEU-MET-INTER
HEU-MET-MIXED
HEU-SOL-THERM
IEU-MET-FAST
LEU-COMP-THERM
LEU-SOL-THERM
MIX-COMP-FAST
MIX-MET-FAST
MIX-MET-INTER
PU-MET-FAST
PU-MET-INTER
PU-SOL-THERM
U233-MET-FAST
U233-SOL-INTER
U233-SOL-THERM

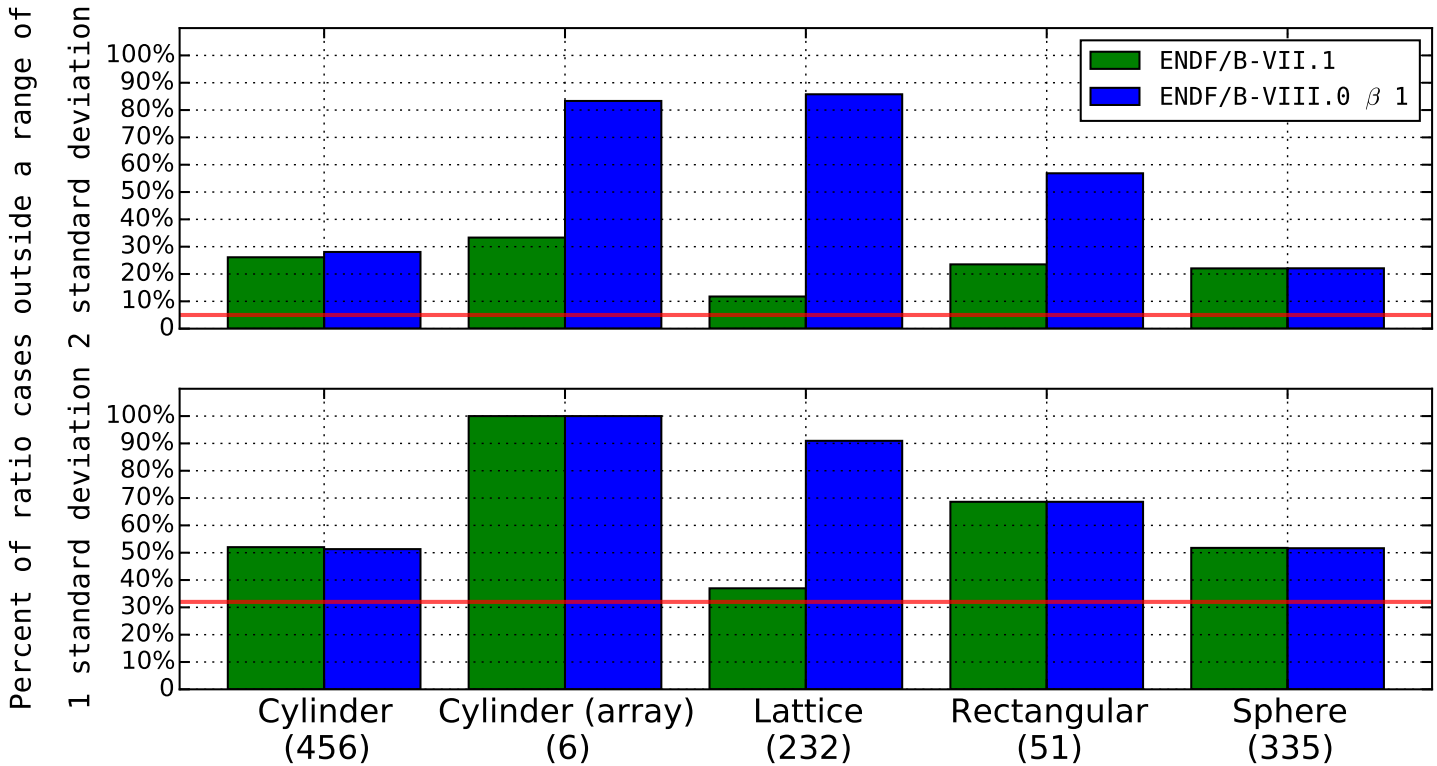
Plots


K-eigenvalue Ratio vs. ^{238}U Atom Density
ENDF/B-VIII.0 β 1
ENDF/B-VII.1
MCNP k_{eff} / Experiment k_{eff}
^{238}U Atom Density (atoms/barn cm)
1.03
1.02
1.01
1.00
0.99
0.98
0.97
0.00
0.01
0.02
0.03
0.04
0.05
HEU-MET-FAST
HEU-MET-INTER
HEU-MET-MIXED
HEU-MET-THERM
HEU-SOL-THERM
IEU-COMP-FAST
IEU-MET-FAST
LEU-COMP-THERM
LEU-SOL-THERM
MIX-COMP-FAST
MIX-MET-FAST
MIX-MET-INTER
MIX-MET-MIXED
PU-MET-FAST
PU-MET-INTER
U233-COMP-THERM
U233-MET-FAST

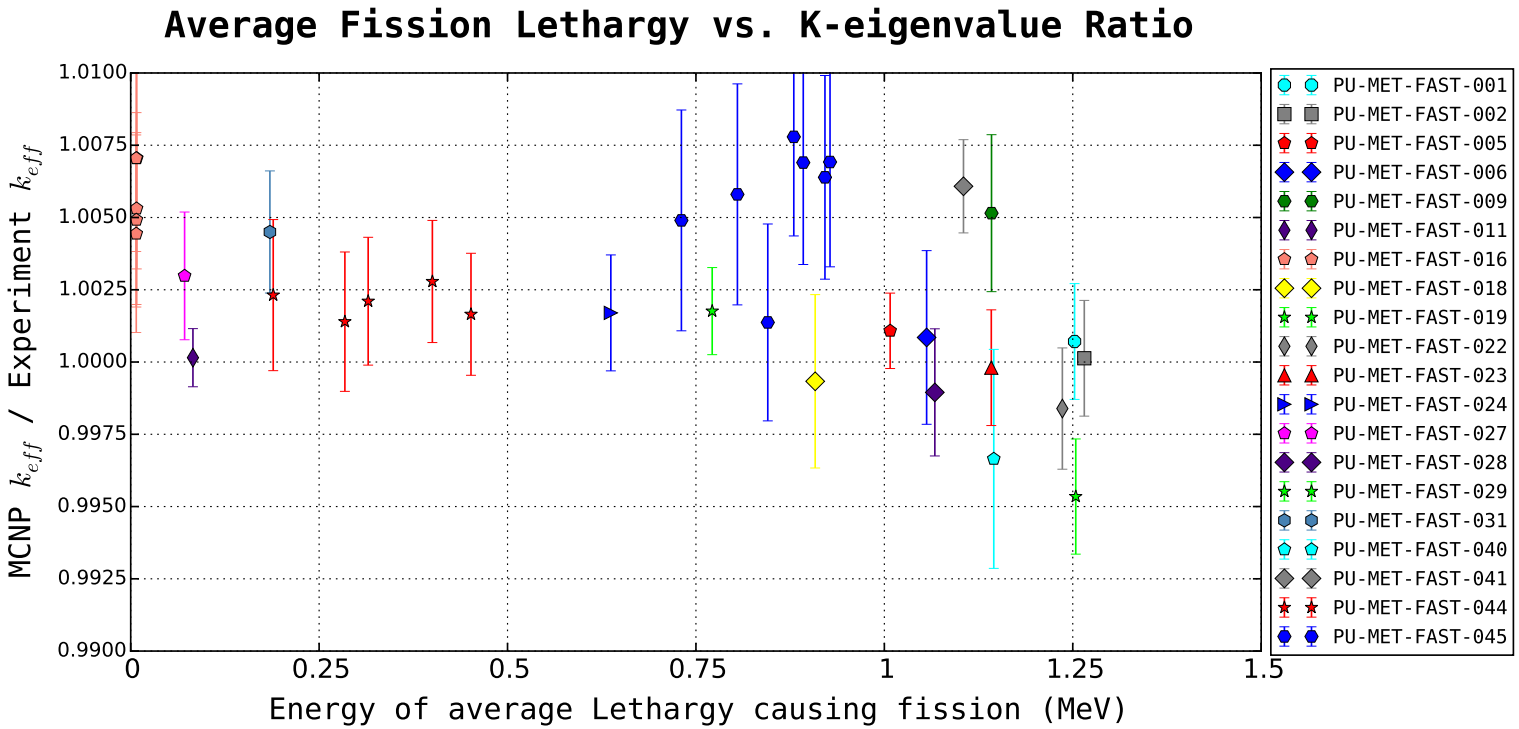
Plots


K-eigenvalue Ratio vs. ^{238}U Mass
ENDF/B-VIII.0 β 1
ENDF/B-VII.1
MCNP k_{eff} / Experiment k_{eff}
^{238}U Mass (kg)
1.03
1.02
1.01
1.00
0.99
0.98
0.97
0
10
20
30
40
50
60
HEU-MET-FAST
HEU-MET-INTER
HEU-MET-MIXED
HEU-MET-THERM
HEU-SOL-THERM
IEU-COMP-FAST
IEU-MET-FAST
LEU-COMP-THERM
LEU-SOL-THERM
MIX-COMP-FAST
MIX-MET-FAST
MIX-MET-INTER
MIX-MET-MIXED
PU-MET-FAST
PU-MET-INTER
U233-COMP-THERM
U233-MET-FAST

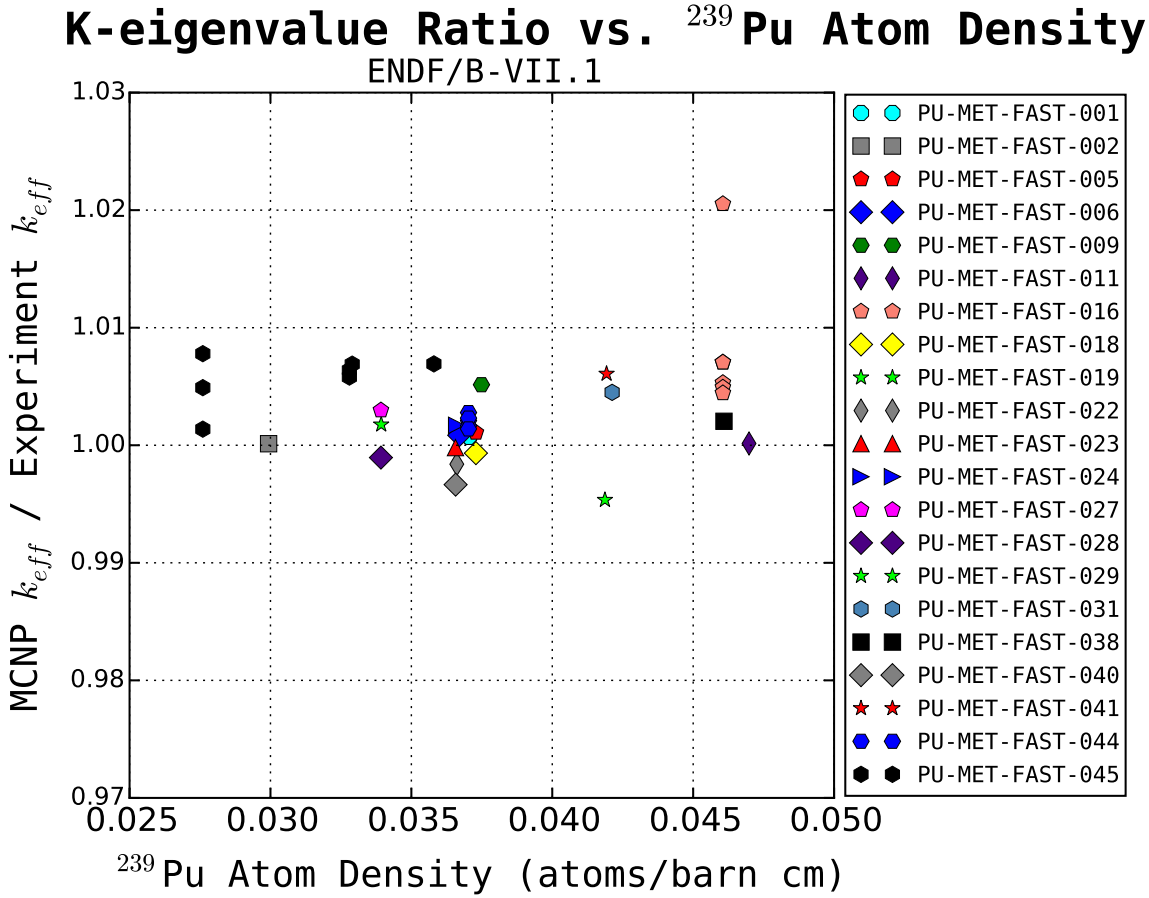
Plots


Percent of ratio cases outside a range of
1 standard deviation 2 standard deviation
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0
ENDF/B-VII.1
ENDF/B-VIII.0 β 1
Cylinder (456)
Cylinder (array) (6)
Lattice (232)
Rectangular (51)
Sphere (335)

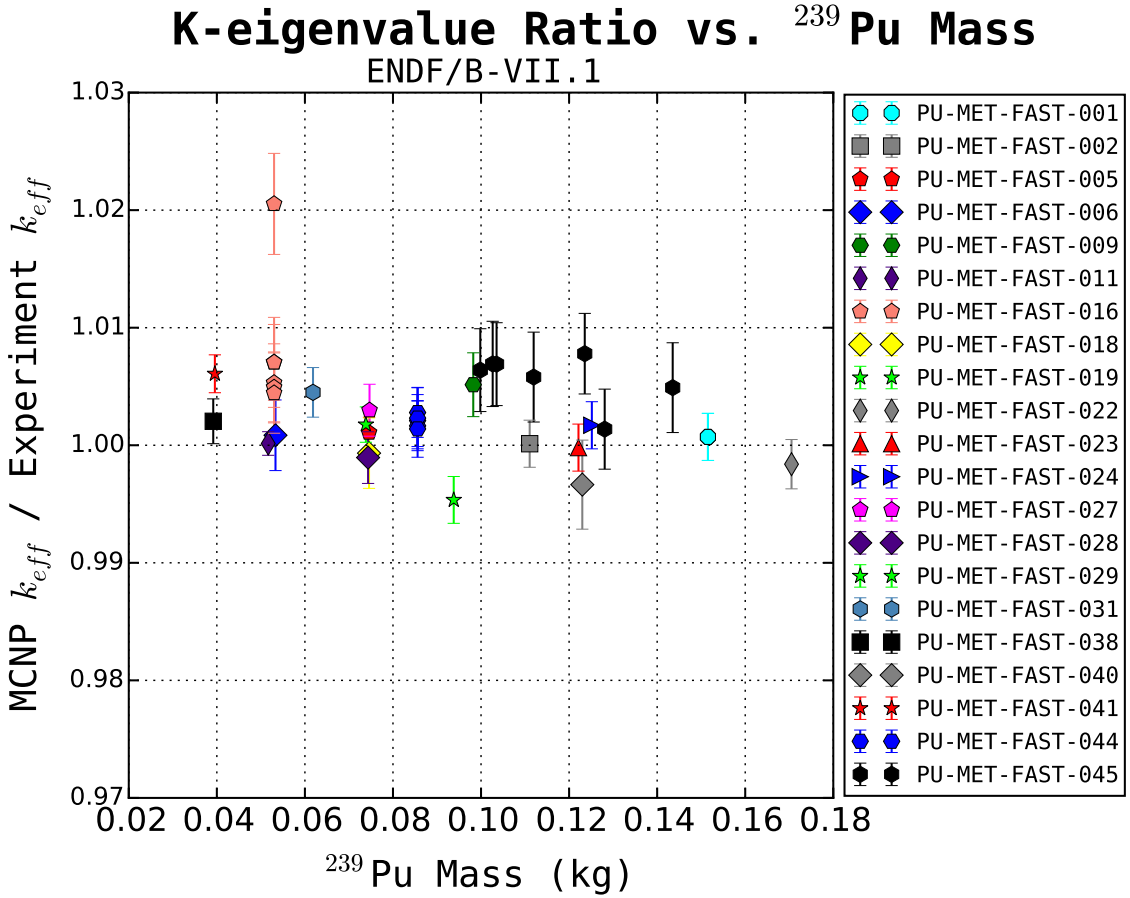
Plots


Average Fission Lethargy vs. K-eigenvalue Ratio
MCNP k_{eff} / Experiment k_{eff}
1.0100
1.0075
1.0050
1.0025
1.0000
0.9975
0.9950
0.9925
0.9900
0
0.25
0.5
0.75
1
1.25
1.5
Energy of average Lethargy causing fission (MeV)
PU-MET-FAST-001
PU-MET-FAST-002
PU-MET-FAST-005
PU-MET-FAST-006
PU-MET-FAST-009
PU-MET-FAST-011
PU-MET-FAST-016
PU-MET-FAST-018
PU-MET-FAST-019
PU-MET-FAST-022
PU-MET-FAST-023
PU-MET-FAST-024
PU-MET-FAST-027
PU-MET-FAST-028
PU-MET-FAST-029
PU-MET-FAST-031
PU-MET-FAST-040
PU-MET-FAST-041
PU-MET-FAST-044
PU-MET-FAST-045

Plots


K-eigenvalue Ratio vs. ^{239}Pu Atom Density
ENDF/B-VII.1
MCNP k_{eff} / Experiment k_{eff}
1.03
1.02
1.01
1.00
0.99
0.98
0.97
0.025
0.030
0.035
0.040
0.045
0.050
^{239}Pu Atom Density (atoms/barn cm)
PU-MET-FAST-001
PU-MET-FAST-002
PU-MET-FAST-005
PU-MET-FAST-006
PU-MET-FAST-009
PU-MET-FAST-011
PU-MET-FAST-016
PU-MET-FAST-018
PU-MET-FAST-019
PU-MET-FAST-022
PU-MET-FAST-023
PU-MET-FAST-024
PU-MET-FAST-027
PU-MET-FAST-028
PU-MET-FAST-029
PU-MET-FAST-031
PU-MET-FAST-038
PU-MET-FAST-040
PU-MET-FAST-041
PU-MET-FAST-044
PU-MET-FAST-045

Plots


K-eigenvalue Ratio vs. ^{239}Pu Mass
ENDF/B-VII.1
MCNP k_{eff} / Experiment k_{eff}
1.03
1.02
1.01
1.00
0.99
0.98
0.97
0.02 0.04 0.06 0.08 0.10 0.12 0.14 0.16 0.18
^{239}Pu Mass (kg)
PU-MET-FAST-001
PU-MET-FAST-002
PU-MET-FAST-005
PU-MET-FAST-006
PU-MET-FAST-009
PU-MET-FAST-011
PU-MET-FAST-016
PU-MET-FAST-018
PU-MET-FAST-019
PU-MET-FAST-022
PU-MET-FAST-023
PU-MET-FAST-024
PU-MET-FAST-027
PU-MET-FAST-028
PU-MET-FAST-029
PU-MET-FAST-031
PU-MET-FAST-038
PU-MET-FAST-040
PU-MET-FAST-041
PU-MET-FAST-044
PU-MET-FAST-045

Future Work

1. Further refinement of programs
   - Static benchmark information needs to be organized (number densities and volumes)
   - Would like to gather leakage information
   - Comment plotting program
2. Analyze the data
   - Regression models and other things, starts with asking the right questions.

Acknowledgements

- Jeremy Conlin, XCP-5
  - Provided guidance and insight throughout this process
  - Helped with understanding the moab client, organizing the benchmark repository, introducing me to Git
- Albert (Skip) Kahler III, T-2
  - Provided the benchmark MCNP input decks and excel sheets with benchmark information, and several plots presented are reproductions of his work.
- Morgan White, XCP-5
  - Responsive to questions and request for the project.
  - Special thanks for organize tours and attending the student symposium.
- Austin McCartney, XCP-5
  - Sitting through questions about Git, python, and bash, and helping on several occations with debugging.

Accomplishments for summer 2016

- ☒ Travel to Japan
- ☒ Publish a paper
- ☒ Swing on a rope swing into a lake
- ☒ Go Contra dancing
- ☒ Make a ton of great friends
- ☒ Star gaze during a meteor shower
- ☒ Run through a monsoon
- ☒ Write a poem
- ☒ Beat Jeremy at a game of Pong
- ☒ Learn lots of stuff
- ☐ Leave Los Alamos unmarked

Future Work

I used to play pretend
Give my codes different names
Used to program rocket ships
and then I'd fly them far away
I used to code in empty space
but Pavel's laughing at my face
saying 'wake up your code won't compile'
yo

-Modification of: Stressed Out by Twenty One Pilots