POLITECNICO DI TORINO

Master's Degree in
Energy and Nuclear Engineering

Master's Degree Thesis

Development of a new thermal-hydraulic module for FRENETIC, a code for the multiphysics analysis of liquid metal-cooled reactors

Supervisors
Prof. Sandra Dulla, PhD
Ing. Giuseppe Francesco Nallo, PhD
Ing. Nicolò Abrate, PhD

Candidate
Antonino Lombardo

Academic Year 2021-2022

Summary

Heavy Liquid Metal Cooled Reactors (HLMCRs) are among the most promising Generation IV concepts. In view of the growing interest towards these systems, HLMCR core design activities are currently being pursued worldwide. The core design of a fission reactor is a complex task that must take into account the intrinsically multi-physics nature of the system. Computational tools that are reliable while being fast running are required both in support of the core design phase and to assess the reactor behaviour during operational and off-normal transients. The specific features of HLMCRs, namely the fast neutron spectrum and the liquid metal coolant, determine a different behaviour with respect to commercial light water reactors, thus calling for the development of specific codes.

FRENETIC is a multi-physics code for the full-core simulation of liquid metal-cooled fast reactors, developed at Politecnico di Torino. The code is capable of performing steady state and transient neutronic (NE) and thermal-hydraulic (TH) coupled calculations, while maintaining a relatively low computational cost thanks to the adoption of simplified physical models. The NE module implements the nodal formulation of the multigroup neutron diffusion equations with delayed neutron precursors, whereas the TH module treats the reactor hexagonal assemblies as separate channels, which are individually modelled as 1D in the axial direction and then thermally coupled to their neighbors in the horizontal directions.

In this thesis, a new TH module for FRENETIC was designed, developed and tested, in accordance with current best practices, which resulted in improved performance, portability and flexibility. Specific attention was devoted to ensure a correct memory management and a high code modularity, so to simplify future developments and/or extensions of FRENETIC. The governing equations were discretized with a finite volume treatment, taking advantage of the incompressibility of liquid metals.

The performance of the new TH module were benchmarked against the previous code version. The newly developed code shows considerably improved performances in steady state and improved performance in transient calculations. Finally, as a reactor-relevant test case, the code is applied to simulate the ALFRED reactor core in steady state and in a representative transient.

Acknowledgements

Thanks to my family that supported me even when it was not so possible. Thanks to people that was near to me and gave me new possibilities during this last year.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The demand for electricity is likely to increase significantly in the near future, as current fossil fuel applications are replaced by processes using electricity, for example in the transport sector. The development of non-CO_2 emitting sources of energy is essential to meet the challenges and create diversification for national countries' choices of a new energy mix. Nuclear energy, which is currently responsible for around 15% of the world's electricity generation, has the chance to play a significant role thanks to: the high capacity factors that are achievable, often more than 90% with long operating cycles, making the units suitable for covering the base-load requirements; essentially negligible operating emissions of carbon dioxide into the atmosphere, compared with alternative electricity production systems. Being an important part of the solution for the coming years and decades, strong and concerted action is required to develop appropriate technologies from the short term to the long term [1].

Alongside the research activities on fusion energy, new concepts belonging to a new generation of fission reactors are under study. These Generation IV reactors will explore the opportunities of providing economically and publicly acceptable electricity, broadening the possibilities for the use of nuclear energy [2].

1.1 Generation IV reactors

To satisfy the expectations Generation IV designs will meet stringent standards compared to current technologies. To achieve these, in the preparation of the road map towards the realization of design, a series of goal were established in four broad areas:

- **Sustainability** Enhanced sustainability is achieved primarily through the adoption of a closed fuel cycle. The main advantages of such systems over current reactor technologies are their improved use of resources and the reduced volume of final waste. Advanced waste management strategies include the transmutation of nuclides such as minor actinides, cost-effective decay-heat management, flexible interim storage. The reduced waste production, thereby improving protection for public health and the environment.
- **Economics** In order to be competitive, Generation IV nuclear energy systems shall

have a clear lifecycle cost advantage over other energy sources, maintaining the same level of financial risk of other energy projects. A high thermal efficiency is also targeted.

- **Safety and Reliability** Their operations will be safe and reliable, decreasing the likelihood and degree of reactor core damage and eliminating the need for offsite emergency response thanks to the adoption of passive safety systems.
- **Proliferation Resistance and Physical protection** To achieve that, all the design systems employing recycling, they avoid separation of plutonium from other actinides and incorporate additional features to reduce the accessibility and weapons attractiveness of materials at every stage of the fuel cycle. Nuclear plants shall have improved physical protection against acts of terrorism.

With these goals in mind, Generation IV International Forum (GIF) selected six reactor technologies for further R&D. These include: Gas-Cooled Fast Reactor (GFR), Lead-cooled Fast Reactor (LFR), Molten Salt Reactor (MSR), Sodium-cooled Fast Reactor (SFR), SuperCritical-Water-cooled Reactor (SCWR), Very-High-Temperature Reactor (VHTR) [3]. The first four of these concepts are characterized by a fast neutron spectrum, the advantages of which shall be described in next chapter.

1.2 Heavy Liquid Metal Cooled Reactors

1.2.1 Fast neutron spectrum

Operating at a higher neutron energy range leads to the possibility to use U-238 and Pu-239 as major fissile materials. The number of neutrons released by fission is higher for Pu-239 than for U-235, thus increasing the possibility to sustain the chain reaction, which is more difficult in a fast spectrum due to the lower cross-section. To operate with a fast spectrum, a coolant that does not significantly interact with neutrons is necessary, since there should be no moderation. For this purpose gases and liquid metals are some of the main candidates.

In addition, LMs are characterized by low neutron absorption and moderation, which makes them ideal for designs which target breeding. In these cases, the design only needs a small surplus of reactivity at the beginning of the cycle, which reduces the overall inventory of fissile material. Fast spectrum reactors have the potential to produce much less long-lived nuclear waste, and some designs can be configured to transmute some of this waste into smaller radioisotopes. Minor actinides can be used as additional fissionable material in fast reactors, as they tend to be preferentially transmuted at high neutron energies, whereas in LWRs they often transform into higher actinides as a result of neutron capture. This leads to the desired enhanced sustainability, at the cost of worsening kinetic parameters. Also, the low moderation leads to a consequent decrease of the reactivity swing and a general reduction of huge reactivity insertion due to changes in the coolant state, including lower risks in case of an accidental withdrawal of a control rod.

The high thermal inertia and negative reactivity feedback of LMs systems offer, in general, more time for corrective operator action, even in case of an unprotected transient during which small positive reactivity feedbacks are counterbalanced by the strong negative

core radial expansion feedback, which limits the reactor power excursion [3]. On the other hand, the short neutron lifetime combined with a low fraction of delayed neutrons and modest feedback coefficients implies faster reactor dynamics and different thermal feedback behaviour with respect to thermal reactors. For example, the Doppler coefficient in a fast spectrum reactor is proportional to the inverse of the temperature, whereas in thermal spectrum reactors it is often parameterized as proportional to the inverse square root of temperature [4].

The fast neutron spectrum and higher flux potentially increase irradiation damage to core structures. Luckily, lead is an excellent radiation shielding material, nevertheless the structure of the core still requires thorough studies to choose the right materials and establish the operating lifetime, as a function of sustainable DPA.

Additionally, the relevant presence of Pu causes a worsening of kinetic parameters. An important characteristic of an LFR, as for other fast reactors, is that the reactor core is not in its most reactive configuration under normal operating conditions and that it is possible to have a positive void reactivity in the center area of the reactor core during events such as core compaction, fuel melting or coolant losses.

1.2.2 LMs characteristics and core design

Among other Generation IV concepts, LFRs choice offers a number of advantages: for example, it allows a neutron spectrum harder than any other concept, due to the very large difference in mass between neutrons and coolant atoms, leading to of high neutron energy and relatively low parasitic neutron absorption. Additionally, it provides a series of benefits from a thermal-hydraulic perspective.

Metal coolants have high thermal conductivity, which enables better heat transfer capabilities, and the larger heat capacity allows in principle to remove heat efficiently from the core under very low flow velocity conditions, reducing in general the potential occurrence of local hot spots. Nevertheless, possible issues might occur due to the possible formation of oxide layers caused by the specific lead properties.The high boiling point of liquid metals allows reactors to be operated at high temperatures, increasing the efficiency of the plant. These high operating temperatures are possible without the need to pressurize the reactor's primary system, reducing the costs and complexity of the system. Also, in any accidental scenario, there is a significant safety margin before reaching coolant boiling, which practically excludes the need to design for the voiding of the core. Thanks to the high boiling point of the lead coolant, boiling is extremely unlikely, but gas/voids can still appear in the core due for example a fission gas release from failed fuel pins or due to steam generator tube leakages or ruptures [5]. In fact, high latent heat, and high thermal capacity of lead, provide significant thermal inertia in the event of a loss of heat sink and guarantee long available time for operator action in case of accidents, reducing the need for offsite emergency response [4].

This type of reactor can have either loop or pool design. However, the pipework connections increase the risk of a primary circuit breach and therefore introduce additional safety challenges. Pool-type designs eliminate this problem but require a larger vessel and routine operations, e.g., for the removal of major components from the pool or for inspection and repair, are more challenging due to the fact that components are under

a visually opaque coolant, hence requiring the usage of X-ray or ultrasonic devices for inspection. The high density of the coolant leads also to a net positive buoyancy force, so all components immersed in the coolant must be anchored. In turn, this larger vessel leads to a larger coolant inventory and thus thermal mass, which is favourable during fault transients.

All the metal coolants considered for LFRs are solids at room temperature, so heaters are required to maintain the coolant liquid during start-up operations but also maintenance activities, as well as during refuelling. The unplanned freezing of liquid metals is a problem regarding the blockage of flow paths.

The higher density of the heavy liquid metals is also a point to notice regarding the potential higher mechanical loads to be experienced by structural components and linked to that, due to the high shear stresses of the coolant, a potentially higher erosion rate of reactor components.

The low coolant moderation permits greater spacing between fuel pins, reducing core pressure losses, thereby increasing the coolant volume fraction in the elementary cell without a significant reactivity penalty. Pins can be spaced either by means of spacer grids or by a single wire which could be wound helically around each fuel pin, thereby providing support to the lattice, and preventing the cladding of adjacent pins from touching. The presence of wire wraps introduces additional challenges for the thermal-hydraulic modelling of the fuel. This affects not only the pressure drop but also the surface heat transfer and the transverse mixing of the flow between sub-channels [4].

In liquid metal-cooled reactors, the fuel assembly is generally encased in a duct, called *wrapper*, providing structural support to the assembly and separation of the coolant flow. The solution with the wrapper has the advantage of enabling different pressure losses through the various FA in order to control the radial distribution of core temperature, but it is disadvantageous from the neutronics viewpoint and requires a larger amount of steel in the core region. The gap between each assembly is open for coolant and the heat transfer from the wrapper to/from the flow in the gap can be important, especially under fault recovery conditions. For example, during passive decay heat removal conditions, transversal heating can become enhanced. However, modelling the process can be tricky because it is highly coupled with the flow in the upper plenum as well as temperature distribution within the core [4].

One of the most significant advantages of lead as a coolant is its low chemical reactivity. In comparison with other coolants, especially sodium and water, lead presents no huge energy release in the event of accident conditions. Usually, the possible fission products and erosive activated particles constitute a source of radioactivity inside the coolant, but LMs have a high retention capability and are inert with both air and water, implying a significant reduction of the potential energy stored in the primary system along with the elimination of fire risks on the nuclear island. The possibility of using air or water as ultimate heat sinks for Decay Heat Removal (DHR) systems without safety threats for their operation, drastically improves their reliability and offers potential for plant design simplifications. The ability of lead to trap and retain fission products, in particular iodine and caesium, constitutes an additional inherent barrier to the release of fission products during accidents. Intrinsic safety is guaranteed by the high compatibility existing between the oxide fuel and LMs and the fact that a loss of coolant accident will not result in

significant pressurization of the containment. In contrast to old reactors type, an LFR would not need to rely on backup power because passively operated DHRs are enabled by the natural circulation capabilities of the lead coolant and thanks to the relative chemical inertness of lead as a coolant, no hydrogen generation would be enabled during accidents [5].

The high operating temperatures and the corrosive properties of lead can result in a high rate of damage to traditional structural materials. In heavy liquid metal reactors, the corrosion of steel components constitute the main concern. During the selection of cladding material for FRs, the harder irradiation environment must be taken into account. New materials must be specifically tested, since Zircaloy, historically used as LWR cladding material, is unsuitable for that operating conditions due the insufficient strength at temperatures relevant for these reactors and to the the limited compatibility with LMs. The cladding material choice thus falls on steels. For what concerns the compatibility of alloys with LMs, the main problem is the solubility in lead of typical steel components such as Fe, Mn and especially Ni, causing a severe dissolution mechanism: with main effect of material loss and compromising of the cladding structural integrity. Viable countermeasures are the adoption of a proper coating layer on the outer cladding surface or the use of alloying elements such as Si, Cr and Al able to promote the formation of a protective oxide layer that must be controlled by proper insertion of oxygen content within the coolant which behaves as a corrosion barrier. The compatibility of ferritic/martensitic and austenitic steels with lead has been extensively studied, however, above 500°C, corrosion protection through the oxide barrier starts to fail, leading to an upper bound of coolant temperature of approximately 550 °C [2]. Above, in fact, the behaviour of lead usually changes from an oxidation/erosion mechanism to aggressive erosion/dissolution of steels grains and progressive irremediable damaging of structural components. Corrosion also pollutes the coolant with the possibility of oxide compound formation and subsequent sedimentation, with the risk of cooling channels plugging.

The different microstructure of austenitic stainless steel and ferritic-martensitic makes their behaviour different under the neutron environment. The advantages are:

- Ferritic-martensitic steels
    1. lower thermal stresses
    2. more resistance to irradiation creep
    3. more endurance to liquid metal corrosion and internal cladding corrosion
- Austenitic steels
    1. less embrittlement caused by neutron irradiation
    2. less susceptible to thermal creep

Due to materials' response to creep and swelling both the opposite alternatives are possible making the individuation of the best one not straightforward. Austenitic stainless steels have been typically preferred to ferritic/martensitic ones as a cladding material in Generation IV reactors in the light of their excellent mechanical behaviour at high temperatures over their swelling weakness. The high swelling suffered by austenitic steels has been partially solved with the development of double-stabilized 15-15Ti steels. When irradiation damage grows further than the sustained 120-130 DPA that can be withstood

by 15-15Ti, other promising steels like the ferritic T91 can be the optimal choice, but long-term irradiation experiments are still ongoing [6].

A viable alternative to lead is its alloy Lead-Bismuth Eutectic (LBE). The lower melting point of LBE brings operational advantages that made this a logical choice for some reactor designs. However is has some important drawbacks, namely: the availability and cost of bismuth; LBE is somewhat more corrosive than lead; final the greatest drawback of the LBE that its relatively large production of polonium-210, which is generated by neutron capture of bismuth[1].

1.2.3 The Advanced Lead-cooled Fast Reactor European Demonstrator

In view of the steps foreseen by the roadmap of the ELFR reactor in Europe, a full-scale demonstration plant, Advanced Lead-cooled FR European Demonstrator (ALFRED), is planned, to demonstrate the feasibility of the LFR technology. The main drawback hindering the industrial deployment of an LFR fleet in Europe is the lack of operational experience, mainly gained in the past by Russian military submarines. The proposed approach to industrial maturity is based on the concept of progressive up-scaling, starting with the zero-power GUINEVERE, then moving to the development of a small demonstrator reactor (ALFRED) for proving the viability of electricity production from LFR systems before moving to the First-Of-A-Kind representative of a commercial ELFR [7]. It should be noticed that, within this staged approach, ALFRED has the double function of a prototype for the ELFR large scale reactor and of a demonstrator for the LFR SMR.

Within the European collaboration for the project Lead-cooled European Advanced DEmonstration Reactor (LEADER), participated by 17 organizations, ALFRED was conceived as the optimum candidate to close the gap in this new nuclear technology and its commercial deployment, by addressing licensing challenges. To this aim, the chosen ALFRED configuration is based on three main pillars: increased safety margins, robust design and scalable size. [8]. To increase the knowledge in various field in preparation for ALFRED, the reactor components, thermal-hydraulics, and materials, will be under study in supporting facilities in 5 parallel projects: ATHENA & ChemLab, ELF & HELENA-2, Hands-ON, Meltin'Pot, Hub [9].

The reactor is a pool type with primary coolant of pure lead in forced circulation that can take advantage of natural circulation in emergency conditions. The core design has been driven by the aim of embedding all the features that are favourable in Design Extension Conditions, so to ensure the respect of the safety limits even in the most extreme conditions. The resulting core is calibrated to a thermal power of 300 MW with assemblies in a pseudo cylindrical scheme. The core inlet temperature is 400°C and the average core outlet temperature is 480°C. The core is composed as follows [10]:

[1]Could be so high that was seen in a small 80 MW, LBE-cooled ADS developed in the 5th Framework Program of Euratom, that the polonium inventory reaches 2 kg in the primary coolant circuit at equilibrium [3]

- 171 hexagonal fuel assemblies. Each element is made of 127 fuel rods in a triangular lattice inside a 4mm of thickness box, made of ferritic/martensitic steel T91; fuel is of MOX type, enriched at max at 30%.
- 12 control rods, used for both normal control of the reactor (start-up, reactivity control during the fuel cycle, power tuning and shutdown) and for scram.
- 4 safety rods, used only for reactor scram.
- 108 dummy elements, surrounding the others to shield the inner vessel from neutrons damage.

1.3 Reactor core modelling

1.3.1 General features of the core modelling problem

Neutronics and thermal-hydraulics are strongly coupled in a fission reactor. It is important to correctly understand the mechanisms through which these two phenomena influence each other, to accurately reproduce the reactor behaviour of the core in a numerical simulation. Neutron flux distribution has a strong influence on the thermal-hydraulics of the reactor by determining the power distribution for the heat transport equations, as well as on the thermo-mechanics, determining swelling and thermal expansions. The effect of the coolant and fuel temperature distributions on the other hand will affect the neutronics by thermal reactivity, changing materials' cross sections.

The high flux and high-temperature combination typical of an LFR requires, for safety reasons, accurate knowledge of the thermal field inside the sub assembly (SA). In line with the Generation IV philosophy, safety is embedded in the design phase from the very beginning, requiring that thermal-hydraulic conditions must be quantitatively assessed at the beginning of the design process. For these reasons, various reliable numerical simulations have been adopted, with complexity ranging from the 1D approaches down to detailed CFD simulations.

The core is the point of contact where all the fields Thermal-Hydraulic (TH), Neutronics (NE), Thermo-Mechanic (TM), and linked design of sub-components, fuel assembly, rods modelling, and materials choices, play together. For this reason, multiphysics models and numerical methods, during the design are intended to support the process.

Core design aims to determine the main parameters that define a reactor configuration providing the desired performances while complying with all the thermal-hydraulic, technological, thermo-mechanical, and economical constraints both in nominal and accidental conditions. Selecting bounds *a priori* is also in line with the defence in depth[2] philosophy through the combination of a safety-embedding system and proper safety margins to secure the source itself of potential hazards.

[2]The safety design based on Defence-in-Depth ensure the reliability of the plant, by design, in every state: normal operation, design basis accidents and design extension conditions. The main goals of the Defence-in-Depth approach is to assure the control of reactivity, remove of heat from the core, confine radioactive materials and limit accidental release [5].

The core thermal-hydraulic design, similarly to the FA one, needs a correct study of the flow distributions to prevent excessive gradients. This means assuring a suitable coolant flow outside the FA itself. Such a flow must be determined given the bypass, established by the thermo-mechanical design of the core base grid in order to provide i) the correct positioning of the elements in the core; ii) the concentrated pressure drops allowing for the assumed by-pass flow rate.

The neutronics design involves the core and the radiation shielding of the nearby structures along with the definition of the control and safety system layouts. The main purposes of the core neutronic design involve the definition of the fuel enrichment zoning which will allow guaranteeing the functionalities of the reactor, respecting all the constraints on the maximum burn-up along with requirements on the cladding and fuel temperatures; to guarantee a flat profile of temperatures, reduce the occurrence of the local hot spot and increase the overall fuel utilization. It is important to avoid hot spot factors which could result in locally enhanced corrosion. A flat temperature profile is also crucial for relaxing the mechanical interaction among the FA components and thus significantly easing its mechanical design.

To account for the above-mentioned phenomena simultaneously, multi-physics simulations are needed, which are usually performed with detailed codes such as coupled CFD-Monte Carlo like Serpent-OpenFOAM, that however still involve a prohibitive usage of computational resources. The use of a Monte-Carlo neutronic code in multi-physics calculations has grown up thanks to the flexibility of the continuous energy model, practically not doing approximations thanks to the proper use of cross sections and the capability to analyze reactor configurations with arbitrary geometrical complexity. However, one of the most important disadvantages is the extreme computational burden, in terms of both time and excessive memory demand, required to carry out a full-core Monte-Carlo simulation. For this reason, instead of Verification-Oriented Codes (VOCs) of this kind, simpler Design-Oriented Codes (DOCs) are preferable during design phases. According to the trade-off between complexity and accuracy, the computational time is reduced as a function of target constraints of the model type, for example focusing on just one portion of the problem [11]. A DOC should be designed targeting the following objectives:

- equilibrium; i.e., searching for a balance between the ability to reproduce experimental data and the complexity of the implemented models and code structure so to maintain a clear relationship among the various core parameters. To preserve the same grade of accuracy between parameters but also try to not increase modelling resolution on terms that are already poorly contributing to the uncertainty on the final results.
- short simulation time; this means opting for a numerical method able to efficiently operate on the particular scale of interest without requiring an excessively complex structure and computational cost.
- clear and focused application domain.

1.3.2 Special features of LFR core modelling

Developing a new code for an LFR, there are several aspects that are different with respect to the case of a light-water reactor, or in general with respect to reactors characterized by a thermal neutron spectrum. Some specific design characteristics allow the user to adopt

a less detailed approach, without loss of accuracy, for example taking advantage of the coolant incompressibility and remarkably high boiling point. Also, neutron kinetics of LFRs is, in general, less complicated than in thermal spectrum reactors. Specifically, since the neutron mean free path is of the order of several centimetres in a fast spectrum reactor, the neutron flux is less affected by the presence of the fuel rod. During a transient, the point-kinetic approximation of a constant flux profile could therefore be more valid than in a thermal spectrum reactor [4]. Coupled temperature-reactivity dynamics, on the other hand, are more complex in an LFR, with radial expansion of the fuel assembly diagrid that must also be properly modeled, and fuel Doppler coefficient that can be relevant.

Transients in Heavy Liquid Metal Cooled Reactor (HLMCR) are characterized by time scales, temperatures, and dynamics remarkably different from the other concepts due to the specific features of LMs as highlighted in section 1.2.2. Therefore, a precise calculation of the fuel temperature is not only of importance for determining the TH operating margins before the FA and fuel cladding failure, but also to obtain the correct reactivity feedback from Doppler.

1.3.3 The FRENETIC code

The Fast REactor NEutronics/Thermal-hydraulICs (FRENETIC) [12] code has been developed for fast multi-physics simulation of coupled neutronics/thermal-hydraulics for steady state and transients of liquid-metal cooled cores, arranged in hexagonal closed assemblies [13]. The purpose of the code is the simulation, during the design and verification phases, of the core behaviour without going into detail at the pin or sub-channel level. The required computational time is reduced by means of a suitable choice of the models adopted, namely:

- a multi-group diffusion model for neutrons, discretized with a coarse mesh nodal method at the assembly level.
- a 1D advection/diffusion model for the flow and heat transfer of liquid metal within each assembly, assuming no crossflow and taking into account the thermal coupling between neighbouring sub-assemblies.

In this way FRENETIC is able to solve the full-core NE-TH problem, providing an axial distribution of fuel and coolant temperature for each FA which is consistent with the fission power generation, evaluated by means of a coarse mesh 3D diffusion approach. The spatial detail of such a calculation is much lower than the one provided by a CFD-Monte Carlo approach, but it is sufficient to provide a reliable estimate of the temperature distribution in the core. The validation activities [14] and [13] carried out with the Politecnico have confirmed the capabilities of FRENETIC and highlighted several possible improvements to the code. For example, the possibility to use a new ad-hoc Thermal-Hydraulic (TH) module and add an extension to allow the simulation of assemblies with stagnant lead.

1.4 Aim of the thesis

From an accurate assessment of the current implementation, the necessity to refactor the FRENETIC adopting state-of-the-art and quality compliant code design strategies

appears clear. With this in mind, in the present thesis a new code design was proposed and implemented based on the usage of a finite volume discretization for the governing equations.

1.5 Thesis outline

This thesis is organized as follows: the first chapter 2 represents an introduction to the problem of designing a new TH module for the FRENETIC code, outlining the code functionalities and modelling strategies, as well as the calculation domain and its spatial discretization. Chapter 3 then describes the set of governing equations adopted to model the hexagonal assemblies (HA), their discretization and the numerical scheme for solving the problem. The following chapter, 4, describes the actual code design and implementation, entering into details of the code functional logic (section 4.1), re-adapted concepts and constitutive relations for the empirical parameters appearing in the equations are (section B) and programming guidelines that have been followed during the code implementation phase (section 4.3). Finally, in chapter 5 a thorough benchmark of the new TH module developed in the present thesis against the old one is presented, showing that results are physically consistent with the already validated previous implementation, and that solution robustness and efficiency have been significantly improved.

Chapter 2

Code design overview

2.1 A quality-oriented approach to code design

The software development was done following a series of steps according to current best practices, aiming at complying with high software quality standards [15]:

- Initial planning: definition of the general plan for the project and review of old implementation
- Requirements definition: identification of measurable requirements of the code to be developed and target goals.
- Software design: development of a conceptual design of the code, where models, mathematical formulation of equations, solver strategy selected to address the foregoing requirements are presented. The software design was reported in the Software Design and Implementation Document (SDID) produced for this new TH module of FRENETIC.
- Code development: coding and testing to check code functionalities.
- Code validation: the code validation phase involves comparing the code with experimental data and/or results of more accurate codes to assess the capability of the code within its anticipated validity domain. At this stage, validation was done comparing the code with the previous implementation testing performance and accuracy of results, section 5.1.

This chapter describes the main choices made for the new implementation of the code. Being part of the initial SDID of this new TH module for FRENETIC, here the requirements that should be fulfilled by the code, the working frame and solution strategies will be described. With respect to the initial design, some modifications were introduced to address problems that arisen during the development and to accomplish all the goals.

2.2 Problem description

The thermal-hydraulic module of FRENETIC aims at computing the axial evolution of the HA-averaged thermal-hydraulic parameters of the coolant (velocity, pressure, temperature) as well as of the average fuel temperature. The computed temperatures are used in order

to compute the new cross sections for the NE module. To this aim, the thermal module solve the mass, energy, and momentum conservation equations for each HA, given the power distribution and boundary conditions.

The temperature dependence of macroscopic cross sections for a given material is obtained through bi-variate linear interpolation with respect to the fuel and the coolant temperatures, that characterize the computational volume in which the material is present [13]:

$$\Sigma(T_c, T_f) = \Sigma(T_{c,0}, T_{f,0}) + \left(\frac{\partial \Sigma}{\partial T_f}\right)_{T_c} (T_f - T_{f,0}) + \left(\frac{\partial \Sigma}{\partial T_c}\right)_{T_f} (T_c - T_{c,0}) \tag{2.1}$$

The nuclear data were generated in previous works with coupling procedures with Monte Carlo-Serpent-2 code, used to collapse the continuous energy data into six groups and to homogenize them over the reactor heterogeneous regions. That was done by means of a suitable spatial homogenization procedure, which was carried out in order to preserve the reaction rate for each material[13].

The NE and TH modules run independently, with their characteristic time steps. They exchange data at user-selected time steps, by successive interchanging of the required physical quantities: temperatures and power deposition. This is done according to the adopted neutronics model, which allows to distinguish different time scales between fast and delayed neutrons, and their effects on the changing amplitude and shape of the power distribution. In this way, thermal feedback can be predicted during transients.

2.3 Calculation domain

The calculation domain extends axially from the beginning to the end of the assemblies in the region at a constant cross section, after the bypass holes and enlargement of the bases, see figure 2.1 for a schematic representation. Radially, the Inter-Wrapper (IW) region between an assembly and the neighbouring one is treated as stagnant lead. The fraction of mass flow flowing here is about 5% of the inlet to the fuel assembly (FA). In normal situations, with the flow in a turbulent regime and no flow blockage, the mass flowing here does not significantly contribute to inter-assembly heat transfer. With the aim to treat the coupling in a modular way, there is the possibility to insert a pseudo resistance, instead of the conduction one, to take care of a different convection behaviour of the flowing lead. The presence of this inter-wrapper flow can have relevant importance also in the transverse direction during off-normal situations that require an overall 3D evaluation of the core thermal-hydraulics. However, these situations are out of the scope of FRENETIC and therefore were not considered.

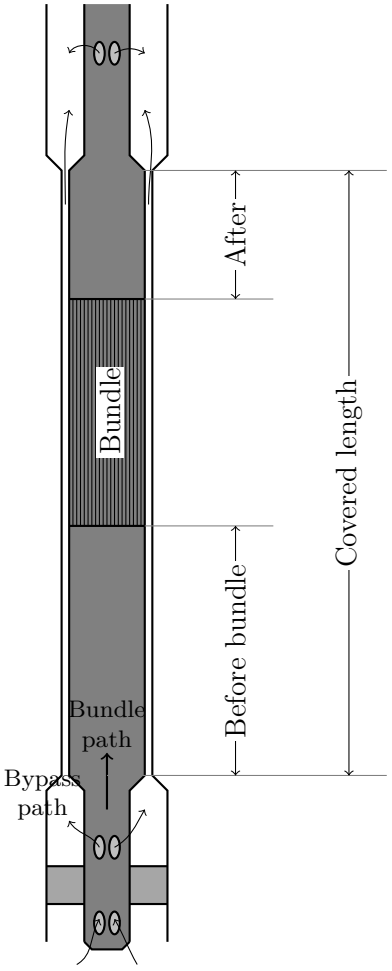


Figure 2.1: Schematic of the bypass flow domain, adapted from [11].

2.4 Required spatial resolution and selected modelling approach

According to the discussed requirements, the TH calculation in FRENETIC needs to compute the average pressure and velocity, the average coolant temperature, and the average fuel temperature, for each assembly at each axial position. The core will be modeled taking care of the different HAs with their own fuel and control rods distributions. Instead, the possible presence of the core barrel, shield, and walls can be considered by means of equivalent assemblies that can be disregarded in the TH solution but are relevant to be considered for their impact on neutronics. Due to the requirement of 1D resolution, the hydraulics can be approached from the single channel point of view, adopting literature correlations for friction factors and heat transfer coefficients. The estimated time scales have shown that since the thermal coupling between neighbouring hexagons is relatively weak compared to the coolant transit along the fuel assembly, the 3D problem can be treated as a series of different parallel weakly coupled 1D problems with similar geometry and physics. The validity of this approach was already discussed in previous work [12]. In order to possibly improve the current implementation, a dynamic model that resolves additional axial equations for dummy rods can be adopted. It should be used during transient totext take care of the heat capacity of non-fuel rods, which currently are neglected in the energy equations but just count in the computation of wetted perimeters and flow area. The possibility to compute in time, with a radial dumped approach, the temperature dependence of non-heated rods can smooth the unphysical coolant temperature peaks that now are found in some cases, e.g. computation of EBR-II core [14].

2.5 Literature review

The full 3D thermal-hydraulic solution is actually achieved by splitting the problem in 1D (axial) + 2D (radial). This approach was already studied in previous experience of the PoliTo research group in the framework of the TH modelling of superconducting magnets for nuclear fusion reactors [16]. The TH module was therefore taken from those previous works and adapted to the requirements of FRENETIC. It solves implicitly the transient 1D mass, momentum and energy balances for each HA in the z direction, then each channel is radially coupled with its surrounding adjacent neighbors by a simple resistive model, to obtain a full-core 3D solution. Linear finite elements are used for the spatial discretization of the set of equations, creating the matrices and resolving the coolant and fuel equations separately. The resulting set of ordinary differential equations is linearized exploiting the frozen coefficients assumption and the coupling matrix is treated implicitly. This maintains the banded structure of the problem preserving high efficiency and decreasing CPU time. Additionally, the various blocks of the problems are decoupled, so that the solution can be computed sequentially for the different blocks.

2.6 Spatial discretization

In the new version of the FRENETIC TH module, the domain discretization is performed with a finite volume approach. Each HA is axially subdivided into a number of control volumes that can be built with a possibly different distributions, for instance to increase the axial resolution in the active core region. We could preserve the presence of a space-dependent cross sectional area, A_i, inside the equations, to leave it general and to allow, in the future, to compute regions with different cross-sectional areas. A smooth transition between two differently refined zones could be implemented, but at the moment a stepwise variation that was present in the previous code version is retained. The volume is discretized with staggered mesh, so the material properties and temperatures are evaluated at the cell centre, determined at the mean of the representative volume, while the velocities are evaluated at cell faces.

2.7 Boundary conditions

The domain of interest is the one between the upper and lower plenum. The code should accept a user-specified distribution of the mass flow rate per HA, determined according to e.g. flow split calculations that are outside the scope of the codes. Accepted boundary conditions must be allowed to be distributed in space, and possibly also varying in time: temperatures in inlet, mass flow with inlet pressure or outlet pressure for the various HAs. Boundary assemblies, i.e. the ones representing the barrel and the lead outside it, are not truly solved. Indeed, a detailed solution would require complex 3D calculations well beyond the scope of FRENETIC, which is concerned with the core region. However, since these regions must be present in the NE module, they must also appear in the TH module (for the moment a logic to simply avoid them in one of the two modules is not available). A viable strategy might be to impose the same temperature of the inlet if the mass flow is zero. Moreover, the coarse-mesh diffusion solver adopted in the NE module is limited in accuracy in boundary regions characterized by strong gradients. However, being far from the region of interest of FRENETIC, these simplifications are acceptable. The barrel and external lead have therefore been simulated in FRENETIC by adding fictitious HAs corresponding to the latter two regions. In this way, the outer zone of the core has been made fully consistent with the one employed in the Serpent model.

The code must also have the possibility of receiving an externally provided heat generation varying in space and time as a result of neutronic flux distribution. As a *goal*[1], there could be the possibility to manage also pressure drop in input, computing the mass flow rate distribution consequently. In that case, the code should provide a hypothesis on initial mass flow rates and from that could enter the loops to solve the selected equations. The code is provided in this case with under-relaxation factors to reach convergence during iterations.

[1] *Goals are weaker requirements that will represent a plus if implemented, otherwise it will be left for future work.*

Chapter 3

Governing equations and discretization

3.1 General assumptions

Classical momentum, mass and energy equations are considered with proper simplifications done according to the desired model resolution and assumptions. The control volume is fixed and no mass flow and momentum exchange in the transverse direction is expected. The problem can be solved either in steady state or in transient conditions, with different approaches regarding coupling effects. The channel is assumed to be vertical, therefore gravity only contributes as a body force in the axial momentum equation.

3.2 Conservation of mass

The spatial discretization adopted will follow the nomenclature in figure 3.1, using standard finite volume method [17].

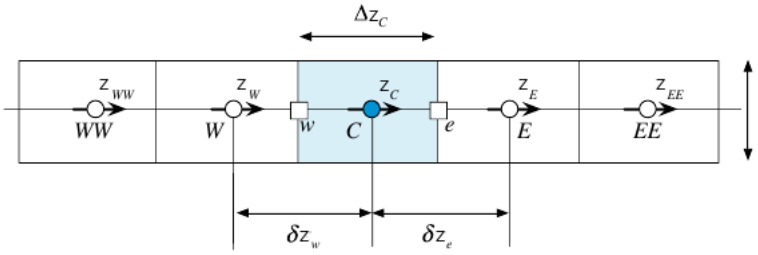


Figure 3.1: Spatial nomenclature for mesh discretization.

The equation for conservation of mass is generally expressed as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot [\rho v] = 0 \tag{3.1}$$

Specializing it for one-dimensional flow and integrating over the volume for the generic C nodal position:

$$\begin{gathered} \frac{d\rho}{dt} + \rho\frac{dU}{dz} = 0 \\ \int \frac{d\rho}{dt}dV + \int \rho\frac{dU}{dz}dV = 0 \end{gathered} \tag{3.2}$$

Due to the incompressibility of the flow, the first term should be zero also in time, while the second, taking advantage of integrals identity and mass flow rate definition, can be expressed as a function of mass flow at volume's faces:

$$\dot{m}_e - \dot{m}_w = 0 \tag{3.3}$$

3.3 Conservation of momentum

The same kind of procedure is done for the conservation of momentum in z direction

$$\frac{\partial[\rho v]}{\partial t} + \nabla \cdot [\rho v v] = -\nabla p + \mu \nabla^2 v + f_b \tag{3.4}$$

The equation for conservation of momentum is particularized for the one dimension and integrated over the volume for the generic C nodal position.

$$\begin{aligned} &Vol\frac{d(\rho_C U_C)}{dt} + \dot{m}_e U_e - \dot{m}_w U_w = -\int \frac{dp}{dz} dV - \rho_C g Vol - \frac{1\Delta z_C}{2D_h} f_C \dot{m}_C U_C - \Delta P_{loc} A_{av} \\ &\frac{d\dot{m}_C}{dt} + \frac{\dot{m}_C^2}{\Delta z_C}\left(\frac{1}{\rho_e A_e} - \frac{1}{\rho_w A_w}\right) = \\ &\qquad = -\frac{\Delta p}{\Delta z_C} A_{av} - \rho_C g A_{av} - \frac{1}{2D_h A_{av} \rho_C} f_C \dot{m}_C^2 - k_{loc} \frac{\dot{m}_C^2}{2\rho_C A_{av} \Delta z_C} \end{aligned} \tag{3.5}$$

where:

$$\frac{1}{\Delta z_C}(\dot{m}_e U_e - \dot{m}_w U_w) = \dot{m}_C^2 \frac{d}{dz}\left(\frac{1}{\rho_C A}\right) = \frac{\dot{m}_C^2}{\Delta z_C}\left(\frac{1}{\rho_e A_e} - \frac{1}{\rho_w A_w}\right) \tag{3.6}$$

The equation is written taking advantage of the mass flow conservation, and so rewritten in terms of mass flow rate, instead of velocities. The dissipation term is written as a function of the Darcy–Weisbach friction factor, which is consistent with the selected approach. The presence of gravity as a body force was taken into account, and the possible presence of a localized pressure drop can be included by providing in input the desired generic k_{loc}, e.g. to consider the presence of spacer grids in case of non wire-spaced assemblies. The integral of pressure drop is discretized in terms of adjacent nodal values with a finite difference approach. For the sake of generality, a distinction can also be done for the areas, where A_e and A_w are the values at cell faces, while A_{av} is an average, so possible area changes could be considered. Different would be also the values of density, between the central average value and the ones at cell faces, that can be computed for example with central discretization as:

$$\rho_e = g_C \rho_C + (1 - g_C)\rho_E = \frac{\Delta z_C}{\Delta z_C + \Delta z_E}\rho_C + \frac{\Delta z_E}{\Delta z_C + \Delta z_E}\rho_E \tag{3.7}$$

in terms of near central cells values, with weight factors to take care of a different possible axial discretization.

3.4 Conservation of energy for coolant

A general form of the energy conservation equation in terms of temperature is considered from [17]:

$$\frac{\partial(\rho c_p T)}{\partial t} + \nabla[\rho c_p v T] = \nabla[k \nabla T] + \dot{q}_v + \rho T \frac{D c_p}{D t} - \left(\frac{\partial(Ln\rho)}{\partial(LnT)}\right)_p \frac{Dp}{Dt} + \lambda\psi + \mu\Phi \tag{3.8}$$

The generic equation, which takes care of pressure and temperature dependence for the specific enthalpy in its derivation, can be particularized for the study case, neglecting some terms of minor importance and others that are zero due to the coolant incompressibility.

- The dissipation term Φ has negligible values with respect to the heat generation in the fuel, also due to the absence of high-velocity gradients.
- The continuity equation implies that $\psi = 0$.
- Since the density is constant inside the volume, also the pressure term is negligible.

Since changes in the evaluation of c_p can be relevant, as a function of the adopted materials correlations, the total derivative of c_p cannot be neglected from equation. The resulting adopted equation is:

$$\frac{\partial(\rho c_p T)}{\partial t} + \nabla[\rho c_p v T] - \nabla[k \nabla T] - \rho T \frac{D c_p}{D t} = \dot{q}_v + \frac{\Pi h^*}{A_{av}}(T_{fuel} - T) + \sum \frac{Q}{A_{av}} \tag{3.9}$$

and discretized as:

$$\begin{aligned} Vol\frac{d(\rho_C c_{p,C} T_C)}{dt} + (\dot{m}_e c_{p,e} T_e - \dot{m}_w c_{p,w} T_w) - k_e A_e \left.\frac{dT}{dz}\right|_e + k_w A_w \left.\frac{dT}{dz}\right|_w = \\ = \rho_C T_C \left[\frac{\partial(c_{p,C})}{\partial t} + \frac{\dot{m}_C}{A_C \rho_C}\frac{\partial c_p}{\partial z}\right] Vol + \dot{q}_{v,C} Vol + \\ + \Pi_f \Delta z_C h^* (T_{f,C} - T_C) + \sum Q_C \Delta z_C \end{aligned} \tag{3.10}$$

$$\begin{aligned} Vol\frac{d(\rho_C c_{p,C} T_C)}{dt} + (\dot{m}_e c_{p,e} T_e - \dot{m}_w c_{p,w} T_w) - k_e A_e \frac{2(T_E - T_C)}{\Delta z_C + \Delta z_E} + k_w A_w \frac{2(T_C - T_W)}{\Delta z_C + \Delta z_W} + \\ - \rho_C T_C \left[\frac{\partial(c_{p,C})}{\partial t} + \frac{\dot{m}_C}{A_C \rho_C}\frac{c_{p,e} - c_{p,w}}{\Delta z_C}\right] Vol = \\ = \dot{q}_{v,C} Vol + \Pi_f \Delta z_C h^* (T_{f,C} - T_C) + \Pi_{nf} \Delta z_C h (T_{nf,C} - T_C) + \sum Q_C \Delta z_C \end{aligned} \tag{3.11}$$

The diffusion term is discretized with a finite difference approach and is pointed to cell faces. The advection term is written with a second order *upwind* scheme to avoid numerical instabilities caused by the central difference scheme found during coding verifications. Temperatures at cell faces are computed as follows, taking care of the possibly different mesh distribution in axial direction:

$$\begin{aligned} T_e &= T_C + \frac{(T_C - T_W)\Delta z_C}{\Delta z_C + \Delta z_W} \\ T_w &= T_W + \frac{(T_W - T_{WW})\Delta z_W}{\Delta z_{WW} + \Delta z_W} \end{aligned} \tag{3.12}$$

Material properties at cell faces can be defined by linear interpolation, weighed according to the spatial discretization step:

$$c_{p,e} = g_C c_{p,C} + (1 - g_C) c_{p,E} = \frac{\Delta z_C}{\Delta z_C + \Delta z_E} c_{p,C} + \frac{\Delta z_E}{\Delta z_C + \Delta z_E} c_{p,E} \tag{3.13}$$

For the conduction constants, is used an equivalent resistance scheme, to better treat the diffusion process, in the form:

$$\frac{1}{k_e} = \left(\frac{1-g_e}{k_E} + \frac{g_e}{k_C}\right) = \left(\frac{\Delta z_E}{k_E(\Delta z_C + \Delta z_E)} + \frac{\Delta z_C}{k_C(\Delta z_C + \Delta z_E)}\right) \tag{3.14}$$

In the equation (3.11) a possible volume heat generation term is considered, for example to possibly account in the future the effect of gamma rays heat deposition outside the rods. The heat transfer terms, for the coupling with fuel rods and other HAs, are treated as in the previous version of the code. In the steady solver, a fictitious term is used to take care of the fact that the average temperature of the fuel is different from the surface temperature seen by the coolant.

Π is the wetted perimeter, and h* is computed as:

$$h^* = h_C \frac{T_{s,C} - T_{pb,C}}{T_{f,C} - T_{pb,C}} \tag{3.15}$$

Possible convective heat transfer to non-fuel rods can be estimated during transients through an iterative procedure of separate time-dependent energy equations for these rods. The term Q_C contains the coupling between an assembly and the neighbouring ones, the loop for the coupling can be up to 6, so specified for hexagonal geometries without loss of generality.

$$Q_C = \Pi_{ij} h_{ij} (T_{C,i} - T_{C,j}) \tag{3.16}$$

where the h_{ij} is the heat transfer coefficient, which will contain information on a series of thermal resistances to model the inter-wrapping gap and walls, between hexagon i and hexagon j.

$$R_{tot} = \frac{1}{h_{conv,i}} + \frac{s_a}{k_{box}} + \frac{s_c}{k_{IW}} + \frac{s_a}{k_{box}} + \frac{1}{h_{conv,j}} \tag{3.17}$$

3.5 Conservation of energy for fuel rods

An energy conservation equation in terms of temperature for solid materials is used for the rods, using heat generation, and coupling with coolant, in the vertical direction, in the form:

$$\frac{d(\rho c_p T)}{dt} - \nabla[k \nabla T] = \dot{q}_v + \frac{\Pi h^*}{A_f}(T_{cool} - T_{fuel}) \tag{3.18}$$

and integrating using the same nomenclature in figure 3.1:

$$\begin{aligned} &Vol\frac{d(\rho_c c_{p,C} T_{f,C})}{dt} - k_e A_e \left.\frac{dT_f}{dz}\right|_e + k_w A_w \left.\frac{dT_f}{dz}\right|_w = \dot{Q}_f \Delta z_C + \Pi \Delta z_C h^* (T_{cool,C} - T_{f,C}) \\ &Vol\frac{d(\rho_c c_{p,c} T_{f,C})}{dt} - k_e A_f \frac{(T_{f,E} - T_{f,C})2}{\Delta z_C + \Delta z_E} + k_w A_f \frac{(T_{f,C} - T_{f,W})2}{\Delta z_C + \Delta z_W} = \\ &= \dot{Q}_f \Delta z_C + \Pi_f \Delta z_C h^* (T_{cool,C} - T_{f,C}) \end{aligned} \tag{3.19}$$

where $\dot{Q}_f$ is the linear heat generation distribution (derived by the NE module or read from input files), Π_f is the total rod perimeter and A_f the fuel cross sectional area, which is treated as constant in the channel. The diffusion term can be discretized like coolant ones with a finite difference approach (3.14). The surface temperature can be analytically linked to the heat generation rate and mean value, assuming a parabolic radial temperature distribution inside the pin. This is strictly correct in steady state where the axial heat conduction is neglected, but can also be adopted when there are small variations of the source or of the boundary conditions.

The surface temperature is defined as follows, for the solid pin:

$$T_s = T_f - \frac{\dot{q}_v r_{pin}^2}{8k} \tag{3.20}$$

and for the annular ones, with internal radius r_v and imposed adiabatic inner side:

$$T_s = T_f - \frac{\dot{q}_v (r_{pin}^2 - r_v^2)}{8k} \left[1 - \frac{ln(r_{pin}/r_v)^2}{(r_{pin}/r_v)^2 - 1} \right] \tag{3.21}$$

Due to the h* approach, the steady matrix cannot be solved in one step, due to the fact that the surface temperature is as a function of the actual distribution that has to be assumed. Moreover, due to the initialization of temperatures profiles, there could be the possibility to have a zero value of h* that would lead to the calculation of temperatures of pins larger than the melting point.

To avoid that, two possibilities are given: a ramp-up of the power, heavily influenced by the ramp step and under-relaxation value for temperatures; a pseudo transient, that usually leads to a fast convergence, in which time terms are used also in the steady case.

The time step used can be derived from the actual physical characteristics of the channel, finding the pseudo transient correct time steps from the specific heat diffusion and momentum diffusion values of the HA. As already done in common codes [18] the time step is chosen keeping the minimum between:

$$\begin{aligned} \Delta t &= \frac{L_f}{\alpha_f} \\ L_f &= min(\sqrt{Vol_f}, max(L_{chan}), \sqrt{A_f})) \\ &\text{and user defined : } L_f = D_{fuel} \end{aligned} \tag{3.22}$$

from the solid material, and for the coolant terms as

$$\begin{aligned} \Delta t &= \frac{L_{cool}}{\nu} \\ \nu &= \frac{\mu}{\rho} \\ L_{cool} &= min(\sqrt{Vol_{cool}}, max(L_{chan}), D_{eq})) \end{aligned} \tag{3.23}$$

The pseudo transient practically consists in adopting an under-relaxation factor that is different from cell to cell. Adopting an implicit Euler scheme for the time discretization

in an iterative system, both methods practically imply an additional source term and contribution to central coefficient A_p; the following similarity between under relaxation ϕ and time step can be seen[19]:

$$\Delta t = \frac{\rho\phi\Delta\Sigma}{A_P(1-\phi)} \qquad \phi = \frac{A_P\Delta t}{A_P(\Delta t + \rho\Delta\Sigma)} \tag{3.24}$$

The application of under-relaxation values is still turned on, to avoid potential problems in start-up procedures in case the power deposition is high.

During transients, the theoretical mean value approach could lead to bad results in terms of expected Ts and related flux coming out from the pin, leading to nonphysical situations such as Ts lower than the coolant temperature or to have an initial reduction of expected Ts during a ramp up of the power deposition. Therefore, a different approach is required. Due to the characteristics of the fuel rod, most of the gradients occur in the radial direction, whereas in axial direction gradients are negligible. So, taking care of the cylindrical shape allows us to safely assume an axially symmetric condition; it was indeed suggested to compute a pure radial model at each axial step to compute the real radial profile inside the pin and the relevant temperatures - maximum, superficial, and main integral - to be provided in output. In this way it was achieved a 1.5D accuracy in the evaluation of these imaginary average rods that represent the temperature profiles for fuel in the HA. The corresponding differential equation is:

$$\frac{d(\rho c_p T)}{dt} - \frac{1}{r}\frac{d}{dr}\left(rk\frac{dT}{dr}\right) = \dot{q}_v \tag{3.25}$$

In finite volume formulation

$$\begin{aligned} Vol\frac{d(\rho_C c_{p,C} T_C)}{dt} - \frac{1}{r_C}\left(rk\frac{dT}{dr}\bigg|_e - rk\frac{dT}{dr}\bigg|_w\right)\Delta z_C &= \dot{q}_{v,C} Vol \\ r_C Vol\frac{d(\rho_C c_{p,C} T_C)}{dt} - \left(r_e k_e\frac{(T_E - T_C)2}{\Delta r_C + \Delta r_E} - r_w k_w\frac{(T_C - T_W)2}{\Delta r_C + \Delta r_W}\right)\Delta z_C &= r_C\dot{q}_{v,C} Vol \end{aligned} \tag{3.26}$$

While in a steady state the simplification of the theoretical profile helped to reduce the computational effort, the radial discretization leads to an iterative procedure due to the coupling between the coolant and the need to impose a convective boundary condition on the outer fuel side. The transient radial profile is computed with a finite element formulation, taking advantage of the current implementation, that already has been developed adding this model approach as alternative of the pure 1D. The common matrix system is solved:

$$\begin{aligned} [C]\{\dot{T}\} + [K]\{T\} &= \{p\} \\ \left(\frac{[C]}{\Delta t} + \frac{1}{2}[K]\right)\{T\}_{m+1} &= \left(\frac{[C]}{\Delta t} - \frac{1}{2}[K]\right)\{T\}_m + \frac{1}{2}\{p\}_m + \frac{1}{2}\{p\}_{m+1} \end{aligned} \tag{3.27}$$

in which the element matrices and external heat load vector are generically formulated, by

the use of shape functions [20], as:

$$\begin{aligned}[C] &= \int_{\Omega} \rho c [N_\theta]^T [N_\theta] d\Omega \\ [K] &= \int_{\Omega} [B_\theta]^T [k] [B_\theta] d\Omega \\ \{p\} &= -\int_S [N_\theta]^T \hat{h} dS + \int_{\Omega} Q [N_\theta]^T d\Omega \end{aligned} \tag{3.28}$$

The radial pin model is also necessary in case the fuel rods are modeled taking care of the possible different layers for the internal gas gap and cladding materials. In this case, even if the mean approach can take care of the presence of an additional thermal resistance, due to simplifications done, the average temperature field could be not so aligned with what is expected with the pure radial model, leading to different possible choices in the modelling treated in section 4.6.1.

3.6 Thimble modelling

To enhance the capabilities of the FRENETIC code, in the past modelling of different kinds of assemblies has been developed. In particular, a specific geometry for the Experimental Breeder Reactor-II was introduced, in which some of the assemblies have a smaller hexagonal box inside the main one. As already studied with the ENEA group [14], in case of the presence of a thimble between the inner and outer box, the heat transfer modelling to the next fuel assemblies should consider the presence of this flowing coolant. The fact that the thimble's flow is non-negligible, in some cases even of the same order of magnitude as the inner box flow, led to the actual so-called "Box-in-the-Box (BiB)" model, in which convective heat transfer of this zone is added to the thermal resistance of inter-wrapper gap. The convective heat transfer coefficient used is computed from the Seban-Shimazaki formula, see section B.2. The temperature for the evaluation of the channels' heat transfer coefficients is currently found by modelling an additional channel, with no power generation, thermally connected to the main inner box and near assemblies by proper heat transfer model [21].

The thermal resistance for the coupling from channel to thimble is:

$$R_{int->thimb,i} = \frac{1}{h_{conv,i}} + \left(\frac{s}{k}\right)_{inbox} + \frac{1}{h_{conv,thimb,i}} \tag{3.29}$$

from thimble to near assemblies:

$$R_{tot} = \frac{1}{h_{conv,thimb,i}} + \left(\frac{s}{k}\right)_{outbox,i} + \left(\frac{s}{k}\right)_{clearance} + \left(\frac{s}{k}\right)_{outbox,j} + \frac{1}{h_{conv,j}} \tag{3.30}$$

The same is done in the new code with some modifications. In particular, now the thimble is directly solved together with the main channel. The equation computed would be again (3.11) without coupling with rods and proper coupling with the near assemblies. The possibility to decrease the computational effort, just considering a heat transfer coefficient at the main temperature of the channel, was discarded, since the differences in

temperatures can be relevant. For example, in the case of instrumented assemblies, in a typical reactor-relevant test case, the values of the thimble profile could be 80°C larger than main channel, with a resulting expected heat transfer coefficient that is two times lower than the actual one. Also, the profiles are different in shape, with peaks on the side of the thimble near the outlet, and the exit region of main channels characterized by an equilibrium thermal exchange. So, to achieve good temperature distributions will be necessary to correctly treat the thimble TH.

3.7 Boundary conditions

Suitable boundary conditions for both the advection and conduction discretizations are adopted. To close the one for the fuel, we can consider that the last part is far from the heated zone and mainly constituted by cladding closure and gas in real rods, so we can reasonably just consider zero heat flux from the last transverse area, being far enough from any possible gradients in temperature. However, on one side, at the first node, a convective boundary conditions is employed to avoid problems during start-up of the steady state

$$T_w = \frac{h\, T_{cool,in} + \left(\frac{k_w}{\Delta z_C/2}\right) T_C}{h + \frac{k_w}{\Delta z_C/2}} \tag{3.31}$$

A similar approach can be applied to the diffusive terms in the coolant equation, while convective boundary conditions are naturally given by imposing the inlet temperatures from the input. Also, to avoid problems in case of null flow rate or no initial heat transfer to the coolant, an additional conductive boundary condition is given for coolant to the first node, adding an additional term to the central value and source vector in the form:

$$\begin{aligned} T_w &= \frac{k_w}{\Delta z_C/2} \\ b_C &= \frac{k_w}{\Delta z_C/2} \cdot T_{in} \end{aligned} \tag{3.32}$$

The radial discretization have imposed an adiabatic boundary conditions on the inner side and a Robin on the outer side.

3.8 Solution scheme

3.8.1 Transient solution

During transient simulations, the governing equations are discretized in both space and time, with the possibility to adopt different solution schemes. An explicit one provides high computational efficiency and simplifies the parallelization of the computational procedure. Yet only few commercial codes have adopted this approach due to a limitation on the size of Δt, due to the stability-limited maximum Δt dependence on the spatial discretization. Nonetheless, implicit is not an ideal scheme as it is of low order and solutions obtained with this scheme are of low accuracy unless small time steps are used[22]. However, in the specific problem solved by FRENETIC, we are forced by the neutronic side to use small time steps, in order to have good accuracy in the time changing of temperature profiles and related cross sections of materials, since the coupling between the two monolithic codes is explicit. The adopted explicit approach for data exchange implies serial calculations, with no actual check on the results provided by the codes; results are used as new boundary conditions for the other code, and vice versa, without any kind of control before proceeding to the next synchronization point. So, if the time step is not sufficiently small, the obtained prediction could be wrong with the delay in power distributions and response of the system to reactivity insertion.

Due to the relatively weak coupling between an assembly and the neighbouring ones, we can treat them as a set of explicitly coupled problems. A coupling done in this way has its own reason in the different time scales and force of the thermal process. Meanwhile, we can use a more stable implicit scheme for the other terms of equations of fuel and coolant, without so increasing the computational cost excessively, with an approach similar to the one already implemented in the M&M code [16]. This leads to decoupling the equations of each assembly at time t_{i+1} from those of the neighbouring ones, with the possibility to solve the system of equations in different ways. It would even be possible to parallelize the solution, grouping the assemblies and assigning them to different cores. Also, there is more flexibility in memory management. Thus, the use of monolithic methods was ruled out preferring the possibility of subdivisions.

In the case of Box-in-the-Box (BiB) assemblies, the inter-channel coupling is done explicitly while the coupling with the main channel is computed implicitly. Indeed, the thickness of the box in this case is usually smaller than the standard box, leading to a stronger coupling between the two flows. With the aim to have a code that can increase the characteristic time step according to time adaptivity, it is better to have an implicit coupling with the main channel, in order to avoid stability-limited time steps Δt. In this way, the two flow fields are iteratively brought at convergence together.

The four sets of equations of each assembly, analysed in section 3, once properly arranged and simplified by the approach, turn out to be decoupled, apart from physical properties dependence on temperature. The decoupling considerably simplifies the numerical solution scheme since, knowing the inlet mass flow, the energy equation can be resolved, so that properties for the momentum equation are found, and pressure distribution can be computed.

3.8.2 Steady state solution

A different kind of approach is needed for the steady state case. The terms associated to the finite difference discretization in time disappear, therefore it is no longer possible to exploit an explicit treatment of the inter-assembly coupling, so that each node now is coupled with the 6 nearby. Different solution schemes can be used:

- Build up a matrix containing the entire system. This would be very expensive from the memory and computation point of view. For example, for a core of 171 assemblies and 350 axial steps, the memory needs of the solution matrix could be predicted as: $(171 \cdot 350 \cdot 2)^2 \cdot byte = 14 \cdot 10^9 \cdot 64/8 \approx 114GB$;
- Adopt a pseudo transient as currently done;
- Adoption of an iterative procedure, changing the transient solution scheme from iteration over time to one for convergence: removing the time terms, starting with an assumption of no heat transfer between assemblies, and then iterating the entire core solution up to convergence, using for the transversal coupling the profile of temperatures found at the previous step.

These last procedures is implemented. The steady-state solver is used also for the initialization and profile determination of temperatures for the initial condition of the transient solver.

3.8.3 Matrix formulation

The two coupled energy equations for each assembly, during steady state, are intended to be solved together in a matrix system in the form:

$$\begin{bmatrix}
\ddots & & & & & & & \vdots \\
\cdots & BB_{i-1}+EE_{i-1} & -EE_{i-1} & CC_{i-1} & 0 & 0 & \cdots & \cdots \\
\cdots & -ee_{i-1} & bb_{i-1}+ee_{i-1} & 0 & cc_{i-1} & 0 & 0 & \cdots \\
\cdots & AA_i & 0 & BB_i+EE_i & -EE_i & CC_i & 0 & \cdots \\
\cdots & 0 & aa_i & -ee_i & bb_i+ee_i & 0 & cc_i & \cdots \\
\cdots & DD_{i+1} & 0 & AA_{i+1} & 0 & BB_{i+1}+EE_{i+1} & -EE_{i+1} & \cdots \\
\cdots & \cdots & 0 & 0 & aa_{i+1} & -ee_{i+1} & bb_{i+1}+ee_{i+1} & \cdots \\
\vdots & & & & & & & \ddots
\end{bmatrix} \cdot$$

$$\cdot \begin{bmatrix} \vdots \\ \vdots \\ TT_{i-1} \\ TT_{f,i-1} \\ TT_i \\ TT_{f,i} \\ TT_{i+1} \\ TT_{f,i+1} \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \vdots \\ QQ_{i-1} \\ QQ_{f,i-1} \\ QQ_i \\ QQ_{f,i} \\ QQ_{i+1} \\ QQ_{f,i+1} \\ \vdots \\ \vdots \end{bmatrix} \tag{3.33}$$

The resulting system is diagonally dominant, with the unknowns vector written keeping at the same axial index coolant and fuel rods. The source terms contain not only heat generation terms, but also known quantities and terms from coupling with other channels. The terms DD, AA, aa, BB, bb, CC, cc contain the multiplicative terms of advection, conduction, etc. from equation (3.11) and (3.19), here indicated with upper and lower case

respectively the ones to coolant and fuel equations. A list of the term used is reported, for coolant:

$$
\begin{aligned}
DD &= +\dot{m}_w c_{p,w} \left(\frac{\Delta z_W}{\Delta z_{WW} + \Delta z_W} \right) \\
AA &= -\dot{m}_w c_{p,w} \left(1 + \frac{\Delta z_W}{\Delta z_{WW} + \Delta z_W} \right) - \dot{m}_e c_{p,e} \left(\frac{\Delta z_C}{\Delta z_C + \Delta z_W} \right) - k_w A_w \frac{2}{\Delta z_C + \Delta z_W} \\
CC &= \dot{m}_e c_{p,e} \left(1 + \frac{\Delta z_C}{\Delta z_C + \Delta z_W} \right) - k_e A_e \frac{2}{\Delta z_C + \Delta z_E} \\
BB &= \dot{m}_e c_{p,e} \left(1 - \frac{\Delta z_C}{\Delta z_C + \Delta z_E} \right) - \dot{m}_w c_{p,w} \left(\frac{\Delta z_W}{\Delta z_C + \Delta z_W} \right) + k_e A_e \frac{2}{\Delta z_C + \Delta z_E} \\
&+ k_w A_w \frac{2}{\Delta z_C + \Delta z_W} - \rho_C Vol \left(\frac{\dot{m}_C}{A_C \rho_C} \frac{c_{p,e} - c_{p,w}}{\Delta z_C} \right) + \Pi_{nf} \Delta z_C h \\
EE &= \Pi_f \Delta z_C h^* \\
QQ &= \dot{q}_{v,C} Vol + \Pi_{nf} \Delta z_C h T_{nf,C} + \sum Q_C \Delta z_C
\end{aligned}
\tag{3.34}
$$

and for fuel:

$$
\begin{aligned}
aa &= -k_w \frac{2}{\Delta z_C + \Delta z_W} \\
cc &= -k_e \frac{2}{\Delta z_C + \Delta z_E} \\
bb &= k_e \frac{2}{\Delta z_C + \Delta z_E} + k_w \frac{2}{\Delta z_C + \Delta z_W} \\
ee &= \frac{\Pi_f \Delta z_C}{A_f} h^* \\
QQ_f &= \dot{Q}_f \frac{\Delta z_C}{A_f}
\end{aligned}
\tag{3.35}
$$

The matrix is stored in a band format and solved with DGBVS, general linear solver for double precision banded matrix, from Lapack libraries [23]. This not only decreases the amount of storage needed, being useless to save a matrix mainly made of zeros if just the 7 main diagonals have elements inside, but also decreases the computation time of the linear system by around 2 orders of magnitude, compared to DGEVS, general solver for matrix systems.

For transients, as mentioned, we have a system for the coolant with DD, AA, BB, CC built in the same way plus the time discretization in a 4xNodes banded matrix solved with DGBVS, iteratively linked to the rods solved in FEM (3.27).

In the momentum equation, pressure drops at each volume elements will be found, to then evaluate the pressure profile as a function of the given inlet or outlet boundary condition. As mentioned before, the possibility is given to have a known pressure drop and unknown mass flow, the solved system, in this case, is the same, but an outer iteration

procedure will be required to correct the assumption on the mass flow rate per HA at iteration $j+1$, of the kind:

$$\begin{aligned} \Delta p_{boundary} - \Delta p_j = \Delta p' => \Delta \dot{m}'(f, L, \Delta p', Deq) \\ \dot{m}_{j+1} = \dot{m}_j + \varphi(\Delta \dot{m}') \end{aligned} \tag{3.36}$$

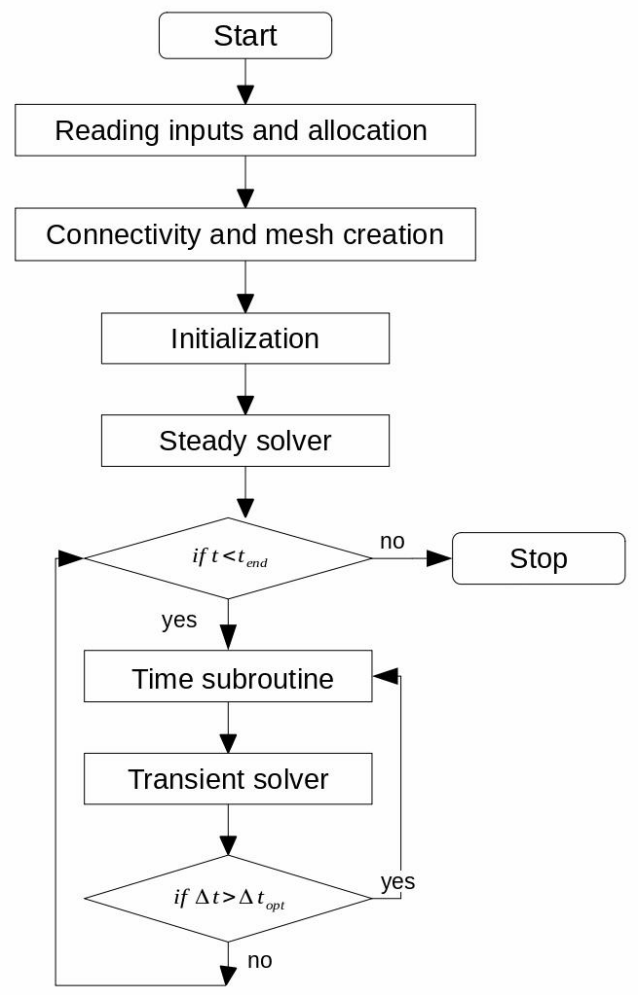


Figure 3.2: TH module logic diagram.

Chapter 4

Code design and implementation

4.1 Code functional modelling

In this section, an overview of the code functional modelling logic is given, from global functionalities to the new main subroutines implemented, by means of Data Flow Diagrams (DFDs) [24] and listing the procedures going towards the lower functions. The code design was performed top-down, starting with the higher-level DFDs and translating them into a sequence of macro-functions. From the macro-functions, the design proceded going into each functional unit and implementing the actual translation of the adopted mathematical modelling approach. According to the top-down design decomposition procedure, the algorithm is broken down into logical subdivisions called *subtasks*. Each subtask can be coded and compiled as an independent unit. They are written with the aim of maximizing modularity and potential future development of brand-new subtasks or expansion to be possibly implemented alongside the ones developed in the present work. Furthermore, a subtask can be tested separately to ensure that it performs properly in standalone mode before combining it into the larger program, reducing the total programming effort, simplifying debugging, and isolating the subtask from undesired side effects.

Of the main code functions in figure 3.2, some parts of the code are simply re-adapted from the ones implemented in the previous FRENETIC TH module, for the time being. Of course, the necessary modifications to comply with the new solver structure and to improve some non-standard or outdated programming practices were performed, such as non-initialization to default values of variables that are intended to be read from the input file, e.g. tolerances. Specifically, the re-adapted code parts are those associated to:

- read inputs regarding geometry and sources.
- generation of assembly map and indexing.
- compute geometrical quantities of interest: wetted perimeters, flow area, fuel cross-section areas.

The *time procedure* evaluate the temporal progression of the transient and the optimal time steps, in case of an adaptive time step procedure is employed. In this case, the value

will be found by keeping the truncation error inserted by the finite discretization under a certain value, determined as a function of a given maximum relative error and of the second-order derivative of the temperature. The approach is based on well know time-step procedures used for the neutron flux transient [25]. The proposed method estimates time step sizes based on the truncation error of the backward-differenced equations, predicting the successive time steps as a function of the known profiles. After the evaluation of the new solution, the time steps are again evaluated and if needed successive iterations at that time step can be performed.

The backward difference approximation is applied to the coolant temadoptedperature distribution and is:

$$\left.\frac{\partial T}{\partial t}\right|_n = \frac{T^{(n)} - T^{(n-1)}}{\Delta t_n} + \tau^{(n)} \tag{4.1}$$

where $\tau^{(n)}$ is the truncation error which can be written using a Taylor expansion as:

$$\tau^{(n)} = \frac{\Delta t_n}{2} \left.\frac{\partial^2 T}{\partial t^2}\right|_n + O(h_n^2) \tag{4.2}$$

in terms of the second order derivative and higher order negligible terms. From equation (4.3), an approximate time step size can be achieved knowing the second order derivative and prescribing an error tolerance to give in input to the code.

$$\Delta t^{(n)} = 2\tau_{toll} \left(\left.\frac{\partial^2 T}{\partial t^2}\right|_n \right)^{-1} \tag{4.3}$$

The second-order derivative is formulated according to:

$$\left.\frac{\partial^2 T}{\partial t^2}\right|_n = \frac{\Delta t^{(n-1)} T^n - (\Delta t^{(n)} + \Delta t^{(n-1)}) T^{n-1} + \Delta t^{(n)} T^{n-2}}{\Delta t^{(n)} (\Delta t^{(n)} + \Delta t^{(n-1)}) \Delta t^{(n-1)}} \tag{4.4}$$

(4.4) is computed punctually, at each volume element and the value held is the one leading to the minimum time step.

The value is compared to other relevant time steps from user inputs, calls from NE, end of the simulation, and optimal time step to keep under control errors given by explicit treatment of the inter-channel coupling. Also, a lower bound is given as an input for the possible next-time step, which can also be used to have fixed time steps.

In the transient and steady solver the governing equations are solved within each HA according to the logic reported in figure 4.1:

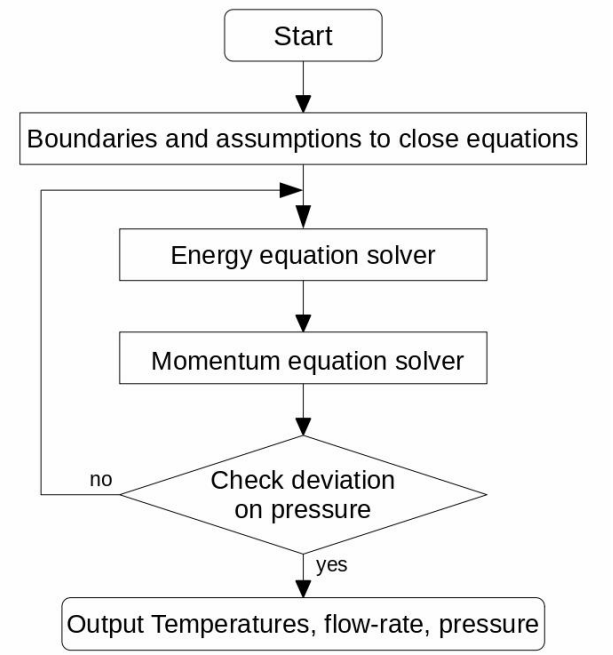


Figure 4.1: Logic diagram for each HAs.

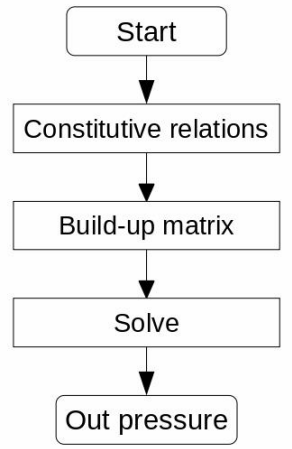


Figure 4.2: Logic diagram for Momentum solver subroutine.

Where a pressure drop boundary condition is provided, the initially assumed profile of pressure and related mass flow vector must be compared with the new obtained from the momentum equation solver and step tuned up to convergence.

The momentum subroutine logic to find the pressure profiles is shown in figure 4.2. The called energy subroutine is the same for time dependent and steady case, however the solved equations and called subtasks will be different. The logic diagram is described in figure 4.3:

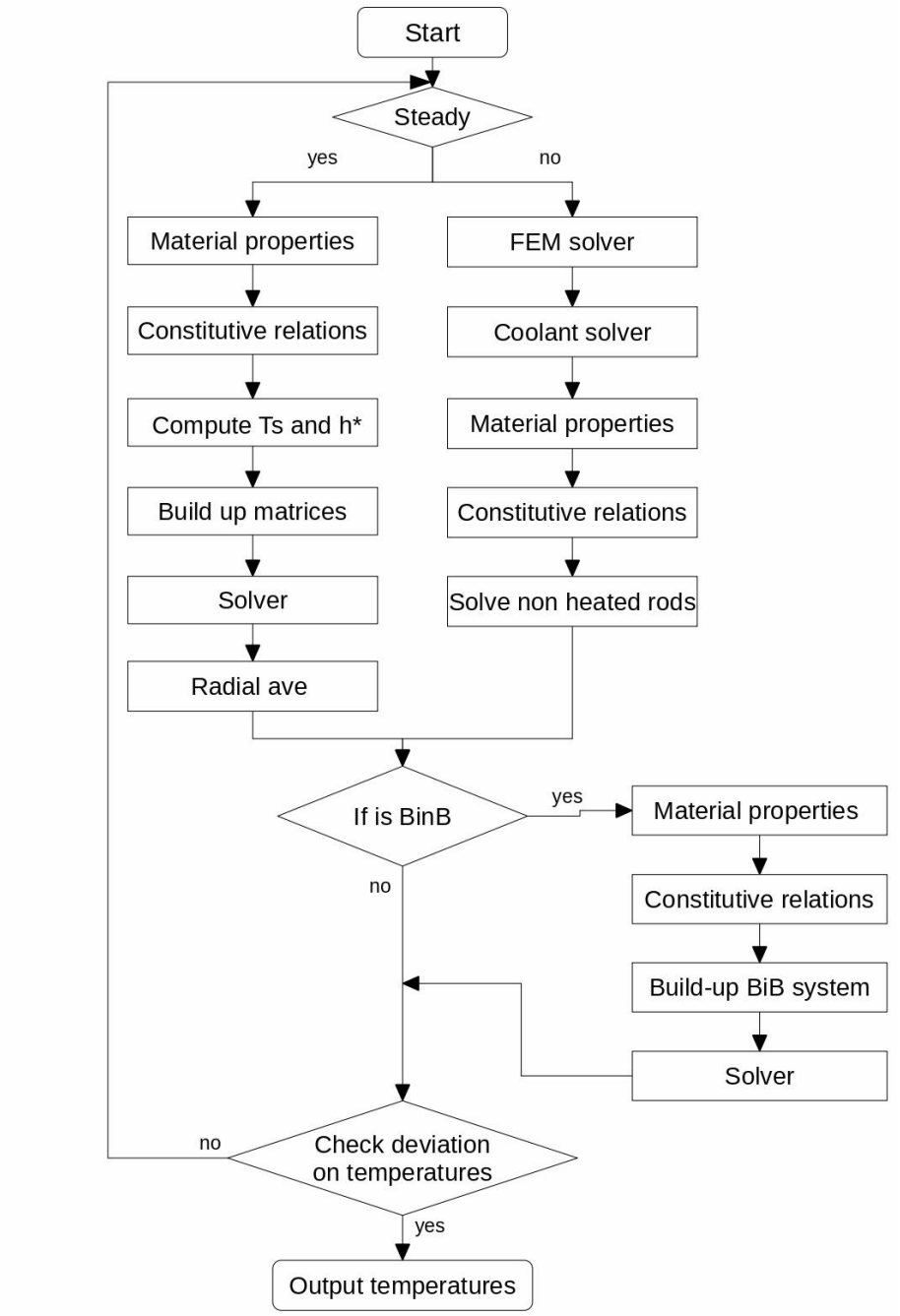


Figure 4.3: Logic diagram for Energy solver subroutine.

In table 4.1 are listed some of the main subroutines with functionalities and related necessary inputs and outputs.

Table 4.1 Main subroutines and required input-output quantities

Subroutine	**Inputs**	**Outputs**	**Functionalities**
Material properties	$\dot{m}, T^{j-1}, P_{wet}$	$\rho, c_p, k, \mu,$ Re, Pe	Call functions to compute materials properties and dimensionless parameters, as a function of temperatures profile at iteration j-1
Constitutive relations	$Re, Pe, D_{eq}, k_{wet},$ $Pitch, D_{rods}, D_{wire},$ $flag_f, flag_{Nu}$	f, h	Compute heat transfer coefficient and friction factor from constitutive correlation, as a function of user choice
Compute $T_{f,s}$ and h^*	$T_{cool}^{j-1}, T_{fuel}^{j-1}, h,$ $k_{fuel}, \dot{Q}_{fuel}, D_{rods},$ $D_{in}, R_{Cl,in},$ $R_{Cl,outer}$	$T_{f,s}, h^*$	Compute $T_{f,s}$ and h* according to the chosen average temperature approach
Build-up matrices	$T_{cool,i}, T_{fuel,i}, h*,$ $k_{fuel}, \dot{Q}_{fuel}, \Pi_f, A_f,$ $\Delta z, cp_f, \rho_{cool}, \rho_f,$ $cp_{cool}, k_{cool}, \dot{m}, A_{cool},$ $\Pi_{nf}, \dot{q}_v, T_{nf}, \rho_{cool,i},$ $Q_i, cp_{cool,i}, \rho_{f,i}, cp_{f,i},$ Δt	$\hat{A}$ and $\hat{b}$ of $\hat{A}\hat{T} = \hat{b}$ matrix system	Contain subroutines for creation of advection, coolant conduction, fuel conduction and source constants to be inserted in the matrix system
Solver	$\hat{A}, \hat{b}$	$\hat{T}$	Call external library to solve the system
Boundaries and assumptions to close equations	$\dot{m}_{in}, T_{in}, p_{in}, p_{out}$	$\dot{m}, p_{dummy}$	Impose boundary conditions
Build-up matrix for pressure	$\dot{m}, \dot{m}_i, A_{cl}, \rho_{cool},$ $D_{eq}, A_{ave}, \Delta k_{loc}$ $Re, D_{eq}, flag_f\ \Delta z$	p, f	Found pressure profile from momentum equation
Solve non heated rods	$\Delta z, T_{cool}, T_{nf,i}, \rho_{nf,i},$ $cp_{nf,i}, h, A_{nf}, D_{nf}$	$T_{nf,i+1}, \rho_{nf},$ cp_{nf}, k_{nf}	Solve non heated rods from axial average time dependent equation
FEM for heated rods	$T_{cool}, T_{f,i}, h, A_f, \dot{q}_v$	$T_{f,i+1}, T_{f,s}, T_{f,c}$	Solve the radial profile of heated rods

To solve the radial profile of fuel rods, at each axial position, the subroutine `step_h1` of the old code was reused with proper modifications. For the BiB modelling it is required to take into account a convective heat transfer resistance as described in 3.6. Since the whole structure of the solver block is changed, the implementation done in the past could not simply be re-used. For this reason, specific assumptions must be performed and a full energy model also for the thimble could be implemented with the logic of figure 4.3 as goal. `Boundaries and assumptions to close equations` impose boundaries for equations, as a function of inputs choice and, related to that, it imposes initial hypothesis for the required profile of temperature, mass flow or to be recomputed materials properties.

4.2 Input/Output description and formatting

4.2.1 Input files description

The code will read input data from formatted input files. There are three main types of files related to: common input (used by both the NE and the TH module), input specific for TH, and files for core assemblies' characteristics. The common input file contains a series of choices and data necessary for both neutronics and coolant modules.

- Type of coupling: active or not, and kind of coupling chosen between file r/w and TISC. The usage of the latter option is currently discontinued, but plans have been made to switch to a memory-based coupling exploiting open-source paradigms such as preCICE [26].
- Number of hexagonal channels; number of different groups of assemblies with equal characteristic and input file, read into the *data* folder; a core arrangement that can be defined by specifying, for each sextant, the number of HA in each row.
- Mesh definition with a number of spatial elements and mesh refinement choice.
- Simulation time end and time step tuning, selection of time adaptivity flag.

Coolant input file contains:

- Numerics: values for definition of tolerance and under-relaxation factors.
- Heating options in space and time, with choice between power deposition from external files, constant values and NE coupling.
- Boundary condition reading options.
- Define how to treat BiB assemblies.
- Definition of output times and spatial locations.

A specific data file for each type of core assembly is also provided. This file contains the information regarding: which HA, according to core mapping, belongs to this specific type; the bundle geometry; the number of heated and unheated rods; selection of adopted correlations.

The coolant mass flow rate in the HA and the fraction going in the thimble region is specified in input. The core inlet temperature andemail pressure are provided as well giving values for each assembly and for each time step in which they change.

4.2.2 Output description

The user will have the possibility to choose at which time step and for what channels print in output the relevant parameters, with vectors formatted in the form: spatial axial position, coolant temperatures, pressure, coolant density, coolant velocity, average rods temperature, central temperature, superficial temperatures, linear power deposition, heat transfer coefficients.

When the file-based coupling approach between the neutronics and the coolant module is selected, the code should give in output coolant and rods average temperatures and iteration time step.

A series of other reports will be automatically produced for code diagnostic giving in output for each channel and for each simulation time, first and last axial values of

quantities of interest and related maximum values for temperatures, velocities and other parameters.

4.3 Programming guidelines

4.3.1 Coding strategy

The code was written in free-form Fortran90 [27] with standard coding procedures for variables declaration, and usage of subroutines aimed at code modularity. The `implicit none` statement was used throughout the code so that all actual variables have to be explicitly defined with `intent()` attribute, for readability and to improve the compiler capabilities to check for unintentional errors. Comments was improved to clarify functions and the meaning of variables. Furthermore, special characters are included in comments so that a Doxygen documentation can be automatically generated. The output take advantage of HDF5 Fortran library [28] to reduce the disk usage and the total number of files produced, as users can already opt for, whereas for the input the standard text file approach is retained, in view of the limited size of the input itself.

The open-source gfortran compiler will be able to compile and run the code, in combination of OpenMP [29] for the possible parallelization of the solver's blocks. However, the strict adherence to standard Fortran 90 programming practices will allow other compilers, if available, to be used (e.g. the Intel compiler suite), which could possibly lead to performance improvements. The code compilation options is specified in a Makefile produced by means of the open-source CMake utility [30].

`Do while` loops for convergent quantities are maintained under control checking that the number of iterations do not exceed the maximum value chosen in input.

4.3.2 Adopted nomenclature

To increase the code readability, standards for the choice of names in the code are used. The coding procedure follow the nomenclature choice done in TIFONE [15]. Subroutine and function names have have their first character capitalized, and dummy arguments have spaces around. Spaces are also employed around `if` statement parentheses. Array names are instead lowercase, and there are no spaces between the arguments and the round brackets. Variables with the `parameter` attribute have uppercase names. Functions from third-party libraries are also named in uppercase. Variable definitions are always preceded by double colon.

4.4 Version control

It is worth mentioning that the new code version was developed starting from a branch of the FRENETIC project, which is under version control via the well-known *git* framework.

4.5 Detailed code design

4.5.1 Data management strategy

While writing the new code version, attention was devoted to modularity as well as to the possibility of future code extensions, following the requirements and guidelines and adopting a top-down approach to design. Even though some parts of the previous code were reused, as previously explained, they were entirely updated in their form and in terms of data access because of the new data structure explained in 4.5.3. Procedures that were unused or redundant were cleaned up or removed, and some procedures were divided into smaller pieces into different files for better code management. The calls to the various created subtasks from each calling subroutine were written by using an explicit interface. This means that every array or structure needed as input and output into procedures is explicitly passed to the calling subroutine. This allows the single routines to be tested independently, without shortcutting the data passage via modules, i.e. adopting *data encapsulation*. The only exception to this rule is represented by common values, like tolerances and time steps, which are taken from modules inside the subroutine itself. For what concerns all the other TH parameters and geometrical characteristics of the specific HAs, they are read from input files where they are written in the form of an explicit-shape dummy array, allocated only during subroutine runtime, and characterized by bounds explicitly specified. In this way, the subroutine will know the shape of each dummy array when it is executed, thus simplifying the compiler checks in terms of vectors dimensions during debug operations, thanks to appropriate flag, being able to detect and report out-of-bounds memory references. This is fundamental during the usage of dynamic allocations of vectors and data structure to avoid memory leaks. The usage of an explicit interface to called subroutines was also done according to the new data management and solver parallel design, so that only the top routines have access to all data sets, while, calling the solver, each channel has access to only its required data. Again, this represents an application of the data encapsulation technique.

4.5.2 Implementation of dynamic allocation

To satisfy goal G6, all the data structures are now organized into user-defined data types, declared in modules and allocatable structures, with pointers to facilitate the dynamic allocation and the usage itself of the data sets contained in each data type. The new management is more efficient from the point of view of the amount of memory usage and during usage operations itself. For example, in exchanging data during time execution, it is way more efficient the swap of one pointer addresses instead of the whole node per node matrix. To allow for the above-mentioned improvements, additional quantities are now required in input, namely:

- `Nchan`: is used to allocate the number of different channels;
- `MaxAxNode`: is the maximum number of nodes in one channel from the mesh composition in `commoninput.dat`;
- `MaxRadNode`: is the maximum number of radial nodes for FEM solver, it is given in input inside `commoninput.dat`, the various HAs then can have their characteristic

radial distribution according to the new data input file. However, MaxRadNode should be strictly larger than the chosen radial element's composition;

- `Maxbcfile`: is the maximum number of read lines inside the input boundary conditions files, it is given in input inside `commoninput.dat`. To avoid arbitrary static allocations of vectors, also the structure of the input vectors was changed introducing to them a dynamic allocation and reading the files just one time, the first when procedures are called, as before.

The dynamic allocation of the vectors is done by ad-hoc subroutines locate'_', where the additional keyword will be the name ScTh, BIB, BC, etc. as a function of the following datasets, giving in input Nchan, MaxAxnode, MaxRadNode and the actions, so, 1 for the allocations and 0 for the deallocations of vectors. Variables names were chosen, maintaining some of the old nomenclatures for the readopted variables and choosing more consistent and understandable names for the new ones. All the data sets are also commented inside the modules' data type definition, describing their meaning and usage, dimensions and expected vectors dimension, with a structure compliant to be read for the Doxygen documentation [31]. An example is reported for the array `tempCl` which contains the solution of coolant temperature profiles for the entire core:

```
1 real(dp), pointer, dimension(:, :) :: tempCl !< (nAxNod, Nchan)
2           !! Average temperature of coolant for each HA at each axial
    node
```

This will results in a series cards inside the Member Data Documentation for the `scTh_t` structure, as shown in figure 4.4, with the reported explanation and hyperlink to code file line.

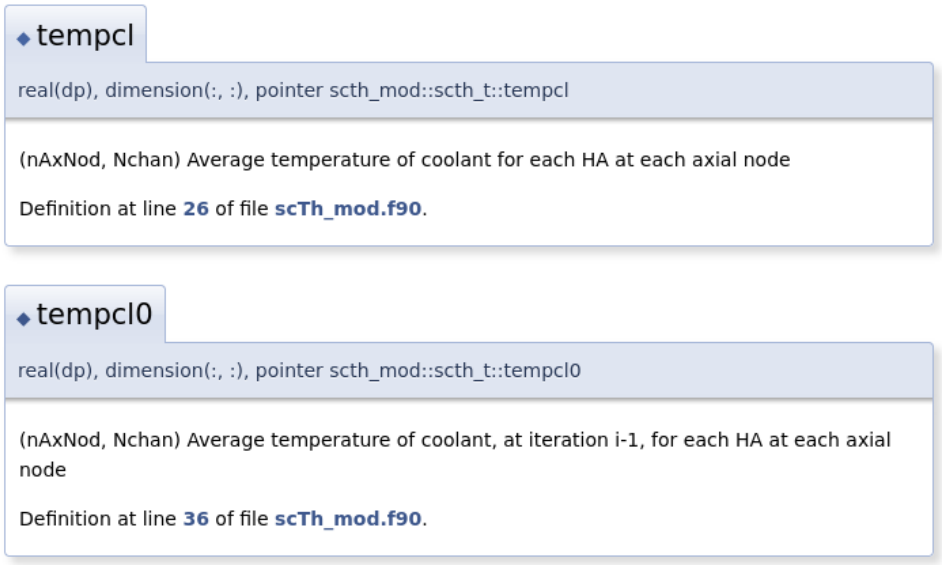


Figure 4.4: Variables `tempCl` and `tempCl0` documentation from the Doxygen HTML report.

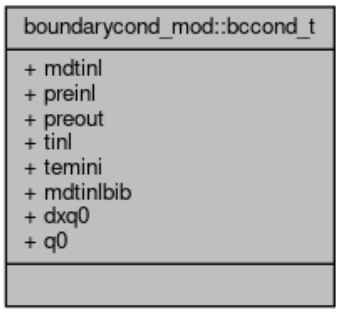
To properly use the dynamically allocated vectors, it was done a proper use of: the

STAT= clause in any ALLOCATE statement to check the returned status, so that a program can be shut down gracefully if there is insufficient memory to allocate the necessary arrays or there were some problems due to vectors already allocated with the same name; the usage of ALLOCATED function to see if the vector was already allocated.

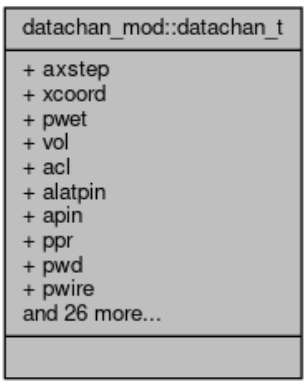
4.5.3 Data structures

As explained, all the vectors and data are assorted into modules here explained:

- The module `casestudy` contains relevant case study parameters like number of channels, kind of run (steady state or transient), coupling options, core geometry model, number of maximum nodes for vectors, kind of boundary conditions, type of coolant material and presence of BiB and non heated rods.
- The module `const` contains a series of constants and parameters common in the code.
- The module `numerics` contains a series of numerical data sets like: tolerances; under relaxations factors; maximum number of iterations; some flags for reading from external files of kloc, qvol etc.; time steps; reference time; internal flags and others parameters used in the code.
- The dynamically allocated vectors' data are all grouped inside the module `typespoint` which contains the pointers to derived data types of pointers containing the quantity of interest:
  1. 'bc', of `BCcond` type (see Figure 4.5a), contains data for boundary conditions and input heating for the TH module standalone run.



(a) Boundary conditions structure

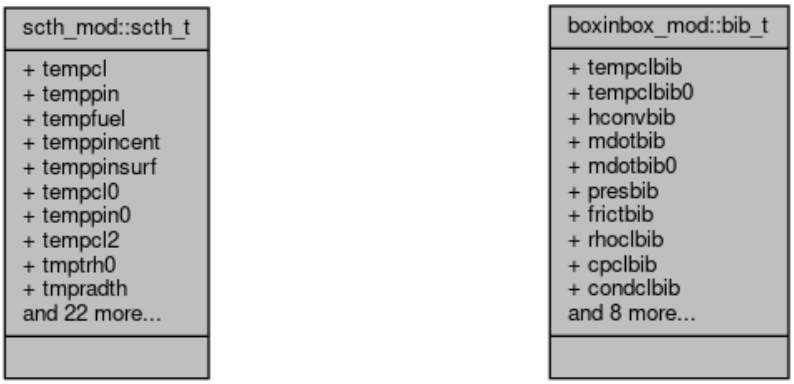


(b) Channel data structure

Figure 4.5: 'bc' and 'dt' data type from Doxygen documentation.

  2. 'dt', of `datachan_t` type (see Figure 4.5b), contains all vectors for what concern main channel geometries, areas, relevant length, fuel dimensions, clearance and box thickness, and radial elements characteristics dimensions, so thickness coordinates of the nodes, lateral perimeter.
  3. 'th', of `scTh_t` type (see Figure 4.6a), contains thermal-hydraulics relevant parameters like temperatures, heat transfer coefficient, flow rate, pressure profile,

heat transfer coupling, heat deposition vectors and material properties of main channels and fuel rods.



(a) Thermal-hydraulics data structure (b) Box-in-the-Box data structure

Figure 4.6: 'th' and 'bb' data type from Doxygen documentation.

4. 'bb', of `BIB_t` type (see Figure 4.6b), contains TH parameters and relevant geometry data for BiB assemblies.
5. 'nf', of `NFth_t` type (see Figure 4.7a), contains TH parameters for non-heated not fuel rods.

(a) Non fuel rods data structure (b) Tisc data structure

Figure 4.7: 'nf' and 'ti' data type from Doxygen documentation.

6. 'ti', of `tisc_t` type (see Figure 4.7b), contains vectors in input and output from TISC subroutines data exchange, possibly amplified in future works.

- ichoice: custom data type (see Figure 4.8). It contains channel choices and flags for runtime such as: materials choices, kind of friction factor, kind of heat transfer correlation to use, number of heated and not heated rods. Channel modelling choices so are grouped, and the specific assembly information can be given in input to subroutines transferring only one term.

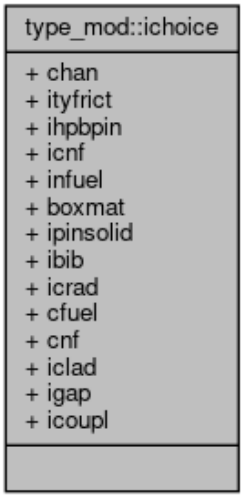


Figure 4.8: Channel modelling 'ichoice' data type.

From the old `module.f90` file, some structures were retained, in particular the module `tisc`, module `sensors` and module `GEOM`, that were slightly changed in order to avoid problems to unchanged subroutines but at the same time adopting a dynamic allocation of vectors also here, as a function of the number of channels actually needed.

4.5.4 Main program refactoring

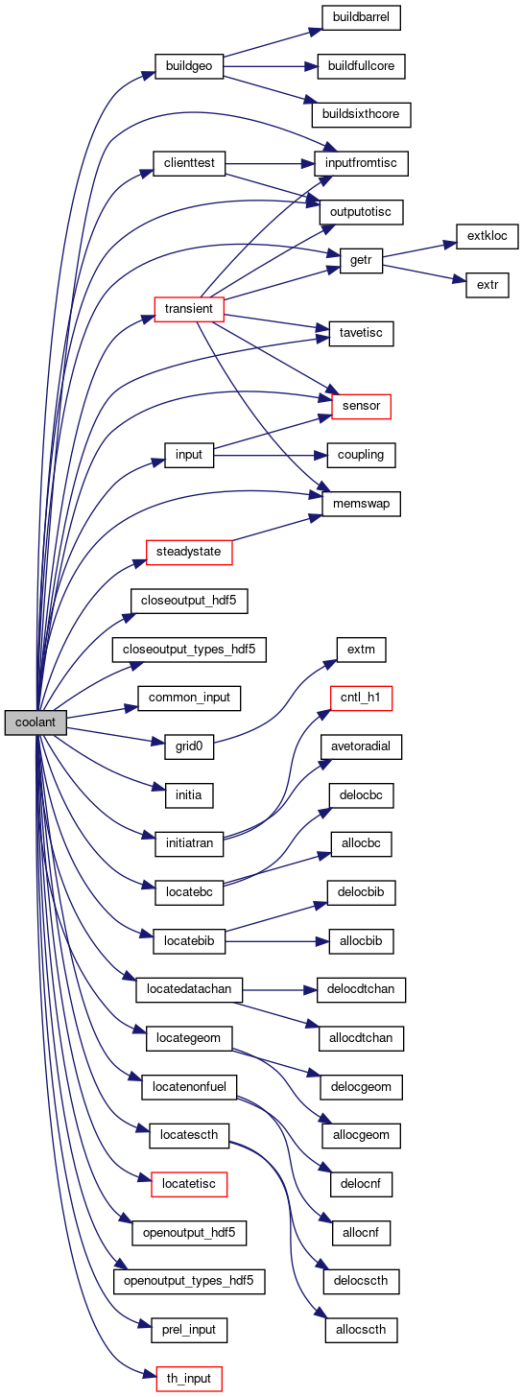


Figure 4.9: Main coolant call graph

The main coolant program now just does the calls to the subroutine for initialization of hdf5 modules, reading from inputs, allocation of variables, creation of channel map, calculations of the relevant geometry, initialization of vectors, calls to steady or transient solver with associated interchange of data file with NE. The call graph from Doxygen documentation is in figure 4.9.

4.5.5 Subroutines re-adapted from previous code implementation

Subroutines for material properties

For the properties of the materials as a function of temperatures, as already said, the old structure was maintained, cleaning up the code and introducing a keyword-based nomenclature for selection between them, through `select case` format structure to simplify the possible introduction of new functions and materials. Some dependencies were removed, like the pressure dependence for properties, since the ones used are all just temperature dependent, being only related to liquid metals and solids. The internal check to ensure that temperatures are greater than zero and in the working reference range, namely `IERRSTOPCOOL` and `IERRSTOPSOLID`, were moved out, so that, computing more properties in the same step, the check is done just once, explicitly calling the subroutines. A new structure was also given to `GETQ`, adopting the new data structure and using a dynamic allocation for `EXTQ`. The same was done for `GETR`, with boundary conditions read as a function of case study choices. Here, it was also added the reading from file of coordinates and values of local K_{loc} pressure losses due to geometrical changes or other local drops.

Calculation of radial pin temperature profiles

The old `heater.f90` for radial profiles determination was rewritten, reducing size, removing unused stuff and intermediate calls, adapting subroutines' calls to new data structure and removing the internal data allocation.

Coupling to NE module

The module coupling procedures with NE were reviewed avoiding to send twice the files during the steady state initialization. In particular, attention was devoted to ensure that the coupling files are exchanged at the correct time step, to solve previous inconsistency that was caused by the constant presence of files in the working folder and the imperfect modules synchronization, causing for example the freezing of the program at end simulation time and, due to that, the loss of simulation data outputs, if the hdf5 file option was turned on.

While working on updating the NE-TH interface, it was found an error in comparing coordinates during the interpolation of coolant and fuel temperatures from TH mesh to NE. The new obtained profiles inside the NE `thtone` subroutine, given an example in figure 4.10, now better represents what are the TH outcomes. However, the profiles of the fuel are still slightly underpredicted in the peak zone, so future developments are required to review the interpolation process and reach the correct target. Now it is already improved

and, during transient, the behaviour of the core could overcome the unexpected results obtained during some validation procedures in the past [14].

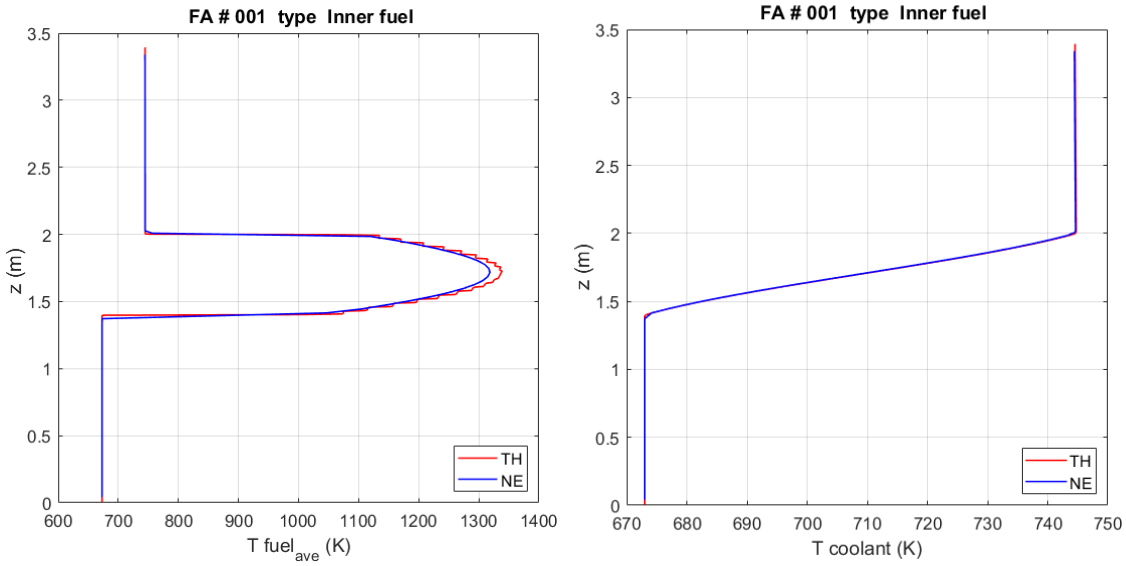


Figure 4.10: Comparison in coolant and fuel average temperatures.

Subroutines for reading input data

The same structure of the previous code was maintained for the input files. In common input the `ADDTH` namelist was added to insert the maximum number of radial elements and dimensions of vectors for reading input files. Instead, the `input.dat` file for coolant was thoroughly changed, removing the unused values and adding all the new numerics and flags variables. The reading of boundary conditions was revisited, inserting the `masbound`, `enebound` and `mombound` closure types into equations, for the different case studies. The usage of values is done maintaining the same logic, reading the external input files or using the values inside `input.dat` as a function of the *initia* variable, that however at the moment is a flag for the entire core while before was for each channel.

Calculation of heat deposition

In the coupling options, the unused downcomer option and thimble modelling options were removed, since this last is now just in one way. Here are inserted instead the reading from file of `fluxext`, for debug, and volumetric heat generations options, for coolant equations in main HA and BiB, to possibly account for the presence of an off-rods heat deposition due to gamma rays. The values are real, so that now can be considered as a percentage of the pin heat deposition and in the future, if implemented a specific function in the NE module, giving a negative value can be used as a flag to take the values from NE coupling files.

Code output

The Sensors routine, which is responsible for the output part, was instead not modified, as the modification of the code output was outside the scope of this thesis. As a matter of fact, the code output was recently updated to print into hdf5 format.

4.5.6 Error handling

All the variables related to the new structure of the code are now pre-initialized and, after reading, it is checked if they respect the reference expected range, to try to guide the user to not make input errors or forget important parameters that could cause the crash or the indefinite run of the code, or worse the production of nonphysical results, in case the default value given by the FORTRAN compiler for example to tolerances was found. An example is reported:

```
if ((TOLTEMP.le.0.0_dp)) then
    write(*,'(a)') 'ERROR. Incorrect TOLTEMP. Allowed values: .GT. 0.0.'
    write(*,'(a)') 'ERROR :: NML=NUMERICS. :: FILE='',INUNIT
    write(*,'(a)') 'Default value: 1.0d-6'
    stop
end if
```

The HA-type file data were also changed, introducing some more rods parameters: actual fuel size, cladding radii, kind of materials, and approach flag. The optional radial file `heater_input.dat` is now here integrated, since during transient the radial method is always employed. However, some parameters are removed, since not needed, and the radial discretization in elements is internally computed as a function of the given number of nodes and radial thickness. In this way, is also possible a different radial discretization for different HAs. The computation of this radial nodalization, wetted perimeters and areas is now performed outside the solver subroutine, in `initia`. In this way also the geometrical parameters for the FEM approach are just computed once.

The rods gas gap thickness, diameters of the pins and cladding, IW gap thickness and box thickness are assumed to be equal among the axial direction, with the possibility to have different values in the 6 hexagonal side for the half thickness of the coolant clearance between assembly boxes.

4.5.7 New subroutines

According to the given guidelines, the new solver was created by following the top-down approach and logic structure in section 4.1. From this a series of subroutines was created:

- `Advection`: compute a 3xnode matrix according to the second order upwind scheme. An initial interpolation to integrate heat capacity central values to cell boundary is performed, using a central scheme, weighted on the actual step length.
- `Avetoradial`: compute the first profile of radial temperature at nodes, to switch from steady to the transient solver. It is found from the theoretical profile as a function of qpindot, coolant temperature, presence of cladding and gap, and kind of fuel pellet, cylindrical or annular.
- `Boundandclose`: set boundary conditions as a function of the case study's choices and found coolant properties in case of frozen coefficient flag.
- `BuildMat`: construction of steady 11xnodes matrix system through proper insertion of conduction, advection and sources terms in the matrix system.
- `Channel`: sequential calls of subroutines to solve energy and momentum equations in each channel.
- `ClntEneSolve`: coolant energy solver used for BiB, not loaded assemblies and main coolant assemblies during transient. It builds a 4xNodes matrix and it is solved.
- `Conduction`: compute a 3xnode matrix according to a central difference conduction formulation, adiabatic boundary conditions are directed imposed. An initial interpolation to integrate heat diffusion central values to cell boundary is performed, using a pseudo resistance scheme, weighted on the actual step length, to preserve energy conservation.
- `Costitutive`: subroutine computes friction factors and heat transfer values, as a function of channel's choices.
- `Emptychannel`: channel solver to compute coolant and pin temperatures outside the central core region, i.e. to account for the additional channels to model the barrel and external lead regions required in the NE module. Using the inlet mass flow rate as a flag, in case is lower than zero a constant temperature equal to the inlet value is imposed, whereas in case is bigger or equal zero, it computes the `ClntEneSolve` and momentum equation. In case the mass flow is zero, the heat transfer is set to 1.0d0 and pressure constant as inlet.
- `EnergySolver`: call to different energy solver's subroutines.
- `Exchange`: re-adaptation of the old `exchange` subroutine. For each channel, search the near ones and solve the equation (3.17) as a function of choices, or adiabatic boundary.
- `InitiaTran`: Initialization of variables used in transient runtime, trough `avetoradial` and calls of radial FEM energy solver of the pin, up to converge to `tolTemp`.
- `Materialprop`: calls for materials properties' computation for coolant and fuel.
- `Momentumsolver`: determine the pressure profile as a function of mass flow rate and friction factors, taking advantage of coolant incompressibility.
- `Radialave`: 'TORAD' computes rods' surface temperature as a function of theoretical profile. 'TOAVE' computes average rod temperature, to be sent to `TISC`.
- `SolveBiB`: BiB coolant's energy equation solver and function to compute the thimble

heat transfer to the main channel.

- `SolveNoNheated`: Non-heated pins' energy equation solver, considering no heat deposition and flat temperature profile, to account for non-fuel rods temperature inertia through an average 1D temperature profile evolution.
- `SteadyEneSolver`: Steady state energy system solver for coupled coolant and fuel system.
- `Steadystate`: Steady state solver, it gets boundary conditions and do parallels calls to `channel` subroutine for TH evaluations, compute error check and inter-coupling thermal exchange up to convergence.
- `Timestep`: Time step assigned as a function of case study choices and actual time point.
- `Transient`: Transient solver, it gets boundary conditions and parallels call to `channel` subroutine for TH evaluations, compute time derivative check, compute inter-coupling thermal exchange, calls for possible output and sent of files to `TISC`.
- `AdaptiveTimeScale`: subroutine to find within each channel a more suitable value of the time step. It predicts the time step for the pseudo transient based on characteristic length scale, velocity field and heat diffusivity.
- `MemSwap`: Simple pointer address swapping to change time-dependent vectors.
- `SecTimeStep`: compute a local second order derivative of coolant temperatures to find an optimal delta time step for the next iteration based on (4.2).

The call graph, from Doxygen documentation, of the transient solver is reported in figure 4.5.7, since it is the most complete in the usage of the newly introduced subroutines, to see the relationship between the different tasks and calling procedures. As already seen in the call graph for the `coolant` main program, the structured approach to code design is reflected by a layer-wise ordered structure of the subroutine calls. The only exception is represented by commonly used modules such as the one for material properties, which are accessed by multiple routines possibly belonging to different layers. From the main graph were removed the calling functions for materials properties for the sake of clarify and make more readable the graph.

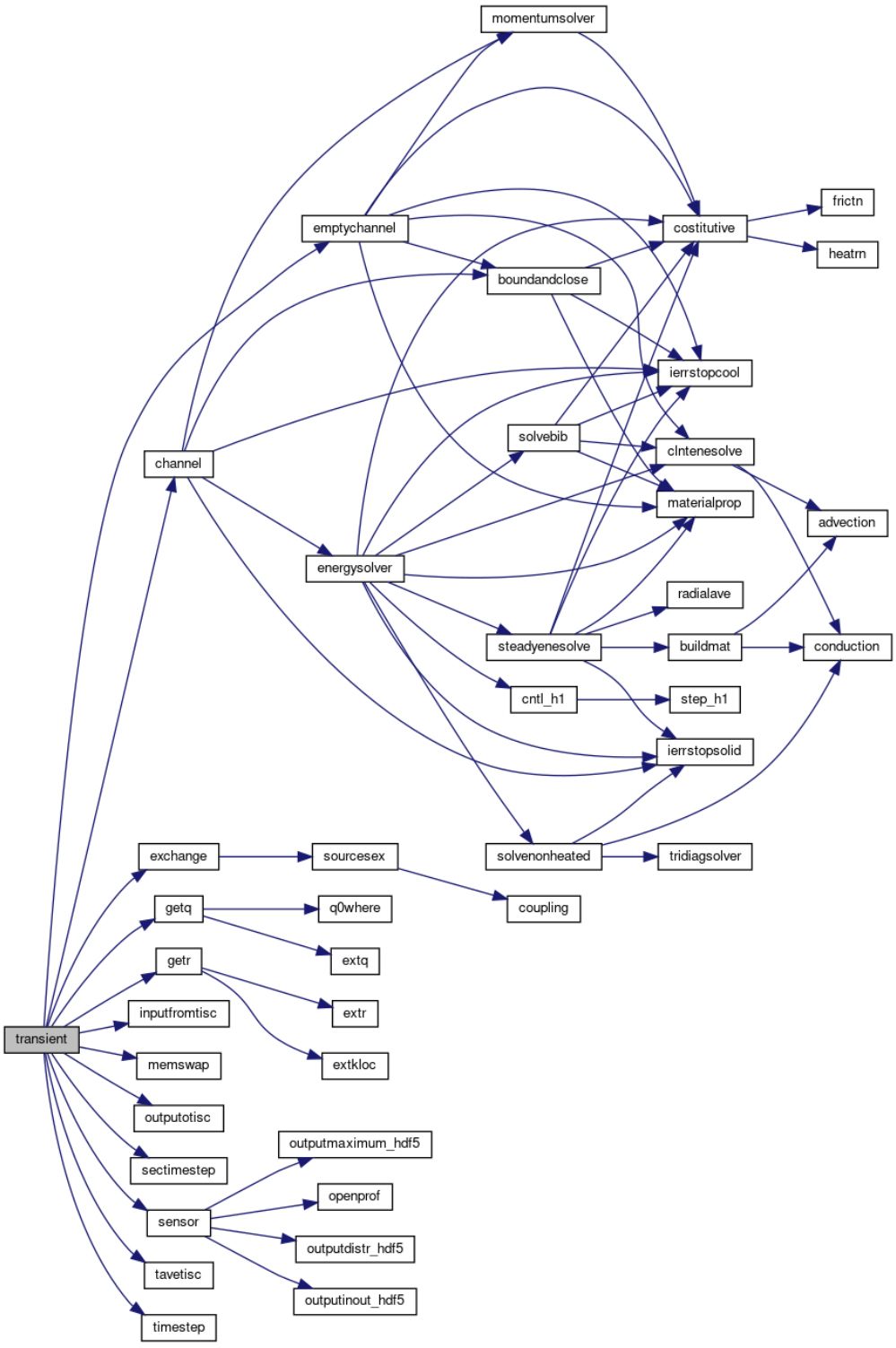


Figure 4.11: Transient calls graph

4.6 Further optimizations of the code design

4.6.1 Numerical schemes for improving the solution efficiency

During the realization of the solver, changes were made to the initial concept to improve performance and validity range of the approach.

To avoid over-solving during iterations procedures it is introduced an an adaptive usage of tolerance reference values. Due to the initial assumption of no channel's flux inter-coupling, the initial steps during progression into steady-state conditions are for sure affected by errors, thus inexact methods can be adopted to cut down the computational time. The number of iterations was reduced by a factor two by the adoption of a Residual Balance method [32], to eliminate over-solving without degrading the convergence rate during steady-state initialization. The iterations, inside the energy solver subroutines are carried out keeping as tolerance the residual norm of the constituent core solver, as in equation (4.5), only just below the residual norm of the maximum error found in the other channels, at the previous core iteration step. In this way, each channel's solver will progress not too far beyond the actual point of the others, after the re-evaluation of the new coupling heat flux between channels is performed, bringing all the core's HAs to convergence together.

$$||R_i^{chan}|| < a \cdot max(||R_{i-1}^1||,\ ...,\ ||R_{i-1}^{NChan}||) \tag{4.5}$$

With constant a specified by the user as $a = 0.1$, and initial reference value for the residual norm of 0.1. Also, since in input is taken one unique `underTemp`, which is used solving all the energy equations, an additional under-relaxation factor was added to the steady solver. The required under-relaxations are in fact different in the two solvers and it is useless to over-relax the transient equations without justification. Thus, a multiplicative parameter of 0.1 was introduced in `SteadyEneSolve`, thus leading to the possibility to accept in input `underTemp` for example of 0.8, keeping the same stability and improve transients' execution time. This will significantly increase performance during the iterations in transient regime. The main solver, in fact, could require more under relaxation only in case there is also the necessity to couple BiB channels.

During preliminary calculation it was seen that a possible source of initial inconsistency could be driven by the temperatures profiles resulting from the theoretical mean average approach of the steady solver, which is different from the average one derived from integration of the spatial radial resolution of fuel temperatures. Differences that were measured ranged from just a small fraction of degree to more than 1 K, in the case of a pin with different layers of gas and cladding and huge power deposition. Even if their presence is accounted for by adding an additional thermal resistance to find the surface temperature, thermal dependence of materials' properties could lead to some differences. To avoid an initial perturbation of the system during the first time steps of the transient, slightly worsening the linearity of the code some changes were did in the solver strategy. Thus, the steady solver is performed in cases in which the ending time of simulation is 0.0d0. If the case study is a transient, the steady solver is used to find the initial spatial profiles of temperatures, after the first send to NE, the initials flag is not reset, so, even if the calling subroutine is the steady-state, the equations solved are the coupled time-dependent

equations with the radial FEM approach. The same is done in the solo run when it is planned the usage of the transient solver. In this way, the successive data exchange, that are necessary for NE stabilization of temperatures and cross sections, are used to also smooth that possible initial inconsistency in the profiles and avoid any possible nonphysical thermal feedbacks at time 0s. Thus, it is performed a pseudo transient in which the time step used during is `deltasteady` given in input, as the previous implementation. The core is solved always keeping the steady tolerance's reference value, so more cycles can be performed in the core solver, sending the profiles always with the same accuracy.

The steady condition is achieved by imposing a relative error of coolant and fuel temperatures under the reference value `tollToSteady`, given in input.

4.6.2 Parallelization

As a consequence of the adopted solution strategy, due to the channels inter-coupling done in an explicit way, a possible parallelization comes out easily to be implemented, having the possibility to assign the resolution of a channel to a different thread. To take advantage of extisting frameworks, the open-source OpenMP libraries were included in the code, through the insertion of a compiler's flag and adequate access to libraries. The adopted procedures were implemented by measuring the effectiveness of the parallelized work, in order to use them only when there is an actual improvement. In particular, the subdivision of the work between different threads is an operation that requires a nonzero computational time by itself, inserting delay in the program. So, for some operations like the simple copy of values between vectors, it was not implemented since they are already extremely fast, while it is preferable to use the parallelization for those operations that require some relevant computational time.

Memory management globally is shared, with some private clause for specific cases, for example, to avoid errors during sums. The coding choice, of giving an explicit interface to subroutines, again leads to simplifications in the implementation. Since all the required information are given during the call of the `channel` solver, there are no possible errors coming from bad indexing or assignment in the parallel execution.

The subdivisions of the work among threads are done in specific piece of the code, by combined parallel work-sharing constructs like:

```
1 !$omp parallel do schedule(dynamic,1)
2     . . .
3 !$omp end parallel do
```

with which the parallel region and kind of schedule are defined. The schedule implemented is dynamic for the solving subroutine of TH channels, since in this case different assemblies, with different heat generations, number of rods, empty channels and presence of BiB, can lead to very different computational time. For other works like the assignment of boundary conditions and inter-channel coupling, a faster static schedule is implemented. In this case, the time required for each channel is practically the same and is useless to use a dynamic schedule that inserts an additional latency due to the listening of the threads for the assignment of the new work, worsening the final times required.

For some blocks like the call to reading the heat source input file and assignment of heat deposition vectors, some more clauses must be added:

```
1 !$omp parallel do schedule(static,1) if(iii>1) private(iii) ordered
```

in this case, the iii variable is set as private to each thread and it is specified that the do loop is solved following the order from 1 to `Nchan`. Furthermore, the parallel region is activated when iii is bigger than 1. That sophistication was done in order to not have problems during the access to regions with allocatable vectors that have to first be allocated the first time.

Thanks to the way OpenMP library were used, the parallelized regions are active only when the `-fopenmp` flag is active in CMake file. The usage of OpenMP, also requires the assignment of the number of threads to be used, that unfortunately cannot be done by reading from file but must be chosen a priori giving a value to `num_threads_TH` inside

`casestudy` module. Then there is also the possibility to change it by setting an environment variable in the terminal where the coolant program is running.

To see the improvement given by the parallelization of the tasks, a case study in a 37 assemblies core is reported 4.12, counting the computational time required to reach a steady state during a run coupled with the NE module. The delay time given by the read-and-write approach is not counted.

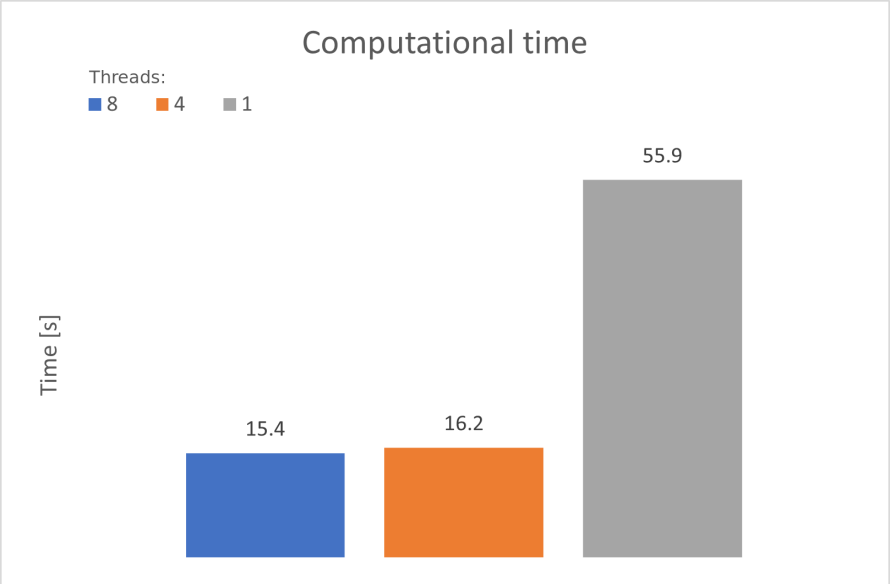


Figure 4.12: Time execution steady-state case.

Moving from 1 to 4 threads it is achieved a division by 4 of the required execution time. Moving further in the adoption of 8 threads the resulting computational time is just a few decreased. Maybe in this case the latency introduced by the subdivision of the work is of the same order of magnitude as the time execution itself, so a more realistic core with more assembly could give better results doubling the threads. It has to be mentioned that the cases were produced with a CPU with 4 cores and 8 threads, so a latency could be inserted due to the fact that the additional threads are not physical.

4.6.3 Inclusion of localized pressure drops

As implemented in the momentum equation (3.5) the code now can take into account localized pressure drops in the HAs. To do that, the `klocx` flag in the input.dat has to be turned on, giving a value bigger than zero. The flag is used to search, in the same folder, a file kloc.dat that contains the wanted local friction factor losses according to:

$$\Delta p_{loc} = k_{loc} \frac{\rho v^2}{2} \tag{4.6}$$

The values are usually computed by experimental evaluation of pressure jumps, as a function of the obstacle interference area. The file must be formatted inserting:

- the number of rows to read
- coordinate, kloc value per each channel at that coordinate

as explained in the subroutine headers note:

```
! # USER'S ROUTINE FOR EXTERNAL K local constant for geometric pressure
    drop
! # ncoord_kloc
! # xcoord | 0 0 0 0 K5 K6 0 K8 0 ... K_Nchan |
```

Following in figure 4.13 is reported an example of evaluated pressure drop considering for example the presence of two grids at coordinate 0.2 m and 3.2 m.

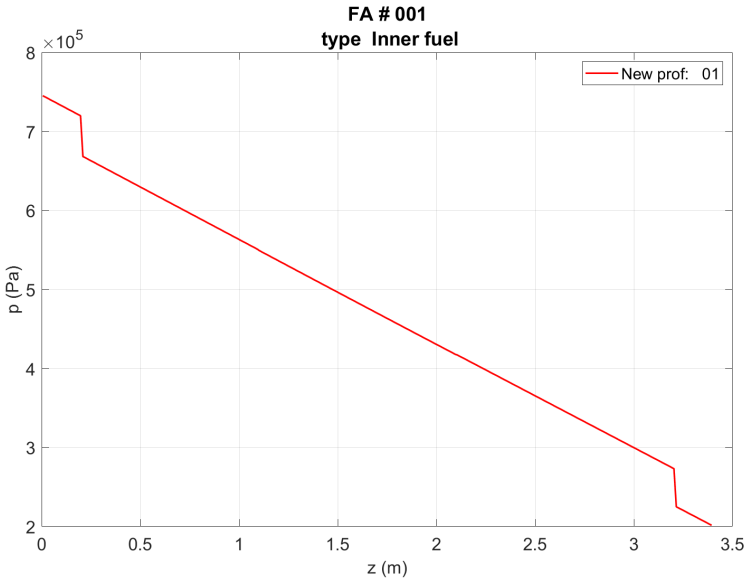


Figure 4.13: Example of computed axial pressure profile in the presence of two grids inducing localized pressure losses.

4.6.4 Transient solver for unheated rods

To enhance code capabilities in better estimating thermal feedbacks, in case of mixed composition assemblies which include unheated rods, a time dependent energy equation was included to solve also for these rods, based on an average 1D approach, to account for the presence of the additional thermal inertia. An example of a potential effect is reported in 5.1.9

Chapter 5

Code testing and benchmark

The code testing phase starts verifying that the new implementation satisfies the software requirements, is capable of correctly processing the inputs and is physically sound at the basic level (e.g. the solution respects the imposed boundary conditions).

5.1 Test cases

After these initial tests, the code functionalities are checked by means of a series of benchmark against the previous code, in a simple single assembly and a mini core of 37 HAs, in different regimes and with various power deposition profiles. The coolant is lead for all the tested cases and the materials properties used are the same as implemented in the old code. Assemblies have all the same in dimension, 3.4 m of height and lateral side of 0.0912 m, and same rods composition: 127 UO_2 rods with diameters of 0.0105 m. Mesh distribution was created similarly, not identical due to a different kind of discretization, with 401 nodes and refined zone from 1.1 m to 2.1 m. Heat transfer coefficients were evaluated with the Kazimi correlation and general bundle friction factor correlation was chosen for the evaluation of pressure drop. Also the new implementation was tested using the pseudo transient for the initialization of profiles and `TOLLTOSTEADY` equal to 1.0E-9. Some of the most relevant considerations are reported to examine the new solver procedures features.

5.1.1 Standalone TH steady-state for single assembly with single rod material

The first case is a steady-state condition in a single assembly with imposed constant 2.5 MW/m linear heat generation, in the region between 1.2 m and 2 m, with a coolant flow rate of 173.3 kg/s and outlet pressure of 2 bar. The power distribution was smoothed up through the use of a small ramp of 0.25 m.

Due to the incapability of the old code to perform a pseudo transient in standalone mode, i.e. when it is not coupled to NE module, the results are taken imposing a long enough transient of 120 s in order to be sure that the outputs results were stabilized and

so convergence from the time dependents terms satisfied. In this case the pure axial model was used.

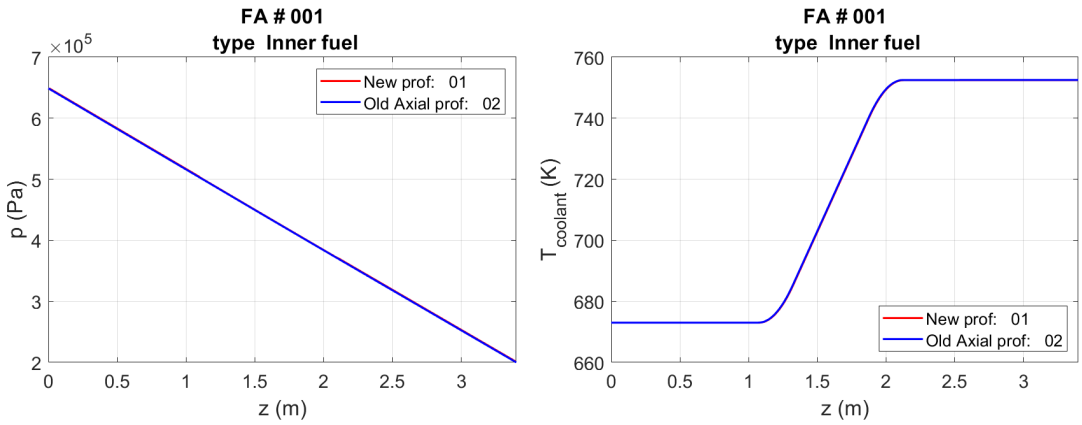


Figure 5.1: Pressure drop and coolant temperature, single assembly constant. heating

The evaluated pressure drops are substantially the same and the required mass flow rate and outlet value of 2 bar is respected by the two codes, figure 5.1. Being based on the same friction factor correlations, the pressure drop profiles results are not so much affected by small differences in materials properties due to temperatures. Therefore, the assumptions made on the simplified approach are justified by the accuracy of results (3.5). In this case with imposed constant heating, significant agreement were found on temperature profiles, outcomes show in shape good substantially the same behaviour. On the other hand the improved evaluation of surface and mean fuel temperature leads to some degree of differences in the evaluation of the mean and center profiles, figure 5.2.

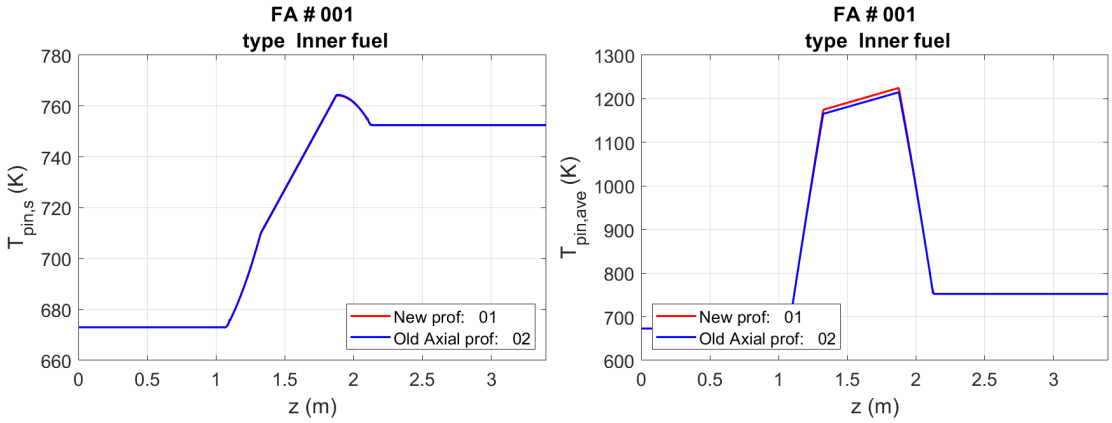


Figure 5.2: Surface rod and mean average temperatures, single assembly constant heating.

It was found that the coolant is slightly colder in the new code with an output value of 752.47 K against the 752.52 K. That found delta is negligible for what concern changing in materials properties and so heat transfer profiles are also practically identical 5.3.

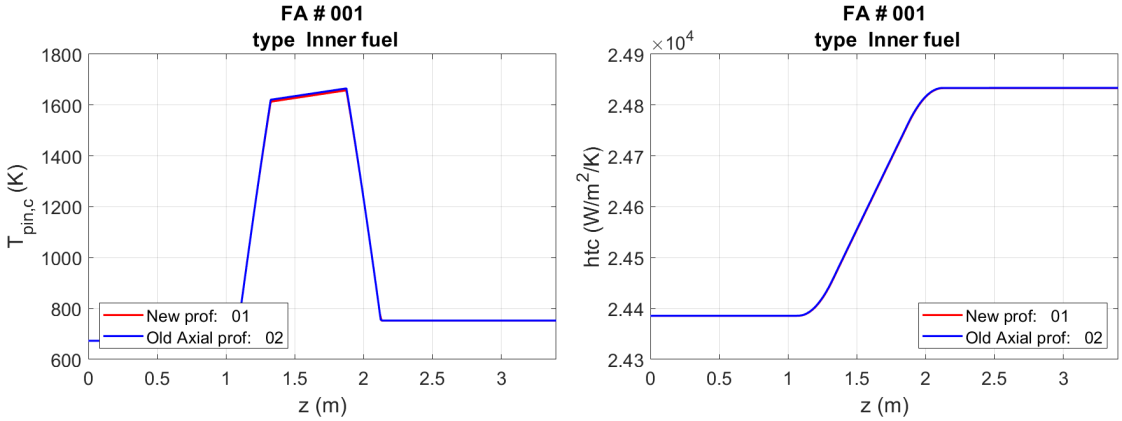


Figure 5.3: Central temperature and heat transfer coefficient, single assembly constant heating.

A check on conservation of energy was done for sake of completeness, comparing the energy deposited in the rods and the rods surface heat exchanging. Results are reported in table 5.1, dimensions are W.

Table 5.1 Energy conservation (W), steady state, single assembly, rods single material

prof	Power dep.	$\dot{Q}_{T_s->T_{cool}}$	ϵ_r %
new TH	1999987.5	1999978.9	4.3E-4
old TH	2000110.5	2000110.5	0.0

In steady state, the power deposited in the fuel pins should result in the same increment of the coolant internal energy between inlet and outlet. The differences in output can be due to the fact that the old code takes into account the dissipation energy deposition from friction effects, neglected in the new version of the energy equation.

It has to be mentioned that the small error in the coupling between Ts and coolant is not really due to a bad coupling in the solver but mainly due to the post-processing phase. After the steady computation, the profiles are processed by the radial approach to possibly send in output the radial fuel profiles. This surface temperature so is not the one computed and used in the `SteadyEneSolver`, leading to the small difference that cause this energy check error in the output subroutine. In fact, as can be seen in the following transient cases test in multi assemblies, where the radial model is already employed during the steady-state evaluation, this error is absent.

5.1.2 Standalone TH transient for single assembly with single rod material

Following the previous case, a reduction of the coolant flow rate to 123.3 kg/s in a short time of 0.4 seconds was imposed. The time steps adopted were fixed in both codes for the sake of a fair comparison. The old code was tested in both radial and axial mode, to see possible

differences in the time-dependent behavior of rods profiles with the different approaches. It is known that in a transient case the axial approach is not strictly correct, since the analytical radial temperature profile is retrieved from the solution of the steady-state heat conduction equations. In view of that, the solver strategy, discussed in section 3.5, that was adopted takes advantage of the best of the two approach: speed in computational time from the analytical model for the steady-state initialization; better estimation of the radial model for transients regime. Coolant temperatures show a better agreement with the old radial approach, since the 1.5D model radial fuel evaluation is practically the same. However, that difference in the outlet coolant temperature still emerges from figure 5.4, demonstrating that is not something related to the fuel model approach.

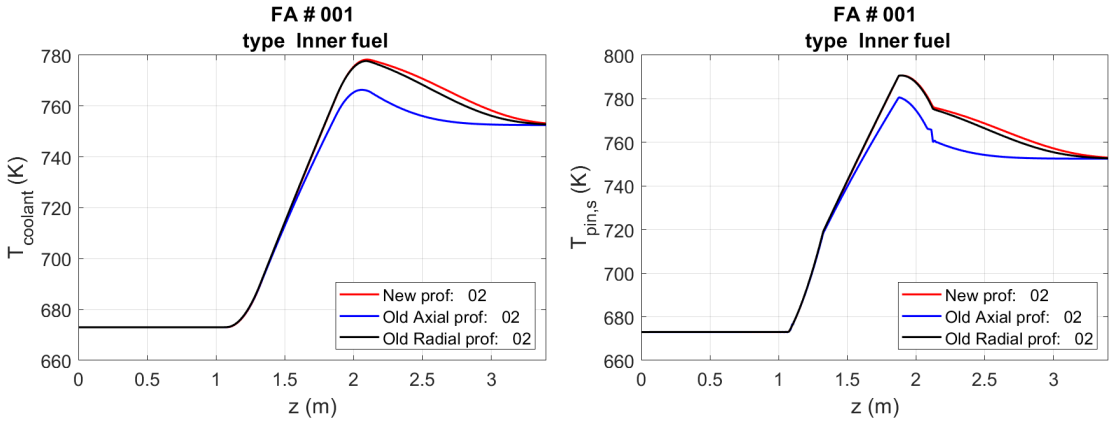


Figure 5.4: Coolant and surface temperatures, single assembly constant heating, time 1.5s.

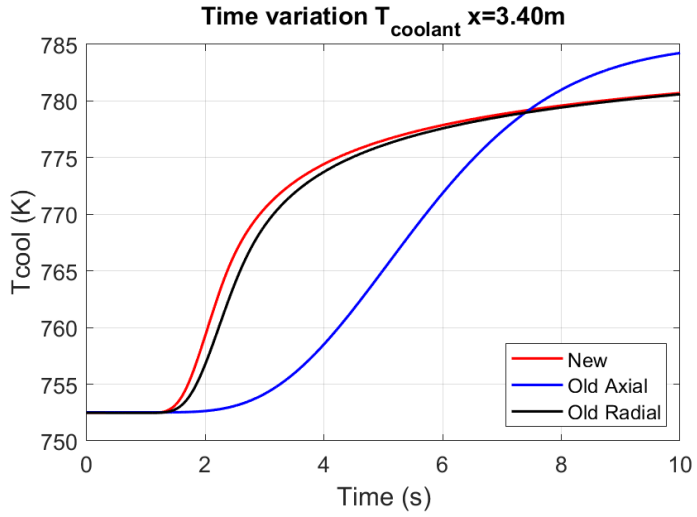


Figure 5.5: Time dependence of the coolant temperature at the outlet

At time 1.5 s models the models show a slightly different time response due to the average mean approach not being well-suitable during transients. At the new steady state, the radial and axial models of the old code are practically the same.

Looking for the system response during transient in figure 5.5, the time variation of lead in outlet is reported, showing the same temporal response of the old radial implementation, thanks to the correct radial modelling of fuel pins. However, it was found like a small thermal delay maybe due to the non adoption of frozen coefficients and better modeling of inertia terms. The conservation of energy check and local temperatures are reported in tables 5.2 and 5.3, at time 0 s and after a transient of 50 s when the outputs are again stabilized.

Table 5.2 Energy conservation (W), transient, single assembly, rods single material.

prof	time	Power dep.	$\dot{Q}_{T_s->T_{cool}}$	ϵ_r %
new	0s	1999987.5	1999978.4	5.0E-6
axial	0s	2000110.5	2000110.5	0.0
rad	0s	2000110.5	/	/
new	50s	1999987.5	1997748.1	0.112
axial	50s	2000110.5	2000110.5	0.0
rad	50s	2000110.5	/	/

Table 5.3 Outlet and exit coolant temperatures, transient, single assembly, rods single material

prof	time	New TH	Old Radial	Old Axial
$T_{cool,exit}$ (K)	0s	752.47	752.48	752.52
$T_{cool,exit}$ (K)	50s	785.04	785,12	785.19

5.1.3 Standalone TH steady-state for a 37 HA core with single rod material

For the multi assemblies case, a mini core of 37 assemblies was prepared as shown in figure 5.6.

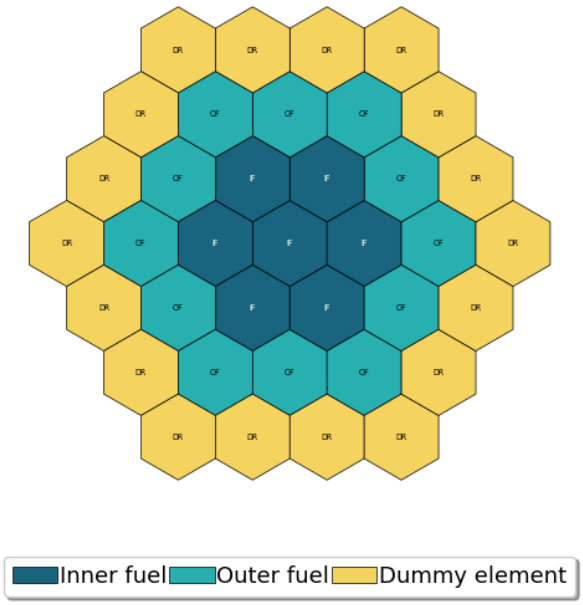


Figure 5.6: Core configuration 37 HAs.

To generate a variable distribution of core temperatures, in this case are applied flat constant power generation of: 2.5 MW/m for the internal assemblies; 2.4 MW/m for the outer region; zero generation for the dummy assemblies. The different flow rate applied boundary condition of: 173.3 kg/s for the internal HA, 145.2 kg/s for the outer ones and 117.5 kg/s for the dummy HA, leads to a peculiar case in which the internal assemblies receive an incoming heat flux from the one in the outer region, that heats up more. Due to the symmetries of the core configuration the results on channels 1, 2, 3, 4 are analysed, thus being representative of the entire core.

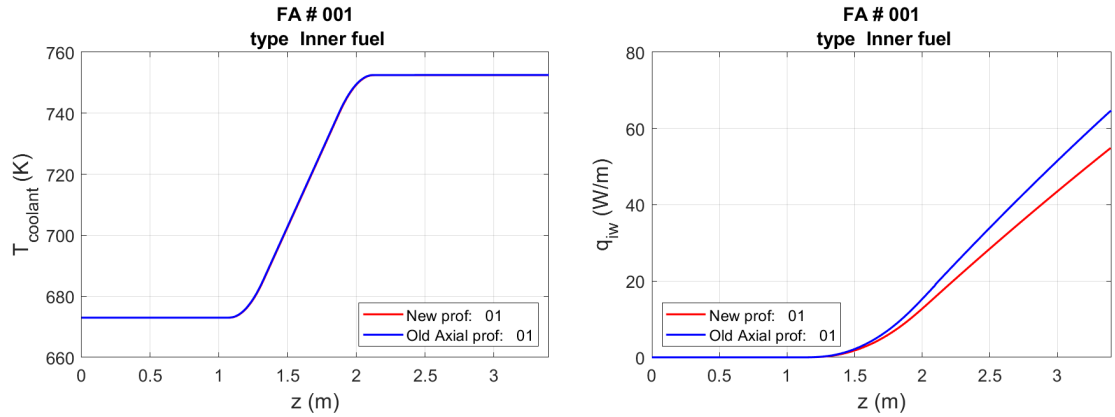


Figure 5.7: Coolant temperature and inter-wrapper power exchange assembly 1, constant heating.

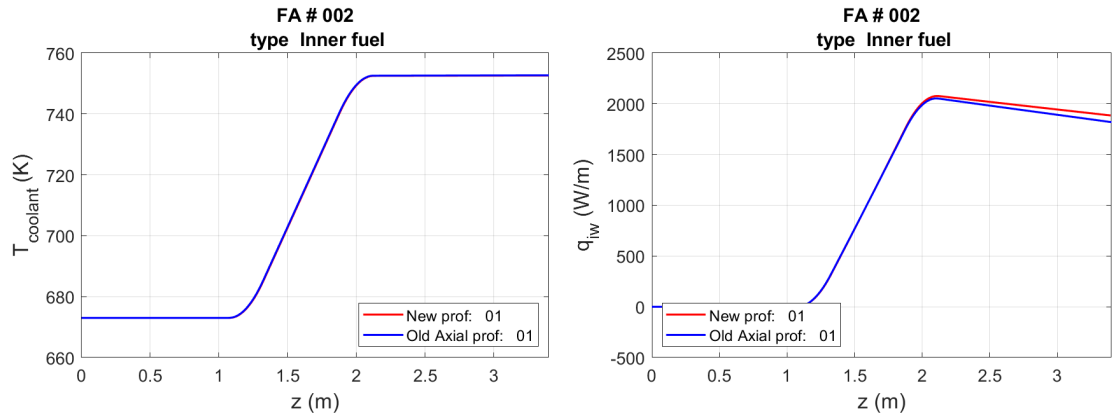


Figure 5.8: Coolant temperature and inter-wrapper power exchange assembly 2, constant heating.

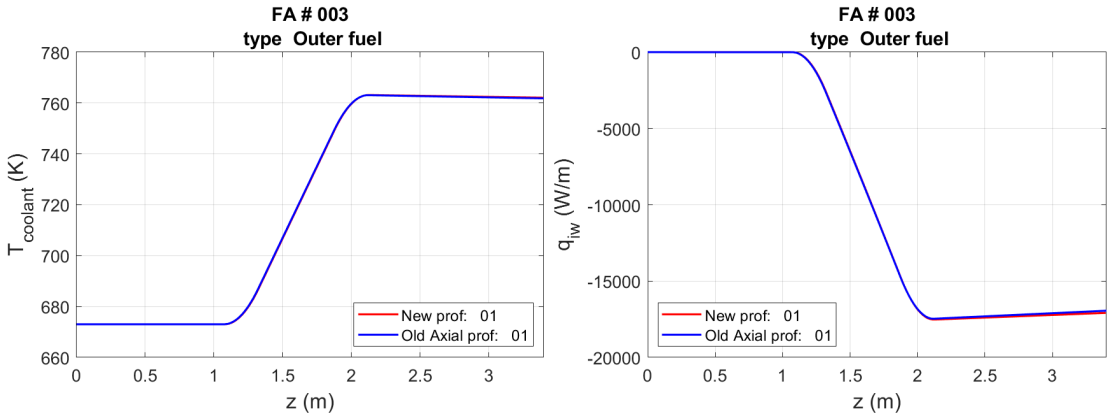


Figure 5.9: Coolant temperature and inter-wrapper power exchange assembly 3, constant heating.

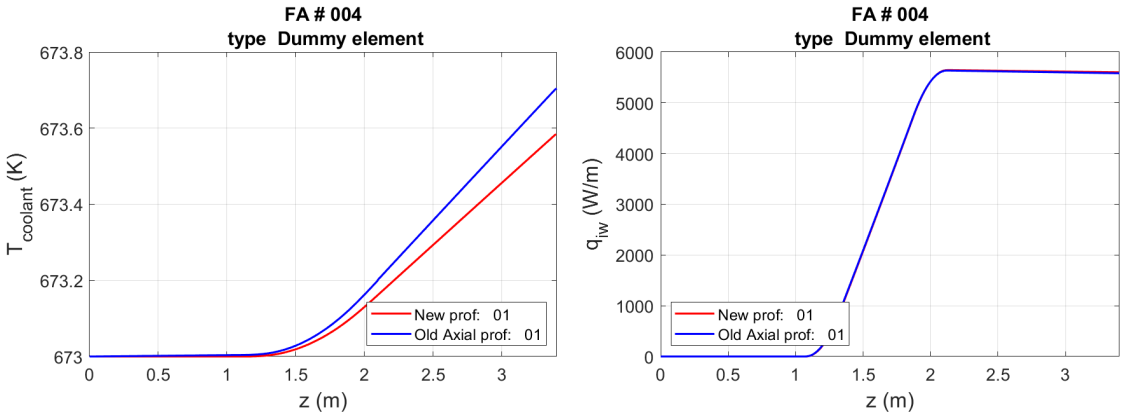


Figure 5.10: Coolant temperature and inter-wrapper power exchange assembly 4, constant heating.

Additionally to what was already seen in the previous single HA cases, the presence of the inter-channel thermal exchange coupled the energy equations of channels, thanks to the additional power source reported in figures 5.7, 5.8, 5.9, 5.10. The resulting profiles are again in good agreement in shape with some differences in the HAs outlet maybe due to the neglected viscous terms. Also it can be appreciated as in this case due to the symmetry of the configuration and equal power generation in the inner core HAs, the contribution of transversal heat transfer in FA 1 is negligible and order of magnitude less in comparison to the others neighbour channels. Due to that the difference with the previous implementation is remarkably less affected by the changes in coolant temperatures.

There are also differences in the treatment of the dummy elements, where the absence of friction energy dissipation, causes a flat profile in the initial part of the channel in the new TH module, while it starts to heat up from the beginning in the old model, resulting

in an outlet temperature slightly higher with the old solver.

Fuel temperature have the same behaviour of 5.2, due to the imposed flat linear power deposition. The energy conservation check is reported in table 5.4 for the four selected assemblies.

Table 5.4 Energy conservation (W), steady state, assemblies 1-4, rods single material.

prof	HA	Power dep.	$\dot{Q}_{T_s->T_{cool}}$	ϵ_r %
new	1	1999987.5	1999978.9	5.0E-6
axial	1	2000110.5	2000127.3	-8.4E-4
new	2	1999987.5	1999978.9	4.30E-4
axial	2	2000110.5	2000127.8	-8.6E-4
new	3	1919988.0	1919979.6	4.4E-4
axial	3	1920106.1	1920183.9	-4.1E-3
new	4	0	0	/
axial	4	0	0.9	/

In this case, also in the new solver the error is not zero but the worse relative error of 6.22E-3% in the outer region could be absolutely acceptable. Furthermore, it will not imply to relevant different degree in rods and coolant temperatures evaluations, at most some decimal digits that will not affects the FRENETIC scope. The errors are probably linked to the explicit treatment of inter assembly coupling or numeric truncation so a case with no channels inter-coupling is proposed.

5.1.4 Standalone TH steady-state for a 37 HA core with single rod material and no inter-wrapper heat transfer

To measure how much this additive exchange term affects the conservation of energy, the mini core is tested in the same conditions as before but with the new introduced `ADIABATIC` condition, for the new channels input, corresponding to `QBOX=0.0d0` in the old, to assume laterally adiabatic channels. Temperature profiles are not reported since the results are nearly identical as before, due to the weak strength of the inter-HA heat transfer with respect to the heat generation inside the rods. In fact, for what concerns the heated assemblies, the conductive coupling to the near boxes is a second order effect in comparison to advection and coupling with the heated rods. Being now the additional source coupling term zero, without external heating the outer region assemblies show a flat profile in the new model since there is not any source term.

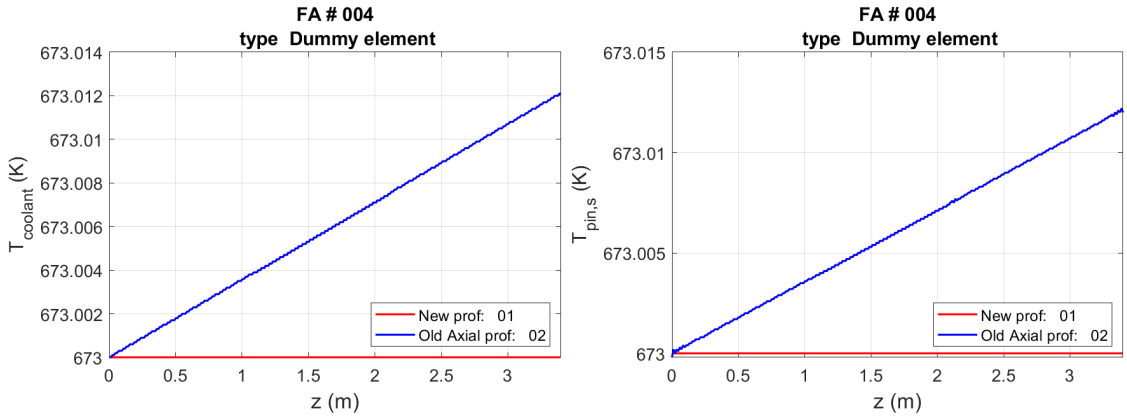


Figure 5.11: Coolant and surface temperatures, HA 4.

From figure 5.11 it can be seen how in the old code the surface temperature exhibit a nonphysical oscillation in at inlet and outlet. Conservation of energy is reported in table 5.5.

Table 5.5 Energy conservation (W), steady state, assemblies 1-4, no inter-channel coupling, rods single material.

prof	HA	Power dep.	$\dot{Q}_{T_s->T_{cool}}$	ϵ_r %
new	1	1999987.5	1999978.9	4.3E-4
axial	1	2000110.5	2000111.4	-4.5E-5
new	2	1999987.5	1999978.9	4.3E-4
axial	2	2000110.5	2000111.4	-4.5E-5
new	3	1919988.0	1919979.6	4.4E-4
axial	3	1920106.1	1920113.3	-3.8E-4
new	4	0	1.44E-09	/
axial	4	0	-0.001	/

From residuals can be seen how in this case, with the same imposed `tolltosteady`, the error, in the channel conservation of the new solver, becomes zero in the new module. This indicates that, in the previous case, convergence was probably not perfectly reached due to the explicit coupling of the channels. However, the error on channel 4 could be derived by some numerical instability given by the steady matrix solver, that was seen to not accept values of relative tolerance of $9 \cdot 10^{-10}$, under this value, in case of 'PSEUDO' method, the `SteadyEnesolve` undergoes numerical oscillations that leads to stop of the programs because the internal maximum number of iterations was reached. Still, from the numerical point of view and conservation of energy and momentum the new simplified and finite volume based solver seems to have a better behaviour and not suffer from any kind of macroscopic oscillations.

5.1.5 Coupled NE-TH steady-state for a 37 HA core with single rod material

To have a more realistic representation of the rods temperatures profiles, some cases coupled to the NE module were performed to have a better model of energy deposition and effects on fuel profiles. The old code was tested with the radial approach to better account for the radial profile of the rods' temperatures. It was imposed the same inlet flow rates of the case with flat heating. The power amplitude for the NE module input was imposed to 25 MW, and the heated zone is between 1.4 m and 2 m with profiles shown in figures 5.12 and 5.13.

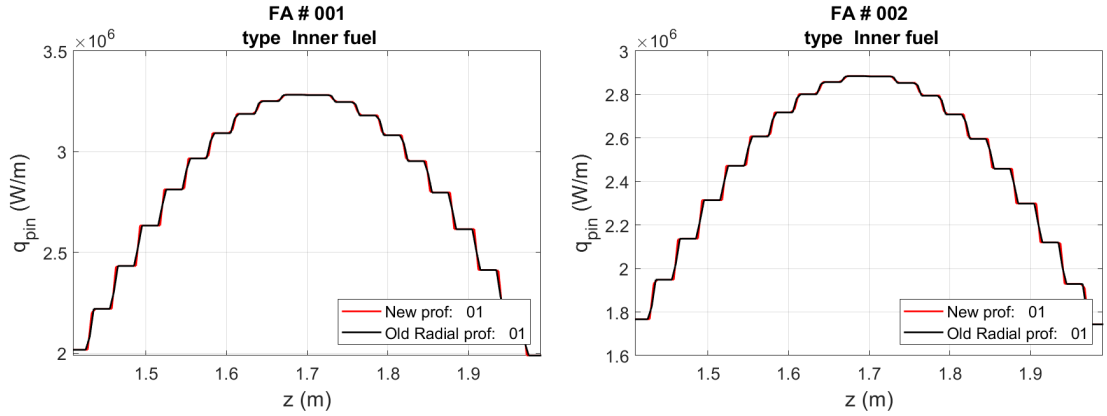


Figure 5.12: Global linear power deposition, HAs 1 and 2.

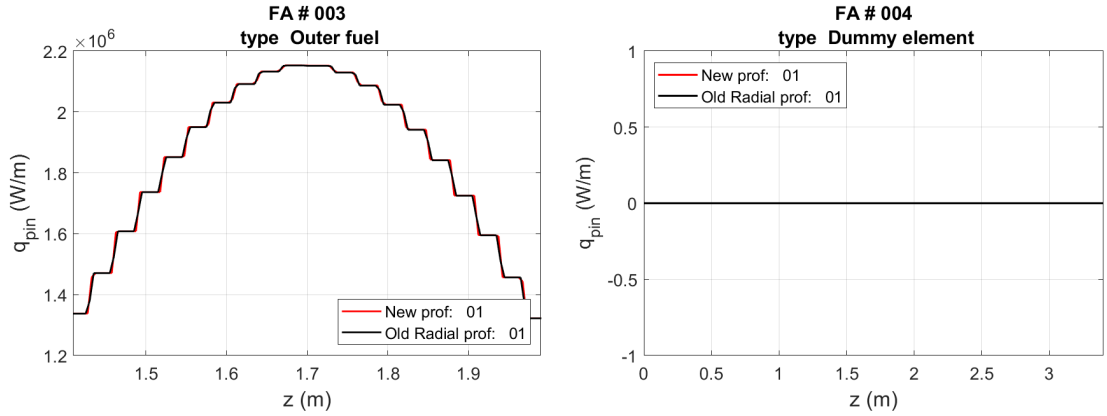


Figure 5.13: Global linear power deposition, HAs 3 and 4.

As obvious, in steady state the power distribution results essentially the same but necessary to validate the updated subroutines for data coupling of the two modules. The inter-channel coupling, in figures 5.14 and 5.15, shows good agreement with the old

implementation, with increased heat fluxes in the second half of the channels in coincidence to the increment of heat transfer coefficients.

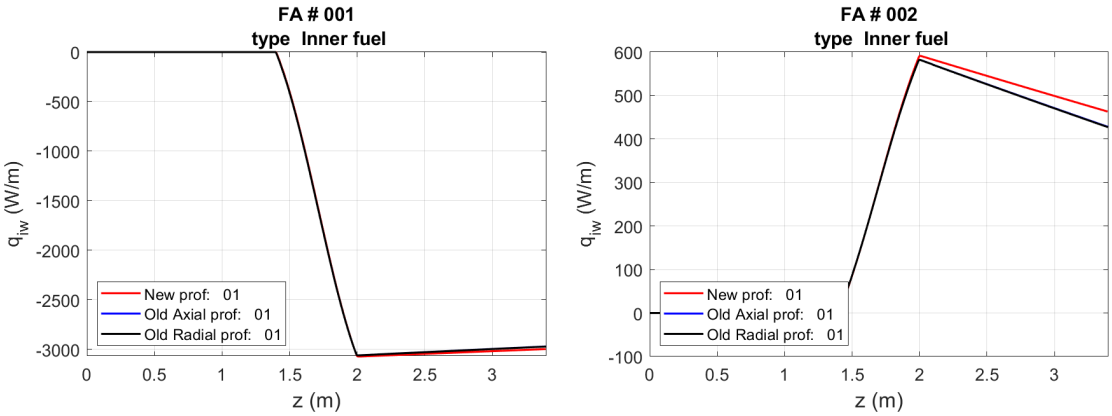


Figure 5.14: Assemblies linear power thermal exchange, HAs 1 and 2.

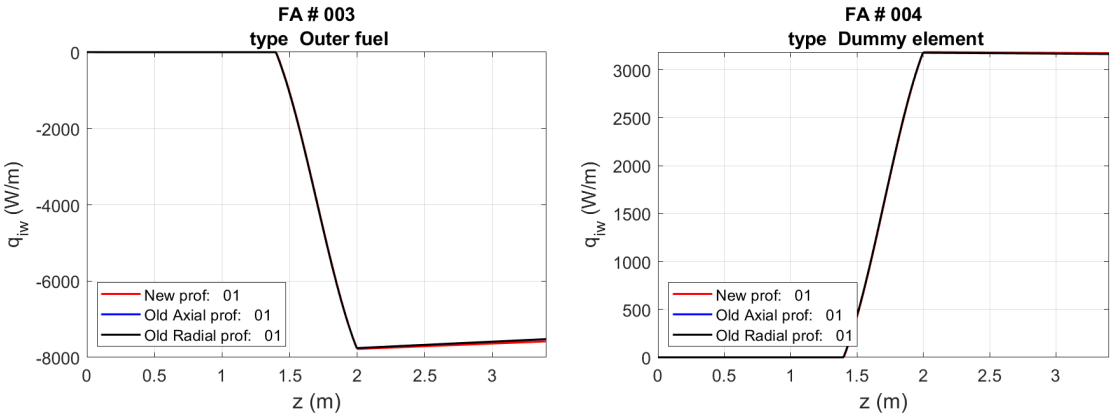


Figure 5.15: Assemblies linear power thermal exchange, HAs 3 and 4.

Coolant heating is slightly higher also in this case but the effects on fuel rods profiles is less notable due to the spatial profile of heat generation. In figures 5.16 and 5.17 are reported the central and surface values in some relevant assemblies, in the refined mesh zone.

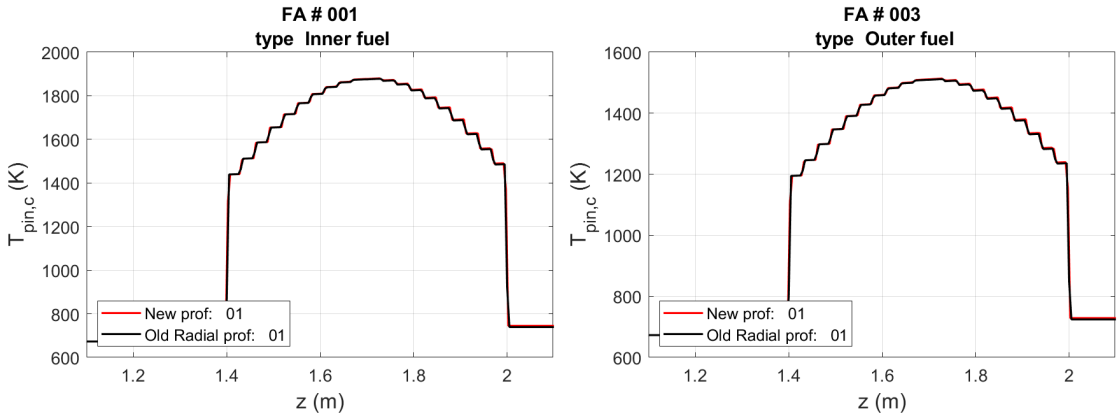


Figure 5.16: Center fuel temperatures, HA 1 and 3.

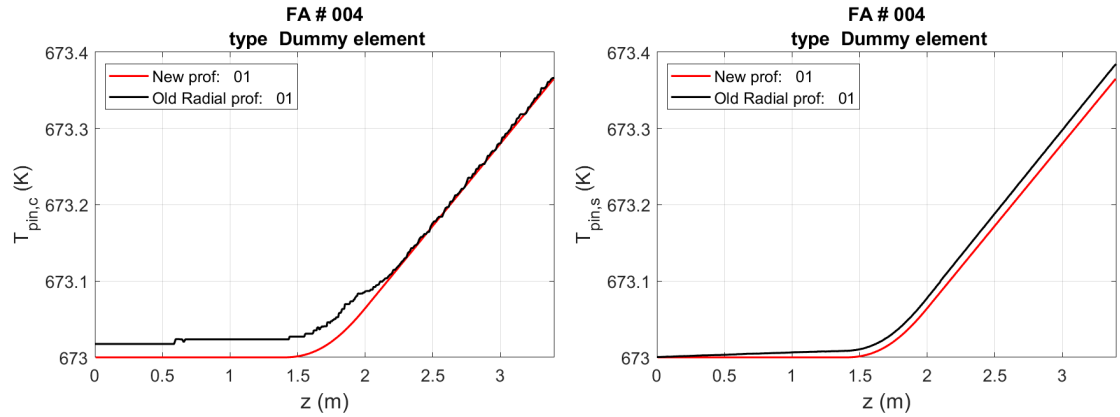


Figure 5.17: Center and surface fuel temperatures, HA 4.

Profiles are in particularly good agreement, but the dummy elements show more smoothed profiles in the new solver. In this case the absence of power deposition shows how, in the old solver, some numerical error builds up in the evaluation of surface temperature, and associated to that, the mean and central one.

This non smooth behaviour can be the result of a bad coupling in the two-points solution scheme, here amplified by the FEM approach. The conservation of energy check is done also for this case reported in table 5.6.

Table 5.6 Energy conservation, steady state with NE, assemblies 1-4, rods single material.

prof	HA	Power dep.	$\dot{Q}_{T_s->T_{cool}}$	ϵ_r %
new	1	1666793.0	1666785.9	4.3E-4
radial	1	1665034.1	/	/
new	2	1464290.0	1464283.5	4.4E-4
radial	2	1462754.1	/	/
new	3	1097141.5	1097136.4	4.6E-4
radial	3	1095979.8	/	/
new	4	0	8.81E-10	/
radial	4	0	-0.001	/

The heat transfer from rods surface to coolant was not computed in the radial 1D method of the old code and so left blank. Relative errors show the same magnitude of previous cases with negligible values in the new solver, probably improvable by performing more iterative core resolution, in this case speeded up by an increase of the NE input under-relaxation temperature value to 0.7. Also, the profiles obtained are not the most precise since, due to the steady state case and given end simulation time of zero seconds, the solution comes from the steady solver based on the mean approach while the surface temperature is found by a radial approach. On the other hand, due to a bug, the radial model of the previous implementation could not work with imposed zero simulation time, so a 0.1 s run was done.

5.1.6 Coupled NE-TH steady state for a 37 HA core with single rod material and no inter-wrapper heat transfer

Aiming at accounting for the effects of inter-channel heat transfer, the assemblies were simulated adiabatic. While the main HAs show good agreement in fuel and coolant temperature, figure 5.18, with usual differences, in this case the old radial approach shows strange results in the outer region.

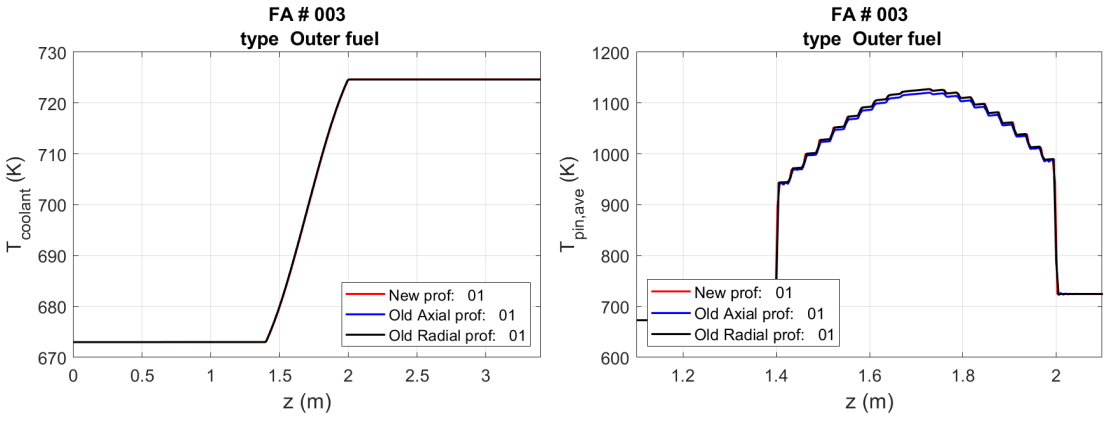


Figure 5.18: Coolant and average temperatures profiles, HA 3.

In case of disabled heat transfer between channels, the dummy elements shows in figures 5.19 and 5.20 a behaviour even worse than before, with a nonphysical step changing in the evaluation of Tave and Tc as a consequence of non-smooth and correct coupling between Ts and coolant, even if convergence should be already reached and the NE exit from steady state. The magnitude of temperatures changes is of course irrelevant - which is probably the reason why this effect was not noticed in previous benchmark activities- but the strange profile suggests that the channels coupling is not responsible for the errors found.

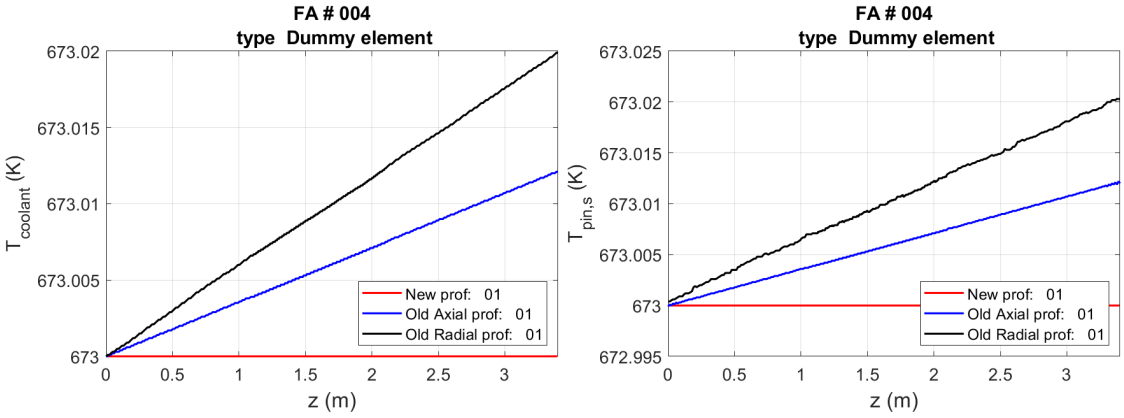


Figure 5.19: Coolant and surface temperatures profiles, HA 4.

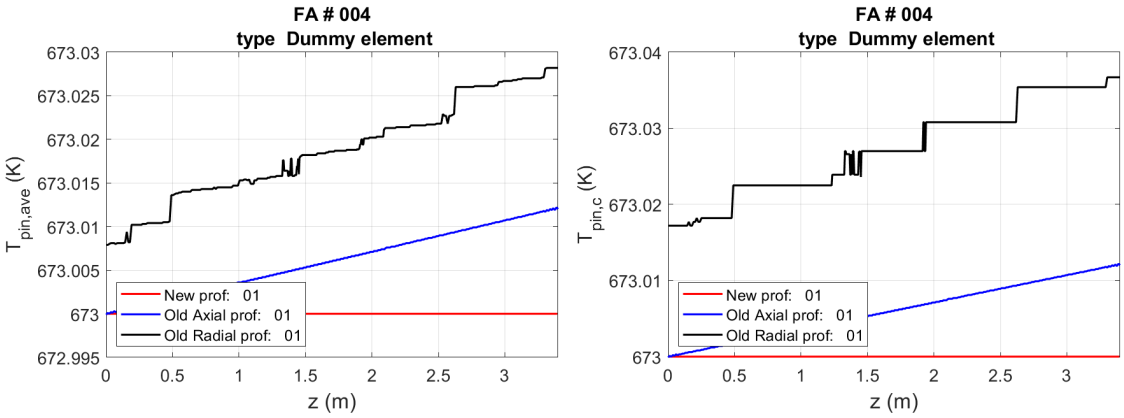


Figure 5.20: Mean and center temperature profiles, HA 4.

In fact, especially the old radial model shows difficulties in convergence with the needs to increase the `TOLLDER` error, although the code runs indefinitely due to oscillations in residuals, hinting that something was wrong in the numerics of the last proposed build. On the other hand, the majority of the improvements given with the new solver result in a general better stability, faster running time and improved consistency of the solver. Looking for the errors in the conservation of energy in table 5.7, the absence of inter-wrapper coupling leads, also in this case, to zero (to machine precision) error in the new solver.

Table 5.7 Energy conservation, steady state with NE, assemblies 1-4, no inter-channel coupling, rods single material.

prof	HA	Power dep.	$\dot{Q}_{T_s->T_{cool}}$	ϵ_r %
new	1	1668347.5	1668340.4	4.3E-4
radial	1	1665037.1	/	/
new	2	1465001.5	1464995.0	4.4E-4
radial	2	1462744.6	/	/
new	3	1096553.5	1096548.4	4.6E-4
radial	3	1095985.2	/	/
new	4	0	1.88E-9	/
radial	4	0	0	/

5.1.7 Coupled NE-TH transient for a 37 HA core with single rod material

A short transient of one second coupled with the NE module is here analysed, comparing the results also with the old axial model. Of particular relevance in this case are the obvious better evaluation of mean average and central values from the radial rods' models 5.21, but not only. The axial profiles show significant oscillations in magnitude in the evaluation of surface temperature profiles, which are shown in figure 5.22, sign that again something was not working properly in the old solver with the adopted numerical parameters.

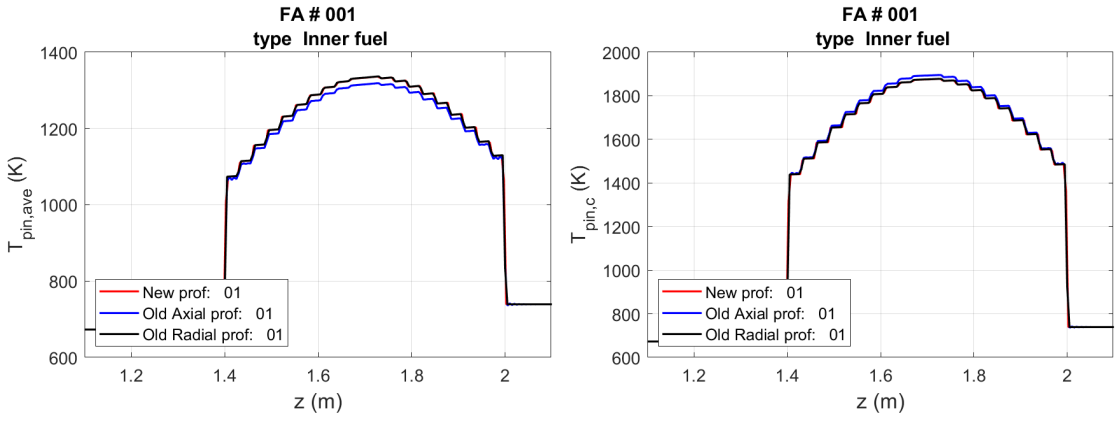


Figure 5.21: Mean and center temperature profiles, HA 1.

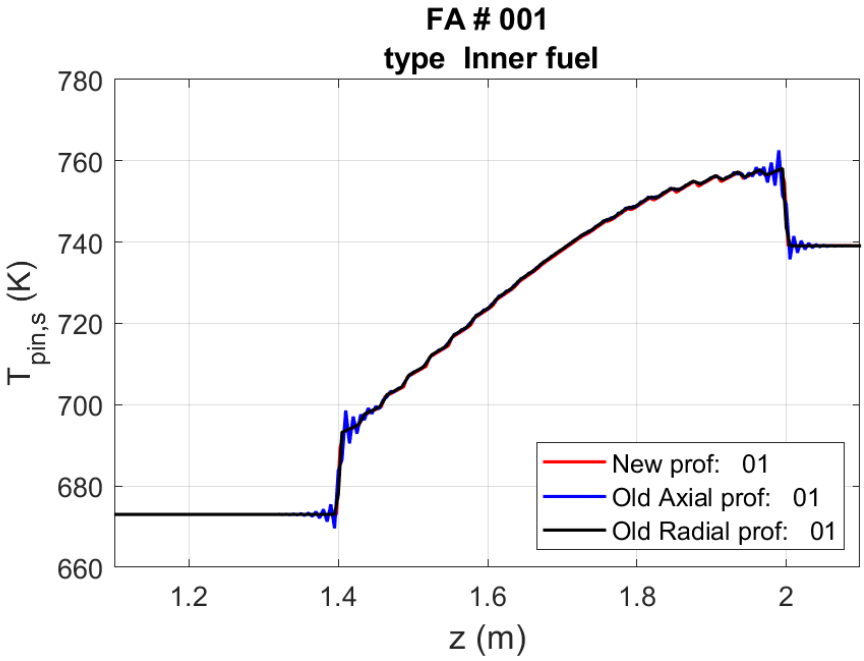


Figure 5.22: Surface temperature profiles, HA 1.

In this case the conservation of energy control, reported in table 5.8, on the heat transfer between rods and coolant seems again well respected by the old axial model but the results are for sure influenced by the presence of that big oscillations. Overall the effect of these oscillations somewhat cancels out, resulting in a less than 1% error in conservation of energy.

Table 5.8 Energy conservation, transient with NE, time 0 s, assemblies 1-4, rods single material.

prof	HA	Power dep.	$\dot{Q}_{T_s->T_{cool}}$	ϵ_r %
new	1	1666783.5	1666783.5	0
radial	1	1665034.1	/	/
axial	1	1664956.8	1665885.6	-0.056
new	2	1464288.0	1464288.0	0
radial	2	1462754.1	/	/
axial	2	1462674.4	1463438.1	-0.052
new	3	1097147.0	1097147.0	0
radial	3	1095979.8	/	/
axial	3	1095916.6	1096559.3	-0.059
new	4	0	4.6E-8	/
radial	4	0	0	/
axial	4	0	-0.0403	/

5.1.8 Coupled NE-TH transient for a 37 HA core considering the actual fuel rod material composition

As a last test case, it is reported one with layered rods. To test code functionalities a series of cases with layered rods were done in parallel, thus accounting for the difference in presence of layers for the gas thickness and cladding material. Here is reported a short transient of 1s coupled with the NE module in which is imposed a reduction of the coolant flow rate of 50 kg/s in the inner 1 HA. Since the profiles are common with the already discussed cases, with just a small increment in fuel rods profiles given by the presence of the layers, it is reported profiles at 1s to highlight also some possible different behaviour of the inertia term. Looking at HA 1, in fact we can see how the profiles show differences in the evolution of coolant temperatures in the channel and, related to that, inter-channel linear power coupling, figure 5.23.

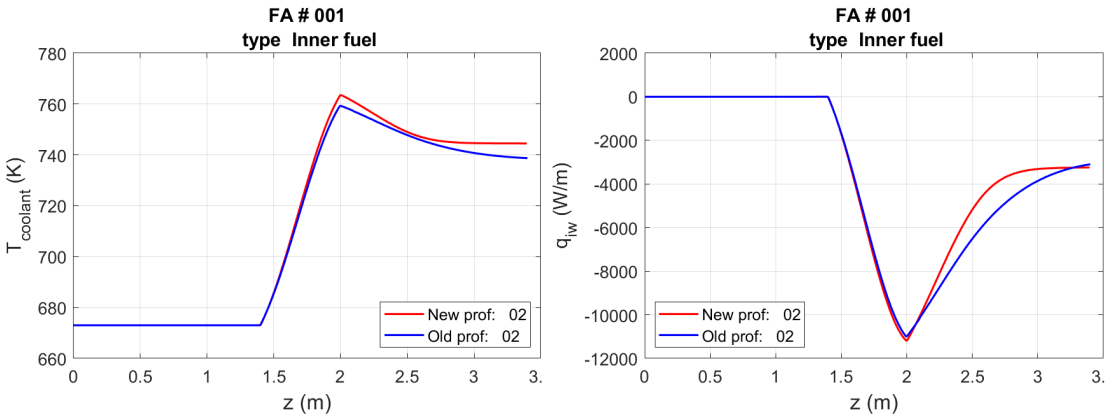


Figure 5.23: Coolant temperatures and inter-channel coupling, HA 1.

The different coolant temperature will also influence the estimation of surface temperature, see figure 5.24. Pressure drops are, as other cases, in good agreement.

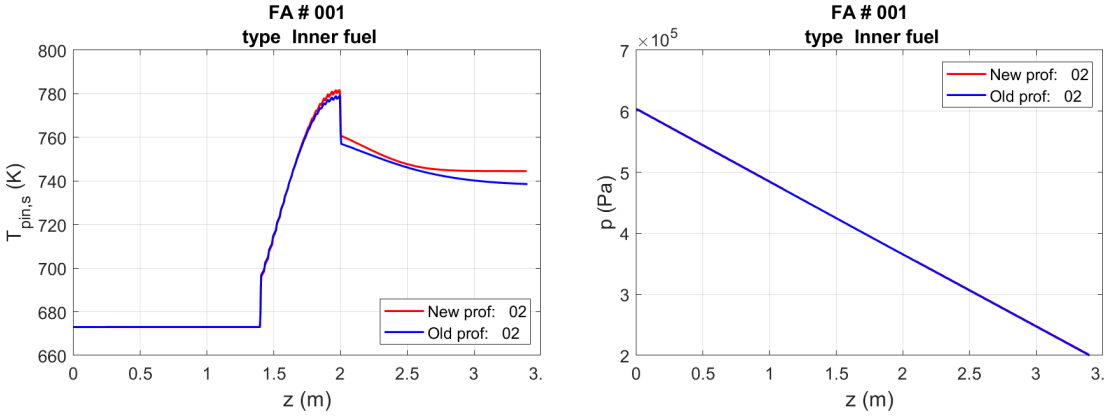


Figure 5.24: Surface temperature and pressure drop, HA 1.

Even if the radial profiles of fuel temperatures are comparable in the two codes, strange results were found from the old radial model in the integration of the mean rods temperature, in all the cases with layered rods. The profiles show a shifted baseline 5.25, nonphysical, even if the evaluations should be the same in function of the radial temperatures and points coordinates. Probably the bad outdated usage and declaration of variables causes some unwanted errors in the old `heater.f90` procedures.

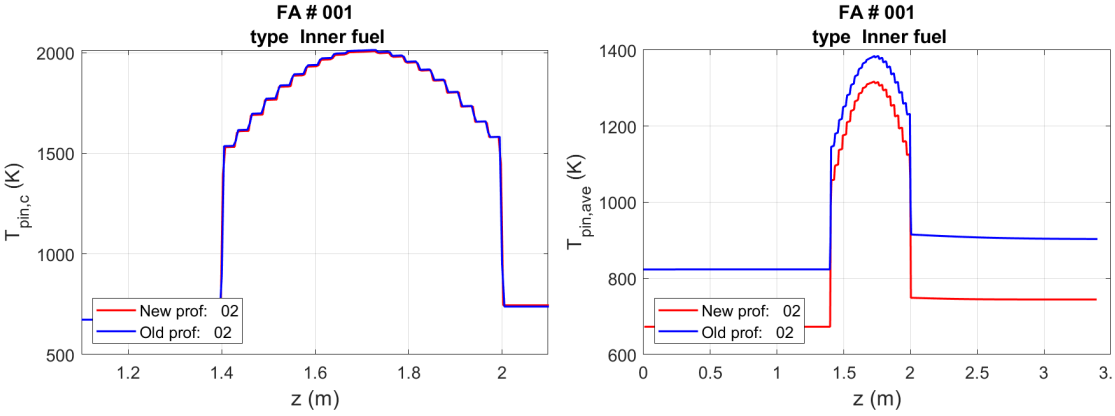


Figure 5.25: Central and mean average temperature, HA 1.

Same considerations can be done for the near channels, perturbed by the central assembly, for which in figure 5.26 the incoming linear power coupling is shown. Probably the slightly different time response is given by different treatment of inertia term and the use of more correct iterative procedures to compute the thermal exchange between coolant and rods as well as a different temporal response of the NE module.

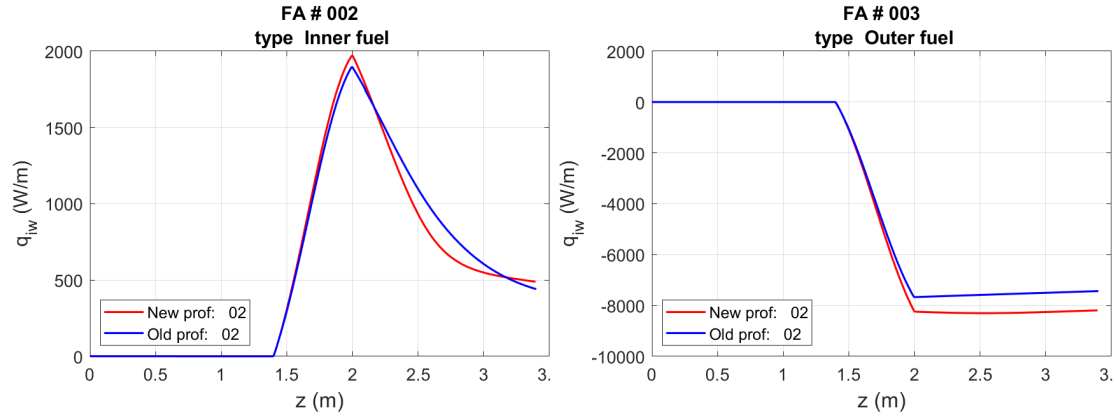


Figure 5.26: Heat transfer inter-wrapper coupling, HAs 2-3.

Again, the worse results of the previous implementation are highlighted by the external non heated assemblies in which the the non-smooth behaviour of coolant leads to a step

changing on fuel profiles modelling 5.27.

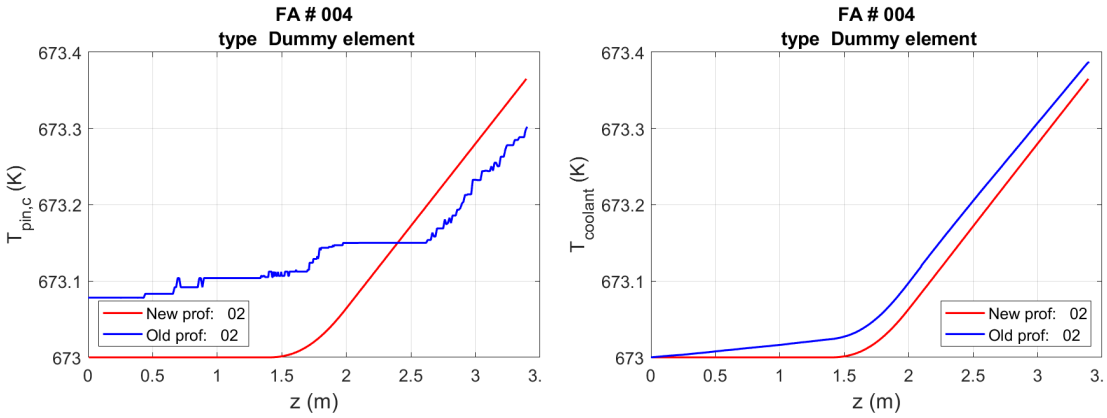


Figure 5.27: Coolant and central fuel temperature, HA 4.

From the energy error check in table 5.1.8, the newly introduced solver shows again negligible errors, under 0.005% in the heated channels, while the old implementation shows a worsening of the coolant/rods coupling, to respect the previous non layered cases, ending in a underestimation of the coolant outlet temperature 722.71 K vs 718.79 K.

Table 5.9 Energy conservation, transient with NE, time 0 s, assemblies 1-4, rods layered materials.

prof	HA	Power dep.	$\dot{Q}_{T_s->T_{cool}}$	ϵ_r %
new	1	1666784.0	1666784.0	0
radial	1	1665305.8	/	/
new	2	1464281.0	1464281.0	0
radial	2	1462912.0	/	/
new	3	1097144.5	1097144.5	0
radial	3	1095867.7	/	/
new	4	0	4.61E-5	/
radial	4	0	0	/

5.1.9 Effect of non-fuel rods

To check how possible peaks and time variations of coolant temperatures are affected by the presence of these rods, it is proposed a short transient in single assembly made of 70 fuel rods and 57 non heated stainless-steel rods. At time 0.1 s the flow rate is reduced from 173.3 kg/s to 123.3 kg/s. The time variation of coolant temperature at exit and 1.8 m is shown in figure 5.28.

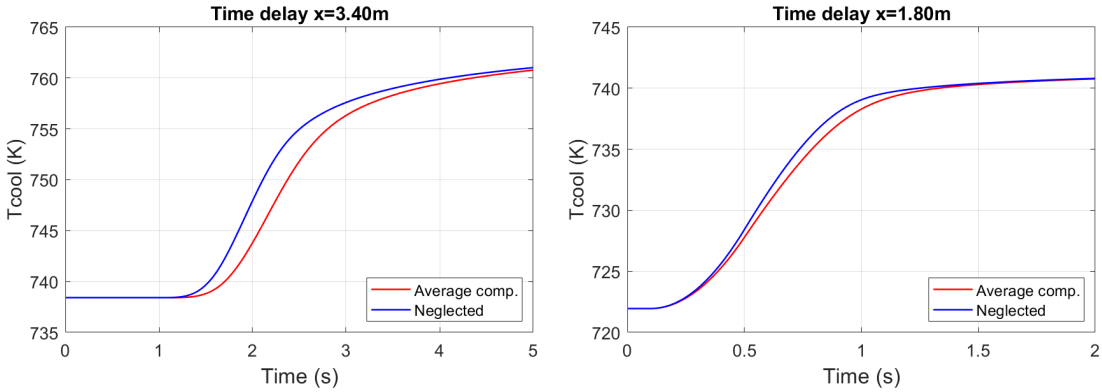


Figure 5.28: Time evolution of coolant.

As expected, the presence of the additional inertia given by the non heated rods, decreases the thermal excursion introducing a time delay thanks to rods/coolant thermal balance. The improvements, carried out in the evaluation of the correct thermal response of the core, could be even more remarkable in a real case with more mixed materials and a real distribution of power distribution, instead of the linear constant imposed here.

5.2 Steady-state ALFRED benchmark

To proof the new introduced code capabilities in a close to real simulation condition, the core of ALFRED (described in section 1.2.3) is modeled the nominal steady state. The core configuration reported in 5.29 was created thanks to the `coreutils` Python toolkit provided by Dr. N. Abrate, using nominal core and assemblies' dimensions. The code is compared with the previous version adopting consistent material properties correlations and channels condition capabilities, according to ALFRED core configuration.

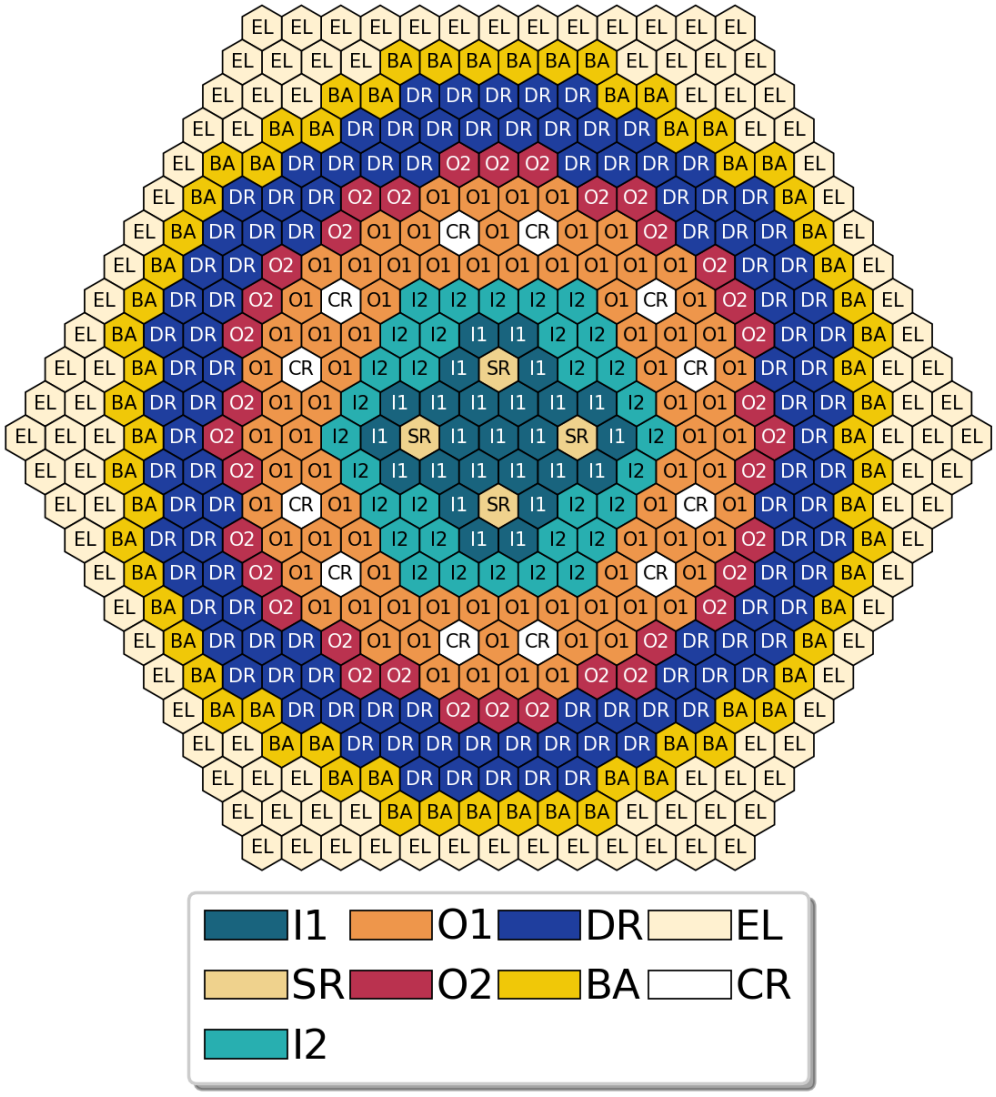


Figure 5.29: Simulation core configuration.

To be consistent with the NE module geometry, the actual assemblies of the ALFRED core are surrounded by a series of fictitious dummy hexagonal assembly to represent the presence of the barrel and the surrounding lead. These channel were simulated as pseudo stationary, mass flow rate of 0.1 kg/s, and one rod of 'SS'A. It must be mentioned that thanks to the new code functionalities for future work those channels can be now model as completely empty and eventually with no flow, thus saving computational time.

During the modelling of this case some problems were found in the channel-oriented core steady-state initialization. To promote the stabilization of temperatures' distribution, avoiding the insurgence of oscillation due to the channel coupling, the conductive inter-channel heat flux is now featured with an under-relaxation factor `tollFlux`, given in input. This slightly worsens the required time for the initial iterations steps, but is necessary to account properly additional channels of this kind, where the main forcing power source is the external coupling, instead of the pins' power deposition. This is an example of how user experience with realistic test cases is essential to complete the code development and verification phase, rendering it ready for future validation campaigns and applications.

The simulation was performed on the same machine taking advantage of the introduced parallelization with 4 threads. The meshes' discretization were of 351 axial elements and an under-relaxation factor of 0.1 was imposed for temperatures in the NE module. In the new TH input the following were set: 1.0E-9 relative error for `TOLLTOSTEADY`; `underTemp` of 0.5; `underFlux` of 0.1 and pseudo transient model. The previous implementation was tested with imposed `TOLLDER` of 1.0E-3, relaxation factor for temperatures of 0.05 and in pure axial model. The resulting computational time, required to reach the steady state condition, is reported in table 5.10 and it confirms the expectations, showing a remarkable reduction in necessary time, resulting in 14x faster convergence.

Table 5.10 Executions times and related max relative error on last core iteration.

prof	Execution time	NE iterations	ϵ_r
New	43m 39s	51	8.48E-10
Old	10h 24m	52	4.23E-08

The results were not only obtained quickly but are also more precise, see the last obtained maximum relative error to previous core iteration.

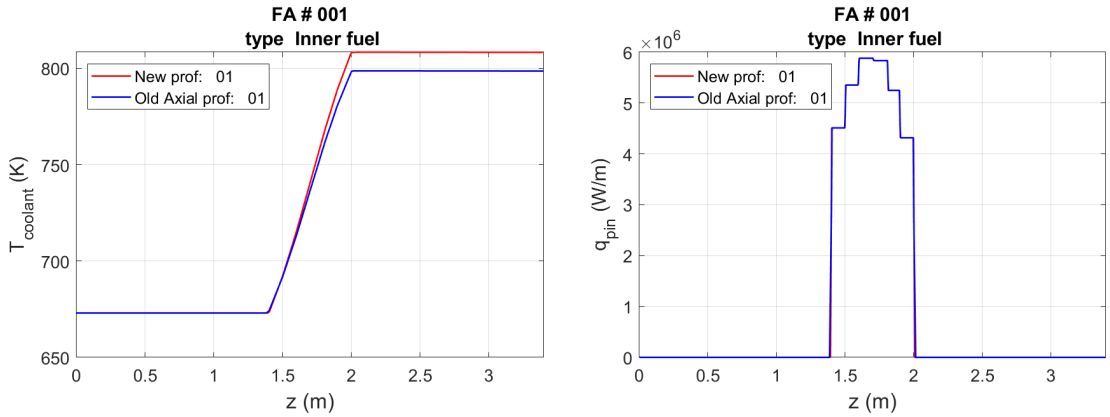


Figure 5.30: Coolant temperature and power deposition, HA 1, ALFRED simulation.

As shown in the previous test cases, the coolant behaviour in the new model is different, following the expected profile, figure 5.30. However, the maximum surface and centre value are overestimated by the old code due to a bad evaluation of the surface profile, showing very big oscillation moving forward from the not loaded to heated zone, figure 5.31.

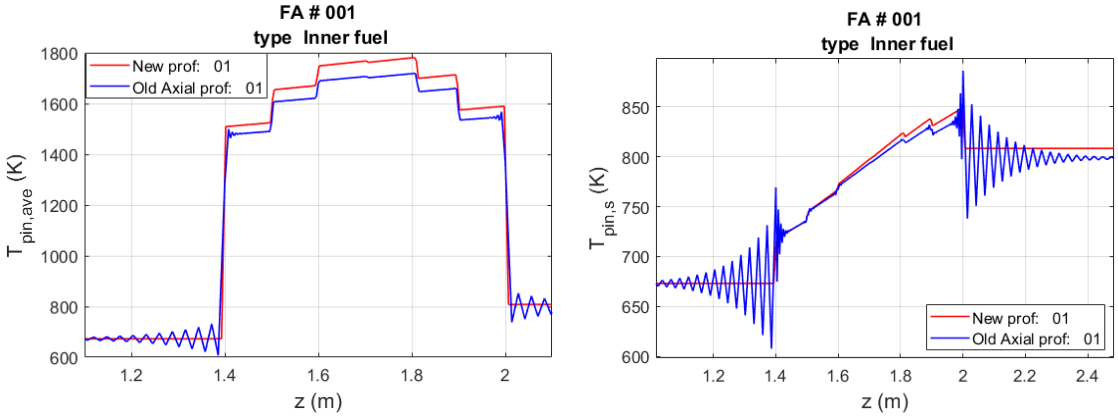


Figure 5.31: Average and surface temperature, HA 1, ALFRED simulation.

To complete the comparison, is has to be mentioned that the computational time can be different in a transient case. In fact, the adoption of the radial approach during the steady state initialization slow down the convergence rate. In the same condition of previous test, results for a 0.001s transient are reported in table 5.11:

Table 5.11 Executions times and related max relative error on last core iteration, for steady state initialization of a transient case.

prof	Execution time	NE iterations	ϵ_r
New rad	1h 58m	49	9.52E-10

These results were obtained with an imposed `STPMINSTEADY` of 1 s but the convergence rate is strictly linked to that. The execution can be improved by selecting value bigger than that. In fact, thanks to the improved stability and implicit time scheme, it was seen in other cases that time step one order of magnitude larger are well accepted increasing convergence rate without affecting the final results.

During transient operations the required computational time, if chosen constant time steps, are comparable and will not affect so much the simulation time. The computational burden in that case is in fact due mainly to the neutronic part, where the usage of more advanced model, such as the Improved Quasi-Static Method, slow down the computation significantly.

From these results, the following conclusions can be drawn: the updated implementation of the TH module is faster and more efficient, taking advantage of the specific physics of the problem to simplify as much as possibile the equations without losing accuracy. Moreover, it is more robust with respect to the selected axial discretization: the oscillations that were shown in the old code results can of course be controlled by changing the axial discretization, as was done in the past, but the present discretization relieves the user from this trial-and-error-like pre-processing.

Chapter 6

Conclusions and future perspective

In this thesis, a new TH module for the FRENETIC code was designed, developed, implemented and tested. The test cases performed proved how the new implementation reaches all the requirement functionalities and satisfies all the proposed goals. Even if less precise in the evaluation of energy dissipation term in the energy equation and on the evaluation of velocities profiles, based on the coolant incompressibility, the newly implemented finite volume formulation shows to be more reliable and more accurate in terms of evaluation of the coolant-rods thermal coupling, thanks to the iterative procedure, and eventually also in the time changing of materials properties with frozen coefficients assumption that can be used or not. All this brings the smoother profiles of temperature, not affected by oscillation even in case of sharp spatial variations of the power deposition or strong inter-channel coupling. Computational time was also improved, with parallelization that results in almost double speed with respect to the previous implementation in steady state, and comparable execution speed during transient calculations.

The approach adopted for code development enhanced modularity and reliability of the code, thus representing constitute a solid base for future improvements of the code, to introduce new features and solve some open questions.

A list of possible points for future developments follows:

- Validate properly the new finite volume approach by comparing it with trusted simulations data obtained with other codes for transient calculations. Preliminary comparisons with results obtained with well-known computational fluid-dynamic tools such as STAR-CCM+ and OpenFOAM (not shown in this thesis) showed very good agreement of the computed solutions for the simple case of an infinite pin lattice. However, more careful studies are required to fully qualify the new TH module.
- Improve theNE-TH coupling, perhaps moving towards NE-driven time steps to follow the neutron shape updates. There is the need of a better time management to avoid that inconsistency in reading of time steps and avoid that during transient the errors build up increasing the magnitude and leading to a not perfect synchronization. This sometimes potentially leads to stop of the program because the imposing next

time step are infinitesimal less than the machine precision. To solve this issue, the implementation of preCICE [26] coupling library for partitioned multi-physics simulations could significantly improve the software robustness. Further improvement can be reached in the initialization phase by the adoption of a residual balance method in NE and TH module coupling to not oversolve the core hydraulics during preliminary computation in which the neutron flux is computed with no optimal effects in thermal feedbacks [32]. Therefore the required running time can be improved.

- Implement a new output procedure, to update the old implementation to better standards and decouple the output of the channels and the output of files for debug purposes. Also, there is the need of adding the output for BiB that now is impossible since it does not represent any additional channels of the system. Designing properly the new output subroutine there could also be the possibility to introduce a restarting operation, in accordance with NE, using appropriate flags and writing on file the vectors of structure 'bb', 'th', 'nf'. In this way there could be the chance to restart some long transients where the looking for target were not reached at the imposed end simulations time. The output phase should be reviewed also in the NE module as, in the development branch for the new TH, is currently bypassed in case the output time is very close to the next transient time, less than the imposed machine precision. This should be easily solvable by moving to on output when the time t is reached instead of when it is started from.
- There is the possibility to send to NE a better averaging of the 'solid' part of materials inside HAs. Since the materials cross sections arise from the homogenization of all the elements inside the core, so fuel, cladding, non-fuel rods etc; to better represent the thermal feedback, inside mixed composition assemblies, it could be possible send a temperature that is averaged to also the non fuel rods, now computed during transient. Also, the validation of non fuel rods temperature validity have still to be demonstrated with a real case.
- Adopt an improved different mesh construction to account for a possible area variation in the channel.
- The validation of thimble modelling in the new TH have still to be properly done, as well as the validation of the new implemented adaptive time steps adoption.

In conclusion, it can be stated that the new TH module, together with a set of Python classes purposely developed to simplify the input generation and post-processing phases, improved the code quality and maturity. Based on these achievements, a short-term plan to render the code open-source appears very feasible. This can potentially increase the number of researchers that adopting FRENETIC for the simulation of full-core coupled NE-TH transients in fast reactors and/or as a platform to test new solution methods.

Appendix A

Material properties

The user will be able to select one among the available correlations for the liquid metal density, viscosity, specific heat and thermal conductivity as already implemented. It should be noticed that adding further materials or different databases in the future is relatively straightforward.

A.1 Coolant properties

The user can choose as before between lead, LBE and sodium correlations according to [33]. With respect to the previous implementation, the user selection now occurs via keywords rather than via numbers, thus simplifying the code utilization. The following nomenclature was adopted:

- 'Pb'
- 'LBE'
- 'Na'
- 'Na-RELAP'

A.2 Fuel rod properties

The user can chose as before between two different kind of fuel and non-fuel materials. The following keywords are available:

- 'UO2'
- 'U5Fs'
- 'B4C'
- 'SS'

A.3 Wrapper properties

Wrapper material properties can be chosen by selection from the input file. They are listed inside cladding materials properties since cladding and wrapper materials are typically selected from the same set. In addition to the previously implemented stainless steels, other possible austenitic steels, ferritic-martensitic steels are adopted, going towards HLMCR constrain and design choices 1.2.2.

The label for the new introduced correlations are:

- '15-15Ti'

$$c_p = 431.0 + 0.177 \cdot T + 8.72 \cdot 10^{-5} / \ T^2 \quad \frac{J}{kg\ K} \tag{A.1}$$

$$k = 8.826 + 1.707 \cdot 10^{-2} \cdot T - 2.315 \cdot 10^{-6} \cdot T^2 \quad \frac{W}{m\ K} \tag{A.2}$$

- 'T91'

$$c_p = 244.0 + 4.677 \cdot T - 1.103 \cdot 10^{-2} \cdot T^2 + 1.151 \cdot 10^{-5} \cdot T^3 - 3.979 \cdot 10^{-9} \cdot T^4 \quad \frac{J}{kg\ K} \tag{A.3}$$

$$k = 17.94 + 2.51 \cdot 10^{-2} \cdot T - 1.45 \cdot 10^{-5} \cdot T^2 \quad \frac{W}{m\ K} \tag{A.4}$$

- 'SS316'

$$\rho = 8084 - 0.4209 \cdot T - 3.894 \cdot 10^{-5} \cdot T^2 \quad \frac{kg}{m^3} \tag{A.5}$$

$$c_p = 462.0 - 0.134 \cdot T \quad \frac{J}{kg\ K} \tag{A.6}$$

$$k = 9.248 + 1.157 \cdot 10^{-2} \cdot T \quad \frac{W}{m\ K} \tag{A.7}$$

- 'He'
  It is also given new temperature dependent values for the modelling of helium gap, giving isobaric properties at 5bar and conduction correlation only as a function of temperature, as the one used in MATPRO code[34]:

$$k = 2.639 \cdot 10^{-3} \cdot T^{0.7085} \quad \frac{W}{m\ K} \tag{A.8}$$

Due to the lack of available correlations in references, for wide range of temperature for T91 and 15-15 materials, their densities are implemented as the one for stainless steels and needs to be updated in the future.

Appendix B

Constitutive relations

Constitutive correlations of empirical nature are employed with the possibility for the user to choose among the available correlations for the friction factor and the Nusselt number. The user can choose between them in input file using a more readable nomenclature through strings label, calling the adopted correlation properly, as for the materials.

Here are listed with the new label, type of correlations and validity range.

B.1 Friction factor

- 'CONST': constant f=0.005
- 'BUNDLE': laminar/turbulent along rod bundles from [35]

$$\begin{aligned} f_T &= \frac{C_{f,T}}{Re^{0.18}} \\ f_L &= \frac{C_{f,L}}{Re} \\ C_{f,L|T} &= a + b_1(P/D-1) + b_2(P/D-1)^2 \end{aligned} \tag{B.1}$$

Where coefficients for C_{fL} and C_{fT} are reported in table B.1

	$1.0 \le P/D \le 1.1$			$1.1 < P/D \le 1.5$		
Subchannel	a	b_1	b_2	a	b_1	b_2
Laminar flow						
Interior	26.00	888.2	−3334	62.97	216.9	−190.2
Edge	26.18	554.5	−1480	44.40	256.7	−267.6
Corner	26.98	1636.	−10,050	87.26	38.59	− 55.12
Turbulent flow						
Interior	0.09378	1.398	−8.664	0.1458	0.03632	−0.03333
Edge	0.09377	0.8732	−3.341	0.1430	0.04199	−0.04428
Corner	0.1004	1.625	−11.85	0.1499	0.006706	−0.009567

Figure B.1: Coefficients for bare rod subchannel friction factor constants in hexagonal array from [35]

- 'BLASIUS': Blasius correlation
  Blasius [36] has proposed a simple correlation for the friction factor in a smooth circular tube:
  $$f = \frac{0.316}{Re_{D_h}^{0.25}} \tag{B.2}$$
  where $Re_{D_h} = (\rho v D_h)/\mu$ is the Reynolds number computed using the hydraulic diameter as the characteristic length. This correlation is applicable in turbulent regime up to $Re < 10^5$.
- 'WIRED': Pressure drop in wire-wrapped hexagonal array pin bundles from [35]
  $$\begin{aligned} &f = \frac{C_{fT}}{Re^{0.18}} \qquad f = \frac{C_{fL}}{Re} \\ &C_{fL} = [-974.6 + 1612.0(P/D) - 598.5(P/D)^2](H/D)^{0.06-0.085(P/D)} \\ &C_{fT} = [0.8063 - 0.9022(log_{10}(H/D))] + 0.3256[log_{10}(H/D)]^2(P/D)^{9.7}(H/D)^{1.78-2.0(P/D)} \end{aligned} \tag{B.3}$$

Correlations 1 and 3 are as a function of Reynolds number, rods pitch and diameters, of rods and wire wrapper.

B.2 Nusselt number

The heat-transfer correlations for liquid metals are derived from experiments performed in dedicated facilities or CFD simulations, and are specific for the kind of rod bundles in terms of shape (e.g. triangular lattice or squared) and extension (7-19-37 pins etc.). Due to the signifcant spread of these experimental data, care should be taken when applying such correlations, also bearing in mind that no one was developed specifically for wired wrapped pins.

- 'BiB': Seban-Shimazaki, liquid metals flow in smooth pipe, uniform axial wall temperature and uniform radial heat flux
  $$Nu = 5.0 + 0.025 Pe^{0.8} \tag{B.4}$$
- 'SCHAD': Schad-modified
  $$\begin{aligned} &Nu = 4.496[-16.15 + 24.96(P/D) - 8.55(P/D)^2] \quad \text{for } Pe \leq 150 \text{ and } 1.3 \leq P/D \leq 2 \\ &Nu = [-16.15 + 24.96(P/D) - 8.55(P/D)^2]Pe^{0.3} \quad \text{for } 150 < Pe \leq 1000 \end{aligned} \tag{B.5}$$
- 'USHAKOV': Ushakov
  $$\begin{aligned} Nu = 7.55(P/D) - 20.0(P/D)^{-13} + 0.041(P/D)^{-2} Pe^{0.56+0.19(P/D)} \\ \text{for } 150 \leq Pe \leq 4000 \text{ and } 1.3 \leq P/D \leq 2 \end{aligned} \tag{B.6}$$

- 'EXPERIMENTAL': Interpolation based on experimental data obtained at the Brasimone research center
- 'MIKITYUK': Mikityuk - Fully Developed Flow in Bare Bundles in triangular lattice

$$Nu = 0.047(1 - e^{-3.8(P/D)-1.0})(Pe^{0.77} + 250.0) \quad \text{for } 30 \leq Pe \leq 5000 \text{ and } 1.1 \leq P/D \leq 1.95 \tag{B.7}$$

- 'KAZIMI': Westinghouse - Carelli-Kazimi

$$Nu = 4.0 + 0.33(P/D)^{3.8}(Pe/100.0)^{0.86} + 0.16(P/D)^5 \quad \text{for } 10 \leq Pe \leq 5000 \text{ and } 1.1 \leq P/D \leq 1.4 \tag{B.8}$$

A comparison of experimental results of wired pins against several correlations [37] concluded that the average Nu was lower than that predicted by the majority of the correlations for bare rod bundles at the same Pe. Results showed values between the Carelli-Kazimi and the Mikityuk correlations, and the slope of the experimental trend is very similar to the correlations. For wire-wrapped fuel, the Carelli-Kazimi correlation being more conservative fit near to the available data so is suggested. For fully developed flow in bare bundles, the derived correlation by Mikityuk shows good agreement with data analysed in [38]. To appreciate the differences between correlations a comparison in the prediction of an average heat transfer inside an assembly as function of Pe number, at 700 K and $P/D = 1.32$, in their validity range, is showed in figure B.2. If not chosen the most suitable correlation, the assumption on values can be deviated by a significant amount.

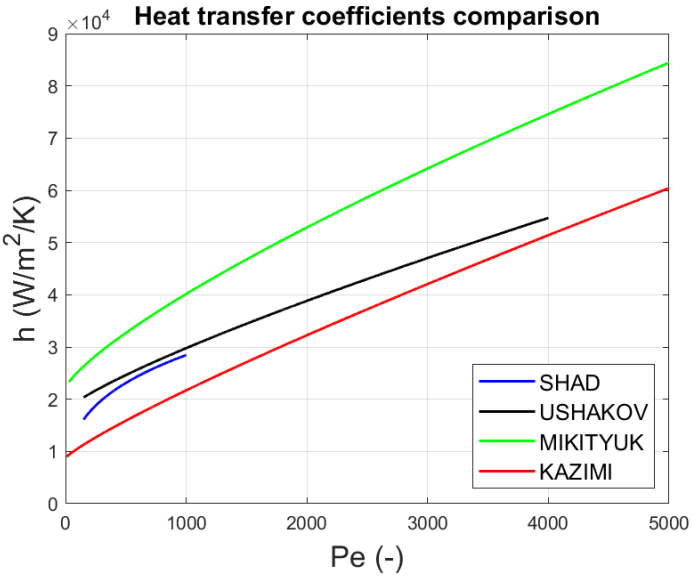


Figure B.2: Comparison correlations for heat transfer.

Appendix C

Core design goals

In this very short appendix, the core design goals - i.e., features that are not mandatory but are worth implementing if time allows - are reported.

- G1: Parallelization of TH module.
- G2: Use of external libraries.
- G3: Possibility to accept in input the core pressure drops.
- G4: Time dependent energy model for the thimbles' coolant.
- G5: Time dependent energy model to account thermal inertia of non-fuel rods.
- G6: Dynamic variables allocations.

Acronyms

ALFRED Advanced Lead-cooled FR European Demonstrator. 6, 78, 79

BiB Box-in-the-Box. 24, 26, 35, 36, 40, 41, 45, 47, 50, 52, 84, 88

CFD Computational Fluid Dynamics. 7, 8, 88

DFD Data Flow Diagram. 31
DHR Decay Heat Removal. 4, 5
DOC Design-Oriented Code. 8
DPA Displacement Per Atom. 3, 5

FA fuel assembly. 4, 8, 9, 13, 64
FEM Finite Element Method. 28, 35, 38, 46, 47, 51
FR Fast Reactor. 5
FRENETIC Fast REactor NEutronics/Thermal-hydraulICs. vii, 9–11, 13–15, 24, 84

GFR Gas-Cooled Fast Reactor. 2
GIF Generation IV International Forum. 2

HA hexagonal assemblies. xi, xii, 10–12, 14, 15, 21–23, 29, 33, 36, 38, 45, 46, 50, 54, 57, 62, 64, 66–76, 84
HLMCR Heavy Liquid Metal Cooled Reactor. 9, 86

IW Inter-Wrapper. 13, 46

LBE Lead-Bismuth Eutectic. 6, 85
LEADER Lead-cooled European Advanced DEmonstration Reactor. 6
LFR Lead-cooled Fast Reactor. vii, 2–9
LM Liquid Metal. vii, 2–5, 9
LWR Light Water Reactor. 2, 5

MSR Molten Salt Reactor. 2

NE Neutronics. x, 7, 9, 12, 15, 32, 36, 44, 45, 47, 50, 51, 53, 67, 70–74, 76, 79, 83, 84

SA sub assembly. 7
SCWR SuperCritical-Water-cooled Reactor. 2

SDID Software Design and Implementation Document. 11
SFR Sodium-cooled Fast Reactor. 2

TH Thermal-Hydraulic. iii, xi, 7, 9, 11, 12, 14, 15, 25, 29, 36, 38, 40, 41, 44, 48, 52, 64, 81, 83, 84, 91
TM Thermo-Mechanic. 7

VHTR Very-High-Temperature Reactor. 2
VOC Verification-Oriented Code. 8

Bibliography

[1] *Technology Brief - Nuclear Power*. Tech. rep. UNECE, 2020 (cit. on p. 1).

[2] A. Stanculescu. *GIF R&D outlook for generation IV nuclear energy systems*. Tech. rep. In Proceedings of the Generation IV International Forum, Paris, France, 2009 (cit. on pp. 1, 5).

[3] Pioro Igor. *Handbook of generation IV nuclear reactors*. Woodhead Publishing, 2016 (cit. on pp. 2, 3, 6).

[4] *Nuclear Heat Transfer and Passive Cooling, Volume 5: Liquid Metal Thermal Hydraulics*. Tech. rep. Frazer-Nash Consultancy, 2021 (cit. on pp. 3, 4, 9).

[5] *Safety Design Criteria for Generation IV Lead-cooled Fast Reactor System*. Tech. rep. The LFR provisional System Steering Committee, 2021 (cit. on pp. 3, 5, 7).

[6] Yvon Pasca (ed.) *Structural materials for generation IV nuclear reactors*. Woodhead Publishing, 2016 (cit. on p. 6).

[7] De Bruyn Didier and et al. «Main achievements of the FP7-LEADER collaborative project of the European Commission regarding the design of a Lead-cooled Fast Reactor». In: *Korean Nuclear Society 4* 281 (2013) (cit. on p. 6).

[8] A. Alemberti, M. Caramello, M. Frignani, G. Grasso, F. Merli, G. Morresi, and M. Tarantino. «ALFRED reactor coolant system design». In: *Nuclear Engineering and Design* 370 (2020) (cit. on p. 6).

[9] G. Grasso, M. Tarantino, and M. Constantin. *ALFRED RDQ&D Infrastructure - PRO ALFRED Conference*. Tech. rep. Falcon, 2020 (cit. on p. 6).

[10] G. Grasso. «The core design of ALFRED, a demonstrator for the European lead-cooled reactors». In: *Nuclear Engineering and Design* 278 (2014), pp. 2040–2053 (cit. on p. 6).

[11] Francesco Lodi. «Development of Core Design Methods and Tools for Gen-IV Heavy Liquid Metal Cooled Reactors». Ph.D. thesis. Alma Mater Studiorum Universit‘a di Bologna, 2017, pp. xlii–324 (cit. on pp. 8, 13).

[12] R. Bonifetto, S. Dulla, P. Ravetto, L. Savoldi Richard, and R. Zanino. «A full-core coupled neutronic/thermal-hydraulic code for the modeling of lead-cooled nuclear fast reactors». In: *Nuclear Engineering and Design* 261 (2013), pp. 85–94. ISSN: 00295493. DOI: 10.1016/j.nucengdes.2013.03.030. URL: http://dx.doi.org/10.1016/j.nucengdes.2013.03.030 (cit. on pp. 9, 14).

[13] G.F. Nallo, N. Abrate, S. Dulla, P. Ravetto, and D. Valerio. «Neutronic benchmark of the FRENETIC code for the multiphysics analysis of lead fast reactors». In: *The European Physical Journal Plus* in press (2019) (cit. on pp. 9, 12).

[14] Dominic Caron, Roberto Bonifetto, Sandra Dulla, Valerio Mascolino, Piero Ravetto, Domenico Valerio, and Roberto Zanino. «Full-core coupled neutronic/thermal-hydraulic modelling of the EBR-II SHRT-45R transient». In: *International Journal of Energy Research* 42 (2018), pp. 134–150 (cit. on pp. 9, 14, 24, 45).

[15] G. F. Nallo. «Modelling liquid metals for nuclear fusion and fission reactors». Ph.D. thesis. Politecnico di Torino, 2021, pp. xxii–288 (cit. on pp. 11, 37).

[16] L. Savoldi and R. Zanino. «M&M: Multi-conductor Mithrandir code for the simulation of thermal-hydraulic transients in superconducting magnets». In: *Cryogenics* 40 (2000) (cit. on pp. 14, 26).

[17] F. Moukalled, L. Mangani, and M. Darwish. *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab®*. Springer, 2021 (cit. on pp. 17, 19).

[18] *ANSYS CFX-Solver Theory Guide*. Tech. rep. ANSYS, Inc., 2009 (cit. on p. 22).

[19] Joel H. Ferziger and Milovan Perić. *Computational Methods for Fluid Dynamics, third, rev. edition*. Springer (cit. on p. 23).

[20] B. L. Wang and Y. W. Mai. «Transient one-dimensional heat conduction problems solved by finite element». In: *International Journal of Mechanical Sciences* 47.2 (2005) (cit. on p. 24).

[21] Roberto Bonifetto, Dominic Caron, Sandra Dulla, Valerio Mascolino, Piero Ravetto, Laura Savoldi, Domenico Valerio, and Roberto Zanino. *Advances in the development of the code FRENETIC for the coupled dynamics of lead-cooled reactors*. Tech. rep. CIRTEN: CERSE-POLITO RL 1572/2015, 2015 (cit. on p. 24).

[22] A.G. Mylonakis, M. Varvayanni, N. Catsaros, P. Savva, and D.G.E. Grigoriadis. «Multi-physics and multi-scale methods used in nuclear reactor analysis». In: *Annals of Nuclear Energy* 72 (2014), pp. 104–119. ISSN: 0306-4549 (cit. on p. 26).

[23] E. Anderson et al. *LAPACK Users' Guide*. Third. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1999. ISBN: 0-89871-447-8 (paperback) (cit. on p. 28).

[24] Roger S. Pressman. *Software Engineering: a Practitioner's approach*. McGraw-Hill, 2001. ISBN: 0073655783 (cit. on p. 31).

[25] Michael W. Hackemack and Justin M. Pounders. «IMPLEMENTATION OF AN A PRIORI TIME STEP ESTIMATOR FOR THE MULTIGROUP NEUTRON DIFFUSION EQUATION IN ASYNCHRONOUSLY COUPLED RELAP5-3D». In: *Proceedings of PHYSOR* (2014) (cit. on p. 32).

[26] *preCICE - The coupling library for partitioned multi-physics simulations.* https://precice.org/ (cit. on pp. 36, 84).

[27] ISO. *Information technology – Programming languages – Fortran – Part 1: Base Language*. Standard ISO/IEC 1539-1:2018. Geneva, CH: International Organization for Standardization, Nov. 2018 (cit. on p. 37).

[28] The HDF Group. *HDF5 User ' s Guide*. 2019. URL: `https://portal.hdfgroup.org/display/HDF5/HDF5+User+Guides` (cit. on p. 37).

[29] *The OpenMP API specification for parallel programming*. https://www.openmp.org/ (cit. on p. 37).

[30] *CMake build system*. https://cmake.org/. Accessed: 2020-01-09 (cit. on p. 37).

[31] *Doxygen documentation generator*. http://www.doxygen.nl. Accessed: 2020-01-09 (cit. on p. 39).

[32] J. P. Senecal and W. Ji. «Development of an Efficient Tightly Coupled Method for Multiphysics Reactor Transient Analysis». In: *Nuclear Energy* 103 (2018) (cit. on pp. 50, 84).

[33] Sobolev Vitaly. *Database of thermophysical properties of liquid metal coolants for GEN-IV*. Tech. rep. 2011 (cit. on p. 85).

[34] Walter G. Luscher and J Geelhood Kenneth. *Material property correlations: comparisons between FRAPCON-3.4, FRAPTRAN 1.4, and MATPRO*. Tech. rep. No. PNNL-19417; NUREG/CR-7024. Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 2010 (cit. on p. 86).

[35] Neil E. Todreas and Mujid S. Kazimi. *Nuclear Systems I - Elements of Thermal Hydraulic Design*. Corporation, Hemisphere Publishing, 1990, p. 689 (cit. on pp. 87, 88).

[36] P. R. H. Blasius. «Das aehnlichkeitsgesetz bei reibungsvorgangen in flussigk eiten». In: *Forschungsheft* 64 (1913), pp. 1–41. DOI: `10.1007/978-3-662-02239-9`. URL: `http://link.springer.com/10.1007/978-3-662-02239-9` (cit. on p. 88).

[37] I. Di Piazza, M. Angelucci, R. Marinari, M. Tarantino, and N. Forgione. «Heat transfer on HLM cooled wire-spaced fuel pin bundle simulator in the NACIE-UP facility». In: *Nuclear Engineering and Design* 300 (2016) (cit. on p. 89).

[38] K. Mikityuk. «Heat transfer to liquid metal: Review of data and correlations for tube bundles». In: *Nuclear Engineering and Design* 239 (2009) (cit. on p. 89).