Politecnico di Torino

Master's Degree in Energy and Nuclear Engineering

Development and Validation of Methods for Modeling HTR Reactor Concepts Using the New-Generation Monte Carlo Code TRIPOLI-5

In collaboration with

Supervisors

Dr. Coline LARMIER

Dr. Andrea ZOIA

Prof. Sandra DULLA

Candidate

Enrico PARDINI

MARCH 2026

Summary

Pebble-bed reactors belong to the class of High-Temperature Reactors (HTRs), one of the six Gen-IV nuclear system candidates. A distinct feature of pebble-bed reactors is that their core has a 'double heterogeneity' structure, with TRISO fuel particles randomly distributed within a graphite matrix in the form of spherical pebbles, themselves randomly distributed within the core, and possibly intermixed with graphite pebbles (without fuel particles) acting as moderator.

In this MSc thesis we address the problem of modeling the transport and multiplication of neutrons within pebble-bed reactors using Monte Carlo simulation. In order to achieve our goals, we consider two aspects: sampling the random geometries typical of such class of reactors, and then solving the eigenvalue problem describing neutron propagation in each realization of the underlying reactor model.

As for the geometries, we extend and improve the CEA's tool CASTOR by developing two new models of random spherical inclusions: the Jodrey-Tory (JT) algorithm and the possibility of using a sphere packing generated with an external Discrete Element Method (DEM) software, a class of deterministic codes used for particulate systems, in view of addressing configurations with high packing fractions. The new implementations in CASTOR are compared with the JT model available in the open-source Monte Carlo code OpenMC. Furthermore, specific tests are performed in order to compare the JT model to the sphere packings coming from the DEM code LIGGGHTS, which is coupled to CASTOR.

Concerning neutron transport problems in pebble-bed reactors, we rely on the Monte Carlo code TRIPOLI-5, also developed at CEA (in collaboration with ASNR and EDF), using the geometries sampled by CASTOR (with either JT or DEM models). For this purpose, we build a model for the benchmark configuration of the HTR-10 pebble bed reactor. We sample a collection of random realization and estimate the average behavior of the key physical observables, namely the fundamental eigenvalue and the fundamental eigenmode, using TRIPOLI-5 for each. Results obtained using the JT model and the DEM code LIGGGHTS are compared and a thorough analysis is performed. Where possible, we further perform similar simulations with OpenMC and its native JT model in order to corroborate our findings.

Acknowledgements

Si è concluso un lungo percorso iniziato a fine settembre 2019, fatto non solo di lezioni ed esami, ma di vita vera. Sono stati anni caratterizzati da tanti bei momenti e da diverse nuove esperienze, tra cui in particolare quelle fuori casa a Torino e in Francia. Allo stesso tempo, sono stati anche anni segnati da tanti momenti di difficoltà, in alcuni casi abbastanza impegnativi, che hanno reso però questo traguardo ancora più significativo. È un percorso che porterò sempre nel cuore, dove ho avuto la possibilità di crescere, soprattutto a livello umano.
Per la realizzazione di questa tesi, tengo a ringraziare i miei tutor al CEA, Andrea Zoia e Coline Larmier, per avermi seguito costantemente e avermi fatto apprezzare il nucleare al di fuori dell'ambito accademico. Ringrazio poi la professoressa Sandra Dulla, relatrice di tesi, non solo per aver contribuito a rendere possibile l'esperienza in Francia, ma soprattutto per essere stata presente anche da lontano.
Guardando al percorso in generale, un ringraziamento speciale va ai miei genitori. Mio padre non è stato un semplice sostenitore, ma piuttosto un compagno di viaggio, che mi ha trasmesso quello che oggi ho consolidato come il mio approccio a questo mondo, che trovo estremamente positivo. Mia madre è stata invece la mia vera sostenitrice emotiva, avendo dovuto farsi carico delle mie fasi più negative, ma proprio per questo forse è quella che più di tutti merita questo traguardo. Fondamentale anche la presenza delle mie sorelle, con cui ho condiviso il quotidiano a casa e che sono probabilmente le persone che meglio mi capiscono. Tengo anche a ringraziare i miei nonni, che hanno sempre avuto grande fiducia in me e, più importante, mi hanno trasmesso quel senso di famiglia che porto nel cuore.
Ringrazio poi tutti gli amici, sia quelli con cui il legame è rimasto e si è consolidato nel tempo, rappresentando il mio porto sicuro, sia le nuove amicizie, con cui ho condiviso nuove esperienze di vita. In particolare, ringrazio tutte le persone che mi hanno permesso di vivere momenti di divertimento, risate e spensieratezza in qualunque forma, quelle con cui ho condiviso un pezzo di percorso, che sia a scuola, all'università o durante l'esperienza fuori casa, e quelle che hanno condiviso le loro fragilità o difficoltà, permettendo di creare un senso di unione indelebile.
Un ringraziamento finale va a me stesso: senza alcun tipo di presunzione, mi ringrazio per aver saputo tenere duro quando è servito.

Table of Contents

List of Tables

List of Figures

Acronyms

HTR
: High-Temperature Reactors

DEM
: Discrete Element Method

RSA
: Random Sequential Adsorption

CRP
: Close Random Packing

JT
: Jodrey-Tory

Part I

Physics overview and code implementation

Chapter 1

Introduction

1.1 Fundamentals of nuclear fission physics

Neutron-induced nuclear fission is a reaction involving the nucleus of an atom, which, after having absorbed a neutron, splits into two or more smaller nuclei, called fission fragments, while releasing energy, in the form of radiation and kinetic energy associated to the emitted particles [1]. In addition, also a given number of neutrons, typically two or three, are emitted in this process [1]. An illustration of a fission reaction is provided in Fig. 1.1.

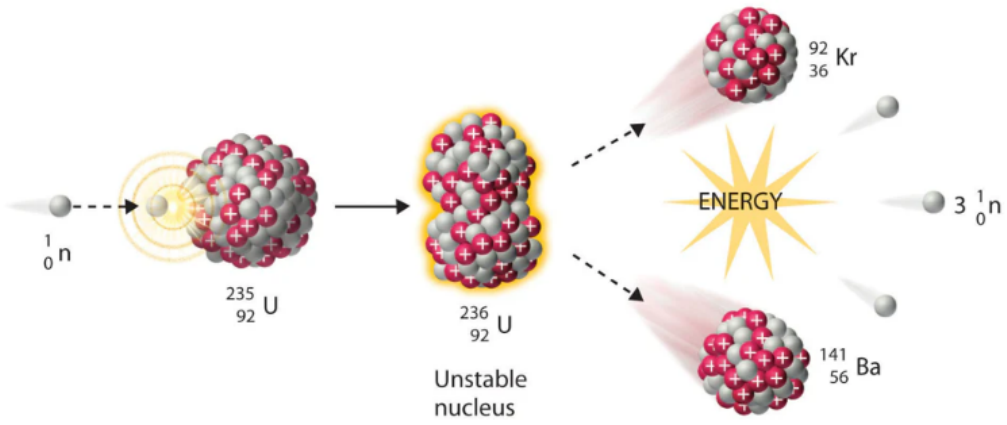


Figure 1.1: Example of a fission reaction [2].

Fission reactions, as all neutron-matter interactions at the microscopic level, are intrinsically stochastic, in that they cannot be predicted *a priori*, but obey a given statistical law. To account for this, nuclear physics defines a quantity known as the cross section. Specifically, the cross section is a quantity proportional to the

probability per unit path of having a given interaction between a neutron and a nucleus [1]. Cross sections depend on the type of interaction considered, the type of material involved in the interaction, and the energy of the colliding neutron.

Not all materials are capable of undergoing nuclear fission: only a limited group of heavy isotopes can. In particular, it is possible to distinguish between [1]:

- Fissile nuclei, which present a significant fission cross section for thermal neutrons (i.e., neutrons carrying low energies), such as ^{235}U.
- Fertile nuclei, which are not fissionable by thermal neutrons, but only by fast ones (i.e., neutrons carrying high energies). These nuclei are also able to undergo the breeding process, through which part of the fertile nuclei is transformed into fissile nuclei by capturing a neutron without leading to fission, as in the case of ^{238}U into ^{239}Pu.

Figure 1.2 shows the trends of the fission cross sections with respect to the energy of the colliding neutron, for three different nuclei, illustrating the differences mentioned above.

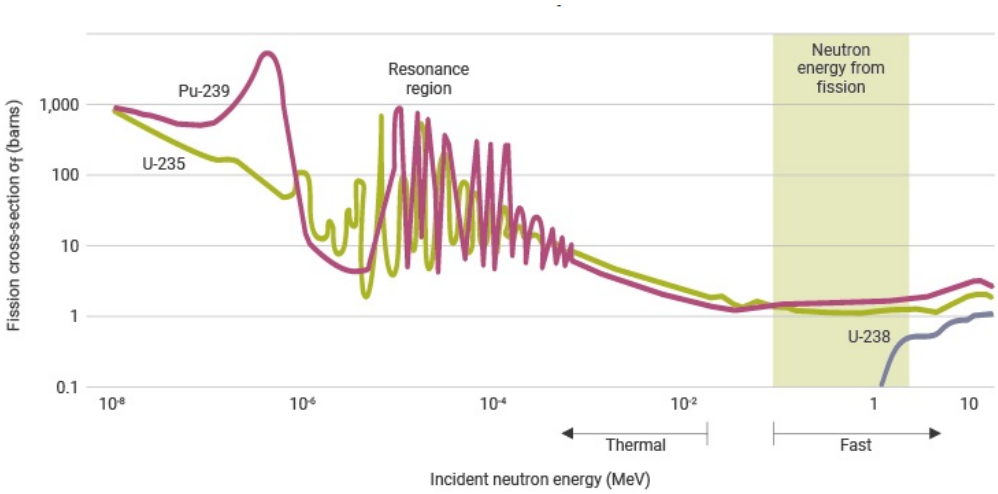


Figure 1.2: Cross sections of three different isotopes [3].

In order to obtain a constant energy production resulting from fission events, it is necessary to have also a constant production of neutrons capable of inducing further fission events. Luckily, for each fission reaction, as anticipated, neutrons are released in an amount which follows a certain statistical distribution, with average around $\nu = 2.5$ for thermal fissions on most fissile nuclides.

The neutrons released can be classified into prompt neutrons, which are emitted simultaneously to fission reactions, and delayed neutrons, which are instead released with a certain delay, following the radioactive decay of some fission product. The

fraction of delayed neutrons over the total number of neutrons emitted by fission is denoted by β; the typical value of this parameter is rather small: for ^{239}Pu it is of the order of 0.3%, whereas for ^{235}U it is of the order of 0.7%.

The neutrons emitted by fission can be employed to achieve the so called neutron chain reactions, where basically the neutrons released by a generation of fissions induce the next generation, and so on [1]. During their life, in addition to fission, neutrons can undergo different interactions with the matter, such as being captured or scattered, and they can also leak from the system boundaries. Each of these phenomena is characterized by a given probability, which again is affected by the material composition and the energy of neutrons (and thus depends on cross sections).

Absorption of a neutron can lead to either fission or capture; capture can be further decomposed as follows [1]:

- Radiative capture, where the nucleus captures a neutron and emits a photon.
- Charged particle emission, where the nucleus captures a neutron and undergoes a transmutation reaction, resulting in a final nucleus chemically different from the initial one, with the emission of a charged particle, such as a proton.

Scattering is a collision between a nucleus and a neutron, which can be elastic or inelastic, leading to the neutron slowing-down and to the modification of its direction of motion [1]. Scattering is the key process by which moderating media (composed of light and weakly absorbing nuclei) slow down neutrons in order to enhance their probability of inducing fission in thermal reactors. In fast reactors, scattering should be in general avoided, in order to minimize the slowing-down process.

Another crucial material utilized in nuclear reactors is the coolant fluid, which is necessary to extract the heat generated by fissions. Indeed, the kinetic energy of the particles emitted by fission processes is rapidly converted into heat because of their Coulomb interactions with the surrounding matter. The heat conveyed by the coolant is used to produce electricity thanks to an appropriate thermodynamic cycle, whose characteristics depend on the type of coolant adopted and on the temperature at which the reactor can be operated [1]. Besides interacting with matter through absorption and scattering, neutrons can also leak from the system boundaries. To minimize neutron leakage and physically contain the core elements, fuel, moderator and coolant are enclosed within a pressure vessel and surrounded by structural and reflecting materials. These structures are designed to withstand high temperatures, neutron irradiation, and mechanical stress, while also shielding the external environment from radiation.

Overall, the entire reactor physics is governed by a competition between neutron production and neutron losses. This continuous balance dictates the life cycle of neutrons in nuclear reactors, as illustrated by the scheme in Fig. 1.3.

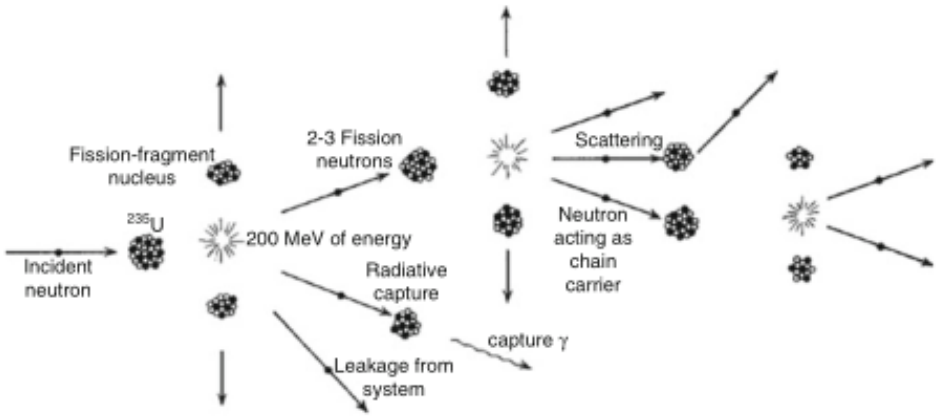


Figure 1.3: Scheme of neutron chain reactions [4].

A crucial parameter to describe the evolution of fission chains is the effective multiplication factor k [5], which represents the ratio between the number of neutrons in two subsequent fission generations. Three regimes can be identified:

- For $k < 1$, which is called sub-critical state, the number of neutrons decreases after each generation, leading to the reduction of the power produced and thus eventually to the reactor shut-down;
- For $k = 1$, which is called critical state, the number of neutrons remains constant: this is the ideal condition of reactor operation, leading to a constant power emission;
- For $k > 1$, which is called super-critical state, the number of neutrons increases with every generation. This condition is required for reactor start-up, but the rate of amplification of the fission chains must be carefully controlled in order to avoid too abrupt increase possibly leading to accidents.

Figure 1.4 illustrates the different trends of the number of neutrons with respect to time, according to the state of the reactor.

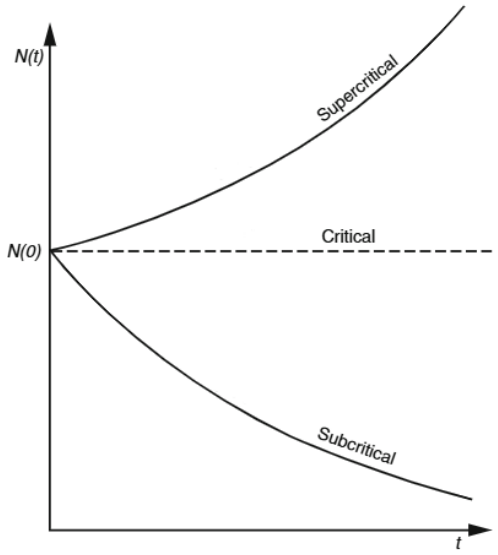


Figure 1.4: Evolution of the neutron population with time for different reactor states [4].

Determining the effective multiplication factor k is one of the most important objectives in the study and analysis of nuclear reactors, since k is directly related to reactor performances and to the associated safety features.

1.2 Fission reactors

Nuclear fission reactors are facilities that allow producing electricity (or more broad thermal power) from nuclear fuel, exploiting neutron-induced nuclear fission reactions. Currently, there are several different types of nuclear reactors, which can be grouped on different generations, from I to IV, their key differences being mostly related to safety features, efficiency, and reactor size. Reactors can also be classified according to the materials employed, such as, mainly, the fuel composition, the type of moderator and the coolant, and consequently also the structural materials. Anyway, not all reactors are conceived in order to produce power: some smaller-size reactors are used e.g. for isotope production (mainly for medical applications), for research in the field of material science, and/or for operator training and teaching.

In the following, a brief review of the main reactor technologies is provided, mainly based on the different materials used for the fuel, the moderator and the coolant [6]:

- Water-Cooled Reactors are based on the utilization of water as both coolant

and moderator. They play a central role in the nuclear industry, since they currently account for the vast majority of all commercial power reactors operating in the world. These reactors can be classified as Light Water Reactors (LWRs), which adopt ordinary water (H_2O), such as Pressurized Water Reactors (PWRs) and Boiling Water Reactors (BWRs); and Heavy Water Reactors (HWRs), which instead employ heavy water (D_2O).

- Gas-Cooled Reactors, and in particular High Temperature Reactors (HTRs), which utilize helium as a coolant and graphite as a moderator. They are currently operating in few countries, such as United Kingdom and China, and they account for a small fraction of all commercial power reactors in the world. Despite this, nowadays the interest in the development of HTRs is increasing, especially due to their high thermal efficiency, the possibility of producing process heat suitable for hydrogen generation and various low-temperature applications, like seawater desalination and district heating.

- Fast Reactors, which are characterized by the absence of the neutron moderator, since fast neutrons are used to sustain the fission chain reactions. The interest in these technologies is growing, since they can provide efficient, safe and sustainable energy, and a considerable waste reduction, allowing for the transmutation of long-lived actinides. The two main reference technologies are Liquid Metal-Cooled Fast Reactors (LMFRs), which exploit liquid metal coolants such as liquid sodium or lead-bismuth eutectic, and Molten Salt Fast Reactors (MSFRs), which use a molten salt mixture both as fuel carrier and coolant. The first category has the advantage of providing an excellent thermal conductivity and allowing high power densities, while the second one benefits from the inherent safety features of molten salt systems.

Figure 1.5 shows the distribution of nuclear reactors nowadays.

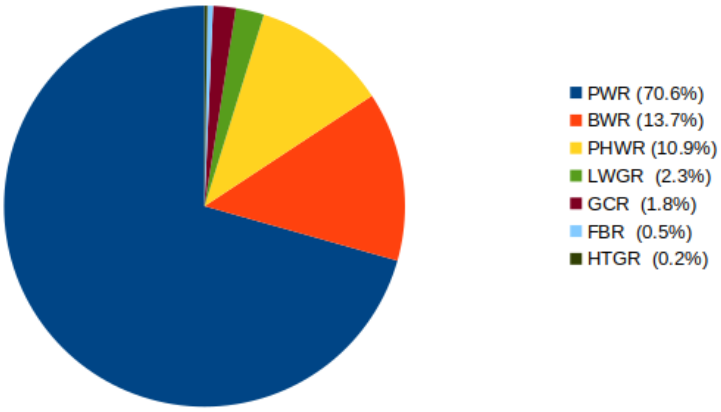


Figure 1.5: Nuclear reactor distribution.

For what concerns the 'generations' of nuclear reactors, a brief overview is given in the following [7]:

1. The first prototypes and experimental reactors built between the 1950s and 1960s, most of which are now decommissioned, represent Generation I.

2. The commercial reactors developed from the 1960s, which today represent the majority of the operating plants, belong to Generation II.

3. Starting from the 1980s, different typologies of reactors, referred as Generation III and III+, were developed, which rely on improved safety systems, often based on passive safety principles, a higher fuel efficiency, and an extended operational life of the plants. There exist different designs, such as EPR, AP1000, APWR and ABWR, that are already in operation or under construction today.

4. Generation IV finally concerns a set of advanced reactor concepts still under development, focusing on sustainability, passive safety, waste minimization, and non-electric applications such as hydrogen production. The concept of small modular reactors, advanced modular reactor, and micro reactors belongs to this generation.

1.3 High-temperature reactors: a brief review

The object of this MSc thesis work concerns High Temperature Reactors. Thus, a more detailed analysis of their characteristics and design is now presented. A short overview of the different HTR concepts is given in Fig. 1.6.

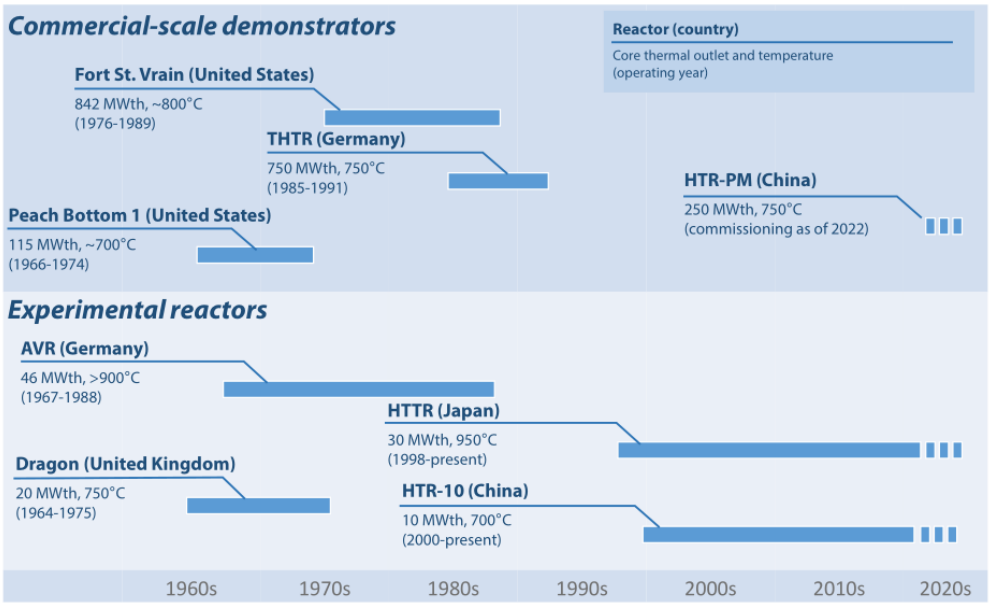


Figure 1.6: Historical panorama of HTR [8].

The first HTR design ever was the 'Daniels Pile', conceived in 1946 in view of optimizing the use of nuclear energy for civil applications, in the wake of the Manhattan Project and the transition from military applications to energy production. Daniels proposed a direct-cycle system using helium as the coolant (which was then also directly used into the thermodynamic cycle), graphite as the moderator, and a core made of structural materials, with enriched uranium carbide as the fuel. His design was conceived as an alternative to LWRs with enriched uranium, which were also under consideration for energy production, but was ultimately discarded due on one side to the Cold War putting all perspectives of civil applications to a rest, and on the other side to the success of PWRs in the naval propulsion program. Nevertheless, Daniels Pile became an important reference point for future HTR development [9].

The first HTR to reach the criticality was the DRAGON reactor in the UK, operating from 1964 to 1977 under the auspices of an international collaboration. It did not generate electricity, but was used exclusively for experimental purposes. In particular, it enabled significant progress for coated particle-fuel and helium-cooled reactor technology. Another crucial experimental reactor was the AVR in Germany, using for the first time the 'Pebble Bed' concept inspired from the Daniels pile ideas [10].

The first HTR to produce electricity was Peach Bottom in Pennsylvania, starting from 1967. It operated successfully for eight years as a prototype reactor to demonstrate the operability of this class of reactors, while being used also for

some tests, especially on advanced coated particles. Two additional significant commercial HTRs were Fort Saint Vrain in Colorado, operating between 1976 and 1989, and THTR-300 in Germany, operating between 1985 and 1989. The latter was a prototype Pebble Bed HTR using TRISO fuel developed thanks to the experience with the AVR [10]. However, their practical application remained quite limited during Generation II, especially due to the higher technological readiness and widespread adoption of LWRs.

In recent years, HTR regained a lot of attention with the emergence of Generation IV reactor concepts proposed by start-ups (both in France, United States and China), in view of their undeniable advantages, such as the inherent safety and high thermal efficiency. In particular, the systems that deserve more attention are the HTR-10, the HTTR and the HTR-PM. The HTR-10 is an experimental reactor, which is China's first major step in the development of modular HTRs; indeed, its main objective is to verify and demonstrate the technical and safety features of modular HTRs and establish an experimental base for the development of heat applications in nuclear processes. It relies on a Pebble Bed fuel, and is based on passive safety and on redundant and diversified components, which allow significantly enhancing the reliability of the system [10].

The HTTR is an experimental reactor, developed in Japan, which reached the outlet coolant temperature of 950 °C in 2004, using TRISO particles into a prismatic fuel assembly, with Generation IV features. As in the previous case, it allowed demonstrating and verifying the advantages represented by this technology [10].

The HTR-PM is a full-scale commercial power plant, which consists of two small reactors (each of 250 MWt) that drive a single 210 MWe steam turbine. Two safety demonstration tests were conducted on the reactor modules, showing that, once the normal energy transfer driven by power was stopped during the tests, the reactors were naturally cooled down, without the intervention of humans or emergency core cooling systems. This is the so-called inherently safe reactor. The main purpose of HTR-PM is to co-generate high-temperature steam up to 500 ℃ and electricity [11].

1.3.1 High-temperature reactors: main features

The historical evolution of all these prototypes and commercial plants has consolidated a specific set of technological choices, like helium as the coolant, graphite as the moderator and ceramic fuel. They represent the core of the HTR technology and will be detailed in the following. Figure 1.7 provides a general scheme for an HTR.

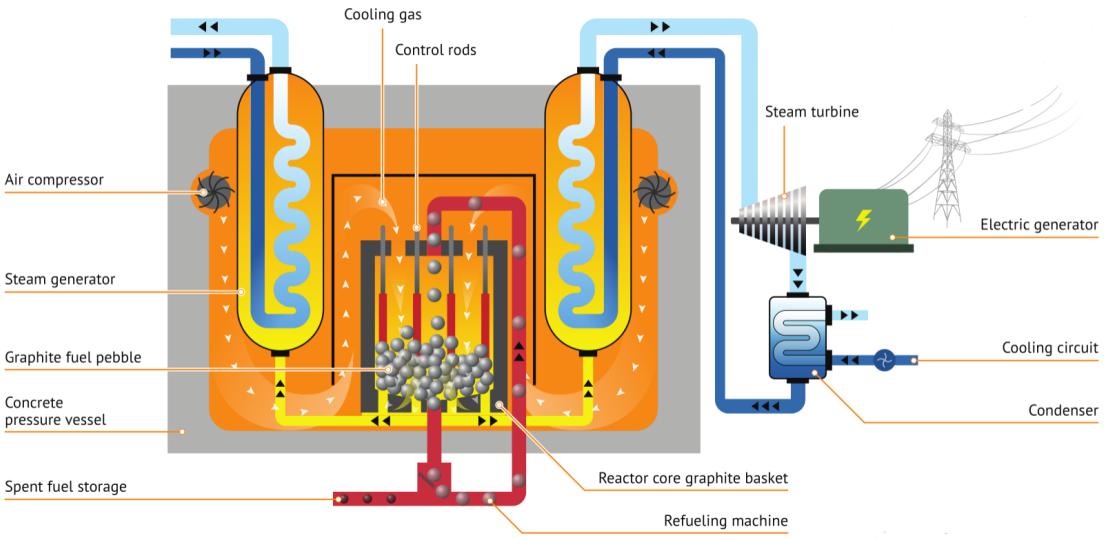


Figure 1.7: HTR scheme [12].

Helium represents a highly reasonable choice as the coolant, as shown by the following favorable characteristics [13]:

- It is chemically inert, meaning that it does not react with the fuel, the moderator or the structural materials, avoiding corrosion issues and consequently enhancing the longevity of reactor components and the system safety.
- It remains stable at high temperatures, enabling operation at high thermal levels, without issues such as cavitation and boiling problems.
- It allows the implementation of a direct thermodynamic cycle (Brayton cycle), which, combined with the high operating temperatures, improves the global system efficiency, even though indirect steam cycles are still widely adopted in near-term deployments (such as the HTR-PM) due to higher technological readiness.
- It has a low neutron activation, increasing the radiological safety of the plant and reducing the shielding requirements.

On the other hand, helium also presents some drawbacks: it has a low specific heat capacity, requiring a high flow rate to be effective (thus a large occupied volume), needs sophisticated turbines, due to the low density, and is relatively rare and expensive. Furthermore, the small atomic size of helium makes it prone to leaking through structural micro-porosities: this characteristic requires very tight manufacturing tolerances, significantly increasing the overall costs of the facility.

Graphite represents a highly reasonable choice in HTR as the neutron moderator. This can be assessed by the following favorable characteristics [14]:

- It has a low neutron capturing cross section, ensuring an efficient moderation without a significant capture of neutrons, which are needed for the fission chain reactions.
- It maintains structural and dimensional stability even at very high temperatures (up to ~ 2500 °C), making it adapted to reactors operating well above the temperatures of the water cooled systems.
- It is characterized by high thermal conductivity, ensuring the improvement of heat removal and the passive safety in accidental scenarios.
- It is chemically compatible with the helium, avoiding corrosion issues or undesired chemical reactions under normal operating conditions.

Nevertheless, it is necessary to underline also some limitations of graphite: it is sensitive to neutron irradiation, which causes dimensional changes, such as swelling and shrinking; it can react with oxygen at high temperatures, leading to oxidation and potential fire risks; and it requires a larger core volume compared to water-cooled designs to attain the same moderating power.

The combined characteristics of helium and graphite result in reactors having significantly larger core volumes than traditional LWRs.

For what concerns the fuel, modern HTR fuels are based on TRISO particles (TRi-structural ISOtropic particle fuel) [15]. Each TRISO particle is made of a uranium, carbon and oxygen fuel kernel, which is encapsulated by three layers of carbon-based and ceramic-based materials that prevent the release of radioactive fission products. TRISO particles are extremely small (approximately 1 mm in diameter) and highly robust: they are structurally more resistant to neutron irradiation, corrosion, oxidation and high temperatures than traditional reactor fuels. This allows them to retain fission products under all normal and accident reactor conditions, effectively eliminating the risk of core meltdown in commercial HTRs. Figure 1.8 illustrates the structure of TRISO particles.

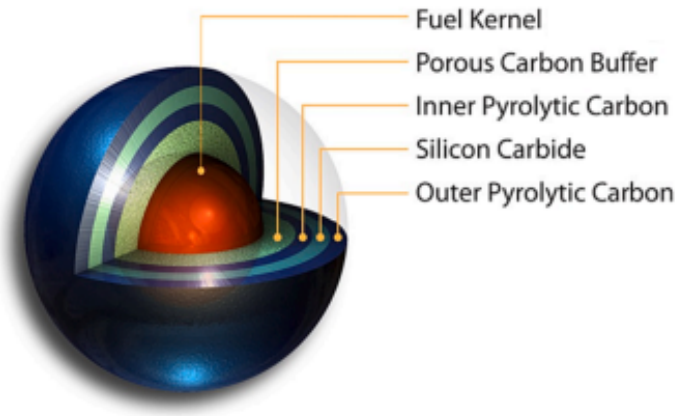


Figure 1.8: Structure of TRISO particles [16].

The fuel can be arranged in different ways within the core; in particular, it is possible to distinguish between two different fuel compositions [8]:

- Pebble Bed fuel, which consists of spherical fuel elements, called pebbles, each containing thousands of TRISO particles encapsulated in a graphite matrix, as represented by Fig. 1.9;
- Prismatic fuel, which consists of prismatic-shaped fuel blocks that contain TRISO particles embedded in a graphite matrix, as represented by Fig. 1.10.

The former has the main advantage of being continuously refueled during operation, avoiding the necessity to stop the reactor, and of having a lower power density with a higher surface area for heat transfer, improving the passive safety. On the other hand, the latter has the main advantage of an easier insertion of control rods and of a simpler modeling and analysis, due to the smaller degree of stochasticity which characterizes the geometry.

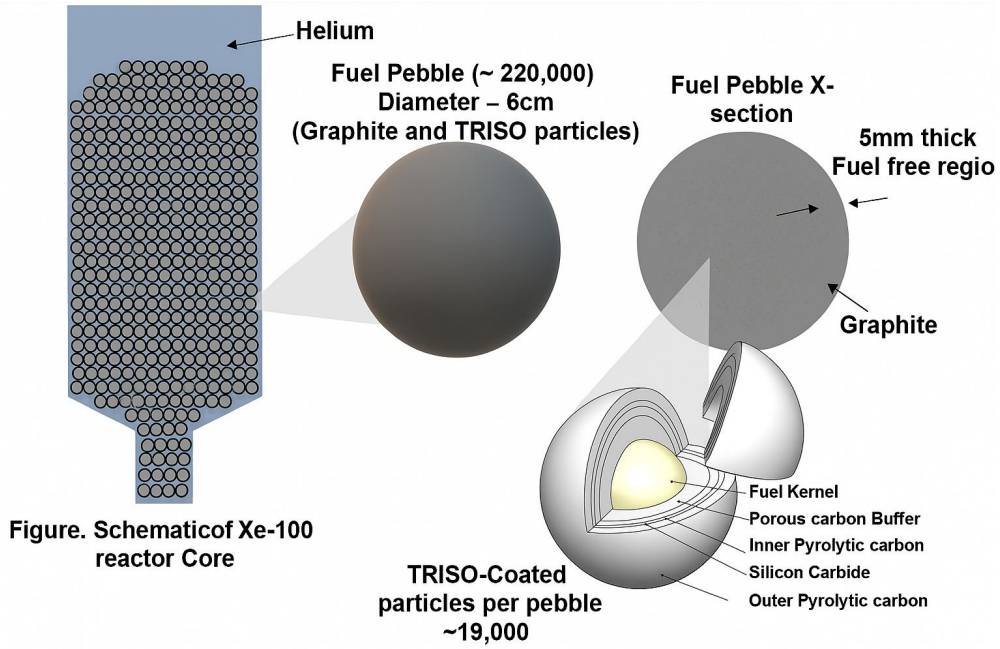


Figure 1.9: Pebble Bed Fuel. [17]

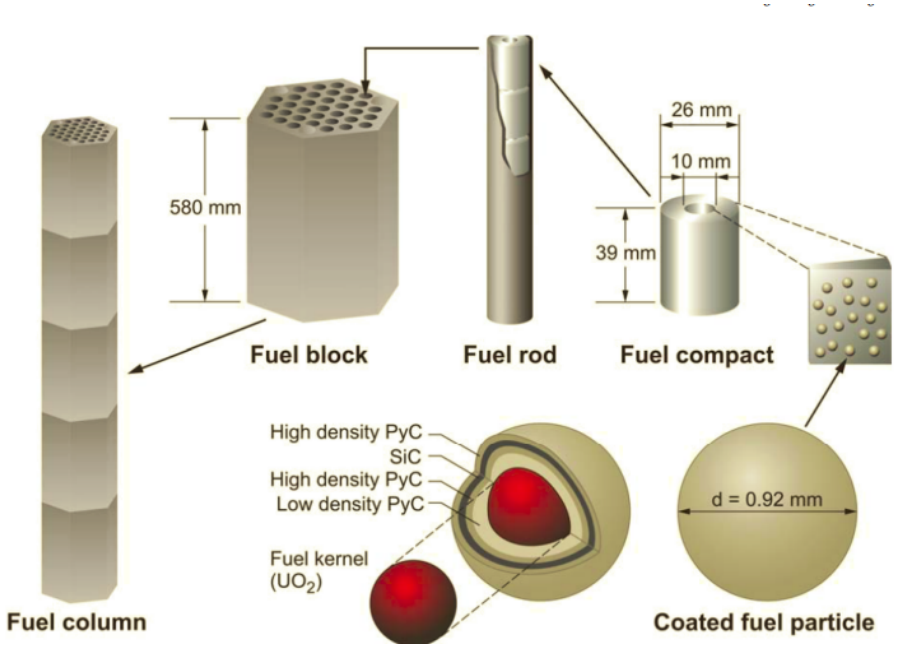


Figure 1.10: Prismatic Fuel. [18]

1.3.2 High-temperature reactors: advantages and drawbacks

As shown in the previous section, HTRs exhibit peculiar characteristics that make them a compelling choice in an energy scenario based on diversification. Their advantages can be summarized as follows:

- First of all, thanks to the use of helium as the coolant, which allows higher outlet temperatures compared to other reactors, and to the possibility of directly using helium in a thermodynamic cycle, the overall efficiency of these reactors is significantly higher, even beyond 50% for Generation IV, corresponding to outlet temperatures around 1000 °C [19].

- Another crucial advantage is their intrinsic and inherent safety, which is the result of several aforementioned material choices: the absolute retention of fission products by TRISO particles up to extreme temperatures [15], the chemical inertness of helium [13], and the massive thermal inertia of graphite [14]. In addition, for the new generation of reactors, the use of passive safety systems allows the reactor to handle transients for several days without any operator intervention.

- Furthermore, thanks to the high temperatures that can be attained in HTRs, they can be used not only for electricity generation, but also for high-temperature heat supply, which is helpful for de-carbonization. In particular, HTRs appear to be particularly suitable in the following scenarios: electricity production, cogeneration, seawater desalination, oil recovery and refining, and the production of hydrogen, ammonia, and aluminum [8].

- Finally, many HTRs belonging to generation IV incorporate modularity, allowing greater adaptability to each site-specific needs and faster construction times [13].

For what concerns the drawbacks, one of the main disadvantages of these reactors is the size of the core and the vessel. This is primarily due to the use of helium, which is much less effective than water in removing heat, leading to the need of larger coolant channels and a larger reactor volumes. Similarly, graphite requires a larger volume than water to be effective in slowing down neutrons, due to its lower moderator efficiency [19].

Another relevant issue is the cost of the plant. Helium is considerably more expensive than water to be procured and managed, and the production of TRISO particles is technologically complex and costly. Concerning the latter, their reprocessing is an issue, due to their complex structure, limiting the possibility

of achieving a closed fuel cycle [20]. For what concerns the graphite, the long-term effects of neutron irradiation must be taken into account, which leads to non-negligible dimensional changes and structural degradation over time [14].

Finally, it is mandatory to consider also that the overall industrial and operational experience with helium and graphite systems is still limited compared to the extensive knowledge accumulated over decades with water-cooled reactors.

In conclusion, on the basis of the previous analysis, it can be assessed that HTRs, despite some challenges, represent a truly promising solution for the future, especially within a scenario based on energy diversification. For this reason, these reactors deserve to be studied with great attention and technical accuracy.

1.4 Simulating HTRs: tools and methods

In order to properly investigate the behavior of a nuclear reactor, it is essential to be able to accurately describe neutron transport and multiplication within the core. For this purpose, Monte Carlo methods are widely used in reactor physics, since they allow modeling complex geometries and heterogeneous media, while integrating the whole neutron and photon interaction laws provided in nuclear data libraries, with almost a complete lack of approximations induced by discretization in either geometry or energy [21].

In particular, Monte Carlo methods rely on the stochastic sampling of particle histories, based on the probability distributions for their interactions with matter, accounting for phenomena such as fission, capture, and scattering. Such simulations allow for the estimation of physical quantities of interest, such as neutron flux, reaction rates, or power distribution, represented by the average of the outcomes over a large number of particle histories [21]. The associated statistical uncertainty is also estimated within the same simulations, which allows quantifying the reliability of the obtained average observables.

One of the key inputs to Monte Carlo simulations is the geometry model of the reactor. This represents a significant challenge for HTRs reactors, particularly for the Pebble Bed designs, since they are characterized by the so called 'double heterogeneity' cores: random spheres (TRISO particles) dispersed within a graphite matrix into random spheres (pebbles), packed by gravity into the reactor core. Consequently, in order to model these systems it is necessary to be able to describe the transport and multiplication of neutrons in 'random media', so called because the materials present at a given point in the fuel (the random spherical inclusions) are known only in probability [22, 23].

To face this kind of problem, CEA (Commissariat à l'énergie atomique et aux énergies alternatives) has recently developed two different approaches [24, 25, 26, 27]:

- The 'explicit' modeling (quenched disorder approach), where the sampling of a collection of random media is achieved from the underlying statistical distribution. The transport equation is then solved for each medium, and averages are taken with respect to replicas. This allows obtaining an accurate description of neutron propagation, at the expense of a very long simulation time. For this purpose, CEA has recently developed a tool, called CASTOR [28], to sample a collection of stochastic media of several kinds.

- The development of 'homogenized' models (annealed disorder approach), in which the presence of randomly distributed spheres in the double-heterogeneity cores is taken into account on average, neglecting the memory effects induced by spatial correlation; this saves simulation time at the expense of a loss of accuracy. The Levermore-Pomraning (LP) and the Chord Length Sampling (CLS) models belong to this category [29, 30, 31, 32]. Observe that several computational strategies, such as the Limited CLS (L-CLS) [33, 34], the Poisson-Box Sampling (PBS) [35] or the Local Realization Preserving (LRP) [36], have been developed in order to partially mitigate this issue, their common feature being a semi-explicit representation of the memory effects that allows attaining higher accuracy with respect to standard annealed-disorder models.

In the present MSc thesis, only the first approach is investigated.

1.4.1 Sampling random media: CASTOR

Concerning the explicit modeling of random media, as anticipated, we will rely on the CASTOR tool [28]. CASTOR, which stands for Construction and Analysis of STOchastic Realizations, has been developed in order to sample three-dimensional random media for particle transport applications and to assess their statistical features.

In particular, three models have been implemented so far: Markov media, based on either isotropic Poisson tessellations or Poisson-Box tessellations, and spherical inclusions, which are the ones considered for the Pebble Bed geometry.

Spherical inclusions can be realized within different container shapes: spherical, cylindrical and box-type. In addition, it is possible to reproduce both mono-dispersed and poly-dispersed spheres (respectively, with a constant or with a variable radius), where the latter can be characterized by a linear or an exponential distribution of the radii.

For what concerns the generation of this type of random media, the legacy version of CASTOR that was available before the beginning of this work allowed for the modeling of non-overlapping three-dimensional spheres, randomly placed into a background matrix, resulting in binary stochastic media. In particular, CASTOR relied on the Random Sequential Adsorption (RSA) algorithm [28], which is based

on sampling uniformly in the matrix the position of each sphere, sequentially, rejecting them in case of an overlap with the previously placed spheres or the boundaries of the domain. This algorithm is shown in Fig. 1.11.

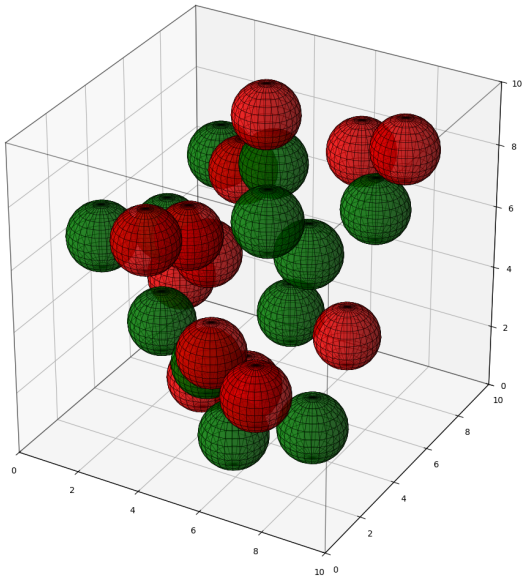


Figure 1.11: Application of Random Sequential Adsorption algorithm for some spheres. Green spheres are accepted, red ones are rejected.

The 'Naive RSA' scales as $O(N^2)$, with N being the number of sampled spheres, due to the constraint of the non-overlapping-sphere. Better performances can be obtained using a fictive Cartesian mesh superposed to the domain, which allows reducing the computational time, with overlaps being verified only for those spheres falling (even partially) within the same mesh cell [28].

Figures 1.12 and 1.13 illustrate realizations of RSA sphere packings sampled with CASTOR, plotted using either ParaView [37] or the visualization tool T4G, developed at CEA [38].

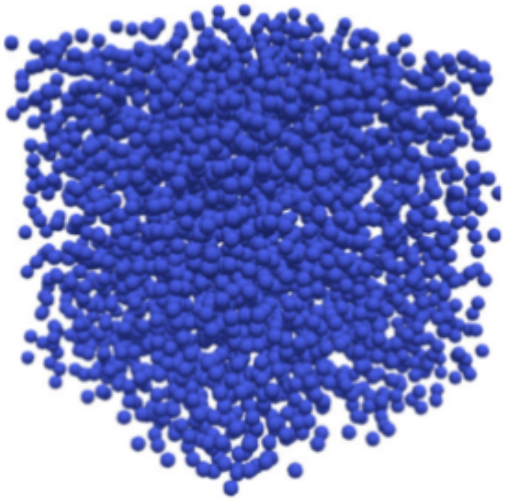


Figure 1.12: CASTOR geometry visualized with Paraview.

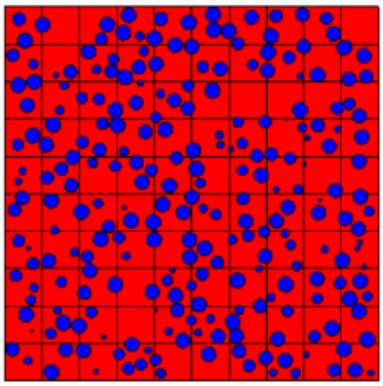


Figure 1.13: CASTOR geometry visualized with T4G.

The RSA algorithm presents a limitation for the maximum achievable packing fraction (i.e. the ratio between the volume occupied by the spherical inclusions and the container) at about 0.38. Thus, using RSA it is possible to realistically reproduce the distribution of TRISO particles into the pebbles, which have a typical packing fraction not higher than $0.3 \sim 0.4$ [39], but the distribution of the pebbles within the core is not achievable, since their typical packing fraction is around $0.6 \sim 0.64$ [40]. For the needs of this MSc thesis, we have thus expanded the perimeter of applicability of CASTOR by developing new features enabling the simulation of random spherical inclusions with high packing fraction. These developments will be described in the following chapters.

1.4.2 TRIPOLI-5

As for particle transport, in the context of this work we rely on the TRIPOLI-5 Monte Carlo code [41], which has been co-developed since 2022 by CEA and IRSN, in collaboration with EDF. The main goal of TRIPOLI-5 is to achieve massively parallel simulations on hybrid CPU/GPU computing architectures. TRIPOLI-5 takes advantage of the experience coming from previous investigations conducted on the PATMOS mini-app, concerning the portability of particle-transport algorithms in High Performance Computing (HPC) environments. At the moment, the main focus of this tool is on reactor physics applications, including multi-physics feedback for stationary and non-stationary configurations. In the long run, TRIPOLI-5 will eventually cover a larger range of applications (encompassing radiation shielding and nuclear instrumentation) and thus supersede the current-generation Monte Carlo codes TRIPOLI-4 [38], developed at CEA, and MORET6, developed at IRSN.

The geometry engine of TRIPOLI-5 is called AGORA, whose modeling capabilities have been designed with a multi-nested description, based on the Constructive Solid Geometry (CSG) paradigm [42]. Thanks to AGORA, TRIPOLI-5 allows building geometry models for reactors by defining planes, cylinders and cones, which in turn allow to define, by the intersection of regions delimited by them, many different shapes. Each shape contains a volume, in which it is possible to subsequently assign a filler, that can be a material composition (homogeneous in the volume) or a 'universe', which is the union of different volumes previously generated. The presence of universes allows creating hierarchical structures, where portions of the geometries are explored by the tracked particles only when the particle is found at the correct 'level' of the structure. AGORA also allows importing external geometries or volumes, which must be generated by external codes in the internal format (.agora files): this feature is particularly useful to import random geometries sampled by CASTOR.

Once a geometry model is created, it is possible to perform the neutron-transport simulation. For criticality simulations as considered in the following, TRIPOLI-5 follows the standard power iteration algorithm. An initial source with a given number of neutrons is sampled, then a given number of inactive cycles (fission generations) are simulated, until convergence is achieved. Then, a given number of active cycles is simulated, over which the required tallies are accumulated.

1.5 Aim of the thesis

The main objectives of the present MSc thesis are the following.

1. The first goal is to develop new algorithms and methods for the spherical inclusion media in CASTOR in order to achieve the packing fraction required

for HTR applications. This is accomplished in two steps. In Chapter 2, we begin by implementing in CASTOR the Jodrey-Tory algorithm [43], which is already available in OpenMC under the name of Close Random Packing (CRP) method [44]. The Jodrey-Tory algorithm allows sampling the 'equilibrium' configuration of a random sphere packing with packing fraction up to 0.66, without having to simulate the dynamics of the single sphere under the action of dynamic forces. For verification, the new model is then compared to the results of OpenMC in Chapter 2. The second step concerns the development of a new coupling between CASTOR and the independent open-source Discrete Element Method (DEM) code LIGGGHTS [45], as detailed in Chapter 3, which enables the sampling of realistic sphere packing configurations obtained by numerically integrating the action of the dynamic forces (gravity, friction, collisions, wall reflection, etc.) acting on the pebbles. The configurations obtained using the LIGGGHTS code are finally compared to those obtained using the Jodrey-Tory algorithm in Chapter 4.

2. The second goal is to use the new tools developed in CASTOR in order to simulate neutron transport in realistic HTR configurations with TRIPOLI-5. For this purpose, we have first further extended CASTOR in order to implement a 'double heterogeneity' model for mixtures of graphite and fuel pebbles: random spheres (pebbles) containing random spheres (TRISO particles), with multi-shells structures for both. This work is detailed in Chapter 5. Based on this model, we finally address the simulation of a benchmark configuration, namely the HTR-10 model, whose specifications are available through the IRPHEEP database [46, 47]. A preliminary analysis is carried out in Chapter 6; a subsequent set of TRIPOLI-5 simulations with the Jodrey-Tory sphere packing model and with the LIGGGHTS sphere packing model is carried out in Chapters 7 and 8, respectively. Whenever possible, an additional set of OpenMC simulations is used as a reference.

Chapter 2

Close random packing

2.1 The Jodrey-Tory algorithm

As mentioned in the previous chapter, the Random Sequential Adsorption (RSA) algorithm adopted in the legacy version of CASTOR is not able to generate spherical inclusion geometries with packing fractions higher than 0.38 [28], which represents a limitation for reproducing a realistic pebble bed geometry whose typical packing fraction may be in excess of about 0.6.

Currently, in OpenMC [44], an open source Monte Carlo particle transport code mainly developed at the Massachusetts Institute of Technology and Argonne National Laboratory, USA, a different method has been implemented for this purpose, namely the Jodrey-Tory algorithm [43] (under the name of Close Random Packing (CRP) method), which is able to sample geometries with a packing fraction slightly above 0.66.

The Jodrey-Tory algorithm works as follows. A given number of points, corresponding to the target number of spheres required for the geometry, is generated randomly inside the container: each point corresponds to the center of an inner and an outer sphere. The inner diameter is set to the minimum center-to-center distance between any pair of spheres, for each iteration. Instead, the outer diameter, which is exploited to identify overlaps between spheres in the algorithm, is initially set to the arbitrary value that would lead to a packing fraction equal to 1, and is progressively reduced at each iteration [43].

The algorithm is based on two different loops. At each outer iteration, a list containing pairs of overlapping spheres is generated. Each sphere can appear only once in the list to avoid conflicting movements during the inner iterations, meaning that only the most severe overlaps are considered. At each inner iteration, the biggest overlap of the list (i.e., the overlap relative to the closest spheres) is eliminated, by moving the two spheres by an equal distance along the line joining

their centers, until these centers are separated by one outer diameter, while ensuring that the spheres remain inside the container. Subsequently, the list is updated, the inner diameter is modified, as described above, and the outer diameter is slowly reduced, according to

$$d_{\text{out}}^{(i+1)} = d_{\text{out}}^{(i)} - \left(\frac{1}{2}\right)^{j} d_{\text{out},0}\frac{k}{N}. \tag{2.1}$$

Here k is the contraction rate, which is 10^{-3} by default and can be modified, N the number of spheres, i the generic iteration index, $d_{\text{out},0}$ the nominal outer diameter, and j is computed from

$$j = \lfloor -\log_{10}(\xi_{out} - \xi_{in}) \rfloor, \tag{2.2}$$

where $\lfloor \cdot \rfloor$ denotes the greatest integer smaller or equal to its argument, and ξ_{out} and ξ_{in} are respectively the packing fractions of the spheres related to the outer and to the inner diameter [43].

The inner iterations continue until the list is empty. The algorithm is stopped under the following conditions.

- If the inner diameter exceeds the real one, which is associated with the nominal density. This indicates that the minimum center-to-center distance between any pair of spheres is bigger than the real diameter, meaning that the nominal density has been reached without overlaps of spheres.

- If the outer diameter approaches the inner one or the list of spheres built in a given outer iteration is already empty. This indicates that the algorithm possibly is not successfully terminated, and thus it can be necessary to reduce the packing fraction or the contraction rate, in order to achieve the target geometry.

A scheme of the Jodrey-Tory is illustrated in Fig. 2.1. In the current implementation available in OpenMC, the sphere packing is actually obtained by using a combination of RSA and CRP, the former being used for the initialization process (further details are provided in the next section). Indeed, RSA performs better than CRP for lower packing fractions, but it becomes prohibitively slow as it approaches the theoretical packing fraction limit at about 0.38 [48].

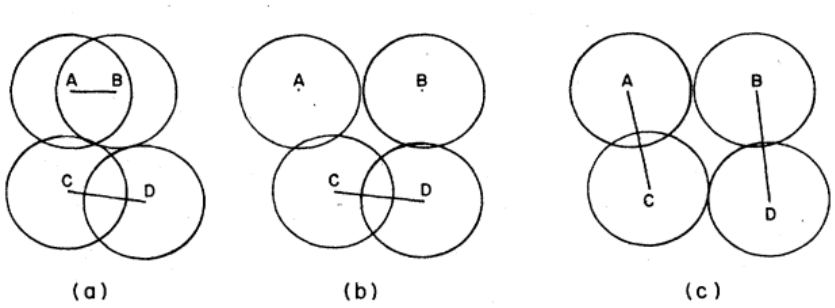


Figure 2.1: Jodrey-Tory algorithm illustration. Stages in the elimination of overlaps in a four-disk system: (a) The worst overlap corresponds to the shortest distance AB. Pair AC is not in the list because the worst overlap involving A is with B and the worst involving C is with D. (b) Once the overlap between A and B is eliminated, the worst overlap corresponds to the pair CD. Pairs AC and BD are not in the queue because the worst overlap involving C is with D and vice versa. (c) With the elimination of the overlap between C and D, pairs AC and BD are placed in the list [43]

.

2.2 Implementation of the Jodrey-Tory algorithm in CASTOR

For the purposes of this MSc thesis work, the Jodrey-Tory (CRP) algorithm was implemented in CASTOR. The overall strategy is the following.

- If the desired packing fraction is below the RSA upper limit, only RSA is used (since it is known to be faster than CRP).
- If a higher packing fraction is required, spheres with a radius smaller than the desired final radius (and therefore with a smaller packing fraction) are initialized within the domain using RSA. This initial configuration of sphere centers is then used as a starting point for CRP.

Due to confidentiality agreements and proprietary rights, it is not possible to disclose the code itself. Therefore, rather than focusing on the specific coding details, we highlight the most relevant aspects of our implementation.

One of the main challenges was to make the new CRP strategy coexist with the legacy RSA implementation in CASTOR. In particular, the handling of the mesh that is used in CASTOR to accelerate the RSA was preserved and successfully reused also for CRP, to speed up convergence. During the geometry generation phase, the CRP relies on identifying sphere pairs with the highest degree of overlap

at each iteration. To handle this computationally intensive task efficiently, we implemented a KD-Tree structure. Operating with a complexity of $O(N \log N)$, this spatial partitioning stage represents the dominant component of the execution time. Once the generation process is complete, it is crucial to verify that the resulting spheres are strictly non-overlapping. For this purpose, we developed and compared three distinct verification methods: a brute-force (naive) approach used as a validation baseline, a mesh-based acceleration technique, and a KD-Tree implementation.

The outcome of this work was the implementation of a new 'constructor' in CASTOR, enabling the sampling of random sphere packings via the Jodrey-Tory algorithm, along the lines detailed above.

2.3 Performance

To thoroughly verify the performance of the CRP as implemented in CASTOR, we compared the sampled sphere packings with those generated by OpenMC, and assessed the corresponding computational costs. In particular, we considered a box shape container of fixed size, with the packing fraction set at 0.4, 0.5 and 0.6, and the number of spheres ranging from 10^3 to 10^6 (achieved by adjusting spheres radii). Being the CRP inherently stochastic, we had to average the sought observables over several realizations, and compute the associated dispersion. For this purpose, we used the formulas

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{2.3}$$

for the average,

$$s = \sqrt{\frac{1}{n(n-1)} \sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{2.4}$$

for the standard error of the mean, and

$$e = \frac{s}{\bar{x}} \cdot 100 \tag{2.5}$$

for the relative standard error of the mean. Here n is the number of samples and x_i is the value obtained for the i-th sample. To ensure reliable results, we performed ten replicas for each configuration, except for cases with a 0.6 packing fraction and a high number of spheres, where statistical dispersion naturally decreases.

To begin with, we performed a set of CRP realizations using OpenMC simulations, where the mesh sizes are computed internally by the code, followed by a second set of CRP realizations using CASTOR, where the mesh is set by the user, to make it comparable to OpenMC ones for a fair comparison. Results are illustrated in Figs. 2.2, 2.3 and 2.4, where each point represents the average over the different simulations, and the corresponding error bar the relative standard error.

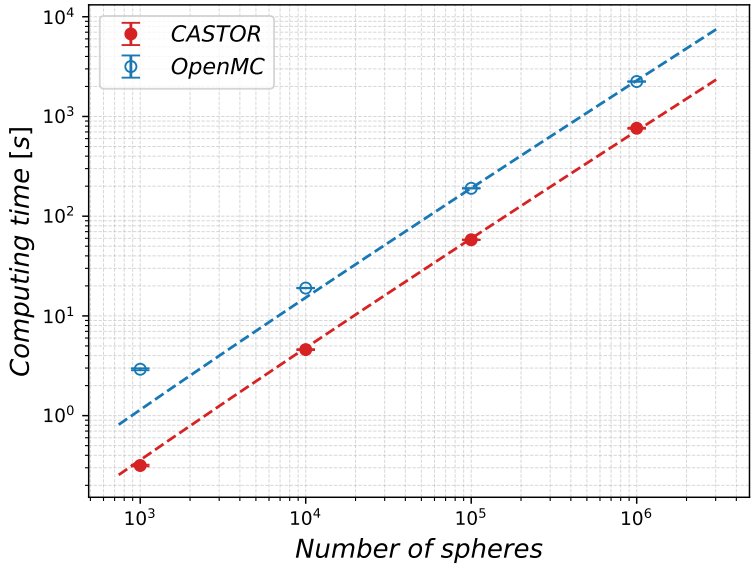


Figure 2.2: Comparison between CASTOR and OpenMC performing the CRP method, for a packing fraction of 0.4. The dashed lines represent the $O(N \log N)$ scaling, which is the expected theoretical slope, particularly for large systems.

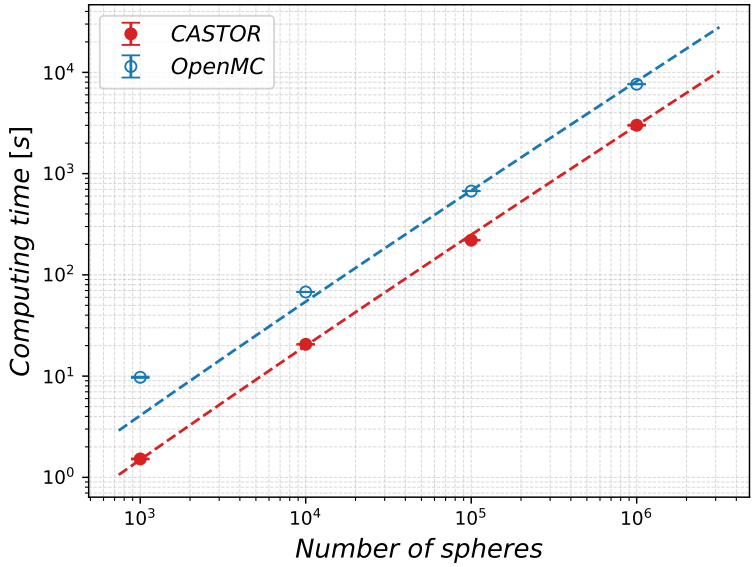


Figure 2.3: Comparison between CASTOR and OpenMC performing the CRP method, for a packing fraction of 0.5. The dashed lines represent the $O(N \log N)$ scaling, which is the expected theoretical slope, particularly for large systems.

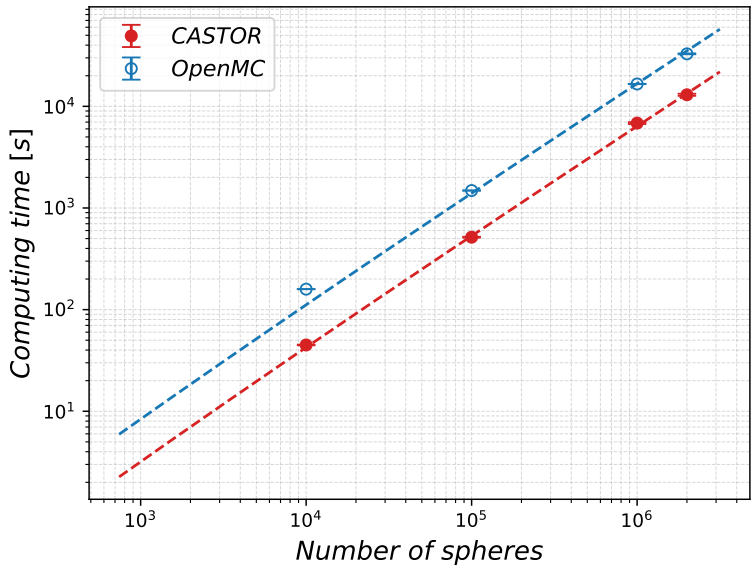


Figure 2.4: Comparison between CASTOR and OpenMC performing the CRP method, for a packing fraction of 0.6. The dashed lines represent the $O(N \log N)$ scaling, which is the expected theoretical slope, particularly for large systems.

As shown in the plots, the simulation results for large N align closely with the predicted $O(N \log N)$ scaling. The slight variations can be attributed to fixed computational costs and the asymptotic convergence of the algorithm as the number of spheres increases.

For a packing fraction of 0.6, both codes fail to sample a configuration with only 10^3 spherical inclusions due to boundary effects. Consequently, an additional simulation with 2×10^6 spheres was performed to maintain four data points for this packing fraction. As mentioned, for the cases of packing fraction of 0.6 with 10^6 and 2×10^6 spheres, respectively eight and seven replicas were sampled, instead of ten. The results for the average values of the sampling times can be considered reliable by looking at the relative standard error, which is always lower than 3%, justifying the chosen number of replicas, as shown by Tabs. 2.1 and 2.2.

Number of Spheres	pf 0.4	pf 0.5	pf 0.6
10^3	1.90%	0.92%	-
10^4	1.07%	0.69%	0.67%
10^5	0.42%	0.85%	0.90%
10^6	1.16%	1.13%	2.04%
2×10^6	-	-	2.33%

Table 2.1: Relative Standard Errors for CASTOR results.

Number of Spheres	pf 0.4	pf 0.5	pf 0.6
10^3	2.24%	1.40%	-
10^4	0.43%	0.19%	0.17%
10^5	0.32%	0.21%	0.09%
10^6	0.37%	0.36%	0.31%
2×10^6	-	-	0.96%

Table 2.2: Relative Standard Errors for OpenMC results.

Overall, this analysis shows that CASTOR is able to apply the Jodrey-Tory algorithm across all the conditions (i.e., the same set of parameters) that were tested with the CRP of OpenMC. Comparing the trends of the execution times, it is possible to conclude that CASTOR is faster than OpenMC for all the configurations. However, we should keep in mind that OpenMC is penalized due to its Python-based user interface, whereas CASTOR is implemented in C++, allowing for a more direct and efficient memory management during geometry generation.

Chapter 3

Discrete Element Method

3.1 The LIGGGHTS code

The Discrete Element Method (DEM) is one of the most powerful and widely used numerical methods aimed at modeling the dynamics of particulate systems. In this method, the trajectories of particles are computed by tracking both their translational and rotational movements, accounting for external forces, such as gravity, and for their interaction between each other and with domain boundaries [45]. Particles are treated as soft or hard spheres that interact when they overlap. The contact forces (both normal and tangential) for each particle pair are calculated with a suitable force model relying on their overlap distance, which represents the deformation between the particles. Physics is solved discretely in the time domain, where at each simulation step the net force is calculated on each particle and Newton's equations of motion are solved. The DEM approach is characterized by a high computational cost, but also by a high reliability, making it a common validation tool for faster codes.

Among the different DEM implementations, LIGGGHTS [49], which stands for LAMMPS Improved for General Granular and Granular Heat Transfer Simulations, is one of the most exhaustive, flexible and widely used open-source tools. It is very suitable for the generation of Pebble Bed geometries; in fact, it has already been used in such applications, as reported, for instance, by Ref. [50, 51]. In this Chapter, we present and test the LIGGGHTS tool, in view of obtaining a faithful model of Pebble Bed geometries, to be used in combination with CASTOR.

3.1.1 Input file generation

To perform LIGGGHTS simulations, it is necessary to prepare an input file, to define the geometry of the considered system and the physics which is behind the filling process of pebbles into the 'container'. The container region is generated

in accordance with the specifications of the reactor of interest, generally applying reflecting boundary conditions for its walls, in order to reproduce the sphere-wall interactions.

Each pebble is represented by the sphere atom style defined in LIGGGHTS. The physical data for the fuel and graphite pebbles of HTR-10 are based on Ref. [50, 51], and are shown in Tab. 3.1.

Parameter	**Value**
Radius	3 cm
Density	1850 kg/m^3
Young's Modulus	10^9 Pa
Poisson's Ratio	0.2
Restitution Coefficient	0.9
Friction Coefficient	0.2

Table 3.1: Pebble data for LIGGGHTS simulations.

Regarding the filling process, the idea is to randomly generate one sphere at a time into an insertion region, defined as a cylinder located at the center of the upper part of the container, with a diameter of 8 cm (the pebble diameter is 6 cm) and a height equal to the diameter of the spheres with the addition of a small tolerance. The initial velocity of the spheres, downward directed, changes between 1 m/s and 1.1 m/s uniformly, and is subsequently increased by gravity. In this way, the insertion process accounts for a certain randomness for both initial position and velocity, which is governed by a 'seed' [49].

The interval between two consecutive pebble insertions is chosen to avoid any possible overlap between particles, i.e., the code waits an amount of time sufficient to ensure that each particle completely leaves the insertion region, before the subsequent insertion. After each insertion, the sphere drops into the container and the consequent interaction with the other pebbles and with the container boundaries are simulated. Figure 3.1 illustrates the filling process, for a given step. The inserting process continues until a fixed number of spheres, according to the considered configuration, are inserted, after which the simulation runs for an additional amount of time, allowing the system to reach an 'equilibrium' state. The equilibrium is considered as achieved when the total kinetic energy of the spheres falls below an arbitrary threshold, which is specifically tailored in accordance to the number of spheres, thus the total mass, inserted in the system.

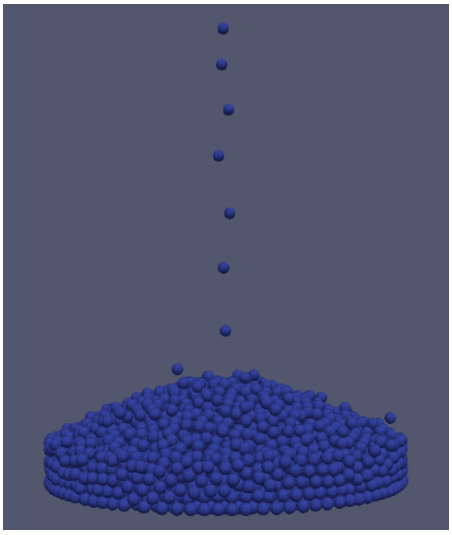


Figure 3.1: Cylindrical container being filled by spheres, simulated by LIGGGHTS and visualized by ParaView.

A critical aspect concerns the time-step choice, which represents the time interval over which the numerical integration is performed, updating positions, velocities and orientations of the spheres. A wrong choice of this parameter may lead to numerical instabilities and a poor resolution of the physics. Its selection is strongly affected by the total mass of particles to be inserted [52]; hence, it has to be carefully chosen for each simulation and its validity must be checked after the simulation. Specifically, the validation of the time step is performed by analyzing the trend of the global kinetic energy of the pebbles starting from the moment the last sphere is inserted. From this point onward, the kinetic energy should monotonically decrease due to energy dissipation, except for occasional temporary spikes caused by localized settling events (e.g., a sphere resting in unstable equilibrium at the top of the bed suddenly dropping to a lower position).

3.2 Building a new constructor in CASTOR

For the purpose of this MSc thesis work, a new constructor was developed in CASTOR, which is able to produce a geometry of spherical inclusions generated by an external code, by reading an input list of sphere centers (with their respective radii). This can be exploited not only for the sphere configurations generated by LIGGGHTS, but also for those produced by the CRP method in OpenMC. Again, it is not possible to disclose any portion of the code due to confidentiality and proprietary rights.

This new constructor serves several critical purposes. First of all, CASTOR is endowed with a large list of tools that can be applied to the geometries read (or sampled) from the code, such as the generation of .vtk and .agora files, which are useful respectively to visualize the geometry produced through ParaView and to export it in the format necessary to run Monte Carlo simulations with TRIPOLI-5.

Geometries generated by DEM codes such as LIGGGHTS, reproducing the physical reality of the filling process, naturally present a conical heap at the top of the container, unlike the flat surfaces produced by the CRP method. An example is displayed in Fig. 3.2.

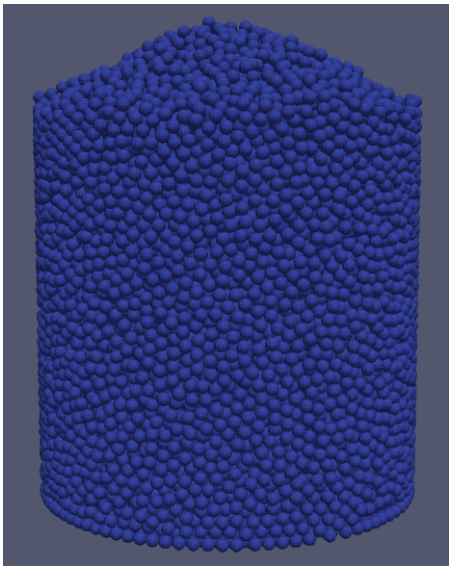


Figure 3.2: Cylindrical container filled with spheres, simulated by LIGGGHTS and visualized by ParaView.

In order to make a fair comparison with geometries produced by CRP, the newly developed constructor offers the possibility of cutting the generated cone, producing a flat container domain, just by considering the container filled strictly below a reference height. This reference height is computed relying on the fact that the topmost empty regions (i.e., where the cone begins to narrow) are always placed at the radial boundaries of the domain. Thus, the idea is to iterate through the list of centers to identify the highest layer of spheres that still extends to the boundaries of the container, and compute the reference value as the average of their heights. Of course, only spheres completely below the reference are saved. An example of truncated geometry is shown in Fig. 3.3.

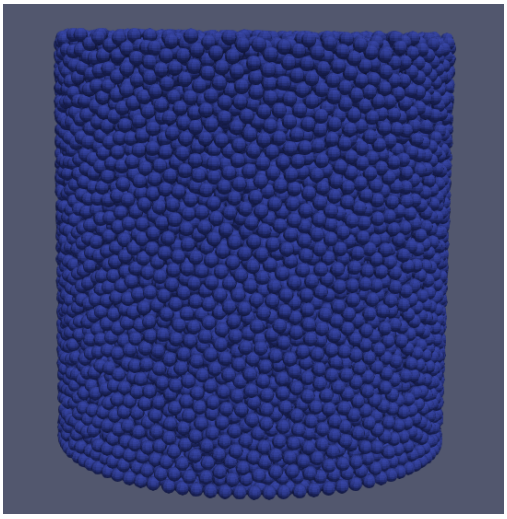


Figure 3.3: Cylindrical container filled with spheres, flattened on the top, simulated by LIGGGHTS and visualized by ParaView.

In addition, we equipped CASTOR with new features aimed at performing detailed statistical analysis of the geometry configurations produced by LIGGGHTS. For instance, contrary to the CRP algorithm, where the packing fraction is a user-defined parameter, here it stems from the choice of the physical parameters that underlie the pebble dynamics. Consequently, for configurations obtained using DEM codes the packing fraction must be estimated a posteriori, once the pebbles in the geometry have attained their equilibrium positions. Remembering that the packing fraction is defined as the ratio between the volume occupied by the spheres and the total container volume, the strategy is to consider only the portion of the container below the reference height. The intersection of the plane defined by the reference height with the topmost spheres naturally leads to the production of some truncated spheres. Therefore, the volume of spheres contained by this portion of the container is computed as the sum of the entire volumes, for spheres fully contained, and spherical cap volumes, for those intersected. This procedure is physically justified since the packing fraction in this central domain is highly representative of the entire core, while greatly simplifying the calculations by dealing with a regular cylindrical shape.

The new constructor can be also used to compute the 'repose angle' which characterizes the distribution of spheres, once equilibrium is attained [46, 47]. The repose angle is the angle of the cone which is formed on the upper side of the container by the accumulation of the pebbles falling from the top. This angle is computed using

$$\tan(\theta) = \frac{h}{r}, \tag{3.1}$$

where h is the height of the cone, computed as the axial distance between the highest point occupied by the spheres and the reference height, and r is the average radius of the cone, computed at the reference height level.

Finally, CASTOR is already endowed with advanced statistical tools based on random media theory. Thanks to the new constructor, these pre-existing features can now be directly applied to the imported geometries to compute, for instance, the chord length distribution, which will be thoroughly discussed in Chapter 4 [28].

3.3 Tests and results

In order to examine the effects of the choice of the time-step τ on the pebble configurations produced by LIGGGHTS, we carried out several investigations. Given the long simulation times associated with LIGGGHTS, we analyzed three different system sizes with a nominal number of spheres N equal to 3.33×10^3, 10^4 and 3.33×10^4. We considered a box-shaped container, with size L chosen according to the number of spheres N that were introduced. We tried to achieve a target packing fraction ξ of 0.6, according to

$$\xi = \frac{\frac{4}{3}\pi \cdot r^3 \cdot N}{L^3}. \tag{3.2}$$

The resulting values for L in the mentioned cases are, respectively, 85.6 cm, 123.5 cm and 184.5 cm.

Since LIGGGHTS naturally produces a conical heap at the top of the bed, a slightly higher number of spheres must be actually provided as input to ensure the complete filling of the box domain. By trial and error, it was found that an over-insertion of approximately 18% was necessary, leading the effective number of spheres to 3930, 11800 and 39300, respectively.

For each configuration it was necessary to find an accurate time-step τ that allows properly integrating the pebble dynamics.

The first configuration that we considered was the one with 3.33×10^3 nominal spheres. Concerning the choice of the time-step τ, as previously discussed, the strategy was to study the decay of the total kinetic energy of the system after the final sphere has been inserted. For this size, the kinetic energy threshold (the value triggering the end of the simulation) was fixed at 3.33×10^{-9} J. Several values of τ were tested, leading to the trends illustrated in Fig. 3.4.

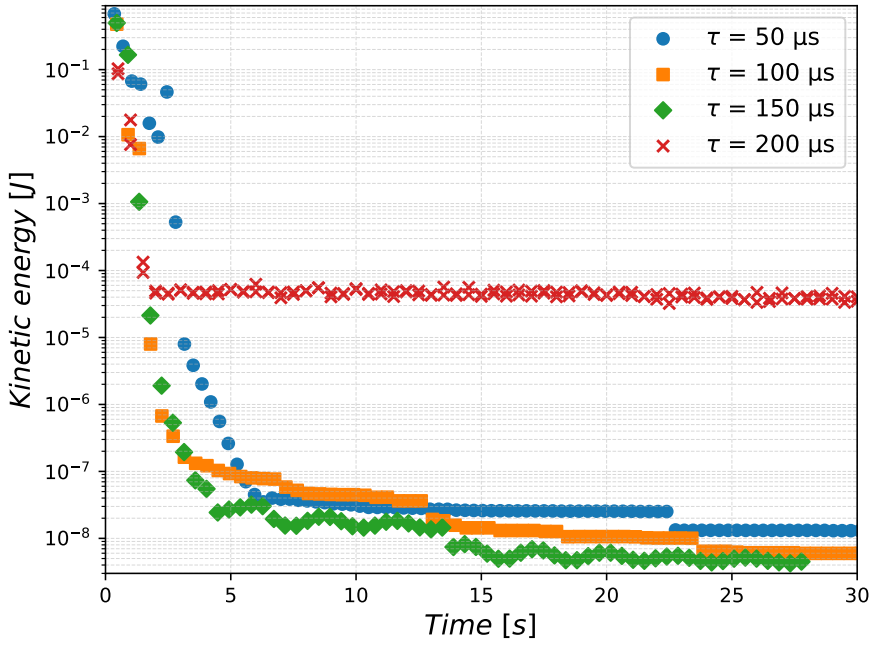


Figure 3.4: Kinetic energy trend after filling the container with N = 3930 pebbles, for different time steps.

The second configuration investigated involved 10^4 nominal spheres. Since the total mass of the pebbles had increased with respect to the previous case, we proportionally raised the energy threshold as well, to 10^{-8} J. Theoretically, the stable time-step τ is expected to be inversely proportional to the square root of the total mass of the spheres [52]; hence, different values were tried, resulting in the findings shown in Fig. 3.5.

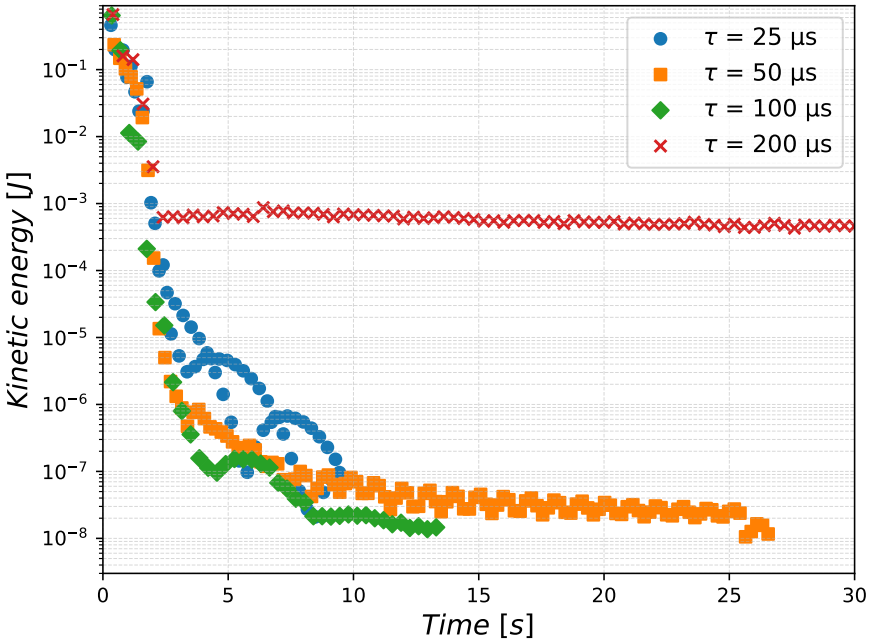


Figure 3.5: Kinetic energy trend after filling the container with N = 11800 pebbles, for different time steps.

The last configuration involved 3.33×10^4 nominal spheres. Following the same reasoning as in the previous case, the energy threshold was further increased to 3.33×10^{-8} J. We tested again different time-steps τ, and the corresponding results are shown in Fig. 3.6.

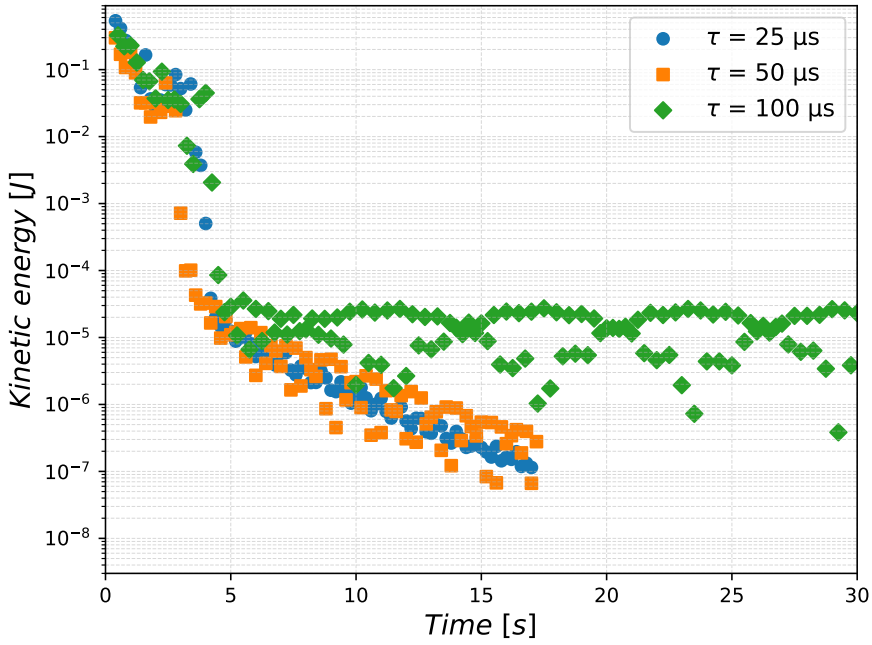


Figure 3.6: Kinetic energy trend after filling the container with N = 39300 pebbles, for different time steps.

The analysis of the effects associated with the choice of the time-step τ leads to the following conclusions.

- Excessively large time-steps τ result in inaccurate integration of the pebble dynamics, as highlighted by the energy trend behavior.
- Conversely, as observed by Fig. 3.5, too small time-steps τ are not a good choice either, since they may produce numerical instabilities in the integration of the pebble dynamics.
- In between these extreme behaviors, there is a rather large window of acceptable values for the time-steps τ. Within this stability region, one should bear in mind that smaller time-steps directly translate to longer simulation times.
- Finally, as theoretically expected, while the total mass of the system (i.e., the number of spheres) increases, the optimal time-step needs to be reduced.

Chapter 4

Chord length distribution

4.1 Introduction

For radiation transport problems involving stochastic mixtures (random arrangements of two or more immiscible materials), the general features of particle propagation can be usefully characterized in terms of the chord length distribution of the random media, and in particular the average chord length associated with each material of the mixture [22, 23, 26].

In essence, a chord of a given material in the mixture is defined as the distance between the entry and exit points of a random line, isotropically and uniformly thrown through the medium, with respect to the material boundaries. Consequently, for a given material, the chord length distribution describes the probability that such a random chord has a specific length. Thus, this distribution is a measure of the material dispersion in the mixture, and provides an estimation of the degree of disorder of the medium and its possible effects on neutron transport across the mixture. For these reasons, it is a key quantity in the modeling of stochastic media. When the full chord length distribution is too cumbersome to estimate, the average chord length (i.e., the first moment of the associated distribution) provides a reliable surrogate that is often used in real-world applications [22, 23, 26].

CASTOR is able to compute chord length distributions for the different random media models that can be sampled, and in particular for spherical inclusions randomly inserted into a background matrix (i.e., the empty volume inside the container left by spherical inclusions), which is the prototype model chosen for the description of pebble bed cores. The estimation is performed as follows. A given number of lines is sampled, generated into the material considered with a random orientation and position (respecting the μ-randomness property [26]). The length of each chord is then automatically determined by the intersection of the line with the boundaries of the material under study. CASTOR attributes a 'color' label to

the chords, according to whether the intersections are with the background matrix, or with the spheres.

The expected chord length distributions for both the background matrix and the spheres, assuming mono-dispersed hard spheres (i.e. spheres with constant radius and without any penetration between them) in three-dimensional settings, are given respectively by [53]

$$P(\ell_0) \approx \frac{3\xi}{4(1-\xi)}\frac{1}{R}\exp\left(-\frac{3\xi}{4(1-\xi)}\frac{1}{R}\ell_0\right) \tag{4.1}$$

and

$$P(\ell_1) = \begin{cases} \frac{\ell_1}{2R^2} & \text{if } \ell_1 \leq 2R \\ 0 & \text{if } \ell_1 > 2R \end{cases}, \tag{4.2}$$

where ξ is the packing fraction, R is the radius of the spheres, ℓ_0 is the length of the chords in the background matrix and ℓ_1 is the length of chords in the spheres.

4.2 Comparison between different models

The comparison of the chord length distributions obtained from different random sphere packing models (e.g., CRP versus DEM models) allows inferring possible discrepancies in the statistical properties of the resulting geometries, which might lead to apparent effects on neutron transport.

For the purpose of this work, we decided to compare geometries sampled with the CRP model of CASTOR against those sampled with the CRP model of OpenMC and those achieved by DEM simulations. For this statistical analysis, geometries sampled with OpenMC and with LIGGGHTS were imported into CASTOR, thanks to the updates discussed in Chapter 3, with the aim of taking advantage of the chord length distribution samplings features. CRP geometries sampled with CASTOR and OpenMC are expected to be in perfect agreement, since they are based on the same construction method (the Jodrey-Tory algorithm), while discrepancies are expected when comparing CRP to DEM models, due to the different approaches followed to generate the sphere packing.

To sample each chord length distribution, 5 replicas of geometries and 50000 sample lines for each were considered, in order to achieve results with a relatively low statistical dispersion.

In order to conduct a fair comparison between models, the geometries must be sampled under the most similar possible conditions. Since LIGGGHTS leads to non-flat configurations because of the simulated dynamics, the chosen strategy was to use the code to fill a higher container, and subsequently import the list of spheres into CASTOR to 'cut' the container to a given reference height, as shown

in Chapter 3: this allows achieving a flat configuration with the same dimensions as the corresponding CRP model in a simple container. For our investigations, a cubical container with a side size of 147.1 cm was considered, filled with pebbles of 3 cm radius.

Regarding the packing fraction, since for LIGGGHTS it is not possible to impose it directly as an input parameter, we chose to first perform the LIGGGHTS simulations, estimate the resulting average packing fraction over the different replicas, and finally use this value as a target parameter for the CRP codes. In our case, it turned out that the estimated packing fraction stemming from the DEM model was about 0.59.

Chord length distributions for the various codes used are represented in Figs. 4.1, 4.2, 4.3, 4.4, 4.5 and 4.6.

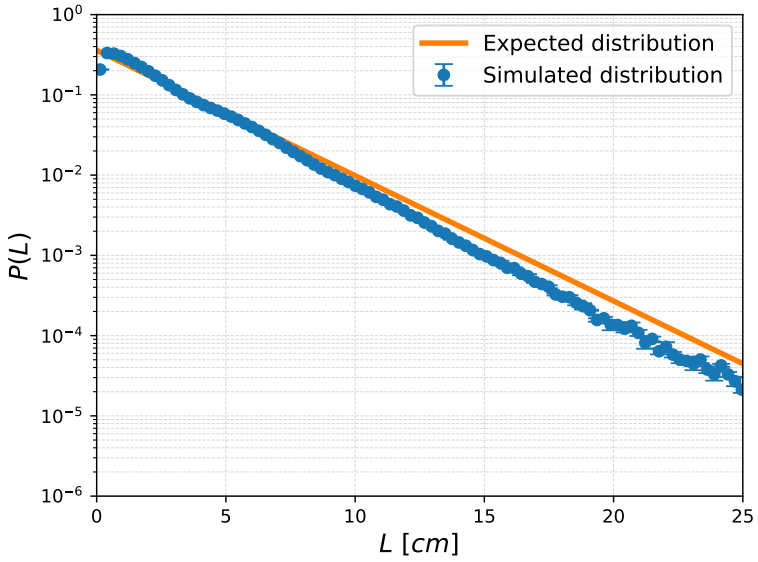


Figure 4.1: Chord length distribution in the background matrix, for CRP geometries generated by CASTOR.

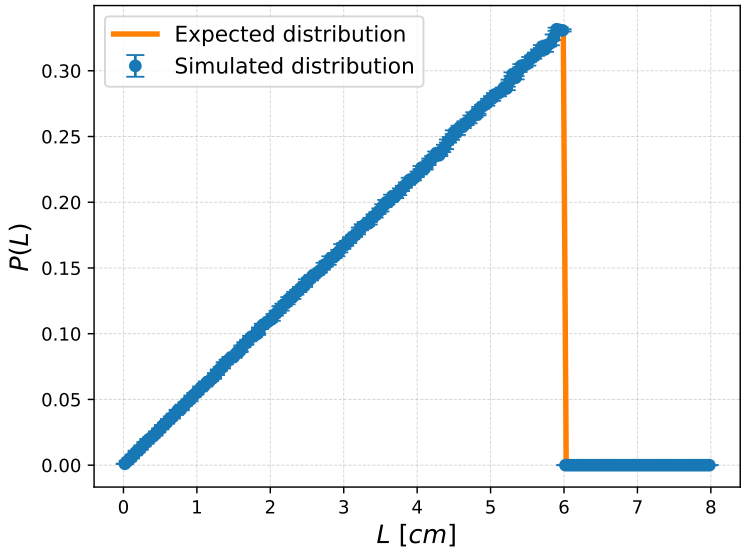


Figure 4.2: Chord length distribution inside the spheres, for CRP geometries generated by CASTOR.

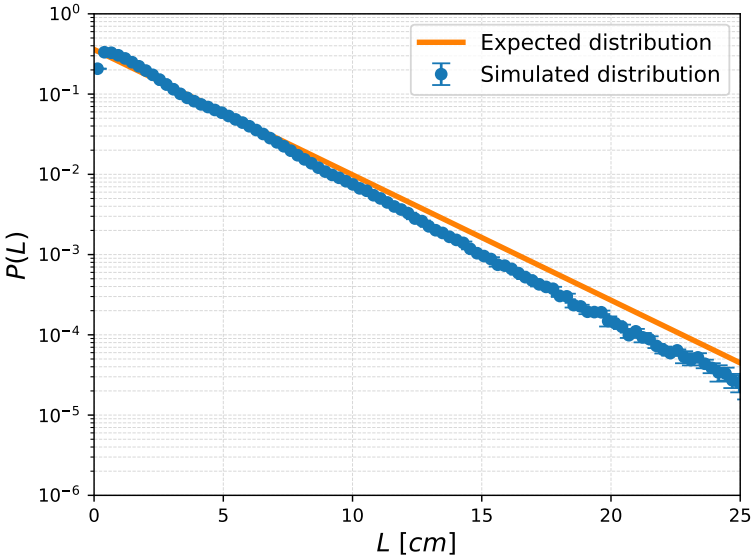


Figure 4.3: Chord length distribution in the background matrix, for CRP geometries generated by OpenMC.

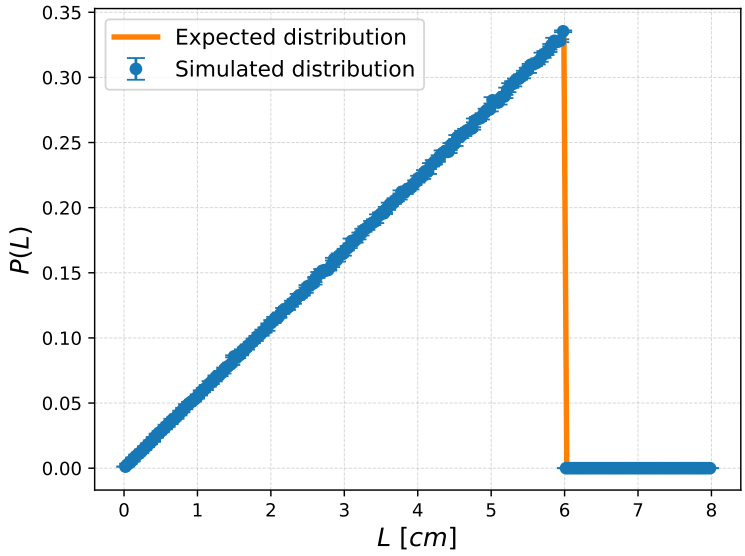


Figure 4.4: Chord length distribution inside the spheres, for CRP geometries generated by OpenMC.

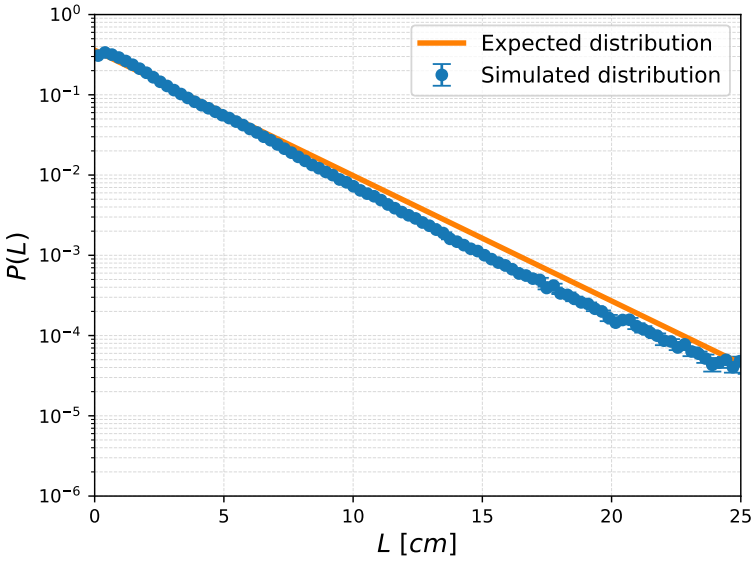


Figure 4.5: Chord length distribution in the background matrix, for DEM geometries generated by LIGGGHTS.

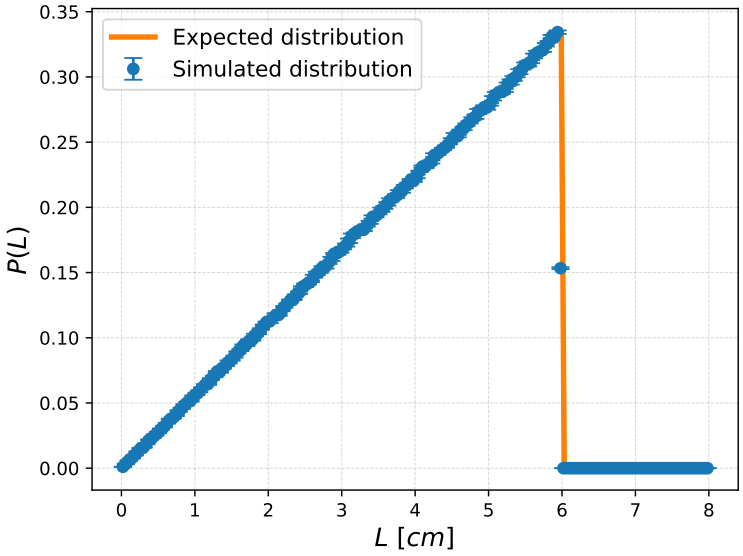


Figure 4.6: Chord length distribution inside the spheres, for DEM geometries generated by LIGGGHTS.

From the previous numerical results, it can be visually assessed that for all codes the simulated distributions are quite close to the expected theoretical ones.

A quantitative assessment was then performed using the t-test for the pair of models to be compared. In essence, the continuous chord length domain is divided into discrete intervals (bins) to construct probability histograms. For each individual bin, the difference between the probability values of the two models is computed and then normalized by dividing it by its corresponding standard error. If this normalized difference falls within ± 3 band, the discrepancy between the two distributions can be attributed just to their statistical dispersion, meaning that they are statistically equivalent; conversely, if the average difference lies outside this band, a statistically significant discrepancy is spotted. This statistical analysis is displayed in Figs. 4.7, 4.8, 4.9 and 4.10.

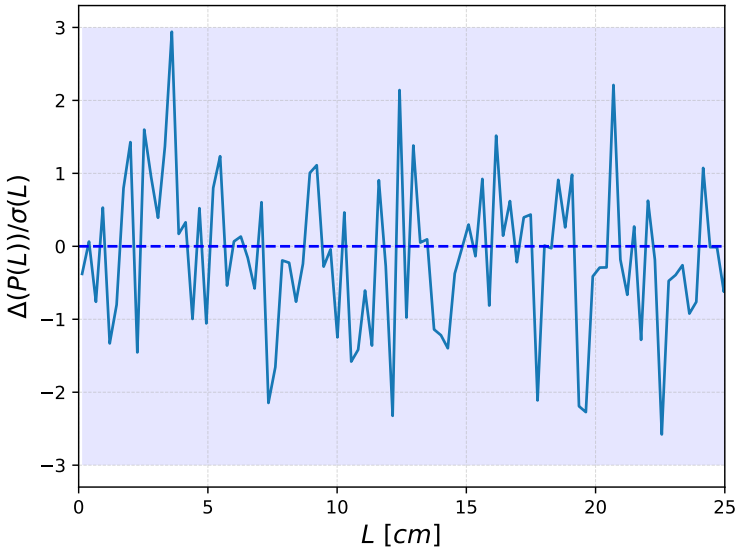


Figure 4.7: Comparison of chord length distribution in the background matrix, for geometries generated by CRP with OpenMC and CASTOR.

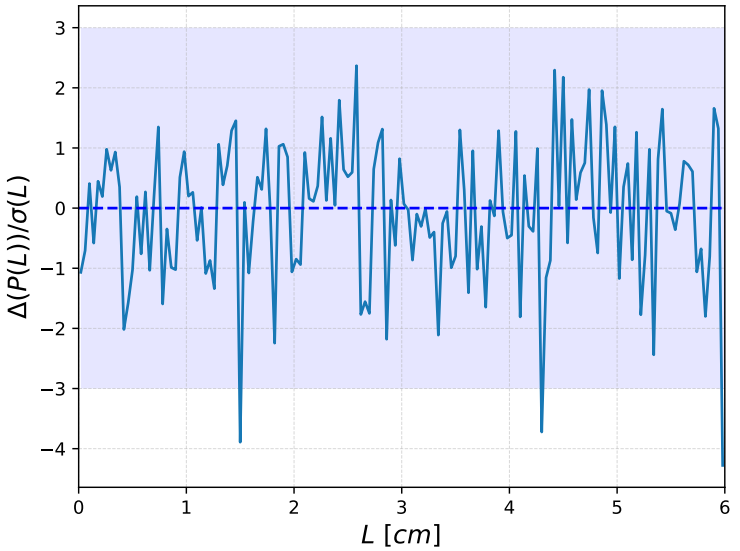


Figure 4.8: Comparison of chord length distribution inside the spheres, for geometries generated by CRP with OpenMC and CASTOR.

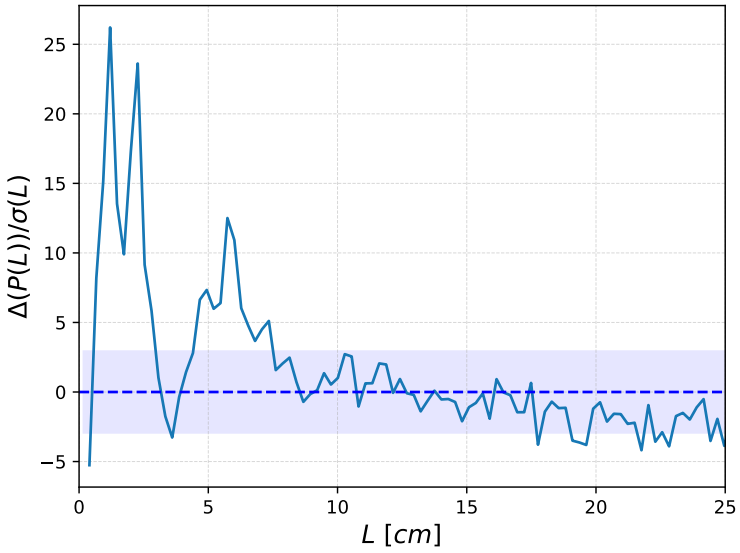


Figure 4.9: Comparison of chord length distribution in the background matrix, for geometries generated by DEM with LIGGGHTS and CRP with CASTOR.

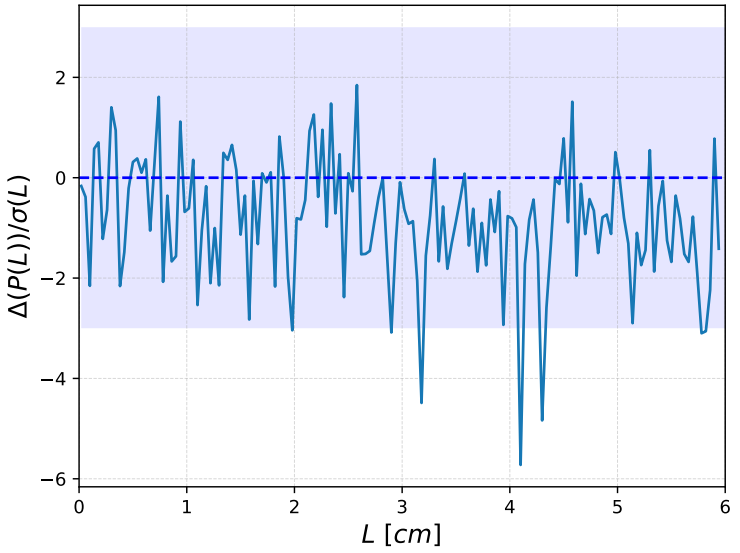


Figure 4.10: Comparison of chord length distribution inside the spheres, for geometries generated by DEM with LIGGGHTS and CRP with CASTOR.

Please observe that the first point in Fig. 4.9 and the last point in Fig. 4.10 have been removed for a better visualization of the results, since large discrepancies were present between the compared models at those extremes. These localized differences are explained by the fact that LIGGGHTS allows for a slight elastic deformation of the spheres during the container filling process. This leads to a highly significant difference in the probability of finding chords with a length close to the pebble diameter inside the spheres, and chords with a length close to zero in the background matrix.

Overall, the statistical analysis based on chord lengths shows that the CRP models of CASTOR and OpenMC are in very good agreement, as expected. On the contrary, statistically significant deviations are observed between the CRP model of CASTOR and the DEM configurations of LIGGGHTS. As anticipated, these deviations stem from the fact that the CRP and DEM models are inherently different approaches to the sampling of the equilibrium distribution of sphere packing. For the case considered here, these discrepancies are larger for the chords in the background matrix than for those within the pebbles: further investigations would be needed in order to verify whether this statement is true in general.

Chapter 5

Modeling the pebble-beds using CASTOR

5.1 Introduction

As discussed in Chapter 1, one of the main objectives of this MSc thesis is to reproduce the entire fuel geometry of Pebble Bed Reactors using CASTOR, accounting for the double heterogeneity of the system. Specifically, two levels of heterogeneity for the materials are present:

- A macroscopic level, related to the random distribution of pebbles within the reactor core;
- A microscopic level, concerning the random distribution of TRISO particles inside each pebble.

For illustration, a simplified two-dimensional example generated with a Python script is shown in Fig. 5.1: TRISO particles are randomly packed inside spherical pebbles, which are in turn randomly distributed into the reactor core. Accurately modeling both heterogeneity scales is essential to achieve reliable simulation results, and this issue is particularly relevant for Monte Carlo neutron transport codes, as the behavior of neutrons strongly depends on the spatial distribution of materials.

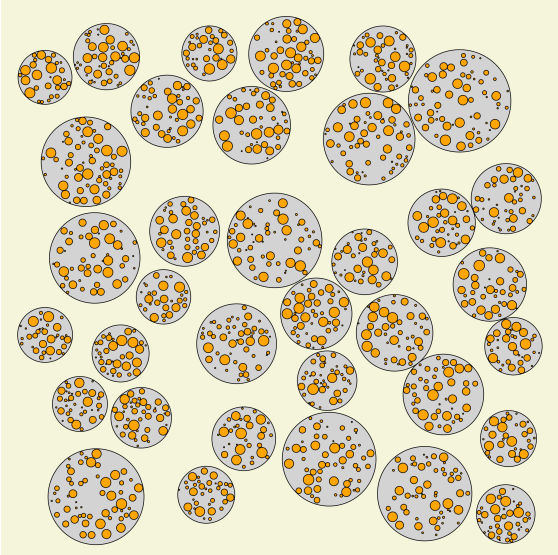


Figure 5.1: Sketch for double heterogeneity problem.

Thanks to the new methods implemented in CASTOR (see Chapter 2), it is now possible to reproduce realistic packing fractions for the distribution of pebbles. To achieve a full double heterogeneity model in CASTOR, the remaining goal is to implement the sampling of nested random spherical inclusions (i.e., smaller random spheres packed inside larger random spheres). This development is described in the following.

5.2 Implementation of double heterogeneity in CASTOR

Initially, for the sake of simplicity, only two different kinds of spherical inclusions were considered: one representative of TRISO particles and one representative of pebbles. The first step of the implementation consisted of defining the internal composition of the pebbles, referred to as the filler, which can be either a universe or a rectangular lattice, depending on whether a mesh is used or not. This filler contains TRISO particles dispersed within a background matrix representing the moderator (graphite).

Once the filler is generated, it is used to populate the internal volume of the pebbles: since only a single filler was defined in this initial stage, every pebble in the system was filled in the exact same way. The pebbles are dispersed into the container, where the background material corresponds this time to the coolant. An example is illustrated in Fig. 5.2 for a $10\times10\times10$ cm^3 cubical container.

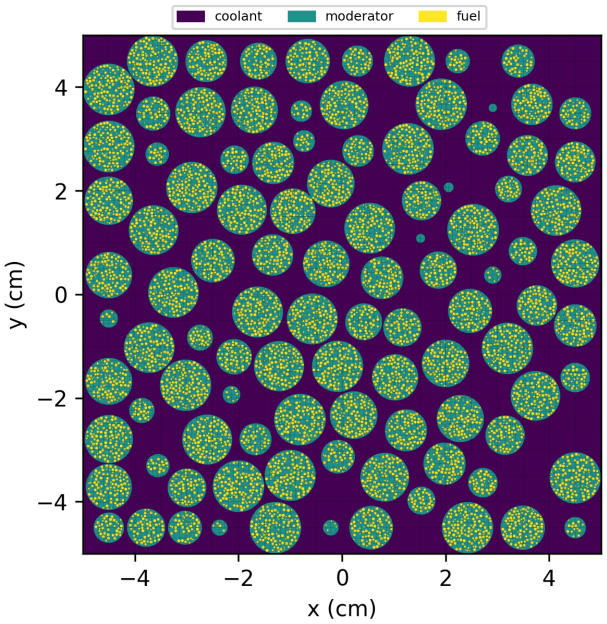


Figure 5.2: Cross-section of a cubical container filled with pebbles, which in turn contain the TRISO particles.

The next step consisted of extending the code capabilities by introducing different types of fillers for the pebbles. In particular, once the macroscopic pebbles are generated, the specific type of filler assigned to each pebble is determined by sampling a user-defined probability distribution. This feature allows modeling, for instance, pebbles filled only by graphite (the so called 'dummy pebbles' often used in HTRs to create fuel/moderator mixtures in the core [46, 47]). An example is illustrated in Fig. 5.3, for a $10\times10\times10$ cm^3 cubical container.

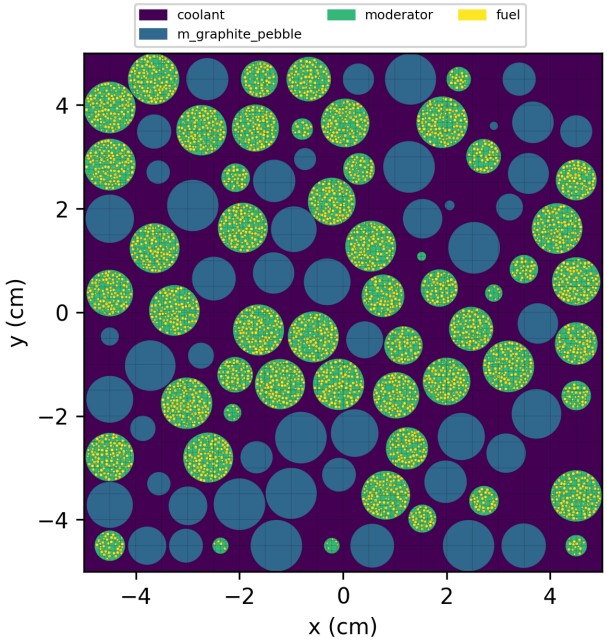


Figure 5.3: Cross-section of a cubical container filled with a mixture of pebbles: some containing TRISO particles and others composed entirely of graphite (dummy pebbles).

The final step consisted of developing a feature enabling multi-layer spherical shells. Indeed, as discussed in Chapter 1, each TRISO particle is composed of a central fuel kernel surrounded by a porous carbon buffer and by three layers, respectively made of pyrolytic carbon, silicon carbide and pyrolytic carbon again. Conversely, the pebbles present a single graphite layer, acting as a fuel-free zone. Thanks to this further development, CASTOR can now sample realistic pebble-bed geometries. An example is illustrated in Fig. 5.4 for a $10\times10\times10$ cm^3 cubical container.

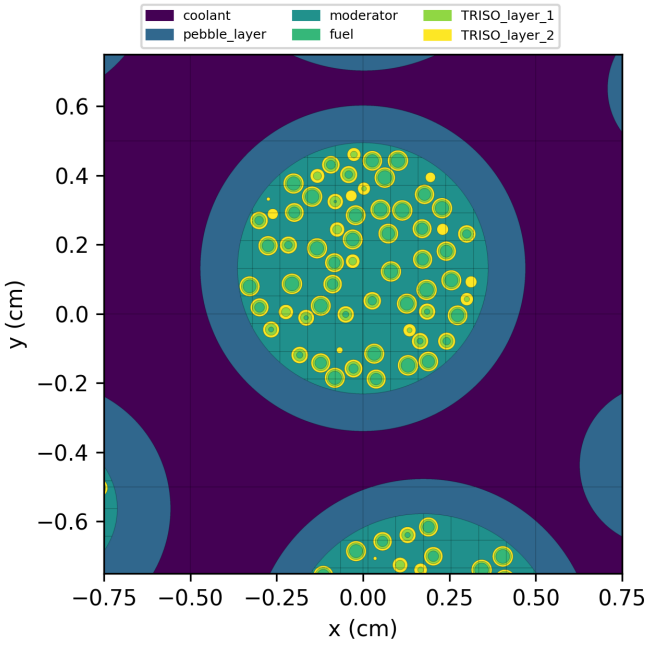


Figure 5.4: Magnified cross-section of a cubical container filled with pebbles with an outer shell, which in turn contain the TRISO particles made of two external layers.

Part II

The HTR-10 benchmark

Chapter 6

Analysis of HTR-10 benchmark

6.1 Introduction

The 10 MW High-Temperature Gas-cooled Test Reactor (HTR-10) is a modular pebble bed type reactor. The HTR-10 was designed with the following main objectives [54]:

- Acquiring experimental validation and operational experience for HTRs;
- Establishing an irradiation and experimental facility;
- Demonstrating the inherent safety features of the Modular HTR;
- Testing electricity and heat co-generation and closed cycle gas turbine technology;
- Completing research and development work for nuclear process heat applications.

Developed in China starting from 1995, it achieved criticality for the first time in December 2000 and began operating at full power conditions in January 2003. The HTR-10 is helium-cooled, graphite-moderated and characterized by Pebble-Bed fuel. It relies on passive safety: for instance, in severe accident conditions, the decay heat of the HTR-10 can be transferred and dissipated into the atmosphere via the passive decay heat removal system. Furthermore, the maximum temperature of the fuel, even during accident scenarios, should not exceed the limit of 1600 °C, below which the SiC layer can retain the fission products within the particles [46, 47]. A scheme corresponding to the axial cross-section of the HTR-10 reactor is illustrated in Fig. 6.1.

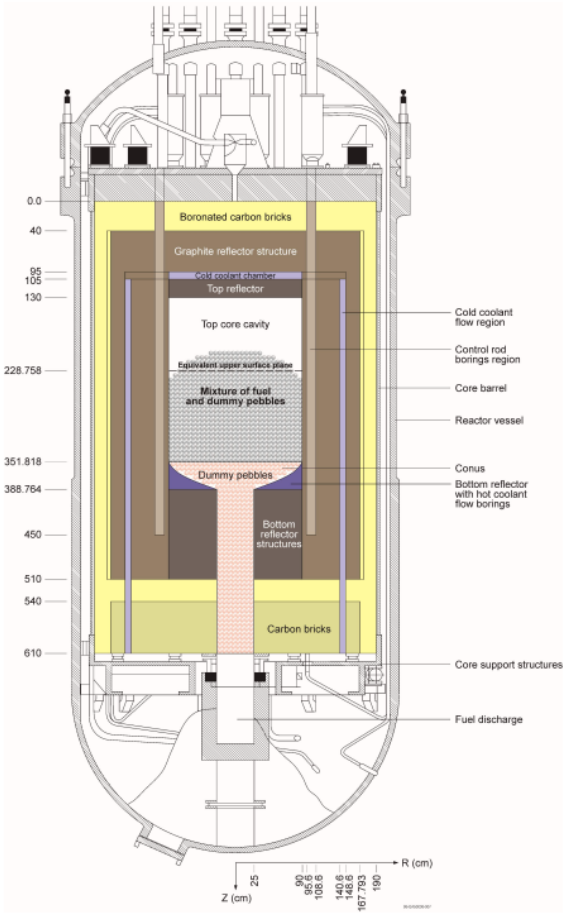


Figure 6.1: HTR-10 reactor sketch [55]

In this second part of the thesis, we present the application of TRIPOLI-5 coupled with new CASTOR tools, specifically to the investigation of a realistic HTR configuration. For this purpose, we focus on the start-up of the HTR-10 reactor, whose specifications have been condensed into an international benchmark of the IRPHEP database [46, 47]. Two different sets of specifications are provided in the literature, one concerning a simplified model of the core (where the actual shape of the pebble bed is approximated by a cylindrical shape with an equivalent height) and one corresponding to the realistic shape of the pebble bed, including the upper 'cone' formed during the fueling. Both correspond to a critical configuration with all control rods withdrawn [46, 47]. This allows for the extensive validation of the novel features of CASTOR and more broadly TRIPOLI-5, over a meaningful application of reactor physics analysis. Whenever possible, OpenMC is also used to perform code-to-code comparisons, enhancing the reliability of the obtained results.

6.2 Geometry model and nuclear data

In order to reproduce the geometry model of HTR-10 through TRIPOLI-5 and OpenMC, we relied on the description provided in Ref. [55].

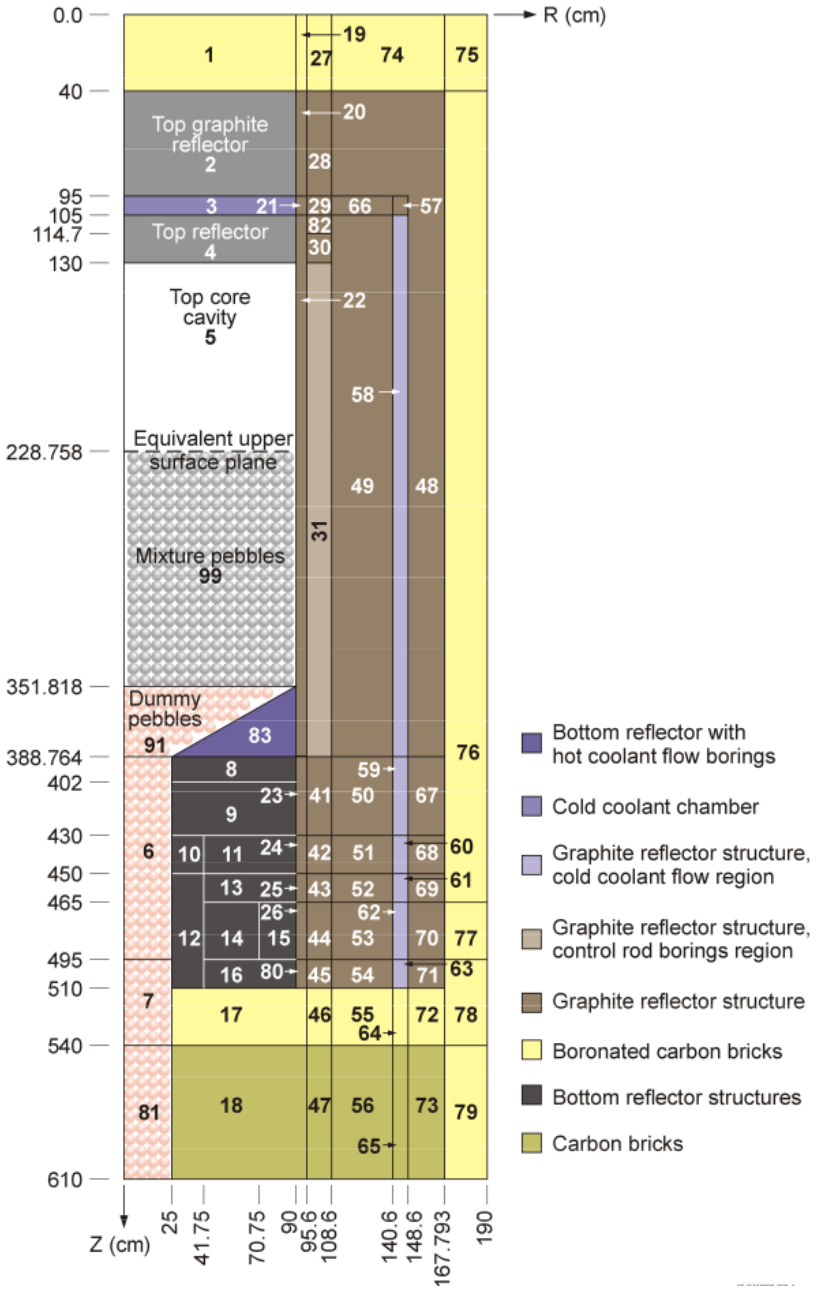


Figure 6.2: HTR-10 section [55]

As illustrated in Fig. 6.2, the reactor is partitioned into sub-regions, each characterized by specific geometrical dimensions and material compositions. Based on this description, a Monte Carlo model for TRIPOLI-5 was built using the geometry engine AGORA, while an equivalent model was developed for OpenMC using its native Constructive Solid Geometry (CSG) engine.

Some of the regions (5, 99, 91, 6, 7 and 81) contain the random configuration of fuel and dummy (moderator) pebbles, with proportions specified in the benchmark. For regions 7 and 81, we followed the benchmark specifications by replacing the random media with a homogeneous mixture, using material densities provided in Ref. [55]. For the other regions, we assigned stochastic realizations of sphere packings, whose details are provided in the following sections, depending on the models adopted (CRP or DEM).

Geometry and material data for the pebbles and TRISO particles are given in Tab. 6.1.

Parameter Description	**Value**
Diameter of fuel and dummy (no-fuel) pebbles	6.0 cm
Volume fraction of pebbles in the core	0.61
Number of fuel pebbles in top core cavity	9627
Number of dummy pebbles in top core cavity	7263
Diameter of fuel zone	5.0 cm
Average number of fuel particles per pebble	8335
Diameter of TRISO kernel	0.5 mm
Coating layer materials (starting from kernel)	Buffer/PyC/SiC/PyC
Coating layer thicknesses (mm), respectively	0.09/0.04/0.035/0.04

Table 6.1: Pebble and TRISO data [55]

Regarding the spatial distribution of the pebbles, a clear distinction is made based on the vertical position of their centers. Pebbles located within the top core cavity (the union of regions 99 and 5) consist of a mixture of fuel pebbles (containing TRISO particles embedded in a graphite matrix) and dummy pebbles. Their respective proportions are given in Tab. 6.1. Conversely, pebbles located below the top core cavity, which belong to the 'discharge' region (union of regions 91, 6, 7 and 81), are modeled exclusively as dummy pebbles (containing only graphite).

For what concerns the materials of the system, the compositions of each volume of the benchmark model above were extracted from Tabs. 6.3, 6.4, 6.5, 6.6 and 6.7, reproduced here from Ref. [55], for convenience. The helium (coolant) is actually replaced by air in the start-up core considered here, as mentioned in the specifications [55].

Nuclide	Atomic Number Density (atoms/barn-cm)
Graphite at 1.76 g/cm^3	8.82418E-2
^{10}B at 4.8366 ppm total boron concentration	9.42800E-8
^{11}B at 4.8366 ppm total boron concentration	3.79489E-7
Carbon in boronated carbon bricks	7.65984E-2
^{10}B in boronated carbon bricks	6.89235E-4
^{11}B in boronated carbon bricks	2.77426E-3

Figure 6.3: Number Densities in Solid Reflector Components [55]

Nuclide	Atomic Number Density (atoms/barn-cm)
^{235}U in kernel	3.99198E-3
^{238}U in kernel	1.92441E-2
^{16}O in kernel	4.64720E-2
^{10}B in kernel	4.06384E-7
^{11}B in kernel	1.63575E-6
Carbon in buffer	5.51511E-2
^{10}B in buffer	1.58513E-8
^{11}B in buffer	6.38035E-8
Carbon in IPyC and OPyC	9.52610E-2
^{10}B in IPyC and OPyC	2.73795E-8
^{11}B in IPyC and OPyC	1.10206E-7
Carbon in SiC	4.77597E-2
Si (natural) in SiC	4.77597E-2
Carbon in fuel matrix	8.67377E-2
^{10}B in fuel matrix	2.49298E-8
^{11}B in fuel matrix	1.00345E-7
Carbon in dummy pebbles	9.22528E-2
^{10}B in dummy pebbles	2.54951E-9
^{11}B in dummy pebbles	1.02621E-8

Figure 6.4: Number Densities in Core Components [55]

Nuclide	Volume Percent in Dry Air[(a)(b)]	Atomic Number Density (atoms/barn-cm)
Nitrogen in dry air	78.084	3.95901E-5
Oxygen in dry air	20.9476	1.06209E-5
Argon in dry air	0.934	2.36778E-7
Hydrogen in vapor at saturation		8.58028E-7
Oxygen in vapor at saturation		4.29014E-7

Figure 6.5: Number Densities of Air Constituents in Experiment Conditions (15°C and 0.1013 MPa) [55]

No. of Zone	Carbon Density (atoms/barn-cm)	Natural Boron Density (atoms/barn-cm)	Description
83	0.851047E-01	0.456926E-06	Bottom reflector with hot coolant flow borings
1	0.729410E-01	0.329811E-02	Boronated carbon bricks
2	0.851462E-01	0.457148E-06	Top graphite reflector
3	0.145350E-01	0.780384E-07	Cold coolant chamber
4	0.802916E-01	0.431084E-06	Top reflector
5			Top core cavity
7	0.572501E-01	0.277884E-08	Dummy pebbles, simplified as graphite of lower density
8	0.781408E-01	0.419537E-06	Bottom reflector structures
9	0.823751E-01	0.442271E-06	Bottom reflector structures
10	0.843647E-01	0.298504E-03	Bottom reflector structures
11	0.817101E-01	0.156416E-03	Bottom reflector structures
12	0.850790E-01	0.209092E-03	Bottom reflector structures
13	0.819167E-01	0.358529E-04	Bottom reflector structures
14	0.541118E-01	0.577456E-04	Bottom reflector structures
15	0.332110E-01	0.178309E-06	Bottom reflector structures
16	0.881811E-01	0.358866E-04	Bottom reflector structures
17,27,46, 55,64,72, 74,75,76, 77,78,79	0.765984E-01	0.346349E-02	Boronated carbon bricks
18,47,56, 65,73	0.797184E-01	0.000000E+00	Carbon bricks
19	0.761157E-01	0.344166E-02	Boronated carbon bricks
20	0.878374E-01	0.471597E-06	Graphite reflector structure
21	0.579696E-01	0.311238E-06	Graphite reflector structure
22,23,25, 26,28,30, 31,41,43, 44,45,48, 49,50,52, 53,54,58, 59,61,62, 63,67,69, 70,71,80, 82	0.882418E-01	0.473769E-06	Graphite reflector structure
24,51,68	0.879541E-01	0.168369E-03	Graphite reflector structure
29	0.524843E-01	1.819690E-05	Graphite reflector structure
42	0.879637E-01	0.162903E-03	Graphite reflector structure
66	0.582699E-01	0.312850E-06	Graphite reflector structure
57	0.728262E-01	0.391003E-06	Graphite reflector structure
60	0.879538E-01	0.168369E-03	Graphite reflector, cold coolant flow region
81	0.847872E-01	0.000000E+00	Dummy pebbles, but artificially taken as carbon bricks

Figure 6.6: Compositions of Zones in Reflector [55]

No. of Zone [a]	Coolant Volume Fraction [b]	N	O (air)	Ar	H (water)	O (water)
83	3.5551E-02	1.38993E-06	3.72877E-07	8.31317E-09	3.05038E-08	1.52519E-08
1	4.7748E-02	1.86679E-06	5.00805E-07	1.11653E-08	4.09691E-08	2.04846E-08
2	3.5081E-02	1.37155E-06	3.67947E-07	8.20327E-09	3.01005E-08	1.50502E-08
3	8.3528E-01	3.26568E-05	8.76083E-06	1.95320E-07	7.16694E-07	3.58347E-07
4	9.0096E-02	3.52247E-06	9.44972E-07	2.10679E-08	7.73049E-08	3.86524E-08
5	1.0000E+00	3.90968E-05	1.04885E-05	2.33838E-07	8.58028E-07	4.29014E-07
7	3.9000E-01	1.52478E-05	4.09052E-06	9.11968E-08	3.34631E-07	1.67315E-07
8	1.1447E-01	4.47541E-06	1.20062E-06	2.67674E-08	9.82185E-08	4.91092E-08
9	6.6484E-02	2.59931E-06	6.97317E-07	1.55465E-08	5.70451E-08	2.85226E-08
10*	4.0560E-02	1.58577E-06	4.25414E-07	9.48447E-09	3.48016E-08	1.74008E-08
11*	7.2253E-02	2.82486E-06	7.57826E-07	1.68955E-08	6.19951E-08	3.09975E-08
12*	3.3478E-02	1.30888E-06	3.51134E-07	7.82843E-09	2.87251E-08	1.43625E-08
13*	7.1278E-02	2.78674E-06	7.47599E-07	1.66675E-08	6.11585E-08	3.05793E-08
14*	3.8613E-01	1.50964E-05	4.04992E-06	9.02919E-08	3.31310E-07	1.65655E-07
15	6.2364E-01	2.43823E-05	6.54105E-06	1.45831E-07	5.35101E-07	2.67550E-07
16*	2.8700E-04	1.12208E-08	3.01020E-09	6.71115E-11	2.46254E-10	1.23127E-10
17,**27**,46,55,**64**, 72,74,75,76,**77**, 78,79	0	0	0	0	0	0
18,**47**,56,**65**,73	0	0	0	0	0	0
19	6.3017E-03	2.4638E-07	6.6095E-08	1.4736E-09	5.40704E-09	2.70352E-09
20	4.5829E-03	1.7918E-07	4.8068E-08	1.0717E-09	3.93226E-09	1.96613E-09
21	3.4306E-01	1.3413E-05	3.5982E-06	8.0220E-08	2.94355E-07	1.47178E-07
22,23,25,**26,28*,30,31**, **41,43,44,45**,48,49,50, 52,**53**,54,**58,59,61,62, 63**,67,69,**70**,71,80,**82***	0	0	0	0	0	0
24*,51*,68*	1.3580E-03	5.3093E-08	1.4243E-08	3.1755E-10	1.16520E-09	5.82601E-10
29*	2.3064E-01	9.0173E-06	2.4191E-06	5.3932E-08	1.97896E-07	9.89478E-08
42	4.0522E-01	1.5843E-05	4.2501E-06	9.4756E-08	3.47690E-07	1.73845E-07
66	3.1516E-03	1.2322E-07	3.3056E-08	7.3696E-10	2.70416E-09	1.35208E-09
57	3.3966E-01	1.3280E-05	3.5625E-06	7.9425E-08	2.91438E-07	1.45719E-07
60	1.7470E-01	6.8302E-06	1.8323E-06	4.0851E-08	1.49897E-07	7.49487E-08
81	3.9000E-01	1.5248E-05	4.0905E-06	9.1197E-08	3.34631E-07	1.67315E-07

Figure 6.7: Compositions of Zones in Reflector [55]

Chapter 7

Application of CRP to the modeling of a simplified sphere packing

7.1 General features

In this chapter, the neutronic analysis on the HTR-10 reactor is performed considering the 'simplified core geometry' provided in the benchmark description (Ref. [55]), applying TRIPOLI-5 coupled to the CRP model of CASTOR. For the sake of comparison and validation, the same analysis is also performed using OpenMC and its CRP model.

With reference to Fig. 6.2, in this analysis volumes 6 and 91 are filled with a homogeneous mixture, which is equivalent to that of volume 7. This simplification is adopted because the randomness related to the distribution of dummy pebbles in these lower regions is assumed to have a negligible effect on neutron transport. In addition, the CRP method cannot be applied, yet, to conical shapes.

Concerning the top core cavity region, a cylinder corresponding to region 99 (up to a defined 'equivalent core height') is filled with pebbles generated by the CRP method, dispersed within the coolant. Although this approach is less accurate than the actual equilibrium configuration of the pebbles that was achieved in HTR-10 for the start-up core, investigations carried out in Ref. [55] suggest that the approximations introduced by this simplified model are acceptable (especially concerning the system reactivity). The remaining part of the core, corresponding to region 5, is filled by the coolant. Subsequently, the geometry engines of TRIPOLI-5 and OpenMC are used to generate the remaining (non-random) part of the HTR-10 model.

7.2 TRIPOLI-5 coupled with CASTOR

Thanks to the new developments for CASTOR described in Chapter 5, it is now possible to model the double heterogeneity geometry of the core.

As detailed in the previous chapter, the top core cavity is characterized by 16890 pebbles, comprising 7263 dummy pebbles and 9627 fuel pebbles. To reproduce the exact number of pebbles and TRISO, their packing fractions were first computed using Eq. (3.2), where the denominator was replaced by the volume of the corresponding container. Subsequently, these computed values were given as input to CASTOR, together with the geometrical parameters and the materials required for these regions. To ensure the exact pebble 'coloring' (i.e., the material assignment), a constrained sampling of the spheres was implemented, such that if, for instance, the dummy target is reached, the next random samples are only of fuel type.

The resulting geometry was then imported into a Python script, which, together with the other non-random regions built using AGORA, allows describing the entire geometry of HTR-10. The TRIPOLI-5 model is illustrated in Figs. 7.1, 7.2 and 7.3.

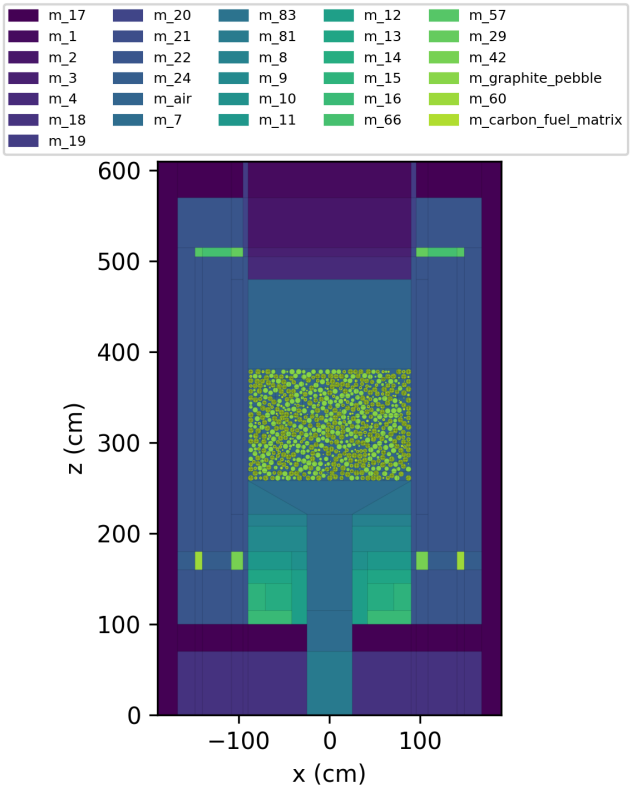


Figure 7.1: Axial cross-section of the HTR-10 model (TRIPOLI-5/CASTOR).

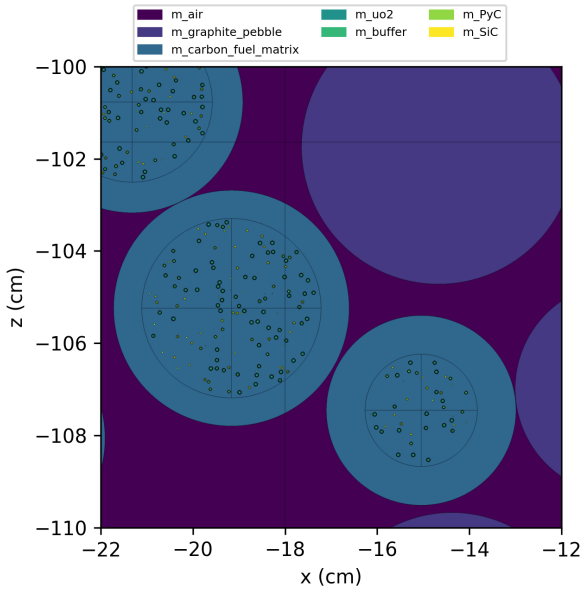


Figure 7.2: Detailed view of HTR-10 pebbles (TRIPOLI-5/CASTOR).

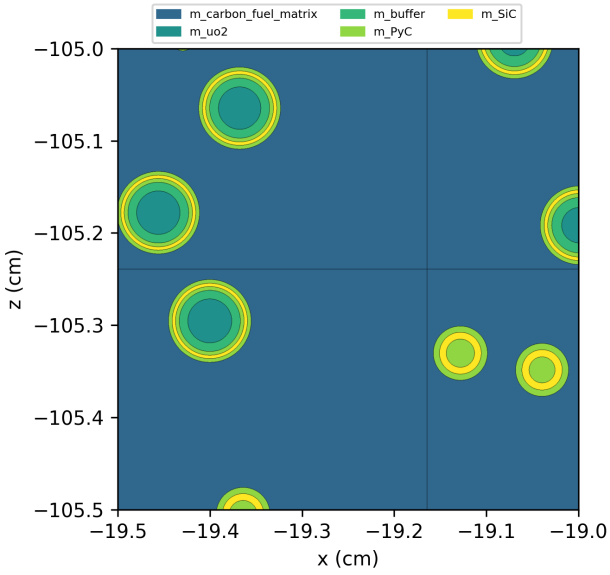


Figure 7.3: Detailed view of HTR-10 TRISO particles (TRIPOLI-5/CASTOR).

7.3 OpenMC with the native CRP model

For OpenMC, the geometry model of HTR-10 was built following the same strategy, the main difference being that the random part of the model was generated using the native CRP tool, instead of resorting to an external tool as we did for TRIPOLI-5. Thus, this time the generation of the region 99 was implemented directly into the Python script used for the generation of the other regions. Here, TRISO particles and pebbles can be generated directly imposing their numbers as input to the code, instead of computing the corresponding packing fraction. As for the 'coloring' of pebbles (fuel or dummy), we applied the same rules used for TRIPOLI-5 model. The OpenMC model is illustrated in Figs. 7.4, 7.5 and 7.6.

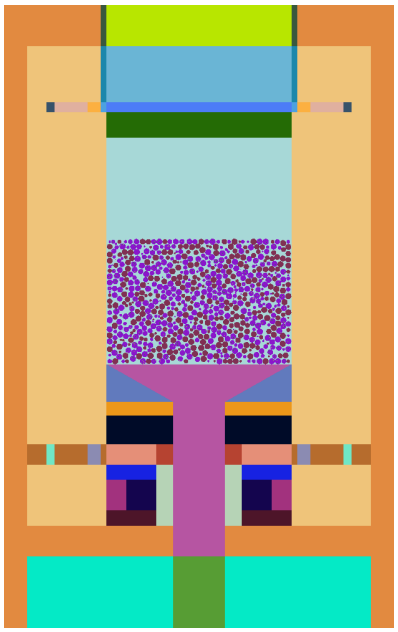


Figure 7.4: Axial cross-section of the HTR-10 model (OpenMC).

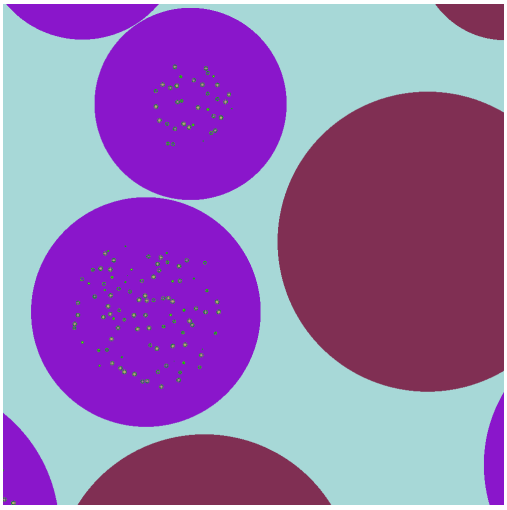


Figure 7.5: Detailed view of HTR-10 pebbles (OpenMC).

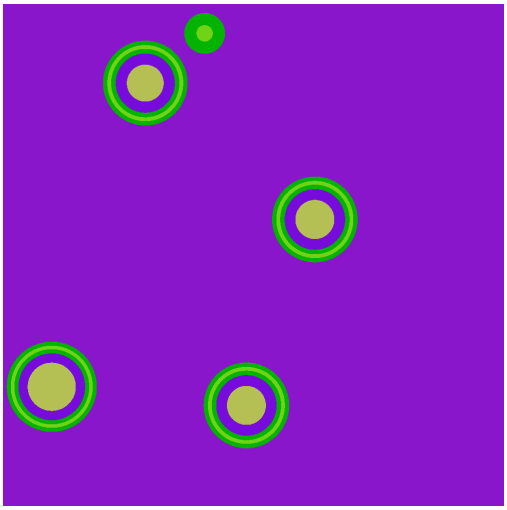


Figure 7.6: Detailed view of HTR-10 TRISO particles (OpenMC).

7.4 Comparison of the TRIPOLI-5 and OpenMC models

In order to verify the consistency between the TRIPOLI-5 and OpenMC models in terms of geometry and material composition, two checks are necessary:

- First, thanks to a converter tool developed at CEA, the whole geometry

generated by OpenMC can be exported, converted into the AGORA format, and subsequently tested with TRIPOLI-5. This yields perfectly mirrored geometries for TRIPOLI-5 and OpenMC, enabling a direct comparison of their material compositions using point-by-point or line-by-line tests [28].

- Second, leveraging the new features developed in CASTOR (see Chapter 3), the sphere centers for pebbles and TRISO particles generated by OpenMC can be read by CASTOR, in order to generate the corresponding AGORA file, to be imported into TRIPOLI-5. By ensuring the random pebble geometry is perfectly mirrored between the two models, this approach allows for the verification of the remaining (non-random) portion of the geometry.

Both tests were performed on a few configurations, and the results successfully confirmed the consistency between the two codes.

7.5 Monte Carlo simulations

We now present the results of the statistical neutronic analysis of the HTR-10 reactor, performed by running multiple Monte Carlo simulations with TRIPOLI-5 and OpenMC on the 'simplified core' models described above. The main observables for our analysis are the effective multiplication factor and the neutron flux.

For each tested configuration, the simulation results discussed in the following were systematically averaged over five independent random replicas, corresponding to independent realizations of the stochastic configurations of the pebble bed cores: the reported uncertainty derives from both the statistical convergence of Monte Carlo simulation (which for the effective multiplication factor lay systematically below 15 pcm) and the dispersion due to the stochastic nature of the configurations.

Each simulation was performed using 100000 neutrons and 1100 batches, where the first 100 were discarded as inactive. The ENDF/B-VIII.0 nuclear data library [56] was used, except where otherwise stated.

For the sake of simplicity, the random TRISO particle distribution within the fuel pebbles was chosen to be the same for all pebbles, which is known to have a negligible impact on the estimated multiplication factor [55].

We begin by considering the impact of randomness introduced by the CRP method, which affects the pebble positions (while preserving the total number of sampled pebbles). The configurations sampled by the CRP method were randomized by changing the seed for the CRP method of both CASTOR and OpenMC. The results obtained for the effective multiplication factor k_{eff} are shown in Tabs. 7.1 and 7.2, respectively.

Description	$k_{\mathbf{eff}}$
Replica 1	1.00201 ± 0.00012
Replica 2	1.00256 ± 0.00012
Replica 3	1.00193 ± 0.00013
Replica 4	1.00182 ± 0.00013
Replica 5	1.00190 ± 0.00013
-	-
Average	1.00204 ± 0.00017

Table 7.1: Effective multiplication factor k_{eff} achieved with TRIPOLI-5 coupled with CASTOR for five different replicas, by modifying only the pebble positions. The corresponding average over the replicas is also provided.

Description	$k_{\mathbf{eff}}$
Replica 1	1.00162 ± 0.00009
Replica 2	1.00198 ± 0.00010
Replica 3	1.00189 ± 0.00010
Replica 4	1.00165 ± 0.00010
Replica 5	1.00236 ± 0.00010
-	-
Average	1.00190 ± 0.00015

Table 7.2: Effective multiplication factor k_{eff} achieved with OpenMC for five different replicas, by modifying only the pebble positions. The corresponding average over the replicas is also provided.

The averages over the replicas yield $k_{\text{eff}} = 1.00204 \pm 0.00017$ for TRIPOLI-5 and $k_{\text{eff}} = 1.00190 \pm 0.00015$ for OpenMC, and suggest that the two models lead to statistically consistent results. Recall that the HTR-10 core as modeled with a cylinder having an equivalent height (provided in the benchmark specifications) is assumed to correspond to a critical configuration. The small offset of both models with respect to criticality (which is consistent with the typical deviations found for other criticality benchmarks) can be attributed to the technological uncertainties and to the impact of the nuclear data library, and is consistent between TRIPOLI-5 and OpenMC.

This finding is complemented by the analysis of the fundamental neutron flux produced by both codes, displayed in Figs. 7.7, 7.8. Fig. 7.9 illustrates the comparison between the average results obtained with the two codes.

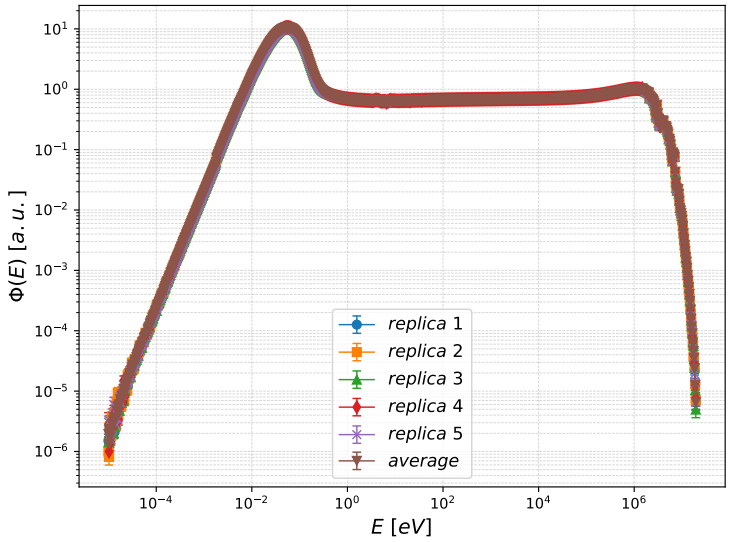


Figure 7.7: Fundamental neutron flux obtained with CASTOR coupled with TRIPOLI-5 for five different replicas. The corresponding average over the replicas is also provided.

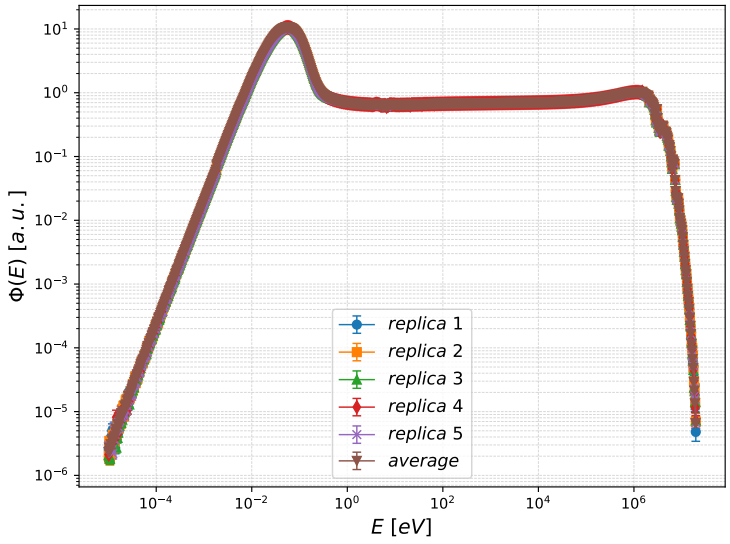


Figure 7.8: Fundamental neutron flux obtained with OpenMC for five different replicas. The corresponding average over the replicas is also provided.

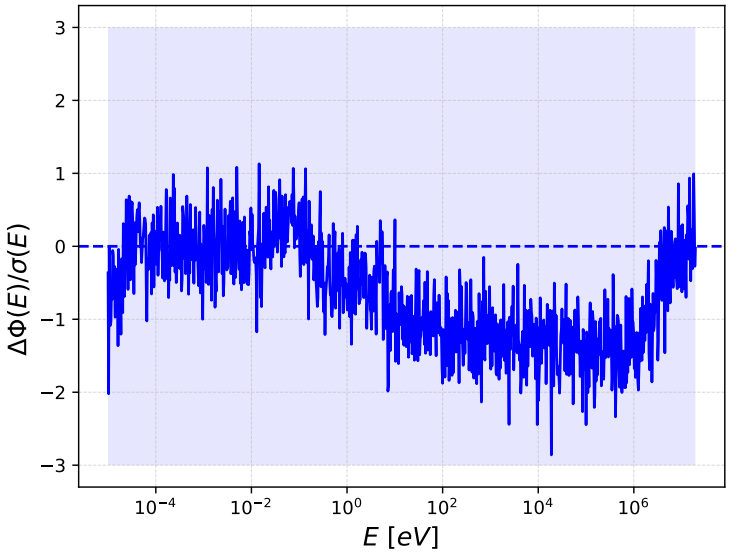


Figure 7.9: Fundamental neutron flux: comparison of the average values obtained with TRIPOLI-5 coupled with CASTOR and OpenMC with its native CRP model.

As observed for the effective multiplication factor, the fundamental mode profiles confirm that the results for the two models remain in excellent agreement, within statistical dispersion.

To investigate the effect of nuclear data, we performed a separate series of replicas for the reference configurations, using TRIPOLI-5 with the JEFF-3.3 library [57], which yields the results in Tab. 7.3.

Description	$k_{\mathbf{eff}}$
Replica 1	1.00101 ± 0.00013
Replica 2	1.00161 ± 0.00013
Replica 3	1.00128 ± 0.00012
Replica 4	1.00139 ± 0.00012
Replica 5	1.00086 ± 0.00013
-	-
Average	1.00123 ± 0.00017

Table 7.3: Effective multiplication factor k_{eff} obtained with TRIPOLI-5 and CASTOR for five different replicas, by modifying only the pebble positions. The corresponding average over the replicas is also provided. The nuclear data library is JEFF-3.3.

The comparison between the averages obtained with the two libraries (Tabs. 7.1 and 7.3) shows a discrepancy of the order of 100 pcm: $k_{\text{eff}} = 1.00123 \pm 0.00017$ using JEFF-3.3 versus $k_{\text{eff}} = 1.00204 \pm 0.00017$ using ENDF/B-VIII.0. This highlights the significant impact of the library choice on the results, as expected.

To isolate the effect of the random 'coloring' procedure from that of the random sphere positioning, we selected a single geometric realization generated with the Jodrey-Tory algorithm and resampled only the pebble types (keeping their positions fixed) over five independent replicas. The results obtained with TRIPOLI-5 are shown in Tab. 7.4.

Description	k_{eff}
Replica 1	1.00097 ± 0.00013
Replica 2	1.00207 ± 0.00013
Replica 3	1.00134 ± 0.00013
Replica 4	1.00096 ± 0.00013
Replica 5	1.00256 ± 0.00013
-	-
Average	1.00158 ± 0.00031

Table 7.4: Effective multiplication factor k_{eff} obtained with TRIPOLI-5 and CASTOR for five different replicas, by modifying only the coloring procedure. The corresponding average over the replicas is also provided.

These findings illustrate that the impact of the random pebble coloring ($k_{\text{eff}} = 1.00158 \pm 0.00031$) is statistically significant, introducing a dispersion of the order of 30 pcm, although smaller than the effect of nuclear libraries. Note that the shift in the mean value compared to the reference case ($k_{\text{eff}} = 1.00204 \pm 0.00017$) is entirely expected, as it simply reflects the reactivity of the specific geometric realization that was kept fixed during this resampling.

The effect of the packing fraction was quantified by adjusting the equivalent height of the core while keeping the number of fuel and graphite pebbles fixed. A packing fraction $\xi = 0.62$ was achieved by setting the equivalent height $h_{eq} = 121.075$ cm. The results obtained with TRIPOLI-5 are provided in Tab. 7.5.

Description	$k_{\mathbf{eff}}$
Replica 1	1.00528 ± 0.00012
Replica 2	1.00562 ± 0.00013
Replica 3	1.00581 ± 0.00012
Replica 4	1.00565 ± 0.00013
Replica 5	1.00568 ± 0.00013
-	-
Average	1.00561 ± 0.00015

Table 7.5: Effective multiplication factor k_{eff} obtained using TRIPOLI-5 with CASTOR for five different replicas, varying only pebble positions, with packing fraction $\xi = 0.62$ and $h_{eq} = 121.075$ cm. The corresponding average over the replicas is also provided.

Conversely, a packing fraction $\xi = 0.60$ was achieved by setting the equivalent height $h_{eq} = 125.111$ cm, which yields the results in Tab. 7.6.

Description	$k_{\mathbf{eff}}$
Replica 1	0.99751 ± 0.00013
Replica 2	0.99767 ± 0.00013
Replica 3	0.99764 ± 0.00012
Replica 4	0.99743 ± 0.00013
Replica 5	0.99767 ± 0.00013
-	-
Average	0.99758 ± 0.00014

Table 7.6: Effective multiplication factor k_{eff} obtained using TRIPOLI-5 with CASTOR for five different replicas, varying only pebble positions, with packing fraction $\xi = 0.60$ and $h_{eq} = 125.111$ cm. The corresponding average over the replicas is also provided.

The sensitivity of k_{eff} to the pebble packing fraction is rather high: a $\pm 1\%$ variation of the packing fraction results in a k_{eff} deviation of approximately ± 400 pcm.

The findings for these two latter cases are complemented by the analysis of the corresponding fundamental neutron flux, which is displayed in Figs. 7.10 and 7.11. Statistically significant deviations are observed; this was expected, since a different distribution of pebbles was used.

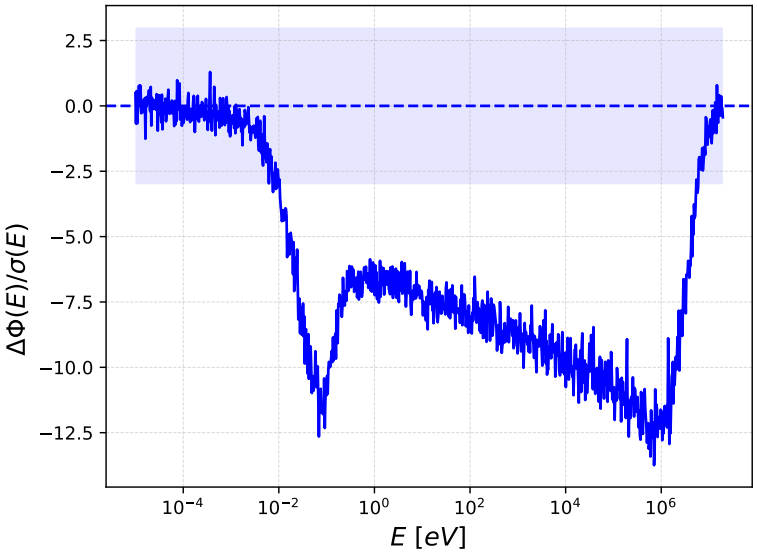


Figure 7.10: Fundamental neutron flux comparison between averages obtained with TRIPOLI-5 and CASTOR, with packing fraction ξ equal to 0.61 and 0.60.

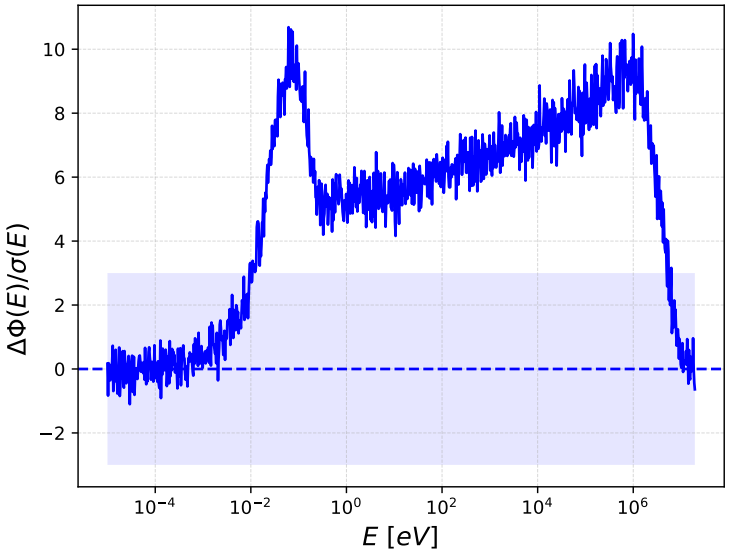


Figure 7.11: Fundamental neutron flux comparison between averages obtained with TRIPOLI-5 and CASTOR, with packing fraction ξ equal to 0.61 and 0.62.

Finally, the impact of the number of TRISO particles in the fuel pebbles was assessed by increasing their packing fraction from 0.05025 (8335 particles) to 0.05125 (8501 particles). The results obtained with TRIPOLI-5 are provided in Tab. 7.7.

Description	$k_{\mathbf{eff}}$
Replica 1	1.00610 ± 0.00013
Replica 2	1.00568 ± 0.00013
Replica 3	1.00607 ± 0.00013
Replica 4	1.00564 ± 0.00012
Replica 5	1.00549 ± 0.00013
-	-
Average	1.00580 ± 0.00017

Table 7.7: Effective multiplication factor k_{eff} obtained with TRIPOLI-5 and CASTOR for five different replicas, by modifying only the pebble positions, with 8501 TRISO particles into each pebble. The corresponding average over the replicas is also provided.

Conversely, decreasing the packing fraction of the TRISO particles to 0.04925 (8169 particles) yields the results in Tab. 7.8.

Description	$k_{\mathbf{eff}}$
Replica 1	0.99804 ± 0.00012
Replica 2	0.99813 ± 0.00013
Replica 3	0.99790 ± 0.00012
Replica 4	0.99812 ± 0.00013
Replica 5	0.99811 ± 0.00013
-	-
Average	0.99806 ± 0.00013

Table 7.8: Effective multiplication factor k_{eff} obtained with TRIPOLI-5 and CASTOR for five different replicas, by modifying only the pebble positions, with 8169 TRISO particles into each pebble. The corresponding average over the replicas is also provided.

The average reactivity worth of each TRISO particle is estimated to be about 1.4×10^{-4} pcm.

Chapter 8

Application of DEM to the modeling of a realistic sphere packing

8.1 General features

Following the investigations on the simplified model, this chapter extends the neutronic analysis of the HTR-10 reactor to its 'realistic core geometry', as defined in the benchmark specifications [55]. To accurately reproduce the pebble arrangement within the core, the DEM code LIGGGHTS is employed, coupled with TRIPOLI-5 for the subsequent Monte Carlo transport calculations.

This transition requires a different approach for the core regions. Referring to Fig. 6.2, regions 99, 5, 91, and 6 are merged to form a single container, used for LIGGGHTS simulations, with the aim of modeling the pebbles dispersed in air.

8.2 DEM input file for pebble sampling

To sample the pebble configuration in the core, the first step concerns the configuration of the LIGGGHTS input file, whose basic structures have been presented in Chapter 3. The main differences to be remarked are listed below.

First, as for the time-step choice, due to the total number of spheres inserted (around 20000), this parameter was set to $80\mu s$, based on the results of Chapter 3.

Second, for the realistic model of HTR-10 provided in the benchmark description, the total number of pebbles is not fixed. The quantity that is fixed is the number of pebbles whose centers belong to the top core cavity region (16890 according to the specifications). Therefore, we relied on a 'while loop' that terminates the

insertion of the pebbles once the target is reached.

Finally, explicit wall properties at the boundaries of the core were added, as specified later, in order to obtain a more accurate analysis. Regarding this, one of the most important challenges concerned the conical shape of the discharge region, since LIGGGHTS is able to reproduce reflecting boundary conditions for walls only in the case of simple primitive shapes, such as planes and cylinders. To accurately model the complex shape of the core and discharge region, it was thus necessary to resort to a surface meshing: for this purpose, the Salome code [58] was used. The meshing refinement chosen is a compromise between the target accuracy of the resulting shape and the simulation time for the pebble dynamics. Figure 8.1 illustrates the final version of the mesh that we selected.

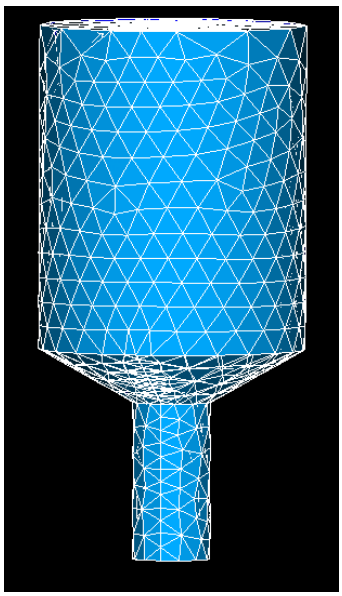
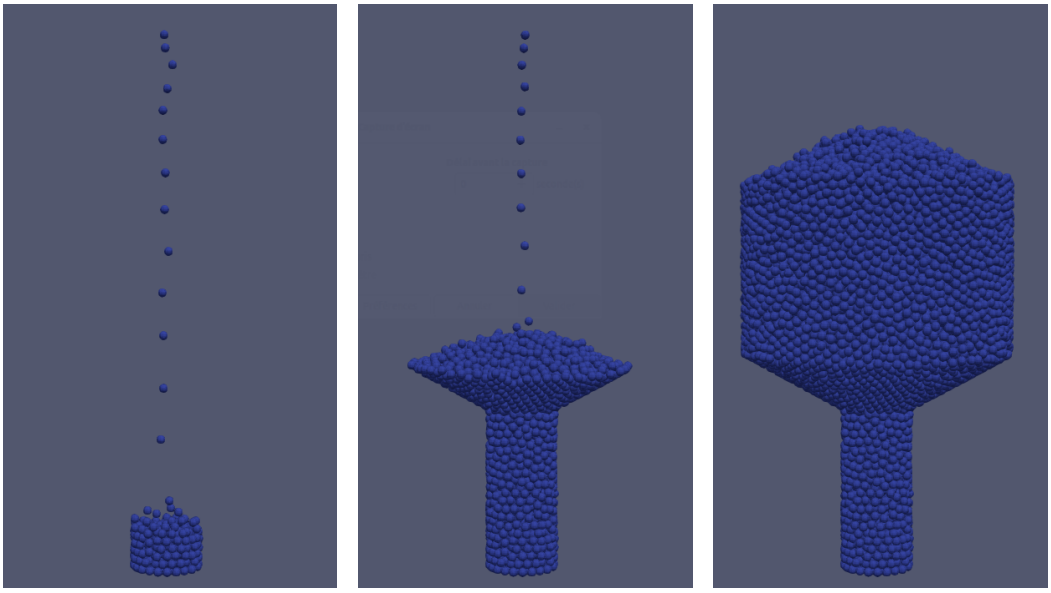


Figure 8.1: Mesh of the container, realized with Salome.

The physical parameters adopted for the LIGGGHTS simulations are based on the literature [50, 51, 59], and are summarized in Tab. 8.1. For illustration, Fig. 8.2 shows the filling process of the HTR-10 container as simulated by LIGGGHTS, at several time steps.

Parameter	Value
Radius of pebbles	3 cm
Density of pebbles	1850 kg/m^3
Young's modulus: pebbles	10^9 Pa
Young's modulus: wall	10^{10} Pa
Poisson's ratio: pebbles	0.2
Poisson's ratio: wall	0.3
Restitution coefficient: pebble-pebble	0.9
Restitution coefficient: pebble-wall	0.3
Friction coefficient: pebble-pebble	0.2
Friction coefficient: pebble-wall	0.2

Table 8.1: Data for the LIGGGHTS simulations needed for the pebble bed configuration of the HTR-10.



(a) 320 pebbles inserted **(b)** 3200 pebbles inserted **(c)** 16890 pebbles inserted above the discharge cone

Figure 8.2: HTR-10 filling process using LIGGGHTS, at different steps.

8.3 Importing and processing the pebbles in CASTOR

After the generation of the pebble centers (with their respective radii) using LIGGGHTS, it is necessary to fill them, including any required external layers, and to produce the AGORA file, to be imported into the TRIPOLI-5 geometry model.

Leveraging the developments discussed in Chapters 3 and 5, the spheres generated by LIGGGHTS can be imported into CASTOR. The code then takes care of the creation of the 'fillers', which can differ among pebbles, according to a probability distribution. In particular, as seen in previous chapters, it is possible to distinguish between dummy pebbles (containing only graphite) and fuel pebbles (containing TRISO particles embedded into a graphite matrix), where the distribution of TRISO particles is also performed by the CASTOR utilities.

According to the benchmark specifications, the HTR-10 core presents a space-varying pebble distribution: regions 6 and 91 are filled only with dummy pebbles, while regions 5 and 99 are filled with a random mixture of fuel and dummy pebbles (exactly 16890 pebbles in total, including 9627 fuel pebbles). To handle this requirement, a specific space-dependent 'coloring' procedure for the pebbles was developed in CASTOR.

The final core configuration corresponding to a single realization sampled using CASTOR and exported to TRIPOLI-5 is shown in Fig. 8.3.

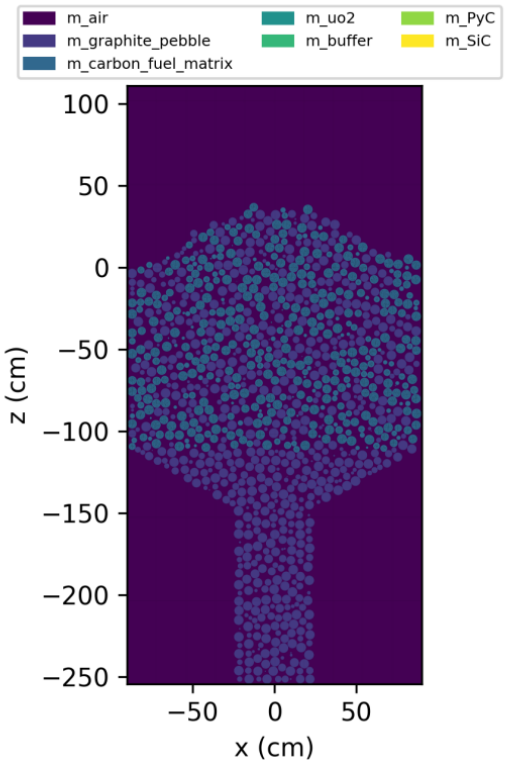


Figure 8.3: DEM-based HTR-10 core configuration for TRIPOLI-5: the pebble positions were determined using LIGGGHTS and the 'coloring' procedure was carried out by CASTOR.

The random portion of the geometry model is then joined to the non-random portion of the TRIPOLI-5 model (which is essentially the same as for the case of the simplified HTR-10 model using CRP, except for the modifications applied to the core container and to the discharge region). The full model of the HTR-10 reactor is shown in Fig. 8.4.

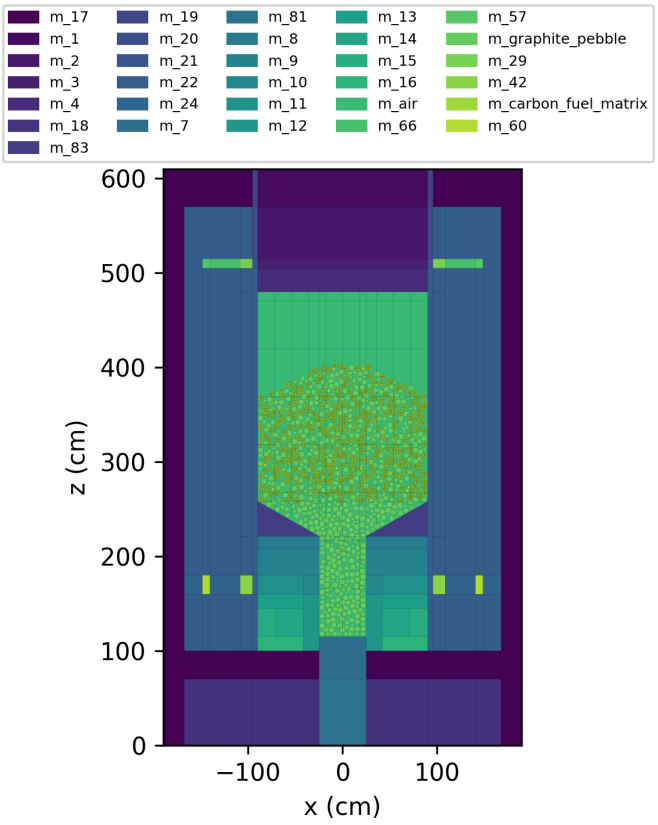


Figure 8.4: Section of the full model of HTR-10 for TRIPOLI-5. The sphere packing was generated using LIGGGHTS, and the 'coloring' using CASTOR.

8.4 Monte Carlo simulations

We now present and discuss the results of the statistical neutronic analysis, performed by running multiple Monte Carlo simulations with TRIPOLI-5 on the realistic model of the HTR-10 reactor. The key observables are again the effective multiplication factor and the neutron flux. The goal is to compare the findings of the detailed model to those obtained in the previous chapter using the simplified model based on the CRP approach and the equivalent core height.

For each tested configuration, the simulation results discussed below were systematically averaged over five independent random replicas: the reported uncertainty derives from both the statistical convergence of Monte Carlo and the dispersion due to the stochastic nature of the configurations.

Each simulation was performed using 100000 neutrons and 1100 batches, where the first 100 were discarded as inactive. Again, the ENDF/B-VIII.0 nuclear data library [56] was used, except where otherwise stated.

As in the previous chapter, the random TRISO configuration within the fuel pebbles was chosen to be identical for all pebbles.

We begin by considering the impact of the randomness introduced by the pebble dynamics simulated by LIGGGHTS on the core filling. In the DEM model, spheres are generated into the insertion region with random positions and velocities, and subsequently their motion under gravity and their interactions with the system are simulated. The final pebble configuration ultimately depends on the random seed governing these initial conditions. Therefore, the configurations sampled by LIGGGHTS were randomized by changing this seed. The corresponding results obtained with TRIPOLI-5 for the effective multiplication factor k_{eff} are shown in Tab. 8.2.

Description	k_{eff}
Replica 1	1.00333 ± 0.00013
Replica 2	1.00254 ± 0.00012
Replica 3	1.00320 ± 0.00012
Replica 4	1.00252 ± 0.00013
Replica 5	1.00231 ± 0.00013
-	-
Average	1.00278 ± 0.00022

Table 8.2: Effective multiplication factor k_{eff} estimated with TRIPOLI-5 and LIGGGHTS for five different replicas, obtained modifying only the pebble positions. The corresponding average over the replicas is also provided.

The final average for the detailed model is rather close to the value obtained using the CRP and the simplified model, as given in Tab. 7.1: $k_{\text{eff}} = 1.00278 \pm 0.00022$ for the detailed model using LIGGGHTS versus $k_{\text{eff}} = 1.00204 \pm 0.00017$ for the simplified model using the CRP.

The corresponding fundamental mode and the comparison with the case of CRP (see Fig. 7.7) are shown in Figs. 8.5 and 8.6, respectively. Interestingly, although the two models yield relatively close results for the reactivity, they exhibit a noticeable difference in their neutron flux profiles.

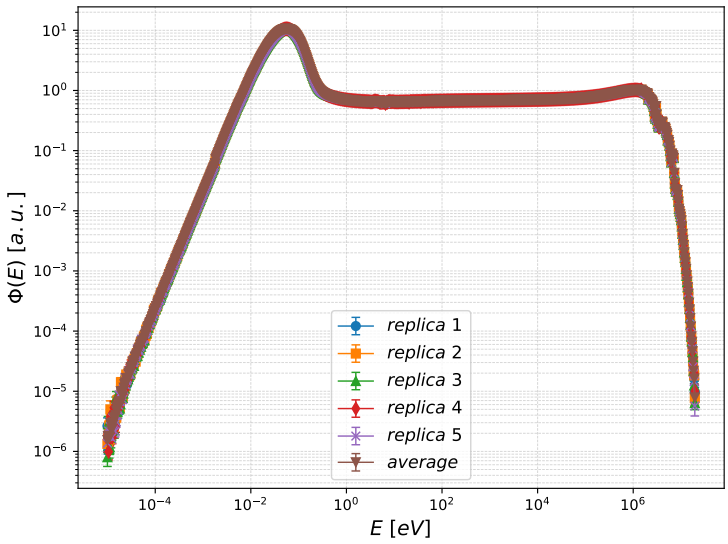


Figure 8.5: Fundamental neutron flux obtained with TRIPOLI-5 and LIGGGHTS for five different replicas. The corresponding average over the replicas is also provided.

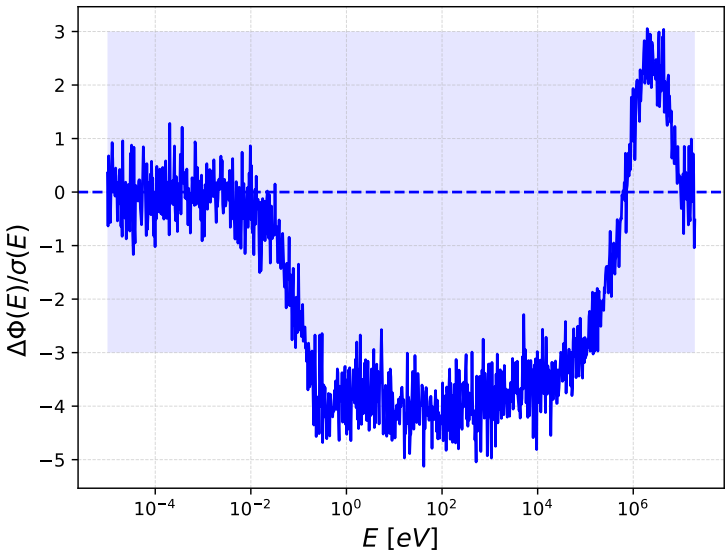


Figure 8.6: Fundamental neutron flux comparison between averages obtained with TRIPOLI-5 using the detailed model (with LIGGGHTS) or the simplified model (with CRP).

Following the approach used for the simplified model analysis, we performed a separate series of simulations for the reference configuration, using TRIPOLI-5 with the JEFF-3.3 library [57], yielding the results presented in Tab. 8.3.

Description	$k_{\mathbf{eff}}$
Replica 1	1.00234 ± 0.00012
Replica 2	1.00167 ± 0.00013
Replica 3	1.00275 ± 0.00012
Replica 4	1.00152 ± 0.00012
Replica 5	1.00146 ± 0.00013
-	-
Average	1.00195 ± 0.00026

Table 8.3: Effective multiplication factor k_{eff} estimated with TRIPOLI-5 and LIGGGHTS for five different replicas, obtained modifying only the pebble positions. The corresponding average over the replicas is also provided. The nuclear data library is JEFF-3.3.

Consistent with the findings of the previous chapter, changing the nuclear data library leads to a discrepancy of the order of 100 pcm: $k_{\text{eff}} = 1.00195 \pm 0.00026$ for JEFF-3.3 versus $k_{\text{eff}} = 1.00278 \pm 0.00022$ for ENDF/B-VIII.0.

Similarly to the analysis conducted in Chapter 7, the effect of the random 'coloring' procedure was singled out with respect to that of random positioning of the spheres, by selecting a given realization sampled with LIGGGHTS and resampling only the pebble types. The corresponding findings obtained with TRIPOLI-5 are given in Tab. 8.4.

Description	$k_{\mathbf{eff}}$
Replica 1	1.00234 ± 0.00012
Replica 2	1.00140 ± 0.00013
Replica 3	1.00225 ± 0.00013
Replica 4	1.00190 ± 0.00012
Replica 5	1.00146 ± 0.00013
-	-
Average	1.00187 ± 0.00021

Table 8.4: Effective multiplication factor k_{eff} estimated with TRIPOLI-5 and LIGGGHTS for five different replicas, obtained modifying only the coloring procedure. The corresponding average over the replicas is also provided.

Consistent with the simplified core model using CRP, these findings show that the impact of the random pebble coloring ($k_{\text{eff}} = 1.00187 \pm 0.00021$) is statistically significant, introducing a dispersion of approximately 20 pcm, although smaller than the effect of nuclear libraries. Note that, as already discussed, the shift in the mean value compared to the reference case ($k_{\text{eff}} = 1.00278 \pm 0.00022$) is expected, reflecting the reactivity of the specific geometric realization that was kept fixed during this resampling.

The effect of friction, which is known to be one of the parameters having the strongest impact on the DEM dynamics for pebbles, was quantified by modifying the pebble-pebble friction coefficient to 0.3. (Recall that in DEM simulations, the packing fraction cannot be explicitly enforced, but emerges as a natural consequence of the system's physical dynamics). As a result, the average packing fraction decreased to $\xi = 0.598$, and the simulations show an appreciable decrease in the effective multiplication factor, as illustrated in Tab. 8.5.

Description	k_{eff}
Replica 1	1.00059 ± 0.00013
Replica 2	0.99991 ± 0.00013
Replica 3	1.00001 ± 0.00012
Replica 4	1.00026 ± 0.00013
Replica 5	1.00009 ± 0.00013
-	-
Average	1.00017 ± 0.00017

Table 8.5: Effective multiplication factor k_{eff} estimated with TRIPOLI-5 and LIGGGHTS for five different replicas, obtained modifying only the pebble positions, with a sphere-sphere friction coefficient modified to 0.3 with respect to the value given in Tab. 8.1. The corresponding average over the replicas is also provided.

This last finding is complemented by the analysis of the corresponding fundamental mode compared to the reference case, which is displayed in Fig. 8.7. The flux exhibits statistically significant deviations, as expected, since modifying the friction parameter leads to a different spatial distribution of the spheres.

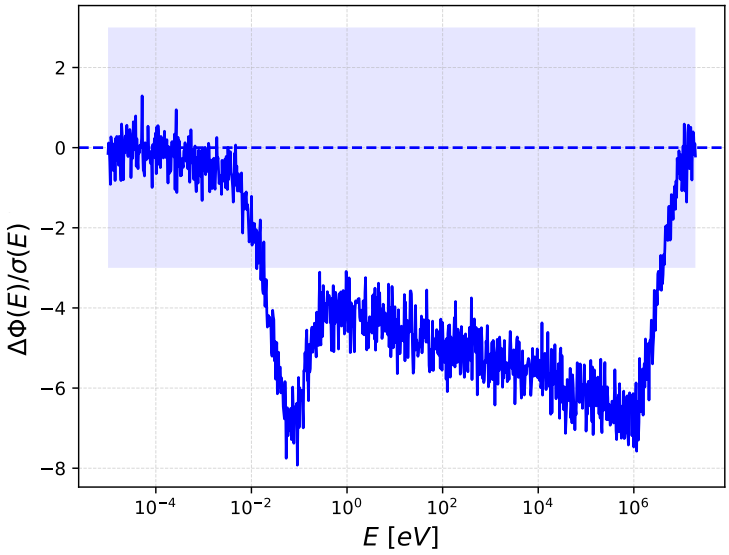


Figure 8.7: Comparison of the fundamental neutron flux estimated with TRIPOLI-5 and LIGGGHTS with a pebble-pebble friction coefficient of 0.2 and 0.3.

The effect of the pebble-wall friction was also investigated by modifying the corresponding friction coefficient to 0.3. In this case, the corresponding multiplication factor displays negligible deviations from the reference value, as shown in Tab. 8.6.

Description	$k_{\mathbf{eff}}$
Replica 1	1.00280 ± 0.00013
Replica 2	1.00252 ± 0.00013
Replica 3	1.00270 ± 0.00013
Replica 4	1.00311 ± 0.00013
Replica 5	1.00239 ± 0.00013
-	-
Average	1.00270 ± 0.00017

Table 8.6: Effective multiplication factor k_{eff} estimated with TRIPOLI-5 and LIGGGHTS for five different replicas, obtained modifying only the pebble positions, with a sphere-wall friction coefficient modified to 0.3 with respect to table 8.1. The corresponding average over the replicas is also provided.

The reactivity worth of individual pebbles was assessed by pouring ten additional and ten fewer (mixed) pebbles compared to the target value of 16890. For the

addition of ten pebbles, the results are given in Tab. 8.7.

Description	$k_{\mathbf{eff}}$
Replica 1	1.00338 ± 0.00013
Replica 2	1.00253 ± 0.00012
Replica 3	1.00372 ± 0.00012
Replica 4	1.00291 ± 0.00013
Replica 5	1.00267 ± 0.00013
-	-
Average	1.00304 ± 0.00024

Table 8.7: Effective multiplication factor k_{eff} estimated with TRIPOLI-5 and LIGGGHTS for five different replicas, obtained modifying only the pebble positions, with 16900 pebbles into the core instead of 16890. The corresponding average over the replicas is also provided.

For the removal of ten pebbles, the results are given in Tab. 8.8.

Description	$k_{\mathbf{eff}}$
Replica 1	1.00292 ± 0.00013
Replica 2	1.00238 ± 0.00012
Replica 3	1.00289 ± 0.00012
Replica 4	1.00243 ± 0.00013
Replica 5	1.00234 ± 0.00013
-	-
Average	1.00259 ± 0.00017

Table 8.8: Effective multiplication factor k_{eff} estimated with TRIPOLI-5 and LIGGGHTS for five different replicas, obtained modifying only the pebble positions, with 16880 pebbles into the core instead of 16890. The corresponding average over the replicas is also provided.

Based on these findings, the average reactivity worth of a single pebble is estimated to be approximately 2 pcm, although better statistics would be needed.

Finally, the impact of the number of TRISO particles in the fuel pebbles was again assessed by increasing and decreasing their packing fraction by the same amount as in the previous chapter. The results are shown in Tabs. 8.9 and 8.10. The reactivity effect closely matches the findings relative to the simplified core model using the CRP.

Description	$k_{\mathbf{eff}}$
Replica 1	1.00697 ± 0.00013
Replica 2	1.00617 ± 0.00013
Replica 3	1.00713 ± 0.00012
Replica 4	1.00626 ± 0.00013
Replica 5	1.00616 ± 0.00013
-	-
Average	1.00654 ± 0.00023

Table 8.9: Effective multiplication factor k_{eff} estimated with TRIPOLI-5 and LIGGGHTS for five different replicas, obtained modifying only the pebble positions, with 8501 TRISO into each pebble. The corresponding average over the replicas is also provided.

Description	$k_{\mathbf{eff}}$
Replica 1	0.99939 ± 0.00013
Replica 2	0.99858 ± 0.00013
Replica 3	0.99952 ± 0.00013
Replica 4	0.99859 ± 0.00012
Replica 5	0.99836 ± 0.00012
-	-
Average	0.99889 ± 0.00025

Table 8.10: Effective multiplication factor k_{eff} estimated with TRIPOLI-5 and LIGGGHTS for five different replicas, obtained modifying only the pebble positions, with 8169 TRISO into each pebble. The corresponding average over the replicas is also provided.

Chapter 9

Analysis of the simulation time

The computer times of the Monte Carlo simulations considered in the previous chapters are now compared. Only the time necessary to perform the Monte Carlo simulation is considered, without accounting for the one necessary for the generation of the geometry.

Due to the randomness of the simulation time, we decided to consider the average over five different simulations for each code, realized over independent geometries. Each simulation is run over 1100 batches, 35 threads (using the same machine for a fair comparison), and 10000 neutrons (this number has been decreased to speed up the simulations, since a linear behavior with respect to the number of neutrons is expected). Our findings are shown in Tabs. 9.1, 9.2 and 9.3

Description	$k_{\mathbf{eff}}$
Replica 1	11545 seconds
Replica 2	11568 seconds
Replica 3	11566 seconds
Replica 4	11489 seconds
Replica 5	11478 seconds
-	-
Average	11529 ± 19 seconds

Table 9.1: Simulation time achieved with TRIPOLI-5 for the simplified model using CRP. Five different replicas are considered, obtained modifying only the pebble positions. The average over the replicas is also provided.

Description	$k_{\mathbf{eff}}$
Replica 1	6921 seconds
Replica 2	6923 seconds
Replica 3	6964 seconds
Replica 4	6913 seconds
Replica 5	6940 seconds
-	-
Average	6932 ± 9 seconds

Table 9.2: Simulation time achieved with OpenMC for the simplified model using CRP. Five different replicas are considered, obtained modifying only the pebble positions. The average over the replicas is also provided.

Description	$k_{\mathbf{eff}}$
Replica 1	11942 seconds
Replica 2	11859 seconds
Replica 3	11927 seconds
Replica 4	11859 seconds
Replica 5	11824 seconds
-	-
Average	11882 ± 22 seconds

Table 9.3: Simulation time achieved with TRIPOLI-5 using the detailed model generated with LIGGGHTS. Five different replicas are considered, obtained modifying only the pebble positions. The average over the replicas is also provided.

Based on these findings, we observe that OpenMC is faster than TRIPOLI-5, which is expected since the development of TRIPOLI-5 started in 2022 and the code is still undergoing optimization for particle tracking routines. Only small differences are found between the detailed and the simplified core models simulated with TRIPOLI-5.

Chapter 10

Conclusions

In this MSc work, we addressed the Monte Carlo simulation of neutron transport in HTR cores displaying a 'double heterogeneity' configuration, with randomly dispersed pebbles each containing randomly dispersed TRISO particles. In order to achieve this goal, several tasks were accomplished, concerning both the possibility of sampling realistic pebble bed configurations using distinct models, and the calculations of reactor parameters of interest for the HTR-10 benchmark configuration.

In the first part of this work, we developed new features in the CASTOR random media sampler. In particular, thanks to these developments, we

- Implemented the Jodrey-Tory algorithm to sample spherical inclusions with a packing fraction up to 0.65; this was a primary requirement for the modeling of realistic HTR pebble bed cores.
- Coupled CASTOR with an external DEM code capable of generating a high-fidelity sphere packing using particle dynamics. For this purpose, we relied on the DEM code LIGGGHTS. This feature can also be usefully applied to read the CRP sphere packing generated by OpenMC. Once imported in CASTOR, these configurations can be displayed, processed, and analyzed using statistical tools such as the calculation of the chord length distribution.
- Sampled realistic geometries of pebble bed reactors, including space-varying distribution of pebbles, filled with graphite or by a 'universe' containing a distribution of TRISO particles embedded in a graphite matrix, and the associated multi-layered structures.

The CRP algorithm implemented in CASTOR was thoroughly verified against the equivalent algorithm available in OpenMC.

In the second part of this work, we built and simulated two versions of the HTR-10 benchmark model: one with a simplified core using CRP for the pebble positions,

and one using a detailed description of the pebble configurations generated using LIGGGHTS.

Simulations were carried out using the Monte Carlo code TRIPOLI-5 and, whenever possible, OpenMC for comparison. A detailed analysis of the simulated models was conducted, aimed at determining the most influential factors on the key observables, namely the effective multiplication factor and the neutron flux.

When TRIPOLI-5 and OpenMC both use CRP, the simulation findings for the simplified HTR-10 configurations are in very good statistical agreement. Conversely, statistically significant discrepancies for the effective multiplication factor and for the neutron flux were detected between the simplified and detailed core models as simulated with TRIPOLI-5, as expected. Specifically, the transition from the simplified approach to the realistic one introduced a reactivity difference of approximately 70-80 pcm. More importantly, while the integral reactivity remained relatively consistent between the two approaches, the DEM configuration exhibited noticeable deviations in the local spatial profile of the neutron flux.

The choice of the nuclear data library, the 'coloring' process, the packing process, the number of pebbles, the number of TRISO particles per pebble, and the physical parameters of the DEM simulations all play a role in the obtained results. Finally, the simulation time was compared between the different codes, showing a small difference between LIGGGHTS and CASTOR for TRIPOLI-5, due to the different pebble distribution. OpenMC is faster than TRIPOLI-5 for particle tracking, which is attributed to the fact that TRIPOLI-5 is not yet fully optimized for these specific geometries.

Future work will concern a more detailed analysis of the differences between the sphere packing generated by the Jodrey-Tory algorithm and by the DEM codes: it would be interesting to investigate, for instance, the connection between the chord length distribution in pebble beds and the particle-transport properties [28]. Also, the impact on reactivity due to the choice of the physics of granular media might be further assessed by coupling CASTOR with other DEM codes, such as ExaDEM, an HPC software solution built on the ExaNBody platform [60].

Bibliography

[1] U.S. Department of Energy. *DOE Fundamentals Handbook: Nuclear Physics and Reactor Theory, Volume 1 of 2*. DOE-HDBK-1019/1-93. Training Manual. Springfield, VA, 1993 (cit. on pp. 3–5).

[2] ScienceReady. *Nuclear Fission, Uncontrolled and Controlled Chain Reactions*. URL: `https://scienceready.com.au/pages/nuclear-fission-and-reactors` (cit. on p. 3).

[3] World Nuclear Association. *Physics of Uranium and Nuclear Energy*. URL: `https://world-nuclear.org/information-library/nuclear-fuel-cycle/introduction/physics-of-nuclear-energy` (cit. on p. 4).

[4] B. Zohuri. «Modeling Neutron Transport and Interactions». In: *Neutronic Analysis for Nuclear Reactor Systems*. 1st. Springer International Publishing, Cham, 2017, pp. 53–101 (cit. on pp. 6, 7).

[5] U.S. Department of Energy. *DOE Fundamentals Handbook: Nuclear Physics and Reactor Theory, Volume 2 of 2*. DOE-HDBK-1019/2-93. Training Manual. Springfield, VA, 1993 (cit. on p. 6).

[6] World Nuclear Association. *Nuclear Power Reactors*. 2025. URL: `https://world-nuclear.org/information-library/nuclear-fuel-cycle/nuclear-power-reactors/nuclear-power-reactors` (cit. on p. 7).

[7] G. Farkas. *From Gen I to Gen III*. 2010. URL: `https://inis.iaea.org/records/tvn9r-pwq12` (cit. on p. 9).

[8] Nuclear Energy Agency (NEA). *High-temperature Gas-cooled Reactors and Industrial Heat Applications*. NEA No. 7629. Paris: OECD Publishing, 2022 (cit. on pp. 10, 14, 16).

[9] M. T. Simnad. «The early history of high-temperature helium gas-cooled nuclear power reactors». In: *Annals of Nuclear Energy* 16 (1991), pp. 25–32 (cit. on p. 10).

[10] H.D. Gougar. «HTR History». In: *Idaho National Laboratory* (2019) (cit. on pp. 10, 11).

[11] J. Fu et al. «Overview and Progress of High Temperature Reactor Pebble-bed Module Demonstration Project (HTR-PM)». In: Tsinghua University, 2014, p. 11 (cit. on p. 11).

[12] *HTGR Schematic Diagram*. 2025. URL: `https://www.energyencyclopedia.com/en/free-downloads` (cit. on p. 12).

[13] D. Wang et al. «Thermodynamics of GT-MHR-250 modular nuclear plant with helium reactor and gas turbine based on the complex Brayton cycle». In: *Annals of Nuclear Energy* 39 (2023), p. 101686 (cit. on pp. 12, 16).

[14] X. Zhou et al. «Nuclear graphite for high temperature gas-cooled reactors». In: *Annals of Nuclear Energy* 32 (2017), pp. 193–204 (cit. on pp. 13, 16, 17).

[15] P. A. Demkowicz et al. «Coated particle fuel: Historical perspectives and current progress». In: *Annals of Nuclear Energy* 515 (2019), pp. 434–450 (cit. on pp. 13, 16).

[16] W. F. Skerjanc and G. J. Youinou. «High uranium loading TRISO particle for microreactor applications». In: *Annals of Nuclear Energy* 414 (2023) (cit. on p. 14).

[17] K.S. Mehta et al. «Neutronics Analysis on High-Temperature Gas-Cooled Pebble Bed Reactors by Coupling Monte Carlo Method and Discrete Element Method». In: *Energies* 17.20 (2024), p. 5188 (cit. on p. 15).

[18] D. Hartantoa and P.H. Liem. «Physics study of block/prismatic-type HTGR design option for the Indonesian Experimental Power Reactor (RDE)». In: *Annals of Nuclear Energy* 368 (2020), p. 110821 (cit. on p. 15).

[19] J. J Iwatsuki et al. «Overview of high temperature gas-cooled reactor». In: *Annals of Nuclear Energy* 5 (2021), pp. 1–16 (cit. on p. 16).

[20] K. Kiegiel et al. «Advanced Nuclear Reactors—Challenges Related to the Reprocessing of Spent Nuclear Fuel». In: *Energies* 18.15 (2025), p. 4080 (cit. on p. 17).

[21] I. Lux et al. *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*. Boca Raton, Florida: CRC Press, 1991 (cit. on p. 17).

[22] G.C. Pomraning. *Linear Kinetic Theory and Particle Transport in Stochastic Mixtures*. River Edge, USA: World Scientific Publishing, 1991 (cit. on pp. 17, 41).

[23] S. Torquato. *Random Heterogeneous Materials: Microstructure and Macroscopic Properties*. Vol. 16. Interdisciplinary Applied Mathematics. New York, USA: Springer-Verlag, 2013 (cit. on pp. 17, 41).

[24] C. Larmier et al. «Neutron multiplication in random media: Reactivity and kinetics parameters». In: *Annals of Nuclear Energy* 111 (2018), pp. 391–403 (cit. on p. 17).

[25] C. Larmier et al. «Benchmark solutions for transport in d-dimensional Markov binary mixtures». In: *Journal of Quantitative Spectroscopy and Radiative Transfer* 189 (2017), pp. 133–148 (cit. on p. 17).

[26] C. Larmier et al. «Finite-size effects and percolation properties of Poisson geometries». In: *Physical Review E* 94.1 (2016), p. 012130 (cit. on pp. 17, 41).

[27] C. Larmier et al. «Monte Carlo particle transport in random media: The effects of mixing statistics». In: *Journal of Quantitative Spectroscopy and Radiative Transfer* 196 (2017), pp. 270–286 (cit. on p. 17).

[28] C. Larmier, M. A. Kowalski, and A. Zoia. «CASTOR, a Monte Carlo Sampler of Random Media for Particle Transport Applications». In: *Proceedings of M&C 2023* (2023) (cit. on pp. 18, 19, 23, 36, 72, 98).

[29] G. C. Pomraning. *Linear kinetic theory and particle transport in stochastic mixtures*. NJ, USA: World Scientific Publishing, 1991 (cit. on p. 18).

[30] D. C. Sahni. «Equivalence of generic equation method and the phenomenological model for linear transport problems in a two-state random scattering medium». In: *J. Math. Phys.* 30 (1989), pp. 1554–1559 (cit. on p. 18).

[31] D. C. Sahni. «An application of reactor noise techniques to neutron transport problems in a random medium». In: *Ann. Nucl. Energy* 16 (1989), pp. 397–408 (cit. on p. 18).

[32] G. B. Zimmerman and M. L. Adams. «Algorithms for Monte Carlo particle transport in binary statistical mixtures». In: *Trans. Am. Nucl. Soc.* 66 (1991), p. 287 (cit. on p. 18).

[33] T. J. Donovan and Y. Danon. «Application of Monte Carlo chord-length sampling algorithms to transport through a two-dimensional binary stochastic mixture». In: *Nucl. Sci. Eng.* 143 (2003), pp. 226–239 (cit. on p. 18).

[34] T. J. Donovan et al. «Implementation of chord length sampling for transport through a binary stochastic mixture». In: *Proceedings of M&C2003*. Gatlinburg, TN, 2003 (cit. on p. 18).

[35] C Larmier et al. «Poisson-Box Sampling algorithms for three-dimensional Markov binary mixtures». In: *Journal of Quantitative Spectroscopy and Radiative Transfer* 206 (2018), pp. 70–82 (cit. on p. 18).

[36] P. S. Brantley and G. B. Zimmerman. «Benchmark comparison of Monte Carlo algorithms for three-dimensional binary stochastic media». In: *Trans. Am. Nuc. Soc.* 117 (2017), pp. 765–768 (cit. on p. 18).

[37] *ParaView — Open-source, multi-platform data analysis and visualization application*. 2025. URL: `https://www.paraview.org/` (cit. on p. 19).

[38] A. Zoia et al. «Overview of the TRIPOLI-4 Monte Carlo code, version 12». In: *EPJ Nuclear Sciences & Technology* 10 (2024), p. 17 (cit. on pp. 19, 21).

[39] B.E. Wells. *TRISO Fuel: Properties and Failure Modes*. Tech. rep. PNNL-31427. U.S. Nuclear Regulatory Commission, 2021 (cit. on p. 20).

[40] A.M. Ougouag and W.K. Terry. *A Preliminary Study of the Effect of Shifts in Packing Fraction on K-Effective in Pebble-Bed Reactors*. Tech. rep. Idaho National Laboratory, 2001 (cit. on p. 20).

[41] D. Mancusi et al. «Overview of TRIPOLI-5, a Monte Carlo code for HPC». In: *EPJ Nuclear Sciences & Technology* 10 (2024), p. 26 (cit. on p. 21).

[42] C. Larmier et al. «Verification Methods for the AGORA Geometry Navigation Engine of the TRIPOLI-5 Monte Carlo Code». In: *Proceedings of M&C 2025*. Denver, USA, 2025 (cit. on p. 21).

[43] W. S. Jodrey and E. M. Tory. «Computer simulation of close random packing of equal spheres». In: *Physical Review A* 32 (1985), pp. 2347–2351 (cit. on pp. 22–25).

[44] P. Romano et al. «OpenMC: A state-of-the-art Monte Carlo code for research and development». In: *Annals of Nuclear Energy* 82 (2015), pp. 90–97 (cit. on pp. 22, 23).

[45] C. Kloss et al. «Models, algorithms and validation for open-source DEM and CFD-DEM». In: *Progress in Computational Fluid Dynamics* 12 (2012), pp. 140–152 (cit. on pp. 22, 31).

[46] X. Yuanhui. «The HTR-10 project and its further development». In: *HTR-2002: Proceedings of the conference on high temperature reactors*. INIS-XA–524. 2002, pp. 8–8 (cit. on pp. 22, 35, 53, 59, 60).

[47] Z. Wu et al. «The design features of the HTR-10». In: *Annals of Nuclear Energy* 218 (2002), pp. 25–32 (cit. on pp. 22, 35, 53, 59, 60).

[48] *OpenMC Python API: pack_spheres Function*. 2025. URL: `https://docs.openmc.org/en/stable/pythonapi/generated/openmc.model.pack_spheres.html` (cit. on p. 24).

[49] *LIGGGHTS® Open Source Discrete Element Method Particle Simulation Code*. 2025. URL: `https://www.cfdem.com/liggghtsr-open-source-discrete-element-method-particle-simulation-code` (cit. on pp. 31, 32).

[50] A. Podlozhnyuk et al. «Model development and verification of LIGGGHTS® open-source DEM software». In: *Journal of Physics: Conference Series* 2048 (2021) (cit. on pp. 31, 32, 82).

[51] V. Rintala et al. «Modeling of realistic pebble bed reactor geometries using the Serpent Monte Carlo code». In: *Annals of Nuclear Energy* 77 (2015), pp. 223–230 (cit. on pp. 31, 32, 82).

[52] M. Rackl and K.J. Hanley. «A methodical calibration procedure for discrete element models». In: *Annals of Nuclear Energy* 307 (2017), pp. 73–83 (cit. on pp. 33, 37).

[53] C. Larmier. «Stochastic particle transport in disordered media: beyond the Boltzmann equation». PhD Thesis. Saclay, France: Université Paris-Saclay, Université Paris-Sud, 2018 (cit. on p. 42).

[54] Y. Xu and K. Zuo. «Overview of the 10 MW high temperature gas cooled reactor—test module project». In: *Annals of Nuclear Energy* 218 (2002), pp. 13–23 (cit. on p. 59).

[55] W.K. Terry et al. «Evaluation of the initial critical configuration of the HTR-10 pebble-bed reactor». In: NEA/NSC/DOC(2006)1 (2006) (cit. on pp. 60–65, 67, 72, 81).

[56] D. A. Brown et al. «ENDF/B-VIII.0: The 8th Major Release of the Nuclear Reaction Data Library with CIELO-project Cross Sections, New Standards and Thermal Scattering Data». In: *Nuclear Data Sheets* 148 (2018), pp. 1–142 (cit. on pp. 72, 86).

[57] A. J. M. Plompen et al. «The Joint Evaluated Fission and Fusion Nuclear Data Library, JEFF-3.3». In: *European Physical Journal A* 56 (2020), p. 181 (cit. on pp. 75, 89).

[58] *SALOME. The open source platform for numerical simulation*. 2025. URL: `https://www.salome-platform.org` (cit. on p. 82).

[59] P.M. Bester et al. «A numerical analysis of the porosity of the HTR-10 packed pebble bed». In: *Annals of Nuclear Energy* 383 (2021), p. 111438 (cit. on p. 82).

[60] *ExaDEM software solution*. 2025. URL: `https://github.com/Collab4exaNBody/exaDEM/` (cit. on p. 98).