**COUPLED UNSTRUCTURED FINE-MESH NEUTRONICS AND THERMAL-HYDRAULICS METHODOLOGY USING OPEN SOFTWARE: A PROOF-OF-CONCEPT**

**Vitor Vasconcelosa,c\*, André Santos****a, Daniel Campolinaa, Germán Thelerb, Claubia Pereirac**

aCentro de Desenvolvimento da Tecnologia Nuclear, Comissão Nacional de Energia Nuclear, Av. Pres. Antônio Carlos, 6627 - 30270-901 Belo Horizonte, MG, Brazil

bSeamplex, Santa Cruz 205, 2300, Rafaela, Argentina

cDepartamento de Engenharia Nuclear (DEN), Universidade Federal de Minas Gerais (UFMG), Av. Pres. Antônio Carlos, 6627 - 30270-901 Belo Horizonte, MG, Brazil

**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.

\*Corresponding author. Tel: +55 31 3069 3466; fax: +55 31 3069 3400

**E-mail address: vitors@cdtn.br, (Vitor Vasconcelos)**

**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 multigroup 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 ordinarymemory 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, Schimidt et al., 2015, Bennett et al., 2016 and 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 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 and Stallman, 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 (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]](#footnote-2)) software in the nuclear engineering field.

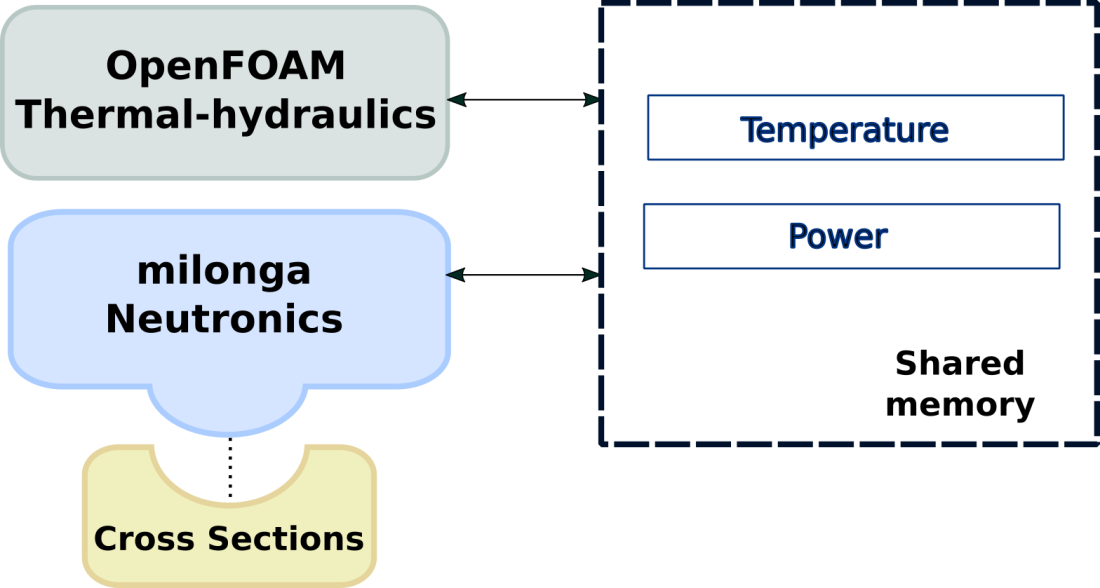
**2. MODELS AND METHODOLOGY**

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. Figure 1 depicts a basic schematic for the coupling methodology.



**Figure 1 - Coupling methodology schematic.**

Cross-sections processing is performed using the *Serpent Monte Carlo* code (Leppänen etal, 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.

**Table 1. Temperatures for fuel, cladding and moderator (Veloso, 2005)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | T1 [K] | T2 [K] | T3 [K] | T4 [K] |
| fuel | 300 | 400 | 500 | 600 |
| cladding | 300 | 396 | 403 | 410 |
| water/moderator | 300 | 309 | 317 | 341 |

**Table 2. Material composition for cross-sections generation with Serpent 2.1.29 (\*Materials code changes for the fuel temperature of 600K from 03.c to 06.c since the tabulated data is available in the library used).**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Material | Material code\* | Atomic fraction |
| fuel | H (in zyrconium hydrade) | 1001.03c/06.c | 3.7525x10-2 |
| Zr | 4000.03c/.06c | 3.7727x10-2 |
| U235 | 92235.03c/.06c | 2.5744x10-4 |
| U238 | 92238.03c/06c | 1.0167x10-3 |
| cladding | Al | 13027.03c | 6.0261x10-2 |
| water/moderator | H | 1001.03c | 6.6653x10-2 |
| O | 8016.03c | 3.3327x10-2 |

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 temperatures. Each cell in the mesh can have its own set of coefficients if the temperature is not uniform through the spatial domain.

**2.2 Thermal-hydraulics**

The *OpenFOAM* C++ library offers numerical solvers for many common problems in continuum mechanics and utilities for data pre and post-processing. In order to solve a one-phase steady state fluid flow including conjugated heat transfer from three different materials, the chtMultiRegionSimpleFoam solver (*OpenFOAM*, 2015) was chosen.

**Table 3. Diffusion approximation coefficients by domain regions and groups.**

|  |  |  |
| --- | --- | --- |
| Group | Coefficients | Parameter |
| Group 1: > 0.625MeV | Diffusion coefficient | D1 |
|  | Absorption cross-section | ΣA1 |
|  | Scattering cross-section | ΣS1.2 |
|  | Neutrons per fission \* Fission cross-section | νΣF1 |
| Group 2: < 0.625Mev | Diffusion coefficient | D2 |
|  | Absorption cross-section | ΣA2 |
|  | Neutrons per fission \* Fission cross-section | νΣF2 |

The fluid dynamic is governed by the momentum, continuity and energy conservation equations. The chosen solver separates fluid regions from solid regions and then solves the appropriated equations for each type of region, taking into account the thermal and physical properties of each material, defined separately for each region.

Differently from common numerical tools, *OpenFOAM* allows equations assembly directly using objects. The energy equation is implemented in a flexible way, allowing for calculations using enthalpy or internal energy.

In the case of solid regions, the energy equation is solved based on enthalpy and after that the corresponding thermal-physical properties are corrected. The energy equation was modified in order to accommodate a source-term to be used by the neutronic coupling scheme. Section 2.3 is entirely dedicated to the coupling methodology and aspects of its work.

The fluid dynamic RANS equations for momentum, continuity and energy conservation were solved. The two equations standard *k-ε* turbulence model (Launder and Spalding, 1974) was selected to calculate the turbulent viscosity. The model assumes that the turbulence viscosity is related to the turbulence kinetic energy (*k*) and dissipation (*ε*). The set of discretized equations are embedded in the solver and solved using SIMPLE algorithm.

Thermo-physical properties are defined for each region as a function of temperature or constant based on the work by Veloso (2015), as presented in Table 4. The coolant properties were considered constant to reduce computational cost as the temperature variations where expected to be small and not affect the proof-of-concept.

**2.3 Neutronics**

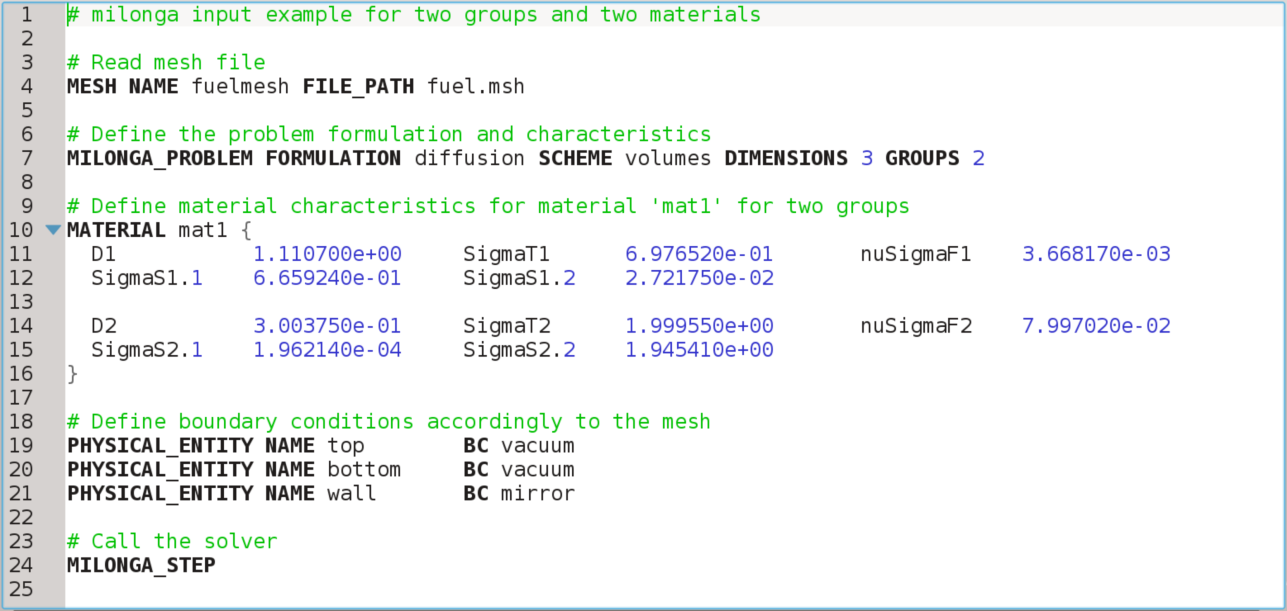
The *milonga* nuclear code is a free and open source piece of software released under GNU license. It heavily relies on other well-known GNU free and open libraries, like PETSc (Balay et al., 2016), SLEPc (Hernandez et al., 2005) and the GNU Scientific Library (Galassi et al., 2009), bringing to it the robustness of these established software and keeping the principles of freedom of software use that GNU encourages. It solves the steady-state multigroup neutron transport equation, either using the diffusion approximation or the discrete ordinates *Sn* method. It can discretize the spatial coordinates using either finite-element or finite-volumes schemes. This ability to use finite-volumes discretization schemes over unstructured grids makes possible, regarding the present problem, to have the same mesh for both thermal-hydraulics and neutronics calculations. With identical meshes, the dependency between fields in each tool can be solved without any additional step and with the same degree of details. Despite limitations of the diffusion approximation under some circumstances (Trahan, 2014), it is the method used in this work due to its faster execution time compared to the *Sn* method. Moreover, with a proof-of-concept methodology in mind, any changes in power and flux can be observed using the diffusion approximation for multigroup steady-state calculations. Despite the cross-sections dependence on temperatures, the diffusion equation can be formulated as linear since the coefficients are kept constant during each step in the iterative calculation.

**Table 4. Thermo-physical properties.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Material | Density [kg/m3] | Specific heat [kj/kg.K] | Conductivity [W/m.K] | Viscosity [μPa.s] |
| fuel | 2705 | 0.892 +  4.44361x10-4*T* + 3.632x10-8*T*2 | 223.7 -  4.756x10–2*T* +  1.0215x10-5*T*2 -  1.8887x10-11*T*3 | - |
| cladding | 6280 | 0.294 +  6.196x10-4*T* -  2.748x10-9*T*2 +  1.354x10-11*T*3 | 22.872 -  4.3131x10-2*T* +  1.124x10-4*T*2 -  1.0039x10-11*T*3 | - |
| coolant | 995 | 4.18 | 0.62 | 797 |

In a standalone case, *milonga* works by reading an input file which describes the problem to be solved. The simplest case for the present problem consists on the basic problem configuration: read the mesh, define the numerical formulation and choose the number of neutron groups for the problem. The next step is to define the macroscopic cross sections for each material (that may further depend on other functions of space such as temperature or poison concentration) matching the materials defined in the mesh. These cross sections are used in the solution of the diffusion equation and must match the material characteristics, like the diffusion coefficients, absorption macroscopic cross sections and scattering cross-sections. These definitions may be given as algebraic expressions of x, y and z, as point-wise defined functions of x, y and z or as a combination of both methods. After the definition of materials, the appropriate boundary conditions are set matching the physical entities defined in the mesh. In order to solve the problem, a simple call command builds and solves the eigenvalue problem. Figure 2 shows a simple input file for *milonga* for the simplest case of spatially-uniform macroscopic cross sections. The material definitions (lines 10-16) are key: these values can be constants like in the example above but they can also be function of x, y and z. Had the cross sections depended on space, algebraic expressions would have appeared instead of constant values. This feature of *milonga* plus the fact that it natively includes instructions to access POSIX named shared-memory segments and semaphores, makes coupling milonga to external codes straightforward.

It is important to mention that *milonga* has a wide range of commands and primitives, ranging from meshing pre-processing to output primitives aimed to post-processing in default formats for data visualization software.

**Figure 2 - A simple input file for *milonga* with line number counting.**

Another key feature of *milonga* which makes external coupling straightforward is that general multidimensional functions (and in particular functions that depend on space) can get data in a grid-wise manner from a set of arbitrary n+1-tuples containing the n independent variables plus the scalar data. The final feature is that these n+1 tuples can be read at run-time from a variety of computational resources, including input files, binary files and shared-memory objects. In particular, to have the corresponding values of cross-sections (and all other coefficients for the diffusion approximation) for each cell depending on other local properties that vary from cell to cell (i.e. temperatures, poison concentration, void fraction, etc.), a properly set of neutronic macroscopic cross sections had to be generated prior to performing the neutronic computation itself.

**2.4 Coupling scheme**

The coupling scheme involves two perspectives: one from thermal-hydraulic, which is *OpenFOAM*, and the other from neutronic, which is *milonga*. The coupling scheme is presented by showing code modifications and technical aspects from the point of view of *OpenFOAM* and then from the perspective of *milonga*, in a separated way.

**2.4.1 Thermal-hydraulics**

The free software characteristic of *OpenFOAM* allows changes and modification of its source code, as mentioned before. So, after properly choosing the conjugated heat transfer solver for the thermal-hydraulic and neutronic problems, it had to be modified in order to attain to the objectives of this work. There were two major[[2]](#footnote-3)\* modifications of the solver:

* the addition of a source-term to the energy equation of a solid;
* the addition of the data structures for communicating through shared memory to the neutronics code.

It is essential to note that, as a project definition, both *OpenFOAM* and *milonga* were set to use the same unstructured mesh. It is enforced to make clear that there is no mapping function and, therefore no error associated to data transformation from one mesh to another in the communication and calculations processes.

To add a source-term for the energy equation of the fuel solid region a new scalar field was defined in the input files to represent the power in the fuel region. The source-code containing the energy equation was modified accordingly to include a source-term. If the field is not present, a null default scalar field is created making the source-term zero.

The coupling between codes was implemented using shared memory taking advantage of the already coded features of *milonga* to make use of it. Shared memory, in computer science, is a piece of memory that can be simultaneously accessed by different programs running in different user spaces. In order to keep read and write access to the shared memory consistent, a set of semaphores was implemented in the *OpenFOAM* solver together with the structures to detect and read the shared memory (Theler et al., 2013).

The methodology for the coupled communication works as follows:

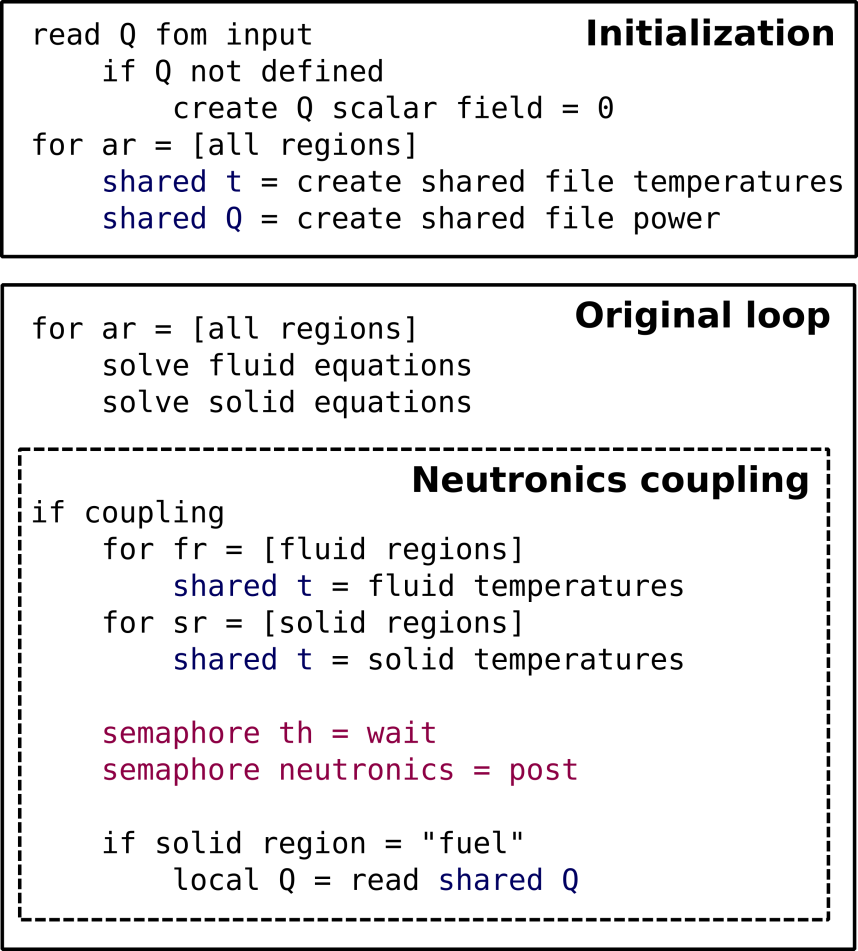
1. the neutronics program *milonga* starts, creates the shared memory and halts until it is able to read the semaphore from thermal-hydraulic;
2. *OpenFOAM* checks if shared memory is present and, if yes, runs the thermal-hydraulic simulation with initial conditions a fixed number of iterations and then signs that to neutronics.
3. *milonga*, when getting the good value for the semaphore, reads temperatures from shared memory, interpolate cross-section data to the current temperature and calculates flux and power. When done, it signs to thermal-hydraulics through its semaphore that it can resume its calculations.
4. From now on, thermal-hydraulic will use the power read from shared memory as a source-term in the energy equation for the solid fuel region. The calculations remain from following steps 3 and 4.

The whole process goes until convergence of thermal-hydraulics. Figure 3 presents the coupling algorithm from the point of view of thermal-hydraulics in a schematic way. Blue commands are related to shared memory manipulation while purple commands control the access to shared memory.

There are some limitations and advantages in the way that the coupling is implemented. For now, a fixed ratio of thermal-hydraulics/neutronics iterations is implemented. In this work a number of 100 iterations of *OpenFOAM* were defined per call to neutronics, based on the maximum number of sub-iterations for neutronics and thermal-hydraulics used by Jareteg (Jareteg et al., 2014). The use of semaphores controls the inter codes communication.

As an advantage, the use of shared memory poses no overhead in data access greater than any other type of memory access. Some works on coupled thermal-hydraulic and neutronic using external files (Ivanov and Avramova, 2007 and Hummel and Novog, 2016) have access time orders of magnitude higher than the achieved using shared memory (Theler et al., 2013). It must be noted that tools for shared memory communication are available as standard in most of the common operational systems available nowadays.

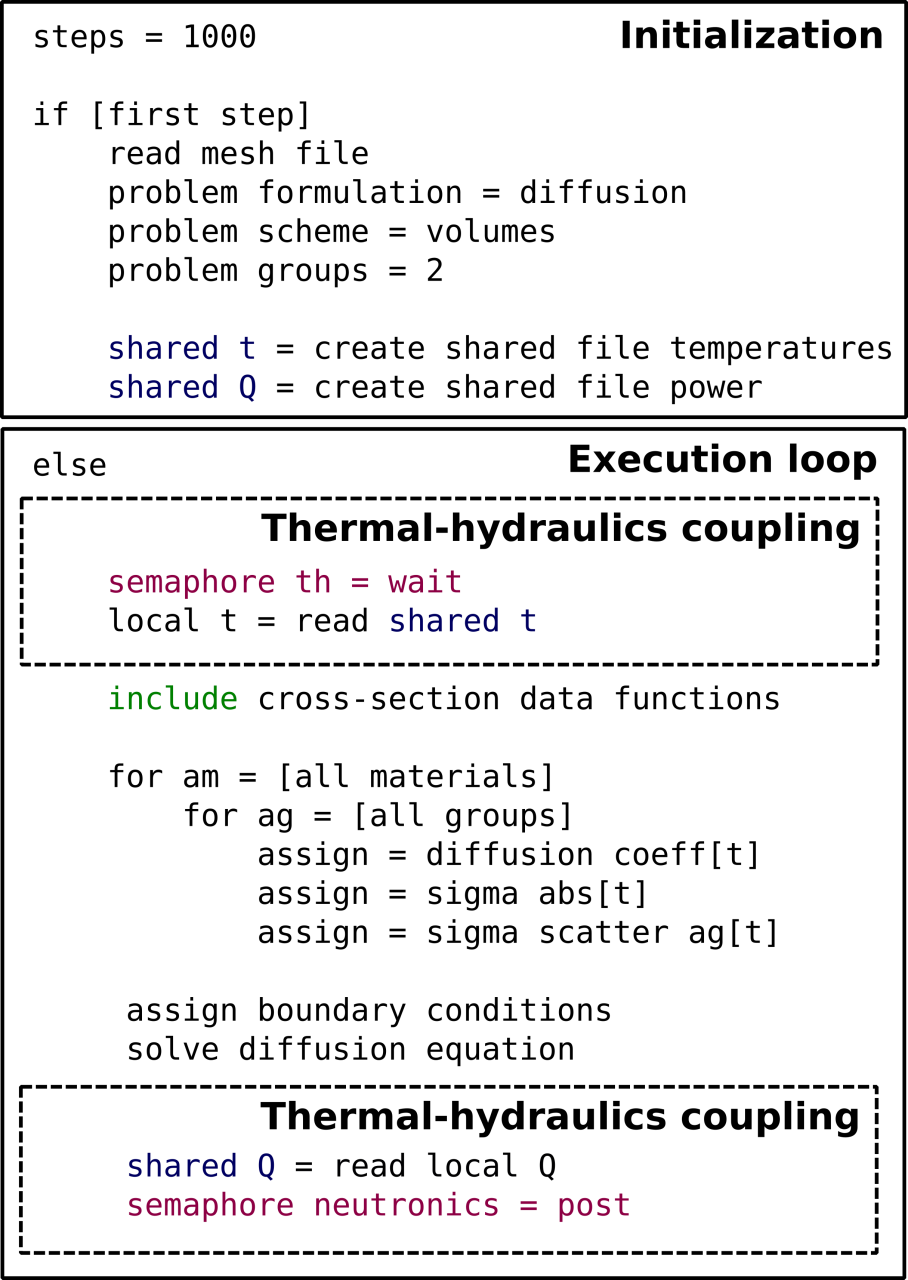
The other main advantage of this coupling scheme is the use of the same mesh for both codes. This allows calculations with the same level of detail for thermal-hydraulic and neutronic also avoiding the use of mapping functions (Silva et al., 2015), which inevitably introduces some approximation errors. There are some elegant solutions for mesh mapping functions which can be consistently applied (Jareteg et al., 2015, Richard et al., 2015 and Schimidt et al., 2015), but also introducing some kind of restriction to the solved problem. Even though *milonga* was designed with the ability to cope with external codes using different meshes, the fact that the neutronics and the CFD code share the same unstructured grid is an unique feature that makes it outstand over the other available core-level neutronic codes.

**Figure 3 - Thermal-hydraulic coupling algorithm schematic. Blue commands are related to shared memory manipulation while purple commands control the access to shared memory.**

**2.4.2 Neutronics**

The coupling methodology establishes *milonga* as responsible for creating and destroying the shared memory segment used to exchange data, reading the temperature distribution from the thermal-hycraulics code, performing the computation of the neutron flux distribution and getting back the power distribution into the OpenFOAM solver. Figure 4 shows a pseudo-code representing the coupling scheme algorithm from *milonga*'s perspective. Commands in blue are related to shared memory manipulation, commands depicted in purple show access to shared memory and the command in green includes definitions in external files.

After solving the diffusion approximation and having the function for the flux, *milonga* has a primitive to integrate the flux over the mesh for a reference power. This value is defined previously in *milonga*'s input file. The power obtained for each cell is thus written in the shared memory, making the power available for the *OpenFOAM* solver. After this point, the coupling cycle is completed and will continue until thermal-hydraulic calculations converge. It is worth noting that *milonga* does not need any modification or re-compilation in order to be coupled using shared memory. Everything is done through the input file and interpreted during run-time.

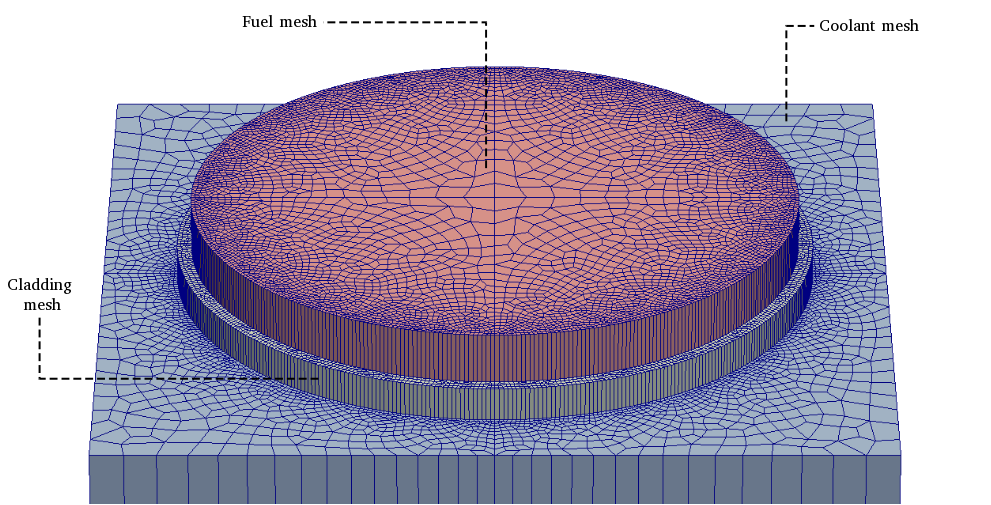
**Figure 4 - Neutronic coupling algorithm schematic.**

**2.5 Physical and numerical model**

The coupled methodology was assessed simulating a fuel element of the TRIGA IPR-R1 reactor, located at the Centro de Desenvolvimento da Tecnologia Nuclear (CDTN), Belo Horizonte – MG, Brazil. The TRIGA fuel has strong reactivity sensitivity with temperature due to the presence of zirconium hydride. This characteristic allows that small temperature differences result in perceivable neutronic variations which makes this model adequate to coupling assessment.

This model consists of the geometry of a pin cell representing an aluminum fuel with no gap considered. The fuel diameter was kept as the real fuel element diameter and the cladding thickness extended to occupy the space of the gap. The model length is the same of the active region of the real fuel element. In order to have a realist neutronic behavior, the ratio of fissile material and moderator for the model matches the ratio of the full reactor. No other solid structures like samarium disks or structural elements were considered, being the model identical at any axial slice. Model sizes are 1.78 cm for fuel pin radius, 1.865 cm for the external cladding radius, 4.57 cm for coolant edges and 35 cm length in axial direction.

An axially extruded unstructured mesh was generated for this work using *Gmsh* (Geuzaine and Remacle, 2009) which is a free software for two or three-dimensional unstructured mesh generation compatible with both *OpenFOAM* and *milonga*. A 2D mesh representing a traversal section of the domain was generated with a global maximum mesh size of 4.0 mm for the whole domain and a 0.3 mm local sizing for the thinnest region of the domain, represented by the cladding, with a growing ratio of 1.2. These parameters resulted in a smoothly grown mesh for the 2D plane. The 3D mesh was completed by the extrusion of 35 layers. The total number of mesh elements is 346,675. A cut showing each material of the generated mesh can be seen in Figure 5. An isometric view of it indicating fluid and solid regions together with boundaries in shown in Figure 6.

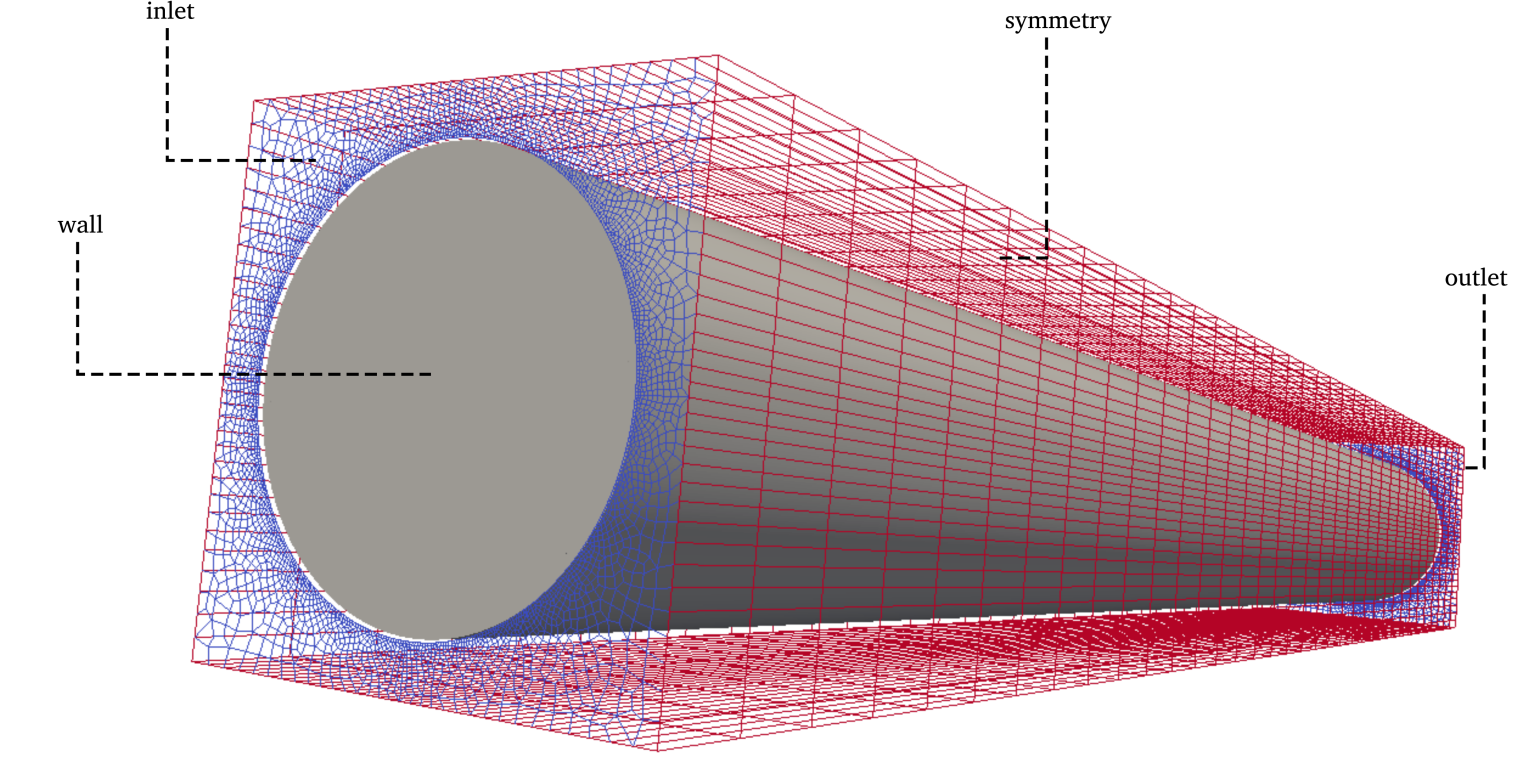
**Figure 5 - Horizontal mesh discretization slice exploded showing the three regions used by thermal-hydraulic and neutronic.**

A simplified mesh problem was selected for this work due to computational limitations. Simulations were performed on one computer as, at the present time, the software is not implemented for parallel simulation.

The setup for thermal-hydraulics and neutronics presented in the following sections is the same for all simulations, with the exception of the initial conditions for the power by volume.

**2.5.1 Thermal-hydraulic boundary conditions and numerical parameters**

There are four different boundary conditions for the thermal-hydraulic simulation, being an adiabatic wall for fuel and cladding extremities, a symmetric condition for the coolant lateral faces and an inlet and outlet in the coolant extremities. In Figure 6 the set of external boundary conditions are depicted. The symmetric boundaries for fluid region are depicted as wireframe in red and inlet and outlet are fluid surfaces shown in blue wireframe. The problem setup is presented in Table 5.

**Figure 6 - External boundaries for the CFD problems.**

**Table 5. Main boundary conditions used for thermal-hydraulic simulations**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Region | Field | Boundary | Type | Value |
| coolant | T | inlet  outlet | fixedValue  zeroGradient | 300 K  - |
|  | U | inlet  outlet | fixedValue  zeroGradient | 0.1 [m/s]  - |
|  | k | inlet  outlet |  |  |
|  | eps | inlet  outlet |  |  |
|  | p\_rgh | inlet  outlet | zeroGradient  fixedValue | -  0 [kg/m.s² =Pa] |
| cladding | T | wall | zeroGradient | - |
|  | Q | wall  internalField | calculated  uniform | -  0 [W/m³] |
| fuel | T | wall | zeroGradient | - |
|  | Q | wall  internalField | calculated  uniform | -  2.276e+07 [W/m³] |

The symmetry boundary is a special type of boundary condition. It is defined in the mesh file as symmetry and it works as if the other side of the boundary has an identical field. In a tridimensional mesh, the symmetry boundary is a surface. Another special kind of boundary is the surface between regions. These surfaces are created during the split of the general mesh and are assigned specific boundary conditions accordingly to the type of region by *OpenFOAM*, mostly in a default way.

However, one of these boundary conditions worth a special mention: to allow heat transfer between adjacent regions, a temperature coupled boundary condition must be defined at all adjacent surfaces between regions. This boundary condition gets information from thermophysical properties files defined for every region and uses this data to calculate the heat flux. More information on boundary conditions available in *OpenFOAM* can be found in its documentation (*OpenFOAM*, 2015). The fluid flow is considered turbulent based on IPR-R1 TRIGA reactor usual operating condition (Veloso, 2005) and the standard κ-ε model (Launder and Spalding, 1974) is used for modeling it.

Simulations were performed using second order discretizations schemes for all equation terms except the divergence, as shown in Table 6. A first order scheme was used for divergence to accelerate and facilitate the numerical convergence, as the results are for a proof-of-concept and an added false diffusion will not interfere with the study’s conclusion.

*OpenFOAM* (2015) allows the use of different discretization schemes for each region and also for each equation term. Utilized matrix solvers and convergence criteria are presented in Table 7.

**Table 6. Discretization schemes.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Type of scheme | | | | |
| Region \ Eq. term | gradient | divergence | laplacian | interpolation | normal do cell face |
| *Coolant* | Least Squares | Bounded Gauss upwind | Gauss linear limited 1.0 | linear | limited 1.0 |
| *Cladding* | Gauss linear | - | Gauss linear uncorrected | linear | uncorrected |
| *Fuel* | Gauss linear | - | Gauss linear uncorrected | linear | uncorrected |

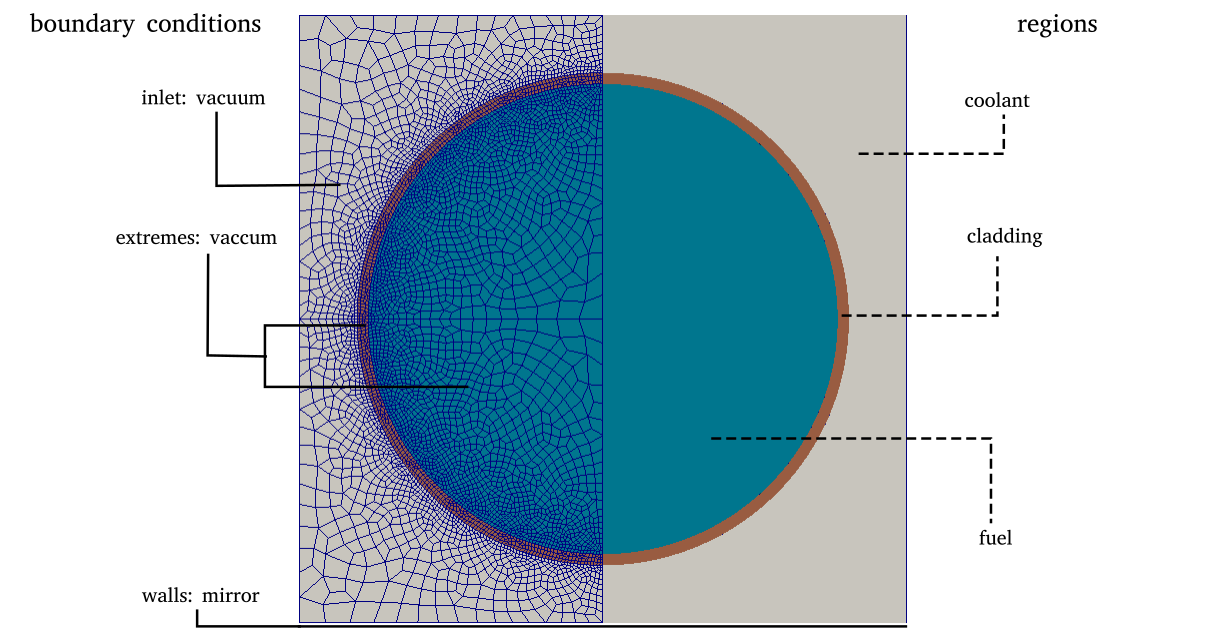
A simulation using the same geometry, mesh and setup was performed using *milonga* to provide a non-uniform power distribution of power to be used as input by *OpenFOAM* (Silva et al, 2016). The results of this work show that the setup is robust to simulations in the range of power distributions encompassing the value of power used in the current work, thus avoiding eventual calculations errors derived from boundary and initial conditions.

**Table 7. Solver settings for each field.**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type of solver | Convergence criteria | Under-relaxation |
| *h* | Preconditioned bi-conjugate gradient | 10⁻⁵ | 1.0 |
| ρ | Preconditioned bi-conjugate gradient | 10⁻⁵ | 0.7 |
| p\* | Generalized geometric-algebraic multi-grid | 10⁻⁵ | 0.7 |
| p\* final | Generalized geometric-algebraic multi-grid | 10⁻⁶ | 0.7 |
| U | Preconditioned bi-conjugate gradient for asymmetric matrices | 10⁻⁵ | 0.7 |
| κ |
| ε |

**2.5.2 Neutronic boundary conditions**

Milonga deals internally with boundary conditions between regions, conveniently sparing the user from an extra amount of work. For the present problem, it is enough to define boundary conditions for the external surfaces which are chosen to represent neutrons leaking from the top and bottom of the model and a symmetric condition at the sides, simulating an infinite lattice. Table 8 shows the boundary conditions used for the simulation, noting that neutrons escape in the top and bottom of the model, while Figure 7 presents the regions of each material and associated boundary conditions in a top view. It is possible to see at right, regions consisting of homogeneous materials. At left, the mesh over domain showing boundary conditions.

**Figure 7 - Top view of neutronic domain.**

**Table 8. Boundary conditions.**

|  |  |  |
| --- | --- | --- |
| Boundary | Type | Corresponding region |
| inlet | Vacuum | Coolant |
| outlet | Vacuum |
| extremes | Vacuum | Cladding and  Fuel |
| walls | Mirror | Coolant |

*OpenFOAM* was run in standalone mode with a constant power value 7.93 kW. This power value is chosen based on the proportional power of the hottest fuel element of TRIGA IPR-R1 (Veloso, 2005) considering the reactor operating at 200 kW. After the simulation, the average temperature for each material is calculated. These averaged temperatures are shown in Table 9.

**3. PROOF-OF-CONCEPT**

The proof-of-concept of the proposed coupled methodology is performed based on different coupled and non-coupled simulations:

1. Non-coupled case: *OpenFOAM* and *milonga* are run separately with constant distributions for temperature and power. *Milonga* is fed with a fixed set of cross-sections. These cases are defined to obtain physical results. These results are the neutronic and thermal-hydraulic references for the coupled cases.
2. Coupled case: in this case, *OpenFOAM* writes cells temperatures to the shared memory segment while reads power from it and *milonga* reads temperatures from the shared memory segment and writes power to it. In order to store data in a shared-memory segment, a 1d-vector is made up from the mesh with all the scalars, and this vector is written/read in/from the shared memory segment.

**Table 9. Averaged temperatures [K] for materials after standalone thermal-hydraulic simulations**

|  |  |
| --- | --- |
| Material | Simulation power 7.93 [kW] |
| fuel | 423.0 [K] |
| cladding | 375.4 [K] |
| water/moderator | 310.1 [K] |

These temperatures were used to feed *milonga* for a non-coupled simulation. The set of temperatures was used to generate fixed cross-sections, interpolated from the data obtained previously from Serpent. The non-coupled simulation gives a power distribution which is then used by *OpenFOAM* as initial condition. This non-coupled simulation used as reference by the coupled simulations can be seen, in fact, as a special case of an initial condition externally coupled calculation. The use of this somehow detailed initial condition field for the power distribution allows to focus the assessment of the coupling methodology on the difference between the non-coupled and coupled simulations, since the initial condition for thermal-hydraulics is already the solution of the diffusion equation for neutrons for the fixed set of cross-sections.

Figure 8 shows temperature and power radially distributed for both coupled and non-coupled simulations while in Figure 9 the axial temperature and power distributions are depicted. Results show an expected behavior: flattened profiles for temperature and power for coupled calculations, indicating the coupling was successfully achieved. The steps presented in the radial plot in Figure 8 are expected and due to the interface between different materials. The difference between coupled and non-coupled calculations shows that locally variations in temperature can have a sensible influence in cross-sections variations and, therefore in the power distribution. This behavior is the expected for the well-known TRIGA fuel elements (Cheng and Roberts, 2017).

|  |  |
| --- | --- |
|  |  |

**Figure 8: Temperature and power in radial directions for coupled and non-coupled simulations.**

|  |
| --- |
|  |
|  |
| **Figure 9: Power and temperature profiles in axial direction for coupled and non-coupled simulations.** |

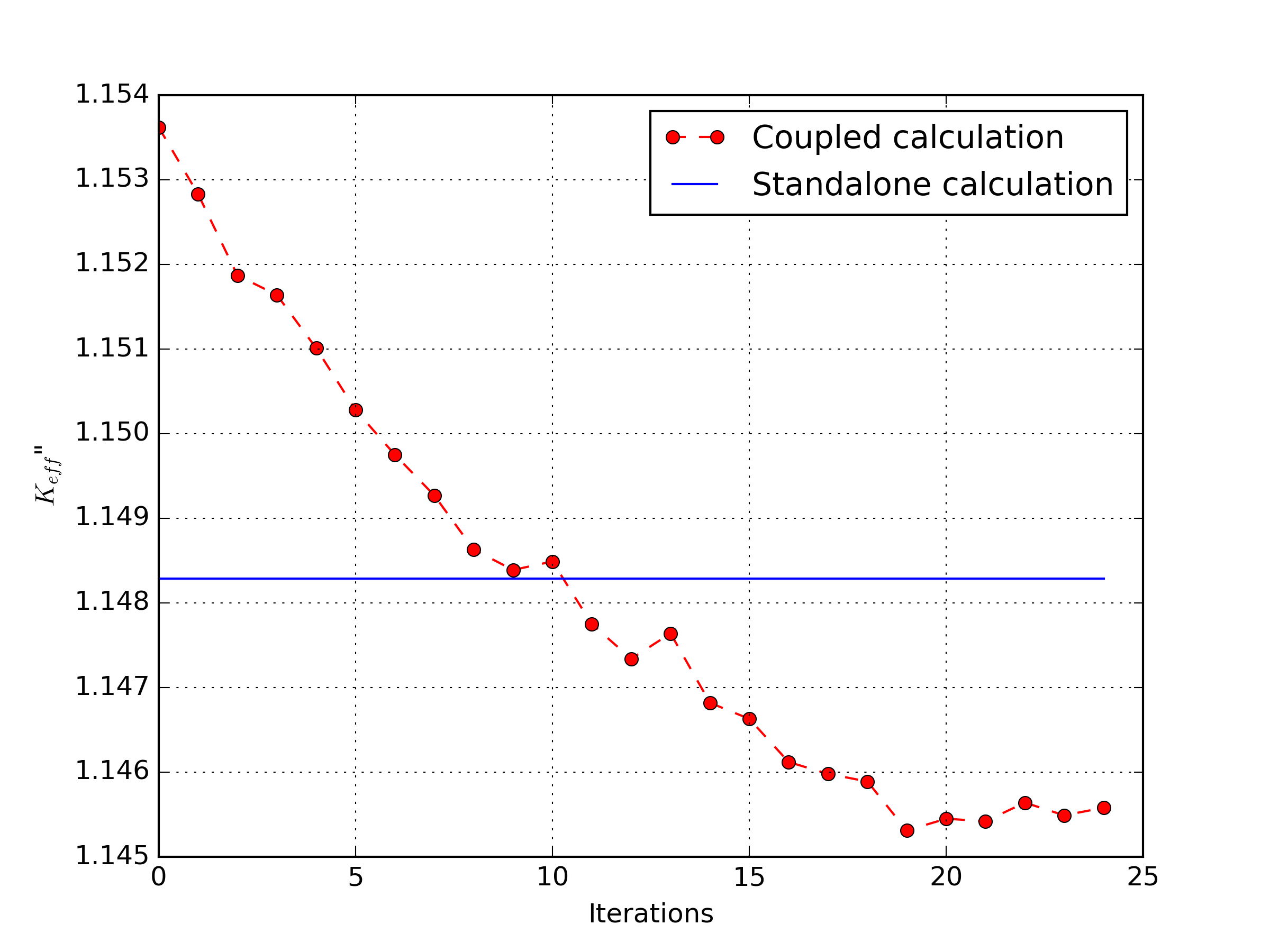
Table 10 presents the keff results from standalone and coupled neutronic simulations. The coupled simulation showed lower values than the standalone calculations under the same conditions. This is due to the feedback from neutronic calculations as the temperature at the higher flux positions will be above the average temperature, changing cross-sections and reducing the reactivity more in this region. This effect outweighs the inverse effect of the extremities of the fuel that will have below average temperatures, as these regions will have a lower neutron flux. These results all indicate that the proposed coupled simulation successfully show the expected behavior.

**Table 10. Results of standalone and coupled neutronic calculations.**

|  |  |  |  |
| --- | --- | --- | --- |
| Fuel pin power [kW] | Full core power equivalence [kW] | Standalone keff | Coupled keff |
| 7.930 | 200 | 1.14829 | 1.14354 |

A graphical view of keff values until calculations convergence is shown in Figure 10. Values for keff at each call to *milonga* during one coupled simulation are plotted. The keff value in standalone is presented as a constant line for the sake of clarity, since it is calculated in only one *milonga* run. It can be noticed that keff for the coupled calculations decreases below the values of standalone simulation after a set of iterations and then remains lower until convergence. Some oscillation of the coupled simulation was observed until achieving convergence after about 20 calls to *milonga*.

The graphical representation of the relative fluxes for the standalone neutronic calculations and the coupled simulations is shown in Figure 11. The results for the coupled calculations show a slightly flattened curve for the neutron flux, with higher values in the extremities (lower than average temperatures) and lower values in the center (higher than average temperatures). As previously discussed, this is the expected behavior for the flux when the differences in temperatures and, consequently, in cross-sections are taken in account, indicating that the coupling has been successfully performed.

**Figure 10: keff results during coupled calculations iterations.**

|  |  |
| --- | --- |
|  |  |

**Figure 11: Radial (left) and axial (right) profiles for relative flux for standalone and fully-coupled calculations.**

In order to visualize the results of coupled calculations relative to standalone execution, a set of images were generated showing the differences in relative fluxes and power between the coupled and non-coupled calculations. Figure 12 shows these differences from different views of the model. In these images the previously described temperature and neutron flux effects can be visualized in terms of flux and power differences.

|  |
| --- |
|  |
|  |
|  |

**Figure 12 - Differences in relative neutron flux and power between standalone and coupled calculation. A – Mid axial slice, B – Mid radial slice and C – Isometric.**

**4. CONCLUSIONS**

In this paper, a coupled neutronics and thermal-hydraulics system based on finite volumes was developed primarily using free software. This coupled-framework uses shared-memory to allow data interchange between two different software without time penalty. It has the advantage of solving neutronics and thermal-hydraulics at the same spatial level of detail, using the same mesh for it. The system is applied to a single TRIGA fuel pin considering only its active portion. The results show the coupling system effectively works and that the multiphysics calculations gives flattened shapes for neutron flux when compared to standalone calculations and, as its consequences, the power shape is also affected.

A comparison between standalone neutronics calculations and coupled calculations with temperatures variation by cell in a fine mesh model show a non-negligible difference in power due to reactivity changes. These results agree with the results found using the Serpent code for calculations using an almost identical geometry for a TRIGA fuel element with a radially applied temperature gradient (Cheng and Roberts, 2017).

Although in its very preliminary version, the system presented in this paper has another innovative characteristic uncommon in other nuclear engineering software tools: it is fully open and free to use and modify. This is not merely a detail, but a form of software development popular in other fields of knowledge. The specificity and history of software development in the nuclear engineering field could have delayed the advances of open software development in this field. However, as recent projects demonstrate (Huff et al, 2016, Romano et al., 2013 and Boyd et al., 2014) - a special remark to *milonga* nuclear reactor code (Theler 2014) - open software development is wider than its technical aspects. It is a form of thinking the whole process of software development and it is not unrealistic to say that it is now a reality in the nuclear engineering field.

**4.1 FURTHER DEVELOPMENT**

There are a lot of room for improvements in the methodology presented in this work and also many technical aspects that can be addressed to make simulations faster and more reliable. Concerning the methodology, the first aspect is to improve the convergence criteria for both software. This can be achieved having a coupled calls based on real-time simulation criteria, like residuals. Another way to make simulations more time efficient is to have codes calculating simultaneously, and not staggered coupling. This can be achieved by improving the semaphore control to a more elaborated set of rules.

The improvements proposed are considering an important limitation of this proof-of-concept methodology: sequential run. *OpenFOAM* offers mature parallel implementations and the coupled development took it in account. In order to fully take advantage of this capability, *milonga* nuclear code should also be able to run in parallel. This situation would be unfortunate, if *milonga* was not freely available. Considering it is available as free software, anybody wishing to tackle this task is welcome to collaborate. This subject brings to the light technical aspects of the coupled software which can be object of improvements. The immediate one is to make *milonga* work in parallel, using domain decomposition techniques. Another way is to use internal threads to solve the eigenvalue problem in parallel while the mesh stays untouched. The use of shared-memory, one of main contributions of this work in coupled-code data exchange, can host the mesh structures, having codes sharing it and thus making use of about half of memory for main data structures. Another possible improvement worth to mention, is to modify *milonga*'s mesh data structures, specifically concerning mesh accessing, which is the bottleneck of neutronic calculations.

As concisely shown, there are a wide number of aspects in the presented implementation and methodology which can be addressed. The open characteristic of both codes are an invitation to anybody interested to contribute to try, test and improve them. This is the spirit of open software and the nuclear engineering field has a lot to benefit from it.

**ACKNOWLEDGMENTS**

The authors are grateful to the Brazilian research funding agencies, CNEN, CNPq, CAPES and FAPEMIG for their support. Furthermore, the first author would like to thank the colleagues Álvaro Bernal, Rafael Miró and Gumersindo Verdú from UPV – Polytechnic University of València, Spain – for their invaluable contributions for this work during a scholarship financed by CAPES – Brazilian Federal Agency for Support and Evaluation of Graduate Education within the Ministry of Education of Brazil - in the basis of PDSE scholarship program. The first author also thanks Tuomas Viitanen for kindly providing pre-processed cross-sections to be used by Serpent Monte Carlo code for modeling zyrconium hydrade in a TRIGA type fuel.

**REFERENCES**

Androutsellis-Theotokis, S., Spinellis, D., Kechagia, M. and Gousios, G. (2010) *Open Source Software: A Survey from 10,000 Feet*. Foundations and Trends in Technology, Information and Operations Management, Vol. 4, Nos. 304, 187-347

Aufiero, M., Fiorina, C. and Laureau, A. (2015) *Serpent-OpenFOAM coupling in transient mode: simulation of a Godiva prompt critical burst,* Proceedings of M&C+SNA+MC, Pages 19-23.

Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., Curfman McInnes, L., Rupp, K., Smith, B. F., Zampini, S. and Zhang, H. (2016) *PETSc Users Manual*, ANL-95/11 - Revision 3.7. Argonne National Laboratory, 2016. http://www.mcs.anl.gov/petsc

Bennett, A., Avramova, M. and Ivanov, K. (2016) *Coupled MCNP6/CTF code: Development, testing and application*, Annals of Nuclear Energy, Volume 96, Pages 1-11, ISSN 0306-4549, <http://dx.doi.org/10.1016/j.anucene.2016.05.008>.

Boyd, W., Shaner, S., Li, L., Forget, B. and Smith, K. (2014) *The OpenMOC method of characteristics neutral particle transport code*, Annals of Nuclear Energy, Volume 68, Pages 43-52, ISSN 0306-4549, http://dx.doi.org/10.1016/j.anucene.2013.12.012.

Fiorina, C., Clifford, I., Aufiero, M. and Mikityuk, K. (2015) *GeN-Foam: a novel OpenFOAM® based multiphysics solver for 2D/3D transient analysis of nuclear reactors*, Nuclear Engineering and Design, Volume 294, Pages 24-37, ISSN 0029-5493.

Fiorina, C., Kerkar, N., Mikityuk, K., Rubiolo, P. and Pautz, A. (2016) *Development and verification of the neutron diffusion solver for the GeN-Foam multiphysics platform.* Annals of Nuclear Energy, Volume 96, Pages 212-222, ISSN 0306-4549.

Cheng, Y. and Roberts, J. A. (2017) *A study of reactivity biases and their dependence on spatial discretization in depleted TRIGA fuel*, Annals of Nuclear Energy, Volume 108, Pages 126-131, ISSN 0306-4549, <http://dx.doi.org/10.1016/j.anucene.2017.04.030>

Galassi, M., Davies, J., Theiler, J., Gough B., Jungman, G., Alken, P., Booth, M., Rossi, F. and Ulerich, R. (2009) *GNU Scientific Library Reference Manual* - Third Edition, ISBN 0954612078.

Geuzaine, C. and Remacle, J.-F., 2009, *Gmsh: a three-dimensional finite element mesh generator with built-in pre and post-processing facilities.* International Journal for Numerical Methods in Engineering 79(11), pp. 1309-1331.

GNU General Public License (2007) Version 3. Free Software Foundation. URL: <http://www.gnu.org/licenses/gpl.html>.

Hernandez, V., Roman, J. E. and V. Vidal. (2005) SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems. ACM Trans. Math. Software, 31(3):351-362.

Huff, K. D., Gidden, M. J., Carlsen, R. W., Flanagan, R. R., McGarry, M. B., Opotowsky, A. C., Schneider, E. A., Scopatz, A. M. and Wilson, P. P. H. (2016) *Fundamental concepts in the Cyclus nuclear fuel cycle simulation framework*, Advances in Engineering Software, Volume 94, Pages 46-59, ISSN 0965-9978, <http://dx.doi.org/10.1016/j.advengsoft.2016.01.014>.

Hummel, D. W. and Novog, D. R. (2016) *Coupled 3D neutron kinetics and thermalhydraulic characteristics of the Canadian supercritical water reactor*, Nuclear Engineering and Design, Volume 298, March 2016, Pages 78-89, ISSN 0029-5493, http://dx.doi.org/10.1016/j.nucengdes.2015.12.008.

Ivanov, K. and Avramova, M. (2007) *Challenges in coupled thermal–hydraulics and neutronics simulations for LWR safety analysis*, Annals of Nuclear Energy, Volume 34, Issue 6, Pages 501-513, ISSN 0306-4549, http://dx.doi.org/10.1016/j.anucene.2007.02.016.

Jareteg, K., Vinai, P. and Demazière, C. (2014) *Fine-mesh deterministic modeling of PWR fuel assemblies: Proof-of-principle of couple neutronic/thermal-hydraulic calculations,* Annals of Nuclear Energy, Volume 68, Pages 247-256, ISSN 0306-4549, http://dx.doi.org/10.1016/j.anucene.2013.12.019.

Jareteg, K., Vinai, P., Sasic, S. and Demazière, C. (2015) *Coupled fine-mesh neutronics and thermal-hydraulics – Modeling and implementation for PWR fuel assemblies*, Annals of Nuclear Energy, Volume 84, Pages 244-257, ISSN 0306-4549, http://dx.doi.org/10.1016/j.anucene.2015.01.037.

Kärrholm Peng, F. (2006) Numerical Modelling of Diesel Spray Injection and Turbulence Interaction. Appendix: Rhie-Chow interpolations in OpenFOAM. Göteborg: Chalmers University of Technology

Launder, B. E. and Spalding, D. B. (1974) *The numerical computation of turbulent flows*, Computer Methods in Applied Mechanics and Engineering, Volume 3, Issue 2, 1974, Pages 269-289, ISSN 0045-7825, <http://dx.doi.org/10.1016/0045-7825(74)90029-2>.

Leppännen, J.,Viitanen, T. and Valtavirta, V. (2012) *multiphysics Coupling Scheme in the Serpent 2 Monte Carlo Code*. Hybrid Monte Carlo Deterministic Methods For Reactor Analysis, Transactions of American Nuclear Society, Vol. 107, San Diego, California, November 11-15.

Leppänen, J., Pusa, M., Viitanen, T., Valtavirta, V. and Kaltiaisenaho, T. (2015) *The Serpent Monte Carlo code: Status, development and applications in 2013*, Annals of Nuclear Energy, Volume 82, 2015, Pages 142-150, ISSN 0306-4549, <http://dx.doi.org/10.1016/j.anucene.2014.08.024>.

Leppänen, J., Pusa, M. and Fridman, E. (2016) *Overview of methodology for spatial homogenization in the Serpent 2 Monte Carlo code*, Annals of Nuclear Energy, Volume 96, 2016, Pages 126-136, ISSN 0306-4549, <http://dx.doi.org/10.1016/j.anucene.2016.06.007>.

Mahadevan, V., Merzari, E., Tautges, T., Jain, R., Obabko, A., Smith, M. and Fischer, P. (2014) *High-resolution coupled physics solvers for analysing fine-scale nuclear reactor design problems.* Phil. Trans. R. Soc. A 2014 372 20130381; DOI: 10.1098/rsta.2013.0381.

OpenFOAM - The Open Source CDF Toolbox: User guide. Version 3.0.1 [S.l.], 2015.

Richard, J. Galloway, J., Fensin, M. and Trellue, H. *SMITHERS: An object-oriented modular mapping methodology for MCNP-based neutronic–thermal hydraulic multiphysics*, Annals of Nuclear Energy, Volume 81, Pages 150-163, ISSN 0306-4549, http://dx.doi.org/10.1016/j.anucene.2015.03.027.

Romano, P. K. and Forget, B. (2013) *The OpenMC Monte Carlo particle transport code*, Annals of Nuclear Energy, Volume 51, January 2013, Pages 274-281, ISSN 0306-4549, http://dx.doi.org/10.1016/j.anucene.2012.06.040.

Schmidt, R., Belcourt, K., Hooper, R., Pawlowski, R., Clarno, K., Simunovic, S., Slattery, S., Turner, J., Palmtag, S. (2015) *An approach for coupled-code multiphysics core simulations from a common input*, Annals of Nuclear Energy, Volume 84, Pages 140-152, ISSN 0306-4549, http://dx.doi.org/10.1016/j.anucene.2014.11.015.

Silva, V. V. A., dos Santos, A. A. C., Mesquita, A. Z., Bernal, A., Miró, R., Verdú, G. and Pereira, C. (2015) *Finite volume thermal-hydraulics and neutronics coupled calculations.* Proceedings of ICAPP 2015, May 03-06. Nice, France.

Silva, V., Campagnole dos Santos, A. A., Theler, G. and Pereira, C. (2016) *Open Software One-Step Coupled Neutronics and CFD Thermal-hydraulics Calculation,* International Journal of Emerging Technology and Advanced Engineering, Volume 6, Issue 11, ISSN 2250-2459.

Theler G., Clausse A., Bonetto F.J., *Solution of the 2D IAEA PWR Benchmark with the neutronic code milonga*, Actas de la XXXVIII Reunion Anual de la Asociacion Argentina de Tecnologia Nuclear, Buenos Aires, 2011.

Theler, G., Omil, J. P. G. and Pellegrino, E. (2013) *A shared-memory-based Coupling Scheme for Modelling the Behavior of a Nuclear Power Plant Core.* Mecánica Computacional Volume XXXII, Asociación Argentina de Mecánica Computacional, Pages 1501-1517.

Theler, G. (2014) *On the design basis of a new core-level neutronic code written from scratch. The free nuclear reactor core analysis code.* Mecánica Computacional Volume XXXIII, Number 48, Numerical Methods in Reactor Physics (B). Available at <https://www.seamplex.com/milonga> (11 January 2018).

Trahan, T. J. (2014) *An asymptotic, homogeneized, anisotropic, multigroup diffusion approximation to the neutron transport equation.* Thesis, Nuclear Engineering and Radiological Sciences, University of Michigan.

Valtavirta, V., Leppänen, J. and Viitanen, T. (2017) *Coupled neutronics–fuel behavior calculations in steady state using the Serpent 2 Monte Carlo code*, Annals of Nuclear Energy, Volume 100, Part 2, 2017, Pages 50-64, ISSN 0306-4549, <https://doi.org/10.1016/j.anucene.2016.10.015>.

Veloso, M. A. (2005) *Avaliação Termo-hidráulica do Reator TRIGA IPR-R1 a 250 kW.* NI-EC3-05/05, Belo Horizonte-MG, Brazil: Centro de Desenvolvimento da Tecnologia Nuclear (CDTN), Comissão Nacional de Energia Nuclear. *(in Portuguese)*

Williams, S. (2010). *Free as in Freedom (2.0): Richard Stallman and the Free Software Revolution* (2nd ed.). (R. M. Stallman, Ed.) Boston: Free Software Foundation.

Zerkak, O., Kozlowski, T. and Gajev, I. (2015) *Review of multiphysics temporal coupling methods for analysis of nuclear reactors,* Annals of Nuclear Energy, Volume 84, Pages 225-233, ISSN 0306-4549, http://dx.doi.org/10.1016/j.anucene.2015.01.019.

1. “Free” as in “free speech” not as in “free beer”. [↑](#footnote-ref-2)
2. Minor modifications concerning technical aspects are left as source-code comments. [↑](#footnote-ref-3)