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

**Vitor Silvaa,d\*, André Santosa, Germán Thelerb, Rafael Miróc, Cláubia Pereirad**

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

bCentro de Innovación Tecnológica Empresarial y Social S.A., Av. Belgrano 758, 2322 Sunchales, Argentina

cInstitute for Industrial, Radiophysical and Environmental Safety (ISIRYM), Universitat Politècnica de València, Camí de Vera s/n, Valencia 46022, Spain

dDepartamento de Engenharia Nuclear, Universidade Federal de Minas Gerais, Av. Pres. Antônio Carlos, 6627 - 30270-901 Belo Horizonte, MG, Brazil

**ABSTRACT**

In this paper we present the development of a fine mesh coupled neutronics/thermal-hydraulics framework exclusively using open source software. The contributions proposed in this work 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 nucler 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 of an open software coupling scheme is proven simulating a quarter of the TRIGA-IPR-R1 reactor fuel.

In order to have a realistic cross-sections variation for the model, a methodology for generating all the required two-group cross-sections data was developed using the Serpent 2 nuclear code. The coupled simulations were performed using a simple mesh of about five thousand elements also generated exclusively with the use of open software. The results show this coupled system gives consistent results and encourages the use of the system with finer meshes towards the simulation a full core reactor model.

\*Corresponding author. Tel: +55 31 3069 3466; fax: +55 31 3069 3411

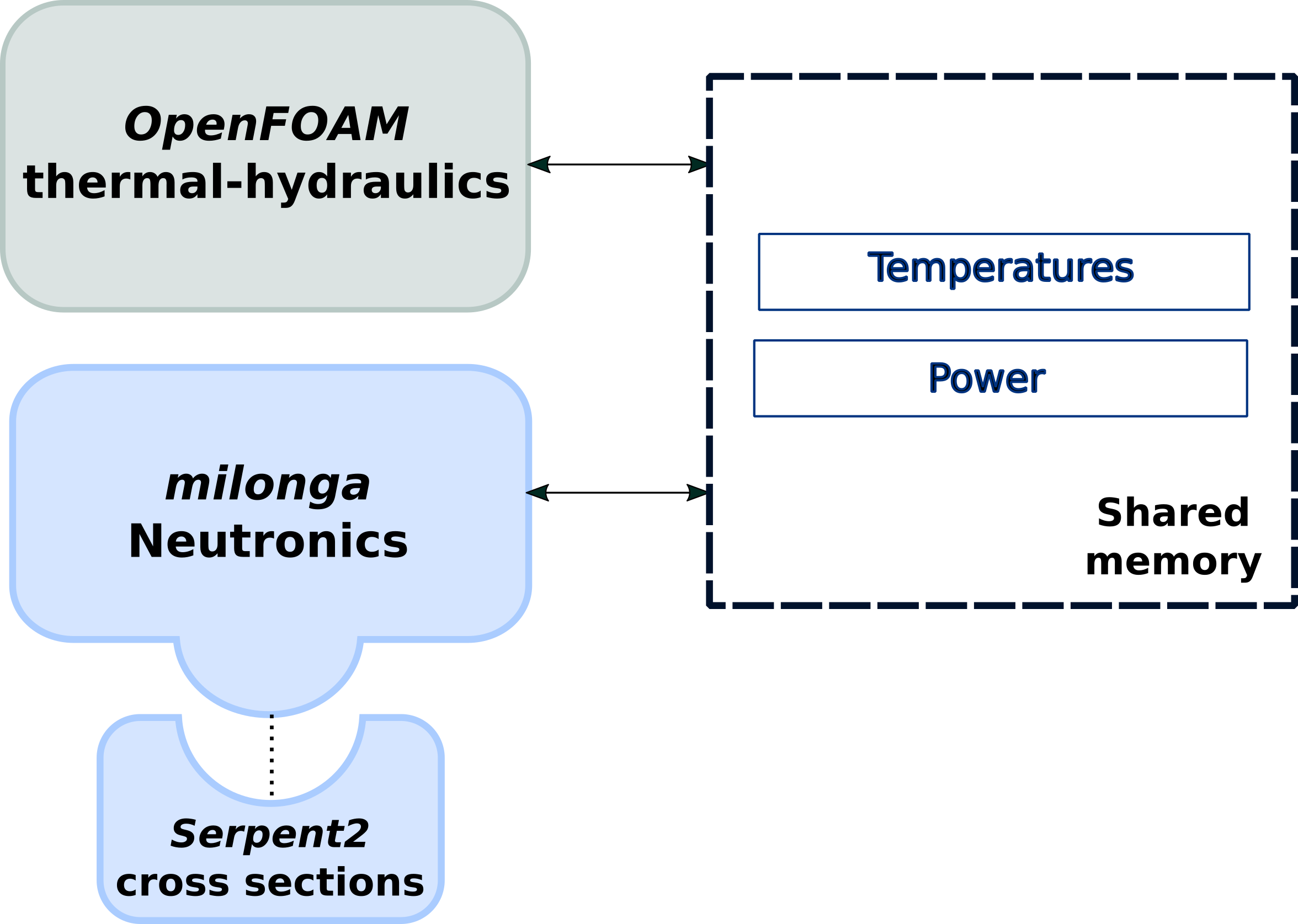
**E-mail address: vitors@cdtn.br, (Vitor V. Araújo Silva)**

**1. INTRODUCTION**

The continuous improvement 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. A recent focus is the use of thermal-hydraulics and neutronics codes to solve nuclear reactor problems in a coupled way. These coupled calculations approaches, also called multi-physics (Leppännen et al., 2012 and Schimidt et al., 2015), offers an interesting way of modeling the feedback from thermal-hydraulics to neutronics and vice versa.

Another interesting aspect of the computer hardware evolution, is the changes in the associated software. The effect of software evolution in order to follow the respective hardware improvement is often neglected, specially 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's impossible to deny open software influence in the society nowadays (Williams and Stallman, 2010). However, more than software itself, free and open source software brings 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). All of these software share a common aspect: they are all open, freely available and, most important, their development is decentralized and anyone interested in participating can download it, make changes and improvements and submit the contribution to the community. This development cycle is, probably, the main strength of an open software project.

This work discusses some efforts initiated as a cooperation development between the Nuclear Engineering Department, Federal University of Minas Gerais and Reactors Technology Service, Nuclear Technology Development Center of Brazilian National Nuclear Commission. In the light of the open software development, the present work aims to introduce a coupled neutronics and thermal-hydraulics framework for nuclear reactors calculations. The thermal-hydraulics calculations are performed by an open software Computational Fluid Dynamics (CFD) toolbox called *OpenFOAM* (OpenFOAM, 2015). For neutronics calculations, a free nuclear reactor core analysis code called *milonga* solves the steady-state multigroup neutron transport equation (Theler, 2014). 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 they are synchronized using semaphores. This scheme poses no overhead for communication other than any ordinarymemory access. The cross-sections processing is performed using *Serpent 2* continuous-energy Monte Carlo reactor physics code (Leppännen, 2013). From continuous cross-section libraries a set of two-groups cross-sections is generated for three different materials at different temperatures, corresponding to the modeled fuel element temperatures variation. Figure 1 depicts a basic schematic for the coupling methodology.

**Figure 1: Coupling methodology schematic.**

In order to validate the coupled framework, a geometry representing a quarter of a TRIGA reactor fuel element is modeled and meshed. It is worth noting that validation in the context means to guarantee that data is properly exchanged from thermal-hydraulics to neutronics and the vice versa, calculations are correctly carried and numerical convergence is achieved in both codes. In other words, the validation of the methodology and the correctness of the implementation are presented for this framework of three-dimensional multi-physics calculations.

In Section 2 the models, codes used and methodology for cross-sections generation, are briefly described. In Section 3, the coupled aspects are discussed involving details and implementation decisions. Section 4 is dedicated to describe the application and results from separated and coupled calculations. Finally, in Section 5, the conclusions and results are presented along with the discussion of future development paths for the framework.

**2. MODELS AND METHODOLOGY**

Before diving into the details of the coupled calculations, it is important to describe the models and methodology used for each code in stand-alone mode. It is also fundamental to remark that both codes solve their equations in a steady-state mode.

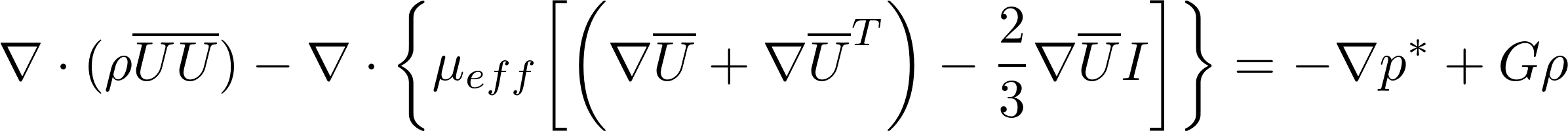
2.1 Thermal-hydraulics

The *OpenFOAM* toolbox offers numerical solvers for many common problems, like heat transfer, two-phase fluid flow, financial problems and many others. In order to solve a one-phase fluid flow including conjugated heat transfer from three different materials, the so-called chtMultiRegionSimpleFoam was chosen.

The fluid dynamics is governed by the momentum equation, the continuity equation and the energy equation. 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 for every region.

For fluid regions, and based in previous values of pressure and density, it solves respectively the equations related to momentum, energy and pressure. Differently from common numerical tools, the solver used, as all other solvers available in OpenFOAM, allows equations assembly directly using objects including discretization methods. That said, the respective equations solved for the momentum are:

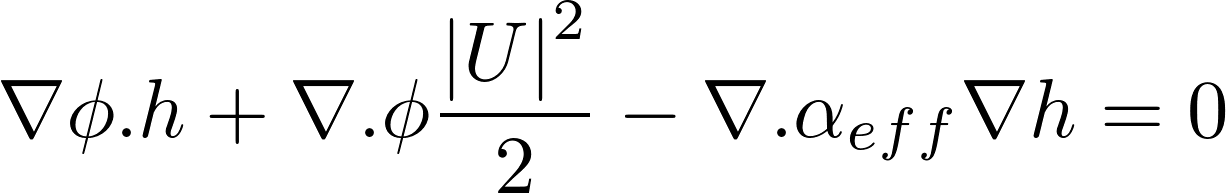
(1)



where the *μeff* term is the effective dynamic viscosity, which is the sum of viscosity and turbulent viscosity. The *Gρ* term comes from a buoyancy adjustment.

The energy equation is implemented in a flexible way, allowing for calculations using enthalpy or internal energy. In this work enthalpy is used and the energy equation is described accordingly:

(2)



where the second therm is the contribution from kinetic energy and the third therm accounts for the heat flux, being *αeff* is the sum of laminar and turbulent thermal diffusivities. It is worth noting that the velocities field is previously calculated as the equations are implemented in a specific order. The next equation to be described is the so-called continuity equation.

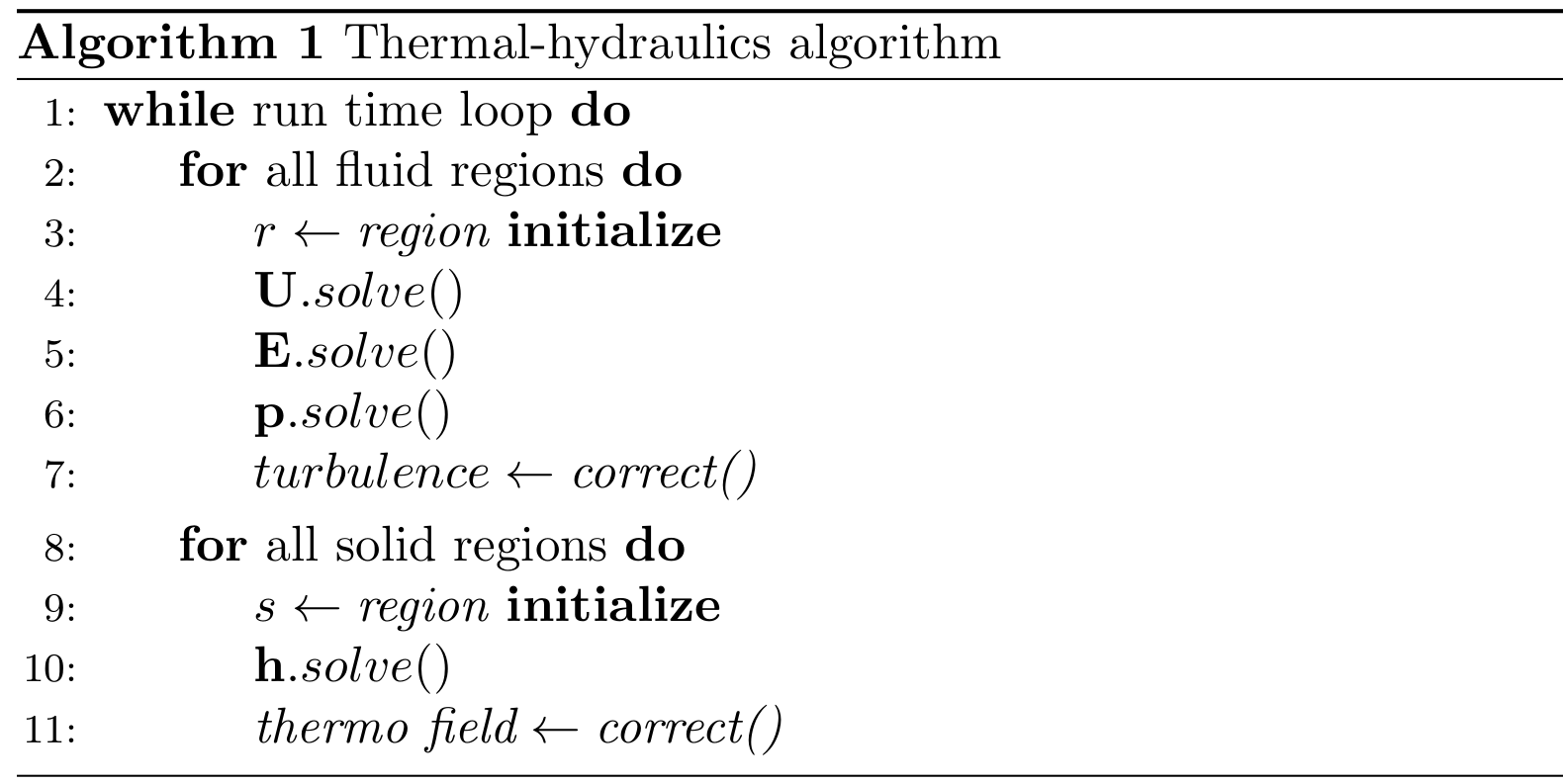
(3)



The transient term is not present for the steady-state problem and the continuity is assured when the divergent of momentum is zero. It is worth mention that OpenFOAM has its own particular way of coupling pressure to velocity (Kärrolhm, 2016) which is beyond the scope of this text.

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 neutronics coupling scheme. Section 4 is entirely dedicated to the coupling methodology and aspects of this work.

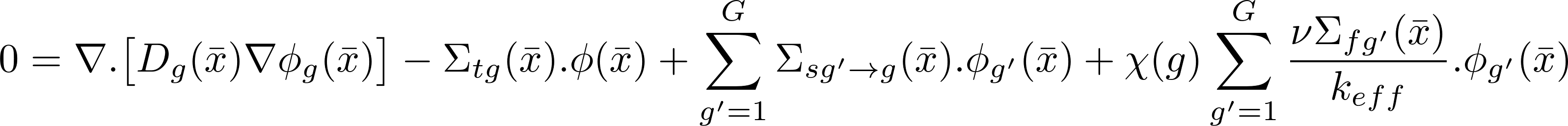
The set of equations are embedded in the solver in an iterative algorithm which is presented as a pseudo-code in Figure 2.

**Figure 2: *OpenFOAM* pseudo-code for the solver used for coupling.**

2.2 Neutronics

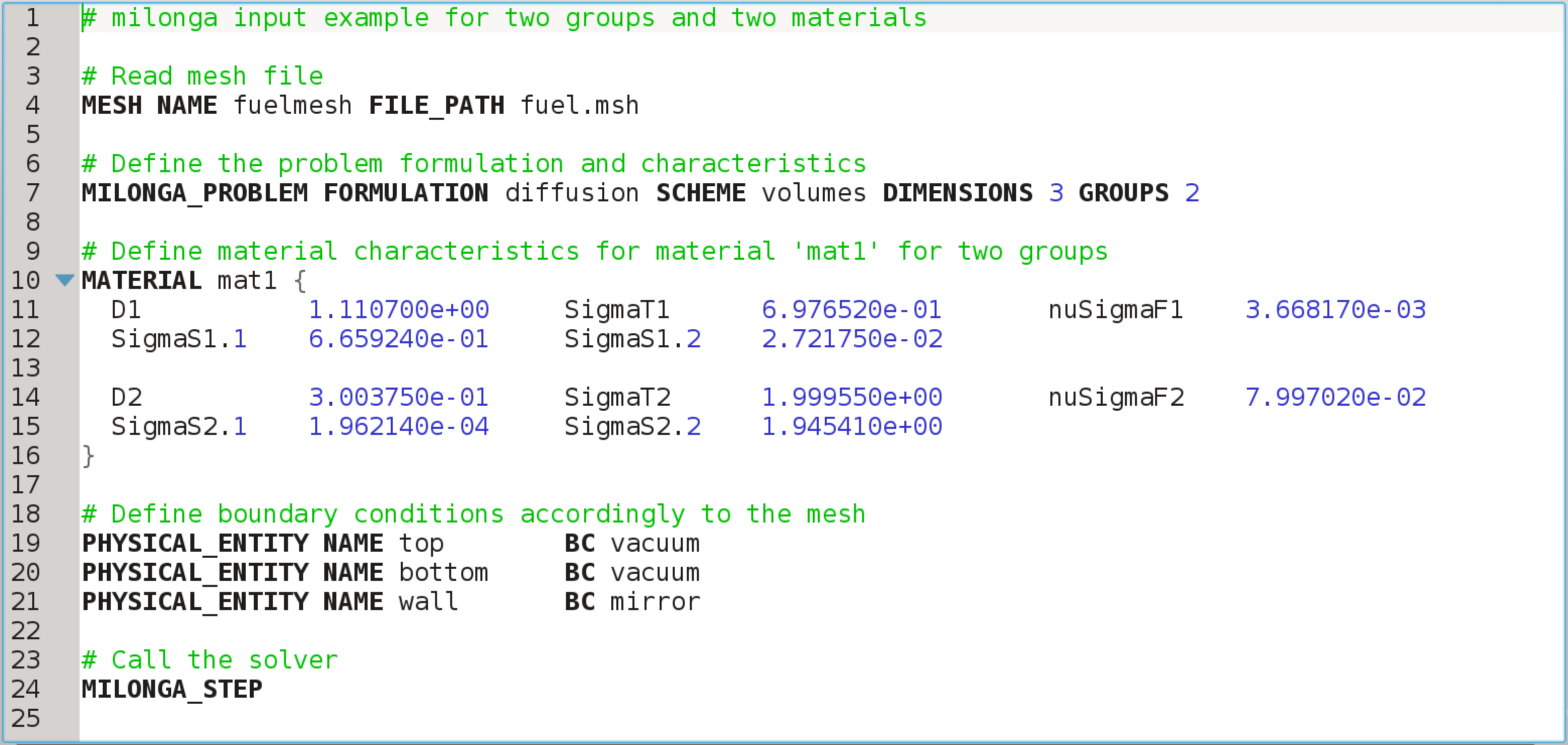
The *milonga* nuclear code is a free and open source piece of software released under GNU license. It heavily relies on other well known GNU libraries, like PETSc (Balay et al., 2016), SLEPc (Hernandez et al., 2005) and GNU Scientific Library (Galassi et al., 2009), bringing to it the robustness of these established software and keeping the principles of freedom of software use that GNU enforces. It solves the steady-state multi-group neutron transport equation, either using the diffusion approximation or discrete ordinates *Sn* method. It can discretize the spatial coordinates using either a finite-element or a finite-volumes scheme. This ability to use finite-volumes discretization schemes over unstructured grids makes it possible, regarding the present problem, to have the same mesh for both thermal-hydraulics and neutronics. With identical meshes, the dependency between fields in each tool can be solved with the same degree of detail. Despite the inaccurate results and 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 calculations can be observed using the diffusion approximation for multi-group steady-state, presented below already discretized in G groups:

(4)



Equation 4 is discretized using the chosen formulation, finite volumes in the present case. 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.

In a stand-alone case, *milonga* works by reading an input file which models the problem to be solved. The simplest case for the present problem consists in the basic problem configuration: read the mesh, define the numerical formulation and chose 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) 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. All of them should be defined for each group of energy. After materials definitions, the appropriate boundary conditions are set matching the physical entities defined in the mesh. In order to solve the problem, a simply call to a command builds and solve the eigenvalue problem. Figure 2 shows a simple input file for milonga.

**Figure 3: A simple input file for milonga with line number counting.**

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, not shown in Figure 3. Some of these features will be presented in Section 4.

A key feature of *milonga* which makes external coupling straightforward is the option to get data in a cell wise way when discretizing the calculation domain as a mesh. In other words, it is possible to have a spacial function dependence for any variable. However, to have the corresponding values of cross-sections (and all other coefficients for the diffusion approximation) for each cell, a properly set of neutronic macroscopic cross sections had to be generated prior to performing the neutronic computation itself.

**3. COUPLING SCHEME**

As mentioned in Section 1, the present coupling has some particularities which are related to the CFD, neutronics and cross-section generation software. A general coupling solution was developed. The implementation was tested and verified using data from the still operational TRIGA IPR R1 reactor. This is a TRIGA Mark 1 in operation since 1961 at the Centro de Desenvolvimento da Tecnologia Nuclear (CDTN), Belo Horizonte – MG, Brazil (Veloso, 2005).

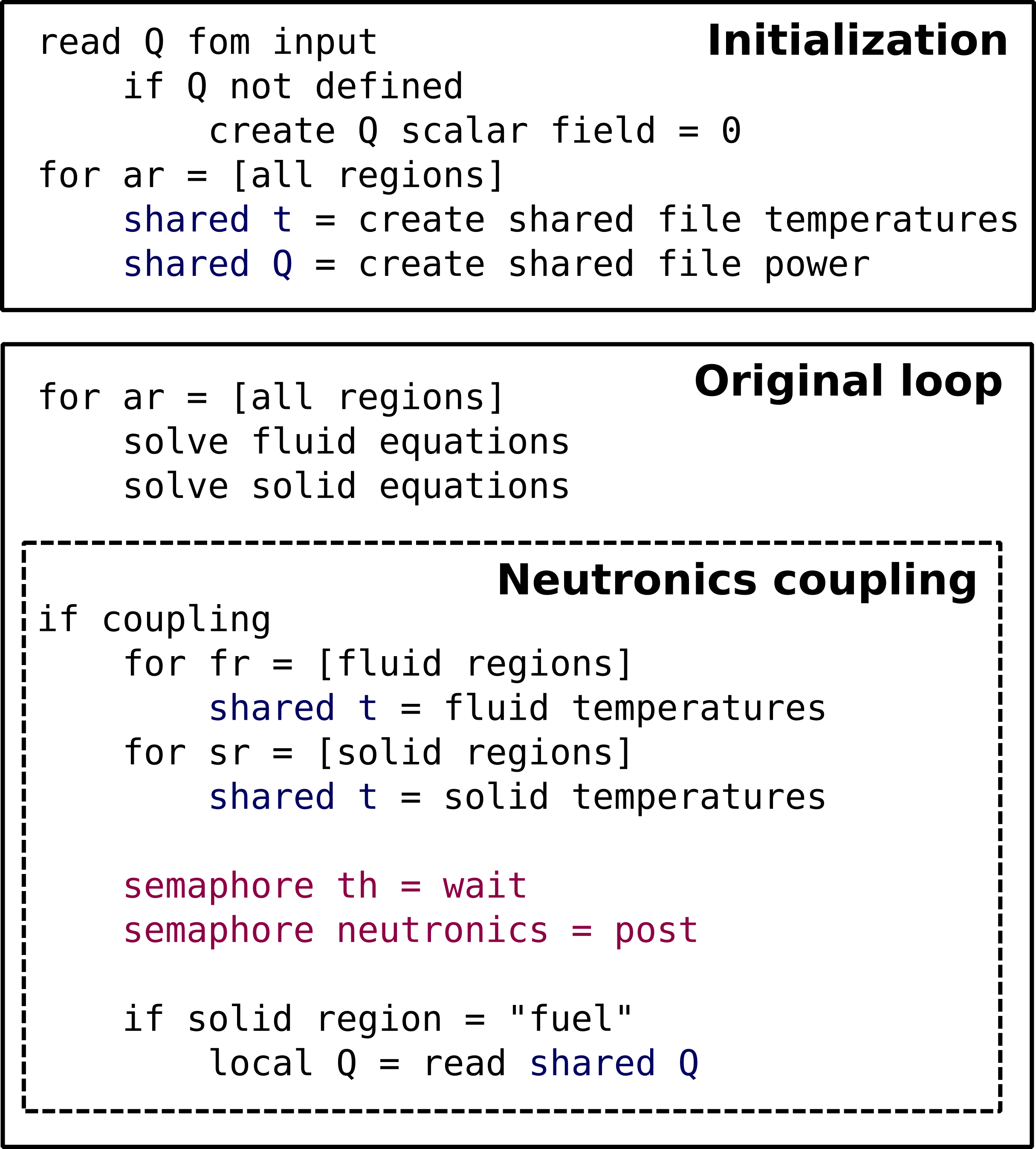
3.1 Thermal-hydraulics

The open software characteristic of OpenFOAM allows changes and modification of source code, as mentioned before. So, after properly choosing the conjugated heat transfer solver for the thermal-hydraulics and neutronics problem, it had to be modified in order to attain the objetives of this work. There were two major modifications to the solver, namely: the addition of a source-term to the energy equation in a solid and 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. This decision had implications in the implementation which will be described in section 4.

The procedure to add a source-term followed the development pattern used in OpenFOAM. A new scalar field was defined in the input files to give power in the fuel. If the field is not present, a null default scalar field is created. This procedure is the same for all solid regions. However, there is one implementation restriction: only the region called “fuel” has the apparatus for communication.

Another major change in the solver is related to how communication is implemented. Taking advantage of the abilities of *milonga* for accessing shared memory, this is the way the coupling is implemented in OpenFOAM solver. 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 OpenFOAM solver together with the structures to detect and read the shared memory (Theler et al., 2013).

The methodology for the coupled communication defines that the neutronics program starts, creates the memory shared memory and halts until it is able to read the semaphore from thermal-hydraulics. Its counterpart checks if shared memory is present and, if yes, runs the thermal-hydraulics simulation with initial conditions a fixed number of iterations and then signal that to the neutronics. Back to milonga, when getting the good value for the semaphore, it reads temperatures from shared memory and calculates flux and power. When done, it signs to thermal-hydraulics that it can proceed. From now on, thermal-hydraulics will use the power read from shared memory as a source-term in the energy equation for solid. The whole process goes until convergence of thermal-hydraulics. Figure 4 presents the coupling algorithm in a schematic way.

**Figure 4: Thermal-hydraulics coupling algorithm schematic. 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 the coupling is implemented. For now, a fixed ratio of thermal-hydraulics/neutronics iterations is implemented. Despite the implemented gather and scatter functions for parallel run, semaphores control and shared memory calls must be changed accordingly. The solution for both these limitations is straightforward and a complete explanation of future work will be presented in the final section.

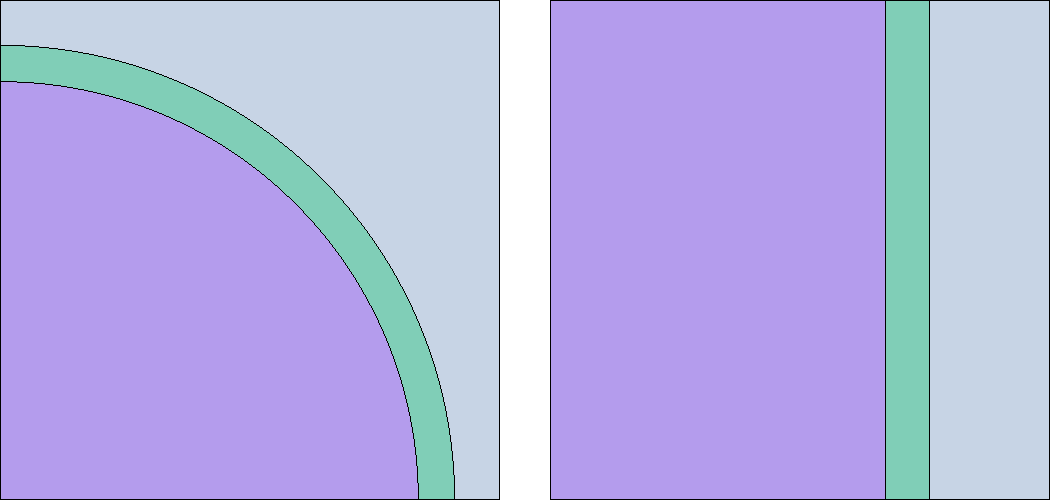
The advantages of this methodology are mainly in the use of shared memory and in the fact of having the same mesh for both codes. Shared memory is a portion of memory accessed by different programs and its use poses no overhead in data access greater than any other memory access. Some works on coupled thermal hydraulics and neutronics using external files have access time orders of magnitude higher than the achieved using shared memory (Ivanov et al., 2007 and Hummel and Novog, 2016). It must be noted that tools for shared memory communication are available in most of the common operational systems available today. The other main advantage in this coupling scheme is the use of the same mesh for both codes. This allows calculations with the same level of detail for thermal-hydraulics and neutronics also avoiding the use of mapping functions (Araújo 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 neutronic 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.

3.2 Cross-section generation methodology

In order to verify the correctness of communication between thermal-hydraulics and neutronics while having a consistent physical validation, a simple methodology was developed to cross-sections generation. The selected tool for this task was the nuclear code *Serpent 2*, which have been widely used not only as a nuclear reactor simulation tool, but also for few groups cross section generation (Fridman and Leppännen, 2011). Despite being distributed without costs for research applications with the source code available, *Serpent 2* cannot be cited in this work in the open software context due to limitations in its license which does not comply with GNU open software license (GNU license, 2007). That said, it is worth noting that *Serpent 2* is used in this work strictly to generate cross sections, not being part of the coupled set of codes.

Recalling that the coupling methodology is based in both thermal-hydraulics and neutronics having the same mesh. Therefore a model representing the mesh was created in *Serpent 2*, matching geometry, materials and boundaries of the reference mesh. The chosen model is a quarter of TRIGA IPR-R1 fuel with aluminum cladding. All geometrical and physical characteristics defined for this model, including reactor elements temperatures and moderator temperatures (which will be describe later in this section), are based in the real fuel element still in use in the reactor (Veloso, 2005). Figure 5x presents the model geometry

Some simplifications in the model do not take in account some characteristics of the whole reactor, namely: fissile material and moderator ratio, geometry variations among neighbor fuel elements nor geometrical representation of the gap between fuel and cladding. These simplifications allow the representation of a quarter of fuel element with symmetric boundaries for neutronics calculations while keeping a realistic approximation of a fuel element in the core.

**Figure 5?: Top-bottom view (left) and side view (right) from the model of a quarter of a TRIGA reactor. Purple represents fuel, green represents cladding and light blue represents water.**

Once with a geometric model, the next step consisted in the isotopic definition of materials. The materials were defined for four different temperatures in order to cover the maximum variation of temperature of reactor elements considering usual operation. Material temperatures were defined using continuous-energy cross-section libraries distributed with *Serpent 2* making use of the available Doppler broadening routine (Viitanen and Leppännen, 2012) and are presented in Table 1.

**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 | 308.5 | 317 | 341 |

One simulation was run for each set of materials temperatures, always with the same setup. The output of each simulation contains data to be used by *milonga* as coefficients in the neutron diffusion approximation equation. These data is then written in *milonga* format as a one-variable function dependent on temperature. Since *milonga* has cell-wise information loaded with the mesh, the temperature of each cell becomes the argument of the one variable functions dependent on temperature. If the temperature is not exactly tabulated, *milonga* provides a simple interpolation by default. Figures 6, 7, 8 and 9 present the variation coefficients for each of the three materials with temperature obtained by *Serpent 2,* respectively: diffusion coefficients, total cross-sections, scattering cross-sections for groups 1 and scattering cross-sections for group 2. In Figure 9 is possible to verify that values obtained for scattering cross-sections from group 2 to 1 are close to zero and can be neglected. However, all data is presented for completeness.

**Figure 6: Diffusion coefficients calculated by Serpent for each material for both groups changing with temperature.**

**Figure 7: Total cross sections calculated by Serpent for each material for both groups changing with temperature.**

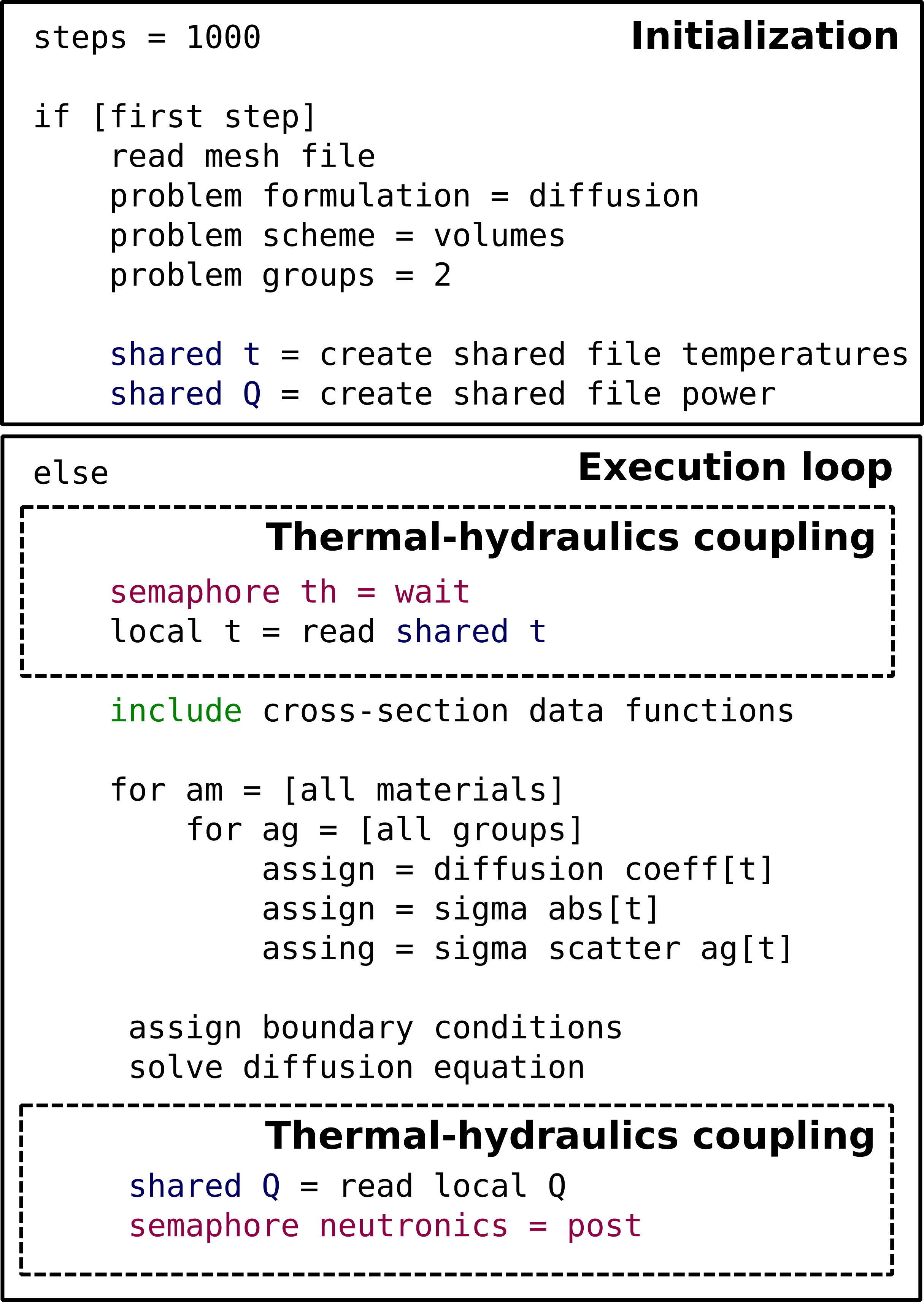
**Figure 8: Scattering cross sections for group 1 calculated by Serpent for each material for both groups changing with temperature.**

**Figure 9: Scattering cross sections for group 2 calculated by Serpent for each material for both groups changing with temperature.**

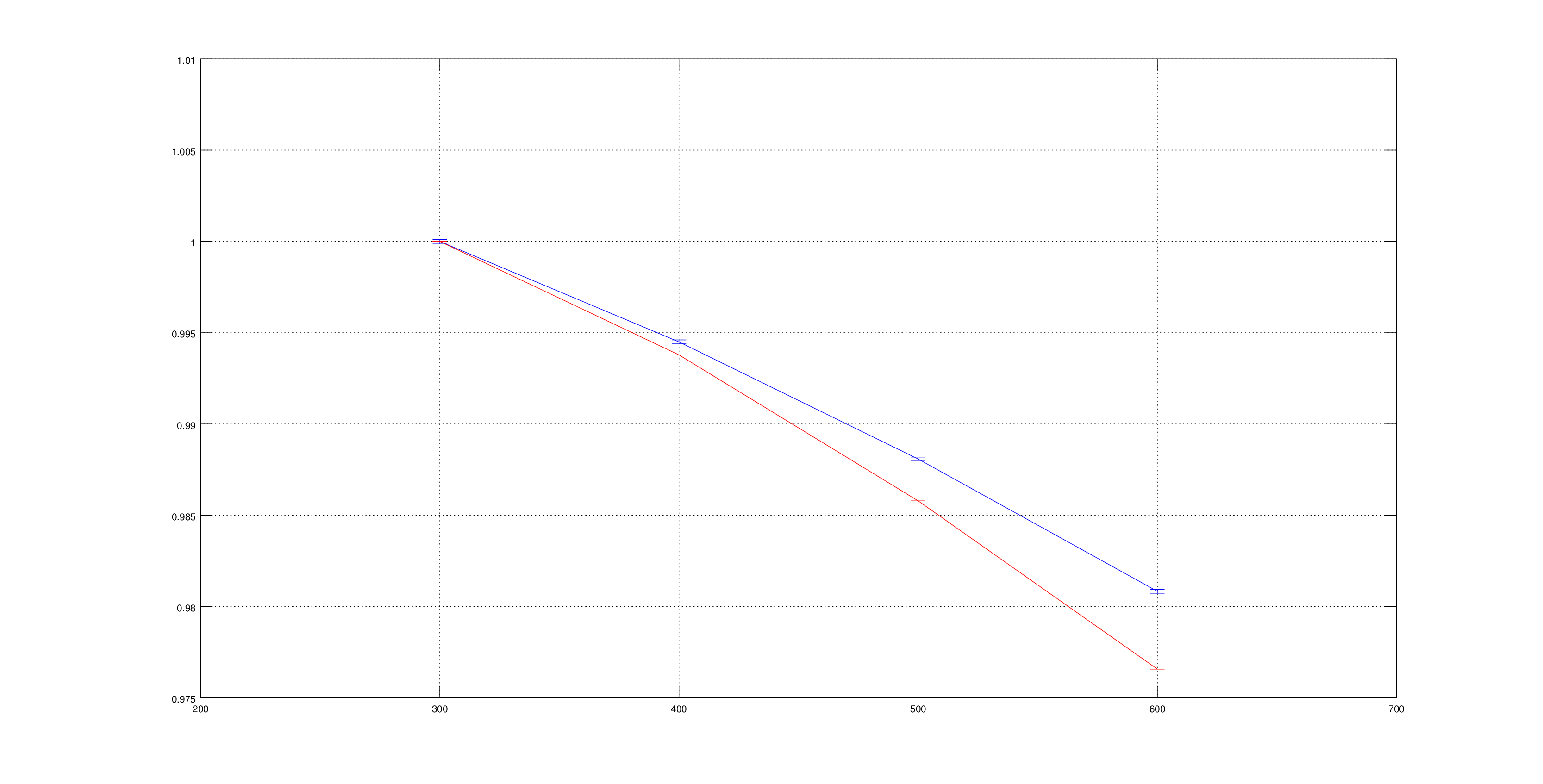
3.3 Coupling methodology

As project definition, *milonga* is the responsible for shared memory creation, neutronics calculations and power feedback to the OpenFOAM solver. As presented for thermal-hydraulics, Figure 10 shows a pseudo-code representing *milonga* input.

After solving the diffusion approximation and having the fuction for the flux, *milonga* has a primitive to integrate the flux over the mesh for a defined power. The calculated power is thus written in a mesh and in the shared memory, making the power available for the OpenFOAM solver. The coupling cycle is complete and will continue until thermal-hydraulics calculation converges.

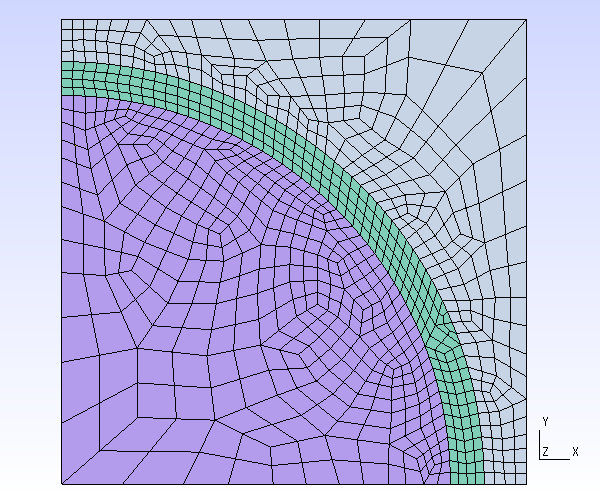
**Figure 10: Neutronics coupling algorithm schematic. Blue commands are related to shared memory manipulation, purple commands shows access to shared memory and the green command includes definitions in external files. Note that milonga does not need any modification or re-compilation in order to be coupled. Everything is done through the input file and interpreted during run-time.**

It is worth noting that in a basic validation of *milonga,* keff values calculated by *Serpent 2* differed from those calculated by *milonga* using the coefficients provided by *Serpent 2* (Figure 11). This is due to the inherent limitations of the diffusion approximation (Trahan, 2014) and are not an issue for the verification of the coupling scheme. Along with the diffusion approximation, *milonga* also implements discrete ordinate methods which can give better results than the diffusion approximation, at the cost of memory and time.

**Figure 11: Differences in keff from Serpent 2 to milonga inherent to the diffusion approximation**

3.4 Coupled calculations

As mentioned in Section 1, a simple mesh was generated representing a quarter of TRIGA-IPR-R1 reactor fuel element. This mesh represents the calculations domains for neutronics and thermal-hydraulics and is graphically depicted in Figure 12. It is extruded from the two dimensional mesh, i. e. the third dimension is generated defining the number of layers and repeating the faces. This gives a mesh which is unstructured, but made of hexahedric elements.

**Figure 12: Mesh view from the top: purple for fuel, green for cladding and light blue for coolant.**

**4. RESULTS**

**5. CONCLUSION AND FURTHER DEVELOPMENT**

**ACKNOWLEDGMENTS**

The first author would like to thank the colleagues Rafael Miró, Álvaro Bernal and Gumersindo Verdú from UPV – Politechnical 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.

**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

Araújo Silva, V. V., 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.

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

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.

Fridman, E. and Leppänen, J. (2011) *On the use of the Serpent Monte Carlo code for few-group cross section generation*, Annals of Nuclear Energy, Volume 38, Issue 6, Pages 1399-1405, ISSN 0306-4549, http://dx.doi.org/10.1016/j.anucene.2011.01.032.

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.

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., 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

Leppännen, J.,Viitanen, T. and Valtavirta, V. (2012) *Multi-physics 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.

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.

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 milonga free nuclear reactor core analysis code.* Mecánica Computacional Volume XXXIII, Number 48, Numerical Methods in Reactor Physics (B).

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.

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)*

Viitanen, T. and Leppännen, J. (2012) *Explicit Temperature Treatment in Monte Carlo Neutron Tracking Routines – First Results.* PHYSOR 2012: Advances in Reactor Physics Linking Research, Industry, and Education Knoxville, Tennessee, USA, April 15-20, on CD-ROM, American Nuclear Society, LaGrange Park, IL.

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.