

# **Development of a coupling approach for multi-physics analyses of fusion reactors**

Zur Erlangung des akademischen Grades eines  
Doktors der Ingenieurwissenschaften  
(Dr.-Ing.)

bei der Fakultät für Maschinenbau  
des Karlsruher Instituts für Technologie (KIT)

genehmigte  
DISSERTATION

von  
Yuefeng Qiu

Datum der mündlichen Prüfung: 12. 05. 2016  
Referent: Prof. Dr. Stieglitz  
Korreferent: Prof. Dr. Möslang



This document is licensed under the Creative Commons Attribution – Share Alike 3.0 DE License (CC BY-SA 3.0 DE): <http://creativecommons.org/licenses/by-sa/3.0/de/>

# Abstract

Fusion reactors are complex systems which are built of many complex components and sub-systems with irregular geometries. Their design involves many interdependent multi-physics problems which require coupled neutronic, thermal hydraulic (TH) and structural mechanical (SM) analyses. In this work, an integrated system has been developed to achieve coupled multi-physics analyses of complex fusion reactor systems.

An advanced Monte Carlo (MC) modeling approach has been first developed for converting complex models to MC models with hybrid constructive solid and unstructured mesh geometries. A Tessellation-Tetrahedralization approach has been proposed for generating accurate and efficient unstructured meshes for describing MC models. For coupled multi-physics analyses, a high-fidelity coupling approach has been developed for the physical conservative data mapping from MC meshes to TH and SM meshes. Interfaces have been implemented for the MC codes MCNP5/6, TRIPOLI-4 and Geant4, the CFD codes CFX and Fluent, and the FE analysis platform ANSYS Workbench. Furthermore, these approaches have been implemented and integrated into the SALOME simulation platform. Therefore, a coupling system has been developed, which covers the entire analysis cycle of CAD design, neutronic, TH and SM analyses. Results can be displayed in the same underlying CAD geometry. This coupling system and its interfaces have been verified using several test models, including Helium Cooled Pebble Bed Test Blanket Module and an ITER Benchmark model. The system was proven to be reliable, robust and efficient.

In addition, CAD and mesh-based capabilities for the open-source Monte Carlo code Geant4 have been developed, and the integration of Geant4 into the system has been achieved. The system is thus ready to be applied for fusion reactors and also other nuclear systems such as accelerator facilities, utilizing any of the standard analysis codes integrated to the system, or external codes through the provided interfaces.



# Zusammenfassung

Fusionsreaktoren sind komplexe Systeme, die aus vielen Komponenten und Teilsystemen mit unregelmäßigen, komplexen Geometrien aufgebaut sind. Die Auslegung solcher komplexer Systeme beinhaltet viele multiphysikalische Probleme, die gekoppelte neutronenphysikalische, thermohydraulische (TH) und strukturmechanische (SM) Analysen erforderlich machen. In dieser Arbeit wurde ein integriertes System entwickelt, mit dem solche gekoppelten multiphysikalischen Analysen für komplexe Fusionsreaktorsysteme durchgeführt werden können.

Hierzu wurde ein erweiterter Monte-Carlo (MC) Modellierungsansatz entwickelt, der für die Umwandlung von komplexen Geometrien in MC-Modelle mit „hybrider Geometrie“, die sowohl konstruktive Festkörpergeometrie als auch unstrukturierte Maschengitter beinhalten, verwendet werden kann. Weiter wurde ein sogenannter „Tessellation-Tetrahedralization“-Ansatz zur Erzeugung von unstrukturierten Maschengittern entwickelt, wodurch eine genauere und effizientere Erzeugung der Gitter für die Beschreibung von MC-Modellen ermöglicht wird. Für gekoppelte multiphysikalische Analysen wurde ein „High Fidelity“-Kopplungsansatz entwickelt. Die Daten werden dabei physikalisch konservativ vom MC-Maschengitter auf TH- und SMMaschengitter abgebildet. Schnittstellen wurden für die MC-Codes MCNP5/6 und TRIPOLI-4 entwickelt, sowie für die CFD-Codes CFX und Fluent und für die FE-Analyse-Plattform ANSYS Workbench. Des Weiteren wurden diese Ansätze implementiert und in die Simulationsplattform SALOME integriert. Damit wurde ein Kopplungssystem entwickelt, mit dem der komplette Analysezyklus, beginnend mit der CAD-Geometrieerstellung, über die Neutronik, Thermohydraulik und Strukturmechanik, durchgeführt werden kann. Ergebnisse können mit der zugrunde liegenden CAD-Geometrie überlagert werden. Dieses Kopplungssystem und seine Schnittstellen wurden auf Zuverlässigkeit, Robustheit und Effizienz hin überprüft, indem mehrere Testfälle untersucht wurden, einschließlich das Modell eines Helium gekühlten Feststoffblanket-Testmoduls mit Partikelbettschüttung und des ITER Benchmark Modells.

Darüber hinaus wurden erweiterte Funktionen und Schnittstellen für den Open Source Monte-Carlo-Code GEANT4 entwickelt. Damit ist es möglich, CAD- und Maschengitter zur Geometriedarstellung mit GEANT4 zu benutzen. Die hierzu benötigten Schnittstellen wurden in das Kopplungssystem integriert bzw. angebunden. Das System kann damit sowohl für Fusionsreaktoren als auch für andere kerntechnische Systeme, etwa Beschleunigeranlagen, eingesetzt werden. Dabei können Standard-Analyse-Codes verwendet werden, die direkt in das System integriert oder über externe Schnittstellen angebunden sind.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Fusion energy and fusion reactor . . . . .	1
1.2	Multi-physics coupled analyses . . . . .	3
1.2.1	Neutronics . . . . .	3
1.2.2	Thermal hydraulics and structural mechanics . . . . .	4
1.2.3	Multi-physics coupled analyses . . . . .	5
1.3	Objective . . . . .	6
<b>2</b>	<b>Theories and methods</b>	<b>9</b>
2.1	Geometry and data . . . . .	9
2.1.1	Boundary representation geometry . . . . .	9
2.1.2	Constructive solid geometry . . . . .	9
2.1.3	Tessellated geometry . . . . .	11
2.1.4	Mesh geometry . . . . .	11
2.1.5	Discretized data . . . . .	12
2.2	Monte Carlo method . . . . .	14
2.2.1	Monte Carlo method in particle transport . . . . .	14
2.2.2	Geometrical step calculation . . . . .	15
2.2.3	Physical step calculation . . . . .	16
2.2.4	Flux and nuclear heating scoring . . . . .	16
2.2.5	Boundary conditions . . . . .	18
2.3	Basics of multi-physics coupled analyses . . . . .	18
<b>3</b>	<b>Investigations on codes, platforms and coupling approaches</b>	<b>23</b>
3.1	Investigations on codes and platforms . . . . .	23
3.1.1	CAD, mesh and data format . . . . .	24
3.1.2	CAD modeling tools . . . . .	26
3.1.3	Mesh generation tools . . . . .	26
3.1.4	Monte Carlo transport codes . . . . .	28
3.1.5	CFD codes . . . . .	29
3.1.6	FEM structural mechanics software . . . . .	29
3.1.7	Visualization software . . . . .	31
3.1.8	Simulation platforms . . . . .	31
3.2	Survey of coupling approaches . . . . .	32
3.2.1	MC geometry conversion approaches . . . . .	32
3.2.2	Coupling interface between MC codes and CFD codes . . . . .	34
3.3	Decisions on candidate selections . . . . .	35

<b>4</b>	<b>A multi-physics coupling system</b>	<b>37</b>
4.1	Design of the coupling system . . . . .	37
4.2	McCad integration into the SALOME platform . . . . .	40
4.2.1	Introduction on the SALOME module integration . . . . .	40
4.2.2	The McCad module . . . . .	41
4.2.3	The conversion process . . . . .	44
4.3	A novel meshing approach for MC geometry modeling . . . . .	46
4.3.1	MC geometry modeling and results scoring . . . . .	46
4.3.2	Quality of MC scoring mesh . . . . .	47
4.3.3	Quality of MC geometry mesh . . . . .	49
4.3.4	A novel mesh generation approach for MC geometries . . . . .	49
4.3.5	Hybrid CAD and mesh conversion for MCNP6 . . . . .	52
4.4	Verifications of MC geometry conversions . . . . .	55
4.4.1	Verification of the TBM test case model . . . . .	55
4.4.2	Verification of the ITER Benchmark model . . . . .	61
4.5	McMeshTran development for multi-physics coupling . . . . .	66
4.5.1	The McMeshTran module . . . . .	66
4.5.2	The coupling process . . . . .	67
4.6	McMeshTran coupling interfaces . . . . .	70
4.6.1	Interface for the MCNP mesh tally . . . . .	70
4.6.2	Interface for MCNP6 results on unstructured mesh . . . . .	72
4.6.3	Interface for the TRIPOLI-4 mesh tally . . . . .	73
4.6.4	Interface for Fluent and CFX . . . . .	74
4.6.5	Interface for ANSYS Workbench . . . . .	77
4.7	Verifications of coupling interfaces . . . . .	78
4.7.1	Verification of the data translation . . . . .	78
4.7.2	Verification of the hybrid geometry . . . . .	82
4.7.3	Verification of the TRIPOLI interface . . . . .	83
4.7.4	Verification of the CFD interfaces . . . . .	85
<b>5</b>	<b>General CAD- and mesh-based MC simulation</b>	<b>91</b>
5.1	Half-space solid . . . . .	91
5.1.1	Definition of the half-space solid . . . . .	91
5.1.2	Implementation of the half-space solid . . . . .	94
5.1.3	CAD conversion for Geant4 . . . . .	97
5.1.4	Test validations . . . . .	101
5.2	Unstructured mesh geometry and scoring . . . . .	105
5.2.1	New Geant4 unstructured mesh elements . . . . .	105
5.2.2	Unstructured mesh scoring . . . . .	107
5.2.3	Coupling interfaces . . . . .	108
5.2.4	Verification of the unstructured mesh scoring . . . . .	109
<b>6</b>	<b>Summary and conclusion</b>	<b>111</b>
6.1	Summaries . . . . .	111
6.1.1	Advanced hybrid MC geometry conversion . . . . .	111
6.1.2	Multi-physics coupling . . . . .	112

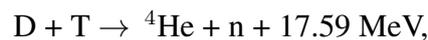
6.1.3	Integrated coupling system . . . . .	112
6.1.4	General CAD and mesh-based MC simulation . . . . .	112
6.2	Conclusions and outlooks . . . . .	113
	<b>Bibliography</b>	<b>117</b>
	<b>Acronyms</b>	<b>125</b>



# 1 Introduction

## 1.1 Fusion energy and fusion reactor

Fusion energy is considered as an ultimate energy source for human-being[Sta10], since the resources for fusion energy is sustainable for millions of years . The fusion energy is released from nuclear fusion reaction, which happens when two light nuclei collide with enough speed. Certain amount of mass is lost after the reaction. According to the equation  $E = mc^2$ , the lost mass is converted into energy, that is the fusion energy. There are three types of realistic feasible fusion reactions —D-T, D-D and D-<sup>3</sup>He reactions, where *Deuterium* (D) and *Tritium* (T) are two isotopes of hydrogen, and <sup>3</sup>He is an isotope of helium. Take the D-T reaction for example:



it produces a <sup>4</sup>He (alpha particle), a neutron and release 17.59 MeV energy. The energy exist in the form of kinetic energy <sup>4</sup>He (3.52 MeV) and neutron(14.07 MeV). Since <sup>4</sup>He is a heavy charged particle remained being confined in the magnetic field, neutron is the main energy carrier that carrying  $\sim 80\%$  of the fusion energy.

In fusion reaction, two nuclei should collide with enough kinetic energy to overcome the repulsive Coulomb force. Therefore, the D-T reaction takes place only in condition when temperature is above  $5 \times 10^7$  K [Sta10]. In such a high temperature, matter exists in the plasma state, which is a neutral collection of ions and unbound electrons. In order to control fusion reactions, the plasma should be confined. Magnetic and inertial confinements are two major technologies. A tokamak is a facility that uses a closed helical shape of magnetic field lines to confine the plasma. It was first built in 1968, are extensively investigated worldwide and become probably the most advanced. The International Thermonuclear Experimental Reactor (ITER), which is under construction and will be the largest experimental tokamak in the world, is aiming at demonstrating the feasibility of thermal fusion power. The next step beyond ITER is the DEMO (DEMONstration Power Plant), which will demonstrate the generation of electric power from fusion energy.

The ITER tokamak machine is shown in Figure 1.1. It is a huge and complex machine with 28.5 m of diameter, 29 m of height and  $2.3 \times 10^4$  tons of weight [ITE14]. This huge machine consists of components blankets, divertors, vacuum vessel, toroidal field coils and poloidal field coils, central solenoid, cryostat and heating systems. These components have complex and irregular geometries, which are great challenges for design

## 1 Introduction

and analyses. The computer-aided design (CAD) technology is strongly relied in the whole process.

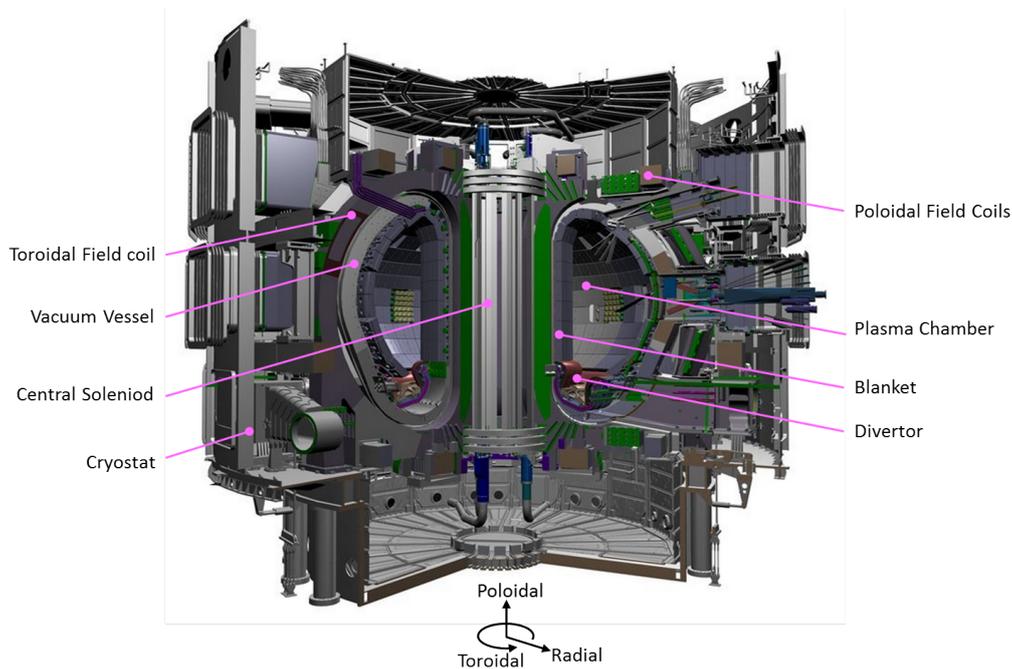
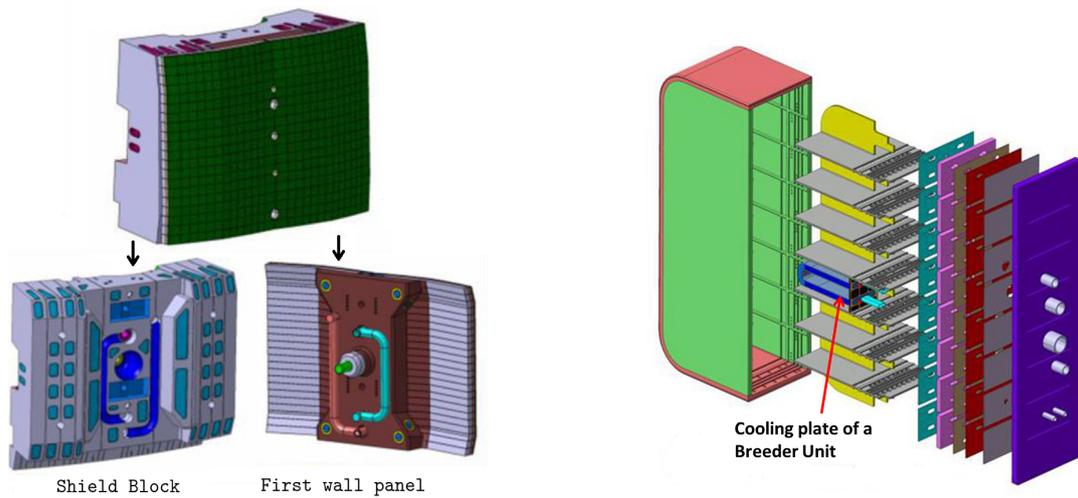


Figure 1.1: The ITER tokamak machine

Another challenge is that, the fusion reactor is a large-scale scientific project covering many disciplines such as plasma physics, neutronics, thermal hydraulics (TH), structural mechanics (SM), cryogenics, superconducting technologies, etc. A typical case is the design of fusion blanket modules.

Blankets surrounding the reaction chamber provide shielding of heat flux and neutrons from the vacuum vessel and superconducting magnets field coils. Since they face directly the high temperature plasma, and high energy neutrons are mostly deposit in the blankets, it has to remove large amount of heat. For ITER, there are totally 440 blanket modules (Figure 1.2(a)) designed to remove up to 736 MW of thermal power by pressurized cooling water [RCC<sup>+</sup>13], and the first wall (FW) facing to the plasma are imposed on up to 4.7 MW/m<sup>2</sup> surface heat flux and 20 MW/m<sup>3</sup> volumetric nuclear heating [TGS<sup>+</sup>91].

In addition, blankets are also responsible for radiation shielding, tritium breeding, etc. ITER blankets do not breed tritium, but ITER provides opportunities to insert Test Blanket Module (TBM), e.g. the Helium Cooled Pebble Bed (HCPB) TBM [CKI<sup>+</sup>09] as shown in Figure 1.2(b), to ensure the self-sufficient of fusion fuel for the future DEMO reactor. To fulfill these goals, intensity coupled neutronic, thermal hydraulic and structural mechanical analyses should be performed for the design of the fusion reactor blanket module.



(a) ITER blanket module, consisting of a first wall panel and shield block with complex flow channels.

(b) HCPB TBM module. It has breeding units for test the sustainability of tritium production for DEMO reactor.

Figure 1.2: ITER blanket (left) and TBM module (right) [CKI<sup>+</sup>09]

## 1.2 Multi-physics coupled analyses

### 1.2.1 Neutronics

The study of neutron behaviors is of significance for the ITER design. Since neutron carries 80% of fusion energy in the D-T reaction, the deposited energy of neutrons and gamma photons causes heat removal issues in in-vessel components and superconducting coils. The radiation damages will affect properties of materials, hence influence the thermal, mechanical, electrical performances of components. Besides that, the radiation hazard on humans is also an important issue.

To simulate the neutron transport, the stochastic method, or in another word Monte Carlo (MC) method, is widely adopted and trusted. A neutron is simulated from its birth in a source to its death in absorption, escape, etc., and probabilities are randomly sampled to determine if reactions are taking place. The simulation is very close to reality, so that it is often called numerical experiment [B<sup>+</sup>03]. MC method is a general method suitable also for other neutral or charged particles, e.g. gamma, proton, etc.

A important characteristic of MC method is that it does not require spatial discretization for describing the problem geometry. In MC simulation, a problem geometry (often called solid, body, or cell) is described by defining its spatial boundaries. The space inside the solid is filled with homogeneous material, and does not need to be subdivided for MC simulation. A Common-used type of geometry is called Constructive Solid Geometry (CSG), which is constructed by Boolean operations of primitives or sub-Euclidean spaces (half-spaces). Unfortunately, in Computer-Aided Design (CAD) another type of geometry called Boundary Representation (BRep) geometry is mostly used, but it is not efficient

## 1 Introduction

for MC simulation in particle tracking computation. Since CAD is very important for fusion reactor design, conversion from BRep to CSG format is essential for modeling complex CAD geometry in MC simulation. Automatic programs, e.g. McCad [LFP14] and MCAM [Wu09], were developed for assisting the conversion process. Manual model simplifications are still necessary for decreasing the model complexity and simplifying extreme complex surfaces, in order to be converted a BRep model into a CSG model.

Another approach for modeling CAD geometry in MC simulation is converting BRep solids into surface meshes. A surface mesh has close boundaries consisted of planar triangle or quadrangle facets for defining a solid volume. It is generated from BRep solid with a approximation process called *faceting*, or *tessellation*. These surface meshes have the advantages of fast particle tracking computation on planar facets and reducing manual effort for some simplification, on the other hand increase computation time due to large amount of facets are produced for non-planar surface.

Although the MC geometry does not require spatial discretization for describing the problem geometry, mesh geometry is often used for obtaining spatial distribution of physical field. This tally technique performs in a way that the mesh only records but not influences the particle behaviors. This technique is called superimposed mesh tally, which is important for computing spatial nuclear heating distribution for multi-physics coupling analysis. Orthogonal mesh is implemented in many MC codes, while it has a nature disadvantage in adapting complex geometry. Recently, a new feature was implemented in MCNP code version 6 [Mar12] that using geometry-adaptive unstructured mesh for describing geometry and obtaining spatial tally, and then a superimposed unstructured mesh tally function was implemented in Serpent-2 [Lep12]. The unstructured mesh based MC simulation is recently being widely and intensively concerned in neutronics analysis, especially for multi-physics coupling analysis.

### 1.2.2 Thermal hydraulics and structural mechanics

Thermal hydraulic analyses, consisting of thermal and fluid dynamic analyses, are greatly concerned in designing of heat intensive fusion device components. Under strong heat flux and volumetric heat load, it is important to calculate the temperature, velocity, and hot-spots, and provides important information for optimizing the design. Structural mechanical analyses provide evaluations on the mechanical performances of those components under strong heat load and high fluid pressure. Thermomechanics analysis (TMA) is one of the important analyses concerning the influence of heat on mechanical properties.

Computational Fluid Dynamics (CFD) analyses are often carried out for three-dimensional (3D) thermal hydraulics studies of fusion blanket components. The numerical method used in CFD codes, e.g. Finite Volume Method (FVM) and Finite Element (FE) Method (FEM), require problem geometries being discretized, and the meshes are greatly concerned in order to get physically correct answers. For structural mechanic analyses, FEM is a mature method suitable for solving general linear and

nonlinear problems. There are many CFD and FE structural analysis codes and software being developed, both commercial and open-source codes.

### 1.2.3 Multi-physics coupled analyses

The coupled neutronics, thermal hydraulics and structural mechanics analyses as shown in Figure 1.3 are based on three major coupling terms — the problem geometry, the nuclear heating data and the temperature distribution.

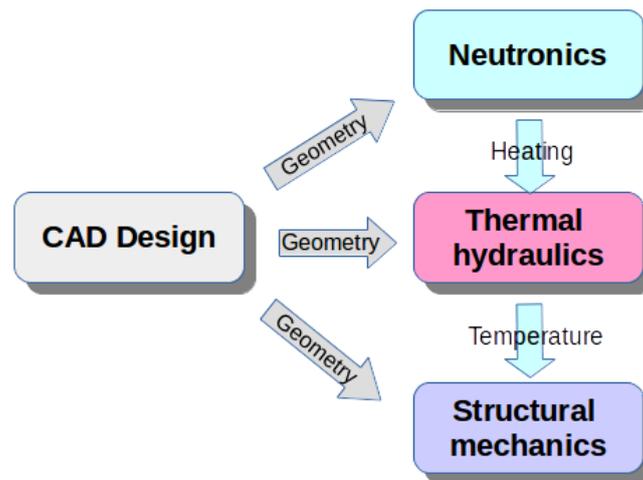


Figure 1.3: Multi-physics coupling process. The major coupling terms are consistent geometries, nuclear heating and temperature field.

This coupled analysis are required to be carried out on consistent geometries. However, they are often inconsistent in between due to the different natures and modeling approaches of MC, CFD and FE simulation. For example, neutronics analysis often performed on the whole fusion reactor model with necessary simplifications, while CFD and FE analyses often focus on particular components with detail fillets and chamfers. Since meshes used in CFD and FE analyses can be highly consistent with CAD geometries, the major issue is the high-fidelity modeling of MC geometry. Currently the MC modeling is still most performed by BRep-to-CSG conversion with heavy manual simplifications on the CAD model. A solution has to be work out to improve the modeling approach, possibly via combining traditional BRep-to-CSG conversion with high-fidelity geometry-adaptive faceting conversion and unstructured mesh model generation as well.

Nuclear heating is probably the most important coupling term between neutronics and thermal hydraulics analyses. A high-fidelity coupled analysis is achieved through data source processing, data transfer and interfacing. Firstly, the data source has to be a large extend overlap with the problem geometry. Because the orthogonal mesh tally has its nature problem in geometry adaptivity, the use of a unstructured mesh tally for obtaining

## 1 Introduction

high-fidelity heating results provides a ideal solution. Secondly, the meshes elaborated for CFD and FE analysis usually differ from that used for tallying nuclear heating in MC simulation, hence high-fidelity and physical conservative data mapping from MC meshes to CFD and FE meshes is required. Last but not least, the interfaces have to be developed not only for post-processing MC results, but also for providing heat source data for CFD and FE simulation. Until this PhD work there are no tools available for helping this 3D mesh-based coupling process.

The calculation of temperature distribution as coupling term is already available with two approaches. The first approach is obtaining temperature distributions on both solid and fluid domains by carried out CFD simulation. Conjugate heat transfer [ANSa] is a technique used for fluid-solid-coupled heat transfer calculations. Another approach is first conducting CFD simulations and obtaining an approximate heat transfer coefficient and fluid temperature on the fluid domain, then using these result for calculating temperature[CKI<sup>+</sup>09] in FE thermal analyses. In both cases the nuclear heating will be taken into account as external heat source. The temperature distribution on the solid domain is provided for thermo-mechanical analysis (TMA). These two coupling approaches are already well-developed in some integrated simulation platforms, e.g. ANSYS Workbench [ANSa].

This multi-physics analysis is an iterative and general process. The CAD design as the basic of the whole process has to be change due to various reasons. For example, an important goal in neutronic analysis is not achieved, or large structure displacement is discovered after TMA. Furthermore, this multi-physics coupling issue exists in many nuclear systems, e.g. in accelerator system. Conducting this large coupling process manually is certainly time consuming and error-prone. An ideal solution is developing a integrated coupling system to enhance the efficiency of the whole coupling process, and apply for multiple nuclear system. Such a huge system will cover many simulation codes and programs, and is very challenging in implementation.

### 1.3 Objective

To perform multi-physics coupling simulations on complex fusion reactors, a suitable system is to be developed within the frame of this PhD work. This work can be broken down into several parts:

1. To develop an advanced MC geometry conversion approach for complex CAD geometries. This approach has to be based on the traditional BRep-to-CSG conversion, and introduce new approach, e.g. geometry-adaptive faceted solid or unstructured mesh, for high-fidelity MC modeling. This is very important for avoiding oversimplification, and keeping the geometry consistence in MC and CFD/FE analyses.
2. To develop a high-fidelity, flexible and generic multi-physics coupling approach. It should observe physical conservation in mapping nuclear heating data, be geometry- and problem-independent in applications, and be ready extensible for

more simulation codes. More importantly, interfaces should be provided for common-used MC codes, CFD and FE structural mechanical analysis software.

3. These approaches are to be implemented in a suitable integrated simulation platform, so that the entire large analysis cycle can be fulfilled. This system has to provide internally links for integrated simulation codes, and interface files for external codes which cannot be integrated, so that the coupling process can be to a large extent simplified and accelerated. Additionally, useful user interaction and data analysis approaches should also be provided.
4. All developed approaches and implementations should be verified with suitable test models, and the results should be evaluated by suitable means.
5. In addition, this coupling approach should be as general as possible, so that it can be not only applied for fusion reactor systems, but also for other nuclear systems, e.g. accelerator systems, in dealing with similar multi-physics problems.

However, the PhD work is not going to solve all problems in the coupled analysis. It is aiming at provide useful tool for dealing with the steady state, linear, thermal problem occurring in most of the coupled neutronics, CFD and FE structural analyses. Time-dependent transient physics, nonlinear thermal feedbacks on neutron cross-sections and structural displacement which need to be considered in particular cases are not studied at this moment, but will be good topics for further developing the coupling system.



## 2 Theories and methods

### 2.1 Geometry and data

Geometry is concern in the coupled multi-physics analysis as crucial input data. It has many forms of representations in different stages of the coupling process. A short introduction is given for better clarification.

#### 2.1.1 Boundary representation geometry

The Boundary Representation (BRep) is a geometry representation technique for CAD system. A solid is represented in items of the “skin” surrounding it [Str06]. Take Figure 2.1 as an example, the solid is formed by several faces. A face is formed by a surface bound by set of edges lying on it, and an edge is a portion of curves limited by vertex (or vertices) on the curve. In CAD system, a wire is defined as a closed set of edges with direction orientations, and a shell is defined as a closed set of faces with forward or reversed orientations.

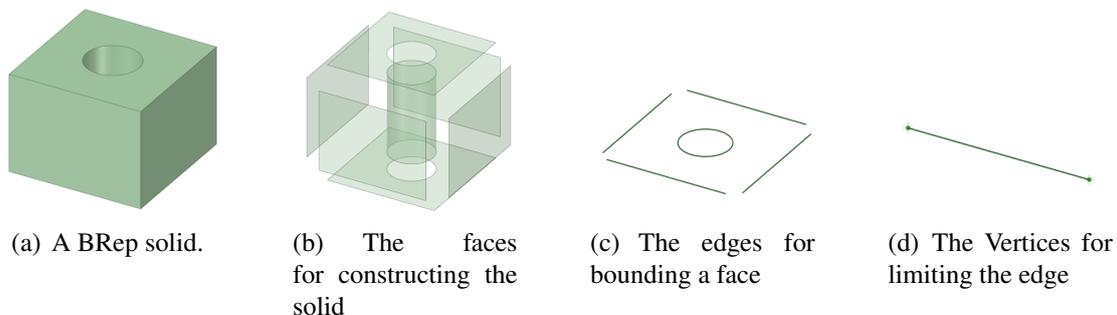


Figure 2.1: The geometry representation of a BRep solid.

#### 2.1.2 Constructive solid geometry

Constructive solid geometry (CSG) is a representation approach commonly used for MC codes. It is constructed by Boolean operations *Intersection*, *Union* and *Subtraction* on basic geometry objects. Based on different basic geometry objects, there are two CSG types — primitive CSG and half-space CSG.

## 2 Theories and methods

The primitive CSG is constructed by Boolean operations on a set of primitives, e.g. Box, cylinder, sphere, cone, and torus. Primitives with analytic surfaces are preferred in MC model, but those with freeform surfaces are still adopted (e.g. in [Col12]). Using this geometry representation, the same geometry created in Figure 2.1 can be created using primitive CSG as Figure 2.2, by subtracting the cylinder from the box.

A half-space CSG is created by Boolean operations of half-spaces defined by algebraic surfaces. For example, a infinite planar surface cuts the 3D Euclidean space into two infinite subspaces, and a closed sphere separates it into a finite and an infinite subspaces. We used surface normal to distinguish two half-spaces create by a surface, and the half-space on the side which the surface normal points to is the *positive* half-space, otherwise is *negative* half-space. For example, the positive half-space of the plane  $z = 0$  is on the positive side of Z-axis, the positive half-space of a sphere is the space outside the sphere. By combining a set of half-spaces we can create a closed solid. For example in Figure 2.3, the solid is built by intersection of positive half-spaces of plane 1, 3, 5, 7, negative half-spaces of plane 2, 4, 6. Analytical surfaces are also preferred for creating half-spaces, such as plane, cylinder, sphere, cone, general quadratic surface, and also torus surface.

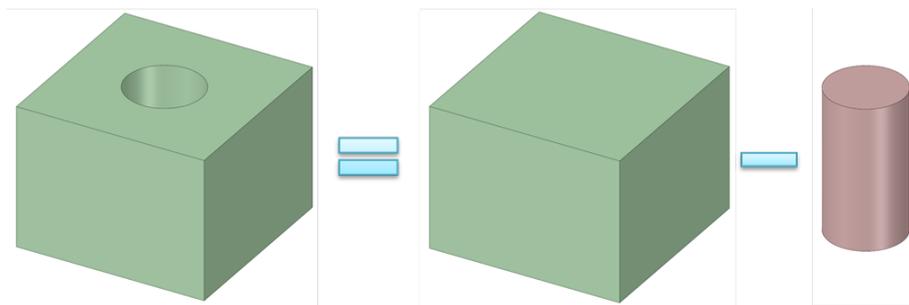


Figure 2.2: The primitive CSG representation. The solid is created by subtracting a cylinder from a box.

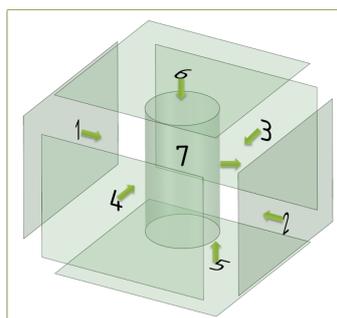


Figure 2.3: Half-space CSG presentation. The solid is constructed by intersections of positive half-space of plane 1, 3, 5, 7 and negative half-space of 2, 4, 6.

### 2.1.3 Tessellated geometry

A tessellated solid is a solid bounded with planar facets, e.g. triangle or quadrangle facets. Curved faces of a CAD solid are approximate with a number of these facets, and the facet size and shape have to be change to adapt to the surface curvature. A large amount of facets will be introduced for a curved, depending on the required precision. Take the tessellated solid in Figure 2.4 for example, the cylinder face is approximated by long and thin triangle facets.

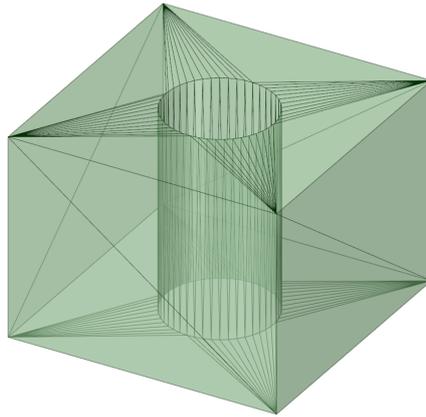


Figure 2.4: Tessellated presentation of a solid. The face of the solid is approximated by triangle facets.

This solid type is intensively studied in computer graphics, and is used for visualization of CAD solid. Since the tessellated solid can precisely represent any CAD solid and can be easily and fast generated by CAD software, it is attractive also for MC transport simulations, and has been utilized in MC codes such as Geant4 [Col12] and DAGMC [TWK<sup>+</sup>09]. However, it is time-consuming to conduct MC simulation using this solid type, when the CAD solids have many curved faces, and hence an acceleration algorithm is needed for speed up the simulation.

### 2.1.4 Mesh geometry

The use of mesh based geometries for discretizing problem domains has a long history in CFD and FE simulations, it have also been adopted for MC simulation since last decade. For adapting the more and more complex geometry, meshes were developed from orthogonal mesh, to body-fit structured mesh, and to unstructured mesh.

In an orthogonal mesh, grid lines are perpendicularly intersected [Wikb]. This mesh type can be built in different type of coordinates, e.g. Cartesian, cylindrical and spherical coordinates. It has nature drawbacks in geometry adaptivity. For example in Figure 2.5(a), the mesh is difficult in representing the hole with a cylinder surface unless using very small intervals. Using rectilinear meshes with non-uniform intervals can model better the this hole by increase resolution in the middle, but the effect is still limited.

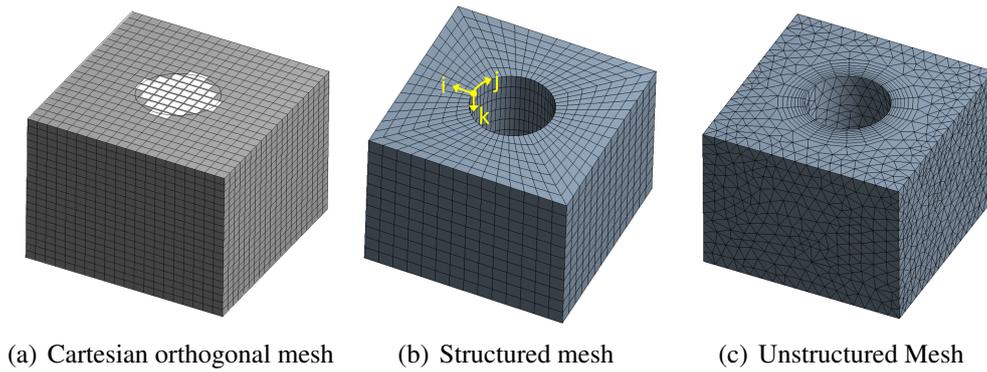


Figure 2.5: Different type of mesh

Structured mesh is not defined on specific coordinates system. Instead, it is on a curvilinear coordinates system which coordinates are adapted to the geometry [TSW10]. In this coordinates system, a grid is considered logical orthogonal, and the nodes can be similarly indexed as an orthogonal mesh. For example, the structured mesh in Figure 2.5(b) can be index with  $i, j, k$ . The structured mesh can be well adapted to the geometry, while it is difficult to map the coordinates to more and more complex geometry. New approaches such as Block Structured Mesh generation is proposed in meshing tools such as ANSYS-ICEM [ANSb], and used often in CFD simulation.

Unstructured meshes uses a different data structured in constructing meshes. There are no curvilinear coordinates and no local  $i, j, k$  indexes. Instead, the nodes are stored in a one-dimensional (1D) array, and the mesh *elements* (the basic partitions of a mesh, also called *cells* in FVM) are defined with a shape type and a list of corresponding nodes constituting it. Theoretically any polyhedra can be used in unstructured meshes; in practice we are often using tetrahedron, pyramid, pentahedron (wedge) and hexahedron as shown in Figure 2.6, in both first- and second order. The first-order element has two nodes in each edge, and the cell faces are linear. The second-order element has an extra node at the mid-point of each edge, so the face can be quadratic surface. Higher order elements exist, but are not commonly used. Unstructured mesh can use several element types for one problem geometry, and it offer high geometry adaptivity. Figure 2.5(c) is a example of unstructured mesh.

### 2.1.5 Discretized data

3D discretized data are often associated with unstructured mesh in several means, as shown in Figure 2.7. The data associated for cell are called *cell data*, and volume averaged data such as mass density, power density are often cell data. Cell data is usually associated with indexes of cells. Another equivalent way is assigning cell data for cell centroid, which is the geometric center of the cell. Data associated with mesh node are called *node data*. Both cell data and node data are kept together with the mesh geometry. *Point data* are data associated with a point defined by values in each coordinates. Both cell data

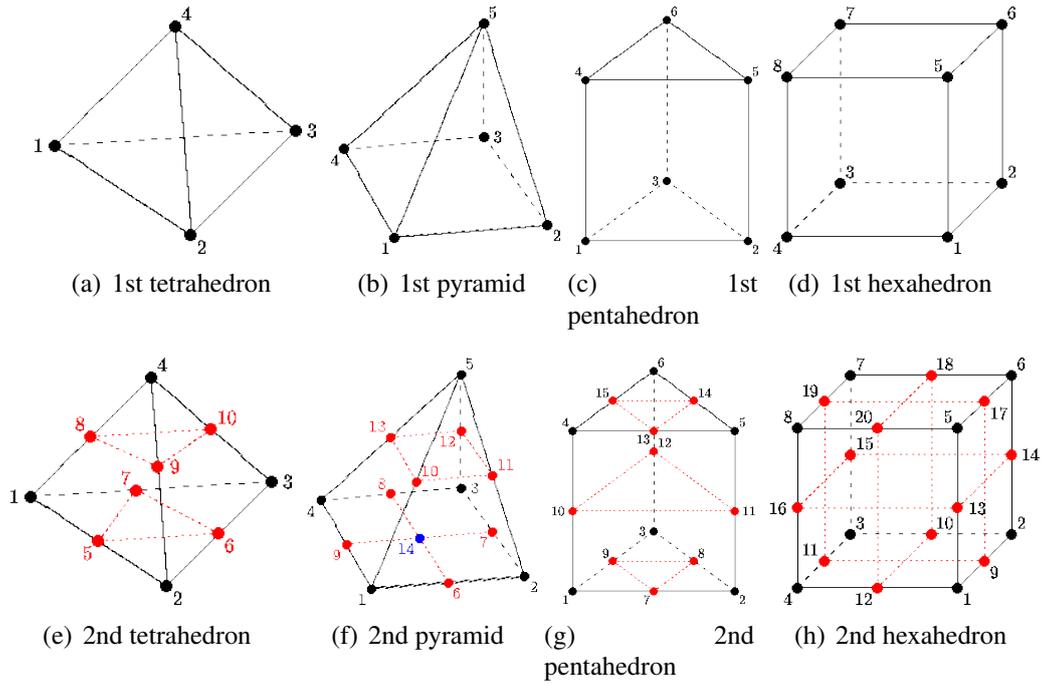


Figure 2.6: First-order (1st) and second order (2nd) element type to represent complex geometries [PBC<sup>+</sup>00].

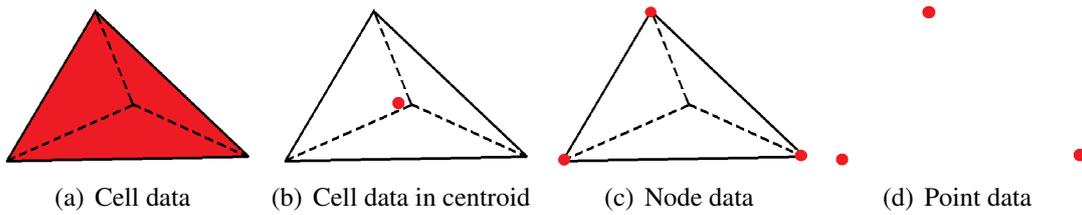


Figure 2.7: Location of discretized data

and node data can be degenerated into point data, without keeping the mesh geometry information.

## 2.2 Monte Carlo method

### 2.2.1 Monte Carlo method in particle transport

The name of Monte Carlo (MC) method is coming from the famous casino Monte Carlo located in Monaco. It is a stochastic method widely used in particle simulations. It simulates the particle from birth in a source or reaction spot, to death in absorption, leakage, or termination. Although it is often said that MC method solves the integral transport equation, it does not need to write the transport equation when using the MC method to solve a problem [B<sup>+</sup>03]. When enough amount of particles histories are simulated, the averaged behavior of particles gives solutions.

For simulating a particle using MC method, there are three important computations — *geometrical step* calculation, *physical step* calculation and *particle scoring*. The geometrical step is the distance from the current location of the particle to the geometry boundary which the particle would cross. The physical step is the distance from the particle position to where the next reaction would occur. Then the physical step is compared with geometry step, and the smaller value will be chosen as the *real step* to decide if the particle is leaving or having a reaction. The real step will be recorded, if it is inside the scoring region.

Figure 2.8 illustrates of a typical neutron history. The MC simulation consists of the following steps [B<sup>+</sup>03]:

1. When the neutron enters the boundary of a material, a geometry step is calculated from the entry point to a possible escape point  $a$ , and a physical step is calculated to position  $1$ , where a neutron scattering reaction might occurred. The physical step is chosen, since it is smaller than the geometry step, and a photon is generated from the reaction and temporary stacked. The neutron continues moving, with an energy and direction randomly sampled from the probability density function of the scattering distribution;
2. A new geometry step is calculated from  $1$  to  $b$ , and a (n, 2n) reaction is randomly sampled at position  $2$ . The physical step is again adopted, and the neutron is terminated in a (n, 2n) reaction and two neutrons are born. One neutron goes on, and another neutron is stacked;
3. A new geometry step is calculated from  $2$  to  $c$ , but a neutron capture happens at position  $3$ , so this neutron is terminated at position  $3$ .
4. The stacked neutron is retrieved. The geometry step to exit the material at position  $d$  is smaller than the physical step to position  $4$ , thus the neutron leaks out of the material.
5. And the remained photon is captured at position  $5$  instead of leaking out at position  $e$ .

## 2.2.2 Geometrical step calculation

Calculation of the geometry step is to find the distance to the nearest intersection boundary. In MC method, the 3D Euclidean space is unambiguously defined as a problem domain and infinite outer space. The problem domain is defined as a set of spatial regions called *solids* or *cells*. They are closed regions with non-zero finite volume, and are filled with homogeneous material. There are many type of solids used in MC simulation, for example, primitive or half-space CSG, tessellated (faceted) solid and unstructured mesh.

For neutral particles, e.g. neutrons and photons, the travel path are straight between two collisions, particle-geometry intersection are similar as intersection of optical ray with geometry object, which is intensively investigated in computer graphic science. For charged particles moving in a magnetic or electric field, the curve paths are usually approximated with small line segments, and in each segment they are treated similarly as ray path [Col12]. Because the geometry step calculation needs to be repeated countless times, the computation efficiency is greatly concerned, which strongly depends on the way of solid description.

Using half-space CSG solid for illustration, a ray with a base point of  $(x_0, y_0, z_0)$  and a normalized direction vector of  $(x_d, y_d, z_d)$

$$(x, y, z) = (x_0 + x_d t, y_0 + y_d t, z_0 + z_d t), \quad \text{where } t > 0. \quad (2.1)$$

intersect with a boundary surface  $f(x, y, z) = 0$  of a half-space solid. In order to find the intersect points, the following equation needs to be solved:

$$f(x_0 + x_d t, y_0 + y_d t, z_0 + z_d t) = 0. \quad (2.2)$$

Depending on the surface type, the equation 2.2 can be a linear, a quadratic, or a quartic equation, and the solutions of these type of equations can be found in [Gla89] and [Arv91]. Since the direction vector of the ray is normalized, the solutions of  $t$  are actually the distance from the base to the intersect points. The negative answer should be given up as it is behind the ray. The intersect point can be solved by substituting  $t$  to Equation 2.1.

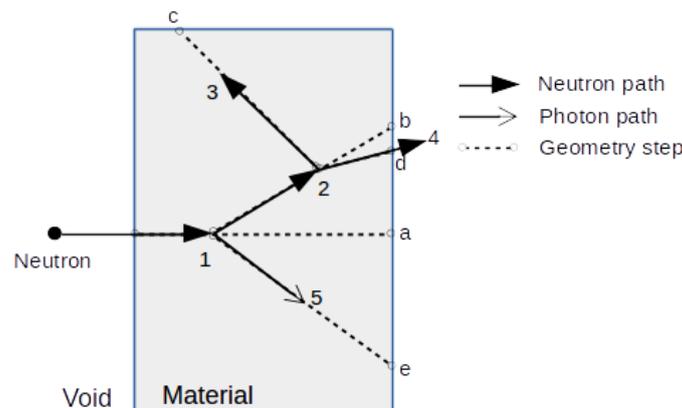


Figure 2.8: A typical neutron history in the Monte Carlo simulation

### 2.2.3 Physical step calculation

Assuming the solid has a homogenized material composition, the physical step of a particle is calculated by random sampling the first collision probability density function, which can be written as the following equation [B<sup>+</sup>03]:

$$p(l) = e^{-\Sigma_t l} \Sigma_t, \quad (2.3)$$

where  $\Sigma_t$  is the macroscopic total cross-section of a nuclei sampling from the material composition, and  $l$  is the travel distance from the starting point. According to the definition of probability density function, this reads to

$$\int_0^{\infty} p(l) dl = \int_0^{\infty} e^{-\Sigma_t l} \Sigma_t dl = 1. \quad (2.4)$$

For sampling a physical step  $L$ , a random number  $\xi$  with ( $0 \leq \xi < 1$ ) is sampled, let

$$\xi = \int_0^L e^{-\Sigma_t l} \Sigma_t dl = 1 - e^{-\Sigma_t L}, \quad (2.5)$$

therefore,  $L$  can be solved by

$$L = -\frac{1}{\Sigma_t} \ln(1 - \xi). \quad (2.6)$$

$1 - \xi$  can be replaced by  $\xi$  because they have an identical distribution, thus the physical step  $L$  is calculated:

$$L = -\frac{1}{\Sigma_t} \ln(\xi). \quad (2.7)$$

### 2.2.4 Flux and nuclear heating scoring

Considering that the MC method “solves” the integral transport equation, the average particle flux  $\bar{\phi}_V$  in a solid with volume  $V$  can be written as [B<sup>+</sup>03]:

$$\bar{\phi}_V = \frac{1}{V} \int dE \int dt \int dV \int d\Omega \nu n(\vec{r}, \hat{\Omega}, E, t), \quad (2.8)$$

where  $n(\vec{r}, \hat{\Omega}, E, t)$  is particle density at position  $\vec{r} = (r_x, r_y, r_z)$  with direction  $\hat{\Omega} = (\Omega_x, \Omega_y, \Omega_z)$  and energy  $E$  and time  $t$ , and  $\nu$  denotes the scalar velocity. Defining  $s$  as the track length variable along the particle track, ( $ds = \nu dt$ ) modifies Equation 2.8 to

$$\bar{\phi}_V = \frac{1}{V} \int dE \int dV \int d\Omega \int ds n(\vec{r}, \hat{\Omega}, E, t). \quad (2.9)$$

The quantity  $\int dE \int dV \int d\Omega \int ds n(\vec{r}, \hat{\Omega}, E, t)$  can be thought of summing up the length of all particle tracks within the solid, thus the average flux  $\bar{\phi}_V$  can be estimated by dividing it by the volume  $V$ . Considering the  $i^{th}$  particle history creates  $m_i$  tracks in inside the volume, and the  $j^{th}$  particle track has particle weight  $w_j$  and track length  $L_j$ , after  $N$

particle histories the normalized volume-average flux  $\bar{\phi}_V$  in the solid can be calculated as:

$$\bar{\phi}_V = \frac{1}{V} \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^{m_i} w_j L_j. \quad (2.10)$$

The nuclear heating tally in a solid volume can be modified from the track length flux tally. The total heating in a solid volume is calculated by

$$\begin{aligned} H &= \frac{1}{m} \int dE \int dt \int dV \int d\Omega \Sigma_t(E) H(E) \nu n(\vec{r}, \hat{\Omega}, E, t), \\ &= \frac{1}{m} \int dE \int dV \int d\Omega \int ds \Sigma_t(E) H(E) n(\vec{r}, \hat{\Omega}, E, t), \end{aligned} \quad (2.11)$$

where  $m$  is the mass, and  $\Sigma_t(E)$  and  $H(E)$  are the total cross section and the heating number per collision of a nuclei random sampled from the material composition. Therefore, the total heating per unit mass after  $N$  particle histories can also be estimated by means of the track length estimator as:

$$\dot{q}_i = \frac{1}{m} \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^{m_i} w_j L_j \Sigma_t(E_j) H(E_j), \quad (2.12)$$

where  $E_j$  is energy of the track  $j$ .

Besides the problem geometry, the track length estimator can also be applied for the geometry representation, which is independent of the problem geometry (so-called *superimposed geometry*). Meshes as superimposed geometry are preferred, they allow to provide 3D spatial distributions with adaptive resolutions.

The results of Monte Carlo simulations are estimated by mean values, thus it is important to know the precision of the answers. Denoting  $x_i$  as a scored quantity (e.g.  $\bar{\phi}_i$  and  $\dot{q}_i$ ) in the  $i_{th}$  particle history, the true mean value is estimated by the sample mean  $\bar{x}$  with the Strong Law of Large Number [B<sup>+</sup>03], where

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i. \quad (2.13)$$

The sample standard deviation of the scores can be estimated by  $S$  in Monte Carlo simulation, given by

$$S^2 = \frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N - 1} \approx \bar{x^2} - \bar{x}^2, \quad (2.14)$$

where  $\bar{x^2}$  is given by:

$$\bar{x^2} = \frac{1}{N} \sum_{i=1}^N x_i^2 \quad (2.15)$$

Since the score result  $x_i$  varies tremendously in each particle history, the standard deviation of the sample means is used for evaluating the accuracy of the sample mean  $\bar{x}$  instead of the standard deviation of the scores, given by

$$S_{\bar{x}} = S/\sqrt{N} \quad (2.16)$$

In the MCNP code, the relative standard error is used which is defined as

$$R \equiv S_{\bar{x}}/\bar{x} \quad (2.17)$$

The relative error is straightforward for representing the precision of the scored result, because it represents the statistical precision as a percentage with respect to the estimated mean. An empirical guideline for interpreting the value  $R$  is that,  $R < 10\%$  represent general reliabilities, and  $R < 5\%$  shows good reliabilities of the results[B<sup>+</sup>03].

## 2.2.5 Boundary conditions

In a MC simulation, the whole universe has to be unambiguously defined. When the particle escapes from the problem geometry and leaves to the outer-space, it provides no further contribution to the simulation. Therefore, the MC model is usually enclosed by an envelope, and a particle traveling through its boundaries will be terminated. The boundary condition applied on the boundaries of this envelope is called *Vacuum Boundary*, where the particle goes through will be immediately killed. A box is often used as envelope, with a certain margin order not overlap with the problem geometry, as illustrated in Figure 2.9.

Another boundary often used in fusion neutronics analysis is *Reflecting Boundary*. When particle with direction  $\hat{\Omega}$  encounter a reflecting boundary surface with unit normal vector  $\hat{n}$ , the direction of the particle changes to

$$\hat{\Omega}' = \hat{\Omega} - 2(\hat{\Omega} \cdot \hat{n})\hat{n} \quad (2.18)$$

It means that the particle is reflected with respect to the surface normal [Rom14]. Reflecting boundary can be applied for the neutronics model of the tokamak machine, due to the circumferential symmetry. For example in Figure 2.9, a sector of the tokamak machine together with reflecting boundary can represent the whole tokamak. By applying this boundary condition, the geometry description is substantially simplified.

## 2.3 Basics of multi-physics coupled analyses

Thermal hydraulics and structural mechanics cover a broad knowledge system with many subjects. This section aims at providing a very rudimentary description aiming to understand the multi-physics coupling, especially how the relevant coupling terms, i.e. nuclear heating and temperature field, link these physics field together.

The fluid dynamics is essentially described by three conservation equations of the mass, momentum and energy. As energy is concerned in this work, the energy equation for an

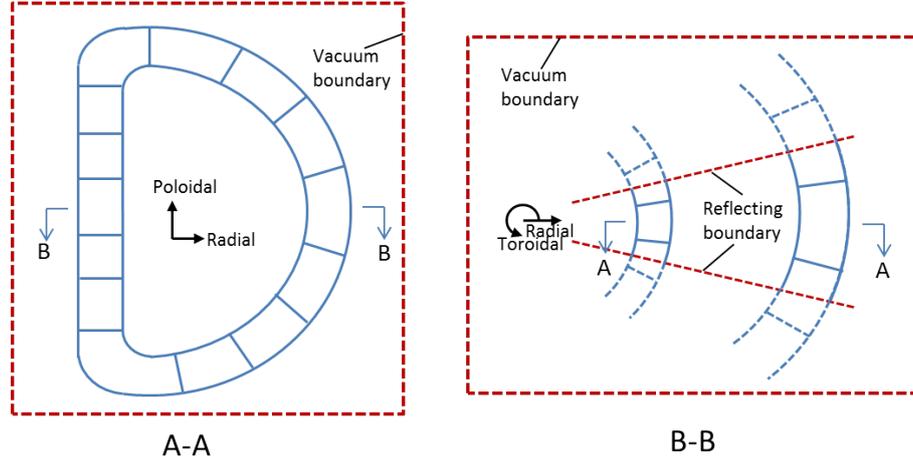


Figure 2.9: Boundary conditions for a neutronics model of tokamak machine. Vacuum boundaries are assigned for the boundary box of the sector model with a margin, and reflecting boundaries are assigned for the lateral plane of the sector.

unsteady, three-dimensional, compressible, viscous flow of a Newtonian fluid is given by [Wen08]

$$\begin{aligned}
 \rho \frac{D}{Dt} \left( e + \frac{V^2}{2} \right) &= \frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( k \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left( k \frac{\partial T}{\partial z} \right) \\
 &\quad - \frac{\partial (up)}{\partial x} - \frac{\partial (vp)}{\partial y} - \frac{\partial (wp)}{\partial z} \\
 &\quad + \frac{\partial (u\tau_{xx})}{\partial x} + \frac{\partial (u\tau_{yx})}{\partial y} + \frac{\partial (u\tau_{zx})}{\partial z} \\
 &\quad + \frac{\partial (v\tau_{xy})}{\partial x} + \frac{\partial (v\tau_{yy})}{\partial y} + \frac{\partial (v\tau_{zy})}{\partial z} \\
 &\quad + \frac{\partial (w\tau_{xz})}{\partial x} + \frac{\partial (w\tau_{yz})}{\partial y} + \frac{\partial (w\tau_{zz})}{\partial z} \\
 &\quad + \rho \vec{f} \cdot \vec{V} \\
 &\quad + \rho \dot{q} ,
 \end{aligned} \tag{2.19}$$

where  $\rho$  is density,  $t$  time,  $e$  internal energy per unit mass,  $\dot{q}$  the volumetric heat per unit mass,  $T$  the temperature,  $k$  the thermal conductivity,  $p$  the pressure,  $u, v, w$  the velocity component in  $x, y, z$  direction,  $\vec{f}$  the body force acting,  $\vec{V}$  the velocity vector, and  $\tau_{ij}$  a stress in the  $j$ -direction ( $j = x, y, z$ ) exerted on a plane perpendicular to the  $i$ -axis ( $i = x, y, z$ ).  $D$  denotes the substantial derivative as Equation 2.20:

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + (\vec{V} \cdot \nabla) , \tag{2.20}$$

where

$$\nabla \equiv \vec{i} \frac{\partial}{\partial x} + \vec{j} \frac{\partial}{\partial y} + \vec{k} \frac{\partial}{\partial z} . \tag{2.21}$$

## 2 Theories and methods

This equation can be derived from studying the energy conservation of an finite volume element in Cartesian coordinates. By introducing the *Navier-Stokes* equations, we can cancel the kinetic energy term from the left side of the Equation 2.19 [Wen08], and this equation can be written as

$$\begin{aligned}
 \rho \frac{De}{Dt} &= \frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( k \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left( k \frac{\partial T}{\partial z} \right) \\
 &\quad - p \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \\
 &\quad + \tau_{xx} \frac{\partial u}{\partial x} + \tau_{yx} \frac{\partial u}{\partial y} + \tau_{zx} \frac{\partial u}{\partial z} \\
 &\quad + \tau_{xy} \frac{\partial v}{\partial x} + \tau_{yy} \frac{\partial v}{\partial y} + \tau_{zy} \frac{\partial v}{\partial z} \\
 &\quad + \tau_{xz} \frac{\partial w}{\partial x} + \tau_{yz} \frac{\partial w}{\partial y} + \tau_{zz} \frac{\partial w}{\partial z} \\
 &\quad + \rho \dot{q} .
 \end{aligned} \tag{2.22}$$

The physics of Equation 2.19 and 2.22 can be separately explained. In Equation 2.19 ,

1. the left side of the first line represents the changing rate of energy, i.e. internal energy and kinetic energy, inside a fluid element;
2. the right side of the first line is the heat transfer into the volume by thermal conduction;
3. the sixth line is the work done by body force, e.g. gravitational, electric, magnetic force;
4. the seventh line is the volumetric heat source, which includes heat radiation, chemical heating, etc. In this work nuclear heating as volumetric heat source is greatly concerned.

and in Equation 2.22 ,

5. the second line is the work done by pressure conducted on the fluid volume;
6. the third to fifth line are work done by the normal and shear stress;

The governing equations are non-linear elliptic partial differential equations coupled with each other. Thus it is nearly impossible to find analytical solutions for a complicate 3D engineering problem. Numerical methods and algorithms are intensively studied, and been developed as a subject within CFD.

Equation 2.19 is applicable not only to fluid but also to solid. Considering a solid, the convective transport terms vanish and Equation 2.19 mutates to the heat conduction equation, which reads to:

$$\rho \frac{\partial(cT)}{\partial t} = \frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( k \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left( k \frac{\partial T}{\partial z} \right) + \rho \dot{q} \tag{2.23}$$

where  $e = cT$ , and  $c$  is the specific heat capacity of the solid material [Lie11]. Assuming  $c$  and  $k$  do not change along with time, space and temperature, the Equation 2.23 can be rewritten as the transient heat conduction equation

$$\begin{aligned}\rho c \frac{\partial T}{\partial t} &= k \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) + \rho \dot{q} \\ &= k \nabla^2 T + \rho \dot{q}\end{aligned}\tag{2.24}$$

In forced convective systems, the convective heat transfer term can be added into the Equation 2.24, by calculating the heat transfer coefficient from a thermal hydraulic simulation. The solution of Equation 2.24 is a temperature distribution varying in time and space.

When the temperature distribution is calculated, a thermomechanical simulation can be carried out for studying the thermal effect on structures, for example the thermal stress

$$\sigma_t = E \alpha (T - T_0)\tag{2.25}$$

where  $E$  is the Young's modulus,  $\alpha$  is the thermal expansion coefficient,  $T$  is local temperature and  $T_0$  is the "stress-free" temperature.



# 3 Investigations on codes, platforms and coupling approaches

Several subjects are related by the coupled multi-physics analysis, and many computation codes software are involved. It is important to conduct a broad and intensive investigation for selecting suitable candidates to establish the coupling system. Meanwhile, a suitable platform has to be chosen for realizing the whole system.

Figure 3.1 shows the mandatory modules and interlinks for the coupling system. It includes CAD software, mesh generators, MC modeling tools, MC codes, coupling tools, CFD codes, FE SM codes, and visualization tools. In addition, suitable CAD formats, mesh formats and discretized data formats are especially important to be investigated as the media for exchanging data between different modules.

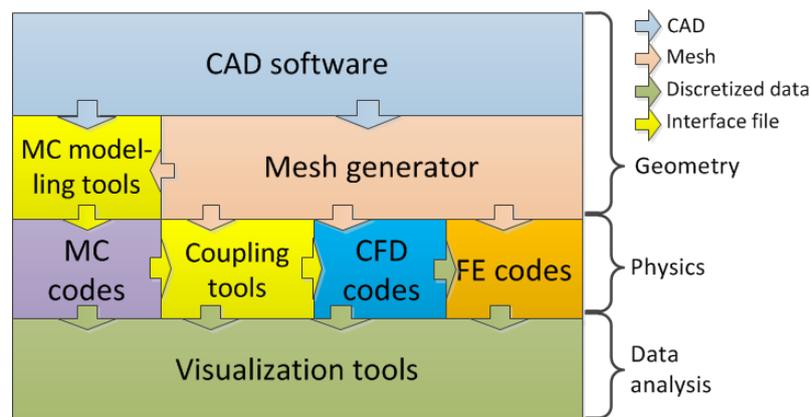


Figure 3.1: Modules and interlinks of an integrated coupled multi-physics simulation system. The arrows indicate necessary data transfer between linked modules.

## 3.1 Investigations on codes and platforms

There are hundreds of codes, software and libraries involved in different modules of the multi-physics analysis. For making decisions, four important criteria are adopted. In this paper, scores 0, 1, 2 are introduced to quantify the qualification capability, in which 2 indicates good, 1 indicates fair, and 0 means dissatisfaction qualification. Besides the following four criteria, complementary ones might be assigned to in some categories.

### 3 Investigations on codes, platforms and coupling approaches

#### C-1 Suitable for fusion applications.

The coupling approach developed here is aiming at meeting the needs of practical engineering applications. The codes and software should be well-accepted for handling typical issues in fusion engineering analysis, such as complex geometries, heterogeneous physical fields, etc.;

#### C-2 Good availability.

Good availabilities of codes are very important for implementing the integrated system. Open-source code will be given a score of 2 due to the free availability. Commercial or licensed codes are carefully scored with 1 if they are intensively applied in fusion engineering analysis.

#### C-3 Compatible with common-used data exchange formats.

Supporting suitable data format is very important for coupling between codes and software. Common-used formats will be based for this evaluation.

#### C-4 Good Application Programming Interface (API).

The chosen codes and software should provide high-level, comprehensive and well-documented API for coupling interfaces in order to facilitate the API learning and the development work.

In the following sections, tables are elaborated to allow for a comparison. As a convention, proprietary and open source codes are separated with a horizontal line. It is noted that the comparison carried out in this work, even though are quantified by scores, is based on a fusion technology view and in the general sense empirical. Therefore, it is only aimed at providing a guide for selecting suitable candidates for the integrated system, but not going to provide a ranking of these tools based on their capabilities.

#### 3.1.1 CAD, mesh and data format

This investigation covers formats for CAD, mesh and discretized data. For a CAD format, the criteria C-1 focuses on data consistency, which is the geometry loyalty during import and export. With respect to the mesh format criteria C-5 and C-6 are introduced, where C-5 indicates the support of both mesh and discretized data, and C-6 indicates the availability of data interpolation functions in the mesh library.

Almost every vendor of CAD software has their only CAD formats. The widely accepted proprietary and open source format are evaluated in Table 3.1. The STEP format is provided by the Open Cascade (OCC) Technology [CAS12], a 3D CAD modeling kernel consisting of reusable open source C++ libraries. STEP format is matching satisfactory in all criteria — fidelity loyalty, open access, and broad acceptance. CATIA and SAT format is constrained by licensing (C-2). IGES is evaluated to be slightly weaker than STEP format in C-1 and C-3 [MAA<sup>+</sup>10]. STL introduces facet approximation of a curved surface, which loses fidelity of the CAD data, but it is very useful for spatial discretization, i.e. mesh generation.

The evaluation of the mesh format is given in Table 3.2. The CFD General Notation System (CGNS)[PBC<sup>+</sup>00] consists of a collection of conventions and open-source

libraries for exchange of CFD data. It defines a general mesh and data format suitable for CFD and FE codes, as well as data visualization software. A small drawback is the data interpolation utility in CGNS system process only point data. MED format [Sala] is a underlying format for the open source SALOME platform[Salb], and offers library functions for mesh management. Its data interpolation functionality is attractive for general use as a coupling interface. VTK format [SAH00] is employed by many computational codes and software for mesh and data visualization. It is provided by the open source Visualization ToolKit (VTK), which is a software system for 3D computer graphics, image processing and visualization. The Comma Separated Value (CSV) format is a raw data format, which is not used for persisting mesh structure. It does not keep geometry information. But it is useful for transferring point-wise data, when a mesh geometry is not required. Abaqus format is an open and general mesh format defined in the Abaqus system. It does not persist discretized results and support data interpolation. It is noted that this format is adopted in MCNP6 for unstructured mesh geometries[Mar12]. Other formats such as Mesh-Oriented datABase (MOAB)[TMM<sup>+</sup>04] and OpenFOAM are not considered, although they are satisfied some aspects.

Table 3.1: Evaluation of the CAD data formats according to the specifications in Section 3.1.1(0 - nonsuited, 1-suited, 2- well suited).

CAD format	C-1	C-2	C-3	C-4	Total
CATIA	2	0	1	1	4
SAT	2	0	1	1	4
STEP	2	2	2	2	8
IGES	1	2	1	2	6
BRep	1	2	0	2	5
STL	1	2	1	2	6

Table 3.2: Evaluation of the mesh and data formats according to the specifications in Section 3.1.1(0 - nonsuited, 1-suited, 2- well suited).

Mesh format	C-1	C-2	C-3	C-4	C-5	C-6	Total
CGNS	2	2	2	2	2	1	11
MOAB	1	2	0	1	2	1	7
MED	1	2	1	2	2	2	10
OpenFOAM	0	2	0	1	2	0	5
CSV	1	2	2	2	0	0	7
VTK	2	2	2	2	2	0	10
Abaqus	2	1	1	2	0	0	6

To sum up the evaluation, the STEP and CGNS are chosen as key formats for respectively CAD and mesh data exchanging. The STL, MED and VTK format are also taken into account, when assessing the candidate codes and software.

#### 3.1.2 CAD modeling tools

Although there are hundreds of existing CAD software, Table 3.3 lists some frequently used package. Based on the purpose of this work, an important criterion C-5 is introduced for assessing CAD software —supporting of the Computer-Aided Engineering (CAE). More specifically, the software should provide functionalities such as geometry error repairing, in order to prepare suitable CAD models for engineering analysis, e.g. CFD, FE, etc.

CATIA and AutoCAD are very powerful in many aspects in Table 3.3, but naturally constrained by criteria C-2, C-3 and C-4. Therefore, they are not considered for our coupling approach. Oppositely, FreeCAD and OpenSCAD are not powerful enough for dealing with complicate and CAE relevant problems. ANSYS DesignModeler (DM) and SpaceClaim are aimed at providing CAE pre-processing on CAD models, especially SpaceClaim, which provides straightforward approaches like pulling, moving, filling and combining for geometry simplifications. The geometry module GEOM of the SALOME platform is based on OCC library in dealing with complex CAD models, and it provides importing and exporting interfaces in the STEP, BREP, STL format. It is able to share data with other SALOME modules, e.g. meshing module, and offers repairing functions for fixing geometry errors. Therefore, it can be considered as a balanced choice for CAD manipulations in our coupling system.

#### 3.1.3 Mesh generation tools

In this sections, C-1 is split into two sub-criteria — C-1.1 and 1.2 for evaluating the tools more specifically. C-1.1 indicates the availability of the common element shapes given in Figure 2.6, i.e. tetrahedron, hexahedron, pentahedron and pyramid. C-1.2 indicates the geometry adaptivity and the mesh tunability, which is somewhat empirical but still valuable for the evaluation.

From Table 3.4 it can be found that the software ANSYS Mesh, ANSYS ICEM-CFD, HyperMesh and ANSA are good commercial meshing tools. For example, ANSYS Mesh provides physics-dependent meshes for simulation modules such as Fluent, CFX and ANSYS Mechanical inside the ANSYS Workbench (WB) platform; ICEM-CFD is a well-known tool for generating hexahedral structural meshes; HyperMesh is efficient in geometry preparations and mesh generations. However, these meshing tools are not matching the C-2 and C-4 criteria very well. They are not preferred in this work to use as meshing module in the integrated system.

For open source meshing tools, the SALOME SMESH is a good choice. Although the mesh tunability is not perfect, SMESH provides common element shapes for various

Table 3.3: Evaluation of available CAD Software for fusion applications according to the criterion in Section 3.1.2(0 - nonsuited, 1-suited, 2- well suited).

Software	C-1	C-2	C-3	C-4	C-5	Total
CATIA	2	1	0	0	2	5
AutoCAD	1	0	0	0	0	1
SpaceClaim	2	1	2	1	2	8
ANSYS DM	1	1	2	0	2	6
FreeCAD	1	2	1	1	0	5
SALOME GEOM	2	2	2	2	1	9
OpenSCAD	0	2	1	1	0	4

Table 3.4: Evaluation of available mesh generation tools for fusion applications according to the criterion set in Section 3.1.3(0 - nonsuited, 1-suited, 2- well suited).

Software	C-1.1	C-1.2	C-2	C-3	C-4	Total
ANSYS Mesh	2	2	1	2	0	7
ICEM-CFD	2	2	1	2	0	7
HyperMesh	1	2	0	1	0	4
ANSA	2	2	0	2	0	6
GAMBIT	2	1	0	0	0	3
Pro-STAR	1	1	0	0	0	2
SALOME SMESH	2	1	2	2	2	9
Gmsh	1	1	2	0	1	5
Netgen	1	1	2	0	1	5
Tetgen	1	1	2	0	1	5

applications. It also offers mesh importing and exporting in the CGNS format, which is an important requirement in this evaluation. Moreover, it supports also internal data sharing with SALOME modules such as GEOM, and provides plug-ins for Netgen, GHS3D, BLSURF, etc. Tetgen is quality tetrahedral mesh generator, which is adopted in Chapter 4 for generating a special kind of tetrahedral mesh.

### 3.1.4 Monte Carlo transport codes

Again, criteria C-1 is split into two aspects indicated as C-1.1 and C-1.2. These MC codes are important to be well-validated for fusion studies (C-1.1), and they are expected to have a good adaptivity for complex geometries (C-1.2), e.g. support of mesh geometries. As these MC codes are all not supporting important formats such as CGNS, the criterion C-3 focuses on the capability of supporting a superimposed mesh tally. The comparison based on the ranks of the evaluation criteria is given in Table 3.5.

As shown in Table 3.5, MCNP is dominating in the entire comparison. It is well-validated and a widely used code in the fusion neutronics analysis. A rectilinear superimposed orthogonal mesh tally is provided in MCNP since version 5, and the unstructured mesh geometry is supported by the new version 6. The main drawback of MCNP is the strict licensing under the export control of the US Department of Energy, hence this code is not free distributable and not being able to be integrated in the source code level. TRIPOLI is a good alternative for our integration system because it is also intensively validated for fusion applications. Although it has limitations with respect to the geometry representation and the source code control, the use of TRIPOLI can be considered as a reasonable approach since version was integrated into SALOME platform. Therefore, it is a good choice for being the MC module in the integrated system if the SALOME platform is adopted at the end.

The Serpent code provides two advanced features — the superimposed unstructured mesh tally and the tessellated solid modeling. However, it is lacking of the coupled neutron-photon transport capability, which is required for fusion neutronics analysis. The open source toolkit Geant4 provides the tessellated solid modeling method as well as coupled neutron-photon transport. Meanwhile it is possible to be integrated in the system. However, some issues have to be addressed, which are specified in Chapter 5.

After this review, the MCNP code considered as primary option mainly, because of its

Table 3.5: Evaluation of different Monte Carlo codes for fusion applications according to criterion in Section 3.1.4 (0 - nonsuited, 1-suited, 2- well suited).

Codes	C-1.1	C1.2	C-2	C-3	C-4	Total
MCNP	2	2	1	2	0	7
TRIPOLI	2	1	1	1	1	6
PHITS	1	1	1	1	0	4
MCU	1	1	1	1	0	4
MCBEND	1	1	1	1	0	4
Serpent	0	2	2	2	0	6
Geant4	1	2	2	1	1	7
FLUKA	1	1	2	1	1	6

strongly routine used in fusion neutronics analyses. TRIPOLI and Geant4 are alternative candidates that can be considered for our integrated system.

### 3.1.5 CFD codes

In order to allow for a multi-physics coupling to a fluid dynamics problem, the CFD codes have to accept the distributed heat source provided by the neutronics calculation. Criteria C-4 is assessing this capability. Additionally, the conjugate heat transfer simulation approach is also important to be evaluated (C-5), as well as the availability of the coupling interface with FE codes for a subsequent TMA (C-6).

From the evaluation illustrated in Table 3.6 it can be found that packages Fluent, CFX and Star-CD exhibit a similar performance, except Star-CD which is not internally coupled with FE structural codes. Fluent and CFX codes are both computational modules of the ANSYS Workbench (WB) platform, and are internally coupled with the ANSYS Mechanical for FE structural analysis. A C language based API User Defined Functions (UDF) is provided in the Fluent code for loading nuclear heating data as heat source, and a similar API provided in CFX is called User Fortran. The NUMECA and PHOENICS are found less suitable for our purposes.

There are several open source alternative CFD codes for integration. Elmer is an open-source finite element software package for multi-physics simulations. One notable drawback of Elmer is that an externally provided distributed heat source cannot be loaded. OpenFOAM [JJT07] (Open Field Operation and Manipulation) is a C++ toolbox for solving comprehensive CFD problems. Library functions of this toolbox are used for programming CFD solvers based on a finite volume method, so that users have very high flexibility. Code\_Saturne is an open source product, which has been applied in fission reactor simulations. An advantage of Code\_Saturne is that it has a version in the SALOME platform, and can be coupled with the FE structural code Code\_Aster. A conjugate heat transfer simulation is achievable by coupling with a transient thermal simulation code called SYRTHES [RP12].

Therefore, the Fluent and CFX codes are considered further to be the most suitable candidates for our integrated system. While the Code\_Saturne is a good open source alternative for future consideration.

### 3.1.6 FEM structural mechanics software

In the view of the coupling for a TMA, six commercial codes and three open source codes are concerned. The criteria C-4 focuses on the availability of coupling with CFD codes, and an additional criterion C-5 is introduced to assess the TMA capability. These two criteria are important for the multi-physics coupling system. The evaluation is given in Table 3.7.

Many commercial codes are ready for fusion applications. The open source code is less attractive, because not many engineering level applications are found, neither with

### 3 Investigations on codes, platforms and coupling approaches

Table 3.6: Evaluation record of some CFD codes in the view of an integrated coupling scheme according to criterion in Section 3.1.5(0 - nonsuited, 1-suited, 2- well suited).

Software	C-1	C-2	C-3	C-4	C-5	C-6	Total
Fluent	2	1	1	2	2	2	10
CFX	2	1	1	2	2	2	10
Star-CD	2	1	1	2	2	1	9
NUMECA	1	0	2	0	0	0	3
PHOENICS	1	0	0	0	2	0	3
OpenFOAM	2	2	2	1	1	0	8
Code_Saturne	1	2	2	1	1	2	9
Elmer	1	2	1	0	2	2	8

a large amount of users. Among the commercial ones, ANSYS Mechanical is preferred, because it is widely accepted, and is ready coupled with CFD codes inside the ANSYS WB platform for TMA analysis. Some codes like ADINA, LS-DYNA, Code\_Aster and Elmer are linked with the in-house CFD codes, while these CFD codes are not preferred for our system. For open source alternatives, Code\_Aster might has an advantage that it has already been integrated into the SALOME platform, but it also has an drawback that it has to be coupled with the Code\_Saturne and SYRTHES simultaneously to conduct a coupled TMA analysis.

Table 3.7: Evaluation of several FEM structural mechanics software packages in the view of fusion applications according to the criteria in Section 3.1.6(0 - nonsuited, 1-suited, 2- well suited).

Software	C-1	C-2	C-3	C-4	C-5	Total
ANSYS Mechanical	2	1	2	2	2	9
Abaqus	2	1	1	0	2	6
Nastran	2	0	0	0	2	4
ADINA	2	0	2	1	2	7
LS-DYNA	2	0	0	1	1	4
COMSOL	2	0	1	1	2	6
Calculix	1	2	1	0	2	6
Code_Aster	1	2	2	1	2	8
Elmer	1	2	1	1	2	7

### 3.1.7 Visualization software

The available visualization tools, for which an evaluation record is presented in Table 3.8, exhibit a satisfying performance. For example, interfaces for common-used CFD and FEM software are provided, as well as user programming API. The open source ParaView and VisIt are preferred due to reduced licensing issues. VisIt is an interactive parallel visualization and graphical analysis tool based on the Visualization ToolKit (VTK) [SAH00]. ParaView is another powerful open-source visualization tool developed also based on VTK. Both support many commercial and free CFD and FEM solvers, e.g. ANSYS Mechanical, FLUENT, CFX, OpenFOAM, etc., and provides C++, python interfaces for user programming. One important information is that the ParaView software has a SALOME integrated version, so that mesh and data can be visualized in the SALOME platform.

Table 3.8: Evaluation of different visualization software for an integrated simulation platform according to the criterion in Section 3.1.7(0 - nonsuited, 1-suited, 2-well suited).

Software	C-1	C-2	C-3	C-4	Total
Tecplot	2	1	2	2	7
EnSight	2	0	2	2	6
AVS/Express	2	0	2	2	6
FieldView	2	0	2	2	6
ParaView	2	2	2	2	8
VisIt	2	2	2	2	8

### 3.1.8 Simulation platforms

The choice of a suitable platform is essential to realize a consistent integrated system. The evaluations have to cover physics simulation capabilities in neutronics, CFD and FE thermal and structural analysis (C-1.1), pre- and post-processing capabilities such as CAD modeling, mesh generations, and data visualizations (C-1.2). As important format for data exchange, STEP and CGNS formats have to be supported by the platform (C-3). Last but not least, suitable APIs are required for constructing the missing but important modules such as MC modeling and multi-physics coupling modules, which are crucial for the integrated system. The evaluations for different platform types are given in Table 3.9.

On one hand, two commercial platforms are suitable according to the evaluation of the physics modules (C-1.1) and pre- and post-processing modules (C-1.2). ANSYS Workbench contains a CAD modeling module, a mesh generator and a visualization module. It provides tailored pre- and post-processing for CFD module ANSYS CFX,

Fluent and Mechanical. COMSOL Multiphysics platform has these capabilities as well, albeit the physics modules are not our first choices. However, the critical issue of ANSYS Workbench and COMSOL Multiphysics platform is the missing capability for developing a new module. Therefore, they are not suitable for realizing our coupling system.

On the other hand, the two open source platforms, SALOME and Elmer, are not perfect in the view of the criteria C-1.1 and C-1.2. The physics modules of Elmer are internally coupled, whereas the CAD modeling, meshing and visualization capabilities are not powerful enough. Although Elmer is open source, new module cannot be straightforwardly integrated. SALOME is a simulation platform for integration of numerical codes. It contains the CAD modeling module GEOM, meshing module SMESH and the visualization module PARAVIS (full integrated ParaView). Although it has no built-in physics modules, the Monte Carlo code TRIPOLI, CFD code Code\_Saturne and FE structural analysis code Code\_Aster are already integrated in the SALOME platform, although they are not considered as first priority in the integration. Most importantly, the SALOME system allows integrating new functional modules. Based on these comparisons, SALOME is concluded to be the suitable platform for establishing the entire coupled system.

## 3.2 Survey of coupling approaches

### 3.2.1 MC geometry conversion approaches

The MC geometry conversion poses the foundation to conduct a neutronics analysis for fusion applications. There are several tools available for this purpose. MCAM (Monte Carlo Automatic Modeling system) [Wu09] is an automatic CAD to MC conversion software developed in FDS team, China; McCad [LFP14] is another open-source program developed at KIT; ANSYS DesignModeler provides also conversion and inversion function for MCNP geometry since version 11.0 [ANSa]; DAGMC [TWK<sup>+</sup>09] (Direct Accelerated Geometry for Monte Carlo) is developed in University of Wisconsin to be a MC geometry kernel using tessellated solids for MC simulation; ROOT [BR97]

Table 3.9: Evaluation of different simulation platforms for an integrated fusion application code system according to the criterion in Section 3.1.8(0 - nonsuited, 1-suited, 2- well suited).

Software	C-1.1	C-1.2	C-3	C-4	C-5	Total
ANSYS WB	2	2	1	2	0	7
COMSOL	2	2	1	1	0	6
SALOME	2	1	2	2	2	9
Elmer	1	2	2	1	1	7

is a system providing a set of object-orientated frameworks implementing with all the functionalities for MC geometry descriptions, particle tracking, and results data analyses. All mentioned codes are evaluated according to the previously mentioned scheme, and the results are shown in Table 3.10.

To be competent for MC geometry conversions in the coupling system, the selected tool should be able to handle highly irregular engineering CAD models (criterion C-1). Among these candidates, MCAM is perhaps the most stable and advanced tool, which is based on a commercial CAD kernel, and it is the reference code for ITER neutronics. McCad and DAGMC are also applied in many ITER neutronics analysis. The MCNP interface ANSYS DesignModeler is not very stable in converting engineering models. ROOT provides only primitives CSG and is thus less suitable for our purpose.

The next criteria C-2 is that the tool should have less license constraints, including less dependencies on the source code of MC codes. MCAM and ANSYS DesignModeler are proprietary software. DAGMC relied strongly on source codes, which is not favorable in our purpose because MCNP is controlled under restrictive license. ROOT relies also on source code of Geant4 and FLUKA. Therefore, the McCad code has an advantage in this evaluation due to its open source nature.

Additionally, the tool has to provide interfaces for the candidate MC codes typically MCNP, and also importing interface for STEP format (criterion C-3). Except ROOT, all tools are able to provide accepted geometry by MCNP. Last but not least, the tools have to be extensible, so that it can be developed with new approach for advanced mesh based MC geometry (criteria C-4). An open source program is preferred, so that the developments can be integrated on the source code level.

Based on this assessment, McCad is chosen as a suitable MC geometry conversion tool for being integrated in our system. It is not a perfect program, but it is an open source package providing a high flexibility for further improvements.

Table 3.10: Evaluation of different MC geometry conversion tools according to the criterion in Section 3.2.1 (0 - unsuited, 1 - suited, 2 - well suited).

Software	C-1	C-2	C-3	C-4	Total
MCAM	2	1	2	0	5
ANSYS DM	1	1	1	0	3
McCad	2	2	2	2	8
DAGMC	2	1	2	2	7
ROOT	1	2	0	2	5

### 3.2.2 Coupling interface between MC codes and CFD codes

The development of coupling interfaces between MC codes and CFD codes poses a big challenge in this work. Neutron physics and CFD are both sophisticated subjects with many scientific codes or programs involved, and they are based on a different physics knowledge basis and demand sophisticated computation methodologies. In this investigation, several literatures on developing coupling interfaces between MC and CFD codes are compared, in order to find a suitable approach for a coupled multi-physics analysis of a fusion system. However, the evaluations of these interfaces are carried out in a different way from previous sections, in order to provide more details of each interfaces.

Investigations have been aiming at evaluating interfaces for MC and CFD codes. Other coupling approaches, which are not based on MC or CFD codes, are ignored due to the suitability for our purposes. Several data mapping schemes have been adopted in these literatures, where the *one-one* scheme is an exact mapping of a cell in MC geometry to an element of a CFD mesh, the *point-point* scheme is mapping data from the nearest point, and the *volume-weighted* scheme is calculating the data weighted by the volumes of a target mesh element overlapping with source mesh elements. Detail descriptions of the point-point and volume-weighted schemes can be found in Section 4.5.2. The methods of obtaining the heating is given, also the availability and the applications of these coupling interfaces are checked.

Some of the coupling interfaces are implemented with a capability of thermal feedback, which is an important issue for coupled MC and CFD simulation. When the neutron cross-section of isotopes, especially the high-Z isotopes which have unresolved resonance regions, the cross-section are temperature dependent due to the Doppler Effect. Therefore, the temperature of different geometries calculated in the CFD codes provide a feedback to the MC simulation for updating the cross-sections, and the coupling process are iterated until converged temperature results are obtained. However, the thermal feedback is not significant for our coupling analysis, as will be discussed below.

The investigations of the different coupling approaches and their main features are summarized in Table 3.11. The first four approaches are obtaining heating on MC geometry cells, and they are mostly using the one-one scheme. These approaches are developed for fission reactor, where geometries are often regular. For fusion application, the two approaches used in [YNA<sup>+</sup>09] and [TWK<sup>+</sup>09] are based on the MCNP mesh tally, and adopting the point-point and volume-weighted mapping schemes for improving the geometry adaptivity. Moreover, the approaches applied for fission reactors have to consider the thermal feedback, since transuranium isotopes exhibit a strong resonance effect. However, the thermal feedback is not taken into account in both [YNA<sup>+</sup>09] and [TWK<sup>+</sup>09]. It is because materials used in fusion systems are mostly composed of low-Z isotopes (except tungsten), which do not have strong Doppler Effect. Therefore, the thermal feedback will not be considered in our coupling interface. It can be also found that most of the approaches are not freely available, except [TWK<sup>+</sup>09] which is based on the MOAB library. However, the MOAB library does not provide a volume-weighted data mapping scheme.

Based on this investigation, a new coupling approach has to be developed. This approach will be a generic approach, which is geometry adaptive, and flexible to allow for a many-to-many coupling instead of a single link between a MC and a CFD codes.

Table 3.11: Essential features to allow for a fusion technology relevant coupling interface between MC codes and CFD codes.

Approach	MC	CFD	Heating tally	Mapping	Thermal feedback	Applica-tion	Avail-able
[Hu08]	MCNP5	Fluent	cell	One-one	yes	fission	no
[BRP09]	MCNPX	CFX	cell	One-one	yes	fission	no
[STD <sup>+</sup> 07]	MCNP5	Star-CD	cell	One-one	yes	fission	no
[CRu11]	MCNP5	Star-CCM	cell	Point-point Point-point	yes	fission	no
[YNA <sup>+</sup> 09]	MCNPX	SC/Tetra	mesh	Volume-weighted	no	fusion	no
[TWK <sup>+</sup> 09]	MCNPX	Star-CD	mesh	Point-Point	no	fusion	yes

### 3.3 Decisions on candidate selections

Based on the broad investigations and inter-comparisons of various codes, software and simulation platforms, decisions have to be made in choosing suitable candidates for the coupling system. For the fusion specific context here the SALOME platform is considered as a suitable platform for implementing the system, and its geometry module GEOM, the meshing module SMESH and the visualization module PARAVIS (or ParaView) are also adopted. Regarding the MC codes MCNP5/6, TRIPOLI-4 and Geant4 and with respect to the CFD codes CFX and Fluent are chosen. As the FE structural analysis code the ANSYS WB platform is selected for solving the physics. Except TRIPOLI-4, these codes are not inside SALOME platform, hence the coupling with these codes have to be loosely implemented by means of external files.

About the MC modeling module in the Figure 3.1, the home developed open source code McCad has been adopted and improved. In this work McCad has been integrated into the SALOME platform with a new capability in manipulating mesh geometries. For the coupling tools, a new module has been developed in this work, providing many-to-many coupling between MC, CFD and FE codes. To fulfill the entire coupling system, data links have to be implemented, to ensure a high fidelity computation at simultaneously acceptable computation time and sufficient spatial accuracy.



# 4 A multi-physics coupling system

## 4.1 Design of the coupling system

A full cycle of coupled fusion multi-physics analyses usually start from 1) geometry design in CAD system, then 2) generation of meshes, 3) converting CAD models into MC neutronic models, 4) carrying out MC simulations, 5) processing and translating the nuclear heating, 6) execution of the thermal hydraulic and structural mechanical analyses, and finally 7) visualizing the result data. The CAD geometry is the major input of this process. Changing of geometry will result in a new process cycle. Therefore, the process will be iterated until an agreement between all modules is reached.

On the basis of the investigations in Chapter 3, a coupling system has been designed as illustrated in Figure 4.1. This system includes 12 functional modules, and they are interacted with each other via 20 data links. In this PhD work, 2 essential modules have to be implemented and integrated into the system, and 14 data links (or interfaces) have to be developed, so that a seamless coupling analysis can be realized. The workflow in the Figure 4.1 is explained in the following and as well as the corresponding tasks are described.

- 1) CAD modeling. The CAD geometries are created from the GEOM module, or imported via STEP, IGES and BREP format. The geometries will be carefully inspected; a geometry error will be fixed if found; and necessary simplifications and modifications are performed. All these works can be executed by the GEOM module.
- 2) Mesh generation. The meshes are generated in SMESH based on the CAD geometries from GEOM, or imported from other meshing tools using a CGNS and a UNV format. These meshes will be provided for the McCad module in generating MC model, or the multi-physics coupling module McMeshTran (see detail explanation in Section 4.5). An initiative studied on evaluating suitable unstructured meshes for MC simulations has to be carried out, since the meshes generated by existing meshing tools are not suitable for MC simulations.
- 3) Geometry conversion. The conversions of CAD and mesh geometries to MC geometries will be performed by the in-house McCad software. In order to take the advantage of managing and manipulating CAD and mesh data, McCad has to be integrated in the SALOME platform. The current McCad version provides interfaces for generating a CSG geometry for MCNP5 and TRIPOLI-4. In order to perform high-complexity geometry conversions, an interface is required for

#### 4 A multi-physics coupling system

converting unstructured meshes geometries accepted by MCNP6. Geant4 is not supporting half-space CSG. Conversion approaches have to be studied for modeling CAD geometry in Geant4. Furthermore, interfaces are needed for sharing CAD and mesh data between the McCad and the GEOM, SMESH module. Additionally, CAD data are expected to be visualized together with simulation results in the visualization module.

- 4) MC simulation. MC simulations are performed with the MC codes MCNP5/6, TRIPOLI-4 or Geant4. The MC geometry conversion provides only the geometry and material information. The particle source specifications, simulation set up and tally specifications are to be provided by users. Orthogonal or unstructured mesh tally also have to be set up for obtaining neutron results, e.g. nuclear heating distribution which forms the basis for the subsequent coupling.
- 5) Data translation. A new data translation module has to be developed and integrated in the system for performing the post-processing of nuclear heating results, mapping of heating data on the target CFD and FE meshes, and data export as heat source for CFD and FE simulations. This module is called McMeshTran (see Section 4.5 for detail explanation). McMeshTran is required to perform manipulations of mesh and data, and data mapping between unstructured meshes with common element types. Also, post-processing interfaces for MCNP5/6, TRIPOLI-4, Geant4, and coupling interfaces for Fluent CFX, ANSYS WB have to be developed. In addition, sharing the meshes with SMESH and sending data directly to PARAVIS will facilitate largely the mesh and data manipulation.

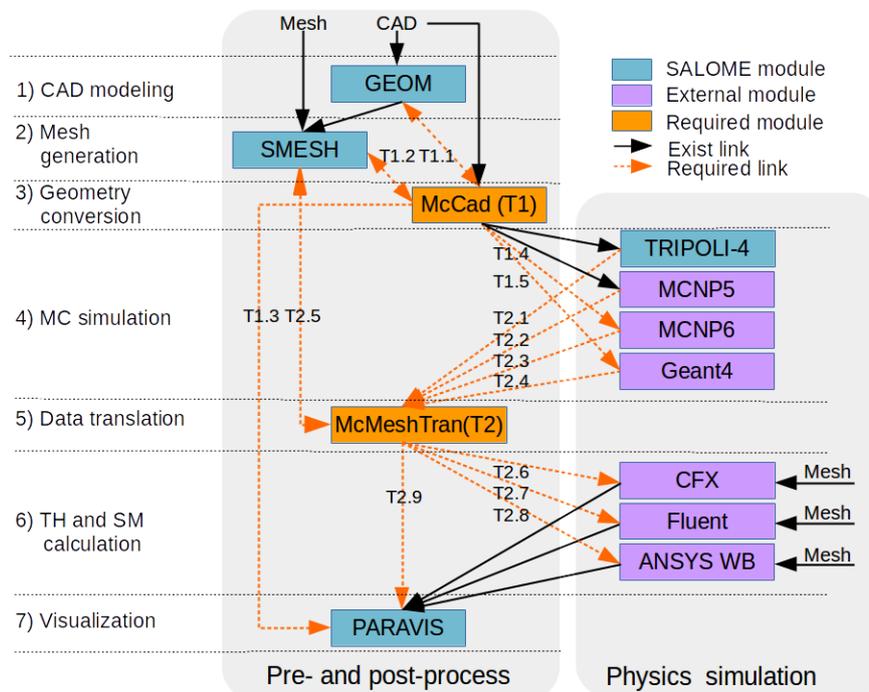


Figure 4.1: Sketch of the integrated coupling system and the computational modules involved. The required links are tagged with an index referring to Table 4.1

- 6) TH and SM analyses. The nuclear heating data are loaded as heat source for CFD and FE analyses. As discussed in Section 1.2.3, there are two approaches to conduct a coupled TH and SM analysis. In the first approach, nuclear heating data have to be written into the source subroutines and loaded during CFD runtime; in the second approach, the heating data are to be provided as point data loaded and automatically mapped to the FE mesh node by ANSYS WB. These coupling interfaces between CFD and FE codes, and also TMA are already available in the ANSYS WB.
- 7) Visualization. PARAVIS is a SALOME version of ParaView. It provides interfaces for post-processing results file of Fluent, CFX and ANSYS Mechanical. Therefore, the CAD, mesh, and results from all the physical modules can be consistently analyzed in one software package.

In order to provide a completed coupling system, two modules and 14 data links have to be developed. They are specified in Table 4.1 and indicated in the Figure 4.1. All of them have been addressed in this work, as discussed in the sections given in Table 4.1. The interface for Geant4 will be discussed later in Chapter 5. Although this system is a complete system, it cannot cover all potential issues in the coupled neutronics, TH and SM analysis. Besides the constraints defined in Section 1.3, this coupling system is also not capable to perform automatic iterative simulation results from the thermal stress induced geometry deformations, as well as the Doppler effect. The geometry deformations in the

Table 4.1: Tasks descriptions for establishing the coupling system.

Tasks	Description	Section
T1	Integration of McCad into SALOME platform	4.2.2
T1.1	Data exchange interface between McCad and GEOM	4.2.2
T1.2	Data exchange interface between McCad and SMESH	4.2.2
T1.3	Visualization of CAD solid in PARAVIS	4.2.2
T1.4	CSG and unstructured hybrid interface for MCNP6	4.3.5
T1.5	CSG and tessellated solid hybrid interface for Geant4	5.1.3
T2	Implement and integration of McMeshTran	4.5.1
T2.1	Post-processing interface for TRIPOLI mesh tally	4.6.3
T2.2	Post-processing interface for MCNP mesh tally	4.6.1
T2.3	Post-processing interface for MCNP6 unstructured mesh results	4.6.2
T2.4	Post-processing interface for Geant4 mesh tally	5.2.3
T2.5	Data exchange interface between McMeshTran and SMESH	4.5.1
T2.6	Coupling interface for CFX	4.6.4
T2.7	Coupling interface for Fluent	4.6.4
T2.8	Coupling interface for ANSYS Mechanical in WB	4.6.5
T2.9	Data visualization interface for PARAVIS	4.5.1

fusion reactor under normal operation are negligibly small, which does not introduce a notable influence on neutronic results. Also, as discussed in Section 3.2.2, the Doppler Effect in fusion material is not significant. Therefore, these two phenomena are not taken into account in the coupling process. However, they might be important to be evaluated in some special cases.

## 4.2 McCad integration into the SALOME platform

### 4.2.1 Introduction on the SALOME module integration

The framework of SALOME platform is shown in Figure 4.2. As discussed in Section 3.1.8, the SALOME platform contains built-in modules GEOM, SMESH and PARAVIS (ParaView). They are mainly relying on software technologies OCC, VTK, MED and CORBA, which are introduced in Chapter 3 except CORBA.

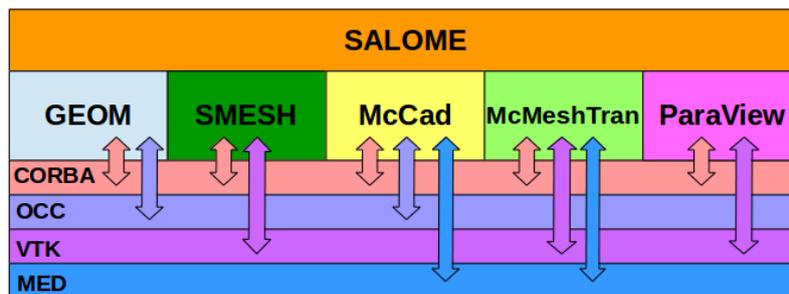


Figure 4.2: The framework of SALOME platform. The arrows indicate modules dependencies on the underlying technologies.

Common Object Request Broker Architecture (CORBA) is a standard defined by the Object Management Group (OMG) that enables data communications of codes and software developed in diverse programming languages and running on diverse computer platforms. A program is conventionally implemented as a *server* module and a *client* module. The server module consists of classes and functions of kernel algorithms, and the client module is implemented with user interactions, visualization, etc. The source codes of the server is wrapped with the CORBA interface layer using an Interface Definition Language (IDL) [Vin97], which produces a proxy source codes for a specific system and program language. The CORBA is employed in the SALOME platform for internal data communications among different modules.

Based on this architecture, every SALOME module is designed with a server and a client, and the client is usually combined with the Graphic User Interface (GUI). Kernel functions are implemented on the server side, and user interactions and data

communications are implemented in the GUI. One module is able to connect with many other servers, so that the data sharing between different modules are achieved. This architecture is also applied for integrating McCad into the SALOME platform.

## 4.2.2 The McCad module

In this work McCad has been integrated in SALOME as a new module. Starting from this section the name *McCad* or *McCad module* implicitly represents the new McCad version in the SALOME platform, while the stand-alone McCad version will be explicitly tagged with the word *stand-alone*. The McCad module is implemented by means of C++ object-oriented programming. As shown in the Figure 4.3, the McCad module developed consists of a server and a client constructed by well-organized classes.

The server consists of geometry object classes and the class for managing them. The geometry object classes include *Part*, *Group* and *Component*. A *Part* contains geometry data, e.g. CAD solids and mesh, which has a homogenized material inside the volume. It will be converted as a MC cell. A *Group* consists of a list of *instances* of *Part*, in other word an object based on the *Part* class. A *Group* is has only one material; A *Component* consists of instances of *Groups*. All instances of *Parts*, *Groups* and *Components* are identified with a unique index *ID*. A *Material* consists of a material number, a name and a density value and isotope composition. A list of *Component* instances and a list of *Material* instances are managed by the *Server Manager*. This manager class is also implemented with methods for managing the geometry objects and materials, and for converting the entire model into a MC model. The conversion is performed with the *McCad Kernel*, where the core algorithms such as decomposition and void generation are implemented. The classes in the server side are all wrapped with IDL interfaces.

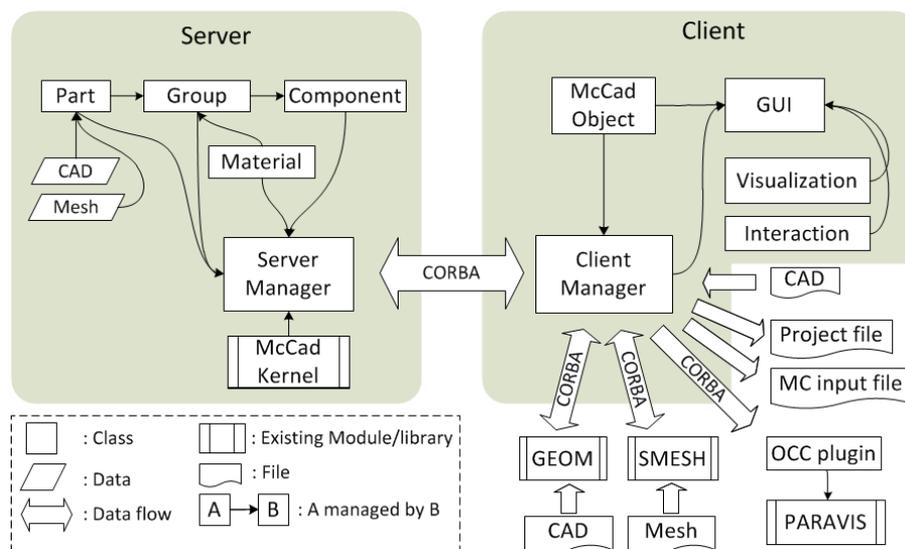


Figure 4.3: The implementation of McCad module. It is consisting of a server and a client separated by the grey area.

The client is implemented with user interactions, visualizations, data persistence and exchanging. A generic *McCad Object* class is created which links with an instance of a geometry object. For example, an instance of a Part is represented in the client side as a McCad Object, and the same for the instances of Group or Component. A McCad Object has a unique *Entry* in the SALOME platform, so that it can be shown in the object-tree or visualized in the view-window. These McCad Objects are created and managed in the *Client Manager* class. The Client Manager is connected with Server Manager via CORBA proxies, which are generated from IDL files, so that user interactions from the *GUI* class and other classes take effect. It is also internally connected to the GEOM, SMESH and PARAVIS modules for sending and receiving data. In addition, it is also responsible for saving the whole project, and exporting the converted MC model into files. Besides these two classes, the GUI class implements methods for creating the graphic windows, building up and managing the visualization, and dealing with the user interactions.

The development of the McCad module requires heavy programming effort, and need also understanding and handling of many technical details and issues on the SALOME platform mechanisms, establishing the CORBA data communication, utilizing the OCC, VTK and MED toolkits. These details and issues are not able to be completely stated in this paper. However, some of the important issues are discussed in the following to provide more information for those people, who are interested in further development of McCad.

#### **Geometry visualization**

Geometry visualization is a basic function for managing the CAD model in McCad. The SALOME platform provides a viewer window for displaying CAD solids. For displaying the CAD solid in this window, the McCad Object should be processed as a instance of the *SALOME Representation* class, which carries the data of CAD geometry, object identification and visualization attributes.

This process is shown in Figure 4.4. For visualizing a McCad object, the CAD data has to be obtained. The ID in the McCad object is used for locating the Part instance in the server. Then the CAD data are fetched from this instance, whereas it cannot be sent directly to the client, because CAD data are not recognized by CORBA. Instead, CAD data has to be encoded as CORBA octet-stream, and decoded from it after being transferred to the client. An instance of *Interactive Object* is created for interactive selection. It is linked with the McCad object via the unique entry. After that, an instance of *AIS Shape* (Application Interactive Services Shape) is created with the CAD data, an Interactive Object, and visualization attributes such as color, display mode, etc. In the end a SALOME Representation is created for displaying the AIS Shape in the viewer window.

#### **Communication with GEOM and SMESH**

The developments of bi-directional interfaces with GEOM and SMESH are required in Table 4.1. Such data transfer is performed through CORBA, and the CAD data are again encoded into octet-stream for transfer. To send the CAD data to the GEOM module, the GEOM server is called in the McCad Client Manager for creating a new *GEOM object* and displaying it in the GEOM GUI. To obtain CAD data from GEOM, the GEOM object associated with the selected object is first located, and the CAD data are then imported into

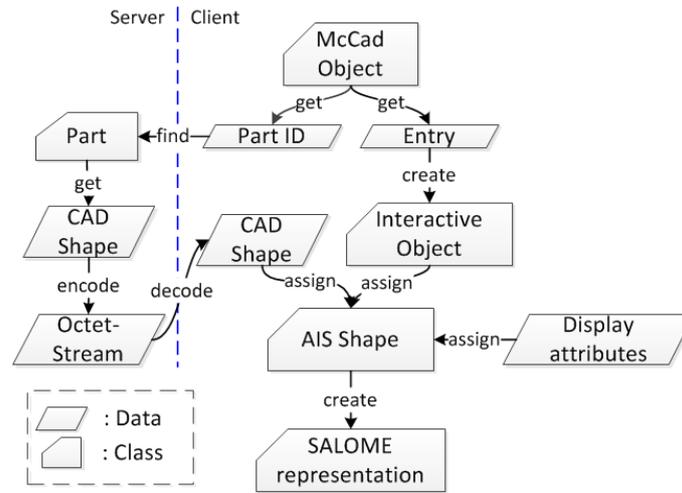


Figure 4.4: McCad visualization process. The process is begin with a McCad Object

the McCad module. For SMESH a similar approach has been used except for encoding and decoding of the data. Since an unstructured mesh is consisting of node coordinates and element connectivities, the mesh can be transferred to SMESH through CORBA as several arrays.

**Communication with ParaView**

The visualization of the CAD geometry together with results is a tool for the data analysis. For example, a superimposed orthogonal mesh tally would not represent the actual geometry, so the geometry is needed for locating the results. To achieve this, a plug-in called *OCCReader* has been developed for visualizing CAD models in ParaView. Additionally, a interface has been developed for sending CAD data to the *OCCReader* in PARAVIS.

Figure 4.5 illustrates the flow chart of the *OCCReader* plug-in. CAD shapes, including edges, faces and solids, are firstly meshed — the curved edges will be divided into line segments, and faces of solids are approximated by triangle facets. This meshing process will be discussed in detail in Section 4.3.4. The next step is converting those line segments and triangle facets from OCC data type into a VTK data object *PolyData*. Finally, the *PolyData* are submitted from the plug-in to ParaView for display.

The *OCCReader* is also adapted to PARAVIS, since PARAVIS is a wrapping module of ParaView. The sending of CAD data from McCad to PARAVIS is again realized through CORBA, by encoding the CAD data.

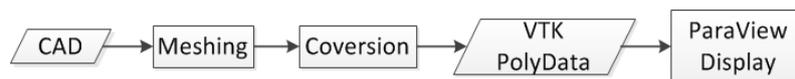


Figure 4.5: The process of CAD geometry for visualization in ParaView, which is implemented as a ParaView plugin.

### 4.2.3 The conversion process

After the McCad integration, the conversion of CAD to MC geometries has been achieved. An advanced conversion approach is provided for converting a hybrid CSG and unstructured mesh geometry for the MC code MCNP6, with the support to the traditional CAD-to-CSG conversions for MCNP5 and TRIPOLI-4. The flowchart of the conversion process is shown in Figure 4.6. It is noted that this process is semi-automatic, with some of the steps achieved interactively by users.

Before the conversion, the imported CAD model should be well-organized in McCad. For example, those parts with different materials are placed in different groups, and those parts, which will be meshed, are placed in separated components. The definitions of part, group and component are given in Section 4.2.2. If a component decided to be converted as CSG, the parts in this component have to be processed with a *decomposition* process, which is a mandatory process to convert a CAD solid into a CSG solid. The decomposition process might fail in certain situations. For example, the CAD solid contains an unsupported surface type for MC codes, or the OCC Boolean operation fails in cutting a CAD solid. In these cases CAD solids are recommended to be sent to GEOM module for a manual geometry simplification or an auxiliary decomposition. If a component is decided to use mesh, a new developed meshing approach, which will be discussed in Section 4.3, can be chosen to generate tetrahedral meshes. If this meshing approach is not preferred or fails, meshes can also be generated or imported in SMESH and assigned to parts in McCad. The meshed components are asked to be enclosed by an envelope and this envelope should not intersect with the mesh and other CSG cells. When the whole model is ready, a void generation process for producing the void space outside and within the solids is invoked for MCNP5/6 and TRIPOLI-4. However, this process is unnecessary for Geant4, since the void space are implicitly handled during the particle tracking. At the end the converted solid and void cells are exported into specific geometry descriptions in MC input files.

The decomposition process plays an essential role to convert CAD geometries into CSG. The methodology of decomposition is that the CAD solids in BRep format are decomposed into a set of *Sense-constant solid* which are constructed by *intersections* of half-spaces. Detail discussions on the decomposition process and also the sense-constant solids are given in Chapter 5. Besides decomposition, another crucial conversion process is the void generation. In MC codes such as MCNP, the whole space in the universe should be unambiguously defined, thus void should be extracted and described in a MC model as well. An efficient algorithm for void generations has been developed in McCad by L. Lu [LFP14].

Theoretically the decomposition and void generation processes do not introduce any geometrical approximations during conversion. In practice, substantial simplifications, so-called *clean-up*, are often needed to pre-process the CAD models, so that they become suitable for the conversion. As long as the CAD is ready, the generated CSG model can exactly persists the CAD volume. When using an unstructured mesh to represent the CAD model, geometrical approximations can hardly be avoided. How to generate precise mesh

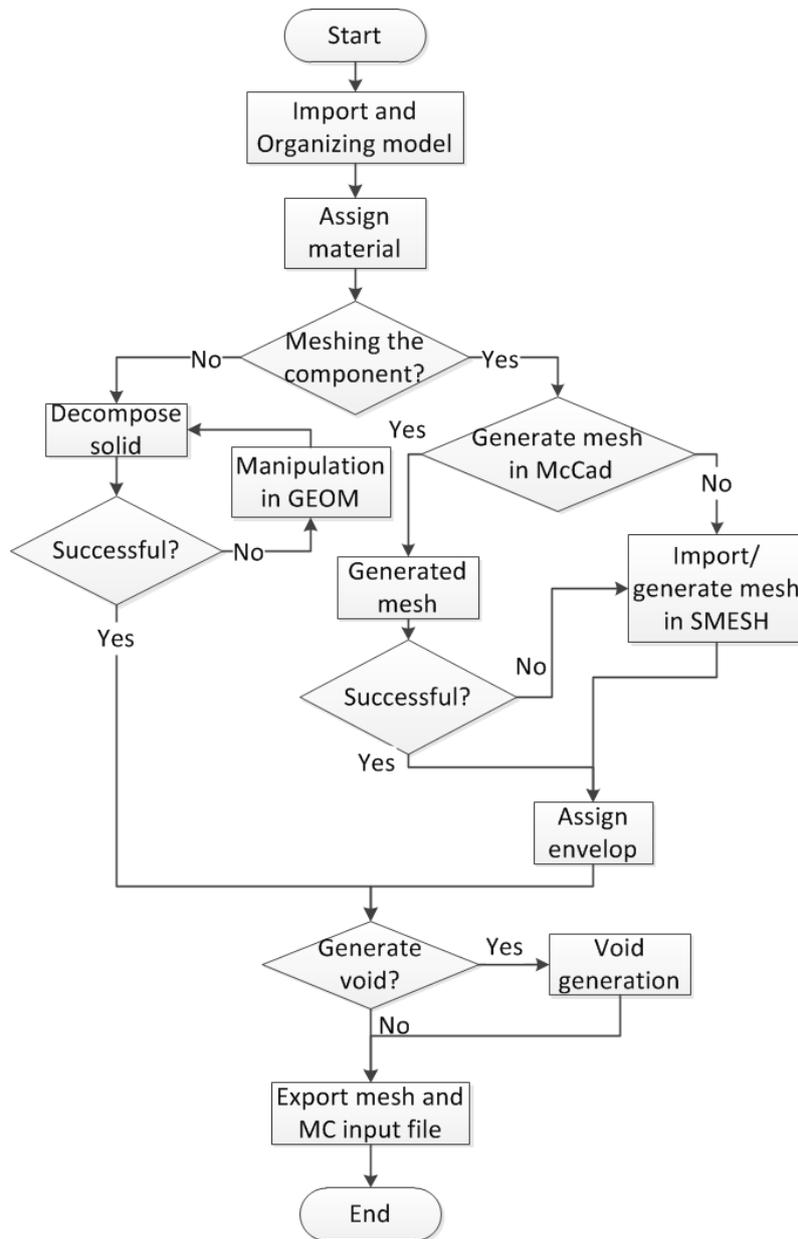


Figure 4.6: The conversion process of McCad. This process is a semi-automatic process with some user interactions.

geometries on CAD models and meanwhile maximize the MC simulation efficiency is a cutting-edge topic, which will be studied in this work in Section 4.3.

## 4.3 A novel meshing approach for MC geometry modeling

Performing MC simulation on mesh geometry is a state-of-art topic. It has not intensively discussed until mesh geometry is supported in the latest MCNP version 6 released in 2013. Although currently there are hundreds of mature meshing tools developed for CFD and FE analysis, whether these meshes are suitable for MC simulations is still not clear. In the section, we try to discuss this topic for the first time. More importantly, a new meshing approach has been developed in this work for MC mesh geometry modeling.

### 4.3.1 MC geometry modeling and results scoring

As J. R. Shewchuk [She02] pointed out, the quality of a mesh depends on the intended application and the numerical method employed. The application of unstructured meshes focus on two aspects, which are the MC geometry descriptions and the result scoring. In contrast to deterministic methods, e.g. FVM and FEM, in which simulation results are produced on the whole mesh geometry, the MC simulation method only records the results on regions, which are set as tally regions (or detectors). For example in the tokamak model depicted in Figure 4.7 results will be obtained only on the tally region of an in-board blanket. Detectors can be set in multiple regions, in the entire problem geometry, or even in a virtual geometry such as a superimposed mesh. Here, the meshes employed for scoring are called *MC scoring meshes* (or scoring meshes), and the meshes employed for geometry descriptions are called *MC geometry meshes* (or geometry meshes).

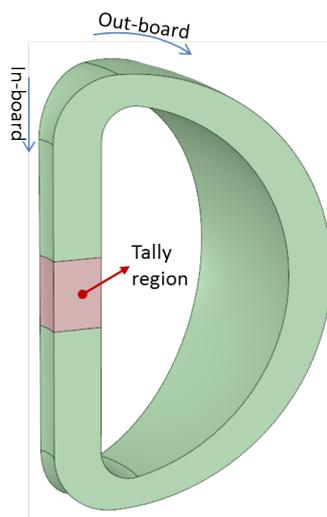


Figure 4.7: A tokamak sector with a tally region specified in one in-board blanket.

### 4.3.2 Quality of MC scoring mesh

Scoring meshes are used with the purpose to provide similar meshes as being applied for CFD and FE analysis. They aim to describe the problem geometry and capture the quantity and gradient of physical field. The element size and shape have strong impacts in mainly convergence performance in CFD and FE simulations. They influence also the statistics of MC results, since numbers of particles go into the element is related to the element size and shape.

The size of an element is usually indicated by its volume. The measurements of element shapes are more complicate. Some measure methods frequently used are *aspect ratio* and *skewness* [Flu, ST06], while the definitions are slightly different from place to place, as illustrated in Figure 4.8. Here, the aspect ratio is defined as the ratio of minimum distance from the element centroid to face centroids to the maximum distance from the element centroid to the nodes. The skewness is defined as the deviation between the element volume  $V_2$  of the element and the volume  $V_1$  of an equilateral element within the same circumsphere, i.e.  $(V_1 - V_2)/V_1$ . In most cases the aspect ratio and skewness give consistent indications on mesh shapes.

The element size and shape are of significance to represent a problem geometry. Because a mesh geometry is an approximation of a CAD geometry, it is required to be adaptive to the actual geometry. The approximation of the curved boundaries is the key issue in reducing the geometry discrepancy. This topic will be detailed discussed in Section 4.3.3. When a scoring mesh is superimposed on the actual geometry, attention should be paid to those elements crossing geometry boundaries, since the results on this element might be homogenized from several materials.

The element size and the shape impact essentially the quantity and gradient of a physical field. A decrease of element size improves the result resolution. A choice of high aspect ratio or high skewness elements may lead to miscellaneous of gradients. For example, good-shaped elements as depicted in Figure 4.9(a) captures much better the gradient than the *bad-shape*, or high aspect-ratio and skewness elements in Figure 4.9(b). A detailed discussion on this topic can be found in [She02].

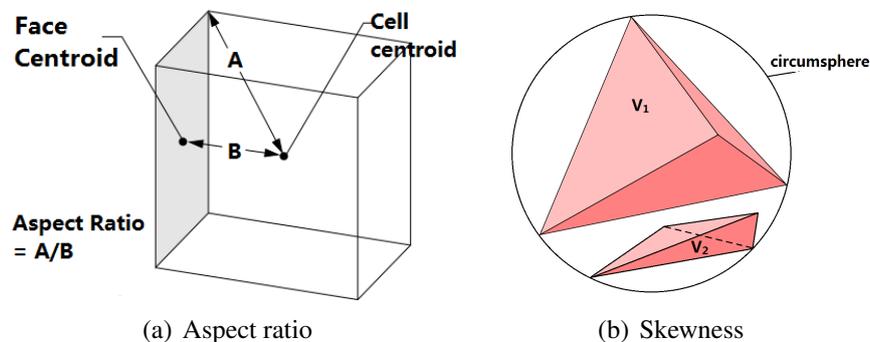


Figure 4.8: Definition of aspect ratio and skewness

#### 4 A multi-physics coupling system

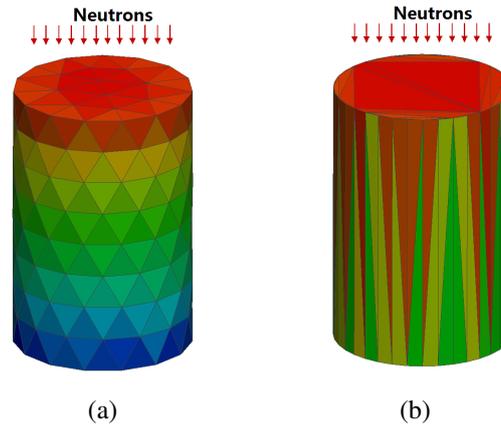


Figure 4.9: Scoring mesh with different element size and shape in revealing the neutron flux distribution.

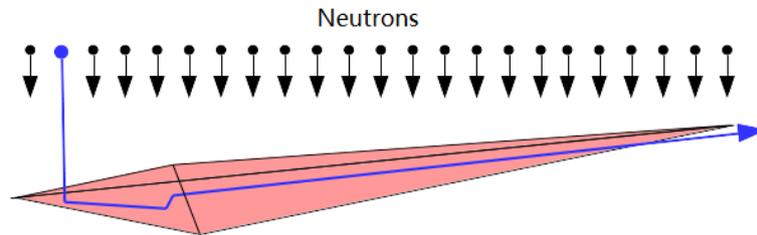


Figure 4.10: A highly skewed element. The particle illustrated in blue will cause statistical problem.

The element size and shape also impacts on the statistics in the neutronics calculations. When same amount of particles are sampled on the meshes of the same geometry, the mesh element with smaller size captures less particles, hence the statistics is worse. Elements with a high aspect ratio cause a statistical problem as well. For example in an element as shown in Figure 4.10, most particle histories will create short track lengths. If a collision happens and the particle is scattered along the direction illustrated in blue color in the figure, a long track line will be created. The wide-spread of track lengths lead to a large sample standard deviation  $S$  specified in Equation 2.14, thus increasing the statistic error of the results.

The studies on these influences are only qualitative nature without any rigorous mathematic proofs, hence some of the statements are not fully correct under certain circumstances. However, the quality of a MC scoring mesh is an interesting topic, which should be studied in more detail in the future.

### 4.3.3 Quality of MC geometry mesh

Different from the scoring mesh, the MC geometry mesh is not concerned on revealing the physical field, neither the statistics. To the leading order it is essential to represent accurately the CAD geometry and to ensure an efficient MC simulation. Based on this consideration, the quality assessment of a geometry mesh deviates considerably from a scoring mesh. Taking the meshes in Figure 4.9 for illustration, the mesh in Figure 4.9(a) has  $\sim 900$  elements, while the volume deviation from the CAD solid is 2.2%. Oppositely, the mesh in Figure 4.9(b) has only  $\sim 100$  elements and the deviation is only 0.5%. Therefore, the mesh in Figure 4.9(b) is preferable for a MC geometry representation.

Considering a solid with a volume size  $V$  represented by a first-order tetrahedral mesh, the absolute volumetric difference  $\delta$  between the mesh geometry and the actual solid is calculated by

$$\delta = \frac{1}{V} \sum_{i=1}^n |V_i - V_i'| \quad (4.1)$$

where  $i = 1, 2, 3, \dots, n$ ,  $V_i'$  is the volume of the element  $i$ , and  $V_i$  is the actual volume represented by the element  $i$ . Theoretically, mesh elements, which are inside the geometry or on the planar boundaries, has identical volume as the actual volume, i.e.  $V_i' = V_i$ . Using  $n_1$  to denote these amount of elements, and  $n_2$  to denote the amount of elements on the curved boundaries,  $n$  in Equation 4.1 can actually be replaced by  $n_2$ .

For improving the geometry accuracy, or in other word reducing the volume difference  $\delta$ , the number of elements  $n_2$  on the curved boundaries have to be increased. Regarding the computation efficiency, the MC simulation time increases with the number of mesh elements  $n$ . For improving the computation efficiency but retain the geometry accuracy, It is better to be keep  $n_1$  as small as possible but remaining  $n_2$  unchanged. In other words, the generation of a good geometry mesh requires elements inside the solid or on planar boundary surfaces to be controlled as less as possible, and elements on curved surfaces being generated appropriately to reduce the volume deviation.

A geometry mesh is considered *economic*, when it has less number of elements (or, small *mesh size*) and high geometry accuracy. An example of a economic geometry mesh is illustrated in Figure 4.9(b). As discussed in Section 4.3.2, this kind of mesh is not suitable for CFD and FE analysis. Therefore, a suitable meshing approach has to be studied to generate economic MC mesh for geometries.

### 4.3.4 A novel mesh generation approach for MC geometries

In this work, a meshing approach called *Tessellation-Tetrahedralization* (TT) is proposed to generate economic geometry meshes. The generated meshes with this approach are named *TT meshes*. In the first phase, the CAD geometries are represented by tessellated surface grids, in which curved surfaces are approximated by triangular facets. In the second phase, tetrahedral meshes are generated conforming to the surface meshes.

Tessellation, in other word *faceting*, is a common technique adopted widely in computer graphics, especially in CAD visualizations. In visualizing a CAD solid, the edges are divided into line segments, and faces are meshed with triangular and quadrangular facets. Computational methods to generate such surface mesh types are available in some CAD libraries like OCC. These surface meshes exhibit a excellent geometry adaptivity and accuracy. The accuracy of the surface meshes to approximate the CAD geometries is tuned by a parameter called *Deflection*, which is the maximum allowable deviation of distance from a mesh edge to the actual algebraic curve or surface. The sequence to create such a surface mesh is the following[Per04].

1. Discretizing an edge into line segments in the parametric coordinate, i.e. U coordinate, and then check the deviation in the 3D Euclidean space. A further division on the edge is made, if the deviation is larger than the deflection, as shown in Figure 4.11(a).
2. Mapping the wires consisting of segmented edges from the 3D Euclidean space to the 2D UV-domain of the face as depicted in Figure 4.11(b). A face in CAD system is defined as a parametric surface with a closed wire as boundaries, and the parametric domain usually defined by the parameters U and V.
3. The mesh generated in the UV-domain on a face is composed of segmented wires, as illustrated in Figure 4.11(c). Then the surface mesh in the UV-domain is mapped into a 3D space for deviation checking. Extra vertices are inserted to reduce the deviation of the mesh edge to the surface. The process is iterative until it falls below the deflection.

These steps are repeated for every face of every solid. In this way, the mesh edge is shorter along the direction with a higher surface curvature and vice versa. A geometry-adaptive surface mesh is created, and the number of facets in this mesh is optimally reduced. Here we called this mesh type *tessellated mesh*. In practice, a single deflection value is not suitable for all solids in one model. A relative deflection (or, *Tolerance*, abbr. *tol.* in the following sections) is adopted, which is a deflection value adjusted to the size of a solid. Usually the length of longest edge of the boundary box is used as estimator of the solid size.

The tetrahedralization is carried out with the open-source tetrahedral mesh generation code Tetgen [ST06]. A method called Constrained Delaunay Tetrahedralization (CDT) is employed, which conducts a Delaunay tetrahedralization on the vertices of the tessellated mesh. In case existing vertices are not sufficient for Delaunay tetrahedralization, very small amount of vertices will be introduced and inserted only on the boundary of the tessellated mesh [ST06], which preserves the accuracy of the tessellated mesh. The mesh generated with CDT method preserve the boundary of tessellated mesh to a large extend. It produces a minimum of mesh elements, thus perfectly meeting the requirement for an economic MC geometry mesh.

The TT meshing approach has been implemented in this work as a meshing functionality of McCad. The TT mesh is suitable for a MC geometry geometry description and the integral results scoring. For scoring distributions of neutron fields on tally region, the

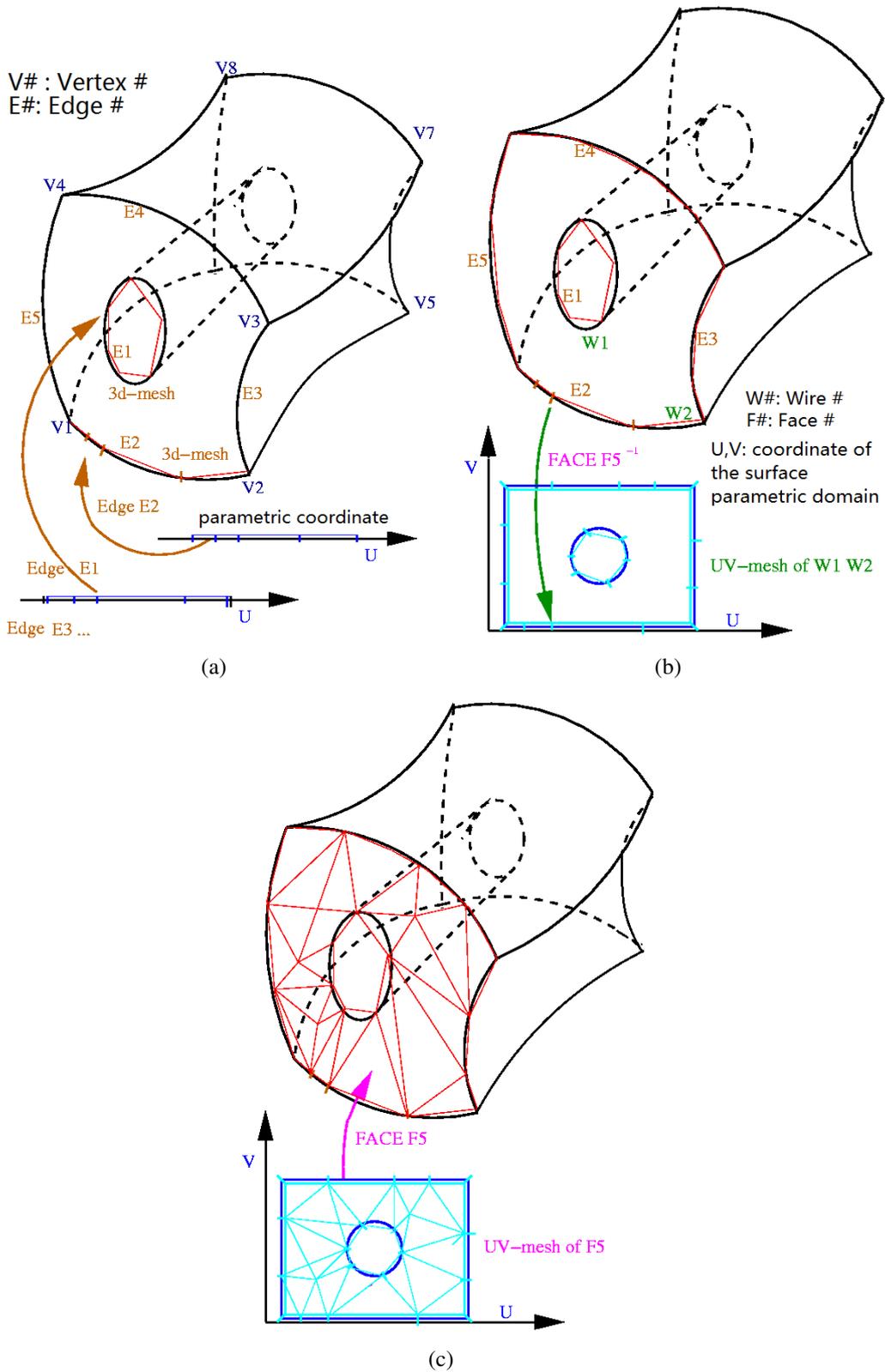


Figure 4.11: The tessellation process. The edge are divided into line segments, and face are meshed with triangle facets.

scoring mesh can be generated or imported from the SMESH module, and used together with TT mesh as shown in the example of Figure 4.12. The next section will discuss, how to use this mesh in MC codes, e.g. MCNP6.

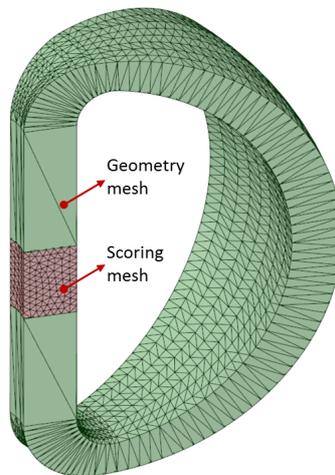


Figure 4.12: A tokamak geometry modeled with a scoring mesh for the tally region in Figure 4.7, and a TT mesh for the normal geometry.

### 4.3.5 Hybrid CAD and mesh conversion for MCNP6

In this work, a new McCad interface has been developed for generating MCNP6 models with hybrid CSG and mesh geometries. As already mentioned in Section 3.1.1, the general Abaqus format is adopted in MCNP6 for unstructured meshes. The meshes in this file are linked with the MCNP input file. Because there are many terms in both files, a short introduction on the terms is given for clear understanding of the interface.

Figure 4.13 is a diagram for illustrating the connections between the Abaqus mesh file and the MCNP6 input file. In the Abaqus mesh file, an *Abaqus part* (briefly as *part* in this paragraph and in Figure 4.13) is an independent mesh. It is constructed by a list of nodes with X, Y, Z coordinates and a list of elements with an element type and nodal connectivities. Elements in a part are grouped into one or several element sets called *elsets*, with each set assigned with a unique *Abaqus material*. These elsets are called *material elsets*. These Abaqus materials are used only for distinguishing elsets, and have neither meanings for real materials nor relations with MCNP materials. A material elset can be subdivided into more elsets called *statistic elsets*. In addition, a part can be instanced into many *instances* by associating them with different transformation matrices, and these instances are grouped into *assemblies*.

In the MCNP input file, the Abaqus material elsets or the statistic elsets contained in the material elsets are defined as *pseudo-cells*. They are linked with a kind of special *MCNP6 cells* which have *MCNP material* numbers but no surface descriptions. These new MCNP6 cells should be filled into a traditional CSG cell called *fill cell* or *envelope cell*, together with a *background cell*, which implicitly represents the void space inside the

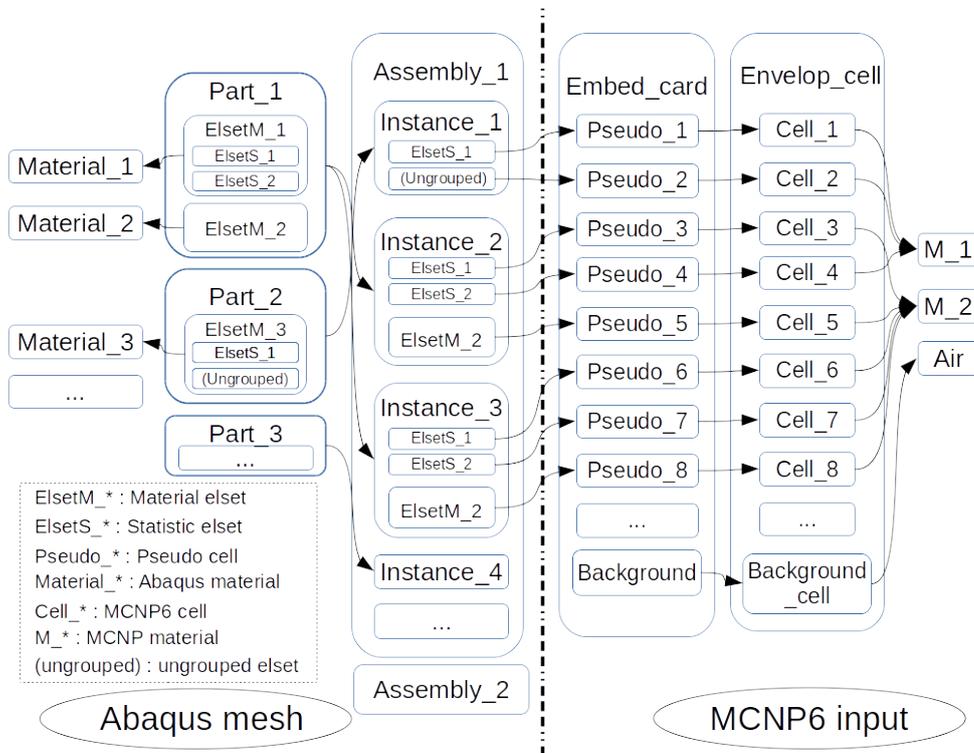


Figure 4.13: An example of a linking between MCNP cells and Abaqus meshes. In this example, three Abaqus parts are created. *Part\_1* has two material elsets and *Part\_2* has one. Each material elset is linked with an unique Abaqus material. *ElsetM\_1* has two statistic elsets, and *ElsetM\_1* has one statistic elset and one ungrouped elset. The Abaqus parts can be copied into one or more instance. The statictic elsets, ungrouped elsets, and material elsets, which are not subdivided, are linked with pseudo cells in the MCNP embed card, and ordered with their occurences in the instance. For example, *Part\_2* is copied as *Instance\_1*, thus the *ElsetS\_1* in *ElsetM\_3* links with the *Pseudo\_1*. At the end each pseudo cell is linked with a MCNP6 cell, specified as a list of MCNP cell number and pseudo cell number pair in the *embed\_card*.

mesh geometries. The linking table of the pseudo-cells and this new type of MCNP cells is specified in an input deck called *embed card*. Other parameters like the input file name, mesh format, etc. are also specified in this card. There are more rules on the specifications of elsets, the indexing of pseudo cells, etc., which can be found in [Mar12].

The above linking scheme is flexible, but also very complicate. Therefore, a simplified linking scheme is adopted in the new MCNP6 interface as shown in Figure4.14. In the McCad module, *McCad part* is defined in Section 4.2.2 as the basic geometry object in McCad, which is different from the Abaqus part. A McCad part is associated with an unstructured mesh, which is exported into the Abaqus file as an Abaqus part. An Abaqus part contains only one material elset with one statistic elset, and is associated by one instance without any transformation. All the instances are put into one assembly. On the other side, a McCad part containing an unstructured mesh is associated with a MCNP6 cell and a pseudo cell in the MCNP input file. If a part does not contain mesh, it will be converted into a traditional MCNP CSG cell. As mentioned in Section4.2.3, all the parts in a component have to be converted either to CSG or to mesh. In addition, the envelope solid assigned to a component will be converted as a envelope cell for filling all the MCNP6 cells. This one-to-one linking scheme is more straightforward for checking geometry during MC geometry preparation.

The MCNP6 hybrid geometry interface has the same features with the traditional MCNP interface in exporting the geometry description and material information. The mesh geometry will be converted from the MED format to the Abaqus format, which is constrained to the six element types supported in MCNP6 — first- and second-order tetrahedra, pentahedra and hexahedra.

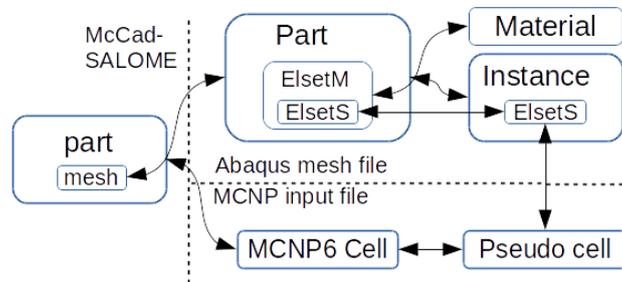


Figure 4.14: Linking scheme of the MCNP6 interface in McCad. The arrow indicates one-to-one linking relation. *ElsetM* and *ElsetS* represent material and statistic elset.

## 4.4 Verifications of MC geometry conversions

Verifications have been conducted for testing McCad in performing CSG and mesh conversions. The CSG model is considered as reference in the following tests since it is a mature and well-validated geometry type in MC codes. Evaluations on accuracy and efficiency of the TT meshing approach on handling complex models are the main purposes of the following two tests — the TBM test case model and the ITER benchmark model.

### 4.4.1 Verification of the TBM test case model

#### TBM test case geometry

A blanket test case model has been created based on the conceptual engineering design of the Helium Cooled Pebble Bed Test Blanket Module (HCPB TBM) [KB12]. The dimensions of this model are  $711 \times 655 \times 484$  mm in X, Y, Z direction respectively. Breeder unites (BUs) are reduced from 16 to 6, resulting in the change of the poloidal (Y) dimension from 1660 mm to 655 mm. The geometrical layout of this model is shown in Figure 4.15.

This TBM test case model is consisting of a first wall (FW), BUs, vertical and horizontal stiffening grids (SGs), lateral caps, and also manifolds and back-plates at the rear of the model. A BU is composed of beryllium pebbles, lithium orthosilicate sphere (OSI) pebble bed and cooling plates (CPs). All structural subcomponents, e.g. FW, SGs, Caps, CPs, are actively cooled by helium in flow channels. The geometry of this model is complex and challenging. Simplifications are introduced for subcomponents including FW, caps, CGs and manifolds, while the flow channels are kept in the structural components. But

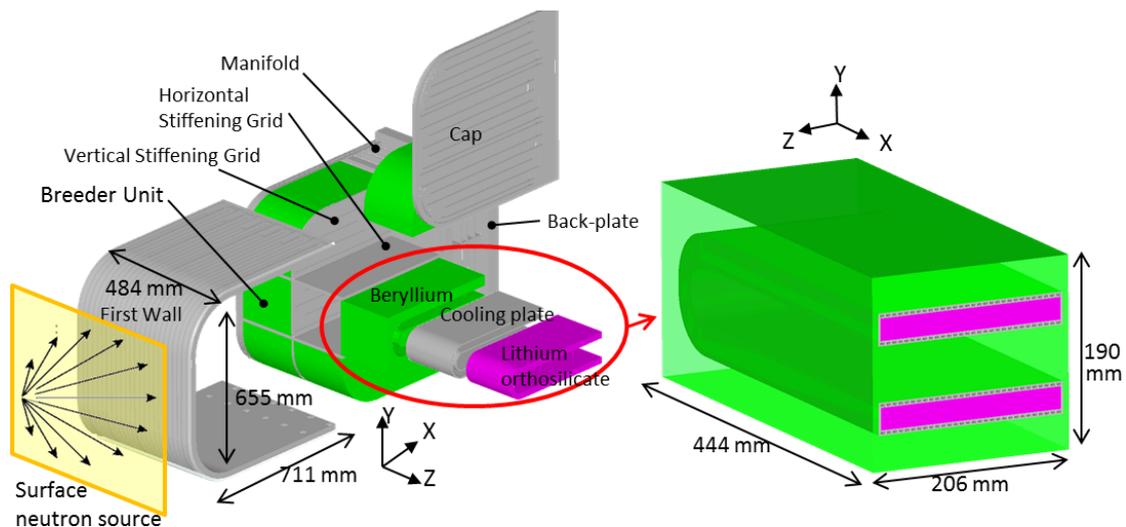


Figure 4.15: Exploded view of the HCPB TBM test case model (left) and its breeder unit (right).

the model is still very challenging for MC simulations. A typical example is that there are 6 pieces of CPs, and each one has 30 cooling channels with dimensions of  $2.6 \times 4.5$  mm. Other detailed geometry configurations, material compositions and properties, operations conditions can be found in [KB12].

### MC model conversions

A CSG model has been converted from the original CAD model using McCad. In order to accelerate the decomposition step of the conversion, manual auxiliary cuts on some CAD solids are conducted for some subcomponents such as the CP. Several overlaps have been detected in the model due to the failure of the overlap detection in McCad. They are fixed manually by checking the geometry using the MCNP plot function. After the conversion, a MCNP model with more than 13000 cells has been created. The CSG model of a CP is illustrated as Figure 4.16(a). The estimation of CPU time for converting this CSG model is rather difficult. The decomposition of the model takes approximately one week, and the void generation takes  $\sim 2$  hours on a computer with Intel Core i5-2500 (3.3GHz) CPU, based on the new algorithm implemented by Lu [LFP14] in 2014.

For TT meshing, meshes for all the TBM sub-components have been generated using the same computer. Five tolerance values (see the defined in Section 4.3.4) —  $1 \times 10^{-3}$ ,  $5 \times 10^{-4}$ ,  $2 \times 10^{-4}$ ,  $1 \times 10^{-4}$ ,  $5 \times 10^{-5}$  are adopted for testing the meshing performance at different tessellation accuracies. The generated meshes for a CP are shown in from Figure 4.16(b) to 4.16(f). It can be seen that the decrease of tolerance increases only the mesh size in the bending region of the CP, thus the volume accuracy is increased accordingly. The meshing processes have been successfully finished without any error. MCNP6 input files are generated as well as the mesh files.

For comparison, a mesh model has been created using ANSYS WB. A first-order (1st-order) mesh represents the TBM test case geometry. The default mesh setup is used in the meshing process, which has no constraint on the element size, smoothing, transition and other controllable parameters in the ANSYS Mesh tool. In order to avoid serious overlaps between different subcomponents, the mesh configurations on the contact boundaries are required to be consistent. The generated meshes for the CP with different tolerance and methods are depicted in Figure 4.16(g). They are imported into the SMESH module via CGNS format, and assigned for corresponding subcomponents in McCad.

Table 4.2 shows the meshing time, the numbers of elements and the volume deviations of the TT meshes. The volume deviations are calculated by

$$Deviation = \frac{\sum_{i=1}^n |V_i - V'_i|}{\sum_{i=1}^n V_i} . \quad (4.2)$$

where  $i = 1, 2, 3, \dots, n$  is the index for subcomponents of the TBM model,  $V'_i$  is the TT mesh volume, and  $V_i$  is the CAD volume of subcomponent  $i$ . Figure 4.17 shows the change of the volume deviation along with the decrease of the tolerance. A good logarithmic relation is found, with a small exception in the result of the tolerance  $1 \times 10^{-3}$ . It means that, by controlling the tolerance, we can effectively achieve a desired volume

deviation. However, the figure also demonstrates that the magnitude of elements and meshing time grow substantially. Fortunately, these values are well below those of the ANSYS mesh, according to the Table 4.2.

The volume comparison of mesh and CAD volume on individual sub-components are shown in Figure 4.18, where the ratios of mesh volumes to CAD volumes are plotted

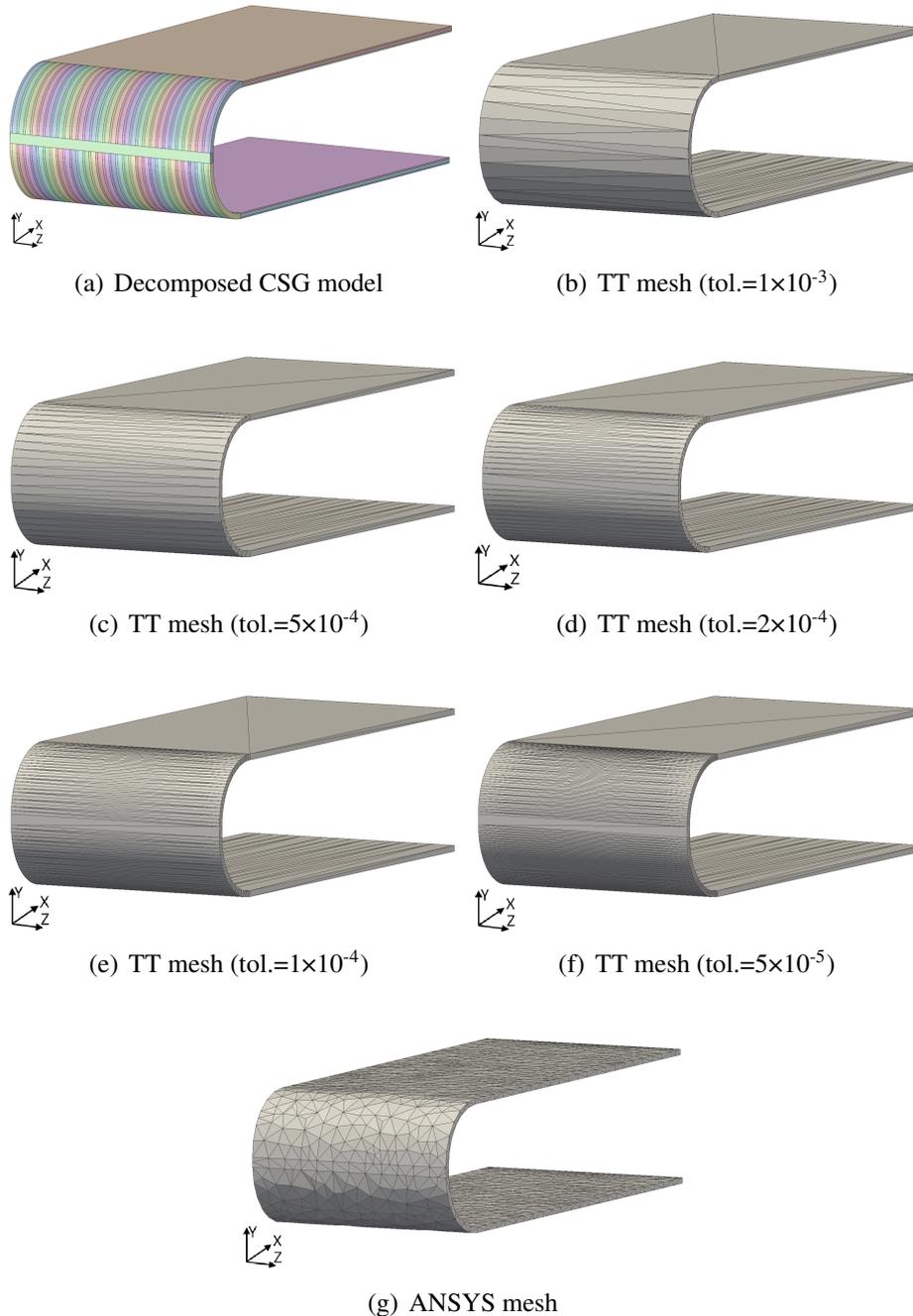


Figure 4.16: The CSG, TT mesh and ANSYS mesh model of a cooling plate of a HCPB breeding blanket.

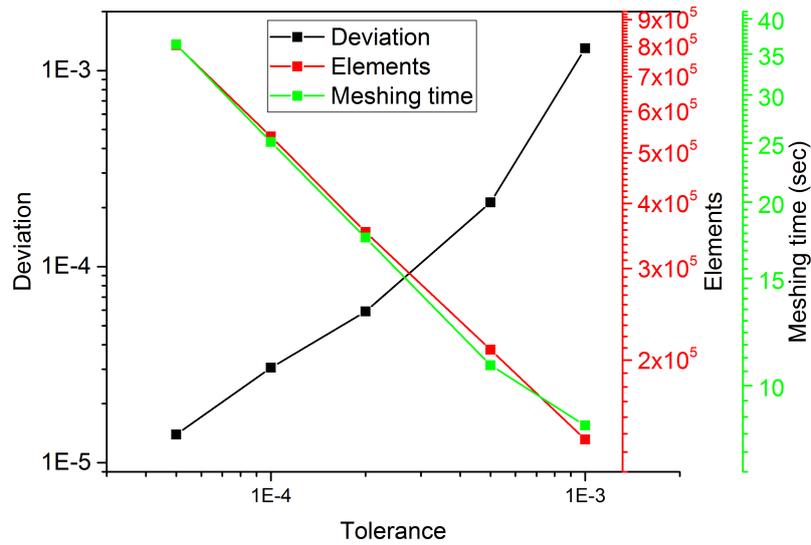


Figure 4.17: The volume deviations, number of elements and meshing time of the TT meshes for the TBM test case model.

Table 4.2: Comparison of TT mesh and ANSYS mesh on meshing time, element amount and volume deviation from the CAD geometry.

	Volume deviation	Meshing time (sec)	Elements
TT mesh (tol. $1 \times 10^{-3}$ )	$1.30 \times 10^{-3}$	8.6	$1.41 \times 10^5$
TT mesh (tol. $5 \times 10^{-4}$ )	$2.13 \times 10^{-4}$	10.8	$2.10 \times 10^5$
TT mesh (tol. $2 \times 10^{-4}$ )	$5.91 \times 10^{-5}$	17.5	$3.52 \times 10^5$
TT mesh (tol. $1 \times 10^{-4}$ )	$3.05 \times 10^{-5}$	25.1	$5.37 \times 10^5$
TT mesh (tol. $5 \times 10^{-5}$ )	$1.39 \times 10^{-5}$	36.3	$8.04 \times 10^5$
ANSYS mesh	$5.08 \times 10^{-3}$	702	$1.15 \times 10^6$

against their volumes. It can be observed from this figure that, the ratios are independent from the volume of the solids, which implies that the TT meshing approach is geometry adaptive. Adopting a tolerance value of  $1 \times 10^{-3}$  almost all meshes exhibit a volume deviation of less than 0.2%, except for the FW, which shows a deviation of 1.2%. A further reduction of the tolerance level to  $5 \times 10^{-4}$  is accurate enough for meshing the whole TBM test case. In this case the total volume deviation amounts to  $\sim 0.02\%$ , and the deviations for most of the subcomponents are within 0.1%.

The mesh generated by ANSYS WB is compared with the TT mesh in Table 4.2 as well. Considering a similar deviation, the comparison shows that meshing time and number of elements for the ANSYS meshes is significantly larger than those of the TT meshes, even though very few constraints are introduced in the meshing procedure. The volume comparisons on individual subcomponents are also shown in Figure 4.18, for ANSYS compared to the TT approach indicating a reduced performance of ANSYS. On one hand these results imply that the FE meshes do not represent suitable MC geometry mesh,

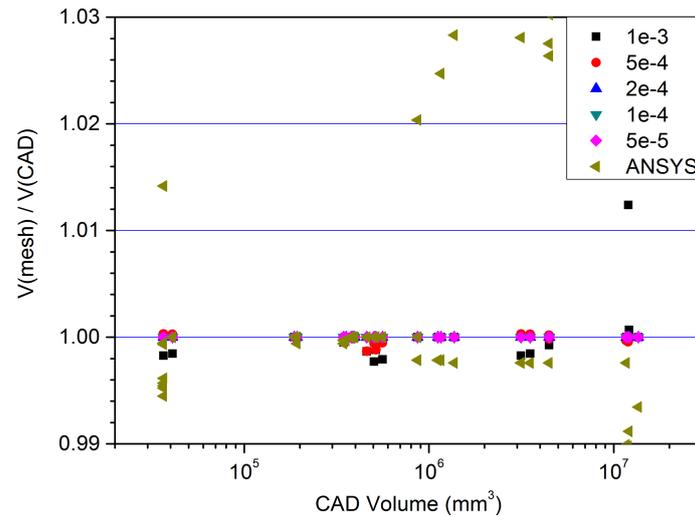


Figure 4.18: Ratios of mesh volumes to CAD volumes of individual sub-components of the TBM test case model.

because the size and shape of mesh elements are controlled in an unfavorable way. On the other hand they prove that the TT meshing approach is an efficient, economic and accurate MC geometry mesh generation method.

### Neutronics simulations

For neutronic calculation, a neutron source is placed at the front surface of the boundary box, as illustrated in the Figure 4.15. The neutrons are mono-energy neutrons with 14.07 MeV, and a ITER-like cosine angular distribution. As illustrated in the Figure 4.15, neutrons have largest probabilities to go in the perpendicular direction, and smallest probabilities to travel paralleling to the surface source. An average neutron wall loading of  $0.72 \text{ MW/m}^2$  has been considered, which corresponds to an incident source intensity of  $1.06 \times 10^{17} \text{ n/s}$ . Reflecting boundaries are assigned on the surfaces of the boundary box in poloidal (Y) and toroidal (Z) directions.

MCNP6-1.1(beta) and FENDL3.0 [LAN11] data libraries have been employed, and the simulations have been carried out on a computing node with 16 cores in the HELIOS supercomputer system at International Fusion Energy Research Centre, Japan.  $10^6$  neutrons are simulated for each case, and the simulation time is shown in Figure 4.19. The figure illustrates that the computational time in case of TT meshes are 7 to 9 times slower than those of the CSG model. The ANSYS mesh exhibits the computational time almost 11 times slower than the CSG.

The nuclear heating results are computed on the beryllium, CP and ISO of the BU highlighted in Figure 4.15. The results are obtained for the CSG, TT mesh and ANSYS mesh model, and given in Table 4.3. The CSG results are considered as references, which have negligible statistical errors below 0.02%. Figure 4.20 illustrate the results of Table 4.3 in the view of difference TT mesh tolerances with respect to the nuclear heating. It can be observed that the physical results are become insensitive to the mesh resolution, when the tolerance falls below  $5 \times 10^{-4}$ . One possible reason is that other sources of errors

#### 4 A multi-physics coupling system

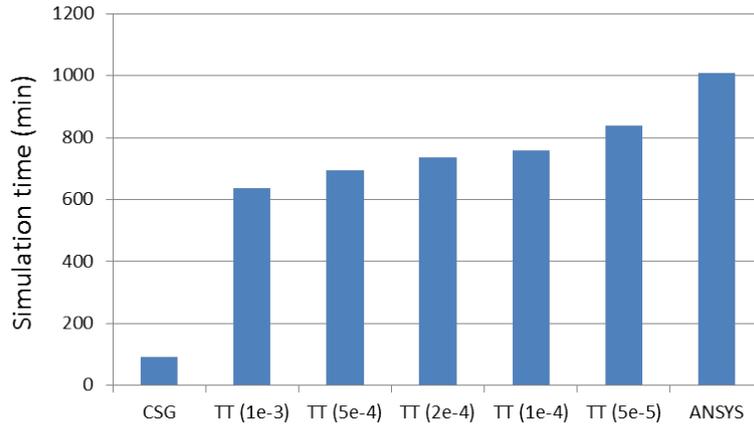


Figure 4.19: Simulation time using MCNP6-1.1(beta) with  $10^6$  particle simples.

Table 4.3: Comparing the nuclear heating results on the BU subcomponents highlighted in Figure 4.15. The results of TT mesh and ANSYS are divided by that of CSG model. The statistical errors of CSG results are beneath 0.02%.

CSG heating (W)	Beryllium ( $1.5555 \times 10^4$ )	CP ( $1.8036 \times 10^3$ )	OSI ( $1.0862 \times 10^4$ )
TT mesh (tol. $1 \times 10^{-3}$ )	$1.0051 \pm 0.07\%$	$1.0006 \pm 0.12\%$	$0.9976 \pm 0.08\%$
TT mesh (tol. $5 \times 10^{-4}$ )	$1.0021 \pm 0.07\%$	$1.0008 \pm 0.11\%$	$0.9982 \pm 0.08\%$
TT mesh (tol. $2 \times 10^{-4}$ )	$1.0022 \pm 0.07\%$	$1.0024 \pm 0.11\%$	$0.9984 \pm 0.08\%$
TT mesh (tol. $1 \times 10^{-4}$ )	$1.0021 \pm 0.07\%$	$1.0008 \pm 0.11\%$	$0.9979 \pm 0.08\%$
TT mesh (tol. $5 \times 10^{-5}$ )	$1.0023 \pm 0.07\%$	$1.0013 \pm 0.11\%$	$0.9979 \pm 0.08\%$
ANSYS Mesh	$0.9973 \pm 0.07\%$	$1.0796 \pm 0.11\%$	$0.9809 \pm 0.08\%$

are become dominant instead of the geometric error, which have not been systematically studied by now. However, considering a tolerance value of  $5 \times 10^{-4}$  is sufficient to attain an accuracy better than 0.3% for the nuclear heating on the TBM test case model.

Regarding the ANSYS mesh, a significant overestimation of the nuclear heating is obtained in the CP. The large difference originates likely from the volume difference between ANSYS mesh and CAD model. A close view on Figure 4.16(g) exhibits that the geometry difference occurs mainly in the bending part of the CP, which is intensively heated by the high neutron flux. Therefore, the ANSYS mesh cannot be conceived suitable for a MC geometry description.

Conversely, the TT meshing approach seems very promising for modeling the CAD geometry in MC codes. The generated TT mesh with the tolerance value of  $5 \times 10^{-4}$  is sufficient for a precise modeling of the TBM test case. The total volume deviation is about  $\sim 0.02\%$  only, and the nuclear heating results agree rather well with that of CSG model within 0.3%.

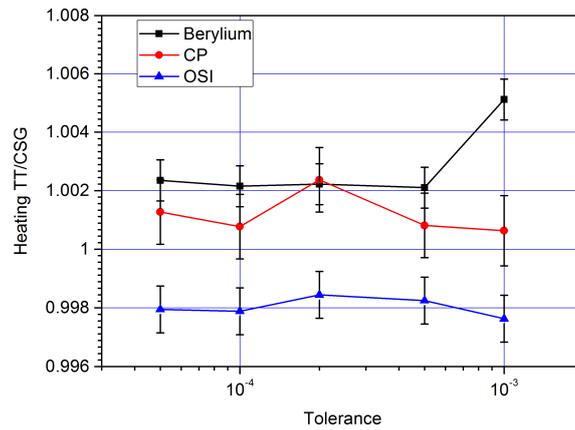


Figure 4.20: Comparison of nuclear heating results from the TT mesh model by dividing them with that of CSG model as function of the chosen tolerance level. The CSG results are considered as reference with statistical error beneath 0.02%.

### 4.4.2 Verification of the ITER Benchmark model

The ITER Benchmark model is a 40° toroidal sector of ITER[WFF<sup>+</sup>08], as shown in Figure 4.21. The model is simplified from the full detail CAD model, and only planar surfaces are included. Some complex components are represented as blocks with homogenized layers. The model consists of more than 900 CAD solids in total. The

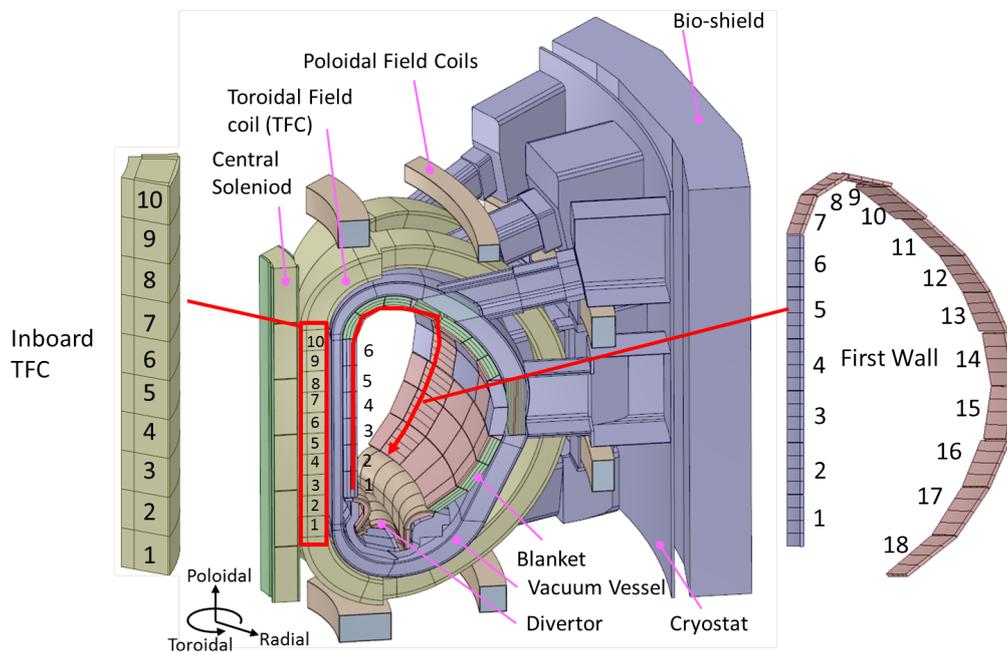


Figure 4.21: CAD model of the ITER Benchmark model. To exam the simulation, every first wall high-lighted is segmented into 4 parts, and the high-lighted inboard TFC is segmented into 10 parts.

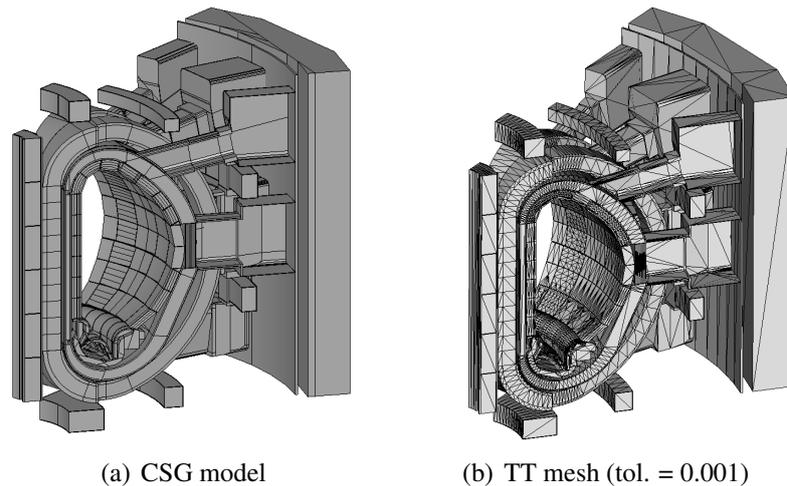


Figure 4.22: McCad converted CSG and TT mesh models of ITER Benchmark.

MCNP CSG model and the TT mesh model have been generated and compared acting as second verification case.

### MC model conversions

The CSG conversion is based on a pre-decomposed CAD model shown in Figure 4.22(a), hence only the void generation are performed for producing the CSG model. The void generation step takes  $\sim 2$  hours computation time on a computer with a Intel Core i5-2500 (3.3GHz) CPU. After the conversion, the generated model is checked with MCNP plots. Discovered geometry errors are solved by modifying the CAD geometries, following the new conversion process.

Similarly to Section 4.4.1, TT meshes are generated using the same computer with five tolerance levels —  $1 \times 10^{-3}$ ,  $5 \times 10^{-4}$ ,  $2 \times 10^{-4}$ ,  $1 \times 10^{-4}$ ,  $5 \times 10^{-5}$ . The generated mesh model with tolerance value of  $1 \times 10^{-3}$  is depicted in Figure 4.22(b). About ten solids encounter occasional failures in meshing, which mainly arise from two reasons. One reason originates from the geometry representation of the CAD solids, for example, a solid might have not fully closed boundaries. There are two solids with this problem and they are fixed by slight modifications of the CAD solids. Another reason is due to failures of Tetgen, when lower tolerances are used. The reason for the Tetgen failure is to be investigated by now. The meshes for these problematic solids are replaced with meshes owing a higher tolerance. Additionally, a failure of MCNP6-1.0 has been identified when using a mesh with only one element. These tiny meshes with several cubic millimeters in volumes are removed. Finally, the mesh envelope is created using a cylinder sector, which fits into the shape of the whole model, and then the MCNP6 input files and the mesh files have been generated.

Figure 4.23 shows the change of the total volume deviation, element amount and meshing time of the TT meshes along with the decrease of the tolerance. The total volume deviation is calculated using Equation 4.2. A logarithmic relation is found in the trend of the volume deviation, except for the tolerance value of  $2 \times 10^{-4}$ . The element amounts

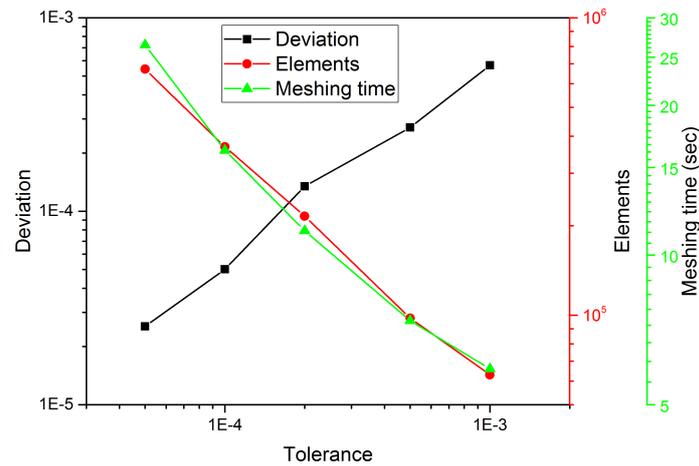


Figure 4.23: Change of the total volume deviation, element amount and meshing time for the TT mesh of ITER benchmark model along with the decrease of the tolerance. The volume deviation is calculated according to Equation 4.2.

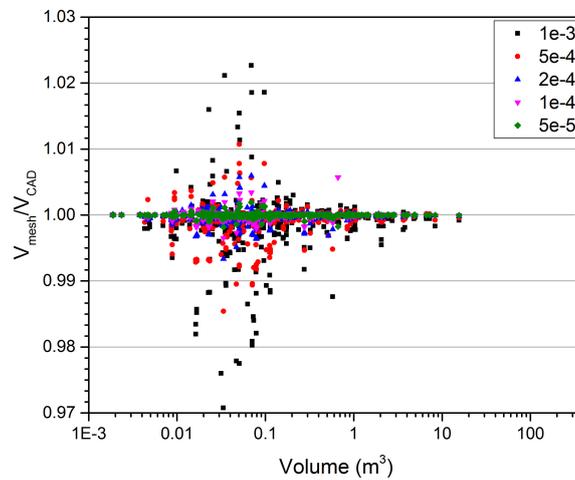


Figure 4.24: Ratios of mesh volumes to CAD volumes for individual solids of the ITER benchmark model.

and the meshing time are less that of the TBM test case (see Figure 4.17 ), even though the ITER benchmark model looks much larger and more complex by eyes. Figure 4.24 shows the volume deviations for individual solids. They are more spread than that of the TBM test case in Figure 4.18, which is due to the irregularity of the geometry shapes in the ITER benchmark model. Most of the volume deviations are below 0.5% in the TT mesh model with a tolerance value of  $2 \times 10^{-4}$ . These data give provides the evidences for a precise and efficient representation by the TT meshes.

### Neutronics simulations

MC simulations are carried out using these two models. MCNP5-1.6 and MCNP6-1.0 are used for computing the CSG model and the TT mesh model, and the JEFF3.1.1 data library is used in both cases to ensure the comparability with respect to the nuclear data.

#### 4 A multi-physics coupling system

The source description is extracted from the ITER B-lite reference model. The lateral faces of the mesh envelope are set as reflecting boundary, as illustrated in Figure 2.9. Due to the limitation in computing resources, only the TT mesh model with tolerance value of  $1 \times 10^{-3}$  has been evaluated and is discussed. The total volume deviation for this mesh model is  $\sim 0.05\%$ , and the deviations for individual solids are mostly within 3%.  $5 \times 10^8$  neutron histories have been simulated in both CSG and TT mesh model. The lost particles are both negligibly few.

The neutron wall loading (NWL) and nuclear heating in the toroidal field coils (TFCs) have been evaluated. The NWL calculation is for comparing the neutron source using the un-collided neutrons, thus materials are not assigned. Because the surface flux tally on the unstructured mesh is not supported yet by MCNP6, the volume fluxes in the FW are obtained in both models using MCNP cell-based tallies. The neutrons are killed after penetrating the FW, in order to prevent neutrons reflecting out again when they travel toward the inboard side. The fluxes have been evaluated in the FW of all the 18 blankets from bottom of the inboard to the top and down to outboard, as shown in Figure 4.21. The evaluated blankets are segmented into 4 pieces in the poloidal direction to increase the resolutions, as depicted in the Figure 4.21 comparing with other FW.

Figure 4.25 exhibits a reasonable trend of the neutron distribution in ITER blankets. Because the source neutrons are isotropic emitted, the fluxes on all the FW will be similar if the plasma volume is in a circular torus shape. As the plasma volume is elongated in the poloidal direction, the FW in the equatorial level, i.e. inboard blanket 3, 4 and outboard blanket 14, 15 shown in the Figure 4.21, have higher neutron fluxes due to the higher probability of neutron radiation. The reverse is the FW in blanket number 8 and 9 which have lower neutron fluxes. As shown in the Figure 4.25, the results have very good agreements within 2%.

The integral nuclear heating value in the TFC cells are calculated using MCNP cell-based energy deposition tallies, with material assigned. Figure 4.26 gives the comparison of heating results in the inboard TFC between of CSG and mesh models. This figure exhibits a reasonable spatial distribution of nuclear heating in the TFC. The maximum value is appear in the segment 6, which is located in the equatorial position between FW 3 and 4, as shown in the Figure 4.21. As discussed in the FW neutron flux calculation, this position has strongest neutron flux than other position in the inboard side. Strong nuclear heating is found also in the TFC segment 2. As shown in the Figure 4.27, there is a gaps of 35 mm between blanket module and the divertor. The TFC segment 2 is less shielded by blankets than other segments, hence the nuclear heating in this segment is significantly larger than segment 1 and 3. The results in Figure 4.26 exhibits that the TFC nuclear heating in the mesh model agree rather well with that of CSG model within the range of statistical error. Both comparisons between the CSG and TT mesh model show acceptable agreements for this verification benchmark. Further evaluations TT mesh models with different tolerances are interesting to carry out.

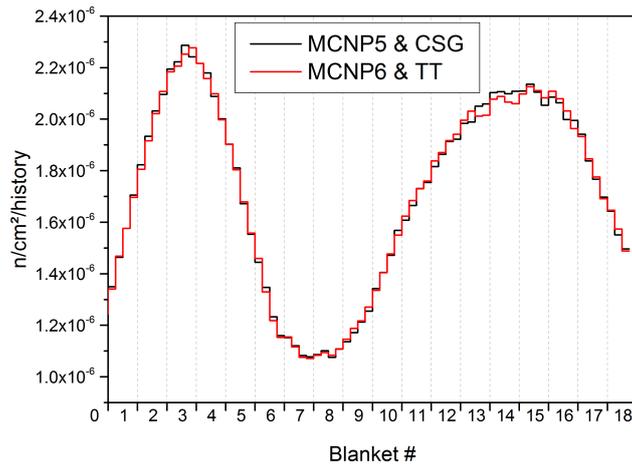


Figure 4.25: Evaluation the flux of un-collided neutrons on the FW, with an order given in Figure 4.21.

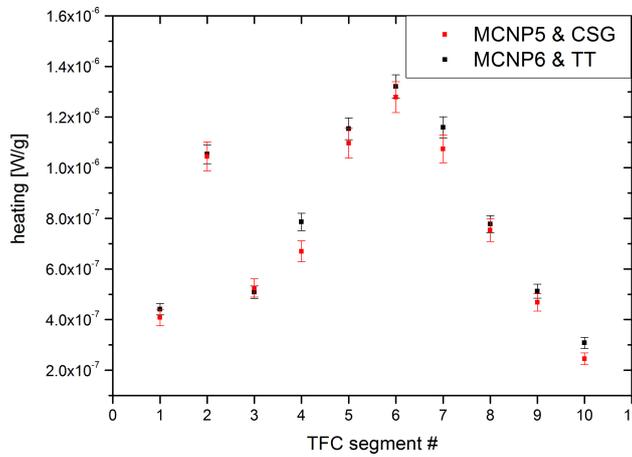


Figure 4.26: Heating in the TFC with an order given in Figure 4.21.

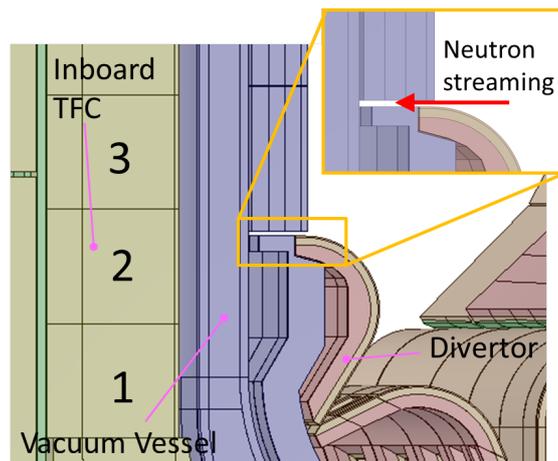


Figure 4.27: Neutron streaming through the gaps between the blanket and the divertor, which leads to higher nuclear heating in the segment 2 of the inboard TFC.

## 4.5 McMeshTran development for multi-physics coupling

McMeshTran represents a MC Mesh and data transformation, translation and transfer tool. It is the essential development in this work to enable a consistent multi-physics coupled analysis. This tool is implemented with a generic data mapping kernel, and interfaces to common-used MC, CFD and FE codes. It has been integrated in SALOME platform for providing an user-friendly GUI with data viewing, management and sharing functions. This section discusses the development of this module, and the coupling processes between MC and CFD/FE codes.

### 4.5.1 The McMeshTran module

The framework of the McMeshTran module, as shown in Figure 4.28, is similar with that of McCad, which is consisted of a server and a client. On the server side, a mesh and the associated discretized data, is represent as a *Part*. The instances of part can be put into a *Group*. The *Sever Manager*, which is based on the MED library, implements methods like mathematic operations, spatial transformations and data interpolations for managing the parts and groups. The On the client side, a *McMeshTran Object*, a *Client Manager* and a *GUI* class are implemented similarly as the McCad client, which has been discussed in Section 4.2.2. The implementation of McMeshTran modules requires handling of many technique issues. One issue is the visualization of McMeshTran mesh and data.

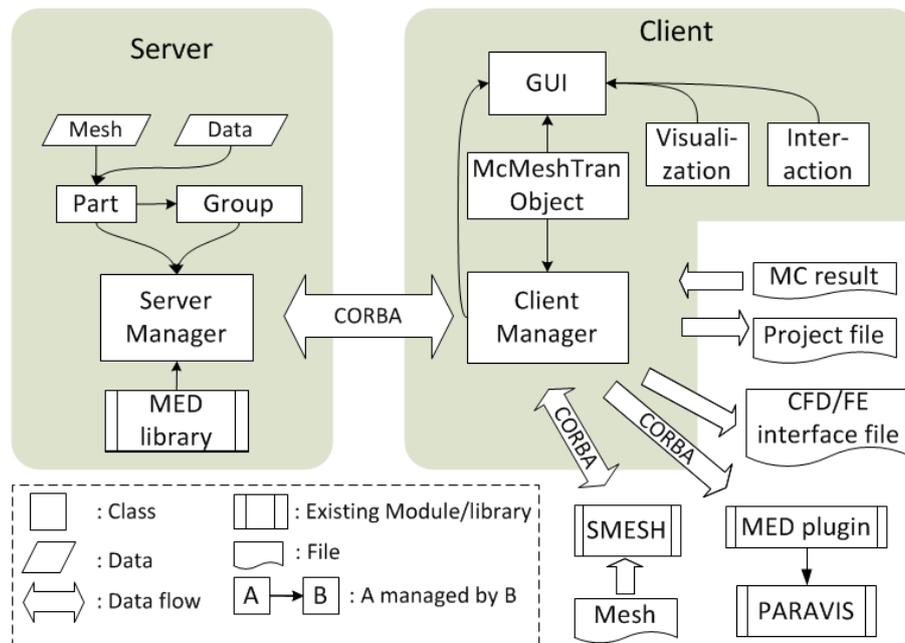


Figure 4.28: The framework of the McMeshTran module.

Two approaches are provided for visualizing mesh and data in McMeshTran. The first way is the visualization in the SALOME VTK viewer window. It requires converting meshes from the MED format to the VTK format. The node coordinates and element connectivities are retrieved for constructing a general VTK object called *vtkDataSet* containing meshes and discretized data. The mapping of a data value to a specific color is required, which is performed by creating a *vtkLookupTable* presented in the VTK viewer.

The second approach is sending the McMeshTran object to the PARAVIS module. The mesh and data are transferred through CORBA, and received by a SALOME built-in plug-in called ParaMEDCorba. This plug-in translates the mesh and data into VTK data type called *vtkMultiBlockDataSet* for representing in PARAVIS. For the development of the interface between McMeshTran and SMESH, a similar approach is used as in McCad as discussed in Section 4.2.2.

### 4.5.2 The coupling process

The coupling process implemented in McMeshTran is schematically shown in Figure 4.29. The MC neutronic results on orthogonal or unstructured meshes are post-processed by MC interfaces. In the mesh/data manipulation phase, the target CFD/FE mesh, where the data are mapped, is import through the mesh interface. And then the MC scoring mesh and target mesh are manipulated in Cartesian coordinate to maintain spatial position consistence. The data are manipulated if needed, e.g. merging results, changing unit, etc. Data mapping is performed for translating the neutronics results, i.e. nuclear heating distribution, to a target CFD/FE mesh with a data interpolation scheme, which reasonably obeys the physical conservation. At the end, the nuclear heating data on the target mesh are exported by the CFD/FE interfaces to specific interface files for CFD/FE simulations. As shown in Figure 4.1, McMeshTran provides interfaces for MC codes MCNP5/6, TRIPOLI-4 and Geant4, and for CFD code CFX, Fluent and FE platform ANSYS Workbench. These interfaces will be detail discussed in Section 4.6. This section focuses to explain the mesh and data manipulation, and the data mapping process.

#### Mesh and data manipulations

Some basic mathematic operation and spatial transformation functions are provided by McMeshTran based on the MED library. Methods provided for manipulating MC results include summation, averaging, and constant factor multiplying. The mesh configurations are assumed to be identical in the summation and averaging operations. The statistical

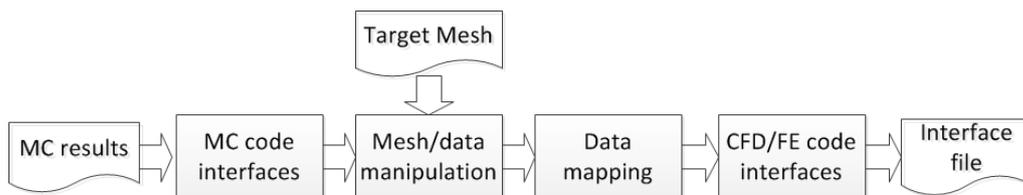


Figure 4.29: The coupling process between MC and CFD/FE simulations.

errors are not handled in these operations. Summing up two results on MC scoring meshes is needed in the case of merging nuclear heating results produced by different particles, e.g. neutron and photon. The averaging operation can be applied to merge two tally results obtained from two independent MC simulations on the same case. The average results is weighted by the number of simulated particles:

$$c_i = \frac{a_i \cdot N_a + b_i \cdot N_b}{N_a + N_b} , \quad (4.3)$$

where  $c_i$  is the average result of the mesh cell  $i$  between  $a_i$  from the first simulation and  $b_i$  from the second simulation,  $N_a$  and  $N_b$  are the number of particles simulated in the first and the second simulation. Multiplying a constant factor is needed for converting the unit and normalizing or unnormalizing the results.

In the MC simulation, the global coordinate system of the entire model is used for the scoring mesh, whereas a local coordinate system is often set up in CFD/FE analysis for a specific component. Therefore, spatial transformations are frequently conducted demanding that the MC scoring mesh is correctly positioned. Methods provided for manipulating meshes are spatial translation, rotation and scaling, which are realized by the MED library.

### Data mapping

Data mapping is a essential process for coupling neutronics with TH and SM analysis. Due to the different physical nature, a mesh configuration for scoring the nuclear heating is not necessarily suitable for a CFD or a FE analysis. High-fidelity and accurate approaches have to be employed for data mapping between general unstructured meshes. In McMeshTran, two data mapping schemes are provided based on the MED library — the volume-weighted and the point-to-point interpolation scheme. The first one is physical conservative, but the second one is not.

Physical conservative volume-weighted data mapping is the first choice in mapping nuclear heating data. The nuclear heating is represented in the form of the power density, which is a volume averaged scalar quantity as data in the element (or cell data). Using the volume-weighted approach, the value in a target mesh cell is calculated from the values of the overlapped source cells weighted by the overlapped volumes. Considering a discretized source field  $\Phi_s$ , e.g. power density, is expressed as target discretized field  $\Phi_t$  using linear remapping method

$$\Phi_t = \mathbf{W} \times \Phi_s , \quad (4.4)$$

where  $\mathbf{W}$  is called interpolation coefficient matrix. For a conservative interpolation, the preservation of integral of  $\Phi$  on any domain is desired. Therefore, any target cell  $T_i$  as shown in Figure 4.30 has to satisfy:

$$\int_{T_i} \Phi_t = \sum_{S_j \cap T_i \neq \emptyset} \int_{T_i \cap S_j} \Phi_s . \quad (4.5)$$

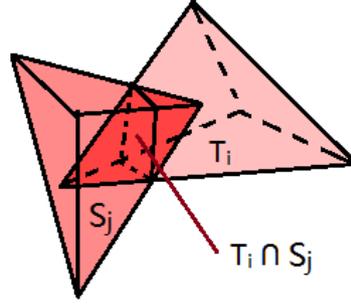


Figure 4.30: Sketch of the volume-weighted data interpolation scheme.

where  $S_j$  is a source cell, and  $i, j$  are respectively the indexes for a target and source cell. Using  $Vol()$  to indicate the volume, the left side of the Equation 4.5 becomes

$$\int_{T_i} \Phi_t = Vol(T_i) \Phi_{T_i} \quad ,$$

and the right side becomes

$$\sum_{S_j \cap T_i \neq \emptyset} \int_{T_i \cap S_j} \Phi_s = \sum_{S_j \cap T_i \neq \emptyset} Vol(T_i \cap S_j) \Phi_{S_j} \quad .$$

Because the physical value is constant within one cell, the coefficient of linear remapping from source cell  $j$  to target cell  $i$  is

$$W_{ij} = \frac{Vol(T_i \cap S_j)}{Vol(T_i)} \quad .$$

Using this interpolation scheme, the integral power in the overlap volume between source mesh and target mesh is preserved during the data transfer. The data mapped on the target mesh are still kept as cell data. The volume-weighted scheme is suitable for mapping nuclear heating data for FVM codes such as Fluent, in which the heat source is required to be located in the cells.

However, nuclear heating data have to be provided on mesh nodes in FE codes, because in FEM continuous physical quantities are discretized as nodal values [Nik04]. The same holds for the transfer of heat source data to the fluid dynamics code CFX, which uses a hybrid method of FVM and FEM. Therefore, cell data are first degenerated to point data located at the cell centroids, and then mapped to mesh nodes (or vertices) using the point-to-point mapping scheme. The value assigned to a mesh node is obtained from the nearest cell centroid, as illustrated in Figure 4.31. It is noted that this mapping approach is not physical conservative, and requires a higher mesh resolution to preserve the data fidelity.

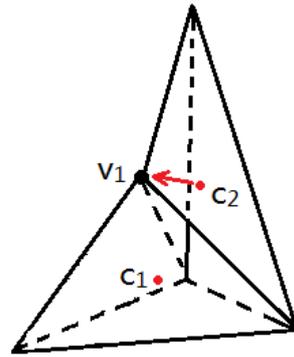


Figure 4.31: Mapping data from mesh cells to a vertice by means of the point-to-point mapping scheme. The data for node  $v_1$  is obtained from  $c_2$  due to the smaller distance than to cell  $c_1$ .

## 4.6 McMeshTran coupling interfaces

The interfaces for MC codes have been designed as general post-processing interfaces, which is not only suitable for nuclear heating results, but also other mesh tally results. Therefore, they can be used as well for general neutronic analyses as post-processing tools.

### 4.6.1 Interface for the MCNP mesh tally

Starting from version 5, the MCNP code provides a function to tally flux, heating and other responses on a 3D superimposed orthogonal mesh. In this work, a reader interface has been developed for post-processing MCNP mesh tallies results.

#### Post-processing mesh tally results

The mesh tally is processed successively in these steps:

- I Processing of the mesh tally configurations from the MCNP input file. The mesh tally output file is not fully self-describing, thus mesh configurations are fetched from the input file and matched with results in the mesh tally file. Mesh coordinates and also other parameters are used to construct the mesh.
- II Processing results from the mesh tally output file, and associating them with the corresponding meshes. This interface is adaptive to all types of ordering of mesh nodes. If mesh tally results are grouped into several energy intervals, the total results summing over all energy intervals will be kept. The statistical errors are processed and restored as well.
- III If a mesh tally is obtained on the cylindrical coordinates, the mesh will be transformed into the global Cartesian coordinates. A special process has to be applied in case of a cylindrical mesh tally, which will be discussed later.

IV Conversion of the orthogonal meshes into unstructured meshes by constructing the element connectivities. The obtained results are associated with the new meshes as well.

Special attention has to be paid for meshes in cylindrical coordinates. These meshes are transformed into global Cartesian coordinates using two steps, as shown in Figure 4.32. The first step is conversion of mesh from the cylindrical coordinates  $R-\varphi-Z'$  to the local  $X'-Y'-Z'$  Cartesian coordinates. The orthogonal cylindrical mesh elements are converted into first-order hexahedra, so that the cylindrical mesh is converted to a structured mesh in a Cartesian coordinate system. Although the curvatures are partly lost after this conversion, the differences are negligibly small, if a sufficiently fine mesh resolution is chosen. At the end these nodes are transformed into the global  $X-Y-Z$  Cartesian coordinates. Another issue is caused by the mandatory rules in MCNP code that the last azimuthal angle should be  $360^\circ$ , and the first radius should be 0. Usually useless elements are created for the first radial interval and the last azimuthal interval. These useless elements are detected and removed automatically by the interface.

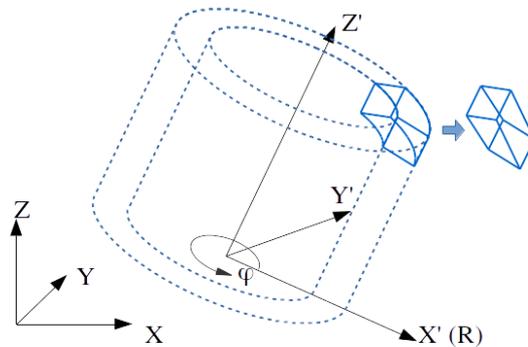


Figure 4.32: Mesh transformation from cylindrical coordinates to global Cartesian coordinates. The mesh is firstly transformed from the  $R-\varphi-Z'$  to the local  $X'-Y'-Z'$  coordinates, and then transformed to global  $X-Y-Z$  coordinates. The cylindrical mesh elements are approximated by using first-order hexahedra.

### Processing of the mesh tally results for coupling analysis

In a superimposed mesh tally, sometimes a mesh cell cannot avoid crossing two or more MCNP geometry cells with different materials. Processing mesh tally results of nuclear heating for CFD/FE analysis require in this view an appropriate handling.

Figure 4.33(a) illustrates a mesh tally cell crossing two geometry cells with materials  $A$  and  $B$ .  $C_A$  and  $C_B$  are unstructured mesh elements which build the geometry cell  $A$  and  $B$  in a CFD/FE mesh. They acquire a nuclear heating value from this mesh tally cell. The neutron or photon flux density  $\bar{\phi}$  calculated in MCNP is averaged over the mesh tally cell, and also the flux values of different materials ( $\bar{\phi}_A$  and  $\bar{\phi}_B$ ) cannot be provided by MCNP separately. To calculate the nuclear heating, two options are provided by MCNP. The first option is to calculate the heating in this mesh tally cell with a mixed material:

$$\dot{q}_{Mix} = \bar{\phi} \cdot \Sigma_{Mix} \quad ,$$

where  $\Sigma_{Mix}$  is the macro heating cross-section of the mixed material. However, this heating value does not represent the actual heating of either  $A$  or  $B$ . For example, if material  $B$  is air,  $\dot{q}_{Mix}$  will largely underestimate the heating in material  $A$ , and heating values mapped to target mesh cell  $C_A$  and  $C_B$  are not accurate. Another option is calculating the heating in  $A$  and  $B$  separately, and each time assuming the whole mesh tally cell is filled with one of the material, as shown in Figure 4.33(b) and Figure 4.33(c). Assuming  $\bar{\phi}_A \approx \bar{\phi}_B$  (usually called *flat flux* assumption), the  $\bar{\phi}$  is a good approximation of both  $\bar{\phi}_A$  and  $\bar{\phi}_B$ . Therefore, the heating power density values in two materials of this cell are

$$\begin{aligned}\dot{q}_A &= \bar{\phi} \cdot \Sigma_A \quad , \\ \dot{q}_B &= \bar{\phi} \cdot \Sigma_B \quad ,\end{aligned}$$

where  $\dot{q}_A$  and  $\dot{q}_B$  are the heating values, and  $\Sigma_A$  and  $\Sigma_B$  are macro cross-sections of material  $A$  and  $B$ . The heating results on these two separate mesh tallies are mapped into the unstructured meshes of  $A$  and  $B$ . Thus the values mapped to  $C_A$  and  $C_B$  are more accurate. This is based on the flat flux assumption, which is often reasonable if a flux gradient on the boundary is not remarkably large, or if the mesh tally cell is small enough to disregard the flux difference.

The second option is preferred for processing a MCNP mesh tally of nuclear heating to be later used in the coupled analysis. In region of interest, mesh tallies of neutron and gamma heating are thereby obtained separately in each materials, and subsequently mapped separately into unstructured meshes of components, according to their material composition.

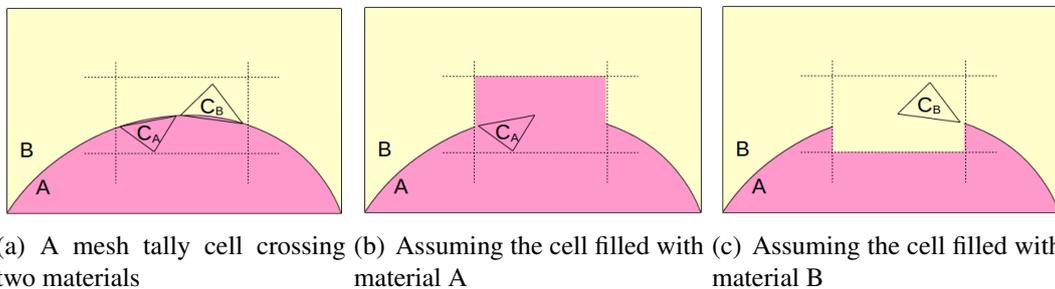


Figure 4.33: Processing of MCNP mesh tallies of nuclear heating crossing two materials.

## 4.6.2 Interface for MCNP6 results on unstructured mesh

The MCNP6 results on the unstructured mesh are written into a EEOUT (Elemental edit output) file in MCNP6. It is a self-describing file which contains all the mesh and data in one file. The interface developed in this work is able to process this output file, reconstruct the mesh and process the data.

All data in this file are organized into 20 data sets (or blocks). Each block contains three segments — the identification line, the title line and the data. Except for the first line, a

block contains at least two and at most three of these segments. The identification line contains six integer values indicating how the title and data are given, e.g. number, size and lines of data. The title line is optional, but it is given when the block has no data. The data line contains the exact number of data as indicated by the identification line. Details on the file structure can be found in [Mar12].

The interface processes the EEOU file based on its file structure. It imports each time a block and check the block type. If the block contains data required by the post-process, then the corresponding subroutine is called for processing it, otherwise this block is simply skipped and next one is imported. The required blocks include calculation parameters, instance information, node coordinates, element connectivities and tally results. The unnecessary blocks include nearest neighbors of elements, materials, cell centroids and volumes. This EEOU file interface has been designed to be adapted to most of the changes in the file, including particle types, element types, energy and time bins, multiplier settings, etc. In the output file, the results for all instances are mixed together. To facilitate the analysis, the results in each instance are separated by the interface.

Unlike any orthogonal mesh tally, the unstructured mesh tally can be adaptive to geometry and physical field, thus it has intrinsic advantages for the coupling analysis. In addition, a unstructured mesh is usually assigned with one homogeneous material, and materials are not be mixed in one element. Therefore, high-fidelity nuclear heating can be calculated using the unstructured mesh tally for multi-physics coupling analyses.

### 4.6.3 Interface for the TRIPOLI-4 mesh tally

Since the TRIPOLI-4 code has been already integrated into the SALOME platform, a full support of this code can further improve the integrity of the coupling system. The TRIPOLI-4 code provides superimposed mesh tally functions for orthogonal Cartesian, cylindrical and spherical meshes. A mesh has a uniform interval, and is constructed by setting a lower point, an upper point and associated intervals for each coordinates. Two types of scores, track length and collision scores are able to be performed by TRIPOLI-4, as well as the responses based on these two types of scores.

The calculation parameters and results are all provided in the TRIPOLI-4 output file, thus only one file has to be processed by the McMeshTran TRIPOLI-4 interface. The parameters of the scores, including associated responses, energy bins, etc. are kept to support the result processing. Because meshes in spherical coordinates are seldom used, only Cartesian and cylindrical mesh tallies are supported by our interface. The orthogonal meshes are converted into unstructured meshes, and finally stored in the MED library. They are processed for the coupling analysis in a similar way as MCNP mesh tally, discussed already in Section 4.6.1.

## 4.6.4 Interface for Fluent and CFX

The nuclear heating processed from the MC outputs is used as the volumetric heat source for the CFD and the FE analyses. In the previous sections the MC output post-processing and nuclear heating data mapping have been discussed. This section discusses how to export the heating data to the Fluent and the CFX code.

### Interface files

In the Fluent code, an API called *User Defined Function (UDF)* is provided for the runtime control of the simulations. The UDF file is written in C language and linked with a Fluent header file for accessing the predefined variables and functions. User-defined source terms such as mass, momentum and energy can be specified in the predefined functions. Take the energy source term as example, the predefined function called *DEFINE\_SOURCE* can be implemented by a Fluent user to provide an associated heat source value for a given element. In the McMeshTran Fluent interface, a UDF file is generated providing this function. The heating data in the UDF file is given as an array associated with the centroids of elements. The UDF files have to be compiled by Fluent code to generate a shared library, which has to be linked dynamically during the simulations. During the runtime of the simulation, the index of a element is provided by Fluent for locating the heat source value. The *DEFINE\_SOURCE* function calculates first the element centroid, and search in the whole nuclear heating array for the corresponding value.

A similar approach is adopted for the CFX interface. A CFX API file called *User Fortran*, which is a Fortran-77 source file, is generated by the interface. A CFX header file has to be linked, and the subroutine *SUBROUTINE\_USER\_SOURCE* has to be implemented to return heat source values for each element. In order to prevent the data to be initialized each time when calling the subroutine, the heating data array is defined using the *DATA* statement, so that the array is initialized in the compilation phase. Because the CFX code is based on a hybrid method of FVM and FEM, the heat source value has to be provided for the element nodes. The User Fortran subroutine has been implemented in a similar way as the Fluent UDF, and has to be compiled by the CFX preprocessing software CFX-Pre before use.

### An accelerated data search algorithm

In the Fluent and CFX interface files, the predefined functions have to search the heating value for given cells or nodes. When a mesh contains millions of cells and nodes, the data searching becomes a critical issue aspect regarding computation time. Therefore, an efficient data searching algorithm has to be implemented for a point cloud.

Even though the same mesh is used, cell centroids provided by CFD codes might not be exactly identical with those calculated by McMeshTran, due to the precision problem. Therefore, the data search cannot rely on exact matching on the centroid coordinates. Search algorithms such as the Hash table algorithm are not applicable due to this reason.

Here, a voxelization data search algorithm has been proposed, as illustrated in Figure 4.34. It consists of two phases—data preparation phase and data search phase. In the first phase, the boundary box of the point cloud is calculated and divided into suitable number of voxels using the dimension of the boundary box in each direction.

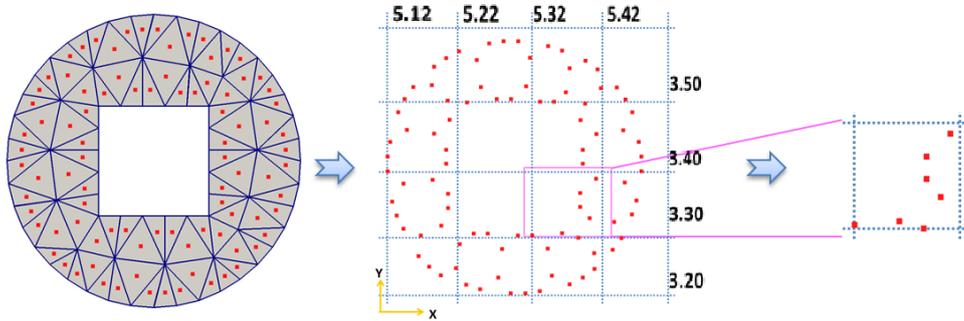


Figure 4.34: Voxelization data search algorithm for CFD interfaces. The point-wise data are grouped into 3D voxels. To search a point, at first the voxel containing it is located.

The sequence illustrated in Figure 4.34 reads to:

1. The range of the boundary box in X coordinate is [5.12, 5.48]. Hence one obtains  $\Delta X = X_{max} - X_{min} = 0.36$ ;
2. Multiplying  $\Delta X$  with a suitable *Amplifier* that is a 10-based exponential figure  $10^n$ , where  $n$  is an integer. Here we choose  $10^1$  for example, we have  $\Delta X \times 10 = 3.6$ ;
3. Rounding it into the upper integer, that is  $3.6 \approx 4$ , and the range is subdivided to 4 sub-ranges along the X coordinate — 5.12, 5.22, 5.32, 5.42, 5.52, where the step interval is the reciprocal of the amplifier, i.e. 0.1.
4. These steps are repeated for the Y and Z coordinates, choosing a suitable  $n$  for each coordinate. Thereby, 3D voxels are constructed. Here  $I, J, K$  are used to indicate the number of sub-ranges in X, Y, Z coordinates;
5. Sorting the points into the voxels. The sorting method is actually similar with the finding method, which is discussed later.

After the preparation step, the data are exported into the interface file, i.e. UDF of Fluent and User Fortran file of CFX, with three arrays as shown in Figure 4.35. The heating data are given in a big 1D array *Array-A* consisting of the X, Y, Z coordinates and the heating values of a list of points. In addition, the information of the 3D voxels is stored as two 1D arrays, which offer one data for each voxel. The value in *Array-B* is the index of the first point in the *Array-A* belonging to the current voxel. A value in the *Array-C* is the number of points contained in current voxel. These three arrays are ordered by the order of voxels. With *Array-B* and *Array-C*, the points in a voxel can be located in the *Array-A* according to the voxel indexes. Information of boundary box, amplifier, etc. are included in the interface file as well.

#### 4 A multi-physics coupling system

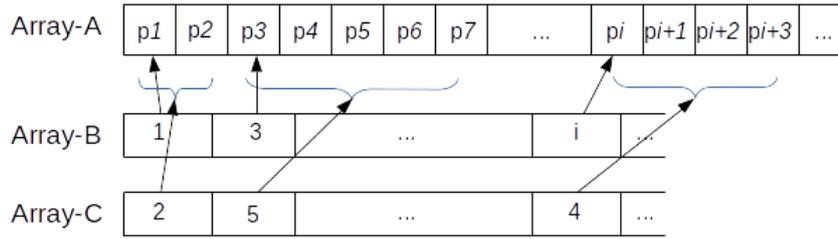


Figure 4.35: Data structure and indexing of nuclear heating data in the CFD interface file. *Array-A* contains coordinates and heating values of all points, *Array-B* contains first index of a voxel in *Array-A*, and *Array-C* contains number of points inside the voxel.

The data search phase is performed during the CFD calculations. When looking for the heating value for a point, e.g. a point with  $X = 5.38$ , the search is executed as follow:

1.  $X - X_{min} = 5.38 - 5.12 = 0.26$ ;
2.  $(X - X_{min}) \times amplifier = 0.26 \times 10 = 2.6$ ;
3. Rounding it to the smaller integer:  $2.6 \approx 2$ , then we find the voxel index for X coordinate that is 2.
4. Calculation of the voxel indexes for the Y and Z coordinate. Using  $i, j, k$  to indicate the voxel index for the X, Y, Z coordinate. By this the position of the first point of this voxel can be found in *Array-B* with the index of  $k \times J \times I + j \times I + i$ , and the number of points contained in this voxel can also be found in *Array-C* with the same index.
5. Using binary or other search algorithms to match the exact point in this voxel.

Regarding the data search, the time complexity of the voxel searching is of  $O(1)$ , which means that the search time does not change with the number of points. The speed-up of this algorithm depends on the number of voxels. If more voxels are used in the search, more speed-ups can be achieved, whereas the memory consumption increases as well. To reach a balance, the number of intervals for each coordinate are suggested to be located between 10 and 100. The distribution of points cloud influences the search speed of this method too. In worse cases, where mesh cells are heterogeneously distributed in one or two voxels, the acceleration effect will not be obvious. In this case, further improvements such as nested voxels have to be introduced.

It should be noted that, in case of the use of single precision float-point in the CFD simulation, cell centroids calculated from CFD codes are not exactly identical as those in the interface file. Other than using the *equal* judgment, a tolerance of  $10^{-6}$  mm is used to prevent search failures. If still failed, the tolerance is enlarged to  $1^{-5}$  mm. Furthermore, in order to prevent a possible failure caused by a point located exactly on the voxel boundaries, a small margin is kept in each voxel, which overlaps with the neighboring

voxels. In any case of failures, the value from last search, which is usually the value of the neighboring cell, is used for the current cell.

### 4.6.5 Interface for ANSYS Workbench

ANSYS Workbench provides a module for importing external data for various purposes. The *Comma-separated Value (CSV)* format is an acceptable format, which is a raw but widely used format. Data in each line are separated by identical number of commas, or other delimiters such as tabs, semicolons. However, it is not able to describe an unstructured mesh. Hence, the data on an unstructured mesh are associated with coordinates of element centroids or mesh nodes. Because FE codes require data to be located on nodes, this CSV format is adequate for FE codes. An interface has been developed for exporting data from unstructured meshes to CSV files. This interface file is applicable for other FEM based software as well, such as COMSOL Multiphysics, Nastran, etc.

## 4.7 Verifications of coupling interfaces

### 4.7.1 Verification of the data translation

The TBM test case model has been adopted in this calculation. The CSG model is adopted, and neutron and photon coupled transport simulations are performed with the MCNP5-1.60 code using the FENDL-3.0 cross-section library. The calculation has been set-up similar to that in Section 4.4.1. Because the calculation conditions are simplified, the results are only suitable for verifying the codes.

The heating data after interpolation has to agree with the MCNP results. Therefore, the integral heating values of the interpolated data have been compared with MCNP directly calculated results. The MCNP results are obtained using the cell-based heating tally, and the integral heating results of the interpolated data are calculated using ParaView. However, the heating distribution of the interpolated data has to be checked as well, which is much more difficult because of the difference between the MC tally mesh and the interpolation mesh. Here, a so-called *Inversion Check* approach has been proposed. The BU highlighted in Figure 4.15 has been checked via the following steps:

1. Tallying the neutron and photon (gamma) heating in the beryllium, CP, OSI, and mixed material with the same mesh configuration covering this BU. The mesh tally setup for obtaining nuclear heating has been discussed in Section 4.6.1. Since a BU contains only three sub-components made of material beryllium, Eurofer and OSI, these sub-components are named here the same as their material. ;
2. Summing the neutron and photon heating results in beryllium, Eurofer and OSI, and interpolating them into three target meshes of these three subcomponents;
3. Merging the three meshes into one mesh, and re-interpolating the data on this mesh into the MC tally mesh, and
4. Comparing with heating mesh tally results on the mixed material. Since the mixed material tally is directly obtained from MCNP, which is considered here as the reference.

To capture the geometry feature on the curved surfaces, a deliberately configured mesh tally with a resolution of  $262 \times 167 \times 98$  intervals (in X, Y, Z coordinates) has been assigned to the BU, as shown in Figure 4.36. After the MC transport calculations, most of the mesh cells have statistical errors below 5% (a recommend level given in Section 2.2.4). Three target meshes for beryllium, Eurofer and OSI has been generated using ANSYS Workbench, with totally  $3.20 \times 10^6$ ,  $1.14 \times 10^6$ ,  $2.41 \times 10^6$  elements.

#### Results and analyses

First, the integral heating of the interpolated meshes are compared with the values obtained from cell-based nuclear heating tallies of MCNP (using the F6 card), as shown in Table 4.4. The statistical errors for these cell-based tallies are less than 0.04%, which are negligible small. The deviation is calculated with the expression  $|P_{MC} - P_{Int}| / P_{MC}$ , where  $P_{MC}$  represents the MCNP tally results,  $P_{Int}$  represents the interpolated results.

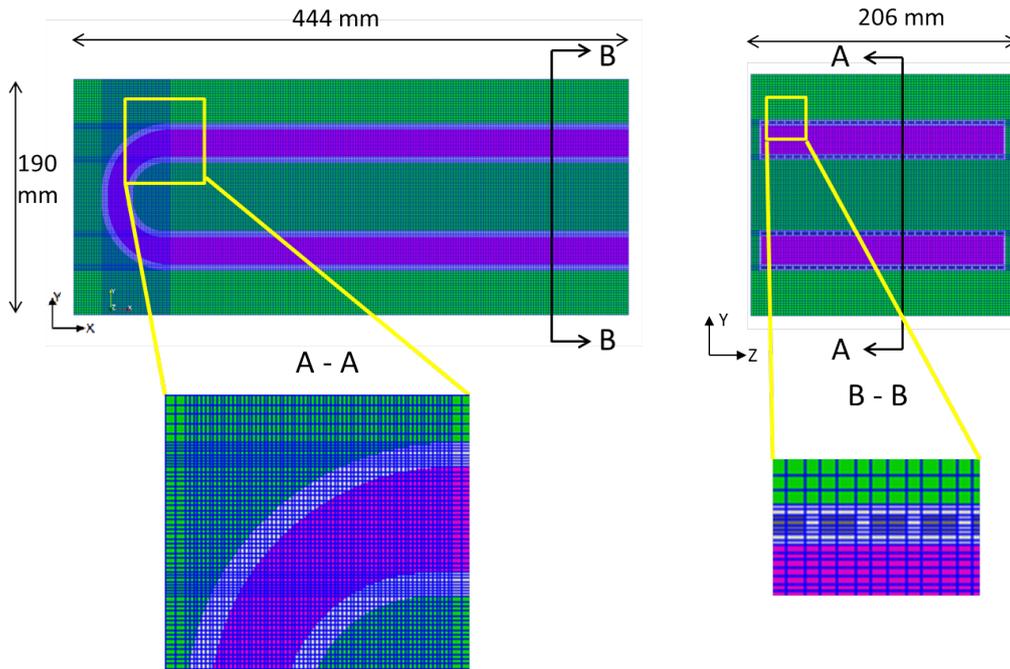


Figure 4.36: Mesh tally cross-sectional view in Z direction (left) and X direction (right)

Table 4.4: Comparison on the integral heating results for a breeder unit using the boundary conditions specified in Section 4.4.1.

Subcomponent	MCNP tally result (W)	Integrated result (W)	Deviation
Beryllium	$1.5555 \times 10^4$	$1.5597 \times 10^4$	0.27%
Eurofer	$1.8036 \times 10^3$	$1.8462 \times 10^3$	2.36%
OSI	$1.0862 \times 10^4$	$1.0882 \times 10^4$	0.18%

The deviations are very small in the beryllium and OSI. A larger deviation in Eurofer is observed, which seems reasonable due to the geometry complexity.

The inversion check is performed as discussed. Figure 4.37 compares two cross-section views at the center plane of the BU. The deviation of the inverted heating values from mix-material tally results are calculated in each cell, using the mix-material tally results as references. Because the statistical errors of both tallies are evaluated with the recommend value of 5%, 10% of deviation between the results of the two tallies can be expected. Therefore, the cells with deviations  $<5\%$  are considered to be acceptable here. The cells with deviations  $>5\%$  are analyzed by categorizing them into five ranges —Range-1 (5%, 10%], Range-2 (10%, 100%), Range-3 [100%], Range-4 (100%,  $\infty$ ), where the square brackets “[” and “]” mean including the bound, and the parentheses “(” and “)” not. These cells are shown in Figure 4.38, and the ratios of number of cells in these ranges to the total number of cells in the MC tally mesh are given in Table 4.5. The cells located in the flow channel have no nuclear heating. Therefore, these cells are ignored in this discussion. Other cells with deviation are located mostly around the flow channel, and they are found

#### 4 A multi-physics coupling system

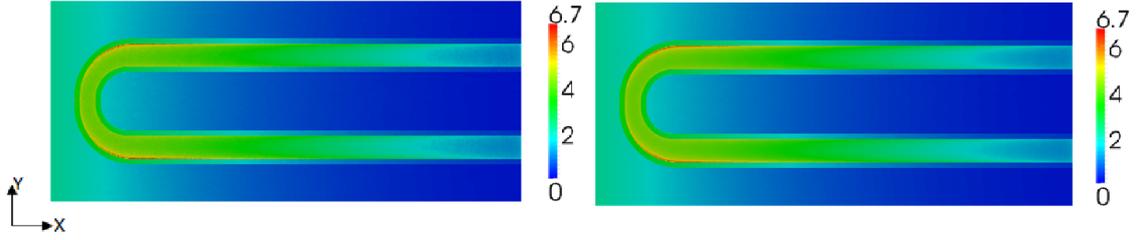


Figure 4.37: Cross-section view of the MC material-mixed heating result (left) and the inverted heating field (right) for a breeder unit using the boundary conditions specified in Section 4.4.1 (unit:  $MW/m^3$ )

Table 4.5: The cells with relative large heating deviation are grouped into four ranges. The ratios of these cells to the total number of MC tally mesh cells are given.

	Range-1	Range-2	Range-3	Range-4
Ratio(%)	0.49	0.44	0.01	0.06

to be influenced significantly by the void space during the re-interpolation step.

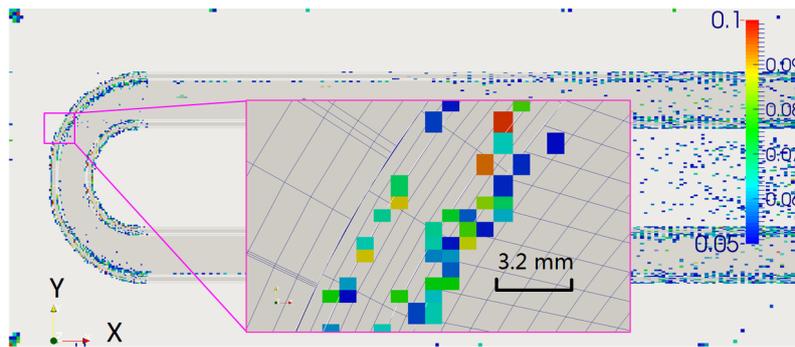
To explain more detail, those cells containing both void and solid are schematically demonstrated in Figure 4.39. Assuming that the flux has a small gradient in Cell-1, and the statistical error is negligibly small, the heating value of the material-mixed tally  $P_{Mix}$  and heating of the mesh tally on Eurofer (material for the CP)  $P_{Eu}$  have  $P_{Mix} = \lambda_1 P_{Eu}$ , where  $\lambda_1$  is the solid volume fraction for Cell-1 in MCNP calculations. Similarly, the heating of the re-interpolated (or inverted) result  $P_{Re}$  for Cell-1 and interpolated result  $P_{Int}$  for Cell-2 have  $P_{Re} = \lambda_2 P_{Int}$ , where  $\lambda_2$  is the overlapped solid volume fraction between Cell-1 and Cell-2. The differences shown in Table 4.5 and Figure 4.38 can be expressed as follow:

$$D = \frac{|P_{Mix} - P_{Re}|}{P_{Mix}} = \left| 1 - \frac{P_{Int} \lambda_2}{P_{Eu} \lambda_1} \right| = |1 - \eta_P \eta_\lambda| ,$$

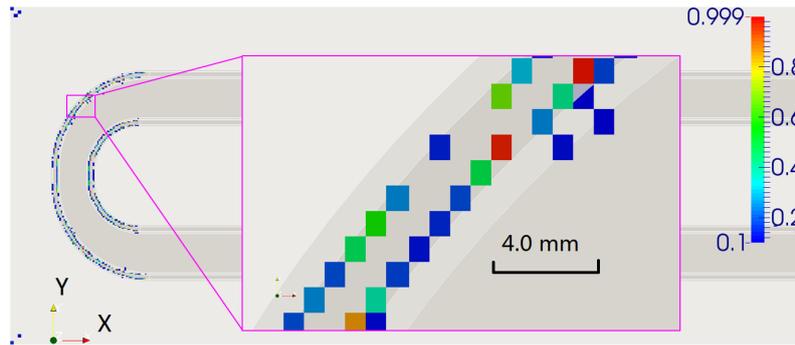
where  $\eta_P = P_{Int}/P_{Eu}$  and  $\eta_\lambda = \lambda_2/\lambda_1$ .

Due to the geometry difference between the MC model and the unstructured mesh,  $\lambda_1$  is not exactly identical with  $\lambda_2$ . When Cell-1 moves from Position-1 to Position-2,  $\eta_P$  remains almost unchanged. However,  $\eta_\lambda$  varies significantly:

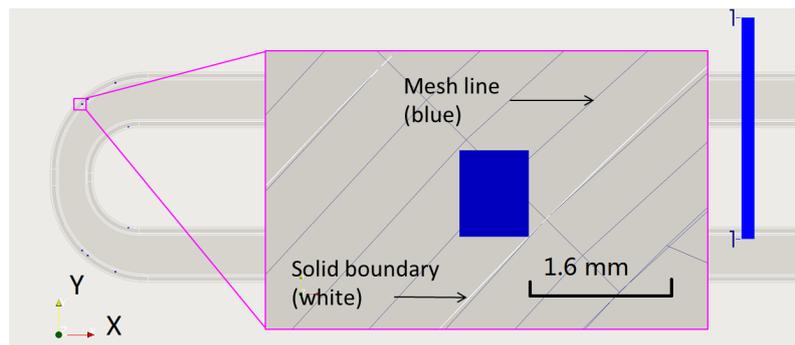
1. In case of  $\lambda_1 > \lambda_2$  (Figure 4.39(a)),  $\eta_\lambda \rightarrow 0$  and  $D \rightarrow 1$ . In such an extreme case as  $\lambda_2 = 0$ ,  $\eta_\lambda = 0$  and  $D = 1$  (as shown in Figure 4.38(c) ).
2. In case of  $\lambda_1 < \lambda_2$  (Figure 4.39(b) ),  $\eta_\lambda \rightarrow \infty$  and thus  $D > 1$  is possible (as shown in Figure 4.38(d) ). In an extreme case such as  $\lambda_1 = 0$ ,  $\eta_\lambda = \infty$  and  $D = \infty$ . Furthermore, when  $\lambda_2 = 0$  (Cell-1 is located totally in void), also  $D = \infty$  .



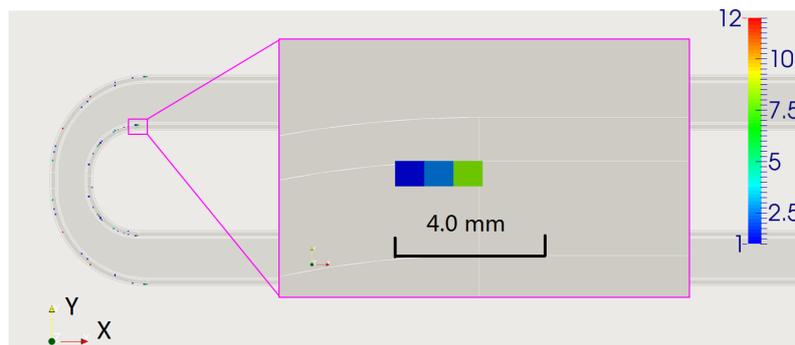
(a) Range-1



(b) Range-2



(c) Range-3



(d) Range-4

Figure 4.38: Cells with high differences in comparing of inverted heating and mixed material heating results.

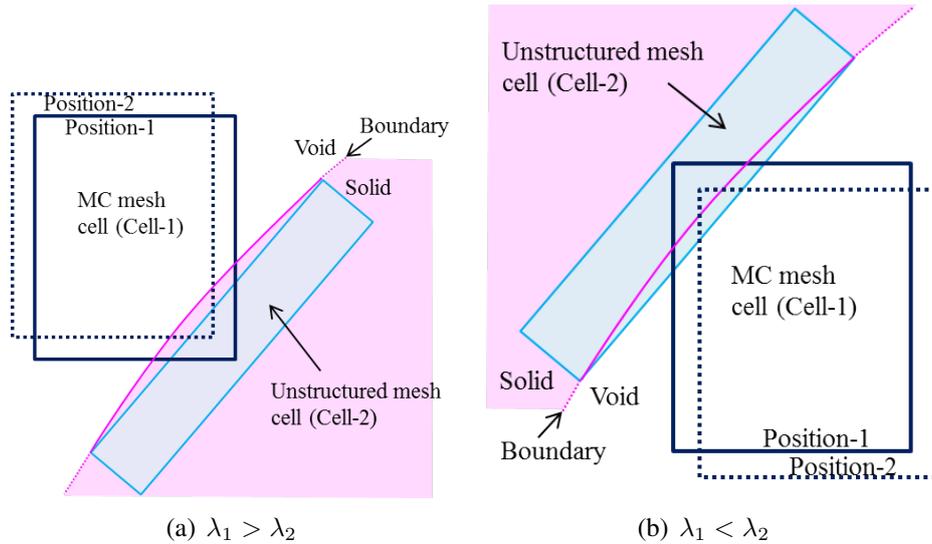


Figure 4.39: Schematic view of cells containing both void and solid volume.

3. In case of  $\lambda_1 = \lambda_2$ , which happens in cells with a pure material,  $D = |1 - \eta_P|$  directly reflects the difference between  $P_{Eu}$  and  $P_{Int}$ . The correctness of these data are preserved during the whole inversion process.

Because  $\eta_\lambda$  is difficult to be determined due to the difficulties in calculating overlapping solid volumes, the contribution of  $\eta_P$  for differences in Figure 4.38(a) and Figure 4.38(b) are complicated to evaluate. And the influences of statistical errors need to be taken into account in Figure 4.38(a), since  $P_{Mix} = P_{Eu} \cdot \lambda_1$  is not necessary valid when statistical errors are considerably large. Also it should be noted that material interfaces have the same issue as the solid-void interface, although the influence is not so strong in this case.

Nevertheless, there are only  $\sim 1\%$  of cells which have differences larger than 5%, which is reasonably small for such a rigorous checking. Besides, target CFD mesh cells cover always one material, thus the significant influence by  $\lambda_2$  would be excluded. Therefore, the MCNP5 mesh tally interface and the data translation can be conceived sufficient in the view of providing accurate heating values.

## 4.7.2 Verification of the hybrid geometry

The approach of hybrid CSG and mesh geometry modeling has been verified using the TBM test case, by replacing the BU (highlighted in Figure 4.15) in the CSG neutronic model with unstructured meshes. Well-configured hexahedral meshes for the three subcomponents — beryllium, CP, and OSI have been generated by ANSYS ICEM for nuclear heating scoring. The CAD and mesh geometries have been converted by McCad. MCNP6-1.0 code and FENDL-3.0 cross-section data are employed for neutron transport calculations, with the similar calculation set-up as Section 4.4.1. The heating results

processed by McMeshTran are shown in Figure 4.40, and integrated volumetric heating data are compared with MCNP5 cell-based integral heating tallies in Table 4.6.

Table 4.6 shows that the integrated heating results of MCNP6 unstructured meshes agree with the MCNP5 cell based tallies for all three sub-components in the BU. The heating distributions in Figure 4.40(a) and Figure 4.40(b) are distributed smoothly. However, it is found that heating in Figure 4.40(c) are not smoothly distributed at the indicated positions. By carefully checking on the mesh of the lateral plate, a tiny mesh overlap has been discovered. Although MCNP6-1.0 has been implemented with overlap detection and treatment, it fails in treating such small overlapping ( $\sim 3 \times 10^{-5}$  mm), which causes particles from the red region in Figure 4.40(c) to travel blindly through the yellow region. By removing such overlaps, the heating for the CP in Figure 4.40(d) becomes correct. Therefore, great caution should be paid to the mesh overlapping, and improvements on the overlap checking are suggested for MCNP6-1.0.

Table 4.6: Integral heating comparison between the MCNP6 unstructured mesh results and MCNP5 cell based tally results.

Subcomponent	MCNP tally result (W)	MCNP UM result (W)	Deviation
Beryllium	$1.5555 \times 10^4$	$1.5787 \times 10^4$	1.49%
CP	$1.8036 \times 10^3$	$1.7596 \times 10^3$	2.44%
OSI	$1.0862 \times 10^4$	$1.0821 \times 10^4$	0.38%

### 4.7.3 Verification of the TRIPOLI interface

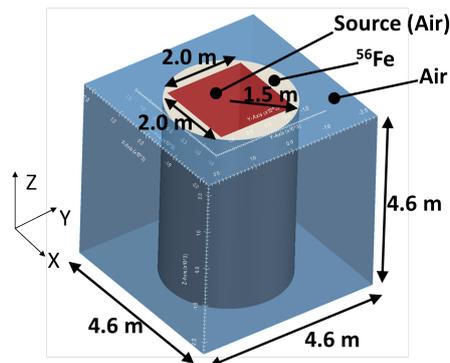
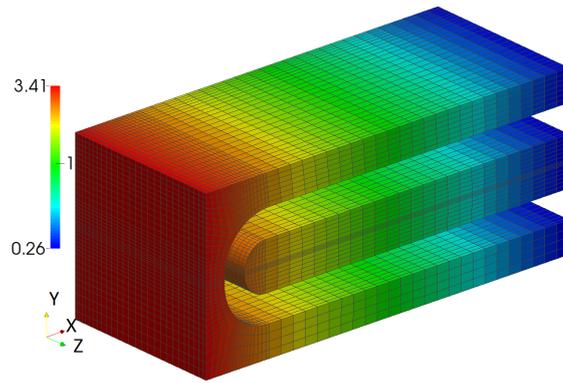


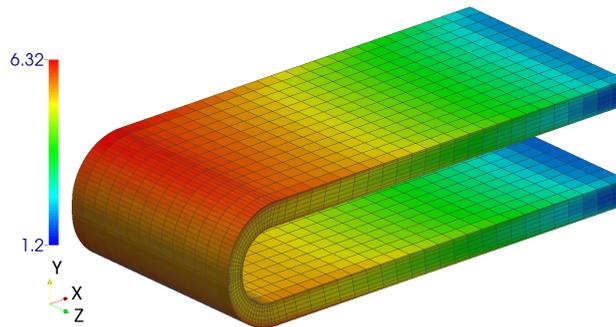
Figure 4.41: Geometry of the iron tube model

A model has been built for testing the TRIPOLI mesh tally interface, as illustrated in Figure 4.41. This tube is 4.6 m in length, and has an outer radius of 1.5 m and an inner cross section of  $1.0 \times 1.0$  m. It is surrounded by air with a dimension of 4.6 m in each direction. Inside the tube is empty void, where the neutrons are generated.

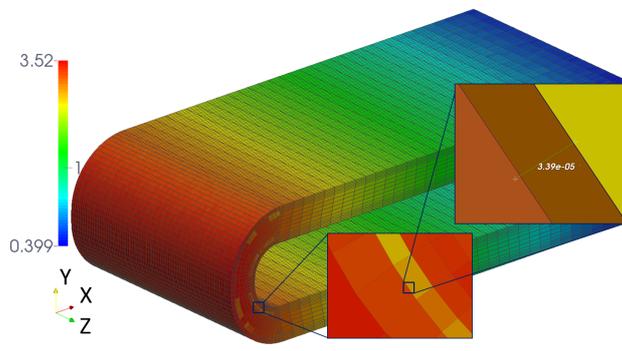
#### 4 A multi-physics coupling system



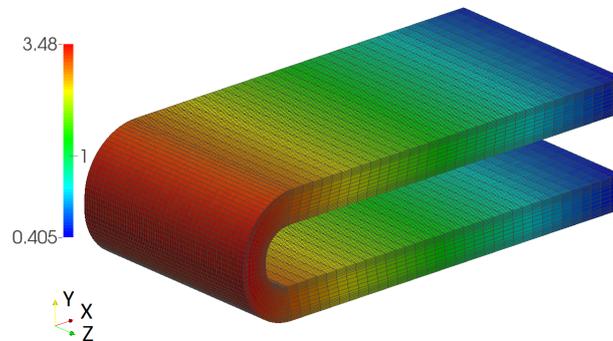
(a) Beryllium



(b) OSI



(c) CP with tiny overlaps



(d) CP without overlapping

Figure 4.40: Computed volumetric heating ( $W/cm^3$ ) of Beryllium, CP and OSI of a BU using the boundary conditions specified in Section 4.4.1 . Relative errors for all elements are below 3%.

The material of the tube is simply the isotope  $^{56}\text{Fe}$ , with a density of  $7.8 \text{ g/cm}^3$ . The air consists of  $^{16}\text{O}$  and  $^{14}\text{N}$  isotopes, with a density of  $1.29 \text{ mg/cm}^3$ . The void is also filled with air. The neutrons are generated homogeneously, constantly and isotropically in the center void volume with energy of  $14.07 \text{ MeV}$ . The space outside the model is defined as vacuum, thus neutrons flying out are terminated. The irregular shape of the tube is useful for testing the tally results of orthogonal and unstructured meshes.

Because the way of processing nuclear heating mesh tally from TRIPOLI output file is identical with that of flux, the neutron flux has been calculated which has a heterogeneous distribution. Since the MCNP mesh tally interface has been well verified in Section 4.7.1, MCNP results are taken as reference. The data library used in TRIPOLI is CEAV5.1.1 [BDD<sup>+</sup>13], which is based on JEFF-3.1.1 [SBB<sup>+</sup>09]. The JEFF-3.1.1 library is also used in MCNP5 to keep consistency. Both Cartesian and cylindrical mesh tallies are used. The Cartesian mesh has  $50 \times 50 \times 50$  intervals in X, Y, Z coordinates, covering the whole model; the cylindrical mesh has  $60 \times 50 \times 50$  intervals in R,  $\theta$ , Z coordinates, ranging from 80 to 200 cm in R, from 0 to  $\pi/2$  in  $\theta$  and from -230 to 0 cm in Z direction. In total  $10^6$  source neutrons are simulated. In TRIPOLI they are divided into 100 batches with  $10^4$  neutrons per batch.

The computed results are shown in Figures 4.42 for both the MCNP and the TRIPOLI calculation. Since TRIPOLI codes is well validated and verified for fusion applications [BDD<sup>+</sup>13], the purpose of this test is not to verify TRIPOLI again, but for verifying the TRIPOLI mesh tally interface in McMeshTran. From Figure 4.42 it can be found that, the TRIPOLI results are in general consistent with MCNP results on the neutron flux values and distribution pattern. This agreement is sufficient to conceive the TRIPOLI mesh tally interface as reliable.

#### 4.7.4 Verification of the CFD interfaces

In order to verify the McMeshTran on its multi-physics coupling capabilities, and also to conduct a cross-check on the interfaces for CFD codes, a CFD simulation for the FW has been carried out using both the CFX and Fluent code. As suggested in [KB12], the 1/6 FW model has been used for the analysis, which is shown in Figure 4.43.

The volumetric heat source has been calculated using MCNP5-1.6 and FENDL3.0 library on the CSG model. A mesh tally applied on material Eurofer steel with cell size of  $5 \times 5 \times 5 \text{ mm}$  is applied, and then is post-processed and interpolated into the mesh of the FW using McMeshTran, as shown in Figure 4.44. The unstructured meshes generated using the ANSYS Workbench are shown in Figure 4.45, with around  $7.85 \times 10^5$  elements in the fluid domain and  $7.88 \times 10^5$  elements in the solid domain. The nuclear heating data is only located in the mesh of the solid domain, since the helium gas almost produce no nuclear heating. The data on cells are remapped into nodes using the point-to-point scheme for CFX.

The plasma-side surface of the FW is subjected to a  $500 \text{ kW/m}^2$  heat flux in radial direction (X direction in Figure 4.43), which decreases to  $0 \text{ kW/m}^2$  along the bending portion. The BU-side is assigned with a uniform heat flux of  $83 \text{ kW/m}^2$  in a direction

#### 4 A multi-physics coupling system

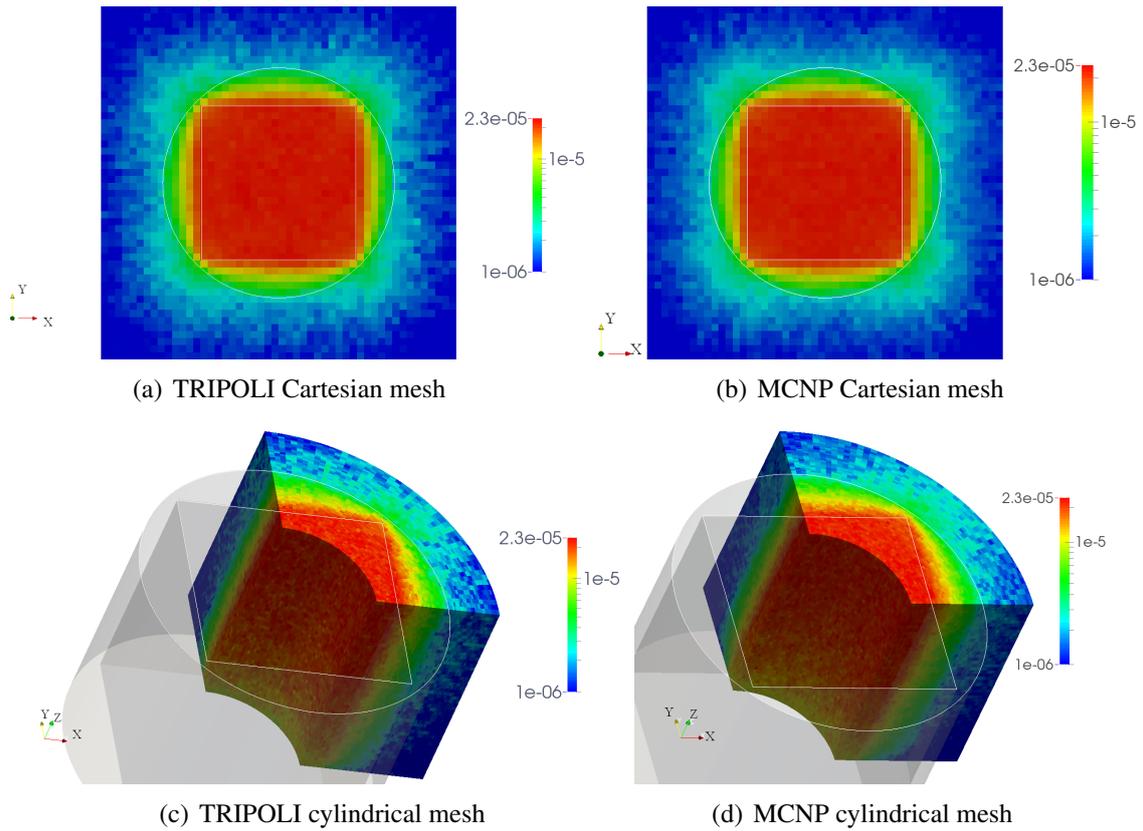


Figure 4.42: Comparison between TRIPOLI and MCNP mesh tallies on neutron flux ( $\text{cm}^{-2}\text{s}^{-1}$ ) normalized to one neutron history. The agreements on the neutron flux values and distribute patterns are shows the reliability of the TRIPOLI mesh tally interface.

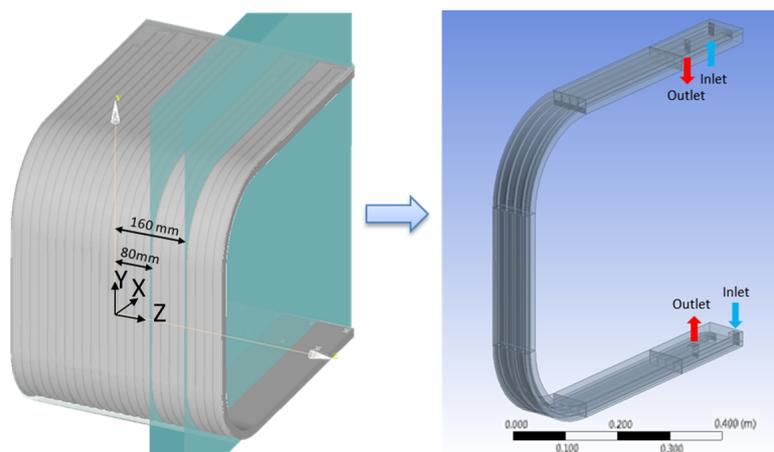


Figure 4.43: The FW model for CFD calculations

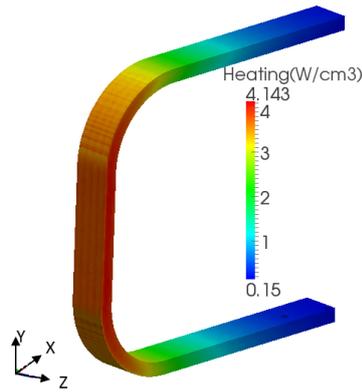


Figure 4.44: Calculated volumetric nuclear heating on the FW.

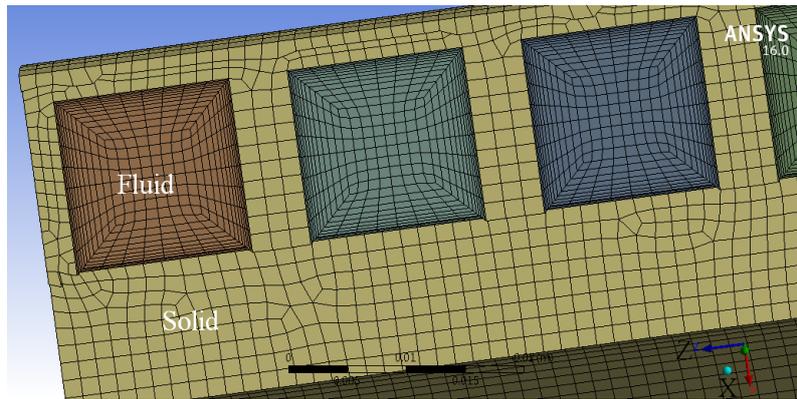


Figure 4.45: Meshes for the fluid and solid domains generated by ANSYS WB. The boundary layers of the fluid mesh consist of 12 layers with an initial layer height of 0.1 mm. The solid mesh has elements size of  $\sim 1.5$  mm.

parallel to the surface normal. Based on the geometry feature, periodic boundary conditions are applied on the lateral surfaces (the slide plane in Figure 4.43). The other outer surfaces are considered adiabatic thus no heat transfer is assumed towards the outside of the domain. The boundary conditions for one cooling channel are as follows: Inlet temperature 300 °C, pressure 80 bar, and mass flow rate 0.1 kg/s. The fluid helium is considered to be incompressible because of the low velocity of  $\sim 80$  m/s. Ideal gas is considered for the helium coolant, which assumes a constant specific heat, but a temperature-dependent density, thermal conductivity and dynamic viscosity. More detail material properties can be found in [KB12].

The conjugate heat transfer approach has been adopted for solving the coupled fluid-solid heat transfer. The two-equation  $k - \omega$  shear stress transport (SST) turbulence model is chosen for closing the turbulent equations. The meshes and all the conditions are set identically in both the CFX and Fluent codes. Furthermore, the pressure-based coupled solver is intently chosen in Fluent to keep consistence with CFX. The calculations with the two codes converged in heat transfer residuals of  $10^{-5}$ , maximum solid and fluid temperatures, and also the outlet temperatures. The wall distance factor  $y^+$ , which is a non-dimensional distance from the wall to the first node of the fluid mesh, has been kept

Table 4.7: Hotspots and outlet temperatures calculated for the configuration in Figure 4.43.

Temperature	CFX (°C)	Fluent (°C)	Diff. (°C)
Solid Max.	474.81	473.55	1.26
Fluid Max.	421.47	423.59	2.12
Outlet Avg.	335.45	334.94	0.51
Solid Max. raise	6.44	6.43	0.01
Fluid Max. raise	5.95	5.84	0.11
Outlet Avg. raise	4.93	4.53	0.40

in a range of 30-300 for almost the whole fluid domain.

As the temperature is of the most concern in this comparison, the fluid and solid temperature are evaluated. The maximum temperature in the solid and fluid domains and the average outlet temperature results are listed in Table 4.7. The differences of 1-2 °C in the figure and the table are considered to be negligibly small in CFD analyses. Additionally, in order to evaluate the contributions of the volumetric heat source, the cases without loading the volumetric heat source are also calculated. The temperature raises are listed as well in Table 4.7. Very close temperature raises are obtained by both codes.

As depicted in the Figure 4.47, the temperature on the FW is higher on the plasma side, due to the strong heat flux from the plasma and also the nuclear heating from neutron and photon. The maximum temperature of 475 °C is lower than the allowable temperature of 550 °C for Eurofer steel [Tav02]. Additionally, using the adiabatic boundary condition yields a conservative temperature estimation. Note that this test case model with half length in Y direction comparing to HCPB TBM is aiming at code verifications, thus the results are not applicable for engineering analysis of the TBM. From Figure 4.47 it can be observed that the CFX and Fluent calculations agree well in the temperature ranges, pattern, and the locations of hot-spots in the fluid and the solid domains. The temperature distributions along a line at the middle of the FW crossing both the solid and fluid domain is given in Figure 4.46, which shows a reasonable agreement. However, some deviations can be still observed in the bending region of the solid domain, where the result accuracy is sensitive to the mesh configuration. Further improvement is required if more accurate result is demand.

The speed-up of the voxelization data search algorithm has been evaluated. The counterpart cases are using the sequential search algorithm, which sequentially searches a point in the entire heating data array by matching the coordinates. The number of cell centroids in the Fluent UDF file is about  $8.75 \times 10^5$ , and the number of nodes in CFX User Fortran is  $4.55 \times 10^5$ , including some redundant points in the voxel margin layer. The computation time and the speed-up of the voxelization data searching algorithm is given in Table 4.8. It can be found that  $\sim 150$  times speed-up is achieved in both CFX and Fluent computations. In addition, although the points in the Fluent UDF file are about two times more than that in the CFX User Fortran file, the influence on the speed-up are

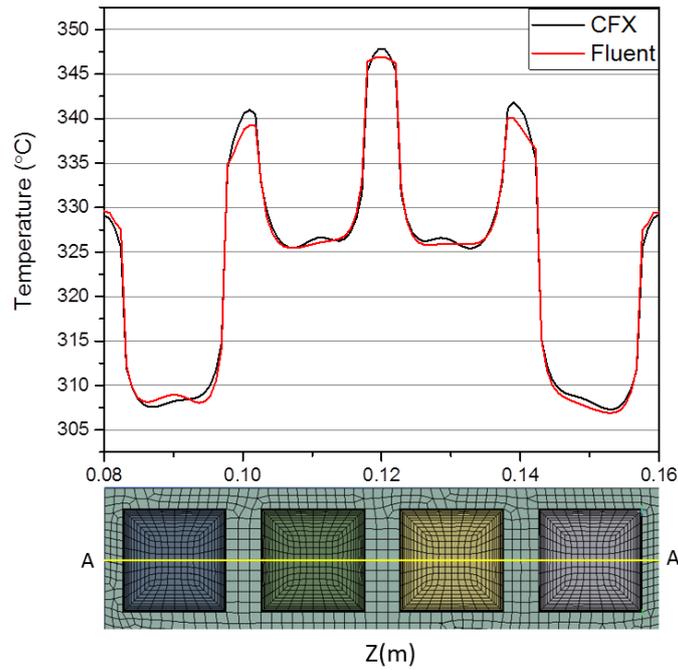


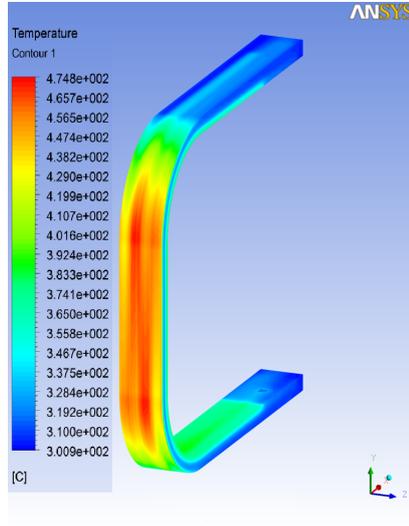
Figure 4.46: Temperature distributions calculated by both CFX and Fluent results along the line A-A crossing the fluid and solid domain. The line begins at point (0.0105, 0.0, 0.08) and ends at point (0.0105, 0.0, 0.16) based on the coordinates given in Figure 4.43.

Table 4.8: The evaluation on the speed-up of the voxelization data searching algorithm.

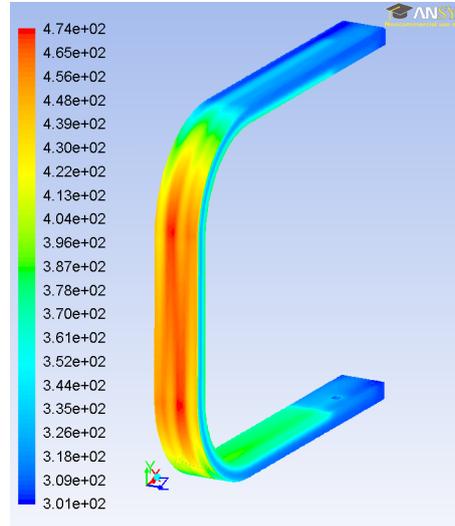
	CFX	Fluent
Points	$4.55 \times 10^5$	$8.75 \times 10^5$
Sequential (min)	100.21	113.33
Voxelization (min)	0.70	0.71
Speed-up [voxel/sequential]	143	159

not evident. Therefore, this evaluation on the voxelization data search algorithm proves good acceleration effect and also insensitiveness to the number of points. However, more computations on different meshes are needed to quantify this further.

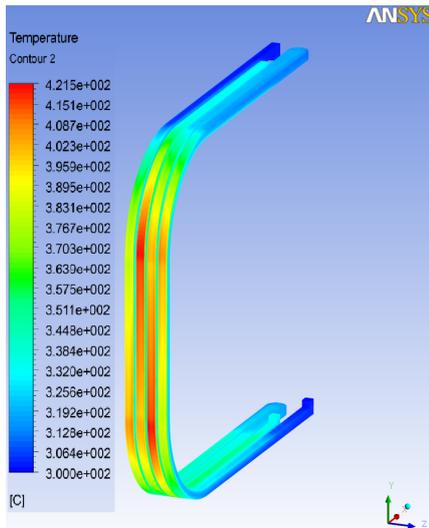
#### 4 A multi-physics coupling system



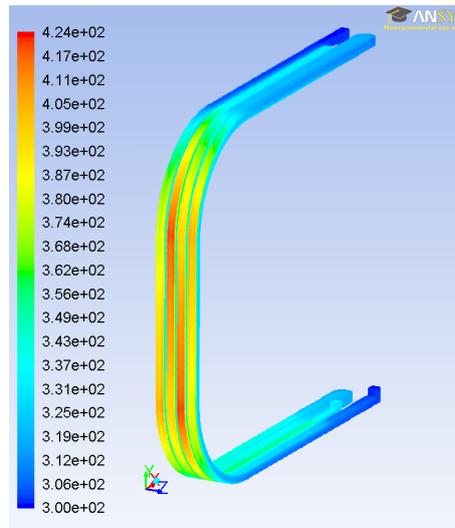
(a) CFX solid temperature



(b) Fluent solid temperature



(c) CFX fluid temperature



(d) Fluent fluid temperature

Figure 4.47: Temperature distributions of the first wall model in Figure 4.43 for the boundary conditions provided in Section 4.7.4.

# 5 General CAD- and mesh-based MC simulation

In Chapter 4, the multi-physics coupling approach for neutronics, thermal hydraulics and structural mechanics has been developed. This coupling approach is ready to be applied for other nuclear systems besides fusion reactors, as long as a general MC code is integrated. Geant4 code is considered as a suitable candidate among many investigated MC codes. It solves the transport problems of almost all kinds of particles, and has been applied in high-energy physics, medical physics as well as reactor physics.

The coupling approach requires the advanced CAD- and mesh-based geometry to be supported by Geant4. Unfortunately, Geant4 is lacking of such capabilities. One reason is that the half-space CSG solid type is not supported, another reason is that the unstructured mesh is not readily usable in Geant4. In this work, the developments of a CAD-based modeling approach and a unstructured mesh based scoring capability have been achieved, as well as the integration of Geant4 into the coupling approach.

## 5.1 Half-space solid

Half-space solid is a type of half-space CSG developed for CAD-based geometry modeling in Geant4. This section discusses the design and development of this solid type, and the corresponding conversion approach from CAD solids. The tests of this solid type in constructing common geometry shapes are given as well.

### 5.1.1 Definition of the half-space solid

A half-space solid is constructed based on the half-spaces as discussed in Section 2.1.2. Although arbitrary algebraic surfaces can be used for creating half-spaces, surface types given in the Table 5.1 are chosen in order to obtain analytic solutions in particle-geometry intersection calculations. A term, *sense* (either 1 or -1), is used for indicating a half-space on which side (either positive or negative) of the surface creating it. In addition, a quantity  $S(x, y, z)$  is used for indicating the position of a point  $(x, y, z)$  regarding to a half-space created by a surface  $f(x, y, z) = 0$ :

Table 5.1: Supported analytic surface types in a half-space solid [B<sup>+</sup>03].

Type	Description	Equation	Entries
Plane	General	$Ax + By + Cz - D = 0$	A, B, C, D
	Normal to X-axis	$x - D = 0$	D
	Normal to Y-axis	$y - D = 0$	D
Sphere	Normal to Z-axis	$z - D = 0$	D
	General	$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - R^2 = 0$	$x_0, y_0, z_0, R$
	Parallel to X-axis	$(y - y_0)^2 + (z - z_0)^2 - R^2 = 0$	$y_0, z_0, R$
Cylinder	Parallel to Y-axis	$(x - x_0)^2 + (z - z_0)^2 - R^2 = 0$	$x_0, z_0, R$
	Parallel to Z-axis	$(x - x_0)^2 + (y - y_0)^2 - R^2 = 0$	$x_0, y_0, R$
	On X-axis	$y^2 + z^2 - R^2 = 0$	R
Cone	On Y-axis	$x^2 + z^2 - R^2 = 0$	R
	On X-axis	$x^2 + y^2 - R^2 = 0$	R
	Parallel to X-axis	$\sqrt{(y - y_0)^2 + (z - z_0)^2} - t(x - x_0) = 0$	$x_0, y_0, z_0, t$
Quadric	Parallel to Y-axis	$\sqrt{(x - x_0)^2 + (z - z_0)^2} - t(y - y_0) = 0$	$x_0, y_0, z_0, t$
	Parallel to Z-axis	$\sqrt{(x - x_0)^2 + (y - y_0)^2} - t(z - z_0) = 0$	$x_0, y_0, z_0, t$
	On X-axis	$\sqrt{y^2 + z^2} - t(x - x_0) = 0$	$x_0, t$
Torus	On Y-axis	$\sqrt{x^2 + z^2} - t(y - y_0) = 0$	$y_0, t$
	On Z-axis	$\sqrt{x^2 + y^2} - t(z - z_0) = 0$	$z_0, t$
	In general form	$Ax^2 + 2Bxy + 2Cxz + 2Dx + Ey^2 + 2Fyz + 2Gy + Hz^2 + 2Iz + J = 0$ $(x^2 + y^2 + \frac{a^2}{b^2}z^2 + r^2 - a^2)^2 - 4r^2(x^2 + y^2) = 0$	A, B, C, D, E F, G, H, I, J r, a, b
On Arbitrary base and axis	Transformation to coordinate with base $(x_0, y_0, z_0)$ and axis $(x_d, y_d, z_d)$	$x_0, y_0, z_0, x_d, y_d, z_d, r, a, b$	

$$S(x, y, z) = \text{sense} * f(x, y, z) \begin{cases} < 0 & \text{if outside the half-space} \\ = 0 & \text{if on the surface} \\ > 0 & \text{if inside the half-space} \end{cases} \quad (5.1)$$

A point is inside a half-space or on the surface only if it is within the semi-algebraic subset  $P \subset \mathbb{R}^3$  [BCR98] :

$$P = \{(x, y, z) \in \mathbb{R}^3 | S(x, y, z) \geq 0\} \quad (5.2)$$

The half-space solid defined in this paper is constructed by Boolean *intersections* of half-spaces. A point is inside or on the boundaries of a half-space solid only if it is inside the closed subset  $Q \subset \mathbb{R}^3$ :

$$Q = \bigcap_i^n P_i \quad (5.3)$$

where  $n$  denotes number of surfaces, and  $i = 1, 2, \dots, n$ .

The half-space solid is a special type of the half-space CSG which is described in Section 2.1.2. The essential difference is that the half-space solid is formed purely by intersection operations of half-spaces. For illustrations, examples are given in Figure 5.1. In Figure 5.1(a), the solid is constructed by intersections of positive half-spaces (sense=1) of plane 1, 3, 5 and cylinder 7, and negative half-spaces (sense=-1) of plane 2, 4, 6. Therefore, it can be directly represented as a half-space solid. For Figure 5.1(b), the solid lays on both positive and negative half-spaces created by plane 1 and 2, which means it cannot be directly represented as a half-space solid, and should be decomposed into two half-space solids using plane 1 or 2 as cut plane, and then recombine them again. For Figure 5.1(c), it also needs to be decomposed since the solid lies on both positive and negative half-spaces of the cylinder surface. The cylinder surface can be used for cutting, or introduce an appropriate auxiliary surface such as plane 1.

Note that a half-space solid is also not equivalent to a *convex* solid. A solid is convex when a ray going out of the solid cannot re-enter the solid again, otherwise it is a *concave* solid. For example, the solid in Figure 5.1(c) is a convex solid, and the two in Figure 5.1(a) and Figure 5.1(b) are concave. However, as we discussed, the solid in Figure 5.1(a) is a half-space solid, and the others are not. In order to distinguish them, we defined the

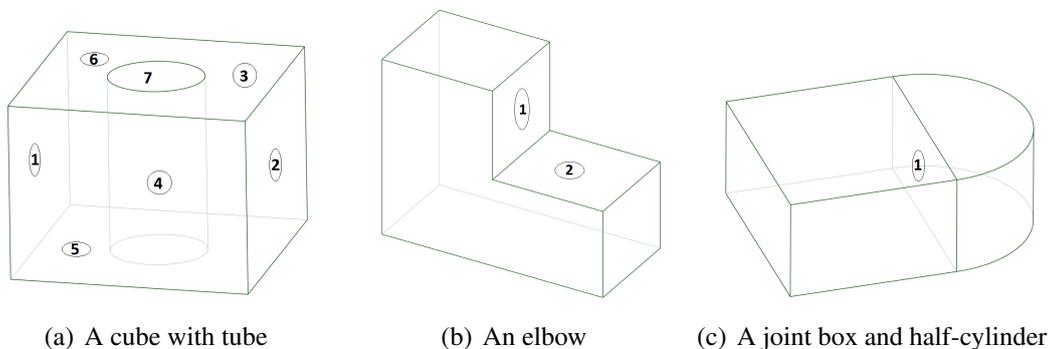


Figure 5.1: Examples for half-space solid. The numbers are notations of surfaces.

half-space solid as a *sense-constant* solid, which means that all the points inside (but not on the surface of) the solid are sense-constant regarding to every bounding surface.

Using only the intersection operation to build the half-space solid would surely introduce certain limitations. However, its capability in representing a CAD solid is not influenced. It is proved that a BRep CAD solid can be subdivided into half-space solids by introducing suitable splitting surfaces, and then represented by these half-space solids by union operations. Detail discussion on this decomposition process will be given in the Section 5.1.3.

## 5.1.2 Implementation of the half-space solid

The half-space solid geometry has been implemented as a new solid type in Geant4. Geant4 offers a standard framework for defining a new solid type. As shown Figure 5.2, a Geant4 solid should provide methods for:

1. Judging of the relative position of a point, whether it is inside, outside or on the surface of the solid;
2. Calculating the distance to enter the solid, if the ray is outside the solid; Or the distance to exit the solid, if the ray is inside the solid;
3. Calculating the safety (or underestimated) distance from a point to the boundaries of a solid along any direction, if the point is outside the solid; Or the safety distance from a point to the boundaries of the solid along any direction, if the point is inside the solid;

These methods have been implemented for the half-space solid, which are discussed in the following.

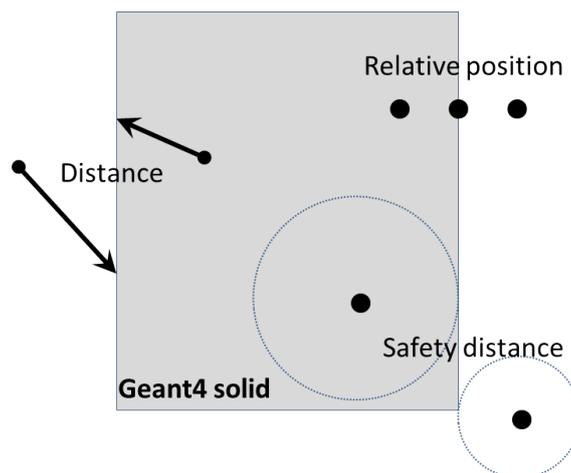


Figure 5.2: Methods should be implemented by a Geant4 solid for judging the relative position of a point, calculating distance of a ray to enter or exit the solid, and calculation of safety distance of a ray to the solid boundaries.

### Relative position

A point is inside or on the boundaries a half-space solid, when it is inside the closed subset  $Q$  given in Equation 5.3. On the other hand, the complement of  $Q$ , denoted as  $\bar{Q}$ , represents the subset outside the half-space solid. Based on the set theory, we get  $\bar{Q} = \bigcup_i^n \bar{P}_i$ . It means that, if a point is outside one of the half-space consisting the solid, then this point is outside the half-space solid. Therefore, the relative position of the particle is calculated by the follow procedures:

1. If  $S(x, y, z) < 0$  is true for one half-space, then the particle is outside the solid.
2. Else if it is on one of the surfaces, then it is on the surface;
3. Else it is inside the solid.

### Distance to enter a solid

The intersection calculation of a ray with a half-space solid is performed by intersection calculations with its bounding surfaces. A set of intersection points are calculated, and those outside the solid are excluded, since they are not the actual intersection points. The points remaining are considered possible intersection points. The smallest intersection distance will be taken as the final solution in most of the cases.

However, there are some exceptions which have to be considered. For example, the ray  $a$  illustrated in Figure 5.3 is crossing exactly through the intersection line of the cylinder and the plane at point 1, and enter the solid in point 2. Point 2 should be the correct entry point, even though point 1 is the nearest one. To avoid faulty answers, an additional probe step is calculated. Taking the point 1 for illustration, a point with a very tiny distance in the front of point 1 is calculated and checked if inside the solid. If yes, point 1 is accepted as valid the intersection point, and intersection distance is calculated, otherwise the ray is just passing by the solid at point 1.

Attention has to be paid also to the precision problem. Take the ray  $b$  for example, the base of the ray is supposed to be on the bounding surface. However, because of the limitation of a float-number in representing a real number, this point might be already inside the

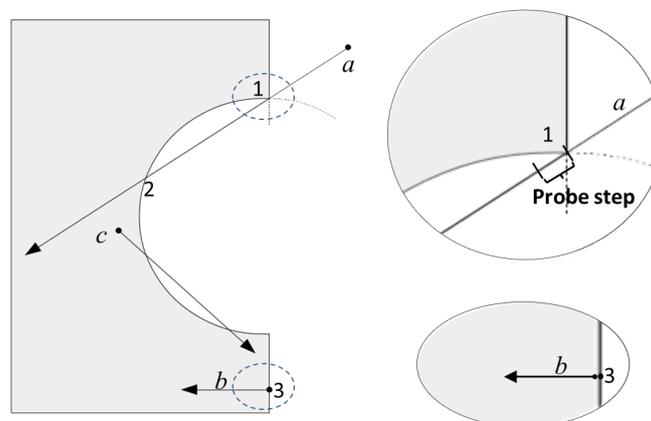


Figure 5.3: Illustrations of special cases for ray intersection calculations.

solid. For solving this issue, the cosine of the angle between direction vector of the ray and vector of the surface normal is calculated, denoted as  $\cos \theta$ . Here we consider the surface normal is pointing outward of the solid. If  $\cos \theta < 0$ , the ray is entering the solid, and the point 3 is accepted as intersection point even if it is behind the base point.

### Distance to exit a solid

The method of calculating the exiting distance is similar as that of calculating the entering distance. One difference is that Geant4 requires checking and returning the surface normal at the exiting point on a concave boundary. For example, the ray  $c$  in Figure 5.3 might re-enter the solid if no collision happens outside the solid. The judgment of concave boundaries depends mainly on the surface types. For planar surfaces, the ray cannot re-enter the solid, when it is going out from it. For sphere, cylinder and cone surfaces, if the solid is on the positive half-spaces, then the ray might re-enter the solid, otherwise it will not. For general quadratic and torus surfaces, it is difficult to give exact answers. In these case a conservative answer is given, that is assuming that they are concave boundaries. Then Geant4 will also include this solid for checking possible re-entering.

### Safety distance from outside

In Geant4, the isotropic distance (or *safety distance*) from a point to the nearest geometry boundary is used for speed-up the particle tracking. Once a physical step is proposed, the step length is first compared with the safety distance. If not greater, the physical step is adopted, and the calculation of exact distance to the geometry boundary is avoided; otherwise, the ray intersection should be calculated, and the physical step is compared with the exact intersection distance. Therefore, the safety distance is an underestimated value which should be less than or equal than the exact intersection distance. In addition, the calculation of the safety distance should be considerably faster than that of the intersection calculation.

The boundary box of the solid is used for safety distance estimations, as shown in Figure 5.4, which is a 2D case but representative for 3D. First the point is checked to which side of the boundary it closes. For the point in Figure 5.4 is close to the upper side of X-coordinate and lower side of the Y-coordinate. And then the distance from the point to this side is calculated. It can be negative if the coordinate in one direction is within the range of the boundary box. Then the largest distance among three coordinates is chosen as the safety distance. In this case the  $\Delta X$  is chosen. If the largest distance is smaller than 0, then the point is inside the boundary box, and the safety distance is considered as 0.

### Safety distance from inside

The estimation of safety distance from inside to the boundaries of the half-space solid requires the calculation of isotropic distance from a point to the boundary surfaces. This calculation is straightforward for plane, sphere, cylinder and cone surfaces, whereas it is time-consuming for general quadratic surfaces and torus. In order not to oppositely *slow down* the calculation, the safety distance calculations for these surfaces are simply abandoned, and a value of 0 is returned instead.

### Tolerance

When conducting Monte-Carlo particle tracking, significant attentions should be paid for the precision problem in the floating-point arithmetic. Take the intersection calculation

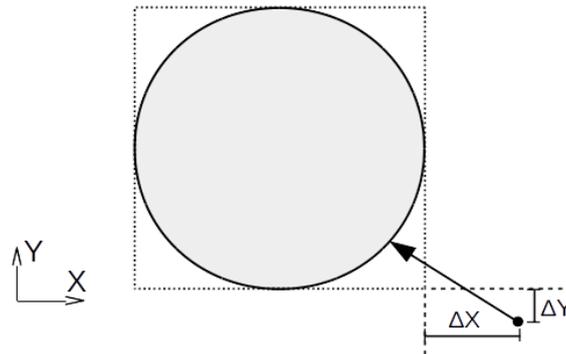


Figure 5.4: Safety distance from a point outside to the solid

for example, an intersection point  $(x_i, y_i, z_i)$  is supposed to have  $f(x_i, y_i, z_i) = 0$  when substituting the point coordinates into the surface equation. However, this might be not true due to the precision problem. Instead,  $f(x_i, y_i, z_i)$  might be very close to, but not equal to 0. Therefore, the intersection point might be very close to the surface instead to be on the surface. To avoid this problem, a tolerance  $\epsilon$  is defined, which is by default  $10^{-9}$  for geometry dimensions. Thus points, which satisfy  $|f(x, y, z)| \leq \epsilon$  are considered to be on the surface. Such tolerance has been used for many occasions, where the equal-to-zero judgements should be avoided.

### Limitations of the half-space solid

The developed half-space solid aims to provide an approach for a CAD based geometry modeling in Geant4. It relies on modeling tools and also Geant4 functionalities to create half-space solids. Firstly, because half-space solid is described a sense-constant solid formed by intersections of half-spaces. It relies on Geant4 Boolean operations, especially the multi-union function, to build a complex solid. Also, it relies heavily on CAD conversion tools for providing important information such as the boundary box, volume size and surface area. It also requires CAD conversion tools for faceting CAD solids, so that they can be displayed by the Geant4 visualization interface. The last important thing is that, the overlap checking for this solid type is expensive. Because Geant4 overlap checking requires random points to be generated on the boundary surfaces, and these random points are calculated by intersections of random rays with the solid. These limitations should be carefully considered when applying the half-space solid for a Geant4 simulation.

### 5.1.3 CAD conversion for Geant4

BRep geometry is commonly adopted in CAD system for geometry representation. A BRep geometry can be converted into half-space CSGs, since the surfaces forming the half-spaces are given by the boundary faces of a BRep solid. Under a subsequence of

decompositions, it has been proven that a BRep solid can be converted to a closed semi-algebraic subset[TT01] by:

$$R = \bigcup_j^m \bigcap_i^{n_j} \{(x, y, z) \in \mathbb{R}^3 | S_{j,i}(x, y, z) \geq 0\} , \quad (5.4)$$

where  $S_{j,i}$  is a polynomial in  $\mathbb{R}^3$  for  $i = 1, 2, 3, \dots, n$  and  $j = 1, 2, 3, \dots, m$  defined in Equation 5.1. Substituting  $Q$  in Equation 5.3 to the Equation 5.4, we can see that the subset  $R$  is actually the union of half-space solids:

$$R = \bigcup_j^m Q_j . \quad (5.5)$$

Therefore, it can be concluded that a BRep solid can be represented by unions of finite numbers of half-space solids.

In general, the surfaces of BRep solid boundaries are not sufficient for a decomposition, except all faces are planar. For example in Figure 5.1(c), it is favorable to introduce an auxiliary cutter surface to decompose the solid. The decomposition method is already implemented in McCad using the algorithms discussed in [LFP14].

In this work, a new McCad interface has been developed to export the decomposed half-space solids for Geant4. An extended version of the Geometry Description Markup Language (GDML) format has been implemented to store the half-space solids. Additionally, the GDML parser in Geant4 has been extended to process the new GDML file.

### Extension of GDML for half-space solids

To achieve geometry data transfer from McCad to Geant4, it is important to define the interface file. There are several possible solutions, for example, existing MCNP or TRIPOLI input files as intermediate files, or a self-defined format, or a general geometry description format which is widely accepted by the MC code. The first solution is straightforward in considering CAD conversion, because efforts to develop a new interface are avoided. However, because the MCNP format is very flexible, it might be very difficult to translate arbitrary MCNP geometries to Boolean combinations of Geant4 half-space solids. For the second solution, a self-defined format has a bad compatibility, and requires a lot of framework design. The third solution has been adopted in this work, using a MC geometry format called GDML.

GDML [CMPS06] is an application independent geometry description format based on XML. It is accepted by the MC codes Geant4, FLUKA, TRIPOLI and ROOT (a Data Analysis Framework [BR97]) for saving geometries. It has self-consistent definitions of syntax, which are specified as a XML schema. The GDML can be extended for a new solid type, as long as the GDML schema is extended. In this work, the half-space solid has been integrated into the GDML description. This is achieved by extending the GDML schema and the Geant4 GDML parsers as well.

The GDML *schema* is a XML schema which defines the structure and convention of a GDML file. It can be used for verifying the contents. According to the GDML schema, a GDML file should include five XML blocks, as shown in Figure 5.5 —definitions, materials, solids, structure and setup. These blocks should be given in a fixed order as listed below.

1. The *Definitions* block contains predefined numerical values of constants, points, rotations, etc., which are used for geometry descriptions.
2. The *Materials* block contains materials with compositions.
3. The *Solids* block contains all Geant4 solid types, including primitive CSG solids, tessellated solids and Boolean solids.
4. The *Structure* block contains *Logical Volumes*, which associates a solid with a material. Logical volumes with specific spatial transformations are called *Physical Volumes*. Physical volumes can be placed again into a parent logical volume, forming a tree-like geometry structure.
5. The *Setup* block specifies the top level physical volume of the geometry tree, which is called the *World*.

The GDML schema has been extended to integrate the half-space solid. As shown in Figure 5.5, the half-space solid is built from a surface block, which contains the surface definitions of this solid, a boundary box specified with upper and lower points, the volume, the area, and also a reference of a polyhedron for visualization purposes. The name of the solid is uniquely given, so that it can be used as reference of the solid for other blocks.

The Geant4 Boolean solid is used for combining half-space solids. Currently, GDML only supports Boolean operations with two solids. To be more flexible, a *nested Boolean solid*

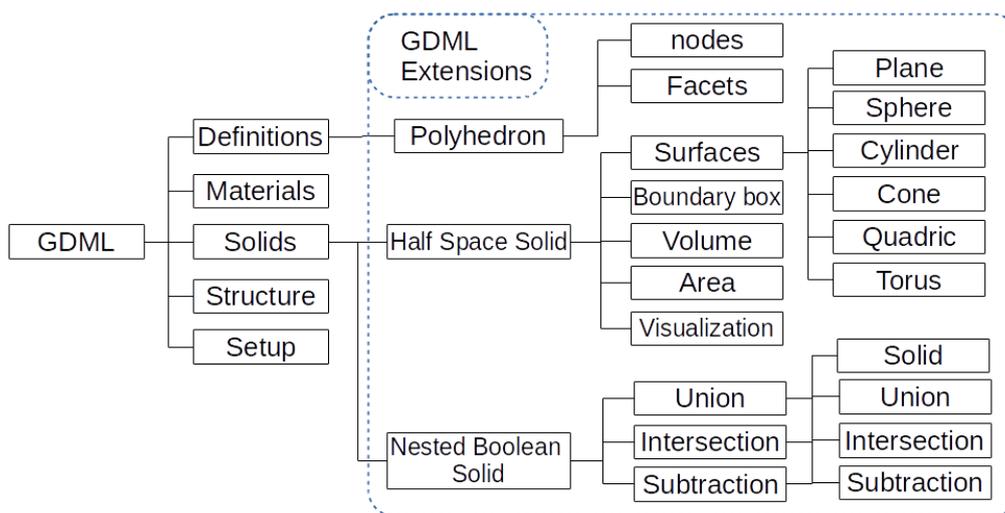


Figure 5.5: The GDML structure and the extensions made for integrating half-space solids for a closed geometry description of engineering bodies.

type has been created. As shown in Figure 5.5, Boolean operators union, intersection, and subtraction can be used in a nested manner.

Visualization of the half-space solids is important for users. In Geant4 code, a solid is visualized by tessellating the solid into a polyhedron, which consists of triangular or quadrangular facets. Because the boundary surfaces of half-space solids are not explicitly constrained by edges, it is difficult to generate facets for the visualization purpose. To avoid huge efforts, these polyhedra are generated by faceting the CAD solids in the conversion tool, and written in a GDML file as a list of nodes and facets. These polyhedra are defined in the definition block, as shown in Figure 5.5. They are referred by names in the solid block.

Additionally, the GDML parser interface in Geant4 has been extended. The Xerces-C++ library[Xer10], which is a XML library, has been employed to process the GDML file. The polyhedrons defined in the definition block are at first computed to build a list of Geant4 visualization objects. The solid block parser has been extended to process the surfaces and other parameters to create new half-space solids. Also, the nested Boolean solids are processed into Geant4 built-in Boolean solids.

### **McCad interface for GDML**

A McCad interface has been developed to export decomposed half-space solids to the extended GDML files. The interface is implemented based on the QtXml library[QtX14], which is a sub-library of Qt for XML processing. The conversion process is similar as described in Section 4.2.3. One difference is that the void space generation is not necessary, because of an implicit void treatment in Geant4. The model is exported by the McCad GDML interface in the following sequences:

1. The names for solid and material have to be checked, because names are used as references.
2. The generation of polyhedrons is carried out exactly as in the tessellation phase mentioned in Section 4.3.4. Attention should be paid for the face orientation in the CAD model. If a face is reverse-oriented, then the orientation of its facets should be also reversed, otherwise it will cause a visualization problem. The polyhedra data are exported to the GDML definition block.
3. If material information has been defined in McCad, it will be also included in GDML files. Otherwise air as material is created for filling the world.
4. In McCad, a part consists of one or more sense-constant half-space solids, and each half-space solid contains a list of surfaces with senses. This information is retrieved and written into the GDML file. The boundary box, volume and surface area are calculated with Open Cascade library functions, and assigned for its solid. The half-space solids are exported to the solid block, and also a Boolean solid is created, which combines these half-space solids into the original CAD solid. A extra envelope solid is also created for the description the world.

5. Solids are associated with materials to build logical volumes in the GDML structure block. Also physical volumes are created by linking them with logical volumes. No spatial transformation has been made for those physical volumes.
6. The physical volume *world* is setup by associating the world solid with air as material.

#### 5.1.4 Test validations

The Geant4 half-space solid has to be intensively tested before application. Geant4 has a series of well-validated primitive CSG solids. They can be used as references for this test. Several Geant4 CSG solids based on first- and second-degree surfaces and the fourth-degree torus surface have been chosen. The chosen solids are shown in the Figure 5.6. Tests have been performed with random points and random rays generated inside the boundary boxes with certain margins to the solid boundaries. Each item has been evaluated with  $10^5$  samples, except the volume comparison. Following items have been tested, covering all mandatory functions for a standard Geant4 solid:

1. Solid volume. The volume size is an important indication for the correctness of a half-space solid. The volumes of the Geant4 primitive CSG solids are calculated analytically, and the volumes of half-space solids are estimated with the MC method. Geant4 offers a general way to estimate the solid volume by generating random points inside the boundary box, and then estimates the volume by the ratio of points inside the solid.  $10^6$  points are sampled for every solid. The estimated volumes would have certain statistical errors, thus would not be identical with the analytical results.
2. Related position of a point. The related positions of the random points are checked in both half-space solid and reference solid. The comparison is passed when all the relative positions are observed to be consistent.
3. Surface normal. Random points are generated on the boundary surfaces of the reference solid, and used to calculate the surface normal of both solids. The comparisons are evaluated by calculating the mode of the difference vector of the two surface normals.
4. Safety distances from outside or inside. For comparing the safety from outside, the random points generated inside the solid are rejected; oppositely, for comparing the safety from inside, the random points generated outside the solid are rejected. Note that the results of the half-space solid and the primitive CSG are not necessarily identical in this test. One reason is that the method to calculate safety from outside is not always the same, another reason is that the safety distance from inside are given up for general quadratic and torus surfaces. In those cases the comparisons are not applicable (N/A).
5. Ray distance to enter or exit a solid. This comparison is of importance for the half-space solid. The rays are generated inside the boundary box. For testing the

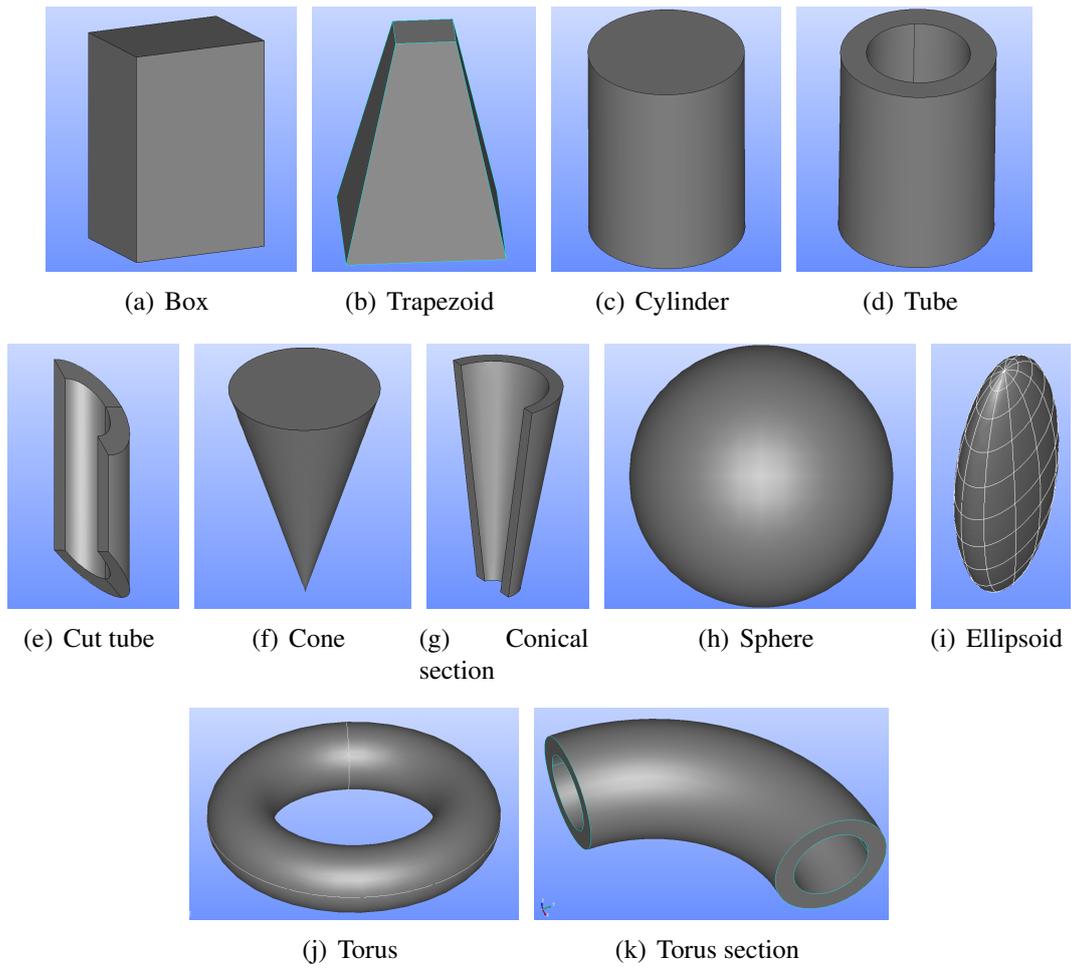


Figure 5.6: Geant4 primitive solids which are chosen for testing the half-space solid.

distance to enter the solid, those rays located inside the solid are rejected, vice versa for testing distance to exit the solid. The obtained intersection distances are compared.

6. CPU time slow down. The computation time of a half-space solid depends on the boundary surfaces it includes. Also, because the code and method are not very well optimized at this moment, the computation speed of this solid type might not exceed the Geant4 primitive CSG solids. The computation time slowdown is calculated.

The results are summarized in Table 5.2. Those results for the unstructured mesh elements will be discussed in Section 5.2. The comparison results for the volumes are relative deviations of the volume values. The computation time slowdown is calculated by dividing the CPU time used for half-space solids with that for the Geant4 primitives; the results for other items are represented using the maximum value among all the samples. The results of the safety distances are not applicable, if the calculation methods in two solid types are different.

Table 5.2: Differences between half-space solids and Geant4 primitives in MC simulations. Unit of distance: (mm).

	Volume (%)	Relative position	Surface normal	Distance to enter	Safety outside	Distance to exit	Safety inside	Slow down
Half-space solid								
Box	0.001	Pass	0	$5.68 \times 10^{-14}$	$1.00 \times 10^{-09}$	$5.68 \times 10^{-14}$	0	5.5
Sphere	0.023	Pass	$2.77 \times 10^{-32}$	$2.64 \times 10^{-12}$	N/A	$2.47 \times 10^{-13}$	0	2.6
Cylinder	0.012	Pass	$2.47 \times 10^{-32}$	$6.39 \times 10^{-13}$	N/A	$1.17 \times 10^{-13}$	0	4.4
Cone	0.156	Pass	$7.43 \times 10^{-23}$	$8.33 \times 10^{-12}$	N/A	$1.69 \times 10^{-11}$	$8.88 \times 10^{-15}$	2.4
Torus	0.163	Pass	$1.20 \times 10^{-29}$	$6.17 \times 10^{-09}$	N/A	$1.20 \times 10^{-10}$	N/A	0.9
Trapezoid	0.014	Pass	$1.08 \times 10^{-25}$	$3.12 \times 10^{-08}$	N/A	$1.32 \times 10^{-07}$	$3.80 \times 10^{-10}$	4.9
Tube	0.133	Pass	$2.47 \times 10^{-32}$	$1.90 \times 10^{-12}$	N/A	$3.48 \times 10^{-13}$	$9.17 \times 10^{-16}$	2.3
Cut Tube	0.099	Pass	$2.47 \times 10^{-32}$	$1.48 \times 10^{-12}$	N/A	$1.85 \times 10^{-12}$	$8.46 \times 10^{-09}$	1.4
Cone section	0.123	Pass	$1.63 \times 10^{-31}$	$1.56 \times 10^{-12}$	N/A	$2.78 \times 10^{-12}$	$7.33 \times 10^{-15}$	1.4
Ellipsoid	0.002	Pass	$1.73 \times 10^{-31}$	$9.35 \times 10^{-12}$	N/A	$9.18 \times 10^{-14}$	N/A	2.4
Torus section	0.175	Pass	$1.92 \times 10^{-29}$	$1.88 \times 10^{-10}$	N/A	$2.24 \times 10^{-11}$	N/A	0.8
Unstructured mesh element								
Box	0	Pass	$1.71 \times 10^{-13}$	$1.14 \times 10^{-13}$	N/A	0	0	1.6
Trapezoid	0	Pass	$4.93 \times 10^{-32}$	$4.26 \times 10^{-13}$	N/A	$8.24 \times 10^{-13}$	N/A	1.4
Wedge	0.012	Pass	$6.16 \times 10^{-32}$	$2.64 \times 10^{-12}$	N/A	$6.82 \times 10^{-13}$	N/A	0.4
Pyramid	N/A	Pass	$1.54 \times 10^{-23}$	$5.09 \times 10^{-08}$	N/A	$4.76 \times 10^{-10}$	N/A	1.3
Tetrahedron	0	Pass	0	0	0	0	0	1.2

The volume differences are within a reasonable level, which can be more accurate if more points are sampled. For all other results, the differences are below the level of geometry tolerance of  $10^{-9}$  mm. There are a few exceptions, e.g. the trapezoid and the torus, which have larger differences than other solids. The surface parameter defined by McCad in this test has only 7 significant digits, which results in the slight differences in the results for trapezoid. The reason for the torus is that, the ray-intersection calculation for torus surface is performed with a numerical algorithm, which has certain residuals. However, these small differences are negligible in practical simulations. Regarding the computation time, the CPU time used for half-space solids is in general more than that of Geant4 primitives. Further optimization on the codes and algorithms are suggested, which will be carried out in the future.

## 5.2 Unstructured mesh geometry and scoring

An unstructured mesh is an attractive geometry type for MC simulations due to its good adaptivity to a complex geometry. It is also valuable for scoring the spatial distribution of nuclear heating, so that the multi-physics coupling can be achieved.

In this work, a set of unstructured mesh elements have been developed as new Geant4 solids, and an interface has been developed for employing unstructured mesh. These mesh elements can be used for modeling complex geometries. In addition, the Geant4 scoring has been upgraded with a new superimposed unstructured mesh scoring function. This scoring function is applicable for all particle types and various physical quantities such as flux, collision, energy deposition, etc.

### 5.2.1 New Geant4 unstructured mesh elements

Four types of common-used first-order elements — tetrahedron, pyramid, pentahedron and hexahedron have been developed as new Geant4 solid types. These elements are shown in Figure 5.7. The reason for choosing first-order element is that the ray intersection calculations are much faster due to their planar boundaries. Second-order elements, which preserved better the surface curvatures, have higher cost in computing the ray intersection with its second-order surfaces.

There are many things in common for these four types of elements. Therefore, they are generalized as one solid type. A few differences are on the number of nodes and surfaces, which affect only on the geometry preparation of each element type.

#### Geometry preparation

A list of 3D points (or nodes) is used for constructing a mesh element. The nodes are ordered exactly according to Figure 5.7, since this ordering reveals the nodal connectivities of this element. A list of surfaces is then created, and each of them contains a list of indexes of nodes constructing this surface. These surfaces represent a closed Geant4 solid. In the geometry preparation step, the volume size of the solid is checked. If

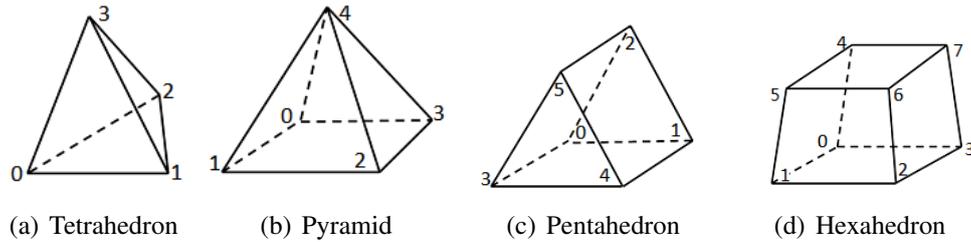


Figure 5.7: Four types of first-order elements and their notation.

the volume is negative, a method is employed to heal the element by reversing the node ordering. Surface area and also the boundary box are calculated. After these preparations, those four types of elements can be considered as a general first-order element solid for simulation.

The method of calculating the volume is splitting a general first-order element into several tetrahedra and calculating their volumes. Attention must be paid on splitting the tetrahedron volume. The nodes of a tetrahedron should be ordered correctly, so that the first three nodes  $N_0, N_1, N_2$  are in anticlockwise orientation, when looking from the fourth node  $N_3$ , as shown in Figure 5.7(a). Otherwise the volume is negative. The volume of the tetrahedron is given by

$$V = \frac{1}{6}(N_0 - N_3) \cdot ((N_1 - N_3) \times (N_2 - N_3)) \quad (5.6)$$

When a negative volume is found, it is probably due to a reverse of node-ordering. For example, rather than ordering the nodes for a tetrahedron and a hexahedron in Figure 5.7(a) and 5.7(d), the nodes might be ordered as shown in Figure 5.8(a) and 5.8(b). This problem can be fixed by swapping nodes into the proposed order in Figure 5.7. If negative tetrahedron volumes are still found, the nodes might be totally disordered. In this case, the error will be reported.

The method for calculating the surface area is splitting each boundary face into triangles, and then calculating the total area of them. For surfaces, which have more than three nodes, nodes might not be exactly coplanar, which means that the surface is creased (as shown in Figure 5.9). To avoid a possible error, the distances of nodes to the plane formed by the first three nodes are checked. If a specified tolerance is exceeded, the error will be reported.

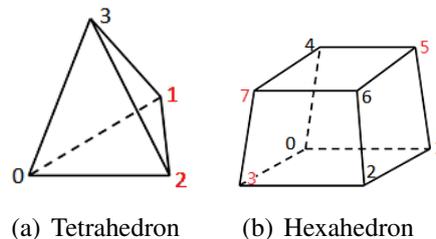


Figure 5.8: Tetrahedron and hexahedron with reversed node ordering.

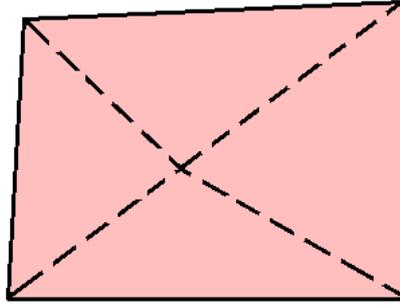


Figure 5.9: A creased planar face caused by four non-coplanar nodes.

### Particle tracking

Similar as a half-space solid, calculation methods for these new solids types have been provided for calculating the relative position, the ray intersection distance and the safety distance. The calculation method is also similar, except the ray intersection. A vector based calculation method discussed in [Bou91] has been used.

There are some comments for using these mesh elements in MC simulations. At first, the mesh overlap checking is extremely time-consuming. Sometimes millions of elements are contained in a mesh, thus it is a huge computational effort to generate random points in each element and check the overlaps. Furthermore, acceleration algorithm is needed to cope with the large number of elements used in MC simulations. The *Smart Voxel* algorithm implemented in Geant4 is very efficient for the mesh geometry, making a use of the mesh feasible in practical simulations.

## 5.2.2 Unstructured mesh scoring

### Parallel world and scoring mesh

Sometimes, it is not reasonable to use structure geometries as detector geometries. For example, if a simple structure block is used for setting hundreds of detectors, then the block should be divided into hundreds of geometries for creating the detector, which induces great burdens on simulations. Geant4 provides an advanced method for defining detector geometry on a so-called *Parallel World*. In a parallel world, geometries are superimposed on the actual geometries, and are not affecting the actual simulation. Therefore, it is obsolete to split the structure geometry when detector geometries are defined in a parallel world. A parallel world can also superimposed on other parallel worlds, thus geometry overlaps between two parallel worlds are allowed.

This feature is suitable for implementing a superimposed mesh scoring. In Geant4, an orthogonal mesh scoring function is already implemented. It supports uniform-interval orthogonal meshes in Cartesian or cylindrical coordinates. Mesh is built in a way that first a box or cylindrical tube sector is built, and then a mesh is constructed either by replicating or dividing the geometry. However, this mesh scoring function is not sufficient for complex models.

## Unstructured mesh scoring for Geant4

In this work, a superimposed unstructured mesh tally (scoring) feature has been developed for Geant4. This development is based on the Geant4 orthogonal mesh scoring, extending all its functions to unstructured meshes. Changes have been made on the detector geometry construction, scoring quantities and filters setup, results displaying and export of the orthogonal mesh scoring function.

A parser interface has been developed for importing meshes into Geant4. VTK format is chosen due to its flexibility in mesh element types and wide acceptance in meshing tools. Four element types, tetrahedra, pentahedra, pyramids and hexahedra, are currently supported. However, the interfaces are designed to be compatible with most of the first- and second-order element types. In this interface, element connectivities and associated nodes are retrieved to setup the elements. Then they are associated with dummy materials to form Geant4 logical volumes. Solids in Geant4 are not explicitly indexed. In order to preserve the element indexes of the mesh, these indexes are assigned for logical volumes as special indexes called *copy numbers*. These indexes are important for the post-processing step, since the data should be relocated to the correct elements in the mesh. In addition, Geant4 multi-functional detectors are assigned for these mesh elements, so that scalar quantities like fluxes, energy depositions, etc by can be scored.

Visualization functions have been also implemented to display meshes and results. These elements are visualized as general polyhedra by splitting their boundary faces into triangles or quadrangles facets. The visualization of results is performed by mapping the values to colors. A color legend is created, which blue color is mapped to the minimum value, and the red color to the maximum value. Both linear and logarithmic color legends are provided.

The export of results is performed by a VTK writer interface developed for Geant4. The results are reordered according to the indexes assigned in the preparation step. The results in the VTK file can be visualized in open-source tools such as VisIt, ParaView, etc.

### 5.2.3 Coupling interfaces

As McCad has been upgraded to generate and manage meshes, these meshes can also be used for Geant4. The VTK mesh format can be directly supported by the MED library. Also meshes can be provided by many other meshing tools via the VTK format.

For importing the results to the coupling interface McMeshTran, an interface has been developed for processing unstructured meshes and data in VTK format. It can be applied for general VTK data files as well as the Geant4 unstructured mesh score results. The interface is based on the VTK library for parsing the files, and to conduct conversion from the VTK format to the MED format.

### 5.2.4 Verification of the unstructured mesh scoring

A similar test as mentioned in Section 5.1.4 has been performed for the four types of mesh elements developed in this work. The Geant4 solids such as box, trapezoid, wedge, pyramid and tetrahedron are used. These shapes are also constructed using the mesh element. Comparison has been shown in Table 5.2. It can be observed that the results for the trapezoid agree better than those of the half-space solids, which is mainly due to some similarity in modeling this shape between the Geant4 trapezoid and the hexahedron mesh element. For the pyramid, because Geant4 does not have this shape, it is built by a degenerated trapezoid. The volume calculation of this solid fails for the Geant4 trapezoid. Also other results for pyramids are exhibit large differences for this reason. After all, the new developed mesh elements for Geant4 pass the tests very well.

Another verification on the new developed unstructured mesh capability has been carried out using the same iron tube model mentioned in Section 4.7.3, and the results have been compared with that of MCNP6. The Geant4 High Precision (HP) neutron model has been applied, which scopes the elastic, inelastic, capture and fission processes based on evaluated data libraries. The G4NDL4.4 [MCOGC12] neutron cross-section library has been adopted, with neutron cross-section data largely from ENDF/B-VII [COH<sup>+</sup>06]. An unstructured mesh of the iron tube is generated using the ANSYS Workbench and for the mesh tally, with 11241 hexahedra and 223 pentahedra elements.  $10^6$  source neutrons have been simulated for both the Geant4 and the MCNP6 calculations.

The Geant4 results are shown in Figure 5.10(a). MCNP6 results are considered as reference, which have a statistical error below 5%. The deviation of the Geant4 results against the MCNP6 results are shown in Figure 5.10(b). It can be observed that the Geant4 results agree in general with MCNP6 results. Figure 5.11 is helpful for making a detail evaluation. Large parts of elements have deviations below 10%, which is reasonable when the statistical error in both MC simulation are around 5%. The amount of elements outside this range is  $\sim 13\%$  of the total amount of elements. Since the sampled particles are  $10^6$ , the statistics has to be improved in both Geant4 and MCNP6 results. However, these

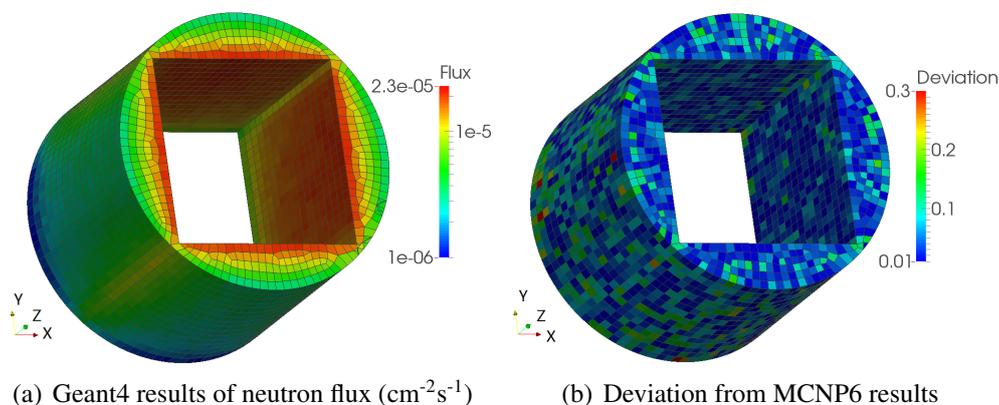


Figure 5.10: Geant4 unstructured mesh tally results and comparison with MCNP6 results

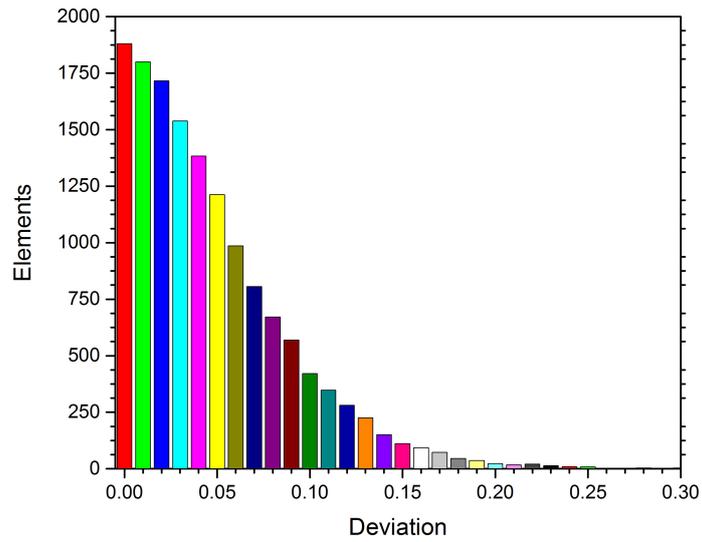


Figure 5.11: The amount of elements in different range of deviations of Geant4 simulation compared to MCNP6 (reference). The elements are grouped into bins according to their deviations, with an interval of 0.01.

results are valuable for showing the reliability of the newly developed Geant4 unstructured mesh scoring function.

# 6 Summary and conclusion

The objective of this PhD study is to develop a multi-physics coupling approach providing advanced MC geometry conversions and generic coupling interfaces for multi-physics analyses. This coupling approach has to be implemented as an integrated coupling system so that the entire analysis cycle can be carried out. This coupling system has to be verified using suitable test cases in order to prove the reliability, robustness and efficiency of the system. In the following sections, the performed work and the achievements are summarized, and conclusions and outlooks are given at the end.

## 6.1 Summaries

### 6.1.1 Advanced hybrid MC geometry conversion

McCad, which is an in-house MC geometry conversion tool, has been integrated into the SALOME platform. Based on the CAD-to-CSG conversion function of McCad, an advanced hybrid CSG and unstructured mesh approach has been developed for MC geometry modeling. A novel meshing approach called Tessellation-Tetrahedralization (TT) has been developed for generating accurate and economic MC geometry meshes. In addition, an interface has been developed generate a hybrid CSG and a mesh model for MCNP6. The McCad module is able to communicate with the SALOME CAD modeling, meshing and visualization modules for sharing CAD and mesh data.

Two challenging models, the HCPB TBM test case model and the ITER Benchmark model have been employed to verify the TT meshing approach. The generated TT mesh models have been compared with CSG models. The TT meshing exhibits a very promising efficient and accurate MC geometry modeling. The neutronic results of the TT meshing models have been compared with that of the CSG models, and a reasonable agreement has been obtained in both cases. The FE mesh model generated by ANSYS Workbench provides inferior capability with regards to MC geometry modeling, compared with the TT mesh model.

The hybrid geometry conversion capability has been also verified with the TBM test cases model. The unstructured mesh geometry of a BU has been taken together with the CSG geometries. The calculation results show that the integral nuclear heating of the unstructured mesh agrees with that of the CSG geometry. However, tiny mesh overlaps influence significantly the simulation results of the heating distributions in MCNP6-1.0.

### 6.1.2 Multi-physics coupling

For realizing the multi-physics coupling analyses from neutronics to TH and SM, a general multi-physics coupling tool McMeshTran has been developed and integrated into the SALOME platform. A generic data translation kernel has been developed based on the MED library, which performs a physical conservative data mapping between almost any unstructured meshes. Mathematic operation, spatial transformation, data sharing and visualization are provided for mesh and data manipulation and analyses.

Interfaces have been developed for a general multi-physics coupling between MC codes MCNP5/6, TRIPOLI-4 and Geant4, CFD codes CFX and Fluent, and the FE platform ANSYS Workbench. The MCNP6 interface provides nuclear heating directly on the unstructured meshes, so that the high-fidelity coupling between neutronics and TH/SM can be achieved. The heating data are exported as source subroutines for CFX and Fluent. A voxelization searching algorithm has been developed to accelerate the data search.

The TBM test case has been adopted to verify the MC data translation and the MCNP5 mesh tally interface. The integral heating of these interpolated meshes are compared to the cell-based tally results, and an excellent agreement is observed. Furthermore, an inversion check has been performed to check the interpolated heating distributions, and the data mapping is concluded to be reliable. For the TRIPOLI-4 interface, a verification calculation has also been carried out using a iron tube model. The results have been compared with that of MCNP5 results and showed a good agreement.

In order to verify the CFD interfaces, a 1/6 FW model of the TBM test cases has been adopted. The nuclear heating data are processed from a MCNP5 mesh tally. Similar meshes and calculation parameters are set up for CFX and Fluent codes, and the temperature in the solid and fluid domains have been evaluated and compared. An excellent agreement between two codes is observed, allowing for a high confidence for future design analyses.

### 6.1.3 Integrated coupling system

An integrated coupling system has been developed, which performs the seamless coupling through the whole analysis cycle starting from the CAD modeling, then the mesh generation, MC geometry conversion, MC simulation, data coupling, TH and SM simulation and finally visualization. The SALOME platform has been chosen for implementing this system. The MC conversion module McCad and the multi-physics coupling module McMeshTran have been developed, and 14 data interfaces have been implemented for completing the system.

### 6.1.4 General CAD and mesh-based MC simulation

This coupling system is a general system, which can be applied for other nuclear system as long as a general MC code is integrated. Geant4 has been chosen as a suitable candidate,

and efforts have been made for integrating Geant4 into the coupling system.

A Geant4 half-space solid type has been developed, so that the CAD solids can be converted using McCad. The GDML format has been extended for describing this new solid type. A McCad interface has been developed for converting CAD solids to half-space solids, and also the Geant4 GDML interface has been extended to parse and construct the half-space solids. Using this approach, the CAD-based MC simulation using Geant4 is achieved. Tests have been carried out for comparing half-space solid types with Geant4 native primitive CSG, and consistent results have been obtained for all tested geometries.

In addition, a superimposed unstructured mesh scoring function has been developed for Geant4 to realize high-fidelity result scoring on complex models. Element types tetrahedron, hexahedron, pyramid and pentahedron have been developed. Also, the superimposed unstructured mesh tally function has been developed, so that physical quantities distributions of various particles can be captured. Furthermore, interfaces have been developed for importing mesh and exporting results with the VTK format. This mesh scoring function has been verified by comparing with MCNP6 unstructured mesh results. There a reasonable agreement have been obtained. Since the VTK format is fully supported by McCad and McMeshTran, coupling the Geant4 code with the integrated CFD/FE codes is enabled similarly as other MC codes.

## 6.2 Conclusions and outlooks

In this work, an advanced MC modeling approach has been developed for converting complex geometries to MC models with hybrid CSG and unstructured mesh geometries. A high-fidelity multi-physics coupling approach has been developed for mapping data from several MC codes to CFD and FE codes using a physical conservative coupling scheme. These two approaches have been implemented and integrated into the SALOME simulation platform, thus a coupling system has been developed, which covers the entire analysis cycle. To extent this coupling system for general nuclear systems, the CAD and mesh-based capabilities have been developed and tested for Geant4. This coupling system and its interfaces have been demonstrated to be reliable, robust and efficient by verifying it with several test cases. It can be concluded that the overall objective of this PhD study has been achieved.

The developed system is a general and flexible system which is compatible with other MC, CFD and FE codes. The coupling system could be extended to other open source codes such as Serpent 2, FLUKA, OpenFOAM, Code\_Aster as mentioned in Section 3.2.2. Another further improvement on the system is developing a higher automation scheme, so that the automatic linking of different modules and the automatic running of calculation sequences can be achieved. To realize a time dependent safety analyses, time-dependent nuclear heating results have to be processed, and interfaces for CFD codes have to be improved as well. Temperature feedback for updating cross-sections in

## *6 Summary and conclusion*

neutronics calculation is required for nuclear systems with a strong Doppler effect. Feedback from structural mechanical calculations to the neutronics would allow taking into account strong geometry deformations caused by thermal expansions. All these aspects pose challenging topics which have to be studied in the future for improving the system especially in the view of engineering applications.

# Acknowledgement

At the end of the thesis, I would like to express my thanks to the people who offer me helps during these four years.

I would like to given my first and deeply, heartfelt thanks to my supervisor Dr. Ulrich Fischer. I am deeply impressed by his kindness, gentle and patience. His is always given positive remarks to me and my work, and always give me valuable suggestions. His encouragements during these four years are undoubtedly the key for my all achievements.

I would like to thank Prof. Dr. Stieglitz for the great helps in improving the manuscript. Through his insightful comments, I've learned a lot of things about the ways of thinking and doing as a scientist. I also appreciate Prof. Dr. Möslang from Institute for Applied Materials (IAM) for being the second reviewer of my thesis.

I am very grateful to my colleagues — Dr. Bastian Weinhorst, Dr. Arkady Serikov, Dr. Keitaro Kondo, Dr. Pavel Preslavitrev, André H'aussler, Peng Lu and other colleagues who are not listed here. Without their helps on doing the calculations and revising the thesis, the way will be much longer to walk. Special thanks to Dr. Lei Lu, a friend in need when I am losing hope in the front of difficulties.

At the end, a people who deserves of particular gratitude is my wife Qinlan Kang. I am very lucky to marry an expert in thermal hydraulics and structural mechanics, who offer me many helps in meshing and CFD/FE simulations. I am thankful for her patience, supports, and always being there for me.



# Bibliography

- [AFG<sup>+</sup>09] John Apostolakis, Gunter Folger, Vladimir Grichine, Aatos Heikkinen, Alexander Howard, Vladimir Ivanchenko, Pekka Kaitaniemi, Tatsumi Koi, Mikhail Kosov, Jose Manuel Quesada, et al. Progress in hadronic physics modelling in geant4. In *Journal of Physics: Conference Series*, volume 160, page 012073. IOP Publishing, 2009.
- [ANSa] CFX ANSYS. Help manual 14.0.
- [ANSb] ICEM CFD ANSYS. User’s manual 14.0.
- [Arv91] James Arvo. *Graphics gems II*, volume 2. Morgan Kaufmann, 1991.
- [B<sup>+</sup>03] Forrest B Brown et al. Mcnp—a general monte carlo n-particle transport code, version 5. *Los Alamos National Laboratory, Oak Ridge, TN*, 2003.
- [BCR98] Jacek Bochnak, Michel Coste, and Marie-Françoise Roy. *Real algebraic geometry*. Springer, 1998.
- [BDD<sup>+</sup>13] Emeric Brun, Frederic Damian, Eric Dumonteil, Francois-Xavier Hugot, Yi-Kang Lee, Fausto Malvagi, Alain Mazzolo, Odile Petit, Jean-Christophe Trama, Thierry Visonneau, et al. Tripoli-4 r version 8 user guide. Technical report, CEA Saclay, Direction de l’Energie Nucleaire, Departement de Modelisation des Systemes et Structures, Service d’etudes des Reacteurs et de Mathematiques Appliquees, DEN/DANS/DM2S/SERMA/LTSD, CEA/Saclay, 91191 Gif-sur-Yvette Cedex (France), 2013.
- [BDD<sup>+</sup>14] E Brun, F Damian, CM Diop, E Dumonteil, FX Hugot, C Jouanne, YK Lee, F Malvagi, A Mazzolo, O Petit, et al. Tripoli-4®, cea, edf and areva reference monte carlo code. *Annals of Nuclear Energy*, 2014.
- [BKR09] Steffen Becker, Heiko Koziolk, and Ralf Reussner. The Palladio component model for model-driven performance prediction. 82:3–22, 2009.
- [Bou91] Paul Bourke. Intersection of a plane and a line, 1991. <http://paulbourke.net/geometry/pointlineplane>.
- [BR97] Rene Brun and Fons Rademakers. Root—an object oriented data analysis framework. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 389(1):81–86, 1997.

## BIBLIOGRAPHY

- [BRP09] Harald Breitzkreutz, Anton Rohrmoser, and Winfried Petry. 3-dimensional coupled neutronic and thermohydraulic calculations for a compact core combining mcnp5 and cfx. 2009.
- [CAS12] Open CASCADE. Open cascade technology, 3d modeling and numerical simulation, 2012. <http://www.opencascade.com>.
- [Chi13] Hank Childs. Visit: An end-user tool for visualizing and analyzing very large data. 2013.
- [CKI<sup>+</sup>09] F Cismondi, S Kecskés, M Ilic, G Légrádi, B Kiss, O Bitz, B Dolensky, H Neuberger, LV Boccaccini, and T Ihli. Design update, thermal and fluid dynamic analyses of the eu-hcpb tbm in vertical arrangement. *Fusion Engineering and Design*, 84(2):607–612, 2009.
- [CMPS06] Radovan Chytracsek, Jeremy McCormick, Witold Pokorski, and Giovanni Santin. Geometry description markup language for physics simulation and analysis applications. *Nuclear Science, IEEE Transactions on*, 53(5):2892–2896, 2006.
- [COH<sup>+</sup>06] MB Chadwick, P Obložinský, M Herman, NM Greene, RD McKnight, DL Smith, PG Young, RE MacFarlane, GM Hale, SC Frankle, et al. Endf/b-vii. 0: Next generation evaluated nuclear data library for nuclear science and technology. *Nuclear data sheets*, 107(12):2931–3060, 2006.
- [Col12] Geant Collaboration. Geant4 user’s guide for application developers. *Accessible from the GEANT4 web page [1] Version Geant4*, 9, 2012.
- [CRu11] Jeffrey Neil Cardoni and Rizwan-uddin. Nuclear reactor multi-physics simulations with coupled mcnp5 and star-ccm+. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, 2011.
- [CS95] LL Carter and RA Schwarz. Mcnp visual editor computer code manual. Technical report, Westinghouse Hanford Co., Richland, WA (United States), 1995.
- [DW98] G Dhondt and K Wittig. Calculix: a free software three-dimensional structural finite element program, 1998.
- [Dyk13] Phillip C Dykstra. The brl-cad package: An overview. Technical report, DTIC Document, 2013.
- [FCY99] Mike Folk, Albert Cheng, and Kim Yates. Hdf5: A file format and i/o library for high performance computing applications. In *Proceedings of Supercomputing*, volume 99, 1999.
- [Fis13] U. Fischer. Design tools and methodologies for neutronics studies. Final Report TA WP12-DTM-04, Karlsruhe Institute of Technology, 2013.

- [Flu] ANSYS Fluent. 14.0 user's manual.
- [FRSF05] Alfredo Ferrari, Johannes Ranft, Paola R Sala, and A Fassò. *Fluka*. Number INFN-TC-2005-11. Cern, 2005.
- [GFK<sup>+</sup>13] D Große, U Fischer, K Kondo, D Leichtle, P Pereslavytsev, and A Serikov. Status of the mccad geometry conversion tool and related visualization capabilities for 3d fusion neutronics calculations. *Fusion Engineering and Design*, 88(9):2210–2214, 2013.
- [GHH<sup>+</sup>14] Yixiang Gan, Francisco Hernandez, Dorian Hanaor, Ratna Annabattula, Marc Kamlah, and Pavel Pereslavytsev. Thermal discrete element analysis of eu solid breeder blanket subjected to neutron irradiation. *Fusion Science and Technology*, 66(1):83–90, 2014.
- [Gla89] Andrew S Glassner. *An introduction to ray tracing*. Morgan Kaufmann, 1989.
- [GM99] E Gomin and L Maiorov. The mcu monte carlo code for 3d depletion calculation. In *Proceedings of International Conference*, volume 2, 1999.
- [Gra99] Jeffrey Grandy. Conservative remapping and region overlays by intersecting arbitrary polyhedra. *Journal of Computational Physics*, 148(2):433–466, 1999.
- [HAL04] Amy Henderson, Jim Ahrens, and Charles Law. *The ParaView Guide*. Kitware Clifton Park, NY, 2004.
- [HCK12] Francisco Hernández, Fabio Cismondi, and Béla Kiss. Thermo-mechanical analyses and assessment with respect to the design codes and standards of the hcpb-tbm breeder unit. *Fusion Engineering and Design*, 87(7):1111–1117, 2012.
- [Hec94] Paul S Heckbert. *Graphics gems IV*, volume 4. Morgan Kaufmann, 1994.
- [Hu08] Jianwei Hu. Coupled neutronics and thermal-hydraulics simulations using mcnp and fluent. *TRANSACTIONS-AMERICAN NUCLEAR SOCIETY*, 98:606, 2008.
- [ITE14] ITER. Website, 2014. <http://www.iter.org>.
- [JJT07] Hrvoje Jasak, Aleksandar Jemcov, and Zeljko Tukovic. Openfoam: A c++ library for complex physics simulations. In *International workshop on coupled methods in numerical dynamics*, volume 1000, pages 1–20, 2007.
- [JT70] MA Jenkins and Joseph F Traub. A three-stage algorithm for real polynomials using quadratic iteration. *SIAM Journal on Numerical Analysis*, 7(4):545–566, 1970.

## BIBLIOGRAPHY

- [KB12] S. Kecskés and L. Boccaccini. Blanket test case definition. Final Report WP12-DTM01-T02-D01, EFDA.D.2D44K Q, Karlsruhe Institute of Technology, 2012.
- [Knu64] Donald E. Knuth. Backus normal form vs. backus naur form. *Commun. ACM*, 7(12):735–736, December 1964.
- [LAN11] D Lopez Aldama and R Capote Noy. Fendl-3.0: Processing the evaluated nuclear data library for fusion applications. Technical report, International Atomic Energy Agency, International Nuclear Data Committee, Vienna (Austria), 2011.
- [LBK<sup>+</sup>09] MJ Loughlin, P Batistoni, C Konno, U Fischer, H Lida, L Petrizzi, E Polunovskiy, M Sawan, P Wilson, and Y Wu. Iter nuclear analysis strategy and requirements. *Fusion science and technology*, 56(2):566–572, 2009.
- [Lep12] Jaakko Leppänen. Serpent—a continuous-energy monte carlo reactor physics burnup calculation code. *VTT Technical Research Centre of Finland*, 2012.
- [LFP14] L Lu, U Fischer, and P Pereslavitsev. Improved algorithms and advanced features of the cad to mc conversion tool mccad. *Fusion Engineering and Design*, 89(9):1885–1888, 2014.
- [Lie11] John H Lienhard. *A heat transfer textbook*. Courier Dover Publications, 2011.
- [MAA<sup>+</sup>10] Suziyanti Marjudi, MF Mohd Amran, Khairul Annuar Abdullah, Setyawan Widyarto, NA Abdul Majid, and Riza Sulaiman. A review and comparison of iges and step. In *Proceedings Of World Academy Of Science, Engineering And Technology*, volume 62, pages 1013–1017, 2010.
- [Mar12] RL Martz. The mcnp6 book on unstructured mesh geometry: User guide. *Los Alamos National Laboratory, Los Alamos, NM*, 2012.
- [MCOGC12] E Mendoza, D Cano-Ott, C Guerrero, and R Capote. New evaluated neutron cross section libraries for the geant4 code. *IAEA INDC (NDS)-0612, Tech. Rep.*, 2012.
- [MED14] MED. Med linear remapping, 2014. <http://docs.salome-platform.org>.
- [MWT05] Steven J Manson, Eric K Williams, and Brian P Triggs. Topact: automated translation from cad to combinatorial geometry for radiation transport analysis. In *Collection of Technical Papers—AIAA Space 2005 Conference and Exposition*, volume 3, pages 1571–1579, 2005.
- [Nik04] GP Nikishkov. Introduction to the finite element method. *University of Aizu*, 2004.

- [NSI<sup>+</sup>06] Koji Niita, Tatsuhiko Sato, Hiroshi Iwase, Hiroyuki Nose, Hiroshi Nakashima, and Lembit Sihver. Phits —a particle and heavy ion transport code system. *Radiation measurements*, 41(9):1080–1090, 2006.
- [P<sup>+</sup>05] Denise B Pelowitz et al. Mcnpx user’s manual version 2.5. 0. *Los Alamos National Laboratory*, 76, 2005.
- [PBC<sup>+</sup>00] Diane MA Poirier, Robert H Bush, Raymond R Cosner, CL Rumsey, and DR McCarthy. Advances in the cgns database standard for aerodynamics and cfd. *AIAA Paper*, 681(2000):1, 2000.
- [Per04] Alain Perronnet. Nef: A mesher based on opencascade c.a.d. software, 2004. <http://www.ann.jussieu.fr/perronnet/mit/mit.html>.
- [PLHB00] JM Proix, N Laurent, P Hemon, and G Bertrand. Code aster, manuel de référence. *Fascicule R*, 8, 2000.
- [QtX14] QtXml. Qtxml-c++, 2014. <http://qt-project.org/doc/qt-4.8/qtxml.html>.
- [R&08] EDF R&D. Code saturne 1.3. 2 theory and programmer’s guide. Technical report, Technical report. <http://www.code-saturne.org>, 2008.
- [RCC<sup>+</sup>13] AR Raffray, B Calcagno, P Chappuis, G Dellopoulos, Zhang Fu, Chen Jiming, DH Kim, SW Kim, S Khomiakov, A Labusov, et al. Iter blanket engineering challenges and solutions. In *Fusion Engineering (SOFE), 2013 IEEE 25th Symposium on*, pages 1–8. IEEE, 2013.
- [RFLG07] Peter Råback, Pirjo-Leena Forsström, Mikko Lyly, and Matti Gröhn. Elmer-finite element package for the solution of partial differential equations. In *EGEE User Forum*, 2007.
- [Rom14] P. Romano. Openmc geometry, 2014. <http://mit-crpg.github.io/openmc/methods/geometry>.
- [RP12] I Rupp and C Peniguel. Syrthes 4.0 user manual, 2012.
- [SAH00] William J Schroeder, Lisa Sobierajski Avila, and William Hoffman. Visualizing with vtk: a tutorial. *Computer Graphics and Applications, IEEE*, 20(5):20–27, 2000.
- [Sala] Salome. Med library. <http://www.salome-platform.org/user-section/about/med>.
- [Salb] Salome. Salome platform. <http://www.salome-platform.org>.
- [SBB<sup>+</sup>09] A Santamarina, D Bernard, P Blaise, M Coste, A Courcelle, TD Huynh, C Jouanne, P Leconte, O Litaize, S Mengelle, et al. The jeff-3.1. 1 nuclear data library. *JEFF Report*, 22(10.2):2, 2009.

## BIBLIOGRAPHY

- [Sch] R. Schneider. meshing software. <http://www.robertschneiders.de/meshgeneration/software.html>.
- [Sch97] Joachim Schöberl. Netgen an advancing front 2d/3d-mesh generator based on abstract rules. *Computing and visualization in science*, 1(1):41–52, 1997.
- [She02] J Shewchuk. What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures (preprint). *University of California at Berkeley*, 73, 2002.
- [SOK<sup>+</sup>12] S Sato, K Ochiai, C Konno, H Morota, H Nasif, M Tanaka, E Polunovskiy, and M Loughlin. Nuclear analyses for iter nb system. In *Proceedings of the 24th IAEA Fusion Energy Conference*, October 2012.
- [ST06] Hang Si and A TetGen. A quality tetrahedral mesh generator and three-dimensional delaunay triangulator. *Weierstrass Institute for Applied Analysis and Stochastic*, Berlin, Germany, 2006.
- [Sta10] Weston M Stacey. *Fusion: an introduction to the physics and technology of magnetic confinement fusion*. John Wiley & Sons, 2010.
- [STD<sup>+</sup>07] Volkan Seker, Justin W Thomas, Thomas J Downar, et al. Reactor physics simulations with coupled monte carlo calculation and computational fluid dynamics. In *International Conference on Emerging Nuclear Energy Systems (ICENES-2007)*, Istanbul, TURKEY, page 5, 2007.
- [Str06] Ian Stroud. *Boundary representation modelling techniques*. Springer, 2006.
- [SWSB09] B Smith, P Wilson, M Sawan, and T Bohm. Source profile analysis for the iter first wall/shield module 13. *Fusion science and technology*, 56(1):57–61, 2009.
- [Tau00] Timothy J Tautges. The common geometry module (cgm): A generic, extensible geometry interface. In *IMR*, pages 337–348. Citeseer, 2000.
- [Tav02] A-AF Tavassoli. Present limits and improvements of structural materials for fusion reactors—a review. *Journal of Nuclear Materials*, 302(2):73–88, 2002.
- [TGS<sup>+</sup>91] K Tomabechi, JR Gilleland, Yu A Sokolov, R Toschi, ITER Team, et al. Iter conceptual design. *Nuclear Fusion*, 31(6):1135, 1991.
- [TMM<sup>+</sup>04] Timothy J Tautges, Ray Meyers, Karl Merkley, Clint Stimpson, and Corey Ernst. Moab: a mesh-oriented database. Technical report, Sandia National Laboratories, 2004.
- [TSW10] Joe F Thompson, Bharat K Soni, and Nigel P Weatherill. *Handbook of grid generation*. CRC press, 2010.

- [TT01] H Tsige-Tamirat. On the use of cad geometry for monte carlo particle transport. In *Advanced Monte Carlo for Radiation Physics, Particle Transport Simulation and Applications*, pages 511–516. Springer, 2001.
- [TWK<sup>+</sup>09] Timothy J Tautges, Paul PH Wilson, J Kraftcheck, BF Smith, and Douglass L Henderson. Acceleration techniques for direct use of cad-based geometries in monte carlo radiation transport. In *International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2009)*, 2009.
- [Vin97] Steve Vinoski. Corba: Integrating diverse applications within distributed heterogeneous environments. *Communications Magazine, IEEE*, 35(2):46–55, 1997.
- [Wen08] John Wendt. *Computational fluid dynamics: an introduction*. Springer, 2008.
- [WFF<sup>+</sup>08] PPH Wilson, RUSSEL Feder, U Fischer, M Loughlin, L Petrizzi, Y Wu, and M Youssef. State-of-the-art 3-d radiation transport methods for fusion energy systems. *Fusion Engineering and Design*, 83(7):824–833, 2008.
- [Wika] Wiki. Cad software. [http://en.wikipedia.org/wiki/Comparison\\_of\\_computer-aided\\_design\\_editors](http://en.wikipedia.org/wiki/Comparison_of_computer-aided_design_editors).
- [Wikb] Wiki. Grid classification. [http://en.wikipedia.org/wiki/Grid\\_classification](http://en.wikipedia.org/wiki/Grid_classification).
- [Wic] Wikipedia. Cad software. [https://en.wikipedia.org/wiki/Comparison\\_of\\_computer-aided\\_design\\_editors](https://en.wikipedia.org/wiki/Comparison_of_computer-aided_design_editors).
- [Wid] Wikipedia. Constructive solid geometry. [https://en.wikipedia.org/wiki/Constructive\\_solid\\_geometry](https://en.wikipedia.org/wiki/Constructive_solid_geometry).
- [WMM97] Todd A Wareing, John M McGhee, and Jim E Morel. Attila. a 3-d unstructured tetrahedral-mesh s n code. In *3-D deterministic radiation transport computer programs. Features, applications and perspectives*. 1997.
- [WSGB04] GA Wright, E Shuttleworth, MJ Grimstone, and AJ Bird. The status of the general radiation transport code mcbend. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 213:162–166, 2004.
- [Wu09] Yican Wu. Cad-based interface programs for fusion neutron transport simulation. *Fusion Engineering and Design*, 84(7):1987–1992, 2009.
- [Xer10] Xerces. Xerces-c++, 2010. <http://xerces.apache.org>.
- [YNA<sup>+</sup>09] A Ying, M Narula, M Abdou, R Munipalli, M Ulrickson, and P Wilson. Toward an integrated simulation predictive capability for fusion plasma chamber systems. *Fusion science and technology*, 56(2):918–924, 2009.

## BIBLIOGRAPHY

- [Zik10] Oleg Zikanov. *Essential computational fluid dynamics*. John Wiley & Sons, 2010.

# Acronyms

<b>1D/2D/3D</b>	One-/Two-/Three-dimensional
<b>ANSYS</b>	ANSYS engineering simulation software
<b>API</b>	Application Programming Interface
<b>AIS</b>	Application Interactive Services
<b>B-Rep</b>	Boundary Representation
<b>BU</b>	Breeder Unit
<b>CAD</b>	Computer Aided Design
<b>CAE</b>	Computer Aided Engineering
<b>CFD</b>	Computational Fluid Dynamics
<b>CFX</b>	Computational fluid dynamics simulation software
<b>CGNS</b>	CFD General Notation System
<b>CORBA</b>	Common Object Request Broker Architecture
<b>CP</b>	Cooling Plates
<b>CSG</b>	Constructive Solid Geometry
<b>CSV</b>	Comma-separated Value
<b>DEMO</b>	DEMONstration Power Plant
<b>EEOUT</b>	Elemental edit output
<b>ESS</b>	European Spallation Source
<b>FE</b>	Finite Element
<b>FEM</b>	Finite Element Method
<b>FLUENT</b>	Computational fluid dynamics simulation software
<b>FVM</b>	Finite Volume Method
<b>FW</b>	First Wall
<b>GDML</b>	Geometry Description Markup Language

## Acronyms

<b>Geant4</b>	A toolkit for the simulation of the passage of particles through matter, developed in CERN, Switzerland
<b>GEOM</b>	SALOME platform CAD modeling module
<b>GUI</b>	Graphical user interface
<b>HCPB</b>	Helium Cooled Pebble Bed
<b>HDF</b>	Hierarchical Data Format
<b>IDL</b>	Interface Definition Language
<b>IFMIF</b>	International Fusion Materials Irradiation Facility
<b>IGES</b>	Initial Graphics Exchange Specification format
<b>ITER</b>	International Thermonuclear Experimental Reactor
<b>MC</b>	Monte Carlo
<b>McCad</b>	Monte Carlo geometry conversion tool
<b>McMeshTran</b>	a MC Mesh and data Transformation/Translation/Transfer tool
<b>MCNP</b>	Monte Carlo N Particles
<b>MED</b>	MED mesh library
<b>NWL</b>	Neutron wall loading
<b>OSI</b>	lithium orthosilicate
<b>ParaView</b>	An open-source visualization software
<b>PARAVIS</b>	ParaView module in SALOME platform
<b>RD</b>	Relative Deflection
<b>SALOME</b>	SALOME open-source simulation platform
<b>SM</b>	Structural Mechanics
<b>SMESH</b>	SALOME platform mesh generation module
<b>STEP</b>	Standard format for the Exchange of Product model data
<b>STL</b>	STereoLithography CAD format
<b>TBM</b>	Test Blanket Module
<b>TFC</b>	Toroidal Field Coils
<b>TH</b>	Thermal hydraulics
<b>TMA</b>	Thermomechanical analysis
<b>TRIPOLI</b>	A 3D general purpose continuous energy Monte Carlo Transport code developed in CEA, France

<b>TT</b>	Tessellation-Tetrahedralization
<b>UDF</b>	User Defined Functions
<b>VTK</b>	Visualization ToolKit