Modern Techniques for Designing Experiments in Zero-Power Reactors

Vincent Lamirand[1]

[1]EPFL, Lausanne, Switzerland

Objectives

To provide participants with modern strategies for designing reactor experiments that deliver reliable, high-value, and reproducible results.

Audience

- Experimental reactor physicists
- Validation and uncertainty-quantification specialists

Main topics

The evolving landscape of nuclear energy, combined with rapid advancements in computational power, detection technologies, and data acquisition systems, has opened new horizons for experimental reactor physics. The need of support for innovative reactor designs and long-term operation demand new experimental design approaches that ensure the value of each experiment, improve accuracy for complex phenomena, and enable rigorous uncertainty quantification, all while ensuring efficient use of limited zero-power reactor facilities.

This workshop will explore the principles and practical applications of modern experimental design in reactor physics. Participants will discuss the challenges and requirements of contemporary experiments, learn about methodologies for optimizing experimental campaigns, and analyze case studies such as the PETALE program in CROCUS. Emphasis will be placed on how to exploit atypical or small-scale reactor systems effectively to generate meaningful validation data.

Over the three hours, the session will alternate between short presentations, interactive discussions, and hands-on exercises demonstrating the trade-offs between experimental complexity, accuracy, and cost. Example topics include noise and modulation techniques, nuclear-data-oriented experiments, new detector types, and uncertainty-driven design optimization.

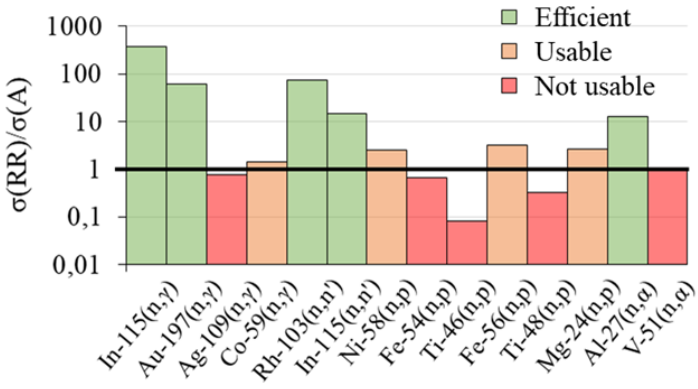

Fig. 1: Example of PETALE experiments: ratios between reaction rate distributions and activity uncertainties for the iron reflector, as used at the design stage.

Materials

Participants are encouraged to bring a laptop equipped with their preferred analysis tools (e.g., Python, MATLAB, or equivalent).