ACTIVE CODING ASSIGNMENTS IN NUCLEAR REACTOR MODELLING

C. DEMAZIÈRE
Chalmers University of Technology
Department of Physics
Division of Subatomic, High Energy and Plasma Physics
SE-412 96 Gothenburg, Sweden
Email: demaz@chalmers.se

In the area of modelling and simulations, the nuclear reactor community relies on many tools. In most of the curricula at universities, emphasis is put on teaching how to use such software to solve practical problems. Due to the complexity of the codes, learning developing input decks, extracting the results and understanding them represent an essential part of the work nuclear engineers and analysts need to get trained on.

Nevertheless, those codes typically rely on many different intertwined algorithms, each based on approximations and limitations. Without proper knowledge of those, an analyst might use a code for application areas not covered by the code. If no warning message is returned by the code in such situations, the analyst might still believe that his/her results are trustable, whereas they are not.

It is thus essential not only to teach using a code but also to make the code users aware of the algorithms and methods on which the code relies, so that the codes are used in their areas of validity.

With this objective in mind, various courses and workshops were developed at Chalmers University of Technology since 2009 on the modelling of nuclear reactor multi-physics. The different areas covered are deterministic neutron transport, thermal-hydraulics, multi-physics coupling and the associated numerical methods. It also resulted in a book recently published on this topic [1]. The courses, workshops and book aim at providing equal level of knowledge in computational reactor physics and thermal-hydraulics, so that the students are able to understand the importance of a faithful modelling of those two inter-related areas of physics in the case of nuclear reactors. This is of special importance for modelling complex reactor transients where such an interplay might play a fundamental role. The different core areas covered are as follows:

— Computational methods for neutron transport at both the pin cell and fuel assembly levels. This area is presented following the solution procedure in fuel pin/lattice codes as much as possible. The various topics covered are: resonance calculations of the cross-sections, determination of the micro-region micro-fluxes, determination of the macro-region macro-fluxes, spectrum correction, preparation of the macroscopic cross-sections for sub-sequent core calculations, where the effect of burnup is also detailed.
— Computational methods used for core calculations. The various topics covered are: treatment of the angular dependence of the neutron flux, treatment of the spatial dependence of the neutron flux, solution procedure for estimating the core-wise position- (and possibly direction-) dependent multigroup neutron flux, methodology used for determining the core-wise space- and time-dependent neutron flux in case of transient calculations.

— Computational methods used for one-/two-phase flow transport and heat transfer. The various topics covered are: derivation of the macroscopic governing equations for fluid flow and heat transfer, derivation of the most typical flow models (two-fluid model, mixture models with thermal equilibrium and specified drift, and the Homogeneous Equilibrium Model), temporal and spatial discretization of the flow and heat transfer models (with emphasis on their stability, consistence, and convergence).
— Neutronic/thermal-hydraulic coupling at the core level. The various topics covered are: general aspects of multi-physics coupling (segregated versus monolithic approaches, coupling terms and non-linearities, information transfer, preparation of the macroscopic material data, spatial coupling), numerical techniques used to solve multi-physics coupling either in a segregated or in a monolithic manner.

Teaching the algorithms used in reactor modelling remain at an abstracted level without having coding assignments along the description of the methods. Moreover, the best approach to understand the algorithms in depth is to implement them. Through code implementation, the students also realize that implementing the described methods represents a major undertaking and that coding computationally efficient algorithms is far from easy. Due to the required implementation efforts, several exercises in form of small projects were created:

— Development of a one-dimensional multi-energy group collision probability solver for modelling fuel pins.
— Development of a one-dimensional multi-energy group diffusion-based solver for modelling reactor cores.
— Development of a one-dimensional Homogeneous Equilibrium Model-based solver for modelling boiling water reactors.
— Development of a one-dimensional one-phase solver for modelling sodium-cooled fast reactors.
— Development of a two-dimensional temperature solver for modelling fuel pins.
— Development of a one-dimensional coupled neutronic/thermal-hydraulic solver for modelling sodium-cooled fast reactors using the Jacobian-Free Newton Krylov method.

To keep the above exercises at a reasonable level of complexity, the dimensionality of the systems was typically reduced to one or two dimensions. Nevertheless, as realistic as possible data were used in terms of, e.g., macroscopic cross-sections and thermal-hydraulic correlations. The non-linear dependence of those data on various fields is also accounted for.

In order to scaffold student learning in the best possible manner, a flipped classroom approach has been progressively introduced since 2013 [2]. The incentive of this pedagogical approach is to deliver the learning elements either asynchronously or synchronously. The asynchronous elements are typically pre-class activities the students have to complete before attending, and for getting prepared to, the synchronous sessions. As a result, such sessions can focus on more complex tasks, such as coding assignments under the close supervision of the teacher. The main advantage of the flipped classroom is to make room for active learning in the classroom. Active student contributions are known to lead to much better learning outcomes [3].

The learning sequence for the students is presented in Fig. 1. The asynchronous elements are made of: (a) studying the textbook, (b) watch short pre-recorded videos lectures, and (c) answer online quizzes to test one's understanding. Whereas the textbook presents the details of the numerical methods, the video lectures are provided as a complement. There are of very short duration and summarize the key points of the concepts presented in the textbook. The videos thus help the students in building their conceptual understanding of the subject. This is further enhanced by the online quizzes.

The essence of the pedagogical approach is to prepare the students to the active learning sessions, which can thus focus on higher order thinking skills, according to Bloom's revised taxonomy for the cognitive domain [4]. Those sessions are structured in the following manner: brief summary of the core concepts presented in the textbook (some of them being used in the subsequent exercises), discussions on the related quiz questions, Q&As, and finally exercises based on coding assignments. The last elements represent the largest contribution to the synchronous sessions.

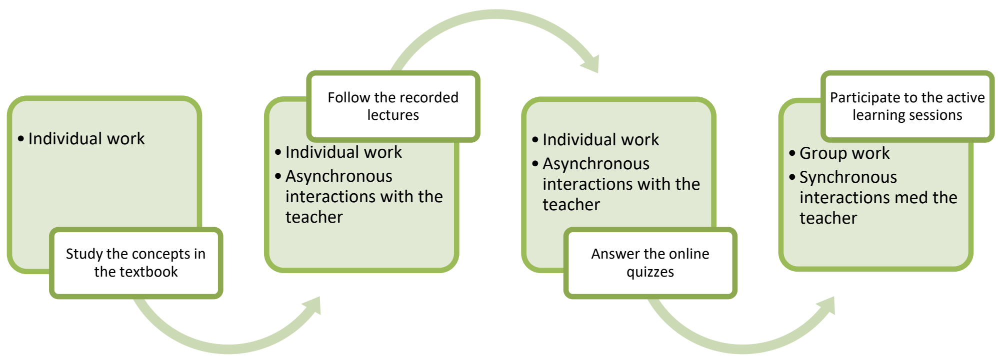

FIG. 1. Illustration of the learning sequence.

The coding assignments are all implemented on MATLAB. In more recent years, a web-based platform, called MATLAB Grader, was used for the assignments. The MATLAB Grader environment seen by the students is illustrated in Fig. 2. MATLAB Grader offers many advantages for coding assignments as compared to letting the students "freely" code the solution algorithms. Incomplete coding templates are provided to the students, with parts of the code already written by the teacher, and the other parts that remain to be completed by the students. This allows to focus the coding assignments on key core concepts the students need to implement, leaving unnecessary details in the already provided lines of codes. Secondly, automatic pre-tests written by the teacher allow the students to test the proper functioning of their code while complementing the missing lines of codes. This contributes to formative feedback the students can immediately incorporate. Learning analytics capabilities are embedded to the platform, thus allowing constantly monitoring student progress, and identifying students needing help. The coding templates also provide the advantage of guiding the students towards good coding practices and efficient implementations. The main advantage of the coding templates lies with the possibility for the teacher to entirely focus during the interactive sessions on helping the students in the implementation of the correct solution and to support them when they most need help. Being able to provide immediate feedback to the students while they are completing the coding assignments is of prime importance if one wants active student participation to the interactive sessions. Student engagement is key to learning.

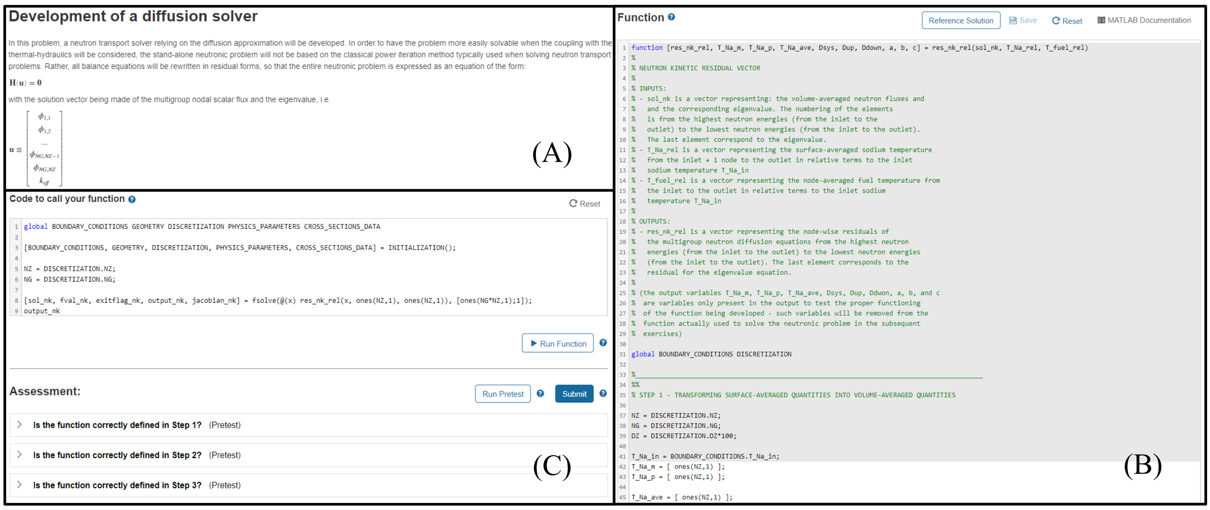

FIG. 2. Illustration of the MATLAB Grader interface seen by the students. Each box represents parts of different areas of the interface. (A) gives a description of the assignment. (B) corresponds to the code the students need to complement. (C) gives access to the MATLAB terminal window and allows testing the proper functioning of the code via pre-tests. The positioning of the different areas is different on the platform and was adjusted in this figure for the sake of clarity.

After complementing the assignments, the students are encouraged to keep their codes alive, especially if the assignments are aligned with the students' on-going research projects. In some cases, students also decided to port their solutions to other environments, such as Python.

REFERENCES

[1] DEMAZIÈRE, C., Modelling of nuclear reactor multi-physics − From local balance equations to macroscopic models in neutronics and thermal-hydraulics, Academic Press/Elsevier (2020).

[2] AKÇAYIR, G., AKÇAYIR, M., The flipped classroom: A review of its advantages and challenges, Comput. Educ. **126** (2018) 334−345.

[3] FREEMAN, S., et al., Active learning increases student performance in science, engineering, and mathematics, Proc. Natl. Acad. Sci. U.S.A. **111** 23 (2014), 8410−8415.

[4] ANDERSON, L.W., et al., A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives, Pearson, Allyn & Bacon (2000).