Exploring Computational Notebooks for Education and Training in Nuclear Engineering

Muhammad Ramzy Altahhan[1], Maria Avramova[1], and Kostadin Ivanov[1]

[1]North Carolina State University
Department of Nuclear Engineering, Raleigh, NC 27695, USA

mralttahh@ncsu.edu, mnavramo@ncsu.edu, knivanov@ncsu.edu

doi.org/10.13182/CONTE23-40475

1. INTRODUCTION

The COVID-19 pandemic has proved that virtual distance learning-courses and communication are important for the higher education systems. In such courses, using state-of-the-art methods of education (lecturing, homework problem solving, performing computer projects, training, conducting exams, evaluating learning outcomes, etc.) becomes important and essential to increase the effectiveness of the course delivery. Furthermore, onboarding training for new employees and interns can also benefit from these state-of-the-art delivery methods to complement the "on-the-job" training approach. We explore in this paper such method based on computational notebooks that can hold both text, multimedia (e.g., images), mathematical equations, as well as program computer code that can be executed in the notebook interactively. For this, we will use the new computer language julia [1] to explore such interactive computational notebooks using a Nodal Expansion Method (NEM) solver implemented inside the notebook, while using the IAEA-3D benchmark as the simulated problem. This solver and simulation were discussed previously under the "Julia for Enhancing Nuclear Engineering Simulations (JENES)" project [2]. The next sections then describe the development of a digital computational notebook and explores the benefits of such approach for education and training.

2. DESCRIPTION OF THE JULIA COMPUTATIONAL NOTEBOOK

julia computer language shares the good features of the Fortran and C++ computer languages (e.g., fast execution, and the column-major orientation like Fortran) while avoiding the negative aspects of legacy computer languages (e.g., lack of intrinsic and implicit modern libraries and packaging). This computer language, through its intrinsic and loadable pre/post-processing packages and modern libraries, can be used to develop a complete scientific computing and analysis platform that can be oriented to benefit the nuclear engineering community, especially those for education and training or for cutting-edge research in optimization theory or statistical analysis. Being a dynamic language, julia is a good candidate for computational notebooks and can be integrated into the notebooks to combine code, text, multimedia, and mathematics while being used to develop tools and codes for education and training as well as research.

There are two possible ways to design notebooks in julia. The first one is associated with Jupyter notebooks [3]. The second one, which is used in this summary, is based on the Pluto package available from the julia hub; the official ecosystem that is the entry point for all things julia. The major benefit of Pluto notebooks is them being run in active sessions, which means that changing the value of one variable will be reflected in any cell (the common block of a notebook) using this variable, and there is not then this idea of sequential execution as found in Jupyter. Another benefit is that the Pluto notebook can also act as a package manager. Hence, it is then possible to load any package inside the notebook instead of adding it manually beforehand in the working environment. After adding Pluto to the current environment in julia, it is then straightforward to open Pluto notebook by just writing "Pluto.run()" in the julia executable command window or "Read-