Annals of Nuclear Energy 115 (2018) 173–185

Contents lists available at ScienceDirect

Annals of Nuclear Energy

journal homepage: www.elsevier.com/locate/anucene

Technical note

Coupled unstructured fine-mesh neutronics and thermal-hydraulics methodology using open software: A proof-of-concept

Vitor Vasconcelos [a,c,*], André Santos [a], Daniel Campolina [a], Germán Theler [b], Claubia Pereira [c]

[a] *Centro de Desenvolvimento da Tecnologia Nuclear, Comissão Nacional de Energia Nuclear, Av. Pres. Antônio Carlos, 6627, 30270-901 Belo Horizonte, MG, Brazil*
[b] *Seamplex, Santa Cruz 205, 2300 Rafaela, Argentina*
[c] *Departamento de Engenharia Nuclear (DEN), Universidade Federal de Minas Gerais (UFMG), Av. Pres. Antônio Carlos, 6627, 30270-901 Belo Horizonte, MG, Brazil*

ARTICLE INFO

Article history:
Received 3 November 2017
Received in revised form 15 January 2018
Accepted 19 January 2018

Keywords:
Neutronics
Thermal-hydraulics
Coupling
Multi-physics
TRIGA
FOSS
Serpent
Simulation
Software

ABSTRACT

The development of a fine mesh coupled neutronics/thermal-hydraulics framework mainly using free open source software is presented. The proposed contributions go in two different directions: one, is the focus on the open software approach development, a concept widely spread in many fields of knowledge but rarely explored in the nuclear engineering field; the second, is the use operating system shared memory as a fast and reliable storage area to couple the computational fluid dynamics (CFD) software *OpenFOAM* to the free and flexible reactor core analysis code *milonga*. This concept was applied to model the behavior of a TRIGA-IPR-R1 reactor fuel pin in steady-state mode. The macroscopic cross-sections for the model, a set of two-group cross-sections data, were generated using the *Serpent* code. The results show that this coupled system gives consistent results, encouraging system further development and its use for complex geometries simulations.

1. Introduction

The goal is to present a coupled neutronics and thermal-hydraulics framework for nuclear reactor calculations. The thermal-hydraulic computations are performed by a free software Computational Fluid Dynamics (CFD) toolbox called OpenFOAM (2015). For neutronic calculations, a free nuclear reactor core analysis code called *milonga* is used to solve the steady-state multi-group neutron transport equation (Theler, 2014). Both codes solve the discretized equations for a fine unstructured mesh using finite volumes method. In this coupled framework, both codes use the same mesh for domain discretization, allowing both codes to solve their problems with the same degree of detail. Data is shared between them through a shared memory segment and the codes are synchronized using POSIX semaphores. This scheme poses no overhead for communication other than any ordinary memory access.

There are many reasons for choosing a computationally demanding method like finite volumes to solve both neutronics and thermal-hydraulics in a coupled manner. The continuous improvements in computers speed and storage capacity have had a deep impact in the way engineers and scientists work on their problems. The nuclear engineering field have been benefiting from computers continuous increasing in processing power, turning former expensive calculation methods into useful and practical tools to solve many different problems in the field. Recently the focus also pointed toward the use of heavy-weight thermal-hydraulic and neutronic codes to solve nuclear reactor problems in a coupled way. It must be remarked that coupled problems have been tackled for a long time (Ivanov and Avramova, 2007), but only recently more computational demanding methods became accessible. These coupled calculations approaches, also called multiphysics (Leppännen et al., 2012; Schmidt et al., 2015; Aufiero et al., 2015; Bennett et al., 2016; Valtavirta et al., 2017), offer an innovative way of modeling the feedback from thermal-hydraulics to neutronics and vice versa. Some of these approaches worth a more than a mention, like the development of a multiphysics OpenFOAM solver called GeN-Foam (Fiorina et al., 2015) capable of solving coupled problems using different fine-meshes in parallel. An extension of this work (Fiorina et al., 2016) focuses on its diffusion

[*] Corresponding author at: Centro de Desenvolvimento da Tecnologia Nuclear, Comissão Nacional de Energia Nuclear, Av. Pres. Antônio Carlos, 6627, 30270-901 Belo Horizonte, MG, Brazil.

E-mail address: vitors@cdtn.br (V. Vasconcelos).

https://doi.org/10.1016/j.anucene.2018.01.021
0306-4549/© 2018 Elsevier Ltd. All rights reserved.

solver and its features and performs verification through comparison to Monte Carlo results.

An important aspect of computer hardware evolution is the respective evolution of the associated software. The effects of changes in software conception and development in order to take advantage of hardware improvement are often neglected, especially in the nuclear engineering field. Despite the healthy discussion on the safety and the advantages and disadvantages of open source software in many fields (Androutsellis-Theotokis et al., 2010), it is impossible to deny the influence of free software in the society nowadays (Williams, 2010). However, more than software itself, free and open source software bring a new culture of software development and, as one would expect, this cultural change reached scientists and engineers in the nuclear field. It goes from Monte Carlo criticality calculations like proposed by the OpenMC project (Romano and Forget, 2013), passing through deterministic approaches for neutron calculations (Boyd et al., 2014), to full nuclear cycle calculations (Huff et al., 2016). A special mention must be made to the MOOSE framework (Gaston et al., 2009) which goes further on solving coupled problems by providing a complete framework in which users can develop tightly coupled multiphysics solvers taking advantage of built-in parallel implementation and many other features. All of these software share a common aspect: they are all open, freely available and, most important, their development is decentralized. This means that and anyone interested in participating can download it and start a development branch to make changes and improvements. Once the developer finishes its work, its contributions are submitted to the main development branch and project maintainers take care of checking and decide if contribution is merged back into the official code distribution or not. This development cycle is, probably, the main strength of an open software project. These are solid reasons to make software presented is this paper open and free.

In this paper, the objective is to the develop a framework for coupled thermal-hydraulic and neutronic calculations which can be freely distributed, used, modified and improved by anyone interested in performing nuclear reactor calculations. The following sections present details of the coupled system developed in this work, which have the following features:

- Neutronics and thermal-hydraulics are calculated using a finite-volumes approach, in steady-state mode, and both pieces of software use the same domain discretization for the computations. Data is exchanged using a *shared-memory* approach. Despite being a well-known method for sharing data in thread-based systems, this form of data access is not widely used in neutronics and thermal-hydraulics coupling. It extends the innovative *shared-memory* approach proposed by Theler et al. (2013) to perform coupled calculations.
- It uses two free and open source codes for neutronics and thermal hydraulics coupling aiming to bring to the light the discussion of the use of open (and free[1].) software in the nuclear engineering field.

2. Models and methodology

In order to test the coupled framework, a geometry representing one TRIGA IPR-R1 reactor fuel pin was modeled. It is worth noting that "to test" in the context of this work means to guarantee that:

1. data is properly exchanged from thermal-hydraulics to neutronics and the vice versa;
2. calculations are correctly carried and numerical convergence is achieved in both codes.

In other words, the coupling methodology and the correctness of implementation are demonstrated through three-dimensional multiphysics calculations. Fig. 1 depicts a basic schematic for the coupling methodology.

Cross-sections processing is performed using the *Serpent Monte Carlo* code (Leppänen et al., 2015), following this code abilities in generating N groups cross-sections. It should be noted that even though *Serpent* is distributed along with its source code, its specific license is not compatible with the GNU Public License (GNU license, 2007) nor with other licenses classified as open-source compliant, and thus is not cited as an example of free software.

Before diving into the details of the coupled calculations, it is useful to describe the models and methodology used for each code in standalone mode. The coupled code has a special feature concerning data exchange, but the way both *OpenFOAM* and *milonga* runs is preserved.

2.1. Cross-sections generation

In order to be able to utilize *milonga* to calculate the power distribution in the modeled fuel pin it is mandatory to have the macroscopic cross-sections for each group so they can be used in the neutron diffusion approximation equation. These input values consist usually of absorption, fission neutron production and group scattering cross-sections In this work, a set of two-group cross-sections is generated for three different materials at different temperatures, corresponding to the modeled fuel element expected temperatures operation range, considering the fuel composition at the beginning of life TRIGA IPR-R1. A simplified model of a fuel pin was built which keeps the fissile material/moderator ratio of the entire TRIGA IPR-R1 core.

Considering this ratio, this model was loaded in the Serpent Monte Carlo code and two-group cross-sections were generated accordingly, following an established methodology for cross-section generation in Serpent (Leppänen et al., 2016). Sets of cross-sections were generated following the aforementioned cross-section generation methodology for three different materials representing fuel, cladding and coolant at the tabulated temperatures presented in Table 1. The composition of the materials are depicted in Table 2. The Serpent code (version 2.1.29) reads a properly chosen set of cross-sections representing the materials of the model. The main cross-sections data come from ENDFB-7 library originally distributed with Serpent 1.1.7. Another set of cross-sections previously prepared for simulations of zirconium hydride in the TRIGA fuel is used, based on ENDFB-7.

The calculations for obtaining cross-sections at the desired tabulated temperatures were made using the Doppler-broadening feature implemented in Serpent for this purpose. Diffusion coefficients, absorption cross-sections, scattering cross-sections and energy released by fission were obtained in this way. After this procedure, sets of coefficients for each material at four different temperatures are available to be used by *milonga*. Table 3 shows the parameters used by *milonga* and which diffusion equation coefficient they physically represent.

These data are written in *milonga* format (i.e. a plain ASCII file with column-wise data) as a one-variable function dependent on temperature. Since *milonga* has cell-wise information loaded with the mesh, the temperature of each cell becomes argument of the one variable function dependent on temperature. If the temperature is not exactly tabulated, *milonga* provides different interpolation algorithms (linear, splines, akima, steffen, etc.). In other words, for each neutronic calculation, a set of coefficients for diffusion equation is calculated based on interpolated values of cell

[1] "Free" as in "free speech" not as in "free beer".