INL/CON-24-79139-Revision-0

Probabilistic Analysis of Uncertainty in ATRC Flux Profiles

July 2024

Changing the World's Energy Future

Alysha Yaeun Allgood

INL/CON-24-79139-Revision-0

Probabilistic Analysis of Uncertainty in ATRC Flux Profiles

Alysha Yaeun Allgood

July 2024

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

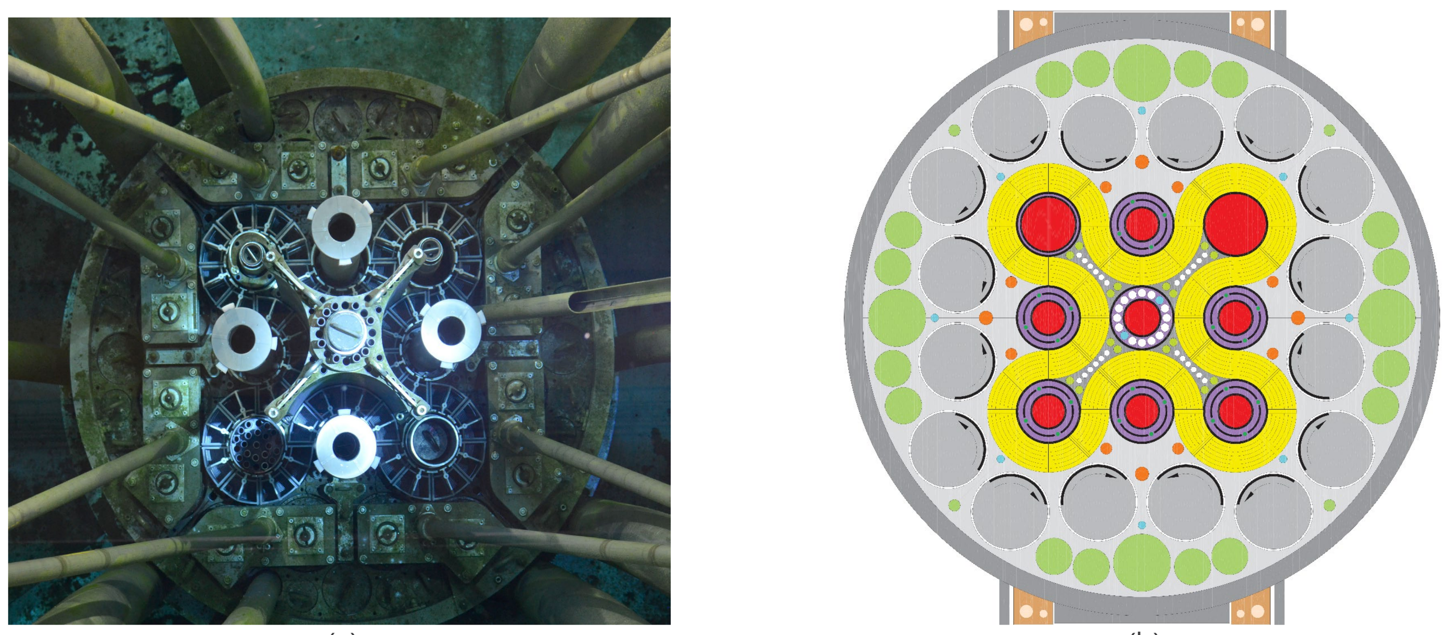
Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Probabilistic Analysis of Uncertainty in ATRC Flux Profiles

Alysha Allgood | University of Utah | G510 - Reactor Engineering | Nathan Manwaring, Andrew Maile

Introduction

The Advanced Test Reactor Critical (ATRC) is a low-power pool type reactor located in the Advanced Test Reactor Complex. ATRC is a replica of the larger ATR design and is used to conduct research and obtain data such as flux measurements, excess reactivity, and loading requirements for ATR experiments before tests are inserted into ATR proper.[1]



(a) (b)

Fig. 1: (a) Overhead view of the Advanced Test Reactor Critical after the 6th core overhaul completed in March 2022. (b) Illustration of the ATR core depicting fuel rods, in-pile experiment loops, and beryllium and hafnium control cylinders.

One method for determining the impact an experiment will have at ATR is by looking at the axial flux profile along the fuel rod in the corresponding ATRC experiment. An axial profile can be determined by placing 25 small aluminum-uranium alloy flux wires at two-inch intervals along a plastic strip known as a flux wand, inserting into specific fuel elements, and measuring the radiation emitted from each wire.[2]

Objective

The flux wand measurements includes large amounts of variation which makes drawing conclusions from the data difficult. The objective of this project is to take the previously collected ATRC data and create a definitive method to propagate the uncertainty from ATRC measurements which would help calculations in the future.

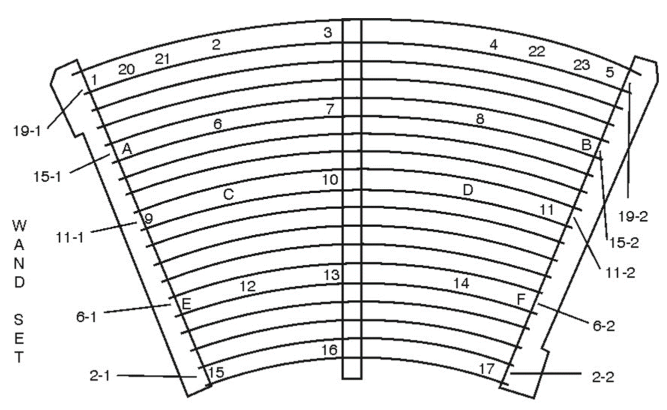


Fig. 2: ATRC Fuel Element Channel Layout for Flux Wand Placement.[3]

Results

To begin the error propagation calculations, the error in measurements from ATRC data needed to be quantified. This was done by finding the 95% confidence interval of the deviations in measurement. This gives a top end of expected error.

(a) (b)

Fig. 3: (a) Fission Rate Data from ATRC measurements with the 95% confidence interval (CI) and mean. (b) Standard deviation of the error in ATRC measurements with the 95% confidence interval and mean.

The equations were then integrated into a Python script that calculated axial peaking factors, axial nodes, power fractions, their respective errors, and probability. The probabilities were then ranked and exported to an Excel file for ease of viewing.

```python
####### Error Calculation #############
#calculating peaking factor error
a = 1/24
b = 1/48
c = 2/24
user_axial_error = []
axial_error_dataframe_creator(user_fission_data_complete, user_fiss_ave, user_axial_error)
user_axial_error_df = pd.DataFrame(user_axial_error, columns = np.linspace(24, -24, 25))

#calculating error in nodal peaking factor
Ptotal = 48
user_axial_nodes_error = []
axial_node_error_dataframe_creator(user_peaking, user_axial_error, user_axial_nodes_error)
user_ax_node_error_df = pd.DataFrame(user_axial_nodes_error, columns = labels_short)

#calculating error in nodal power fraction
node4 = 11
node5 = 13
node6 = 15
node7 = 17
node8 = 20
A = 1
B = 2
user_power_fraction_error = []
power_fraction_error_dataframe_creator(user_peaking, user_axial_error, A, B, user_power_fraction_error)
user_power_frac_node_error_df =  pd.DataFrame(user_power_fraction_error, columns = labels_short)

########### Uncertainty Calculations ##############
user_ax_node_mean = pd.concat([axial_node_df, user_axial_node_df], axis = 0).mean()
user_power_frac_mean = pd.concat([power_frac_df, user_power_frac_df], axis = 0).mean()
user_axial_uncertainty = []
user_power_uncertainty = []
uncertainty(user_axial_peaking_nodes, ax_node_mean, user_axial_nodes_error, user_axial_uncertainty)
uncertainty(user_power_frac_nodes, power_frac_mean, user_power_fraction_error, user_power_uncertainty)
user_axial_uncertainty_df = pd.DataFrame(user_axial_uncertainty, columns = ['4', '5', '6', '7', '8'])
user_power_uncertainty_df = pd.DataFrame(user_power_uncertainty, columns = ['4', '5', '6', '7', '8'])

probability = 1- ((1/2)**10)
user_complement = []
user_relative = []
```

Fig. 4: A small portion of the developed Python code.

	A	B	M	N
1		Flux Run	Complement Percentage	Relative Percentage
2		Name	0	Relative Percentage
3				
4	0	19-2FE13ch19	99.78547629	-0.1169817
5	1	19-2FE13ch15	99.71136817	-0.191162258
6	2	19-2FE13ch11	99.86364834	-0.038733238
7	3	19-2FE13ch6	99.80804902	-0.094386908
8	4	19-2FE13ch2	99.80303509	-0.099405733
9	5	19-2FE15ch19	99.30525603	-0.597671385
10	6	19-2FE15ch15	99.68269764	-0.219860822
11	7	19-2FE15ch11	99.76291859	-0.139561453
12	8	19-2FE15ch6	99.52631015	-0.376401178
13	9	19-2FE15ch2	99.70487622	-0.197660563
14	10	19-2FE17ch19	99.58643066	-0.316221902
15	11	19-2FE17ch15	99.6808488	-0.221711469
16	12	19-2FE17ch11	99.79133193	-0.11112034
17	13	19-2FE17ch6	99.77106884	-0.131403235
18	14	19-2FE17ch2	99.77131333	-0.131158507
19	15	19-2FE19ch19	99.66575296	-0.236822063
20	16	19-2FE19ch15	99.86674476	-0.03563379
21	17	19-2FE19ch11	99.87833188	-0.024035341
22	18	19-2FE19ch6	99.88858796	-0.013769236
23	19	19-2FE19ch2	99.98707217	0.084811241
24	20	18-6FE23ch19	99.91713757	0.014808282
25	21	18-6FE23ch15	99.83741601	-0.064991211
26	22	18-6FE23ch11	99.9784987	0.076229392
27	23	18-6FE23ch6	99.28150269	-0.621447945
28	24	18-6FE23ch2	99.99942059	0.09717173
29	25	18-5FE33ch19	99.81148072	-0.090951853
30	26	18-5FE33ch15	99.90260599	0.000262492
31	27	18-5FE33ch11ECAR4241	99.86816009	-0.034217075
32	28	18-5FE33ch11bad	99.87806949	-0.02429799
33	29	18-5FE33ch6	99.9274066	0.025087354
34	30	18-5FE33ch2	99.99122337	0.088966506
35	31	18-5FE35ch19ECAR4241	99.8889153	-0.013441574
36	32	18-5FE35ch19bad	99.87769175	-0.024676102
37	33	18-5FE35ch15	99.88857054	-0.013786674

Fig. 5: A small portion of the Python code output.

Methods

The standard uncertainty is the combination of all independent, random errors in all input parameters. For axial profiles, the input parameters are the nodal axial peaking factors and power fractions. The 48-inch fuel rod is split into 9 nodes of differing sizes shown below. For this project only nodes 4 through 8 were analyzed as they are the nodes that impact safety when perturbed in an unconservative manner.[4]

Table 1: Positions of all nodal points along the fuel rod

Node #	Size [in]	Position
1, 9	8	[+24 : +16], [-16 : -24]
2, 8	6	[+16 : +10], [-10 : -16]
3, 4, 5, 6, 7	4	[+10 : +6], [+6 : +2], [+2 : -2], [-2 : -6], [-6 : -10]

Flux run measurements from previous ATRC were read into Python as a dataframe using the Pandas library. The ±24 positions (x ±24) were calculated via linear extrapolation in accordance with GDE-179.[5] Flux measurements were converted into axial peaking factors by dividing by the average calculated using Equation 1 below.

$$Ave = \frac{2\cdot x(+22) + \frac{x(+20)}{2} + x(+18) + \ldots. + x(-18) + \frac{x(-20)}{2} + 2\cdot x(-22)}{24} \qquad (1)$$

Next, the Power Fractions and nodal Axial Peaking Factor were calculated using Equations 2 and 3.[4]

$$f_i = \frac{P_i}{\sum P_i} = \frac{P_i}{P_{tot}} \qquad (2)$$

$$F_{a-i} = \frac{\frac{P_i}{L_i}}{\frac{P_{Tot}}{\sum L_i}} = \frac{P_i(\sum L_i)}{P_{tot}(L_i)} = fi\frac{\sum L_i}{L_i} \qquad (3)$$

The power fraction and axial peaking factor inputs were assumed to be independent and normally distributed. Equation 4 was used to calculate the total error from all inputs.

$$\sigma^2_{f(x_1, x_2, x_3 \ldots)} = \sum_{i=1} (\sigma_{xi} \cdot \frac{\delta}{\delta x_i}[f(x_1, x_2, x_3 \ldots.)])^2 \qquad (4)$$

The partial derivative of each input is calculated by taking the partial derivative with respect to every variable.

$$\frac{\delta}{\delta x_i}[f(x_1, x_2, x_3 \ldots.)]) = \frac{\delta F_{ai_4}}{\delta x} = \frac{\delta F_{ai_4}}{\delta x_{+24}}, \frac{\delta F_{ai_4}}{\delta x_{+22}}, \frac{\delta F_{ai_4}}{\delta x_{+20}}, \ldots., \frac{\delta F_{ai_4}}{\delta x_{-24}} \qquad (5)$$

In this case each position is a different variable, thus the partial derivative of each position is the summation of partials for each position with respect to that node; this results in a system of equations for each different variable. An example for peaking factor node 4 is shown below.

$$\frac{\delta F_{a4}}{\delta x_{+24}} = \frac{\delta F_{a4}}{\delta x_{-24}} = \frac{-12(x_{+6} + 2x_{+4} + x_{+2})}{(P_{total})^2} \qquad (6)$$

$$\frac{\delta F_{a4}}{\delta x_{[+22,\,+8],[0,\,-22]}} = \frac{-12(x_{+6} + 2x_{+4} + x_{+2})(2)}{(P_{total})^2} (occurs\ 20\ times) \qquad (7)$$

$$\frac{\delta F_{a4}}{\delta x_6} = \frac{\delta F_{a4}}{\delta x_2} = \frac{P_{total}(12) - 12(x_{+6} + 2x_{+4} + x_{+2})(2)}{(P_{total})^2} = 12 * [\frac{P_{total} - (x_{+6} + 2x_{+4} + x_{+2})(2)}{(P_{total})^2}] \qquad (8)$$

$$\frac{\delta F_{a4}}{\delta x_4} = \frac{P_{total}(12)(2) - 12(x_{+6} + 2x_{+4} + x_{+2})(2)}{(P_{total})^2} = 12 * [\frac{(2)P_{total} - (x_{+6} + 2x_{+4} + x_{+2})(2)}{(P_{total})^2}] \qquad (9)$$

After the total error was calculated, the cumulative distribution function (CDF) was taken to find the probability of no unconservative perturbation. From there the complement probability was calculated using Equation 10 to find the probability of unconservative perturbation.

$$P(perturbed) = 1 - P(PowerNodes \cap AxialNodes) \qquad (10)$$

Finally, the relative change was calculated to see how much perturbation each compares to an unperturbed profile.

$$C = \frac{x_2 - x_1}{x_1} \qquad (11)$$

Conclusions and Next Steps

The developed Python code can successfully calculate the axial peaking factor, power fraction, and error from a given dataset of axial measurements. From this data, the probability of perturbation can be calculated and sorted from highest to lowest compared to an unperturbed profile. In the future this can be utilized to:

- Model error based on characteristics that cannot be measured.
- Recommend new thermal-hydraulic analyses based on past axial measurements.
- Re-evaluate legacy experiment characteristics requiring axial measurements.

Acknowledgements

Special thanks to Nathan Manwaring and Andrew Maile for their guidance and feedback for the duration of this project and to the Reactor Engineering group for sharing their knowledge and support.

References

[1] *Advanced Test Reactor*, Idaho National Laboratory, INL, https://inl.gov/atr/.

[2] *ATR Critical Facility Axial Fission Profile Acceptance Criteria (with Assessments of ICE and MICE Experiments)*, Idaho National Laboratory, INL, https://inl-edms.inl.gov/pls/inl_docs/doc_3?f_doc=TRA-ATRC-1546

[3] *ATRC Fuel Element Flux Run Procedure*, Idaho National Laboratory, INL, https://inl-edms.inl.gov/pls/inl_docs/doc_3?f_doc=OP-3.6

[4] *Evaluations of Variations in the ATR Axial Power Distribution on Core Safety Margins*, Idaho National Laboratory, INL, https://inl-edms.inl.gov/pls/inl_docs/doc_3?f_doc=ECAR-2179

[5] *Reduction of ATRC Data*, Idaho National Laboratory, INL, https://inl-edms.inl.gov/pls/inl_docs/doc_3?f_doc=GDE-179