

TALLINN UNIVERSITY OF TECHNOLOGY  
School of Science

Krislin Sartakov 212052YAFM

**RADIATION TRANSPORT SIMULATIONS IN A HUMAN  
PHANTOM**

Master's Thesis

Supervisor: Marti Jeltsov  
PhD

Tallinn 2024

TALLINNA TEHNIKAÜLIKOOL  
Loodusteaduskond

Krislin Sartakov 212052YAFM

# **KIIRITUS SIMULATSIOONID INIMFANTOOMKEHAS**

Magistritöö

Juhendaja: Marti Jeltsov  
PhD

Tallinn 2024

## **Author's Declaration of Originality**

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Krislin Sartakov

01.01.2024

The thesis complies with the requirements for Master's theses.

Supervisor: Marti Jeltsov

01.01.2024

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>9</b>
<b>2</b>	<b>RADIATION THERAPY</b>	<b>10</b>
2.1	External beam therapy	11
2.2	Brachytherapy	11
<b>3</b>	<b>MONTE CARLO METHOD</b>	<b>13</b>
<b>4</b>	<b>METHODOLOGY</b>	<b>14</b>
4.1	Meshing	14
4.2	OpenMC	17
4.2.1	Materials	18
4.2.2	Geometry	19
4.2.3	Settings	19
4.2.4	Tallies	20
4.2.5	Cross sections	21
<b>5</b>	<b>SIMULATIONS</b>	<b>22</b>
5.1	Adult male phantom	22
5.1.1	Source site: Heart	24
5.1.2	Source site: Liver	24
5.1.3	Source site: Stomach	25
5.1.4	Source site: Spleen	25
5.1.5	Source site: Bladder	25
5.1.6	Target: Teeth	26
5.1.7	Source site: Heart with Cs-137	26
5.1.8	Source site: Mandible	27
5.1.9	Source placed outside the body	28
5.2	Discussion	28
5.2.1	Results	29
5.2.2	Future direction	33
<b>6</b>	<b>Summary</b>	<b>35</b>
<b>7</b>	<b>Acknowledgements</b>	<b>36</b>

<b>References</b> . . . . .	<b>37</b>
<b>Appendix 1 – Non-Exclusive License for Reproduction and Publication of a Graduation Thesis</b> . . . . .	<b>40</b>
<b>Appendix 2 – OpenMC code</b> . . . . .	<b>41</b>

## Abstract

The the workflow of Cubit Coreform and OpenMC were introduced. With these softwares a simplified adult male phantom was prepared and used in Monte Carlo particle transport code to simulate gamma radiation transport. The phantom consists of 234 volumes, 370 surfaces and 47 material compositions.

Different theoretical situations were created and their corresponding dose rates compared. The source was mostly placed at the torso and the dose rate received by the brain was viewed. Once the source was placed at the pelvic region, the particles couldn't reach the brain.

The simulation results showed that the distances between the target and the source site, the material composition of the volumes and the isotope used all influenced the dose rate received by the target. An identical simulation with Co-60 and Cs-137 was created. The Cs-137 isotope with smaller energy value did produce a smaller dose rate at the target site. Also two situations were created to compare the difference of dose rates in situations where the isotope is either placed inside or outside of the phantom. One source site was located at the mandible of the phantom and other was located outside of the phantom. The results showed that the dose rate received from an external source is half of what the phantom receives internally.

The thesis is written in English and is 40 pages long, including 7 chapters, 18 figures and 2 tables.

# **Annotatsioon**

## **Kiiritus simulatsioonid inimfantomkehas**

Töö käigus tutvustati lugejale Cubit Coreform ja OpenMC tarkvarasid, millega valmistati ette lihtsustatud inimfantomkeha ja viidi läbi kiiritus simulatsioonid. Inimfantom koosneb 234 kehast, 370 pinnast ning 47 materjalist.

Võrdlemiseks loodi erinevad teoreetilised olukorrad, kus enamjaolt paigutati kiirgusallikas inimfantoomi kõhuõõnde ning võrreldi aju poolt saadud doosikiiruseid. Simulatsioonid näitasid, et kui allikas viia ajust kaugemale vaagnapiirkonda, siis vaadeldavasse organisse enam osakesi ei jõua.

Tulemustes selgus, et doosikiirus on mõjutatud allika ja vaadeldava organi vahelisest kaugusest, ümbritsevatest materjalidest ning kasutatud isotoobist. Läbi viidi identsed simulatsioonid nii Co-60 kui ka Cs-137 isotoobiga, mille käigus oli näha, et väiksema energiaga Cs-137 tekitas väiksema doosikiiruse. Samuti viidi läbi kaks sarnast simulatsiooni, kus ühes paigutati kiirgusallikas inimfantoomi ning teises viidi allikas fantoomist välja. Antud simulatsioonide tulemused näitasid, et olukorras kus kiirgusallikas asetseb väljaspool fantoomi, on saadud doosikiirused poole väiksemad.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 40 leheküljel, 7 peatükki, 18 joonist, 2 tabelit.

## List of Abbreviations and Terms

AM	Adult male
CAD	Computer-aided design
CSG	Constructive solid geometry
DAGMC	Direct Accelerated Geometry Monte Carlo
ICRP	International Commission on Radiological Protection
MC	Monte Carlo

## List of Figures

1	Difference between healthy cell division and cancerous cell division [3]. . .	10
2	Brachytherapy in prostate cancer [8]. . . . .	12
3	TriMesh surface on a skull. . . . .	15
4	TetMesh of a brain. . . . .	16
5	Adult male human phantom with triangle elements. . . . .	16
6	Adult male human phantom with transparent skin. . . . .	23
7	Adult male human phantom cross section. . . . .	23
8	Interaction points in heart to brain simulation. . . . .	24
9	Particle tracks in spleen to brain simulation. . . . .	25
10	Interaction points in bladder to brain simulation. . . . .	26
11	Interaction points with Cs-137 source. . . . .	27
12	Particle tracks in mandible to brain simulation. . . . .	27
13	Particle tracks when the source is placed outside of the phantom. . . . .	28
14	Dose rates in different organs. . . . .	29
15	Dose rate difference between the brain and teeth. . . . .	30
16	Dose rates with different isotopes. . . . .	31
17	Dose rates between inside and outside source placement. . . . .	32
18	Dose assessment study results with Co-60 source [28]. . . . .	33

## List of Tables

1	Elements provided by ICRP [20]. . . . .	18
2	Simulation Results and Dose Rates . . . . .	29

# 1. Introduction

With the rising capacity of computer calculations the application of radiation transport simulations is increasing. Today highly complex models are used to carry out simulations in either nuclear reactors, in the surrounding environment or in human phantoms. There are a multitude of Monte Carlo based particle transport codes available, for example GEANT4, MCNP and OpenMC.

With the growing number of cancer patients each year different approaches to radiation therapy are studied [1]. The Monte Carlo particle transport codes offer a way to gain insight into the treatment process. With cancer treatment simulations radiation transport and storage in different parts of the human body can be studied. This offers the physicians a way to approach the possible reduction of damage to nearby healthy cells.

In this thesis the author uses a simplified human phantom provided by International Commission on Radiological Protection and OpenMC, a Monte Carlo code developed and mostly used for reactor simulations, to build and carry out radiation transport simulations. According to the developers of OpenMC and the employees of Cubit Coreform, neither of the softwares have been used previously in an attempt to simulate radiation transport inside a human. The main goal of the thesis was to successfully prepare a human phantom geometry for OpenMC. Once the phantom was successfully integrated into the simulation, the secondary goal was to carry out radiation transport simulations in order to see how the dose rates change depending on the source placement and isotope.

## 2. RADIATION THERAPY

In radiation therapy ionizing radiation is used to treat cancer. The goal is to administer maximum amount of radiation needed to kill the cancerous cells but spare the environment surrounding it. Tumors are the result of abnormal cellular behaviour. During mitosis one mature cell divides into daughter cells. In a normal biological process these daughter cells continue to divide until a mature cell with a specific purpose is achieved. But in case of cancer the daughter cells keep on dividing with no mature cell with purpose ever resurfacing. [2]

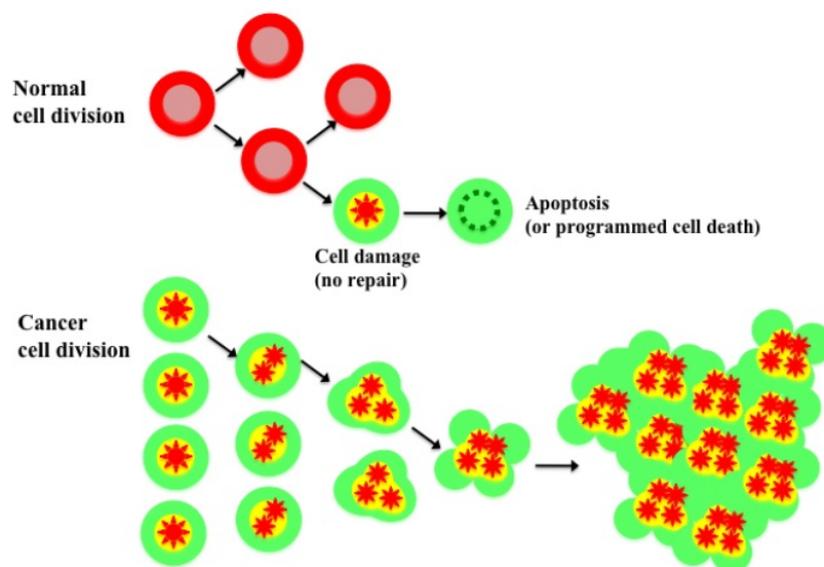


Figure 1. Difference between healthy cell division and cancerous cell division [3].

The cause of cancer stems from mutations in the genes. In 2015 in USA alone approximately 665 500 new cancer cases are registered every year. Approximately 585 700 of those patients die. [2]

Radiation therapy is one of many methods to treat cancer. Radiation is not always the optimal choice and largely depends on the patients condition. Historically radiation therapy could not be used if the cancer had spread across the patient body but today with oligometastatic cancer treatment, up to five metastasis can be treated [4]. [2]

Sometimes radiation may be more effective than surgery. An example of this is laryngeal cancer. The cancerous cells develop in the tissue of larynx, which resides in the throat between the base of the tongue and trachea, near the vocal cords. With surgery the vocal

cords would get removed while with radiation therapy they can be saved. Another example is prostate cancer. While surgery might leave the patient impotent, radiation therapy does no harm to the patients sexual organs. Radiation therapy is sometimes also used as a pain relief. If the tumor is pressing on a nerve and causing the patient pain, radiation can be used to shrink the cancer. [2]

There are two types of radiation therapy: external beam therapy and brachytherapy.

## **2.1 External beam therapy**

External beam therapy uses external beams to treat cancer. Three different types of particles are mostly used:

- photons;
- protons;
- electrons.

Most common photon sources in radiation therapy are X-ray machines, radioisotope sources or a linear particle accelerator. [5] High-energy X-rays are used to treat tumours deep inside the patients body [2]. One disadvantage of using photons in external beam therapy is the fact that photons produce divergent beams which not only affect the tumor but also the neighbouring healthy tissues. Once the photon beam reaches its target, it continues moving through the patients body until it exits on the opposite side. [5] The traveling photon beam is governed by inverse square law, scattering and attenuation, which make the dose estimation a complicated task. A direct measurement of dose distribution inside the body is almost impossible yet the knowledge about doses in the patient are crucial for physicians. [6]

Proton beams however do not scatter and they stop once they have reached their target. This offers a way to treat the cancer while not harming the surrounding healthy tissues. [5] Electron beams are used to treat cancers closer to the surface of the skin. The electron beams do not carry enough energy to reach deeper layers of body. [5]

## **2.2 Brachytherapy**

With brachytherapy a radioactive nuclide is placed inside the patient. The source is placed either in close proximity to the tumor or straight inside it [7].

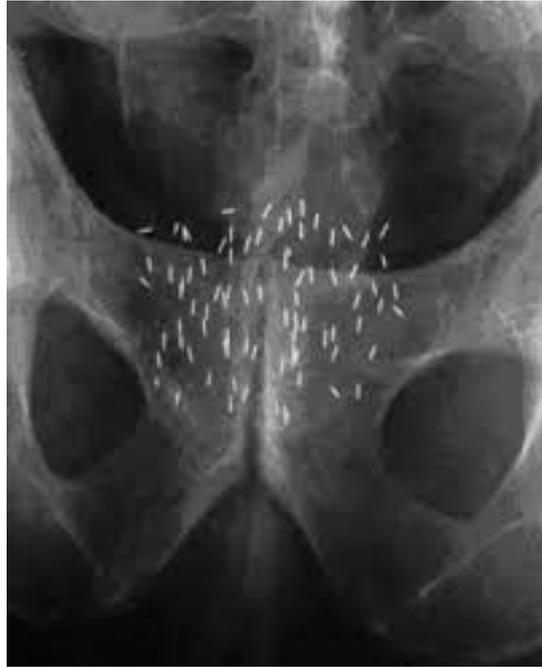


Figure 2. Brachytherapy in prostate cancer [8].

Common nuclides used are Cobalt-60, Iridium-192, Ruthenium-106 and Iodine-125. This method offers a chance to deal high levels of radiation to the tumor while sparing the healthy tissue surrounding it. The source can either be left inside or removed after a sufficient amount of dose is administered to the patient. [2] Temporary implants can be inside the tumor starting from 10 minutes up to 7 days, depending on the strength of the source. While permanent implants are placed inside the tumor and stay there until the source decays. [7] [9]

### 3. MONTE CARLO METHOD

The Monte Carlo (MC) method is a collection of computational techniques which are used to solve mathematical problems through random sampling [10]. It is a powerful statistical method developed at the Los Alamos National Laboratories during the age of first atomic bombs [11]. In essence a distribution of stochastic input values must be defined that incorporate random variability into the model. From there the computer repeatedly runs the simulation with different sample distributions and the result is a set of possible outcomes. MC method is useful in problems where no closed form solution exists. [12] One example of that is the linear Boltzmann equation:

$$\Phi(P) = S(P) + \int_{\Omega} K(P' \rightarrow P)\Phi(P')dP', \quad (3.1)$$

where  $P$  represents particle position, direction and energy,  $S$  is the source,  $\Phi(P)$  is density of radiation at  $P$  from the source and from contributions scattered into  $P$  from all  $P'$ .  $K$  is the scattering kernel and the domain of integration  $\Omega$  consists of all positions, energies and directions. [13]

The linear Boltzmann equation describes the accurate transport of radiation through matter and it is notoriously difficult to solve analytically. Thus the MC method serves its advantage by offering a way to solve the equation without needing expert input from the user. [13] [14] With MC simulations the life of a particle is simulated. Starting from its emission to the final interaction with the surrounding material. It simulates the random trajectories of individual particles through a bulk of material, where random number generators are used to select the step lengths and the interaction type. The interaction type is determined by the probability distribution of the materials cross section. [15] Once the interactions and different outcomes are repeated for a large amount of particles, the result is a detailed simulation of particle transport process in a defined geometry. [16] At each interaction site the energy transfer to the material is stored and different output values can be calculated, such as particle flux [17].

The use of MC method in dosimetry has substantially increased. Its usage ranges from calculating dosimetric quantities for simulations to radiotherapy treatment planning. [18] MC method is used routinely in clinical applications where it accurately and efficiently predicts particle interactions with tissues. Together with the patients computed tomography scans it is used for dose calculations. Although today most radiation treatment is done with photons, the real future of MC in medicine lies with proton treatment planning and dose assessment. [19]

## 4. METHODOLOGY

The human phantom used in this work was acquired from International Commission on Radiological Protection (ICRP) Publication nr. 145: "Adult Mesh Type Reference Computational Phantoms" [20]. The phantom is a mathematical representation of human anatomy. It was specifically created for simulating dosimetry, for both external and internal exposure to ionizing radiation.

The computer-aided design (CAD) files were meshed using Cubit Coreform, employing its academic license, and the simulations were developed and executed using OpenMC.

Cubit Coreform is a computer-aided software developed for hexahedral meshing for simulation model preparation [21]. It specialises in robust 2D and 3D finite element meshes with the goal of reducing computation time [22].

OpenMC is an open source Monte Carlo neutron and photon transport code developed for nuclear reactor simulations. It simulates photons and neutrons stochastically moving through predefined models that aim to represent a real-world experiment. [23]

The results of the simulations were compared to a dose assessment study done in 2014 where gamma radiation exposure was simulated to workers based on a 2005 radiation accident at Meet Halfa, Egypt.

### 4.1 Meshing

The adult male (AM) CAD model was imported into Cubit Coreform. The model was cleaned up and simplified for computational and meshing purposes. Volumes featuring extensive overlaps with multiple others were excluded, such as the blood vessels, bone marrow, select bones, and the muscle layer. Additionally, various contents, including urine and air, were removed from within the organs. If an organ contained multiple volumes, they were all combined together.

Once all the overlaps were fixed and necessary volumes removed the materials were defined. In order to assign a material name to a certain volume it first needs to be grouped. This allows the user to assign materials simultaneously to multiple different volumes. The material names assigned in Cubit Coreform must be identical to the ones defined in OpenMC materials file. Then the boundary conditions must be applied to all external volumes. The last step is to create a graveyard volume around the desired geometry. The graveyard region serves as a barrier between the simulation and real-world. After a particle crosses that barrier, it ceases to exist. [24]

The last step before the model is ready to be exported is meshing. TriMesh was used to

mesh the surfaces and TetMesh was used for volumes. The TriMesh function fills the surface with triangle shaped elements (figure 3). TriMesh method is commonly used as the default approach for generating TetMesh in volumes. [22] TriMesh and TetMesh methods were used due to the fact that direct accelerated geometry monte carlo (DAGMC) uses triangle meshes to track particle movements. [25]

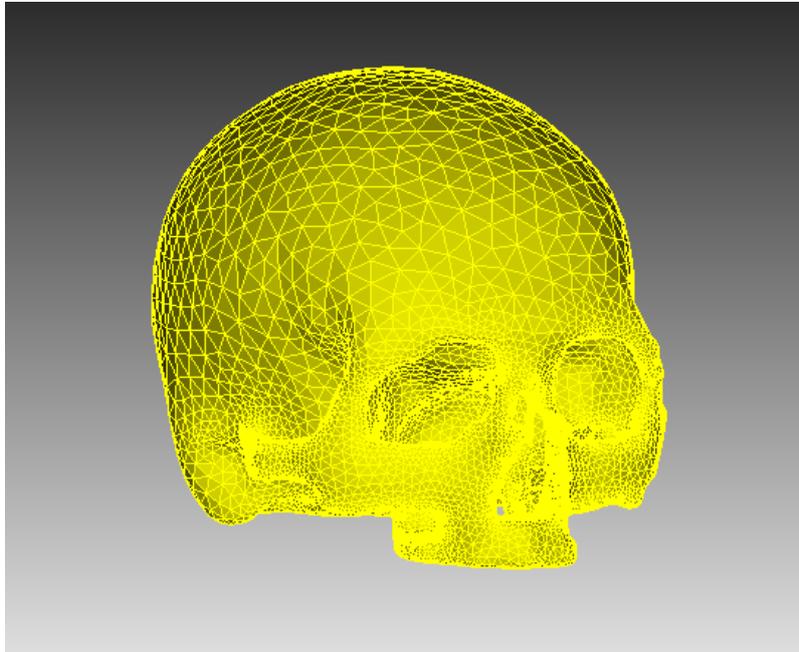


Figure 3. TriMesh surface on a skull.

After the triangles were created for the surface, the three dimensional volume were filled with tetrahedral elements (figure 4).

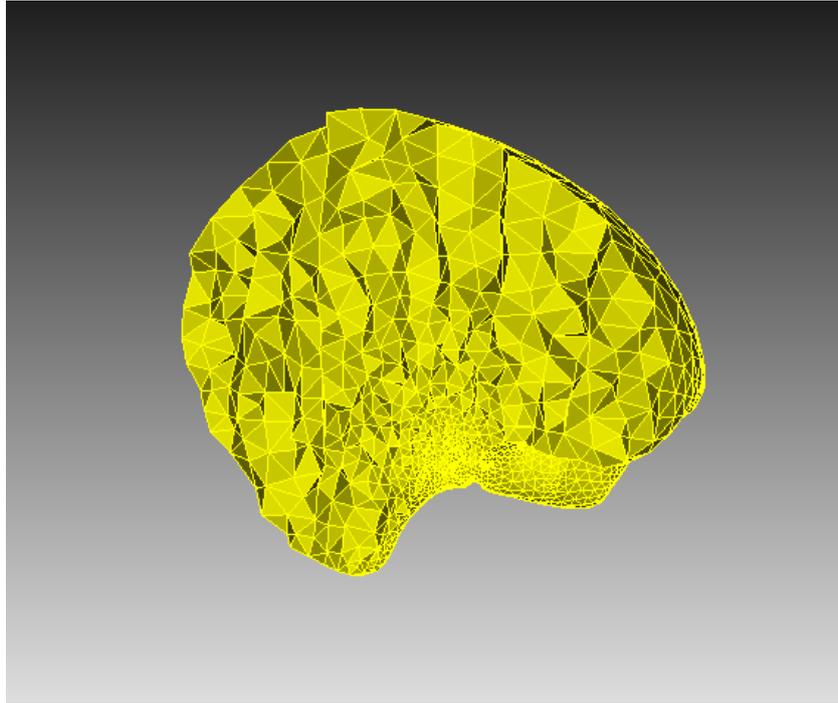


Figure 4. TetMesh of a brain.

The surface representation of the AM model can be seen in figure 5.

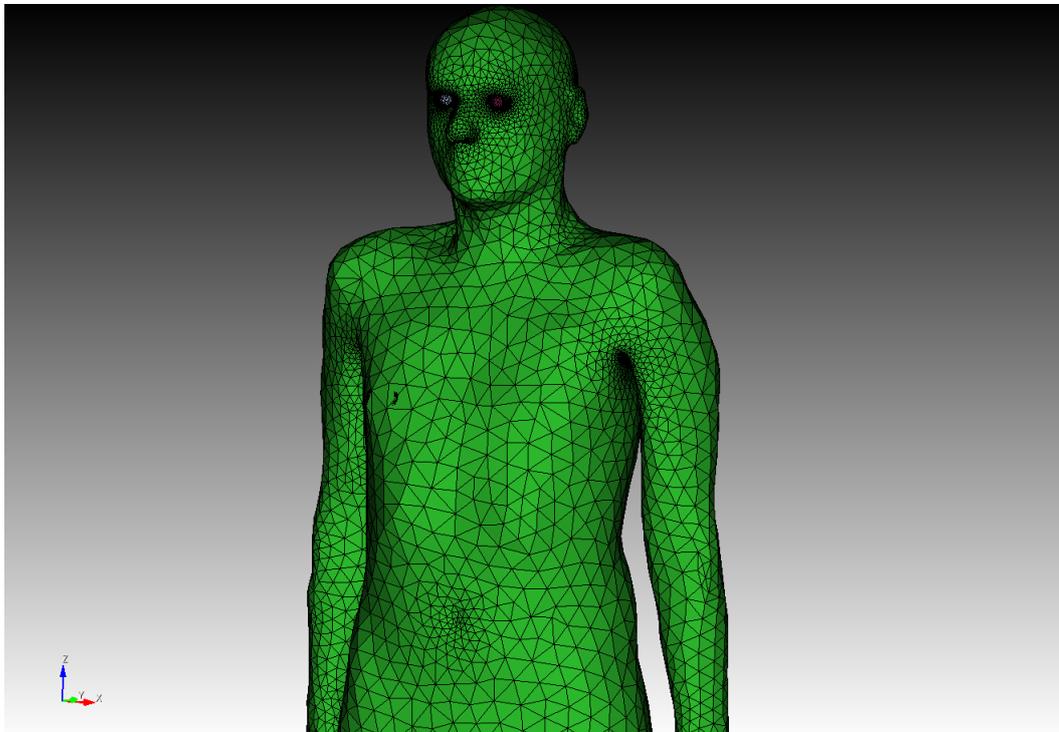


Figure 5. Adult male human phantom with triangle elements.

## 4.2 OpenMC

OpenMC needs several input files to run the simulations successfully. Those files are: material.xml, geomtery.xml, settings.xml and tallies.xml.

Once the input files have been created the simulation has to be initialized. It will read the input files, generate data structures and necessary variables. The pseudo-random number generator is automatically initialized, cross-section data can either be linked at the building phase of the software or entered manually by the user and the source site must be specified. After these processes, the simulation is ready to run.

The particle transport simulation tracks one particle at a time. The simulation starts by determining in which cell the particle resides at. This is determined by sampling the source site defined during the initialization of the program. Then the cross sections and the distance to the nearest boundary of the particle are determined. From which the simulation calculates the distance to the next collision. The distance can be found with:

$$d = -\frac{\ln\xi}{\sum_t}, \quad (4.1)$$

where  $\xi$  is a pseudo-random number sampled from uniform distribution on  $[0,1)$  and  $\sum_t$  is the materials cross section.

Now there are two possible outcomes depending on which is closer: distance to the collision site or distance to the boundary. If the distance to the collision site is closer then the particle will undergo the collision. If the distance to the boundary is closer then the particle is moved closer to the boundary which initiates the process anew and the initial cell is determined again.

Once the particle has moved to the boundary the probability of collision with any nuclide of the boundary material is calculated with:

$$P(i) = \frac{\sum_{t,i}}{\sum_t}, \quad (4.2)$$

where  $\sum_{i,t}$  is the total cross section of the material  $i$ .

Once the nuclide undergoing collision is identified, the microscopic cross section  $\sigma_x$  determines the reaction type  $x$ . The probability of reaction  $x$  happening is:

$$P(x) = \frac{\sigma_x}{\sigma_t}, \quad (4.3)$$

where  $\sigma_t$  is the total microscopic cross section of the nuclide. [26]

The three different types of reactions the particle can have with the nuclide is either scattering, fission or absorption. If the reaction is scattering then the outgoing energy and angle is sampled from an appropriate distribution. If it's absorption or fission then the

particle dies and necessary fission sites are created for the fission bank.

After all the particles have been run through the simulation the program calculates and writes statistical data about the predefined tallies. [26]

### 4.2.1 Materials

The materials defined in OpenMC must be identical to the ones set up previously in Cubit Coreform. This system allows OpenMC to find the right geometries where the simulation is set up to run in. The human phantom material composition was provided in the ICRP nr. 145 documentation. The elements provided by ICRP can be seen in table 1.

Nr	Element	Nr	Element
1	Hydrogen	8	Iodine
2	Carbon	9	Iron
3	Nitrogen	10	Calcium
4	Oxygen	11	Potassium
5	Sodium	12	Chlorine
6	Magnesium	13	Sulfur
7	Phosphorus		

Table 1. Elements provided by ICRP [20].

The materials in OpenMC must be defined as a set of elements and their corresponding atomic fractions. The atomic fractions are re-normalized based on the total density of the material. [26] For example the material for spleen is set up accordingly:

```
spleen=openmc.Material(name='spleen')
spleen.add_element('H',10.3)
spleen.add_element('C',11.2)
spleen.add_element('N',3.2)
spleen.add_element('O',74.3)
spleen.add_element('Na',0.1)
spleen.add_element('P',0.2)
spleen.add_element('S',0.2)
spleen.add_element('Cl',0.2)
spleen.add_element('K',0.3)
spleen.set_density('g/cm3',1.060)
spleen.id=3
```

where on the first line the name is defined. The following lines add necessary elements

and their atomic fractions. The second to last line sets the density of the material in  $\frac{g}{cm^3}$  and the last line specifies the corresponding material ID found in Cubit Coreform.

## 4.2.2 Geometry

In OpenMC the users have the option to either create their own constructive solid geometry (CSG) which allows to build arbitrarily complex models in Euclidean space or work with CAD models through DAGMC. [26] The preparation of CAD models for use in OpenMC with DAGMC requires prior processing in Cubit Coreform, as detailed earlier in this chapter.

In the generic workflow the volumes and surfaces need to be grouped and meshed, material names must correlate with OpenMC and users must establish boundary conditions for the volumes. In the absence of specific conditions, surfaces are automatically assigned a transmission boundary condition. Users have the flexibility to set boundary conditions as either vacuum, transmission or reflecting based on simulation requirements. Also temperatures can be set for desired volumes. [24]

Additionally OpenMC support universe based geometries. Users can define their geometry as a universe rather than a material. These universes can then be multiplied, proving particularly advantageous when the model comprises of multiple identical parts, such as a nuclear reactor fuel pin assembly. But with DAGMC each geometry has to be defined as a DAGMC Universe as a reference that a DAGMC file is being used in the model. DAGMC Universes can then be used to fill CSG cells if desired. [26]

## 4.2.3 Settings

The third xml file needed for the simulation is the execution settings file. In this file different choices have to be made in order to represent as truly as possible the real life situation which the user wishes to simulate. To start the run mode has to be set. The user can choose either between eigenvalue, fixed source, volume, plot or particle restart [23].

1. The eigenvalue mode does criticality calculations. This run mode is used when the source of neutrons includes a fissionable material. It's usually used for simulations of nuclear reactors, nuclear bombs, spent fuel pools etc. [26]
2. The fixed source run mode uses a specified external source to run the simulations. This run mode was used in this thesis.
3. The volume run mode can be used to calculate the size of specified volumes in the geometry. This run mode is useful to check for any irregularities in the geometry. The user must define a bounding box surrounding the volume of interests in which

the simulation will sample points within the box to find what fraction of the points were present in the desired domain of calculation.

4. The plot mode allows the user to visualize the geometry in either two dimensional or three dimensional plots.
5. The particle restart mode uses the particle restart file to only simulate one single source particle. [23]

In the fixed source run mode the total number of particles is divided into user defined batches, in the official documentation it is recommended to use at least 30 batches to satisfy the central limit theorem [23]. 200 batches was used in this thesis.

The particle count per generation defined by user is also needed to be set for the setting file. The more particles present per generation, the smaller will be the stochastic uncertainty of the simulation results. For this thesis 35 000 particles was used in each batch, the choice was made based on the advice of other OpenMC users and the computational power and time available.

The definition of the source encompasses specifying the energy distribution, angular distribution, and particle type. For this thesis the angular distribution is uniformly set as isotropic, while the energy distribution depends on the source element. Photon particles are chosen to simulate gamma radiation. The energy distribution is set up as a probability mass function. For example, Co-60 energy distribution is set as:

```
source.energy = openmc.stats.Discrete([1.1732e6, 1.3325e6],
                                       [0.5, 0.5])
```

where  $1.1732 \cdot 10^6$  and  $1.3325 \cdot 10^6$  are energy values in eV and 0.5 are the probabilities of the particle having each energy value.

#### 4.2.4 Tallies

Tallies in OpenMC represent an estimation of a mean of some random variable in interest. The user can apply different filters in order to specify in which region of the geometry a predefined score should be obtained from. An example of possible filters are: universe, cell, material, surface and mesh. An example of different scores are: flux, total reaction rate, surface currents and neutron production from scattering. A tally can be written in a following form:

$$X = \int dr \int d\Omega \int dE f(r, \Omega, E) \psi(r, \Omega, E), \quad (4.4)$$

where  $\int dr \int d\Omega \int dE$  are the filters and  $f(r, \Omega, E)$  are the scores. [26]

In this work, the material, particle and energy function filter was used to score the value of flux.

#### **4.2.5 Cross sections**

In nuclear physics cross sections represent the probability of a certain reaction occurring between two particles for a given set of parameters. There are three important types of cross sections: absorption, scattering and fission. Cross sections can also be either macroscopic or microscopic. Microscopic cross sections describe the probability of interaction between a particle and a single specific atom. Macroscopic cross sections describe the probability of interaction between a particle and a bulk material that consists of multiple atoms, for example water, concrete, steel etc. [27]

In OpenMC the user can use pregenerated cross section libraries that have been processed into HDF5 format [23]. These libraries have to be linked during the installation process of the software.

## 5. SIMULATIONS

Different theoretical situations were created, where the AM phantom was exposed to radiation in different source locations. The source was placed inside soft tissues, bones, outside of the body and in different distances from the target volume. Most of the simulations were done with Co-60 source, one simulation was done with Cs-137 to compare the dose rates in identical simulations. Activity of 20 Ci was used for both sources [28].

All simulations were carried out with 35 000 particles per batch and the simulation consists of 200 batches. The source is used as a point source and the particles simulated are photons. The target tally or score, specified for OpenMC to overwrite is the flux of particles at the target volume.

Once the simulation has finished and the tallies have been written, the flux score is converted into  $\mu Sv/hr$ . OpenMC writes the flux tally in units  $pSv \frac{1}{cm^3}$  per source particle. In order to convert it to dose rate we have to multiply the flux tally by the number of particles per second and divide it by the desired phantom volume.

The information about the primary and secondary particle tracks were imported after the simulation. Both particle tracks and their interaction points were visualised in ParaView.

### 5.1 Adult male phantom

The AM phantom used in this thesis is a rudimentary model of human anatomy. All of the results in this thesis in no way correspond to real-life treatment results since the phantom used is simplified, lacks any hemodynamic processes and has no cell life cycles included. The AM phantom consists of 234 volumes, 370 surfaces and stand around 176 cm tall. The volumes are grouped between 47 material groups. A transparent view of the phantom can be seen in figure 6 and its cross section in figure 7.

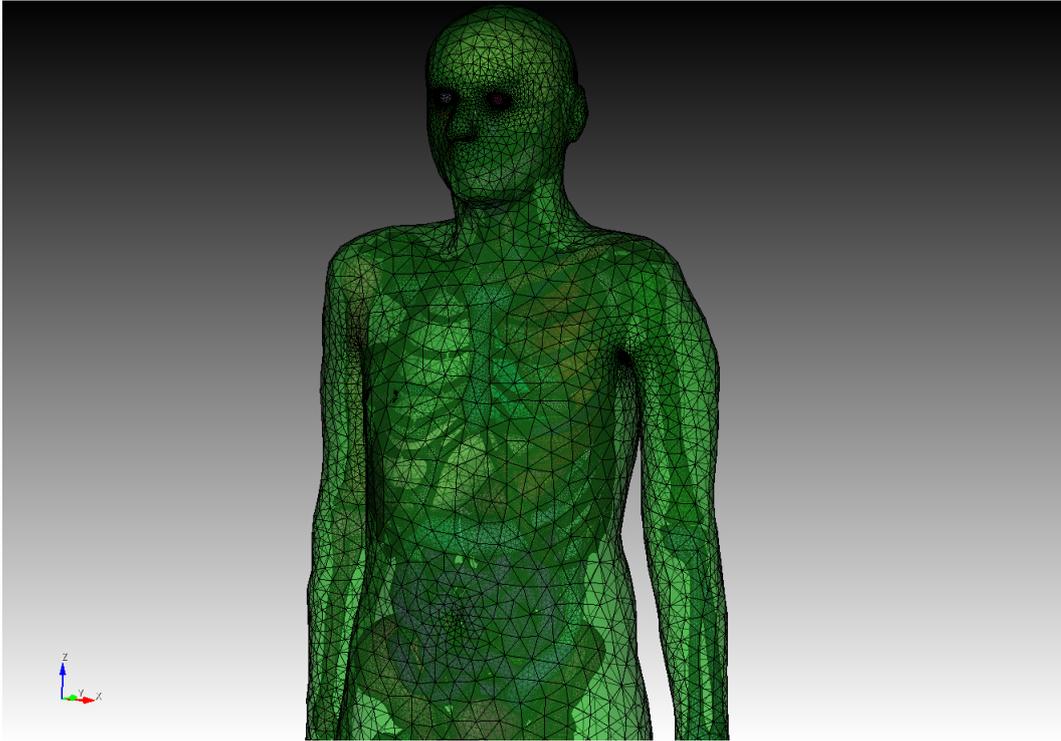


Figure 6. Adult male human phantom with transparent skin.

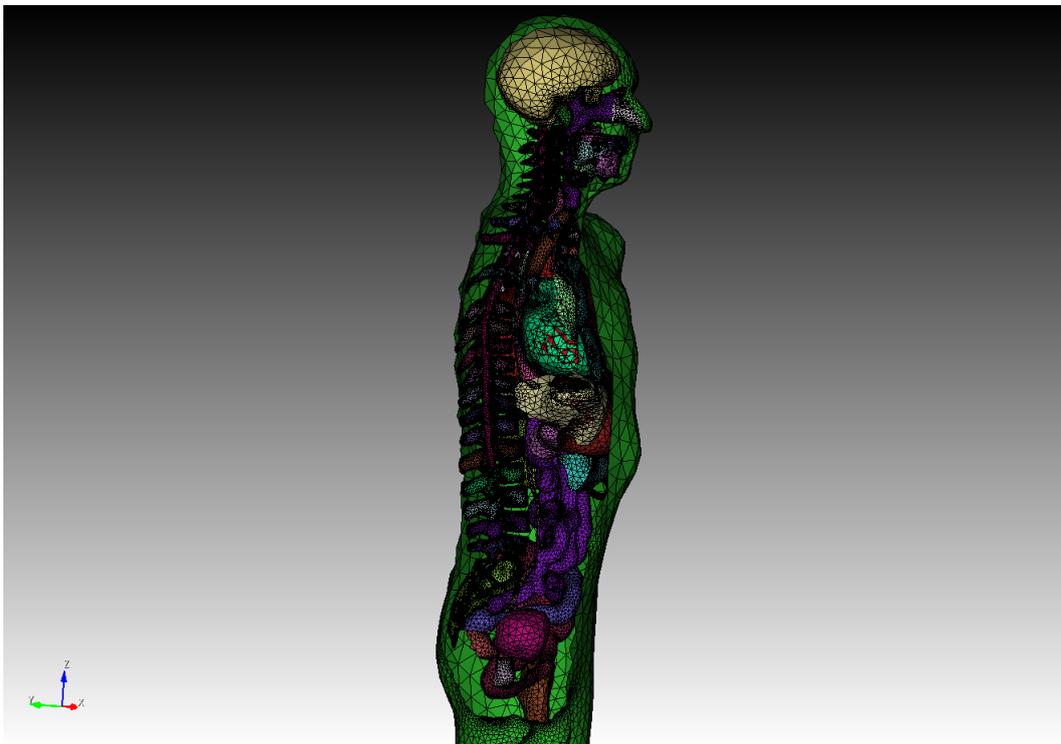


Figure 7. Adult male human phantom cross section.

All simulations except for one were done with Co-60 source. Co-60 source energy values and their corresponding probabilities are  $1.1732 \cdot 10^6$  eV with 0.5 probability of realizing

and  $1.3325 \cdot 10^6$  eV with 0.5 probability of realising. One of the simulations was replicated with Cs-137, which has the source energy of  $6.617 \cdot 10^5$  eV, to see the difference in dose rate received from different isotopes.

### 5.1.1 Source site: Heart

The source was placed at the middle of the heart volume and the target organ is the brain. The dose rate received by the brain is  $5.62 \cdot 10^5 \mu Sv/h \pm 1.34 \cdot 10^{-3}$ .

For reference, during prostate cancer radiotherapy patients receive the dose around  $8.00 \cdot 10^7 \mu Sv$  and during the Chernobyl accident in 1986, the recovery operation workers received the approximate dose of  $1.20 \cdot 10^5 \mu Sv$  [29] [30].

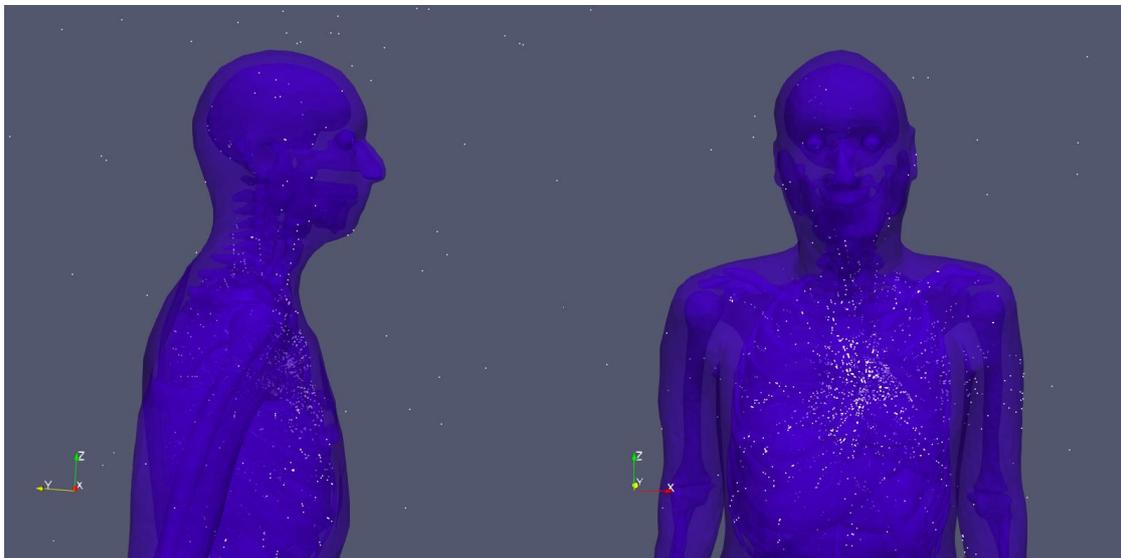


Figure 8. Interaction points in heart to brain simulation.

In figure 8 the particle interaction points can be seen inside the human phantom. The highest concentration of interactions resides near the source in the chest. Some interaction points can be seen in the brain in side and front view. The interaction points seen in the background are particles entering the graveyard zone surrounding the human phantom.

### 5.1.2 Source site: Liver

For this simulation the source was now placed at the liver volume and the dose rate received by the brain was viewed. The goal was to see a decrease in radiation dose received by the brain since liver is located further from the brain than heart.

The dose for brain in this simulation was  $3.58 \cdot 10^5 \mu Sv/h \pm 1.03 \cdot 10^{-3}$ , which is about 36% lower than when the source is placed inside the heart.

### 5.1.3 Source site: Stomach

In this simulation the source was placed inside the stomach. The stomach in its proximity lies fairly close to the liver. Their difference in location mostly lies in the phantoms z-coordinate. The stomach volume is closer to the pelvis.

The dose received by the brain is  $1.81 \cdot 10^5 \mu Sv/h \pm 7.42 \cdot 10^{-4}$ , which is around 68% lower than the first simulation.

### 5.1.4 Source site: Spleen

In this simulation the dose rate received was  $306.86 \mu Sv/h \pm 1.47 \cdot 10^{-5}$ . The dose rate is much smaller than other organs produced.

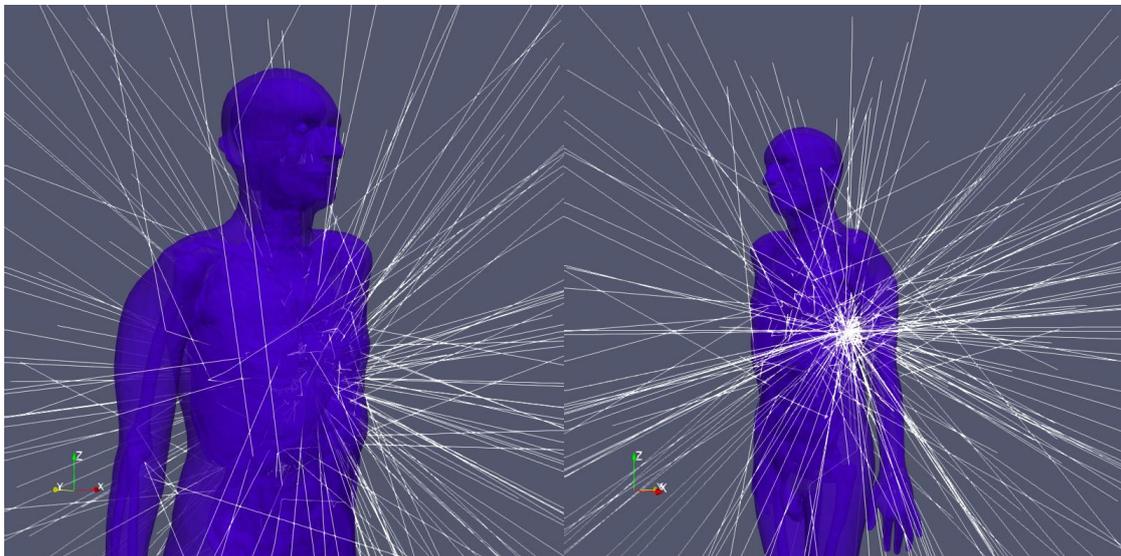


Figure 9. Particle tracks in spleen to brain simulation.

Visualising the particle tracks (figure 9) shows that most particles exit the phantom near the source site. The spleen is located right under the ribs, close to the skin while the other organs simulated before reside more towards the center of the phantom. Due to that most particles escape the geometry and don't contribute to the flux received by the brain.

### 5.1.5 Source site: Bladder

The source was taken further away from the target. The result of the simulation, where the source was placed inside the bladder, was  $29.00 \mu Sv/h \pm 5.21 \cdot 10^{-6}$ .

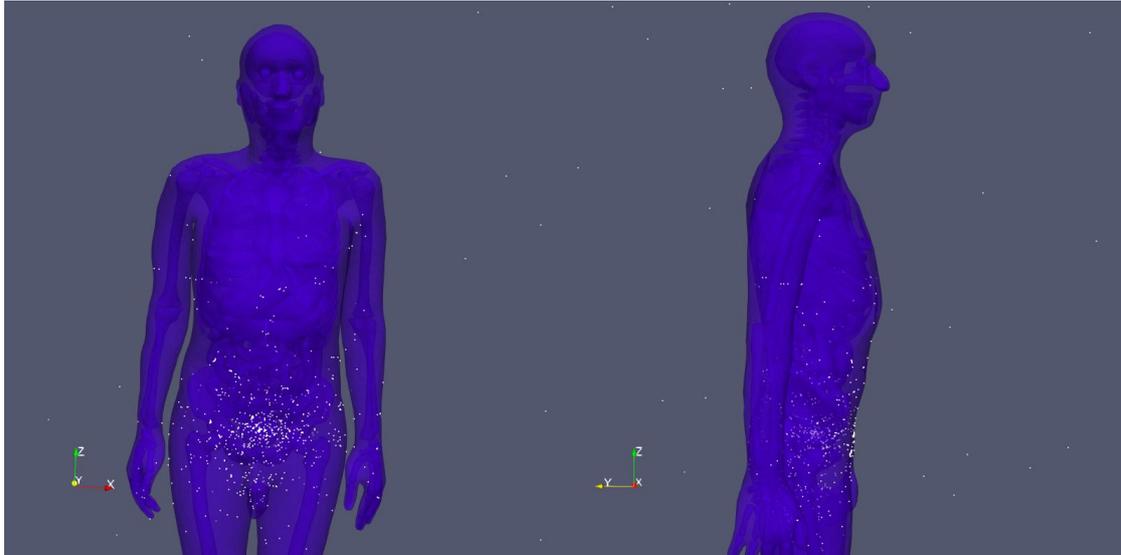


Figure 10. Interaction points in bladder to brain simulation.

The interaction points (figure 10) show that most of the particles remain near the pelvic area of the phantom and don't reach the brain volume. This result shows that when simulating the dose rate received by the brain, its best to stay near the torso and chest area of the phantom. At the pelvic region the particles no longer reach the desired target organ.

### 5.1.6 Target: Teeth

For this simulation the source was placed inside the heart and the dose received by the teeth was calculated. The teeth have the highest material density in the phantom. The aim of this simulation was to see, if the higher density and closer proximity play a role in the received dose rate.

The dose rate received by the teeth was  $3.12 \cdot 10^5 \mu Sv/h \pm 4.53 \cdot 10^{-5}$ , which is approximately 44% smaller than the dose rate received by the brain. The dose rate is smaller even though the target is closer to the source.

### 5.1.7 Source site: Heart with Cs-137

In this simulation we reproduced the first simulation, where the source was placed inside the heart and the dose rate at the brain is observed. Instead of Co-60 the phantom was radiated with Cs-137.

The dose rate received by brain was  $2.93 \cdot 10^5 \mu Sv/h \pm 7.55 \cdot 10^{-4}$ , which is around 48% smaller than the dose rate received with Co-60 source isotope.

The particle interaction points are visualised in figure 11.

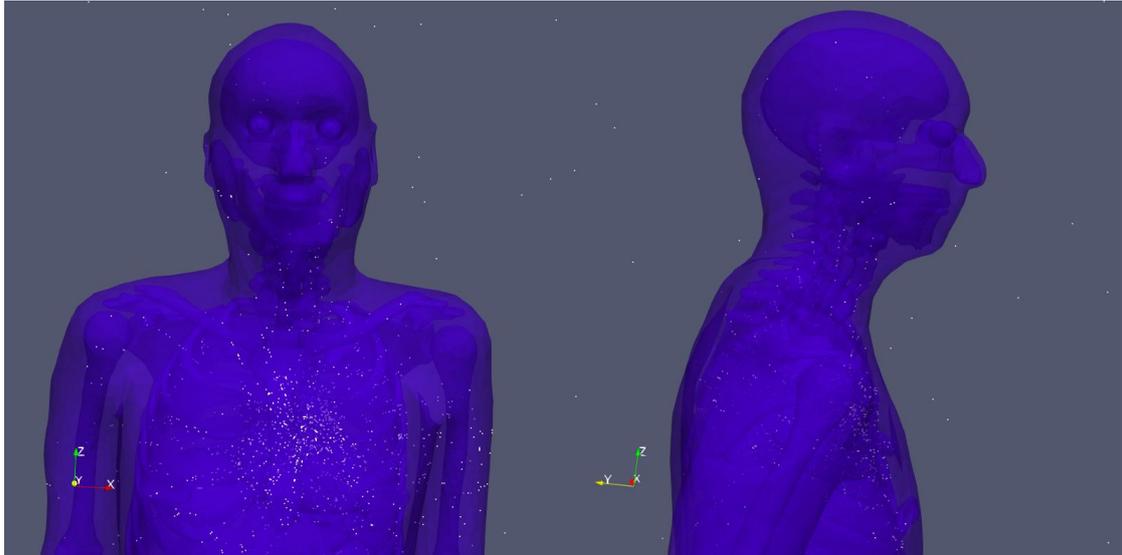


Figure 11. Interaction points with Cs-137 source.

### 5.1.8 Source site: Mandible

In this simulation the source was placed inside the mandible or the jawbone.

The dose rate received by the brain in this simulation was  $3.17 \cdot 10^3 \mu Sv/h \pm 4.93 \cdot 10^{-5}$ .

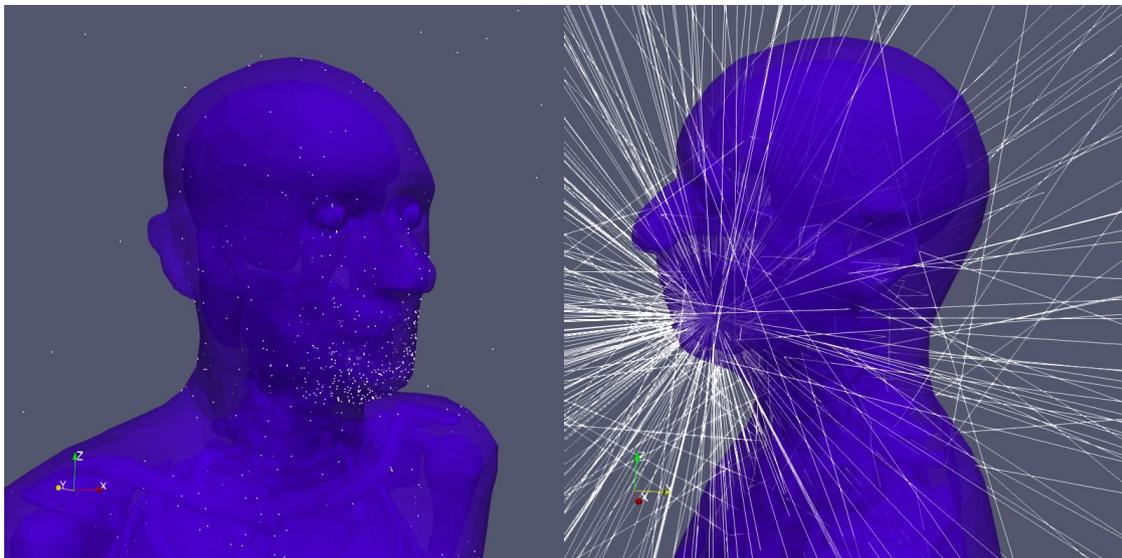


Figure 12. Particle tracks in mandible to brain simulation.

In figure 12 we can see that that most of the particle interactions remain in the vicinity of the mandible. Even though the source is closer to the target than in the first simulation, the dose rate is smaller. This can be due to several reasons:

Firstly the placement of the source is once again closer to the outer boundary of the

phantom. Just like with spleen, we once again see a large amount of particles leaving the geometry near the source site.

Secondly due to the higher amount of calcium, the bones absorb more irradiation than soft tissue organs [31]. In this case most of the particles transfer their energy at the jawbone and thus the dose rate received by the brain itself is smaller.

### 5.1.9 Source placed outside the body

In the last simulation the source was placed outside of the phantom but near the mandible in order to compare dose rate results. The dose rate received by the brain was  $1.35 \cdot 10^3 \mu Sv/h \pm 2.87 \cdot 10^{-5}$ , which is around 57% smaller than when the source is placed inside the mandible.

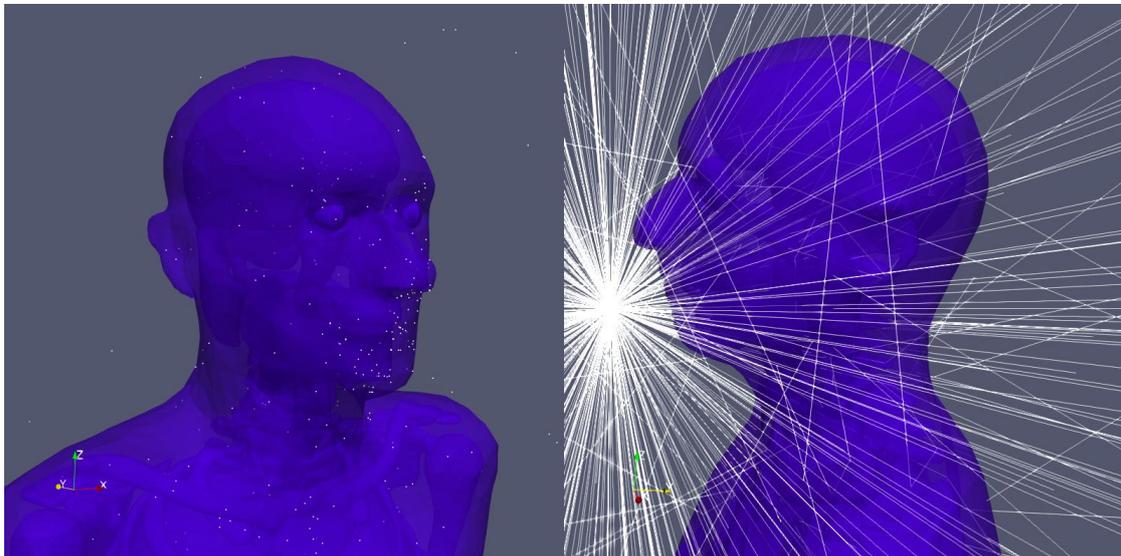


Figure 13. Particle tracks when the source is placed outside of the phantom.

The particle tracks and interaction points can be seen in figure 13.

## 5.2 Discussion

Multiple simulations were carried out with the prepared human phantom. The source was placed at different locations and the dose rate received by a target volume was viewed. The following subsections discuss the results of the simulations and propose future directions that could be taken with this work.

## 5.2.1 Results

The aim of the simulations was to test the geometry created. To see how the dose rates change depending on the placement, materials surrounding it and the source isotope used. To do that the source was placed in various parts of the geometry and their corresponding dose rates checked. The results were influenced by the distance between the source and the target and the materials in the vicinity. All the simulation results can be seen in table 2.

Table 2. Simulation Results and Dose Rates

Isotope	Target site	Source site	Dose rate ( $\mu\text{Sv/h}$ )
Co-60	Brain	Heart	$5.62 \cdot 10^5$
Co-60		Liver	$3.56 \cdot 10^5$
Co-60		Stomach	$1.81 \cdot 10^5$
Co-60		Spleen	306.86
Co-60		Bladder	29.00
Cs-137		Heart	$2.93 \cdot 10^5$
Co-60	Teeth	Mandible	$3.17 \cdot 10^3$
Co-60		Outside	$1.35 \cdot 10^3$
Co-60		Heart	$3.12 \cdot 10^5$

The decrease of dose rates dependent on the location is visualised in figure 14.

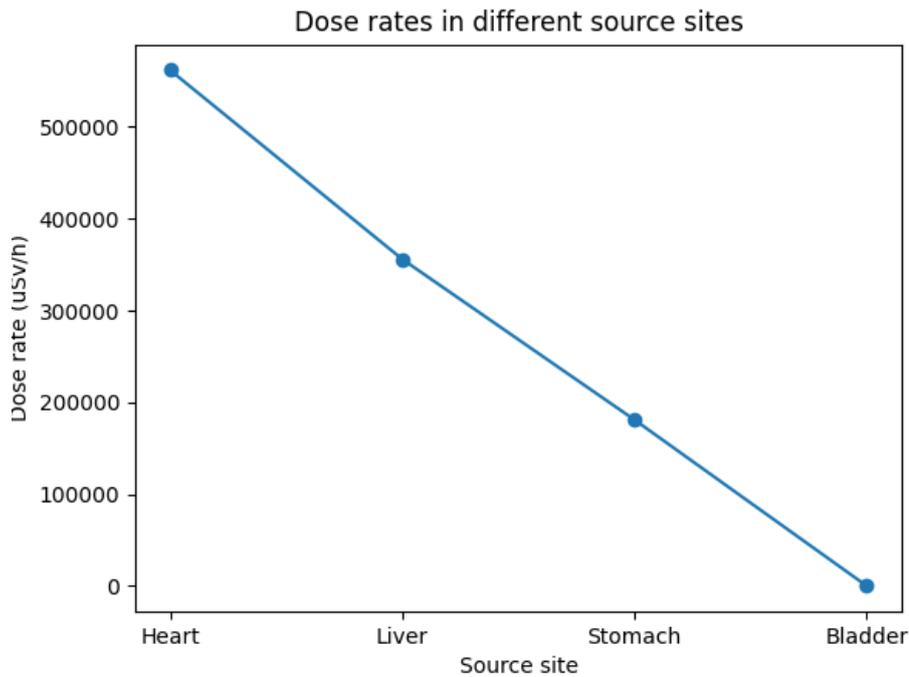


Figure 14. Dose rates in different organs.

Figure 14 shows the decrease of dose rates when the source is moved further away from the brain. When the source is placed at the bladder the particles don't reach the brain and thus its dose rate is minuscule compared to others. Spleen isn't represented in the graph due to the fact, that most of the particles left the phantom through the skin. But overall the simulation results are as expected. The further the source is placed from the target, the smaller the dose rate received.

In the next simulations the effectiveness of material density was tested. Two simulations were carried out with identical source sites but different target sites with different densities. One target site was the brain and the other target site was teeth. If no material densities would have been taken into consideration then the latter simulation should have a higher dose rate due to having closer proximity with the source. But the outcome was the opposite.

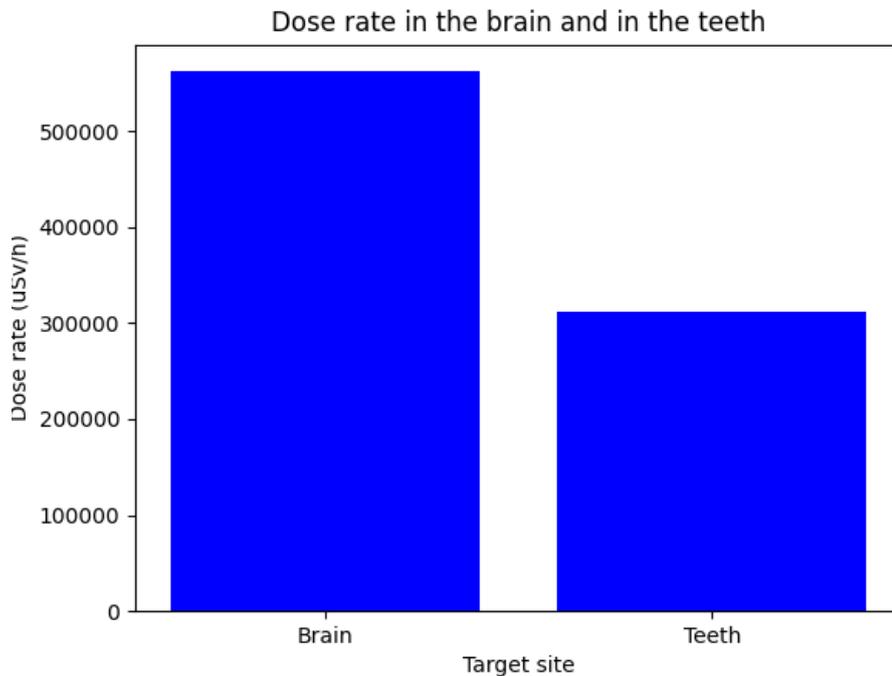


Figure 15. Dose rate difference between the brain and teeth.

In figure 15 we can see that the dose rate differences are substantial. Teeth have the material density of  $2.688 \text{ g/cm}^3$  while brain has the density of  $1.041 \text{ g/cm}^3$ . The teeth volume should have a higher dose rate due to the higher density and calcium content. But in this case the volumes of the targets should be taken into consideration as well. We have a point source with isotropic scattering and a smaller target by the volume size. In figure 8 we can see that there is a scarcity of particle interactions near the region. Thus the smaller dose rate can be due to the size of the target but further simulations would need to be carried out to know for certain.

Cs-137 and Co-60 isotopes were compared to each other as well. The simulations were

identical, where both were placed in the heart and the dose rate received by the brain was calculated.

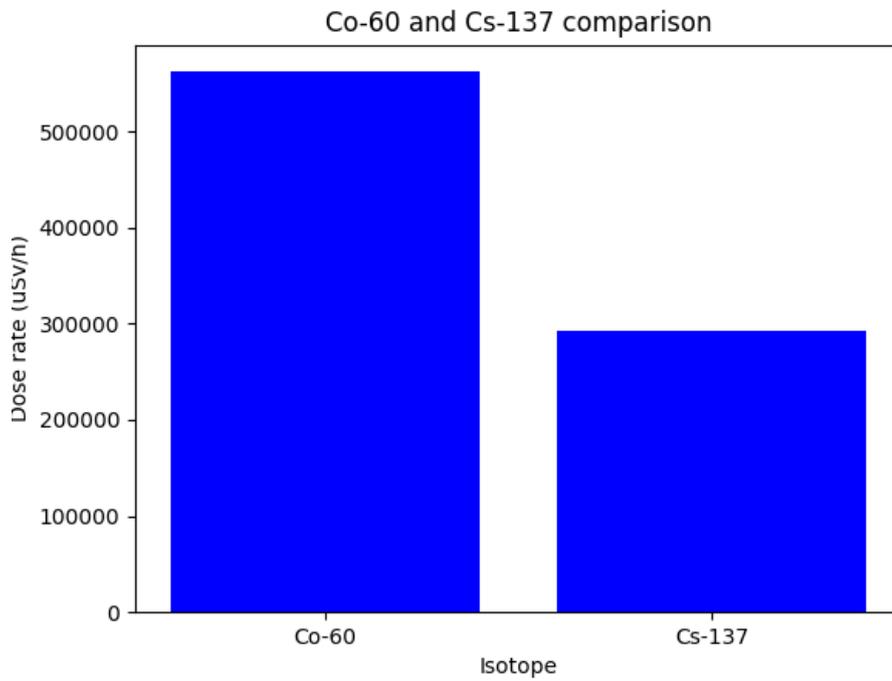


Figure 16. Dose rates with different isotopes.

Figure 16 shows differences in dose rates when using Co-60 or Cs-137 source. The results were as expected and served their purpose in demonstrating how different energy levels produce different dose rates.

In the last simulations the Co-60 source was placed inside the mandible and near the mandible but outside of the phantom.

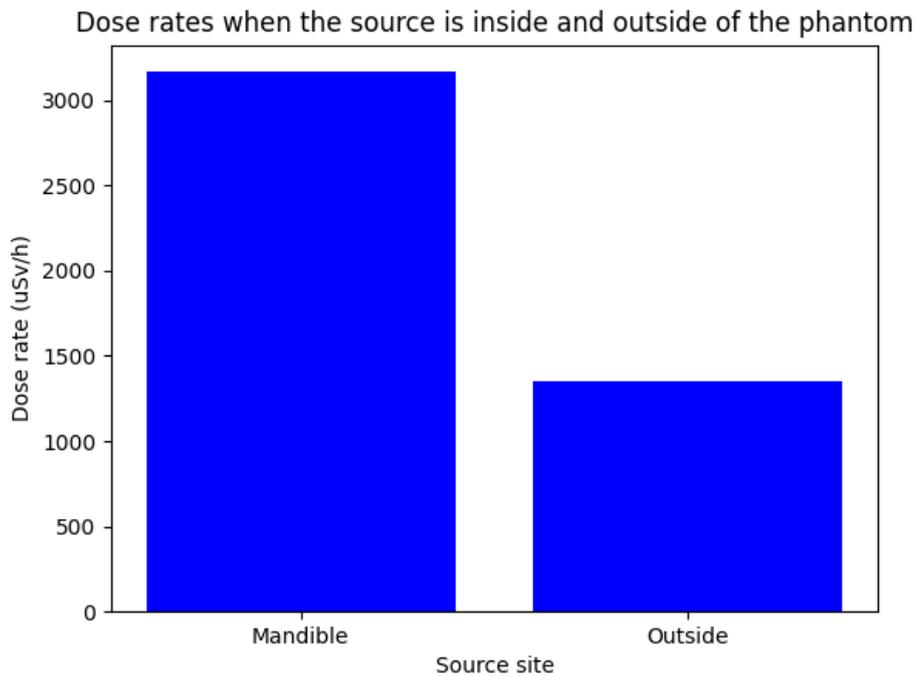


Figure 17. Dose rates between inside and outside source placement.

Figure 17 shows the difference in dose rates. The external source produces nearly half of the dose rate that the phantom receives internally. The dose rates are smaller than in other simulations. The mandible is again with a higher density and thus influences the dose in that regard.

Overall the simulations served its purpose in demonstrating the different aspects of the geometry.

### **Validation**

The source activity of 20 Ci was selected based off an study done by Eman Massoud in 2014 called "Dose Assessment for Some Industrial Gamma Sources with an Application to a Radiation Accident". In their work they used Co-60 and Cs-137 among other isotopes with the activity of 20 Ci. They simulated the gamma radiation exposure to workers based on a 2005 radiation accident at Meet Halfa, Egypt.

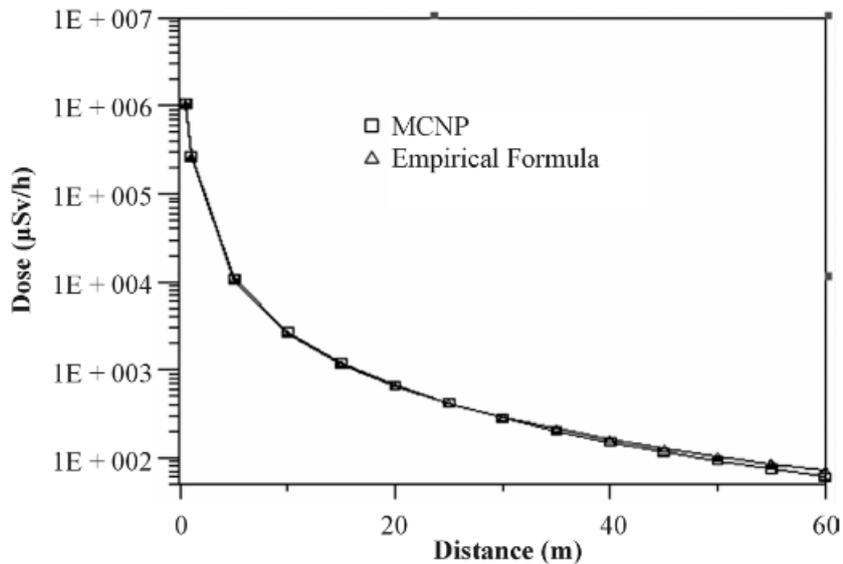


Figure 18. Dose assessment study results with Co-60 source [28].

Figure 18 shows their dose rate results with Co-60 source isotope. Even though the same source isotope and activity was used, their results are quite different. At 0.5 m from the source the dose rate received is  $1.1 \cdot 10^6 \mu Sv/h$ . In this thesis, the highest dose rate received was  $5.62 \cdot 10^5 \mu Sv/h$ . Additionally dose rates at magnitude  $10^3$  was achieved approximately 20 m from the source while in this thesis dose rates at that magnitude were acquired from inside the phantom.

With Cs-137 isotope at 0.5 m MCNP calculated the dose rate of  $2.94 \cdot 10^5 \mu Sv/h$  while in this thesis the dose rate received from Cs-137 was  $2.93 \cdot 10^5 \mu Sv/h$ . While the dose rates are similar, the difference in distance from the source is around 20 cm between the OpenMC and MCNP simulation.

Presumably the main reason for these different results lies in the phantom used for the simulation. Even though minimal information was provided about the human model used in the MCNP simulations, this thesis works with a simplified model. The model doesn't consists of all possible materials that make up a human being. Another possible reason could be the different particle transport code used. MCNP has been extensively utilized for simulations related to human scenarios and may thus offer a wider array of tally filters suitable for dosimetric simulations.

## 5.2.2 Future direction

This thesis works with a simplified adult male phantom, which offers a chance for interesting topics for future physics students.

Due to the complexity of the model and the time limitation some of the volumes had to be

removed. In future works the whole phantom could be implemented and their corresponding dose rates compared. This would offer a change to see how much, for example, the muscle layer affects the particle interactions in different parts of the phantom.

Once the phantom is completed, it would be interesting to see how it performs in different particle transport simulation codes. A comparative study between OpenMC and MCNP for example could be carried out.

ICRP nr. 145 publication also includes a full adult female phantom. The female and male phantoms have different organs, different densities and different organ sizes to work with. An interesting comparative study could be done between the two to demonstrate the differences in dose rates.

According to the developers of OpenMC, their particle transport code hasn't been used with complete human phantoms yet. This could offer a great opportunity for collaboration. In highly complex geometries the code is put to test and it could offer useful input for both sides.

Once the whole phantom is implemented a more difficult study could be done with hemodynamics. Hemodynamics play a vital role in the transport of radiation through human body. With hemodynamics added and an ethical clearance the phantom results could be compared to real life cancer treatments.

And lastly, since most of the focus in this thesis was set on geometry preparation, it offers a change to optimise the code. OpenMC offers multiple directions a user can take to carry out their simulations. In this thesis a code was created to demonstrate the successful creation of the phantom. But the limitations of the code were unexplored.

## 6. Summary

The main goal of the thesis was to successfully prepare a human phantom geometry for OpenMC. Once the phantom was successfully integrated into the simulation, the secondary goal was to carry out radiation transport simulations in order to see how the dose rates change depending on the source placement and isotope. The human phantom CAD model was acquired from the ICRP and prepared in Cubit Coreform. During the preparation of the geometry, the phantom was simplified, overlaps were removed, material names defined for all the volumes and the whole model was meshed. A simulation was set up in OpenMC, where all necessary input files were created and the prepared phantom was used as a DAGMC geometry. The flux score is tallied from a specific material and later converted into dose rate. The simulations consists of 200 batches and 35 000 particles. Nine simulation scenarios were created, where mostly Co-60 isotope was used as the source. The source was placed at different volumes and the dose rate received by mostly the brain was calculated.

The simulations showed how the dose rate decreases depending on the distance from the source and how different materials influence the dose rate. Simulations where the source was placed further from the brain but inside a soft tissue organ produced higher dose rates then simulations where the source was closer to the brain but placed inside a higher density bone. The simulations where the source was placed inside an organ produced dose rates at the magnitude of  $10^5 \mu Sv/h$ . The simulation where the source was placed inside the mandible and outside of the phantom produced dose rates at the magnitude of  $10^3 \mu Sv/h$ . The smallest dose rates were received in simulations where either the particles didn't reach the brain or left the geometry near the source site.

The results obtained from the simulations were compared to a dose assessment study done in 2014 and possible future directions for this topic were suggested.

## 7. Acknowledgements

I extend my gratitude to the individuals and communities who played pivotal roles in the successful completion of this thesis.

First and foremost, my sincere appreciation goes to Dr. Marti Jeltsov, my supervisor, for providing unwavering academic guidance and support throughout this journey.

I am indebted to Dr. Rodrigo Gonzalez Gonzaga De Oliveira, Dr. April Novak, Dr. Patrick Shriwise and Hando Tohver for generously sharing their time and expertise during moments of uncertainty, enriching my understanding of the subject matter.

Special acknowledgment is extended to Dr. Sergei Nazarenko and Kätlin Tiigi, whose insights on the medical aspects of this thesis were indispensable.

A heartfelt thanks to the OpenMC and Cubit Coreform communities for their consistent and invaluable assistance. Their support significantly contributed to the development and execution of this thesis.

Each of you has played a unique role in the realization of this work, and for that, I am sincerely grateful.

## References

- [1] *Age-standardized rate (World) per 100 000, incidence, males and females*. [Accessed: 18-12-23]. URL: [https://gco.iarc.fr/overtime/en/dataviz/trends?populations=75200&sexes=1\\_2&types=%200&multiple\\_populations=1cancers=0](https://gco.iarc.fr/overtime/en/dataviz/trends?populations=75200&sexes=1_2&types=%200&multiple_populations=1cancers=0).
- [2] Dennis Leaver Charles M. Washington. *Principles and Practice on Radiation Therapy*. Elsevier, 2016.
- [3] Avid Khamenehfar. “Reversal of Multidrug Resistance Fluorescently Measured in Single Cancer Cells Captured in the Microfluidic Chip”. In: 2011. URL: <https://summit.sfu.ca/item/15621>.
- [4] *Oligometastatic cancer: What patients with stage IV cancer should know*. [Accessed: 01-01-24]. URL: <https://www.mdanderson.org/cancerwise/what-is-oligometastatic-cancer--what-patients-with-stage-iv-cancer-should-know.h00-159617067.html>.
- [5] *External Beam Radiation Therapy for Cancer*. [Accessed: 08-11-23]. URL: <https://www.cancer.gov/about-cancer/treatment/types/radiation-therapy/external-beam#top>.
- [6] Maria Mania Aspridakis Anders Ahnesjö. “Dose calculations for external photon beams in radiotherapy”. In: *Phys. Med. Biol.* (1999). DOI: 10.1088/0031-9155/44/11/201. URL: <https://iopscience.iop.org/article/10.1088/0031-9155/44/11/201>.
- [7] *Cancer treatment: Brachytherapy*. [Accessed: 08-11-23]. URL: <https://www.iaea.org/topics/cancer-treatment-brachytherapy>.
- [8] Marek KANIKOWSKI et al. “Permanent implants in treatment of prostate cancer”. In: *Reports of Practical Oncology and Radiotherapy* 13.3 (2008), pp. 150–167. URL: <https://www.sciencedirect.com/science/article/pii/S1507136710600065>.
- [9] *Brachytherapy to Treat Cancer*. [Accessed: 08-11-23]. URL: <https://www.cancer.gov/about-cancer/treatment/types/radiation-therapy/brachytherapy>.
- [10] A.M. Johansen. “Monte Carlo Methods”. In: *International Encyclopedia of Education (Third Edition)*. Ed. by Penelope Peterson, Eva Baker, and Barry McGaw. Third Edition. Oxford: Elsevier, 2010, pp. 296–303.

- [11] Helmut G. Katzgraber. “Introduction to Monte Carlo Methods”. In: *arXiv: Statistical Mechanics* (2009). URL: <https://api.semanticscholar.org/CorpusID:263770498>.
- [12] Peter Bonate. “A Brief Introduction to Monte Carlo Simulation”. In: *Clinical pharmacokinetics* 40 (2001), pp. 15–22.
- [13] James Arvo and David Kirk. “Particle Transport and Image Synthesis”. In: *SIGGRAPH Comput. Graph.* 24 (1990).
- [14] Dr. Lucas Norberto Burigo. “Introduction to Monte Carlo Particle Transport”. In: *The Physics of Charged Particle Therapy*. 2020.
- [15] J. Seuntjens and D.W.O. Rogers. “Monte Carlo Applications in Measurement Dosimetry”. In: 2009. URL: <https://www.aapm.org/meetings/09SS/documents/05Seuntjens-MonteCarloIntro.pdf>.
- [16] Jaakko Leppänen. “Development of a New Monte Carlo Reactor Physics Code”. In: 2007. URL: <https://serpent.vtt.fi/serpent/download/P640.pdf>.
- [17] Gregory R. Hart et al. Ying Liang Wazir Muhammad. “A general-purpose Monte Carlo particle transport code based on inverse transform sampling for radiotherapy dose calculation”. In: *Sci Rep* (2020).
- [18] Pedro Andreo. “Monte Carlo simulations in radiotherapy dosimetry”. In: *Radiation Oncology* (2018).
- [19] H Paganetti. “Monte Carlo simulations will change the way we treat patients with proton beams today”. In: *Br J Radiol.* (2014).
- [20] N. Petoussi-Hens et al. C.H. Kim Y.S. Yeom. “Adult Mesh-type Reference Computational Phantoms”. In: 2020. URL: <https://www.icrp.org/publication.asp?id=ICRP%20Publication%20145>.
- [21] *Advanced hex meshing for challenging simulations*. [Accessed: 15-09-23]. URL: <https://coreform.com/products/coreform-cubit/>.
- [22] *Coreform Cubit 2023.11 User Documentation*. [Accessed: 15-09-23]. URL: [https://coreform.com/cubit\\_help/cubithelp.htm](https://coreform.com/cubit_help/cubithelp.htm).
- [23] *A Beginner’s Guide to OpenMC*. [Accessed: 20-10-23]. URL: <https://docs.openmc.org/en/latest/usersguide/beginners.html>.
- [24] *Code-Specific steps for OpenMC*. [Accessed: 20-10-23]. URL: <https://svalinn.github.io/DAGMC/usersguide/codes/openmc.html>.

- [25] Patrick C. Shriwise et al. "Particle tracking acceleration via signed distance fields in direct-accelerated geometry Monte Carlo". In: *Nuclear Engineering and Technology* 49 (2017), pp. 1189–1198. URL: <https://www.sciencedirect.com/science/article/pii/S1738573317303145>.
- [26] *Theory and Methodology*. [Accessed: 23-10-23]. URL: <https://docs.openmc.org/en/stable/methods/index.html>.
- [27] *Cross Sections*. [Accessed: 13-11-23]. URL: [https://courses.washington.edu/mengr430/au07/handouts/cross\\_section.pdf](https://courses.washington.edu/mengr430/au07/handouts/cross_section.pdf).
- [28] Eman Massoud. "Dose Assessment for Some Industrial Gamma Sources with an Application to a Radiation Accident". In: *Open Journal of Modelling and Simulation* (2014).
- [29] Thomas N. Eade et al. "WHAT DOSE OF EXTERNAL-BEAM RADIATION IS HIGH ENOUGH FOR PROSTATE CANCER?" In: *Int J Radiat Oncol Biol Phys*. (2007).
- [30] *Assessments of the radiation effects from the Chernobyl nuclear reactor accident*. [Accessed: 28-12-23]. URL: <https://www.unscear.org/unscear/en/areas-of-work/chernobyl.html>.
- [31] Michaela R. Reagan Samantha Costa. "Therapeutic Irradiation: Consequences for Bone and Bone Marrow Adipose Tissue". In: *Front Endocrinol (Lausanne)* (2019).

# Appendix 1 – Non-Exclusive License for Reproduction and Publication of a Graduation Thesis<sup>1</sup>

I Krislin Sartakov

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis “Radiation transport simulations in a human phantom”, supervised by Marti Jeltsov
  - 1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
  - 1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons’ intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

01.01.2024

---

<sup>1</sup>The non-exclusive licence is not valid during the validity of access restriction indicated in the student’s application for restriction on access to the graduation thesis that has been signed by the school’s dean, except in case of the university’s right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

## Appendix 2 - OpenMC code

```
%matplotlib inline
import openmc
from IPython.display import Image
from dagmc_geometry_slice_plotter import plot_axis_slice
import math
import os
import matplotlib.pyplot as plt
import numpy as np

#materials
skin=openmc.Material(name='skin ')
skin.add_element('H',10.0)
skin.add_element('C',19.9)
skin.add_element('N',4.2)
skin.add_element('O',65.0)
skin.add_element('Na',0.2)
skin.add_element('P',0.1)
skin.add_element('S',0.2)
skin.add_element('Cl',0.3)
skin.add_element('K',0.1)
skin.set_density('g/cm3', 1.089)
skin.id=1

mandible=openmc.Material(name='mandible ')
mandible.add_element('H',7.7)
mandible.add_element('C',33.2)
mandible.add_element('N',3.0)
mandible.add_element('O',42.0)
mandible.add_element('Na',0.2)
mandible.add_element('Mg',0.1)
mandible.add_element('P',4.1)
mandible.add_element('S',0.2)
mandible.add_element('Cl',0.1)
mandible.add_element('K',0.1)
mandible.add_element('Ca',9.3)
mandible.set_density('g/cm3',1.271)
mandible.id=6
```

```
thoracispine=openmc.Material(name='thoracispine')
thoracispine.add_element('H',10.0)
thoracispine.add_element('C',40.3)
thoracispine.add_element('N',2.8)
thoracispine.add_element('O',43.1)
thoracispine.add_element('Na',0.1)
thoracispine.add_element('P',1.0)
thoracispine.add_element('S',0.2)
thoracispine.add_element('Cl',0.2)
thoracispine.add_element('K',0.1)
thoracispine.add_element('Ca',2.1)
thoracispine.add_element('Fe',0.1)
thoracispine.set_density('g/cm3',1.070)
thoracispine.id=7
```

```
sacrum=openmc.Material(name='sacrum')
sacrum.add_element('H',10.5)
sacrum.add_element('C',42.6)
sacrum.add_element('N',2.7)
sacrum.add_element('O',42.6)
sacrum.add_element('Na',0.1)
sacrum.add_element('P',0.3)
sacrum.add_element('S',0.2)
sacrum.add_element('Cl',0.2)
sacrum.add_element('K',0.1)
sacrum.add_element('Ca',0.6)
sacrum.add_element('Fe',0.1)
sacrum.set_density('g/cm3',1.033)
sacrum.id=8
```

```
ribs=openmc.Material(name='ribs')
ribs.add_element('H',8.8)
ribs.add_element('C',34.6)
ribs.add_element('N',3.1)
ribs.add_element('O',44.4)
ribs.add_element('Na',0.1)
ribs.add_element('Mg',0.1)
ribs.add_element('P',2.6)
ribs.add_element('S',0.2)
ribs.add_element('Cl',0.1)
ribs.add_element('K',0.1)
```

```

ribs.add_element('Ca',5.8)
ribs.add_element('Fe',0.1)
ribs.set_density('g/cm3',1.170)
ribs.id=9

pelvis=openmc.Material(name='pelvis')
pelvis.add_element('H',9.4)
pelvis.add_element('C',40.9)
pelvis.add_element('N',2.6)
pelvis.add_element('O',40.0)
pelvis.add_element('Na',0.1)
pelvis.add_element('Mg',0.1)
pelvis.add_element('P',2.0)
pelvis.add_element('S',0.2)
pelvis.add_element('Cl',0.1)
pelvis.add_element('K',0.1)
pelvis.add_element('Ca',4.5)
pelvis.set_density('g/cm3',1.121)
pelvis.id=10

lumbaraspine=openmc.Material(name='lumbaraspine')
lumbaraspine.add_element('H',9.5)
lumbaraspine.add_element('C',38.0)
lumbaraspine.add_element('N',3.0)
lumbaraspine.add_element('O',43.6)
lumbaraspine.add_element('Na',0.1)
lumbaraspine.add_element('P',1.6)
lumbaraspine.add_element('S',0.2)
lumbaraspine.add_element('Cl',0.2)
lumbaraspine.add_element('K',0.1)
lumbaraspine.add_element('Ca',3.6)
lumbaraspine.add_element('Fe',0.1)
lumbaraspine.set_density('g/cm3',1.108)
lumbaraspine.id=11

clavicles=openmc.Material(name='clavicles')
clavicles.add_element('H',8.9)
clavicles.add_element('C',40.9)
clavicles.add_element('N',2.5)
clavicles.add_element('O',38.5)
clavicles.add_element('Na',0.1)

```

```

clavicles.add_element('P',2.7)
clavicles.add_element('S',0.2)
clavicles.add_element('Cl',0.1)
clavicles.add_element('K',0.1)
clavicles.add_element('Ca',6.0)
clavicles.set_density('g/cm3',1.157)
clavicles.id=12

cervicalspine=openmc.Material(name='cervicalspine')
cervicalspine.add_element('H',10.3)
cervicalspine.add_element('C',41.6)
cervicalspine.add_element('N',2.8)
cervicalspine.add_element('O',42.8)
cervicalspine.add_element('Na',0.1)
cervicalspine.add_element('P',0.6)
cervicalspine.add_element('S',0.2)
cervicalspine.add_element('Cl',0.2)
cervicalspine.add_element('K',0.1)
cervicalspine.add_element('Ca',1.2)
cervicalspine.add_element('Fe',0.1)
cervicalspine.set_density('g/cm3',1.049)
cervicalspine.id=13

cartilage=openmc.Material(name='cartilage')
cartilage.add_element('H',9.6)
cartilage.add_element('C',9.9)
cartilage.add_element('N',2.2)
cartilage.add_element('O',74.4)
cartilage.add_element('Na',0.5)
cartilage.add_element('P',2.2)
cartilage.add_element('S',0.9)
cartilage.add_element('Cl',0.3)
cartilage.set_density('g/cm3',1.099)
cartilage.id=14

scapulae=openmc.Material(name='scapulae')
scapulae.add_element('H',8.4)
scapulae.add_element('C',37.3)
scapulae.add_element('N',2.7)
scapulae.add_element('O',40.4)
scapulae.add_element('Na',0.1)

```

```
scapulae.add_element('Mg',0.1)
scapulae.add_element('P',3.3)
scapulae.add_element('S',0.2)
scapulae.add_element('Cl',0.1)
scapulae.add_element('K',0.1)
scapulae.add_element('Ca',7.3)
scapulae.set_density('g/cm3',1.201)
scapulae.id=15
```

```
sternum=openmc.Material(name='sternum')
sternum.add_element('H',10.4)
sternum.add_element('C',42.1)
sternum.add_element('N',2.8)
sternum.add_element('O',42.7)
sternum.add_element('P',0.5)
sternum.add_element('S',0.2)
sternum.add_element('Cl',0.2)
sternum.add_element('K',0.1)
sternum.add_element('Ca',0.9)
sternum.add_element('Fe',0.1)
sternum.set_density('g/cm3',1.041)
sternum.id=16
```

```
teeth=openmc.Material(name='teeth')
teeth.add_element('H',2.3)
teeth.add_element('C',9.5)
teeth.add_element('N',2.9)
teeth.add_element('O',42.6)
teeth.add_element('Mg',0.7)
teeth.add_element('P',13.5)
teeth.add_element('Ca',28.5)
teeth.set_density('g/cm3',2.688)
teeth.id=17
```

```
eyes=openmc.Material(name='eyes')
eyes.add_element('H',11.2)
eyes.add_element('C',0.4)
eyes.add_element('N',0.1)
eyes.add_element('O',88.3)
eyes.set_density('g/cm3',1.031)
eyes.id=18
```

```
bladder=openmc . Material (name=' bladder ')
bladder . add_element ( ' H ' , 10.5)
bladder . add_element ( ' C ' , 9.6)
bladder . add_element ( ' N ' , 2.6)
bladder . add_element ( ' O ' , 76.1)
bladder . add_element ( ' Na ' , 0.2)
bladder . add_element ( ' P ' , 0.2)
bladder . add_element ( ' S ' , 0.2)
bladder . add_element ( ' Cl ' , 0.3)
bladder . add_element ( ' K ' , 0.3)
bladder . set_density ( ' g/cm3 ' , 1.040)
bladder . id=19
```

```
colon=openmc . Material (name=' colon ')
colon . add_element ( ' H ' , 10.5)
colon . add_element ( ' C ' , 11.4)
colon . add_element ( ' N ' , 2.5)
colon . add_element ( ' O ' , 75.0)
colon . add_element ( ' Na ' , 0.1)
colon . add_element ( ' P ' , 0.1)
colon . add_element ( ' S ' , 0.1)
colon . add_element ( ' Cl ' , 0.2)
colon . add_element ( ' K ' , 0.1)
colon . set_density ( ' g/cm3 ' , 1.037)
colon . id=2
```

```
tongue=openmc . Material (name=' tongue ')
tongue . add_element ( ' H ' , 10.2)
tongue . add_element ( ' C ' , 14.2)
tongue . add_element ( ' N ' , 3.4)
tongue . add_element ( ' O ' , 71.1)
tongue . add_element ( ' Na ' , 0.1)
tongue . add_element ( ' P ' , 0.2)
tongue . add_element ( ' S ' , 0.3)
tongue . add_element ( ' Cl ' , 0.1)
tongue . add_element ( ' K ' , 0.4)
tongue . set_density ( ' g/cm3 ' , 1.050)
tongue . id=20
```

```
thyroid=openmc . Material (name=' thyroid ')
```

```
thyroid.add_element('H',10.4)
thyroid.add_element('C',11.8)
thyroid.add_element('N',2.5)
thyroid.add_element('O',74.5)
thyroid.add_element('Na',0.2)
thyroid.add_element('P',0.1)
thyroid.add_element('S',0.1)
thyroid.add_element('Cl',0.2)
thyroid.add_element('K',0.1)
thyroid.add_element('I',0.1)
thyroid.set_density('g/cm3',1.051)
thyroid.id=21
```

```
stomach=openmc.Material(name='stomach')
stomach.add_element('H',10.5)
stomach.add_element('C',11.4)
stomach.add_element('N',2.5)
stomach.add_element('O',75.0)
stomach.add_element('Na',0.1)
stomach.add_element('P',0.1)
stomach.add_element('S',0.1)
stomach.add_element('Cl',0.2)
stomach.add_element('K',0.1)
stomach.set_density('g/cm3',1.037)
stomach.id=22
```

```
spleen=openmc.Material(name='spleen')
spleen.add_element('H',10.3)
spleen.add_element('C',11.2)
spleen.add_element('N',3.2)
spleen.add_element('O',74.3)
spleen.add_element('Na',0.1)
spleen.add_element('P',0.2)
spleen.add_element('S',0.2)
spleen.add_element('Cl',0.2)
spleen.add_element('K',0.3)
spleen.set_density('g/cm3',1.060)
spleen.id=3
```

```
small_intestine=openmc.Material(name='small_intestine')
small_intestine.add_element('H',10.5)
```

```
small_intestine.add_element('C',11.4)
small_intestine.add_element('N',2.5)
small_intestine.add_element('O',75.0)
small_intestine.add_element('Na',0.1)
small_intestine.add_element('P',0.1)
small_intestine.add_element('S',0.1)
small_intestine.add_element('Cl',0.2)
small_intestine.add_element('K',0.1)
small_intestine.set_density('g/cm3',1.037)
small_intestine.id=23
```

```
pancreas=openmc.Material(name='pancreas')
pancreas.add_element('H',10.5)
pancreas.add_element('C',15.8)
pancreas.add_element('N',2.4)
pancreas.add_element('O',70.4)
pancreas.add_element('Na',0.2)
pancreas.add_element('P',0.2)
pancreas.add_element('S',0.1)
pancreas.add_element('Cl',0.2)
pancreas.add_element('K',0.2)
pancreas.add_element('Ca',0.2)
pancreas.set_density('g/cm3',1.044)
pancreas.id=24
```

```
lungs=openmc.Material(name='lungs')
lungs.add_element('H',10.2)
lungs.add_element('C',10.8)
lungs.add_element('N',3.2)
lungs.add_element('O',74.8)
lungs.add_element('Na',0.1)
lungs.add_element('P',0.1)
lungs.add_element('S',0.2)
lungs.add_element('Cl',0.3)
lungs.add_element('K',0.2)
lungs.add_element('Fe',0.1)
lungs.set_density('g/cm3',0.415)
lungs.id=25
```

```
liver=openmc.Material(name='liver')
liver.add_element('H',10.2)
```

```

liver.add_element('C',13.2)
liver.add_element('N',3.1)
liver.add_element('O',72.3)
liver.add_element('Na',0.2)
liver.add_element('P',0.2)
liver.add_element('S',0.3)
liver.add_element('Cl',0.2)
liver.add_element('K',0.3)
liver.set_density('g/cm3',1.060)
liver.id=26

kidneys=openmc.Material(name='kidneys')
kidneys.add_element('H',10.3)
kidneys.add_element('C',12.6)
kidneys.add_element('N',3.1)
kidneys.add_element('O',72.9)
kidneys.add_element('Na',0.2)
kidneys.add_element('P',0.2)
kidneys.add_element('S',0.2)
kidneys.add_element('Cl',0.2)
kidneys.add_element('K',0.2)
kidneys.add_element('Ca',0.1)
kidneys.set_density('g/cm3',1.053)
kidneys.id=27

humeri=openmc.Material(name='humeri')
humeri.add_element('H',8.1)
humeri.add_element('C',35.4)
humeri.add_element('N',2.8)
humeri.add_element('O',41.0)
humeri.add_element('Na',0.2)
humeri.add_element('Mg',0.1)
humeri.add_element('P',3.7)
humeri.add_element('S',0.2)
humeri.add_element('Cl',0.1)
humeri.add_element('K',0.1)
humeri.add_element('Ca',8.3)
humeri.set_density('g/cm3',1.233)
humeri.id=48

heart=openmc.Material(name='heart')

```

```

heart.add_element('H',10.4)
heart.add_element('C',13.5)
heart.add_element('N',2.9)
heart.add_element('O',72.2)
heart.add_element('Na',0.1)
heart.add_element('P',0.2)
heart.add_element('S',0.2)
heart.add_element('Cl',0.2)
heart.add_element('K',0.3)
heart.set_density('g/cm3',1.051)
heart.id=28

breasts=openmc.Material(name='breasts')
breasts.add_element('H',10.6)
breasts.add_element('C',32.4)
breasts.add_element('N',3.0)
breasts.add_element('O',53.5)
breasts.add_element('Na',0.1)
breasts.add_element('P',0.1)
breasts.add_element('S',0.2)
breasts.add_element('Cl',0.1)
breasts.set_density('g/cm3',1.021)
breasts.id=29

brain=openmc.Material(name='brain')
brain.add_element('H',10.7)
brain.add_element('C',14.3)
brain.add_element('N',2.3)
brain.add_element('O',71.3)
brain.add_element('Na',0.2)
brain.add_element('P',0.4)
brain.add_element('S',0.2)
brain.add_element('Cl',0.3)
brain.add_element('K',0.3)
brain.set_density('g/cm3',1.041)
brain.id=4

adrenals=openmc.Material(name='adrenals')
adrenals.add_element('H',10.4)
adrenals.add_element('C',22.8)
adrenals.add_element('N',2.8)

```

```
adrenals.add_element('O',63.0)
adrenals.add_element('Na',0.1)
adrenals.add_element('P',0.2)
adrenals.add_element('S',0.3)
adrenals.add_element('Cl',0.2)
adrenals.add_element('K',0.2)
adrenals.set_density('g/cm3',1.036)
adrenals.id=30
```

```
prostate=openmc.Material(name='prostate')
prostate.add_element('H',10.5)
prostate.add_element('C',25.1)
prostate.add_element('N',2.7)
prostate.add_element('O',60.7)
prostate.add_element('Na',0.1)
prostate.add_element('P',0.2)
prostate.add_element('S',0.3)
prostate.add_element('Cl',0.2)
prostate.add_element('K',0.2)
prostate.set_density('g/cm3',1.031)
prostate.id=31
```

```
oesophagus=openmc.Material(name='oesophagus')
oesophagus.add_element('H',10.4)
oesophagus.add_element('C',22.3)
oesophagus.add_element('N',2.8)
oesophagus.add_element('O',63.5)
oesophagus.add_element('Na',0.1)
oesophagus.add_element('P',0.2)
oesophagus.add_element('S',0.3)
oesophagus.add_element('Cl',0.2)
oesophagus.add_element('K',0.2)
oesophagus.set_density('g/cm3',1.037)
oesophagus.id=32
```

```
wrist_and_hand_bones=openmc.Material(name='wrist_and_hand_bones')
wrist_and_hand_bones.add_element('H',9.6)
wrist_and_hand_bones.add_element('C',50.4)
wrist_and_hand_bones.add_element('N',1.7)
wrist_and_hand_bones.add_element('O',30.8)
wrist_and_hand_bones.add_element('Na',0.1)
```

```

wrist_and_hand_bones.add_element('P',2.2)
wrist_and_hand_bones.add_element('S',0.2)
wrist_and_hand_bones.add_element('Cl',0.1)
wrist_and_hand_bones.add_element('Ca',4.9)
wrist_and_hand_bones.set_density('g/cm3',1.109)
wrist_and_hand_bones.id=33

ulnae_and_radri=openmc.Material(name='ulnae_and_radri')
ulnae_and_radri.add_element('H',9.6)
ulnae_and_radri.add_element('C',50.4)
ulnae_and_radri.add_element('N',1.7)
ulnae_and_radri.add_element('O',30.8)
ulnae_and_radri.add_element('Na',0.1)
ulnae_and_radri.add_element('P',2.2)
ulnae_and_radri.add_element('S',0.2)
ulnae_and_radri.add_element('Cl',0.1)
ulnae_and_radri.add_element('Ca',4.9)
ulnae_and_radri.set_density('g/cm3',1.109)
ulnae_and_radri.id=34

tibiae_fibulae=openmc.Material(name='tibiae_fibulae')
tibiae_fibulae.add_element('H',9.6)
tibiae_fibulae.add_element('C',50.4)
tibiae_fibulae.add_element('N',1.7)
tibiae_fibulae.add_element('O',30.8)
tibiae_fibulae.add_element('Na',0.1)
tibiae_fibulae.add_element('P',2.2)
tibiae_fibulae.add_element('S',0.2)
tibiae_fibulae.add_element('Cl',0.1)
tibiae_fibulae.add_element('Ca',4.9)
tibiae_fibulae.set_density('g/cm3',1.109)
tibiae_fibulae.id=35

spinal_cord=openmc.Material(name='spinal_cord')
spinal_cord.add_element('H',9.6)
spinal_cord.add_element('C',50.4)
spinal_cord.add_element('N',1.7)
spinal_cord.add_element('O',30.8)
spinal_cord.add_element('Na',0.1)
spinal_cord.add_element('P',2.2)
spinal_cord.add_element('S',0.2)

```

```

spinal_cord.add_element('Cl',0.1)
spinal_cord.add_element('Ca',4.9)
spinal_cord.set_density('g/cm3',1.109)
spinal_cord.id=36

femora=openmc.Material(name='femora')
femora.add_element('H',9.6)
femora.add_element('C',50.4)
femora.add_element('N',1.7)
femora.add_element('O',30.8)
femora.add_element('Na',0.1)
femora.add_element('P',2.2)
femora.add_element('S',0.2)
femora.add_element('Cl',0.1)
femora.add_element('Ca',4.9)
femora.set_density('g/cm3',1.109)
femora.id=37

ankles_and_foot_bones=
openmc.Material(name='ankles_and_foot_bones')
ankles_and_foot_bones.add_element('H',9.6)
ankles_and_foot_bones.add_element('C',50.4)
ankles_and_foot_bones.add_element('N',1.7)
ankles_and_foot_bones.add_element('O',30.8)
ankles_and_foot_bones.add_element('Na',0.1)
ankles_and_foot_bones.add_element('P',2.2)
ankles_and_foot_bones.add_element('S',0.2)
ankles_and_foot_bones.add_element('Cl',0.1)
ankles_and_foot_bones.add_element('Ca',4.9)
ankles_and_foot_bones.set_density('g/cm3',1.109)
ankles_and_foot_bones.id=38

ureter=openmc.Material(name='ureter')
ureter.add_element('H',10.5)
ureter.add_element('C',25.1)
ureter.add_element('N',2.7)
ureter.add_element('O',60.7)
ureter.add_element('Na',0.1)
ureter.add_element('P',0.2)
ureter.add_element('S',0.3)
ureter.add_element('Cl',0.2)

```

```
ureter.add_element('K',0.2)
ureter.set_density('g/cm3',1.031)
ureter.id=39
```

```
trachea=openmc.Material(name='trachea')
trachea.add_element('H',10.5)
trachea.add_element('C',25.1)
trachea.add_element('N',2.7)
trachea.add_element('O',60.7)
trachea.add_element('Na',0.1)
trachea.add_element('P',0.2)
trachea.add_element('S',0.3)
trachea.add_element('Cl',0.2)
trachea.add_element('K',0.2)
trachea.set_density('g/cm3',1.031)
trachea.id=40
```

```
thymus=openmc.Material(name='thymus')
thymus.add_element('H',10.5)
thymus.add_element('C',25.1)
thymus.add_element('N',2.7)
thymus.add_element('O',60.7)
thymus.add_element('Na',0.1)
thymus.add_element('P',0.2)
thymus.add_element('S',0.3)
thymus.add_element('Cl',0.2)
thymus.add_element('K',0.2)
thymus.set_density('g/cm3',1.031)
thymus.id=41
```

```
salivary_glands=openmc.Material(name='salivary_glands')
salivary_glands.add_element('H',10.5)
salivary_glands.add_element('C',25.1)
salivary_glands.add_element('N',2.7)
salivary_glands.add_element('O',60.7)
salivary_glands.add_element('Na',0.1)
salivary_glands.add_element('P',0.2)
salivary_glands.add_element('S',0.3)
salivary_glands.add_element('Cl',0.2)
salivary_glands.add_element('K',0.2)
salivary_glands.set_density('g/cm3',1.031)
```

```

salivary_glands.id=42

rectum=openmc.Material(name='rectum')
rectum.add_element('H',10.5)
rectum.add_element('C',25.1)
rectum.add_element('N',2.7)
rectum.add_element('O',60.7)
rectum.add_element('Na',0.1)
rectum.add_element('P',0.2)
rectum.add_element('S',0.3)
rectum.add_element('Cl',0.2)
rectum.add_element('K',0.2)
rectum.set_density('g/cm3',1.031)
rectum.id=43

gall_bladder=openmc.Material(name='gall_bladder')
gall_bladder.add_element('H',10.5)
gall_bladder.add_element('C',25.1)
gall_bladder.add_element('N',2.7)
gall_bladder.add_element('O',60.7)
gall_bladder.add_element('Na',0.1)
gall_bladder.add_element('P',0.2)
gall_bladder.add_element('S',0.3)
gall_bladder.add_element('Cl',0.2)
gall_bladder.add_element('K',0.2)
gall_bladder.set_density('g/cm3',1.031)
gall_bladder.id=44

ET2=openmc.Material(name='ET2')
ET2.add_element('H',10.5)
ET2.add_element('C',25.1)
ET2.add_element('N',2.7)
ET2.add_element('O',60.7)
ET2.add_element('Na',0.1)
ET2.add_element('P',0.2)
ET2.add_element('S',0.3)
ET2.add_element('Cl',0.2)
ET2.add_element('K',0.2)
ET2.set_density('g/cm3',1.031)
ET2.id=45

```

```

ET1=openmc.Material(name='ET1')
ET1.add_element('H',10.5)
ET1.add_element('C',25.1)
ET1.add_element('N',2.7)
ET1.add_element('O',60.7)
ET1.add_element('Na',0.1)
ET1.add_element('P',0.2)
ET1.add_element('S',0.3)
ET1.add_element('Cl',0.2)
ET1.add_element('K',0.2)
ET1.set_density('g/cm3',1.031)
ET1.id=46

```

```

BB1=openmc.Material(name='BB1')
BB1.add_element('H',10.5)
BB1.add_element('C',25.1)
BB1.add_element('N',2.7)
BB1.add_element('O',60.7)
BB1.add_element('Na',0.1)
BB1.add_element('P',0.2)
BB1.add_element('S',0.3)
BB1.add_element('Cl',0.2)
BB1.add_element('K',0.2)
BB1.set_density('g/cm3',1.031)
BB1.id=47

```

```

materials=openmc.Materials([skin, mandible, thoraicspine, sacrum,
                             ribs, pelvis, lumbar spine, clavicles,
                             cervical spine, cartilage, scapulae,
                             sternum, teeth, bladder, colon,
                             tongue, thyroid, stomach, adrenals,
                             spleen, small_intestine, pancreas,
                             lungs, liver, kidneys, humeri, heart,
                             breasts, brain, eyes,
                             salivary_glands, prostate,
                             oesophagus, wrist_and_hand_bones,
                             ulnae_and_radii, tibiae_fibulae,
                             spinal_cord, femora,
                             ankles_and_foot_bones, ureter,
                             trachea, thymus, rectum,

```

```

gall_bladder , ET2, ET1, BB1])
materials.export_to_xml()

#geometry
dagmc_univ=openmc.DAGMCUniverse(filename='AM.h5m')
                                .bounded_universe()
geometry=openmc.Geometry(root=dagmc_univ)
geometry.export_to_xml()

#plotting
plot = plot_axis_slice(
    dagmc_file_or_trimesh_object='AM.h5m',
    view_direction='z',
)

plot.show()

plot = plot_axis_slice(
    dagmc_file_or_trimesh_object='AM.h5m',
    view_direction='x',
)

plot.show()

plot = plot_axis_slice(
    dagmc_file_or_trimesh_object='AM.h5m',
    view_direction='y',
)

plot.show()

#Source
source=openmc.Source()
source.space=openmc.stats.Point((3.07, -30, 46.8))
source.angle = openmc.stats.Isotropic()

#Co60
source.energy = openmc.stats.Discrete([1.1732e6,1.3325e6],
                                       [0.5,0.5])

#Cs137

```

```

#source.energy = openmc.stats.Discrete([6.617e5], [1.0])
source.particle = 'photon'

#settings
settings=openmc.Settings()
settings.source=source
settings.batches=200
settings.inactive=50
settings.particles=35000
settings.run_mode = "fixed source"
settings.photon_transport = True
settings.DAGMC=True
settings.export_to_xml()

#tallies
energy_bins_p, dose_coeffs_p =
openmc.data.dose_coefficients(particle='photon', geometry='AP')
energy_function_filter_p =
openmc.EnergyFunctionFilter(energy_bins_p, dose_coeffs_p)
energy_function_filter_p.interpolation == 'cubic'

photon_particle_filter = openmc.ParticleFilter(["photon"])
material_filter=openmc.MaterialFilter(brain)

dose_tally = openmc.Tally(name="dose_tally")
dose_tally.scores = ["flux"]
dose_tally.filters = [
    material_filter,
    photon_particle_filter,
    energy_function_filter_p,
]
my_tallies = openmc.Tallies([dose_tally])
my_tallies.export_to_xml()

#running the model
!rm *.h5
openmc.run(tracks=True, threads=8)

#post-processing
sp=openmc.StatePoint('statepoint.200.h5')

```

```

tally=sp.get_tally(name='dose_tally ')
df=tally.get_pandas_dataframe()
print(df)

dose_tally_result = sp.get_tally(
    name="dose_tally"
).mean.flatten()[0]

photons_per_second = 740000000000

dose_conversion1 = dose_tally_result * photons_per_second
dose_conversion2 = dose_conversion1 / 1430.994645
dose_conversion3 = dose_conversion2 * 1e-6
dose = dose_conversion3 * 60 * 60

print(f'The tally result is {dose_tally_result}
and the dose is {dose:.2f} uSv/h')

#Converting particle tracks for ParaView
os.system('openmc-track-to-vtk tracks.h5 -o tracks.vtk')

```