INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE FÍSICA Y MATEMÁTICAS

A MONTE CARLO SIMULATION FOR STEREOTACTIC BODY RADIATION THERAPY FOR LUNG CANCER

BY

RAQUEL IVON
SANCHEZ ESTRADA

THESIS SUBMITTED TO THE COLLEGE OF PROFESSORS IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN MATHEMATICAL PHYSICS AT THE

INSTITUTO POLITÉCNICO NACIONAL

THESIS ADVISORS:
DR. ARTURO DELFÍN LOYA (ININ)
DR. EDMUNDO DEL VALLE GALLEGOS (IPN)

MEXICO CITY, MEXICO, DECEMBER 2017

SIP-13-BIS

INSTITUTO POLITÉCNICO NACIONAL
SECRETARIA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REGISTRO DE TEMA DE TESIS
Y DESIGNACIÓN DE DIRECTORES DE TESIS

Ciudad de México, a **7** de **Septiembre** del **2017**

El Colegio de Profesores de Estudios de Posgrado e Investigación de **ESFM** en su sesión **Ordinaria** No. **03** celebrada el día **29** del mes de **Marzo de 2017** conoció la solicitud presentada por el(la) alumno(a):

SANCHEZ	**ESTRADA**	**RAQUEL IVON**
Apellido paterno	Apellido materno	Nombre (s)

Con registro: | B | 1 | 5 | 0 | 9 | 3 | 4 |

Aspirante de la: **Maestría en Ciencias Fisicomatemáticas**

1.- Se designa al aspirante el tema de tesis titulado:
"A Monte Carlo simulation for stereotactic body radiation therapy for lung cancer"

De manera general el tema abarcará los siguientes aspectos:

Se anexa hoja

2.- Se designan como Directores de Tesis a los Profesores:
Dr. Edmundo del Valle Gallegos y Dr. Arturo Delfin Loya

3.- El trabajo de investigación base para el desarrollo de la tesina será elaborado por el alumno en:
Las instalaciones de la Escuela Superior de Física y Matemáticas

que cuenta con los recursos e infraestructura necesarios.

4.- El interesado deberá asistir a los seminarios desarrollados en el área de adscripción del trabajo desde la fecha en que se suscribe la presente hasta la aceptación de la tesis por la Comisión Revisora correspondiente:

Directores de Tesis

Dr. Edmundo del Valle Gallegos

Dr. Arturo Delfin Loya

Aspirante

Raquel Ivon Sanchez Estrada

Presidente del Colegio

SEP
ESCUELA SUPERIOR DE FISICA Y MATEMATICAS
SECCION DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
IPN

Dra. Concepción Mejía García

SIP-14-BIS

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de **México** siendo las **14:00** horas del día **8** del mes de **Septiembre** del **2017** se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de la **ESFM** para examinar la tesis titulada:

"A Monte Carlo simulation for stereotactic body radiation therapy for lung cancer"

Presentada por el(la) alumno(a):

SANCHEZ	**ESTRADA**	**RAQUEL IVON**
Apellido paterno	Apellido materno	Nombre(s)

Con registro:

B	1	5	0	9	3	4

aspirante de la :

MAESTRÍA EN CIENCIAS FISICOMATEMÁTICAS

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Directores de tesis

DR. EDMUNDO DEL VALLE GALLEGOS

DR. ARTURO DELFIN LOYA

M. EN C. CARLOS FILIO LÓPEZ

DR. HERNANI TIAGO YEE MADEIRA

DRA. NORMA SÁNCHEZ SALAS

PRESIDENTE DEL COLEGIO DE PROFESORES

SEP
ESCUELA SUPERIOR DE
FISICA Y MATEMATICAS
SECCION DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN
IPN

DRA. CONCEPCIÓN MEJÍA GARCÍA

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México, D.F. el día 13 del mes de Noviembre del año 2017, la que suscribe Raquel Ivon Sanchez Estrada, alumna del Programa de Maestría en Ciencias Fisicomatemáticas con número de registro B150934, adscrita a la Escuela Superior de Física y Matemáticas, manifiesta que es la autora intelectual del presente trabajo de Tesis bajo la dirección del Dr. Arturo Delfín Loya y del Dr. Edmundo del Valle Gallegos y cede los derechos del trabajo titulado *A Monte Carlo simulation for stereotactic body radiation therapy for lung cancer*, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso de la autora y/o directores del trabajo. Este puede ser obtenido escribiendo a las siguientes direcciones fis.ivonsanchez@gmail.com, arturo.delfin@inin.gob.mx y evalle@ipn.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

RAQUEL IVON SANCHEZ ESTRADA

Nombre y firma del alumno(a)

Abstract

> "Every patient with cancer should have access to the best possible care regardless of constraints such as geographic separation from adequate facilities and professional competence, economic restrictions, cultural barriers or methods of health care delivery" - Inter-Society Council for Radiation Oncology

Radiotherapy cases containing severe heterogeneities are expected to benefit the most from Monte Carlo verifications if an unclear result of the deterministic calculation is anticipated, . According to the Task Group from the American Association of Physicist in Medicine, the strongest motivation for a Monte Carlo verification comes from the treatment of lung cancer. This becomes evident as there are percentage differences between the calculated and the measured dose as big as 30 % in some cases. Therefore there is a strong motivation to provide a method to implement general purpose Monte Carlo codes into the clinical practice in Mexico. The current work consists to determine the feasibility of applying a Monte Carlo simulation for lung cancer as an analysis and research tool for clinical comparisons.

Keywords: Monte Carlo simulation, SBRT, lung cancer, MCNP6, Geant4

Resumen

Si se anticipa un resultado poco claro del cálculo determinista, se espera que los casos de radioterapia que contienen heterogeneidades severas sean los que más beneficien de las verificaciones Monte Carlo. Según el grupo de trabajo de la Asociación Americana de Física en Medicina, la motivación más fuerte para una verificación de Monte Carlo viene del tratamiento para cáncer de pulmón. Esto se hace evidente ya que existen diferencias porcentuales entre la dosis calculada y la dosis medida hasta de 30% en algunos casos. Por lo tanto, existe una fuerte motivación para proporcionar un método que pueda implementar códigos Monte Carlo en la práctica clínica en México. El trabajo actual consiste en determinar la viabilidad de aplicar una simulación de Monte Carlo para el cáncer de pulmón como herramienta de análisis e investigación para comparaciones clínicas.

Palabras clave: Simulación Monte Carlo, SBRT, cáncer de pulmón, MCNP6, Geant4

Acknowledgments

I would like to express my deep gratitude to my advisors, Dr. Arturo Delfin Loya and Dr. Edmundo del Valle Gallegos, for the continuous support, patient, motivation, and immense knowledge. Their guidance was essential for the development of this project.

I thank Dr. Hernani Yee, Dr. Norma Sánchez, and MSc. Carlos Filio for their insightful comments and various perspectives.

My sincere thanks also go to the ININ research group for giving me the opportunity to access the research facilities.

I would like to thank Dr. C-M Charle Ma for his valuable guidance and for providing me with the tools and advice that I needed to choose the right direction to complete this work.

I deeply thank MSc. Mariana Hernández for her trust, support, encouragement and valuable advice throughout my years of study.

I would like to thank my family: my mother and my sister for supporting me through this process.

And finally, I immensely thank Enrique Betancourt for his immense help and unfailing support, this accomplishment would not have been possible without you.

Contents

Abstract **i**

Resumen **iii**

Acknowledgments **v**

Contents **vii**

List of figures **x**

List of tables **xii**

Acronyms **xiii**

1 Radiotherapy **1**

1.1 The treatment planning process 1

1.2 Dose calculation methods for treatment planning 3

1.3 Monte Carlo radiotherapy treatments 4

1.3.1 Monte Carlo for deterministic algorithm modifications 4

1.3.2 Monte Carlo radiotherapy treatments in Mexico 6

1.4 Objectives and scope of the thesis 7

2 Physics and clinical aspects of photon beams in Monte Carlo simulation **8**

2.1 Monte Carlo codes 8

2.1.1 Monte Carlo components 9

2.2 Modeling of photon transport 10

2.2.1 The anatomy of a history 14

2.3 Modeling of electron transport 14

2.3.1 Condensed history technique 15

2.4 Benchmarking 15

3 Monte Carlo treatment planning systems in radiotherapy **17**

3.1 Beam Modeling 17

3.1.0.1 Full linac simulation 18

3.1.0.2 Phase space simulation 18

3.2 Benchmarking, uncertainties and validation of the treatment 19
3.2.1 Percentage Depth Dose 20
3.2.2 Beam profiles 20
3.2.3 Heterogeneous phantoms 21
3.2.4 Statistical Uncertainties 22
3.2.5 Noise reduction technique 22
3.2.6 Variance reduction techniques 23
3.2.7 Verification of the simulated treatment 23
3.3 Dose calculation 24
3.3.1 Dose calculation in treatment planning systems 24
3.3.2 Dose calculation in Monte Carlo simulation 25
3.3.2.1 Monte Carlo for deterministic algorithm dose-modifications 25
3.3.3 Computer tomography based simulations in Monte Carlo 26
3.4 Monte Carlo for lung cancer 27
3.4.1 Monte Carlo simulation for stereotactic lung radiotherapy . . . 28

4 Monte Carlo method 30
4.1 Monte Carlo N-Particle code 30
4.1.1 MCNP input file 31
4.1.2 Phantom construction 32
4.1.3 Beam simulation: phase space 34
4.1.4 Dose calculation 35
4.2 Geant4 code 36
4.2.1 Geant4 input file 36
4.2.2 Phantom construction 39
4.2.3 Beam simulation phase space 40
4.2.4 Dose calculation 41

5 Results and discussion 42
5.1 Beam calculation 42
5.1.1 MCNP code 43
5.1.2 Geant4 44
5.2 Depth dose curves 45
5.3 Beam profile 48
5.4 QUASAR phantom 51
5.4.1 MCNP 51

5.4.2 Geant4 52
5.4.3 MCNP and Geant4 52
5.5 DICOM voxelized phantom 53

6 Conclusions and future work **59**

Bibliography **61**

A Geant4 phase space classes **66**
A.1 iaea config file 66
A.2 iaea header file 69
A.3 iaea phsp file 74

B MCNP Codes **85**
B.1 Percentage depth dose simulation 85
B.2 QUASAR phantom geometry simulation 91

C Geant4 water phantom input file **94**
C.1 Water Phantom simuation 94

List of Figures

1.1 Linear accelerator Novalis-Tx 2

2.1 The number of papers published by the different versions of EGS*, MCNP*, and their relationship (the comparison is performed by similar versions of the EGS and MCNP codes). 10
2.2 The relative importance of interaction plotted as a function of photon energy and the atomic number of the material. 11
2.3 (a)Compton scattering of a photon from electron, as a result of the collision, the photon is scattered leading to (b) an ionized atom. 12
2.4 Logic flow of photon transport in Monte Carlo simulation. 13
2.5 Logic flow for class I and II Monte Carlo algorithms for electron transport simulation. 16

3.1 Full linac simulation components and location representation of the phase space. 19
3.2 PDD curves in water for a 10x10 cm^2 field in a SSD of 100 cm for different photon beams. 21

4.1 Water phantom simulation for PDD measurements, the tallies are contructed along the z axis. 33
4.2 QUASAR phantom. 34
4.3 Geant4 geometrical hierarchy of volume specifications. 38
4.4 Geometrical representation of the voxelixed phantom based CT-images 40

5.1 Geant4 visualization beam for a 6 MeV, 10×10 cm^2 beam. 45
5.2 Depth dose curve simulation comparisons of MCNP and Geant4 46
5.3 Depth dose curve simulation from Geant4 47
5.4 Depth dose curve simulation from MCNP 47
5.5 Beam profile along x axis from MCNP and Geant4 comparison for a 10×10 cm^2 field 49
5.6 Beam profile along y axis from MCNP and Geant4 comparison for a 10×10 cm^2 field 49
5.7 Beam profile along x axis from MCNP and Geant4 comparison for a 4×4 cm^2 field 50

5.8 Beam profile along y axis from MCNP and Geant4 comparison for a $4 \times 4\ cm^2$ field 50
5.9 Quasar phantom with the circular tallies simulated from a a) QUASAR front view and b) xial view 51
5.10 Percentage Depth Dose comparisons in the QUASAR phantom for MCNP and Geant4 52
5.11 a) QUASAR circular tallies at $(0, 0, -100)$ and b) the QUASAR dose distribution from a 4×4 beam at 0 *degrees* at the circular tallies . . . 53
5.12 DICOM voxelized image simulation in Geant4 for a a) 0 degrees beam angle and b) 45 degrees beam angle. 55
5.13 DICOM voxelized image simulation in Geant4 for a a) 73 degrees beam angle and b) 135 degrees beam angle. 55
5.14 DICOM voxelized image simulation in Geant4 for a a) 180 degrees beam angle and b) 225 degrees beam angle. 56
5.15 DICOM voxelized image simulation in Geant4 for a a) 270 degrees beam angle and b) 315 degrees beam angle. 56
5.16 Simulated dose distribution for the DICOM voxelized phantom in Geant4 57
5.17 Simulated dose distribution for the DICOM voxelized phantom merged with the CT image used in Geant4 58

List of Tables

1.1 Research planning systems developed at research institutions. 5

4.1 Density range and material relantionship 39

4.2 Usage of public functions defined in the IAEA classes 41

5.1 Angular direction of the simulated beam 54

Acronyms

ASCII	American Standard Code for Information Interchange
CPE	Charged Particle Equilibrium
CSDA	Continuous Slowing Down Approximation
CT	Computed Tomography
DICOM	Digital Imaging and Communications in Medicine
EGS	Electron Gamma Shower
ETRAN	Electron TRANsport
FUCAM	Fundación del Cáncer de Mama
Geant	GEometry ANd Tracking
HU	Hounsfield Units
IAEA	International Atomic Energy Agency
ICRU	International Commission on Radiation Units and Measurements
IMMS	Instituto Mexicano del Seguro Social
IMRT	Intensity Modulated Radiation Therapy
KERMA	Kinetic Energy Released per unit MAss
MC	Monte Carlo
MCNP	Monte Carlo N-Particle
MU	Monitor Units
NSCLC	Non-Small Cell Lung Cancer
OAR	Organ At Risk
PB	Pencil Beam
PDD	Percentage Depth Dose
PENELOPE	Penetration and ENErgy LOss of Positrons and Electrons
phsp	phase space
PMMA	Polymethylmethacrylate (Acrylic)
PTV	Planning Target Volume
RTP	Radiotherapy Treatment Planning
SBRT	Stereotactic Body Radiation Therapy
SSD	Source to Surface Distance
TPS	Treatment Planning System

1

Radiotherapy

The goal of radiotherapy treatment of cancer is to give a tumoricidal dose to the tumor while minimizing complications in healthy tissues [1]. Depending on the type and progression of the tumor, radiotherapy can be used alone or in conjunction with surgery or chemotherapy. External radiation therapy is the most common form of radiotherapy. The radiation is usually given by a linear accelerator that can produce x-rays, electrons, protons, etc. The treatment schedule, number of fractions, total dose of radiation and type of treatment depends on the type of cancer, and they are decided by the physician [2] [3] [4].

The radiotherapy treatment planning (RTP) has had a considerable advance in the last 40 years, especially in calculation algorithms [5]. RTP demands a calculation method that can accurately determine the dose distribution in the region of interest. For performing the dose calculation a set of data, such as standard depth dose and profile charts, are required by the engine. This data can be obtained in the commissioning process of the treatment machine or linear accelerator as shown in Figure 1.1 [4].

1.1 The treatment planning process

Radiotherapy treatment planning is a broader process that encompasses all the steps involved in planning a patient's treatment: from diagnostic to treatment verification and delivery. It is therefore a complex process that involves the effort of many departmental personnel [1] [2] [6].

Figure 1.1 Linear accelerator Novalis-Tx

After imaging and diagnostic, the initial step in the RTP process is the simulation. The optimum patient position and immobilization for the tumor localization is determined, and it is necessary to maintain this position during the treatment. The simulation requires specialized equipment like computed tomography (CT) simulators or scanners [1].

Posterior to the simulation, the treatment planning or dose planning step begins. This step requires a computerized planning system (computer software) which includes a graphical display, input and output devices, for entering and hardcopy printouts of the patient information and treatment. The beam number, size, dose, and arrangement are calculated and optimized according to the technique, type of therapy, the dose to the region of interest and normal structures. A radiation oncology physicist evaluates the final dose distribution within the patient, and then the radiation oncologist approves it [1] [2].

Before the treatment delivery, a treatment verification must be performed to check the accuracy of the planned treatment. This includes a dosimetric and geometric verification, for the evaluation of dose calculations and cover issues related to patient

positioning and immobilization, respectively [2]. The radiation dose must be within 5% of the prescribed dose, according to with the International Commission on Radiation Units and Measurements [7].

Lastly, after the final approval of the treatment plan, the radiotherapy treatment is delivered according to the schedule specifications of the radiation oncologist [1].

1.2 Dose calculation methods for treatment planning

Radiotherapy treatments use the information calculated by the treatment planning systems (TPS). The clinical outcome can be improved if accuracy in the dose calculation is further improved [8]. Therefore, the accuracy of dose calculations is crucial to the quality of treatment planning and consequently to the doses delivered to patients undergoing radiation therapy [9].

There are various TPS algorithms for dose calculation in radiotherapy, for instance: Pencil Beam (PB), superposition, Clarkson, Monte Carlo method, etc. A common limitation for photon dose calculation in TPS is to ignore the impact on electron transport, and photon scatters from patient heterogeneities [10]. The analytical pencil beam algorithm produces the biggest error near small high-density heterogeneity [2] [11]. Ignoring the presence of tissue heterogeneities when planning the isodose distribution may produce errors which exceed 30% of the prescribed dose [12].

The Monte Carlo (MC) methods have shown to be the most accurate method for radiotherapy dose calculations. It considers all aspects of photon and electron transport within a heterogeneous media [8]. MC is based on a statistical simulation of random sampling [2]. However, generating an acceptable statistical dose distribution comes accompanied by an increase of time for performed the calculation [8].

1.3 Monte Carlo radiotherapy treatments

Advances in computer technology in the last decade have significantly reduced the calculation time required for the treatment planning system in radiotherapy. Because the superior accuracy of MC method over the deterministic models, several researchers have suggested that MC will become the dominant method for TPS dose calculation [13] [9].

The strongest motivation for the need for MC dose calculation comes from the treatment planning for lung cancer [9]. Pencil beam algorithms, for example, overestimate the dose in the lung. Depending on the location and size of the tumor this could lead to a severely underdosed lung tumor and also affected normal tissues, particularly healthy lung. The differences between deterministic algorithms and MC methods, for lung cancer treatments, might be as big as 20% compared to the prescribed dose [9]. Evaluate the dose distributions becomes significative when the accuracy of dose distributions for tumors and normal tissues clinically impact the outcome of the treatment. For this reason, the evaluation of new techniques, such as stereotactic radiotherapy for the treatment of early stage lung cancer, using MC methods are emphatically encouraged by many authors [3] [8].

1.3.1 Monte Carlo for deterministic algorithm modifications

The deterministic algorithms are the most implemented algorithms in the commercially available treatment planning systems. Recent investigations point towards a planning process in which Monte Carlo Method is used in combination with a standard deterministic TPS, or as a reevaluation tool of clinical studies, especially if an unclear result of the deterministic calculation is expected. The implementation of MC dose calculations has taken place in many research institutions. The research institutions that use research planning systems based on the MC method are enlisted in Table 1.1.

Table 1.1: Research planning systems developed at research institutions.

Research institution	MC algorithm and implementation
University of California, Los Angeles [5]	MCNP4 - In a standard lattice geometry photon particle transport is optimized using a user-friendly interface.
Memorial Sloan-Kettering Cancer Center [14]	EGS4 - For a MC full linac simulation, the computation gives information about the primary and scatter radiation of the treatment head.
Stanford University and Fox Chase Cancer Center [15]	EGS4 - The MC algorithm EGS4-based is called MCDOSE uses phase space files and is coupled with the FOCUS system by Elekta.
Virginia Commonwealth University [16]	EGSnrc - Evaluates the radiation transport through the multileaf collimator.
University of Michigan [17]	EGSnrc - Calculate the dose with the code DPM based on phase space files.
McGill University [18]	EGSnrc - An enviroment, based on MC calculation, models a treatment planning system.

1.3.2 Monte Carlo radiotherapy treatments in Mexico

From the first Cobalt unit in 1949, the evolution of radiotherapy in Mexico has seen a fast growth [19]. Nowadays broad medical centers and hospitals can offer radiotherapy to cancer patients, a list of them are presented below:

- Hospital General de México
- Instituto Nacional de Cancerología
- Instituto Mexicano del Seguro Social, IMSS
- Centro Oncológico de Chihuahua
- Fundación del Cáncer de Mama, FUCAM
- Hospital Regional de Alta especialidad de Ixtapaluca
- Hospital Médica Sur
- Hospital Ángeles
- Instituto Nacional de Ciencias Médicas y Nutrición Salvador Zubirán
- The ABC Medical Center, Cancer Center
- Instituto Nacional de Neurología y Neurocirugía

The radiotherapy treatment to be delivered is planned in commercially available TPS as iPlan - BrainLab, Varian - Eclipse, Phillips - Pinnacle, Elekta - XiO, etc. From all the various TPS only Brainlab from iPlan include a Monte Carlo method for planning. The other deterministic methods usually include correction algorithms for heterogeneous media. Besides the long calculation times when planning with MC, the implementation of MC method in a commercially TPS comes along with a substantial increment of the price. In Mexico City, only two medical centers have a license for the use of MC in TPS: The ABC Medical Center, Cancer Center and the Instituto Nacional de Neurología y Neurocirugía.

1.4 Objectives and scope of the thesis

The scope of this work is to determine the feasibility of applying a Monte Carlo simulation for lung cancer as an analysis and research tool for clinical comparisons. Simulations of water phantom for commissioning purposes and a heterogeneous phantom are required. To simulate real data of an accelerator used in radiotherapy, it is recommended to use the phase space data for external beam radiotherapy properly validated provided by the IAEA phase space database.

The implementation of a Monte Carlo algorithm for clinical comparisons will require to correlate, analyze and modify the calculated doses provided by the treatment planning system. Any modification to the treatment plan should be carried out with major care, regarding the clinical experience from the medical physicist. The implementation requires that the medical physicist understands every possible pitfall associated with the Monte Carlo clinical implementation. The Monte Carlo simulation proposed does not intend to replace a conventional treatment planning system but serves as an analysis tool for clinical comparisons.

Future work will concentrate on performance modifications, including the simulation of more particles, benchmarks, and clinical comparisons.

2

Physics and clinical aspects of photon beams in Monte Carlo simulation

Over the past few years, the growth of speed processing at a lower cost along with the availability of general purpose Monte Carlo software have contributed to the accelerated increment of the employment of Monte Carlo techniques. Still, the application of MC software in clinical applications consumes greater time than the analytical approaches. The range of applications of MC simulations in medical physics is vastly diverse; including applications as brachytherapy dosimetry, calculation of radiation protection quantities, particle transport problems in radiotherapy and validation of commercial treatment planning systems for radiotherapy. MC provides an accurate transport statistics of the resulting particles through the patient, considering the internal structures, heterogeneous media, etc. [20] [21]

2.1 Monte Carlo codes

Monte Carlo method is the statistical approach that uses the probability distributions, governing the individual ionizing particles in a medium, to simulate every random particle path [22] [20]. In the matter of radiation therapy, MC can compute the average values of macroscopic quantities by simulating a huge number of histories. The simulation of the particles trajectories models the probability distributions governing the random nature of the physical processes involved. The simulation of the random trajectories of every particle is calculated by the generation of random numbers [20] [23].

The most widely used general purpose Monte Carlo codes in radiation therapy are

EGSnrc [24], MCNP [25], Geant4 [26] and PENELOPE [27]. MCNP (Monte Carlo N-Particle) was originally developed as software for neutron-photon transport for reactor calculations, after the implementation of electron transport in 1990, it became a significant participant in the medically related fields. Also, MCNP can simulate more than 34 different kinds of particles along with more than 2000 heavy ions of practically all energies. EGS (Electron Gamma Shower) and PENELOPE perform the simulation of coupled transport of photons and electrons. Because EGS was developed specifically for medically related simulations, it has experienced a significant grown in medical physics, especially for low energy dosimetry. Lastly, MCNP and EGSnrc have played a dominant role in the radiation treatment planning and dosimetry research, Figure 2.1 [22] shows the relationship between the number of papers published by EGS and MCNP. Besides, EGS and MCNP have shown approximately the same efficiency calculation in simple geometries when there are not variance reduction techniques [9]. Presently, Geant4 has also present a significant growth, matching the medical-related contributions of MCNP. Geant4 code was originally developed for simulating particle physics. Similar to MCNP, it has successfully implemented applications for radiation therapy related fields. Geant4 can simulate neutrons, protons, pions, etc. Still, it has been demonstrated that photon transport is practically identical in the energy range of radiotherapy [28].

2.1.1 Monte Carlo components

Every Monte Carlo general purpose code consists of four components:

- the cross section data,
- the algorithms for particle transport,
- the specification of the geometry, and
- the output information.

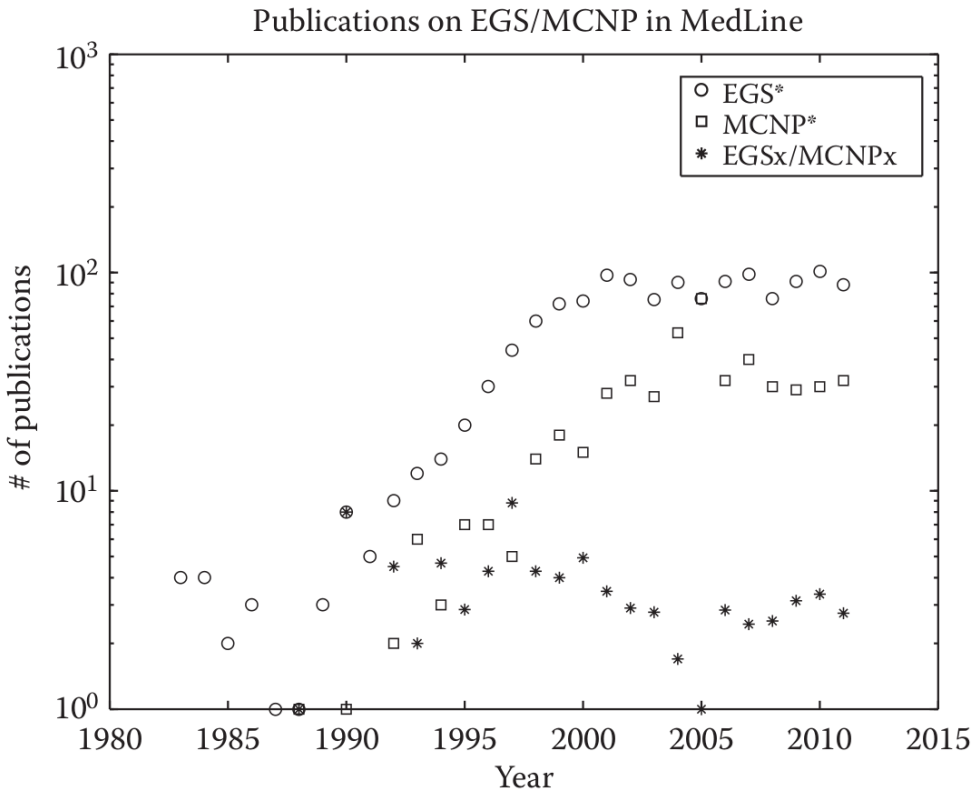


Figure 2.1 The number of papers published by the different versions of EGS*, MCNP*, and their relationship (the comparison is performed by similar versions of the EGS and MCNP codes).

The cross section data and the algorithm used can influence the physics of the simulation highly, while the geometry and the information of interest required by the simulation have a significant impact on the running time. Each MC general code manages their cross section data, based on two different algorithms, to be discussed later. It must be emphasized that every MC dose determination must be carefully benchmarked. For the case of energy lost by the photon interaction, the cross-sections employment becomes crucial. The quality of the calculated results depends strongly on the cross section utilized [20].

2.2 Modeling of photon transport

As a result of the inherently stochastic nature of Monte Carlo, each photon simulated passes through a series of interactions whose outcome is not predictable beforehand.

Each history is computed to be an independent random event. The code models the distance to a collision, the interacting material, and the most probable interaction, along with the probability distribution based on the cross section intrinsic of every code. The relation between the relative importance of the three principal modes of photon interactions are shown in Figure 2.2 [29].

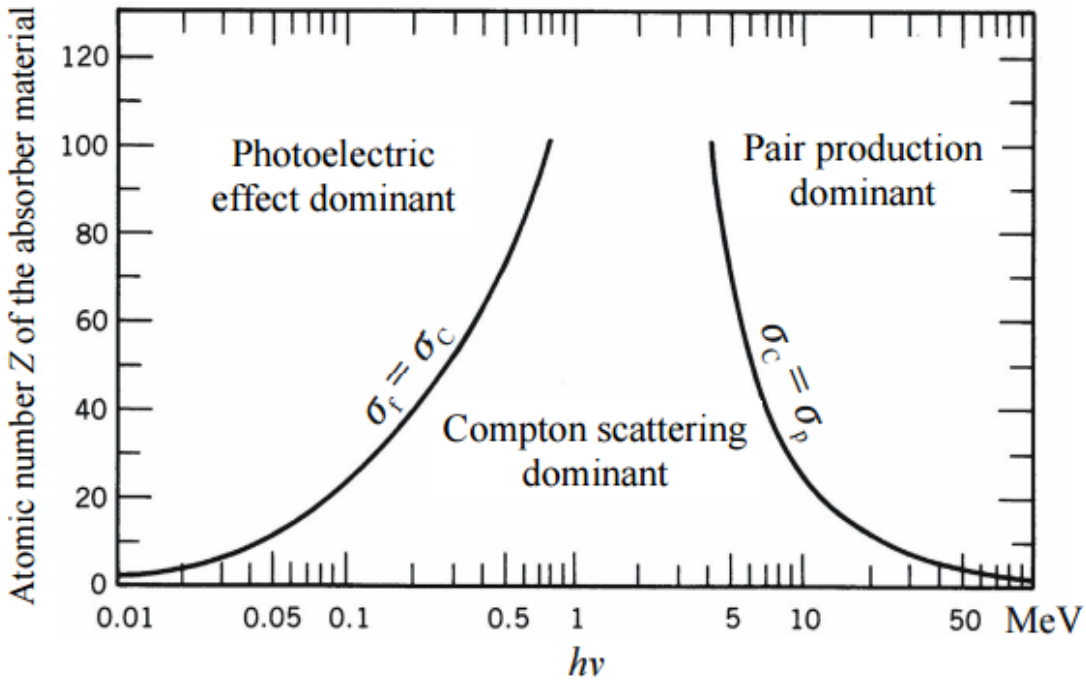


Figure 2.2 The relative importance of interaction plotted as a function of photon energy and the atomic number of the material.

Due to the range of energies and the atomic number for its occurrence, the most common photon interaction in radiation therapy is the Compton interaction. Compton interaction occurs when a photon interacts with a free electron, or in a loosely bound orbital. The photon energy is then transferred to the electron as kinetic energy, with the remaining energy carried out by another photon. The Compton effect leaves the atom in an unstable state, leaving the atom with a vacancy. Figure 2.3 ilustrates the Compton effect.

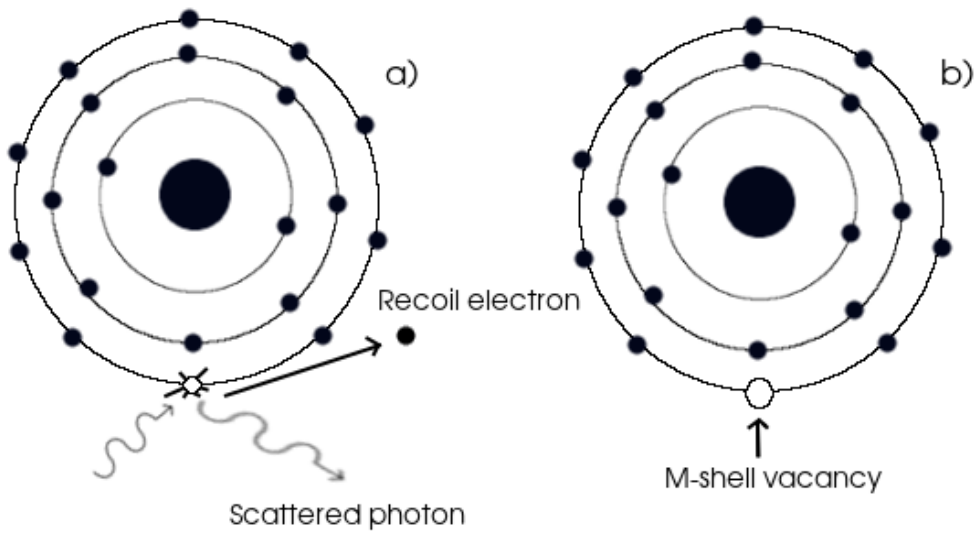


Figure 2.3 (a)Compton scattering of a photon from electron, as a result of the collision, the photon is scattered leading to (b) an ionized atom.

Other less probable physical interactions in medical simulations are the photoelectric effect and pair production. For lower photon energies the photon is absorbed by the atom, and a photoelectron is ejected, this process is called photoelectric effect, leaving a vacancy in a shell of the atom. Thus, the binding of the electron is essential for its occurrence. The probability for a photoelectron to be produced depends highly on the atomic number Z and the energy $h\nu$ of the photons. The probability varies as $Z^4/(h\nu)^3$. For energies above a few mega electron volts, the dominant interaction is the pair production. A photon with energy $h\nu \geq 2mc^2$, meaning at least twice the electron rest energy, the photon interacting with the field nucleus is absorbed creating an electron-positron pair. The probability varies with the atomic number as Z^2 [20].

To determine the quantities of interest required by Monte Carlo simulation, MC samples the photon probability distribution by the generation of random numbers as shown in Figure 2.4 [20].

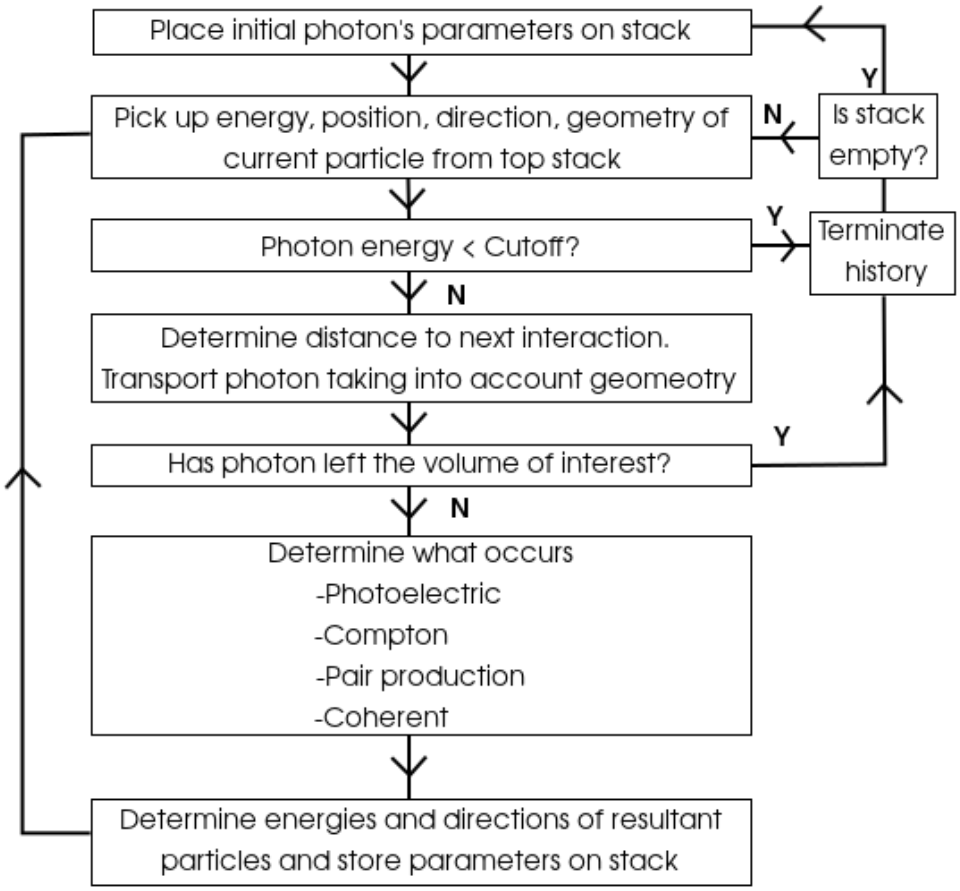


Figure 2.4 Logic flow of photon transport in Monte Carlo simulation.

Initially, the photon is carried to the location of the first interaction, considering the most probable type of interaction based on the partial cross section for the different interactions. Once the interaction is selected is then simulated based on the algorithms describing the kinematics of the interaction. For example, if the Compton effect is simulated, the energy and direction of the photon sampled and the unbound electron are computed. The electron simulation is then put in "stack" for its later computation. The photon sampled is computed to the next interaction until it gets absorbed and the secondary particles got transported.

2.2.1 The anatomy of a history

If a photon with energy E collides with a homogeneous medium, the probability of interaction with the medium after a length l is

$$p(s)dl = \mu(E)e^{-\mu(E)l}dl, \tag{2.1}$$

where $\mu(E)$ is the linear attenuation coefficient of the medium. The coefficient $\mu(E)$ depends on the medium where the photon travels and the photon energy. The total linear attenuation coefficient is the sum of the individual contributions of photoelectric effect, Compton effect and pair production as $\mu(E) = \mu(E)_{photoelectric} + \mu(E)_{comptom} + \mu(E)_{pairproduction}$ [30]. The calculation of the distance to the location of the first interaction l is derived from the cumulative probability curve $P(l)$ that considers the probability of an interaction between the travel distance l and $l + dl$ given by the expression $\mu e^{-\mu l}dl$. The cumulative probability is then calculated by

$$P(l) = \int_0^l \mu e^{-\mu s}ds = 1 - e^{-\mu l}. \tag{2.2}$$

In this set of equations μ is the total cross section given in cm^{-1}.

If in the simulation throughout a heterogeneous media the compute distance exceeds the distance to the nearest media medium, the photon is then transported directly to the boundary. Once the photon gets to the boundary, the next interaction is computed using the cross sections of the new interacting medium. The track of the photon is computed within the new medium without changing the flight direction [28] [23].

2.3 Modeling of electron transport

While the simulation of photons in MC undergo a small number of discrete interactions per track, the simulation of electrons may be subject to a large number of interactions as they pass through matter. The MC simulation of every electron interaction is then considered unrealistic. Consequently, the condensed history technique is employed, where the interactions are divided into two different models. The two models simulate

the electron energy loss in a variety of ways. These codes are referred as class I and class II algorithms [23] [20] [28].

2.3.1 Condensed history technique

All the general purpose Monte Carlo codes mentioned are based on the condense history technique, developed by Berger and Rogers and Bielajew in 1963 and 1988, respectively [22] [28] [20].

The class I condense history approach considers that the primary electron is not influenced by the generation of a secondary electron. The creation of secondary electrons is modeled explicitly to simulate the energy loss in each step. Therefore, the energy losses and angular deflections, associated with individual events, are grouped together. MCNP and ETRAN are examples a class I type MC code [22] [28] [20].

Class II condensed history approach groups soft interactions into one step. The catastrophic interactions, or hard collisions, are simulated independently. The threshold between soft and catastrophic is determined by the accuracy of the requirements and the physics of the problem. EGSnrc, Geant4, and PENELOPE are examples of class II MC codes [22] [20].

It must be noticed that for modeling radiative energy losses, codes as ETRAN and MCNP, use a class II algorithm. Although, class I include complete energy-lose, independent of the electron transport cut-off, because of the correlation between primary and secondary particles, class II algorithms claim to be more accurate than the class I. Nevertheless, this correlation only became significant in very specific circumstances, and rarely can lead to real significant differences. The logic flow of electron transport for class I and II algorithms are shown in Figure 2.5 [23] [20].

2.4 Benchmarking

Due to the importance of MC quantities to model medical related problems, it is imperative to benchmark every MC code. The benchmarking is computed by comparing calculated output data with high-quality experimental data or properly evaluated MC codes. To estimate the statistical uncertainties of the code, it is

advisable to compare with data provided by simple geometry, for obtaining the most substantial comparisons. Comparison with the percentage depth dose (PDD) and profiles beams in water is the most suitable comparison for validating the MC code, quantities that will be described in the next chapter [20].

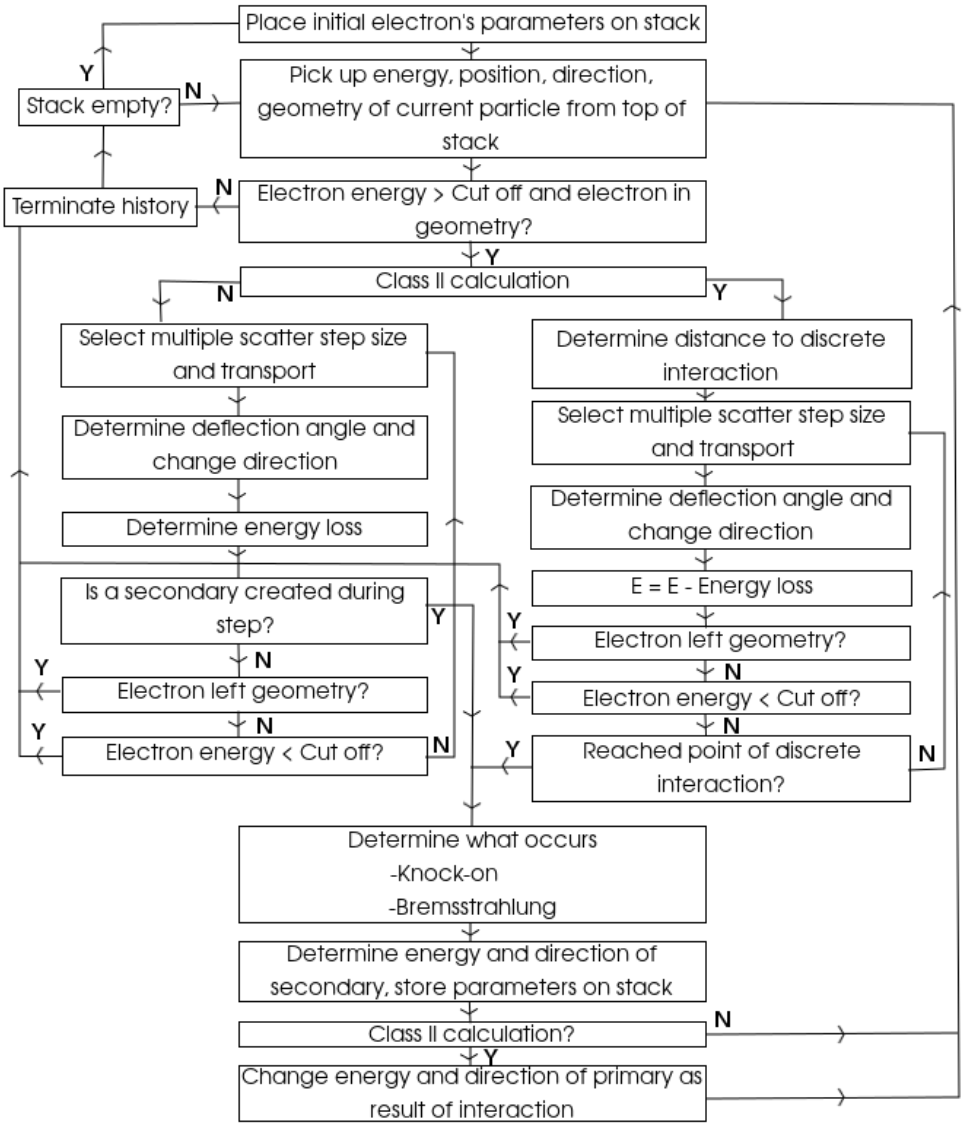


Figure 2.5 Logic flow for class I and II Monte Carlo algorithms for electron transport simulation.

3

Monte Carlo treatment planning systems in radiotherapy

In the last few decades, dose calculation algorithms have evolved from handmade empirical calculations to more accurate analytical codes used in the treatment planning systems (TPS). To deliver an accurate treatment, TPS have improved further, starting with heavily analytic algorithms, as Pencil Beam, to more accurate codes as convolution-superposition, Clarkson, and Anisotropic Analytical Algorithm.

Since the first appearance of MC in radiotherapy in the 1970s, several authors coincide that Monte Carlo method is the most accurate method for modeling radiation transport in radiotherapy. Because of its stochastic nature, MC is the only method capable of account all the aspects of particle transport within a heterogeneous medium. Nevertheless, the long calculation time needed to reach an accurate and precise dose distribution is still a disadvantage over other analytical methods [28] [22].

3.1 Beam Modeling

An accurate Monte Carlo simulation is only achievable when the radiation beam is correctly modeled. For MC photon treatment planning, several beam models have been developed. The most common models for beam simulations are the full linac simulation and the phase space simulation.

3.1.0.1 Full linac simulation

Full linac simulation strongly depends on the complete knowledge of the components of the linac. Information about the material composition, the dimensions of the components and their geometrical configurations must be accurately modeled. The simulation of a linac photon beam begins with an electron beam as a primary source. The electron source enters the linac head, hitting the target and producing bremsstrahlung photons that pass throughout a primary collimator, the flattening filter and finally by secondary collimators and beams modifiers. Usually, the simulations of the monitor ion chamber and the field mirror are not typically considered in MC linac simulations [22] [28]. A visual representation of the linac components simulated is shown in Figure 3.1 . Despite the advances in computer power, full linac simulation still is extremely long, for most of the calculation time is spent in the simulation in the linac head.

3.1.0.2 Phase space simulation

The data used in the phase space (phsp) is obtained by an accurate simulation of a full linac simulation. A proper simulation of the phase space can entirely replace a full linac simulation and become a "virtual linac". The geometry location of the phase space file is shown in Figure 3.1 . The phase space file is generated at the bottom of the linac and contains all the parameters of a vast number of particles that exit the linac [22] [28]. The information stored in the phsp files is:

- the position of the particle (x, y, z)
- the direction cosines (U, V, W)
- the kinetic energy of the particle E
- the particle statistical weight
- the type of particle: photons, electrons, positrons, protons, and neutrons
- the history of the particle, if the particle belongs or not to a new history

Because the information required by the phsp file comes from an MC linac simulation, it is imperative that all the results are extensively evaluated. The International Atomic

Energy Agency (IAEA) has developed a project for creating a characteristic phsp data of accelerators used in medical radiotherapy by compiling existing data that have been properly validated [31]. The IAEA project allows users to access to phase space files from different commercial therapy units adaptable for MC simulation. Despite the significant amount of storage required by the phsp files, the phase space beam modeling still being advantageous as it is not always possible to obtain the linac information from the vendors.

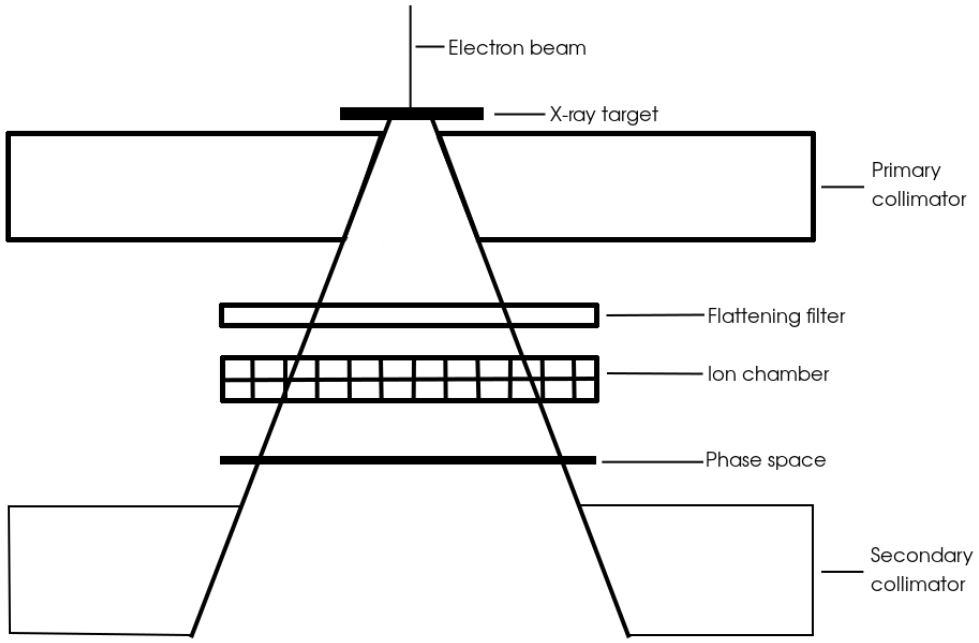


Figure 3.1 Full linac simulation components and location representation of the phase space.

3.2 Benchmarking, uncertainties and validation of the treatment

The evaluation of Monte Carlo codes is made by comparing the computed data against high-quality experimental data. As a first approximation, comparisons against other Monte Carlo codes provides a nominal way to demonstrate the accuracy of the calculations. To ensure precise comparisons, all the parameters used in every Monte Carlo code must be the same. For fully linac simulations the calculated and experimental comparison is compulsory to eliminate coding errors and to estimate the

uncertainties associated with the approximations made in the MC code. The IAEA phase space simulations for linear accelerators have already been properly validated [20] [31].

Although phsp simulations are previously validated, measurements like PDD and lateral profiles must be performed to confirm the proper implementations of the phsp in the MC code [20].

To provide a specific benchmark, it is advisable to reproduce a "typical" case, easy to implement between codes and other algorithms. The recommended voxel size is $5 \times 5 \times 5 \times \ mm^3$ to ensure a voxel on a central axis and to represent a realistic size. The voxels are to be filled with water. The benchmark beam should be collimated to $10 \times 10 \ cm^2$ and utilize the energy required for the dose calculation [20].

3.2.1 Percentage Depth Dose

The characterization of the central axis dose distribution is performed by normalizing the dose at depth with the dose at a reference depth. The dose distribution $D_{max} = 100\%$ is normalized at the depth of dose maximum z_{max}. The percentage depth dose (PDD) is defined as $PDD = 100D_d/D_{d_0}$, where D_d is the absorbed dose at the depth d and $D_{d_0} = D_{max}$ the reference depth. The parameters that affect the PDD are the depth in a phantom, the field size, the source to surface distance and the beam energy. The calculated PDD for various energies are shown in Figure 3.2 [32] [33].

3.2.2 Beam profiles

Information of the off-axis data is measured by beam profiles perpendicular to the central beam axis at a given depth. The usual parameters for the beam profiles measurements at reference depth are usually 10 cm or z_{max}, a field size of 10×10 cm^2 at the energy of interest. Beam profiles are part of the information provided by the off-axis ratio, defined as the measurements of ratio of dose at an off-axis point to the dose on the central beam axis at the same depth [33].

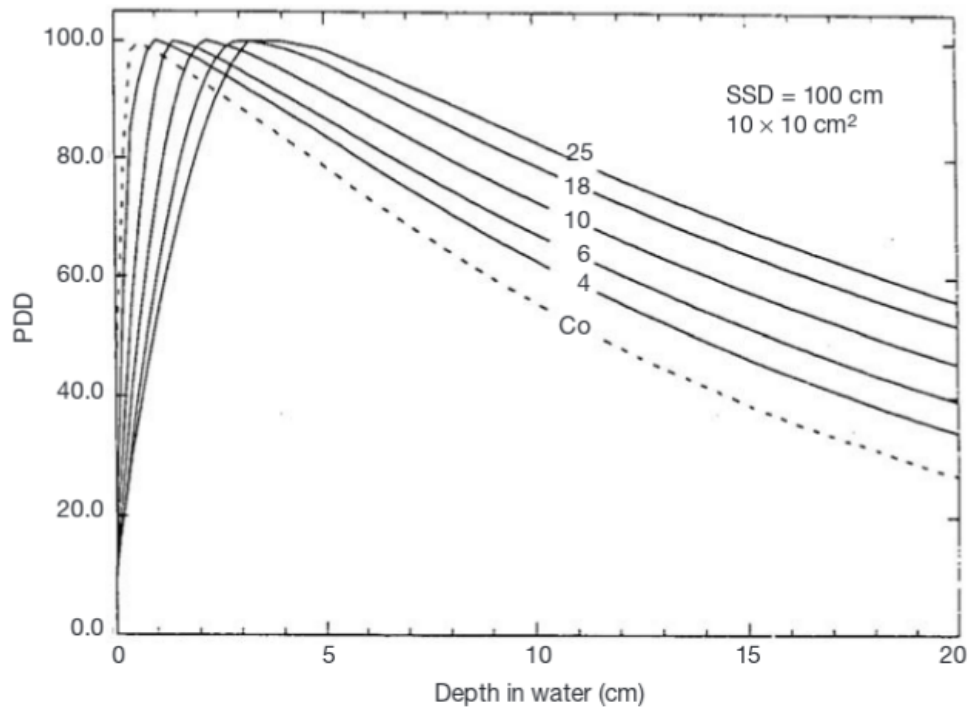


Figure 3.2 PDD curves in water for a 10x10 cm^2 field in a SSD of 100 cm for different photon beams.

The uniformity of the dose profiles is characterized by the beam flatness and symmetry. The beam flatness F is calculated by

$$F = 100 \times \frac{D_{max} - D_{min}}{D_{max} + D_{min}}, \tag{3.1}$$

where D_{max} and D_{min} are the maximum and minimum dose on the beam profile, respectively, in the central 80% of the beam width. At usual parameters $F \leq 4\%$ is recommended [33] [32].

3.2.3 Heterogeneous phantoms

Percentage depth dose and beam profile measurements are performed in homogeneous water phantoms. However, in clinical practice, the beam travels through the different patient tissues heterogeneities, and the isodose distributions may vary considerably to the distributions computed in the water phantom. The effects of heterogeneities in the

dose deposition depends on the density and the atomic number of the medium and the quality of the beam. The outcome may be an increase or decrease in the attenuation of the primary beam or the secondary electron fluence [34] [33].

Therefore, before the validation of a particular treatment, it is recommended to perform a benchmark measurement in a heterogeneous phantom. Ma et al. [34] suggest the employment of a PMMA phantom with inserts of different materials for point measurements and dose verification. The heterogeneous phantom can also be employed for the validation of treatment planning as well [22].

3.2.4 Statistical Uncertainties

For the Monte Carlo dose calculation within a patient or phantom, there are two sources of statistical uncertainties: the ones generated from the simulation of the linac and those resulting from fluctuations in the patient or phantom [9]. The latent variance is defined by the uncertainties caused by the statistical fluctuations in the phase space. The outcome of a calculated phase space will be a finite latent variance.

The calculation of the uncertainty $s_{\bar{x}}$ of a scoring quantity X is calculated for a number N of independent events as follows

$$s_{\bar{x}} = \sqrt{\frac{1}{N(N-1)}(\frac{\sum_{i=1}^{N} X_i^2}{N} - (\frac{\sum_{i=1}^{N} X_i}{N})^2)}, \tag{3.2}$$

where X_i is the quantity of interest scored during the history i. The calculation of the uncertainty presented is called history by history method. Particularly, when using phase space beam simulations, X_i might be a weight quantity, while N is equal to the total number of independent histories [35]. Monte Carlo for treatment planning related simulations must be able to offer an uncertainty in dose calculation within 3% [28].

3.2.5 Noise reduction technique

Unless several histories are sampled in an MC simulation, the dose distribution will carry some degree of noise in the result. The approach for denoising an MC dose

distribution is performed using image processing algorithms, similar to image restoration techniques. The method used for denoising an MC image must be carried out avoiding the introduction of systematic errors, especially around beam edges [28].

3.2.6 Variance reduction techniques

Due to the long computational time for MC simulations used in radiotherapy, a variety of techniques have been developed to reduce the calculation time called variance reduction techniques. Although parallel calculations or software improvements could increase the efficiency of MC calculations, do not make the MC algorithm faster. Some of the techniques used in variance reduction techniques are cut-off energies, uniform particle splitting, range rejection, and so on [22].

3.2.7 Verification of the simulated treatment

The verification of the simulated treatment must be performed before the delivery of the treatment. The aim of the verification is to ensure that the calculated dose and the delivery dose are within the tolerance ranges [7] [36]. The techniques and protocols implemented to perform the verification vary from facility to facility. Still, the most common methods include point dose measurements or comparisons of 2D dose distributions. The simulated treatment is transferred and delivered to a dummy phantom for its verification, using an ionization chamber, a dosimetry film or a 2D detector array [37] [38] [39]. Nevertheless, if patient heterogeneity on the dose distribution is not included in the calculation, the calculated and measured dose distributions can differ considerably than the dose distribution delivered to the patient [34].

Monte Carlo method can be used as a tool for verification of radiotherapy simulated treatments. An accurate MC simulation can perform several verifications that cannot be directly measured. MC verification can be simulated in a heterogeneous phantom or, if possible, in the voxelized patient geometry [23] [34].

The verification of the treatment in an MC simulation must be performed under the same conditions as the patient treatment as photon energy, number, direction, and field size of the beams delivered, etcetera [22]. The dose distributions obtained, in a phantom or directly in the patient geometry obtained from the Computed Tomography

(CT), are then compared with the results obtained from the treatment planning system. Because the anatomical information of the patient also affects the accuracy of the final dose distribution, for verification purposes, it is desirable to simulate the complete treatment in the patient CT geometry taking into consideration that it increases the user memory usage greatly [40] [41].

3.3 Dose calculation

The absorbed dose is defined by the mean energy $\bar{\varepsilon}$ imparted by ionizing radiation to matter of mass m in a finite volume V [29], meaning

$$D = d\bar{\varepsilon}/dm. \tag{3.3}$$

The energy imparted ε is defined as

$$\varepsilon = (R_{in})_u - (R_{out})_u + (R_{in})_c - (R_{out})_c + \sigma Q, \tag{3.4}$$

where $(R_{in})_u$ and $(R_{out})_u$ are the radiant energy of uncharged particles entering and leaving V, $(R_{in})_c$ and $(R_{out})_c$ are the radiant energy of the charged particles entering and leaving the volume V, and σQ is the net energy derived from the rest mass in V [29]. Consequently, the mean energy imparted $\bar{\varepsilon}$ is the sum of all the energy entering the volume of interest V minus all the energy leaving the volume, taking into account any mass–energy conversion within the volume. The units of absorbed dose is joule per kilogram (J/kg) of Gray (Gy) [42] [33] [43].

3.3.1 Dose calculation in treatment planning systems

To calculate the absorbed dose within the medium the TPS takes into account the linac calibration measurements. The TPS uses the conversion factors, obtained in the calibration in an open $10 \times 10\ cm^2$ field in water, to associate the dose calculated with the dose measured.

The dose delivered to a patient aiming to a certain point depends on the depth of the point of interest, the penetrating power of the beam, the type of tissue, the

size of the field, and so forth [44]. Because of the physical nature of photons and charged particles used in radiotherapy, it is impossible only to irradiate the tumor. For that reason, in clinical practice, the prescribed treatment is delivered taking into consideration the dose to the target, the dose constraints of the surrounding tissue or organs at risk (OAR), and a safety margin that takes into account setup uncertainties and body motion of the patient [3] [28].

3.3.2 Dose calculation in Monte Carlo simulation

Due to the different characteristic of every Monte Carlo dose algorithm the dose calculation steps highly depend on the type of software used. However, the information required for simulating a radiotherapy treatment plan is the same for all [22]. For instance,

- the beam quality (phase space files) based on the characteristics of the treatment,
- the beam specifications, like number of beams, direction, and position relative to the patient,
- the patient geometry, mass density, and composition (obtained by the CT imaging data set or a similar phantom geometry),
- and the patient-specific beam line devices (blocks, jaws, wedges, etc.).

3.3.2.1 Monte Carlo for deterministic algorithm dose-modifications

As mentioned before, the implementation of MC simulation for the quantifications of differences computed in commercially TPS has taken place in many research institutions. The Monte Carlo calculated distributions are used to quantify dosimetric differences computed in the commercially TPS. An example, of a simplified method using Pencil Beam and Monte Carlo calculation presented by Laub et al. [45] may be summarized as follows:

1. Calculate the treatment in a PB algorithm conventionally.
2. Compute the MC treatment with the beam parameters from the PB calculation in step 1.

3. Compare the dose maps obtained by both calculations. Obtain new beam settings for PB optimization.

4. The calculations end when both algorithms converge, otherwise step two and three must be repeated.

Still, one of the biggest problems in mixing calculation methods is that the optimization may not converge [45]. In that case, Monte Carlo could be used as a tool for optimization of Monitor Units (MU) of the individual beams, where one MU is equal to a specific dose of radiation. Usually, most of the linacs are calibrated to deliver $1MU = 1cGy$ to a depth of dose maximum z_{max}. Then, the treatment plan calculated with a deterministic algorithm can be recalculated with MC, and the resulting MUs are compared [28].

3.3.3 Computer tomography based simulations in Monte Carlo

The most accurate patient information is obtained from the computer tomography data images of the patient. The dose distribution is then stored in the voxels based on the CT information. Nevertheless, the use of CT images comes along with a significant impact on the calculation time due to the time spent in the geometry boundaries checks and scoring.

The relationship between the mass density and material composition on the CT is related to its Hounsfield Units (HU). The Hounsfield Unit is defined by

$$HU = 1000 \times \frac{\mu - \mu_{water}}{\mu_{water} - \mu_{air}} \tag{3.5}$$

where μ is the attenuation coefficient of the material study and, μ_{water} and μ_{air} are the attenuation coefficients of the water and air, respectively. Once the HU is calculated the electron density ρ_e is then computed as

$$\frac{\rho_e}{\rho_{ew}} = 1 + \frac{HU}{1000} \tag{3.6}$$

where ρ_{ew} is the electron density of the water. The mass density ρ can thus be calculated as a linear function of the electronic density ρ_e as

$$\rho = \rho_e \frac{n}{n_w} \frac{\rho_w}{\rho_{ew}} \tag{3.7}$$

where n is the number of atomic units per electron for the medium, and for the water $n_w = 1.802$, and ρ_w is the water mass density [28] [46] [5].

The material composition of each voxel in the CT images is determined according to the HU. The correlation between HU and elemental compositions is commonly divided by the most common materials: air, lung, fat, water, muscle, and bone. This division has proven to be sufficient to obtain an accuracy of 1 % [28].

3.4 Monte Carlo for lung cancer

The biggest differences between the dose distributions calculated with analytical treatment planning systems and MC simulation are found in the heterogeneous medium, especially in low-density tissues. This is the result of the analytical approaches for the calculation of electron transport in different media, for example for lung treatment planning calculations. For this reason, lung Monte Carlo simulation has been pointed out as one of the strongest motivation for analytical treatment planning systems verifications. Analytical TPS can miscalculate the absorbed dose to the lung given an under dosage to the tumor as substantial as 30 %, and for more recent algorithms as 8%, along with problems related to the dose imparted to healthy tissues and normal lung [5] [9]. Furthermore, the utilization of advanced techniques as stereotactic radiation therapy increases the importance of an accurate dosimetry and clinical importance of MC treatment verifications [5].

3.4.1 Monte Carlo simulation for stereotactic lung radiotherapy

Lung cancer is the most common cause of death for cancer in men and the fourth in women in the word. In Mexico, lung cancer is the second most common cause of death by malignant tumors in men and the eighth in women, this represents an increment of 50 % of deaths for lung cancer in only a decade [47]. Non-small cell lung cancer (NSCLC) affects 75-80 % of patients with lung cancer. It is expected that the number of patients diagnosed with NSCLC with an early stage increment considerably [48]. Surgical resection is the standard treatment for early stage lung cancer. However, radiotherapy becomes an option when the patient is medically inoperable.

Stereotactic Body Radiation Therapy for lung cancer (SBRT) is a known invasive cancer treatment that allows the delivery of high doses of radiation hypofractionated, by multiple beams guided by a set of coordinates. The term stereotactic refers to the fiducial markers used as a set of coordinates for planning and delivered the treatment [48]. Some of the most important dosimetric characteristics of SBRT for early stage lung cancer are

- The treatment schedule is 5 fractions of 10 Gy.
- A minimum size field of 3.5 cm is required.
- CT images must allow the view of the fiducial markers along with the patient anatomy.
- Non-opposing, noncoplanar beams are preferable.
- A minimum of seven non-opposing beams should be used.
- The field aperture must correspond to the projection of the Planning Target Volume (PTV) (beam's eye view).
- The use of IMRT should be considered only when the target coverage and limits to the normal tissue are not archivable with 3D conformal planning.
- The use of Clarkson or Pencil Beam algorithms must be credentialed before the delivery of the treatment.
- The acceptable criteria across the PTV is 5% of dose difference or 5 mm of the distance to agreement criteria.

- The treatment plan should be normalized such that 100% corresponds to the center of mass of the PTV.
- The prescription isodose surface will be chosen such that 95% of the target volume (PTV) is conformally covered by the prescription isodose surface and 99% of the target volume (PTV) receives a minimum of 90% of the prescription dose.
- The prescription isodose surface selected must be less than 60% of the dose at the center of mass of the PTV and more or equal 90% of the dose at the center of mass of the PTV.

4

Monte Carlo method

This chapter describes the various Monte Carlo tools and methods used for the simulation. The simulation procedure includes the phase space beam modeling, the validation of the computed beam, the phantom construction, the simulation based CT-images, and the process to calculate and read the dose distribution information.

4.1 Monte Carlo N-Particle code

Monte Carlo N-Particle (MCNP) is a general purpose Monte Carlo code that can simulate several particles, including photons, electrons, neutrons, coupled neutron/photon/electron, heavy ions, etc. The energy particle simulation capability varies from particle to particle; the energy range simulation for photons and electrons is from 1 keV to 1000 MeV, and for neutrons is from 10^{-11} MeV to 20 MeV. MCNP includes an extensive collection of cross-section data, flexible tallies for dose calculation, tally plotters, and a large group of variance reduction techniques [49].

MCNP is a type I class algorithm. The algorithm follows the path of many particles from a source throughout its life to its death by absorption, annihilation, etc. MCNP photon simulation can sample incoherent and coherent scattering, absorption in pair production with a local emission of annihilation radiation, bremsstrahlung, and the possibility of fluorescent emission after photoelectric absorption. MCNP electron simulation follows a continuous-slowing-down approximation (CSDA). The CSDA is used for the simulation of positrons, k-shell x-rays, and bremsstrahlung.

4.1.1 MCNP input file

MCNP reads the information created in an input file, containing the geometry specification, the material description, the source characteristics, the type of output information desired (tallies), and the variance reduction techniques to be utilized.

The structure of the MCNP input file is

Title card
Cell cards
blank line delimiter
Surface Cards
blank line delimiter
Data card
blank line terminatior (optional)

The geometry specifications are taken from the surface and cell cards. The surfaces used to create the geometry of a problem are written on the surface cards.

MCNP geometry is defined regarding regions or volumes constructed by intersections, unions, and region or volume complements. The boolean operator between all volumes is described in the cell card. The cell card also includes the information related to the mass density of the geometry created.

The MCNP data card contains all the information related to the type of particle, the material chemical composition, the radiation sources, the desired output information (tallies), the cross section libraries, the variance reduction techniques to be utilized, etcetera. The material specification needs to include the material number, the precise chemical designation of the element that compound the material and the elemental composition. The material density is specified in the cell card. For example, input line of the material information for water is

$$M1\ 1000\ -0.11190$$
$$8000\ -0.88810$$

Where the first, second and third entry indicates the material number, the isotope ZAID number (that contains six digits or less ZZZAAA in which ZZZ is the atomic

number Z and AAA is the atomic mass number A) and the elemental mass fraction of hydrogen and oxygen, respectively.

The command SDEF describes the source and type of radiation particles and is uses to define all the characteristics of the source problem. The source simulation comprises isotropic sources, isotropic sources with histogram or continuum of energies, isotropic volumes sources, line sources, plane sources, collimated sources, etcetera.

The tally commands are used to specify the type of output information desired, as surface current, the average flux in a cell, energy deposition, etcetera. The tallies are identified by tally and particle; every tally works with specific particles. All the tally results are normalized to be per starting particle.

4.1.2 Phantom construction

Two phantoms were simulated in MCNP, a water phantom for beam calibration purposes and a heterogeneous lung phantom. The parameters applied for every simulation as field size, depth in water or the phantom are determined according to the measurement.

The water phantom simulated consisted of a water tank of $675 \times 645 \times 560\ mm^3$, the acrylic wall thickness of 15 mm, equivalent to a real phantom, were also simulated. For the PDD and the beam profile, the depths of measurements are simulated at z_{max} (of the energy of 6 MeV used) and 10 cm, respectively. The tallies created depend on the output data required. The simulation characteristics of every calculation were as follows,

- For the PDD simulations, the tallies were constructed along the z axis, from the surface of the simulated water to 30 cm depth. The field size used was $10 \times 10\ cm^2$. The PDD geometry constructed in MCNP is shown in Figure 4.1 .
- For the beam profiles simulation, the tallies were constructed along the x and y axis, in the range of $(-7\ cm, 7\ cm)$ for both cases, for a 10 cm depth. The field sizes used were $10 \times 10\ cm^2$ and $4 \times 4\ cm^2$.

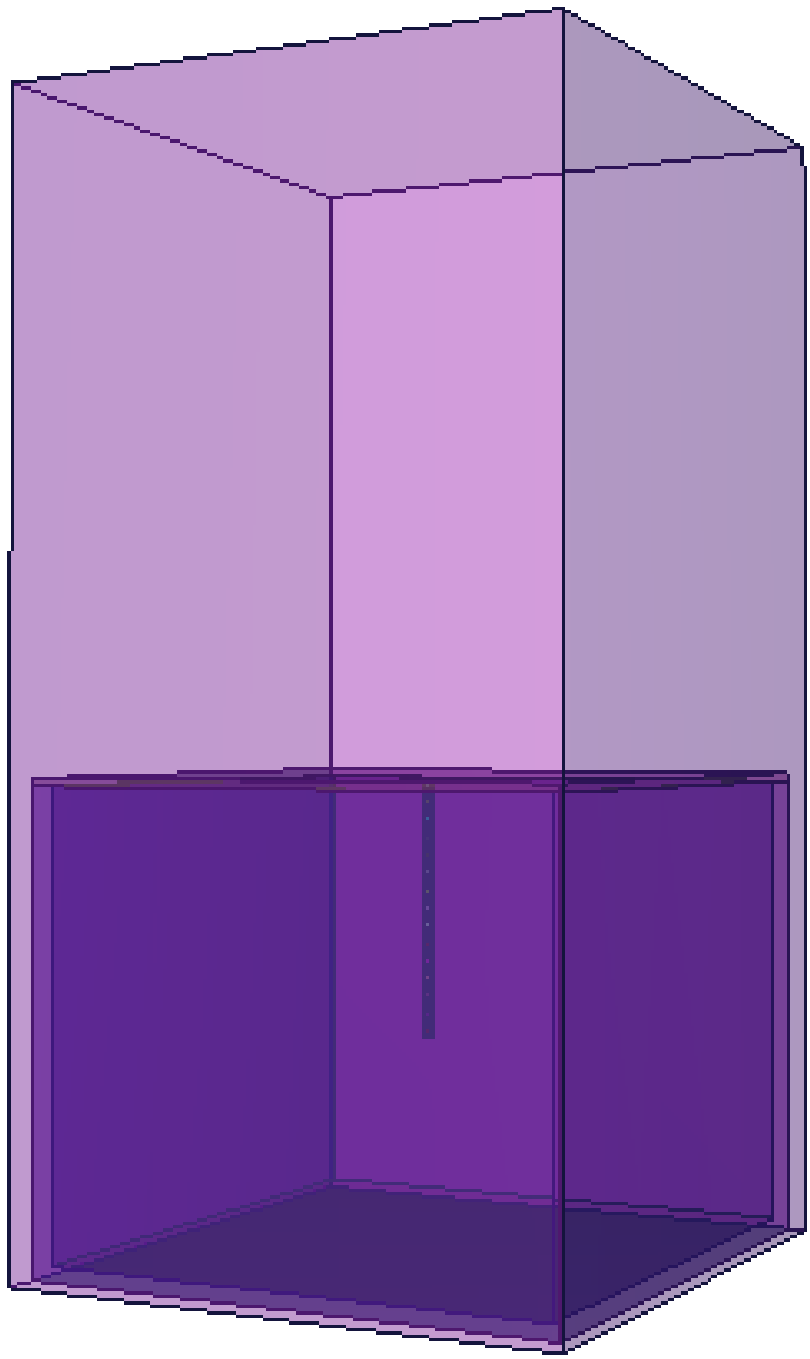
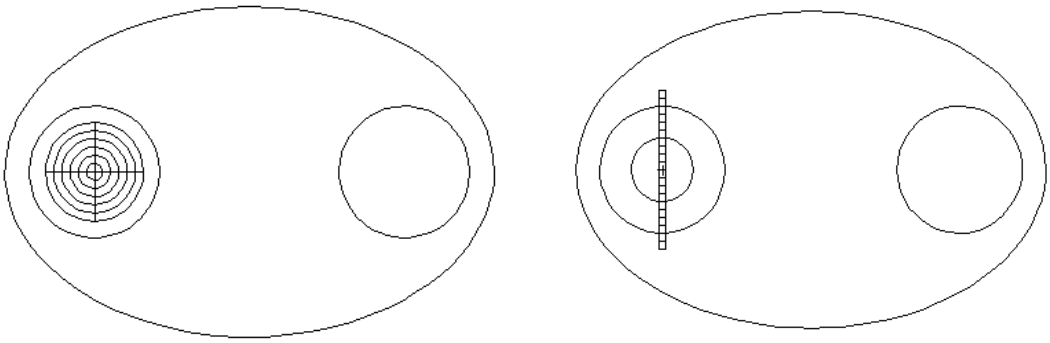


Figure 4.1 Water phantom simulation for PDD measurements, the tallies are contructed along the z axis.

The heterogeneous phantom simulations consisted of a simulation of a real phantom called $QUASAR^{TM}$ Multi-Purpose Body Phantom [50]. The phantom is constructed of a solid acrylic with two cylindrical lung inserts. The main body component is an oval shape of 20 cm high, 30 cm wide and 12 cm long. The cylindrical lung inserts dimensions are 8 cm diameter and 12 cm long. In addition to the $QUASAR^{TM}$ simulation, a "virtual" cylindrical tumor was simulated in the center of a lung insert. The isocenter was simulated to coincide with the center of the tumor. The field size was computed according to the width of the tumor, 4×4 cm^2, and it was simulated to be directed only from above the phantom. The energy used was 6 MeV. Two forms of tallies were created, circular tallies including the tumor and part of the lung and tallies along the z axis that include the body of the phantom, part of the lung and the tumor, both geometries are shown in Figure 4.2 .



(a) Circular tallies.

(b) Tallies along z axis.

Figure 4.2 QUASAR phantom.

4.1.3 Beam simulation: phase space

The beam simulation takes the phase space information from the International Atomic Energy Agency (IAEA) [31]. The beam simulation in MCNP was done in two stages, the conversion of the phase space data from IAEA-phsp to ASCII format, and the implementation of an MCNP input.

The phase space data is written in the IAEA phsp format, that has not a native implementation for MCNP. IAEA format cannot be directly open by any program, and its interface is only available for MC codes as Geant4, EGSnrc and PENELOPE [31]. The generic converter from IAEA to ASCII was used to extract all the phase

space information. The information obtained from the phase space was: the position of the particle (x, y, z), the direction cosines (U, V, W), the kinetic energy of the particle E, the particle statistical weight, the type of particle: photons, electrons, positrons, protons, and neutrons, and the history of the particle (if the particle belongs or not to a new history).

Because MCNP is an algorithm class I the history of the particle, extracted from the files, was eliminated from the phase space information. The remaining information was completely used and implemented in a readable input for MCNP. Due to the excessive amount of storage information, the phase space was divided along the x axis into 100 and 40 sections for the 10×10 cm^2 and 4×4 cm^2 fields, respectively. The execution of the MCNP input files was continuous. The output information was stored and the sum for all the contributions. The division, the execution of the program, and the sum of the output information were performed with code written in Python.

The center of phase space was placed in $(0, 0, 0)$. The field size of the phase space used for the performance tests with the water phantom was 10×10 cm^2 and 4×4 cm^2, for the heterogeneous measurements, the phase space was cut to the dimensions of 4×4 cm^2 in a code written in Python.

4.1.4 Dose calculation

MCNP poses three tallies for dose calculation, the tallies F4, F5, and *F8. The tally F4 computes the dose using the method of KERMA approximation, and the tally F5, called the fluence tally, calculates the dose by adding the tally multiplier of fluence to dose equivalent. One of the most common methods for dose calculation in MCNP is the KERMA approximation method [51]. The use of this method is only possible for conditions where low energy photons are simulated (because their secondary electrons have a very short range), and when charged particle equilibrium (CPE), or at least transient CPE conditions, exists. However, if there is an interface between different materials, CPE conditions are not fulfilled.

Whenever CPE does not hold, the absorbed dose must be calculated by the *F8 tally. For this work purposes and methods, the *F8 tally was used in every simulation. The *F8 tally gives the average energy deposited in a tally cell per source particle in MeV. To calculate de absorbed dose in every cell, it is necessary to divide the result of the energy stored by the cell mass; the cell mass information is provided in table 30

of the output MCNP information [51].

4.2 Geant4 code

Geant4 (GEometry ANd Tracking 4) is an MC toolkit for simulating particles through matter based on object-oriented technology. Geant4 is written in C++, providing great flexibility to redefine the components if needed. It has the capability to simulate a wide variety of particles, the default class $G4ParticleDefinition$ represents particles such as electrons, protons, and photons. If any other particle as deuterons or heavy ions is needed, it is necessary to include other particle class. The user can also select the type of physical process implementation, different to the set provided [52].

Geant 4 can simulate the physics of particle decay, electromagnetic interactions, hadronic interactions, and so on. The physical process allows particle tracking simulation of particles down to 250 eV. Geant4 utilizes a class II electron transport algorithm, meaning that the primary electron loses energy by producing secondary electrons above the discrete electron creation cut-off energy, below the energy threshold the electron energy is deposited locally [52]. Each type of particle has a suggested threshold, the default considerations for the cut-off energies are the following:

1. The production of secondary particles depends on each process inherent limits.
2. All the particles produced are tracked up to zero range.
3. Each particle has an own cut range that depends on the interaction material.

Geant 4 has been used in medical research as microdosimetry, radioprotection, and radiotherapy verifications. Some open Geant4 medical related projects are G4DNA, G4MED, and GATE [53].

4.2.1 Geant4 input file

Geant4 input file is divided into different categories:

- Global, for the unit system, constants, and random number handling.
- Material, for physical characteristics as material composition and mass density.
- Geometry, for the description of geometrical structures.
- Particle, for the implementation of default particles or others.
- Track, for the classes of tracks used by the process.
- Processes, for the implementation that models physical interactions.

It is advisable to create a *main*() function that contains every executable class of the simulation. The *main*() file must include the classes

- *G4RunManager.hh*
- *DetectorConstruction.hh*
- *PhysicsList.hh*
- *PrimaryGeneratorAction.hh*

The *G4RunManager.hh* class manages the initialization procedure. This class must include the information about how the detector is constructed, the particles and physical processes, and how the primary particles are produced.

The *DetectorConstruction.hh* class, contains the volumes to be used and the detector geometry. The volume construction takes into consideration the physical, logical and solid volume. Every volume is constructed by specifying its shape and physical characteristics, and then positioning it inside a containing volume, the former volume is called the daughter volume and the later the mother volume. The solid volume is used to specify the volume and dimensions of the object simulated. The logical volume describes the material of the volume, the detectors if used, and the geometrical properties.The *G4Material*, for example, describes the macroscopic properties of matter as density, state, temperature, and pressure. The geometrical hierarchy consists of a word volume that contains the global coordinate system, the mother volume that can be placed in the world volume more than once (containing all the daughters in their physical volume), and the logical volume that can be also

placed more than once in the mother volume. A representation of Geant4 geometrical hierarchy is shown in Figure 4.3 .

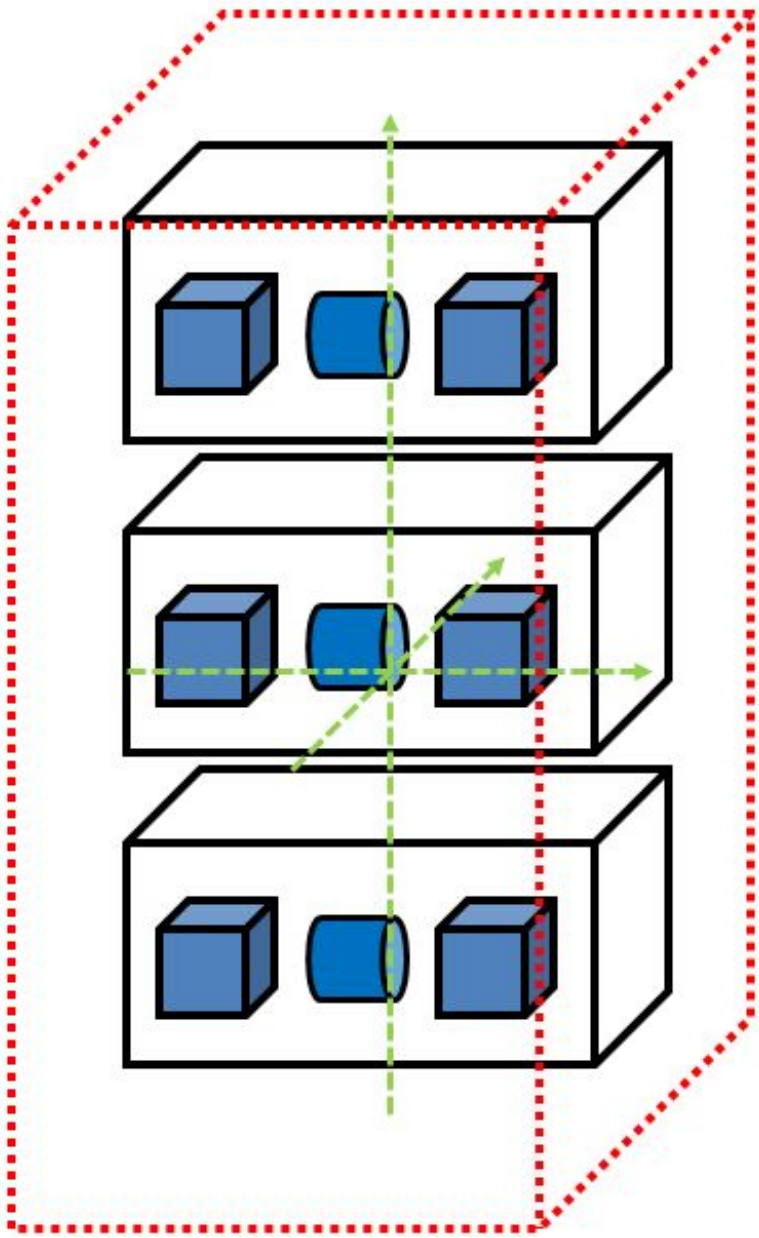


Figure 4.3 Geant4 geometrical hierarchy of volume specifications.

The *PhysicsList.hh* defines all the particles and physical processes to be used in the simulation. The particle construction requires to be called by the function $ConstructeParticle()$. The user must define the primary particles and the secondary

particles generated by the physics processes used. The threshold particle energies are determined in this sections; the threshold is defined by range cut-off energy or distance. A default threshold for gammas, electrons and positrons can be used with the function $SetCutWithDefault()$.

The $PrimaryGeneratorAction.hh$ class specifies how a primary event is created. In this class, the main characteristics as direction, energy, momentum, position, type, and the particles to be used are defined. If a phase space beam generator is used then the class $G4IAEAphspReader$ must be included.

4.2.2 Phantom construction

The phantom simulation in Geant4 consisted of a water phantom, a heterogeneous phantom $QUASAR^{TM}$ [50], and a voxelized phantom based on CT lung images. The dimensions, materials, and isocenter were the same as the MCNP simulation for the water and the heterogeneous phantom. The detector construction in both phantoms was simulated along all the logical volume of the water tank, and the tumor and lung insert simulated in the $QUASAR^{TM}$ phantom.

The voxelized phantom based on CT lung images followed the Geant4 DICOM documentation [52]. The densities used are extracted from the HU and associated with the materials using the ICRU Report 46 [54] as shown in section 3.3.3. The relationship between mass density and the material association is shown in the Table 4.1.

Density range g/cm^3	**Materials**
[0.100, 0.800]	Lungs
[0.919, 0.979]	Adipose
[0.979, 1.004]	Breast
[1.004, 1.109]	Liver
[1.109, 1.113]	Muscle
[1.113, 1,400]	Trabecular bone
[1.496, 1.654]	Dense bone

Table 4.1: Density range and material relantionship

The DICOM module is exportable to read images only by adding the class initialization $DicomHandler.hh$ into the $main()$ Geant 4 function. An example of the voxelized based CT images is shown in Figure 4.4 .

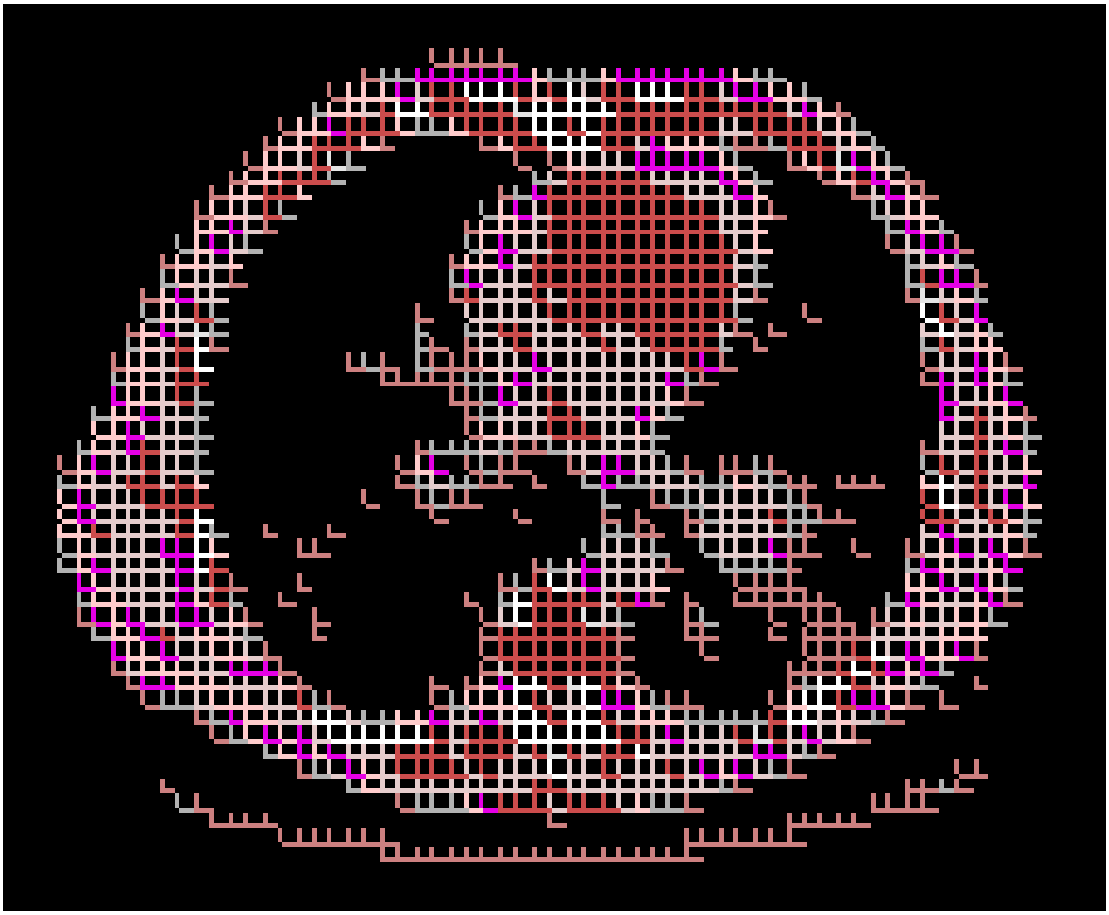


Figure 4.4 Geometrical representation of the voxelixed phantom based CT-images

4.2.3 Beam simulation phase space

The IAEA-Geant4 interface includes the classes *G4IAEAphspReader* and *G4IAEAphspWriter*. The IAEA phase space stores photons, electrons, positrons and neutrons particles. The IAEA classes defined public functions for the simulation of the source, the class R stands for *G4IAEAphspReader* and the class W for *G4IAEAphspWriter*, shown in Table 4.2 .

IAEA function	Class
iaea_check_file_size_byte_order()	R
iaea_copy_header()	W
iaea_destroy_source()	W/R
iaea_get_constant_variable()	R
iaea_get_extra_numbers()	R
iaea_get_max_particles()	R
iaea_get_maximum_energy()	R
iaea_get_particle()	R
iaea_get_total_original_particles()	R
iaea_get_type_extra_variables()	R
iaea_get_used_original_particles()	R
iaea_new_source	W/R
iaea_print_header()	W
iaea_set_constant_variable()	W
iaea_set_extra_numbers()	W
iaea_set_paralell()	R
iaea_set_record()	R
iaea_set_total_original_particles()	W
iaea_set_type_extrafloat_variable()	W
iaea_set_type_extralong_variable()	W
iaea_update_header()	W
iaea_write_particle()	W

Table 4.2: Usage of public functions defined in the IAEA classes

4.2.4 Dose calculation

The dose calculation in Geant4 is obtained by exporting the output information of the detectors constructed in the simulation, obtained by a voxel grid witch size is specified by the user. Because the output information is extracted from all the logical volume, it is necessary to cut the information desired from the detector. The exported data is then analyzed in a Python program for analysis purposes.

5

Results and discussion

The Monte Carlo simulation process developed encompasses the geometry constructions, the beam simulation, and the analysis and extraction of the output information required to calculate the absorbed dose. The MC calculations were performed in MCNP and Geant4.

Before the verification of treatment in a phantom, it is necessary that verification measurements are performed. The verification measurements act as a benchmark model, eliminating coding user errors. Percentage depth dose simulations and dose profile curves were measured at reference conditions for a beam energy of 6 MeV. The measurements of both MC codes were compared against each other and analyzed according to the limit criteria of each measurement.

5.1 Beam calculation

The beam calculation information for both MC codes is taken from the International Atomic Energy Agency (IAEA) [31]. Although the beam information implementation was different for Geant4 and MCNP, the initial parameters were the same. The simulated linear accelerator was the ELEKTA Precise for the 6 MeV energy. For all the simulations 50 million phase space particles were computed and variance and noise reduction techniques were not considered. The statistical uncertainties were computed for all the detectors using the history by history method presented in the section 3.2.4.

5.1.1 MCNP code

To model the beam in MCNP, the IAEA phase space information was introduced in as an input information readable by the code. Because MCNP does not have a native implementation for the IAEA phase space, a generic converter from IAEA to ASCII was used to extract the beam information. A Python program was used to convert the phsc information into an MCNP beam input. The Python program removes the information of the initial position of the particle (x, y, z), the direction cosines (U, V, W), the kinetic energy of the particle E, the particle statistical weight, and the type of particle: photons, electrons, positrons, protons.

Only the photon particles were extracted from all the MCNP simulations, meaning that the electron contamination was not computed. However, the MCNP particle mode for photons and electrons was used. The Python program wrote the particle information as point sources. The beam was restricted to a 10×10 cm^2 and 4×4 cm^2 field in the $z = 0$ axis. After the beam was limited, a Python program divided the entire phase space beam size into 100 and 40 segments, respectively. The sections were divided along the x axis, meaning that each segment has a dimension, of 1 mm and 10 cm along the x and y axis. Another Python program converts the "raw" information of every file into a readable input file for MCNP. An example of how the particles were simulated is written below.

```
MODE  P  E
SDEF  P:4 PAR=2 pos d3 erg fpos d4 vec fpos d5
C Particle coordinate positions
si3 L 0.01956 -1.63582 0.0 &
0.07967 0.10135 0.0 &
0.01956 -1.63582 0.0 &
C Particle statistical weight
spe 1 &
1 &
1 &
C Energy in MeV
ds4 l 1.63866 &
1.15732 &
2.55346 &
C Direction cosines
```

```
ds5 L 0.01956 -1.63582 0.0 &
0.07967 0.10135 0.0 &
0.01956 -1.63582 0.0 &
```

5.1.2 Geant4

Geant4 possesses an interface for the IAEA phase space simulations. However, the lack of proper documentation makes this type of Geant4 simulation a not so much user-friendly interface. As mentioned before the phase space must include two essential classes: *G4IAEAphspReader* and *G4IAEAphspWriter*. The class *G4IAEAphspReader*, read the phase space file and converts it into an input file readable for Geant4. For example, the particle properties are enlisted as

```
std::vector<G4int>* theParticleTypeVector;
std::vector<G4double>* theEnergyVector;
std::vector<G4ThreeVector>* thePositionVector;
std::vector<G4ThreeVector>* theMomentumVector;
std::vector<G4double>* theWeightVector;
std::vector< std::vector<G4double> >* theExtraFloatsVector;
std::vector< std::vector<G4long> >* theExtraIntsVector;
```

Besides the *G4IAEAphspReader*, the classes necessary for reading the phase space file are

- *iaea_config.h*, this class specify the type and characteristic information extracted from the phsp
- *iaea_header.h*, this class reads the IAEA-header and remove the information about some histories simulated, type of linear accelerator, global photon energy cutoff, beam size, etc.
- *iaea_phsp.h*, this class exports all the particle information contained in the phsp file, energy particle, position, etc.

The whole code for this classes is included in Appendix A. A visual representation of the simulated beam is shown in Figure 5.1.

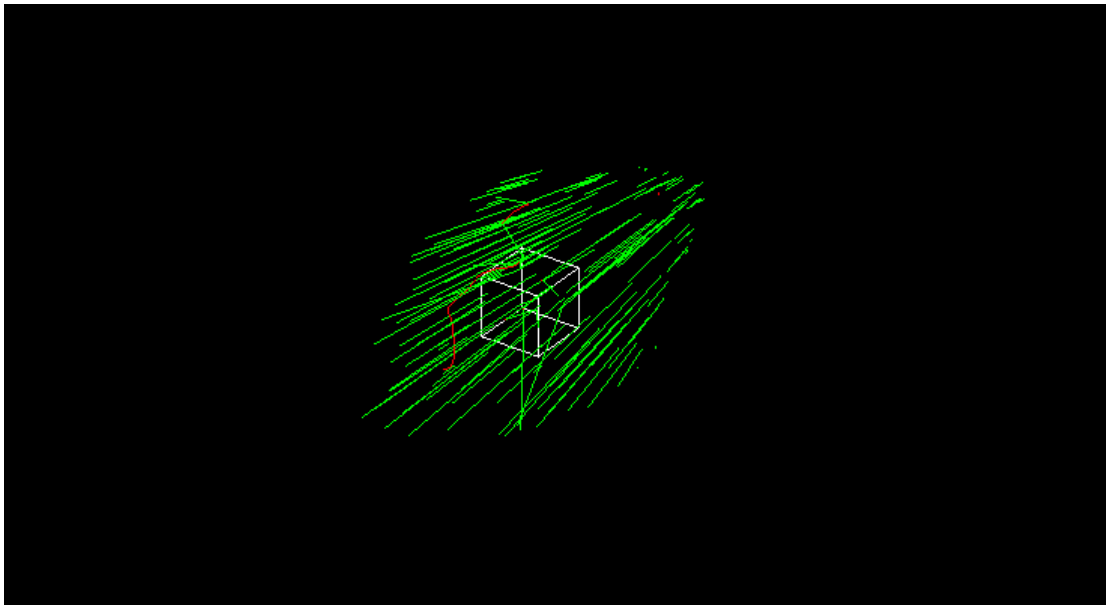


Figure 5.1 Geant4 visualization beam for a 6 MeV, 10×10 cm^2 beam.

5.2 Depth dose curves

The calculation of the depth dose curves (PDD) were computed in MCNP and Geant4 for a 10×10 cm^2 IAEA phase space 6 MeV beam.

The input file for MCNP is shown in Appendix B. The construction of the water phantom in Geant4 is provided in Appendix C.

The information for the dose calculation for MCNP was extracted from the detectors simulated, for Geant4 the dose information was extracted from the z axis in $x = 0$ and $y = 0$, from the beginning of the water phantom and 30 cm depth. Every simulation was repeated 3 times in both codes with different random seeds. The statistical uncertainty calculation was performed in each detector for both codes.

The maximum percentage difference of 2.6% was found in the 22.5 cm depth. For most of the depths in the simulated PDD, MCNP predicts a slightly greater absorbed dose than Geant4. The PDD results are within the statistical uncertainties, these differences are insignificant taking into account that the electron contamination in MCNP was not simulated and that the number of particles simulated is low. For a proper clinical simulation, it is advisable to simulate 300 million particles instead of

the 50 million simulated. Minor disagreements in the entrance regions of the PDD exists. The construction of more detectors in the entrance region and more histories simulated are required to validate this results correctly. In the Figures, 5.2 - 5.4 the MCNP and Geant4 results are shown.

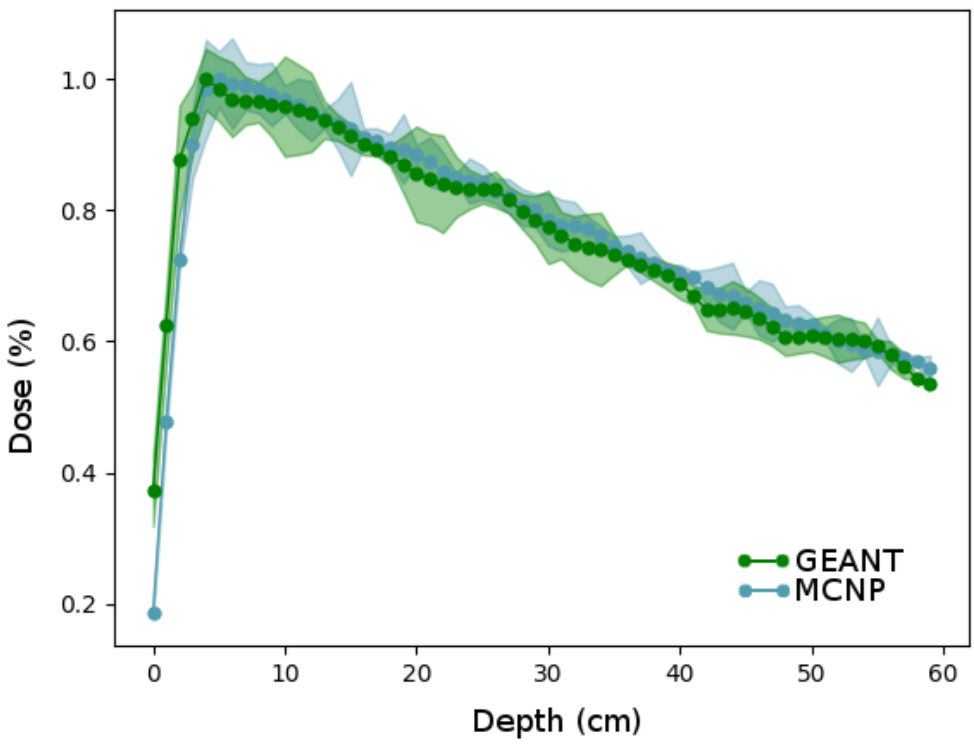


Figure 5.2 Depth dose curve simulation comparisons of MCNP and Geant4

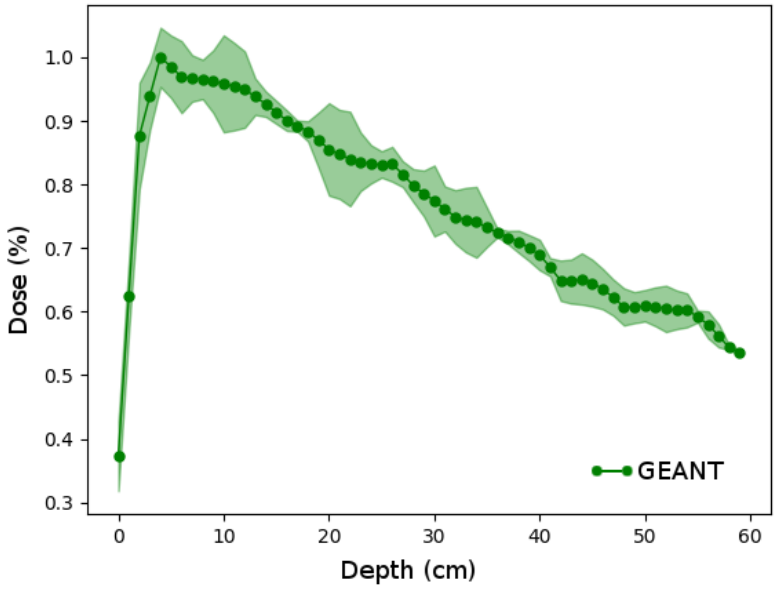


Figure 5.3 Depth dose curve simulation from Geant4

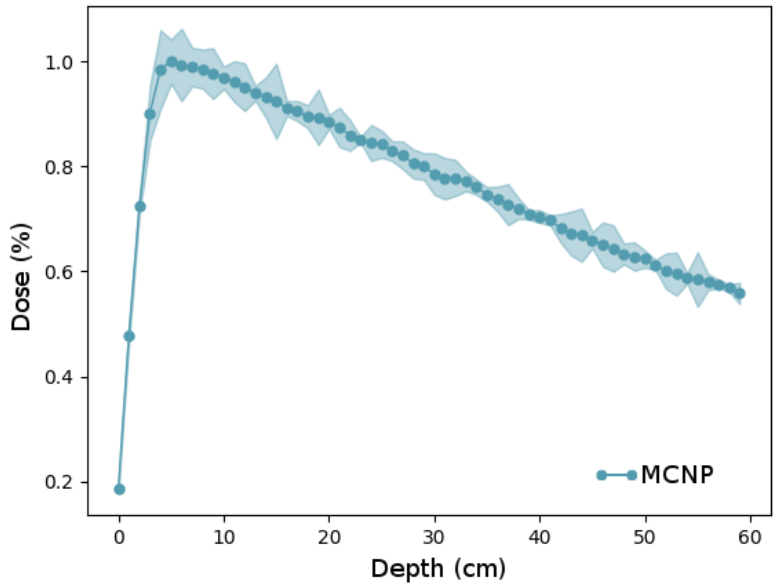


Figure 5.4 Depth dose curve simulation from MCNP

5.3 Beam profile

The calculation of the x and y profile curves were computed in MCNP and Geant4 for a 10×10 cm^2 and a 4×4 cm^2 IAEA phase space 6 MeV beam. The geometry phantom is the same as the used for the calculation of the percentage depth dose.

For MCNP, the information for the dose calculation was extracted from the detectors simulated, for Geant4 the detectors were simulated in the x and y axis, for a 14 cm length. Again, the simulation was repeated 3 times in both codes with different random seeds.

For the 10×10 cm^2 field the maximum percentage difference of 3.7% and 4.8% was found in the length $x = -5$ cm and $y = 1.5$ cm, for the x and y beam profiles, measured from the center of the beam. Both Monte Carlo codes predict matching dose curves. The characteristic bell shape of the profile curves is clearly reached. The calculated flatness for MCNP and Geant4 were for the x axis 1.03%, and 4.3%, and for the y axis 1.6%, and 3.7%, respectively. The results are between the range recommended. Along the x axis, Geant4 predicts a greater absorbed dose than MCNP, however, along the y axis MCNP shows a slightly greater dose than Geant4. For the 4×4 cm^2 field, the characteristic bell shape of the profile curves is clearly reached. Both codes showed a good agreement within the statistical uncertainties for both fields, although again the number of particles simulated is low. In the Figures, 5.5 - 5.8 the MCNP and Geant4 results are shown.

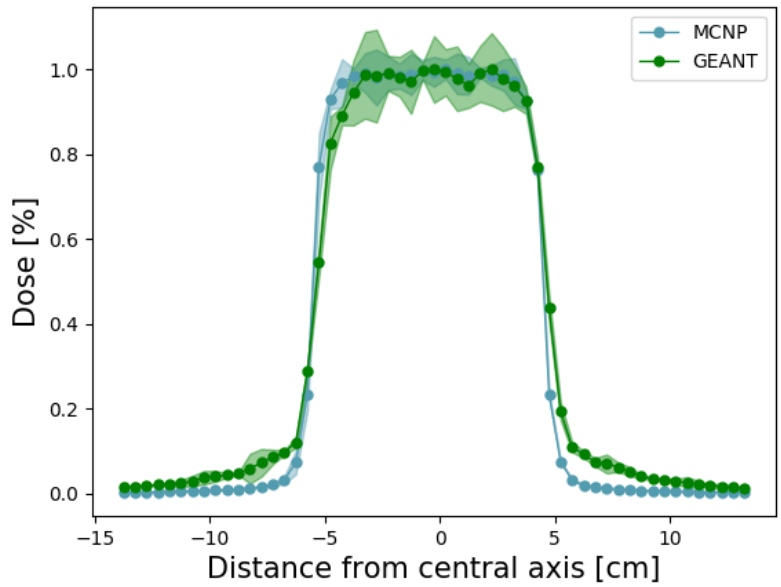


Figure 5.5 Beam profile along x axis from MCNP and Geant4 comparison for a $10 \times 10\ cm^2$ field

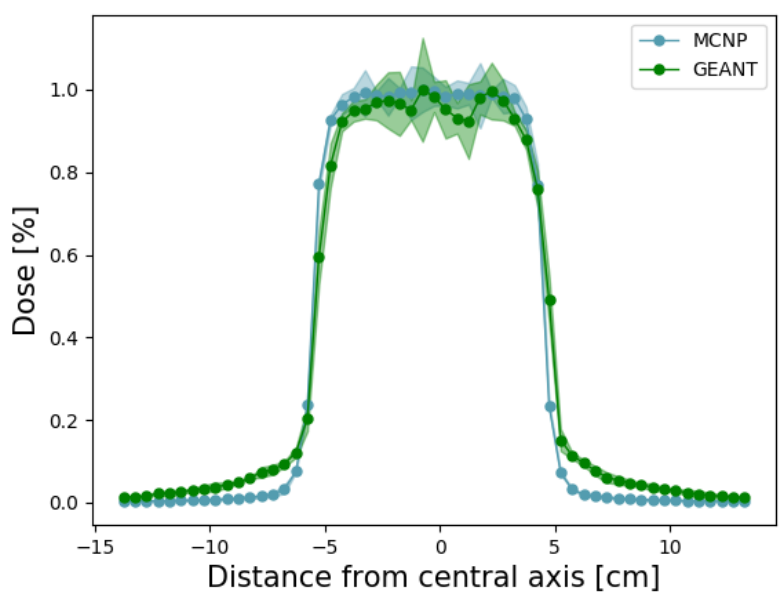


Figure 5.6 Beam profile along y axis from MCNP and Geant4 comparison for a $10 \times 10\ cm^2$ field

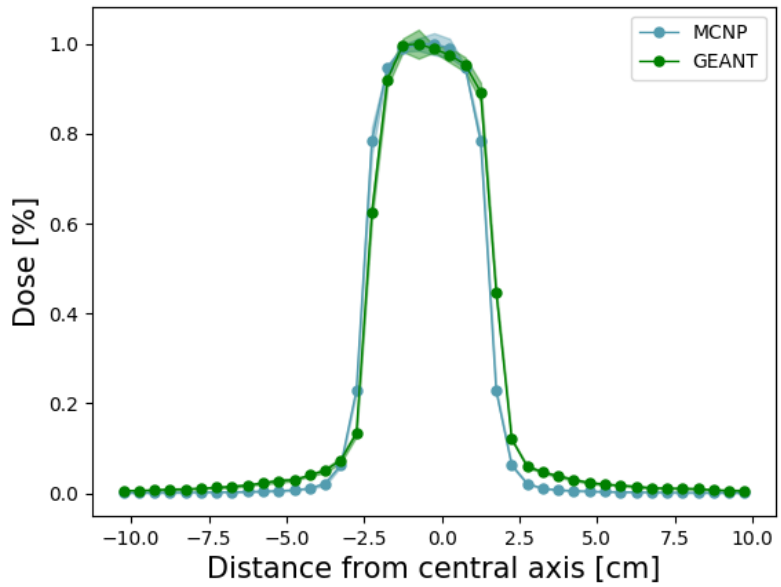


Figure 5.7 Beam profile along x axis from MCNP and Geant4 comparison for a $4{\times}4\ cm^2$ field

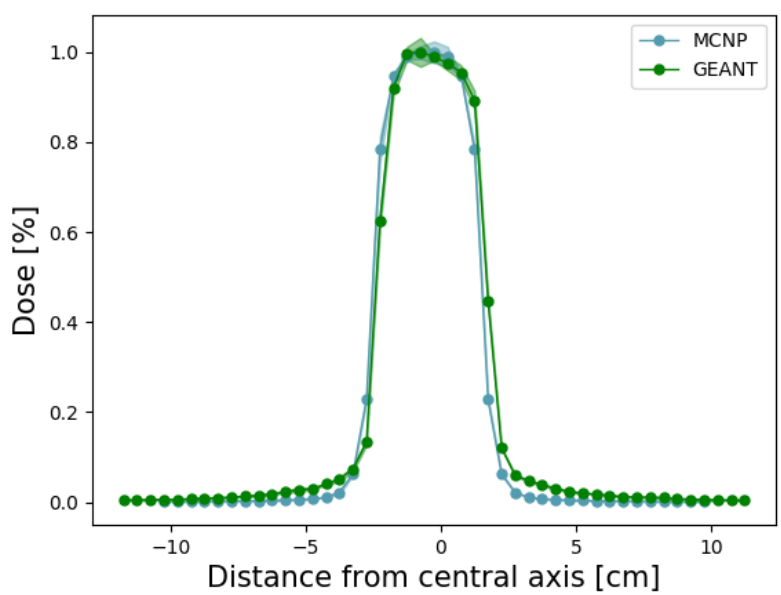


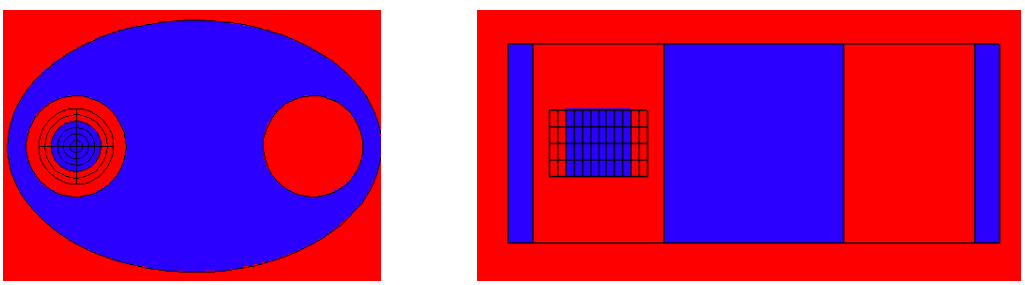
Figure 5.8 Beam profile along y axis from MCNP and Geant4 comparison for a $4{\times}4\ cm^2$ field

5.4 QUASAR phantom

The QUASAR simulation for both codes take the real measurements of the $QUASAR^{TM}$ Multi-Purpose Body Phantom. The measurement was extracted as percentage depth dose calculations and dose distributions. The characteristics of the simulations were the same for both codes, a 4×4 cm^2 beam size of 6 MeV were simulated.

5.4.1 MCNP

For MCNP, the information for the dose calculation was extracted from the detectors simulated. The construction of the detectors was computed for both calculations, the circular detectors, and the square detectors, for dose distributions and PDD measurements, respectively. The circular detectors consider the area of a virtual tumor and a partial area of the lung, as seen in Figure 5.9.



(a) (b)

Figure 5.9 Quasar phantom with the circular tallies simulated from a a) QUASAR front view and b) xial view

The simulation consists of a single beam pointing from the $z = 0$ axis. The result of the QUASAR simulation is shown in Figure 5.10. The dose distribution resulted as expected. A minimal energy reaches the lung area, and the dose is deposited into the tumor. The visualization of the distribution was made in a program in Python. As expected the dose in the tumor began to diminish as it reaches the center of a tumor, for this reason, several beam angles, in clinical practice, are simulated from a different distance and directions.

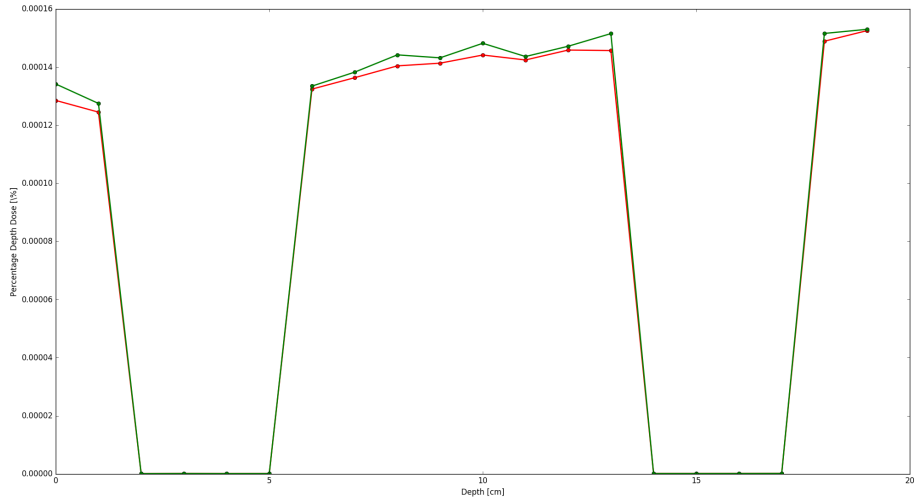


Figure 5.10 Percentage Depth Dose comparisons in the QUASAR phantom for MCNP and Geant4

5.4.2 Geant4

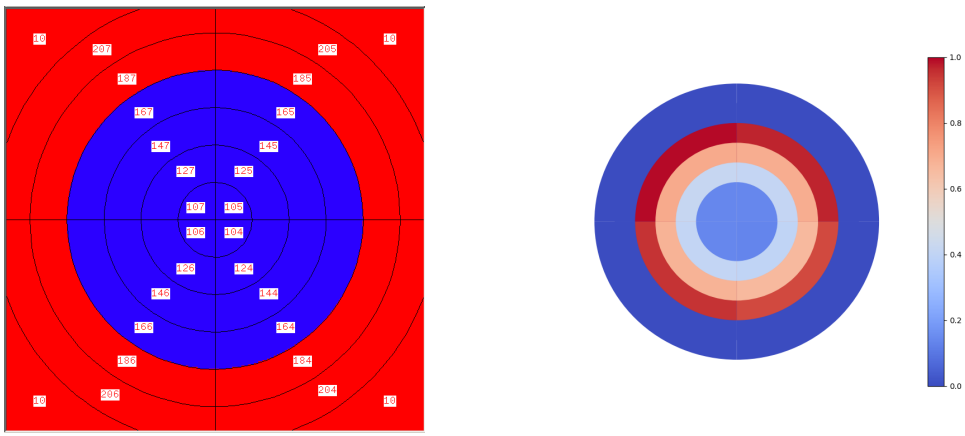
The information extracted from the QUASAR coincide with the plane analyzed in MCNP. The information was obtained to coincide the geometry of the MCNP circular and square detectors.

The simulation also consists of a single beam pointing from the $z = 0$ axis. The result of the QUASAR simulation is shown in Figure 5.11. Again, a minimal energy reaches the lung area, and the dose is deposited into the de tumor. The visualization of the distribution was made in a program in Python.

5.4.3 MCNP and Geant4

The comparison of the dose distributions for MCNP and Geant4 were computed. The percentage depth dose and dose distributions were calculated and compared to both codes. Noise reduction techniques were not used for this simulation. The maximum percentage difference was 4.32% in the percentage depth dose calculations. The analysis of the dose distribution took only percentage difference, for the detector size made impossible a gamma index evaluation. The maximum percentage difference was of

3.89% in the dose distribution calculations. Both comparisons are shown in Figure 5.11. The results indicate a good agreement within the statistical uncertainties under the circumstances examined. Once more, Geant4 predicts a slightly greater absorbed dose than MCNP.



(a) (b)

Figure 5.11 a) QUASAR circular tallies at $(0, 0, -100)$ and b) the QUASAR dose distribution from a 4×4 beam at 0 *degrees* at the circular tallies

5.5 DICOM voxelized phantom

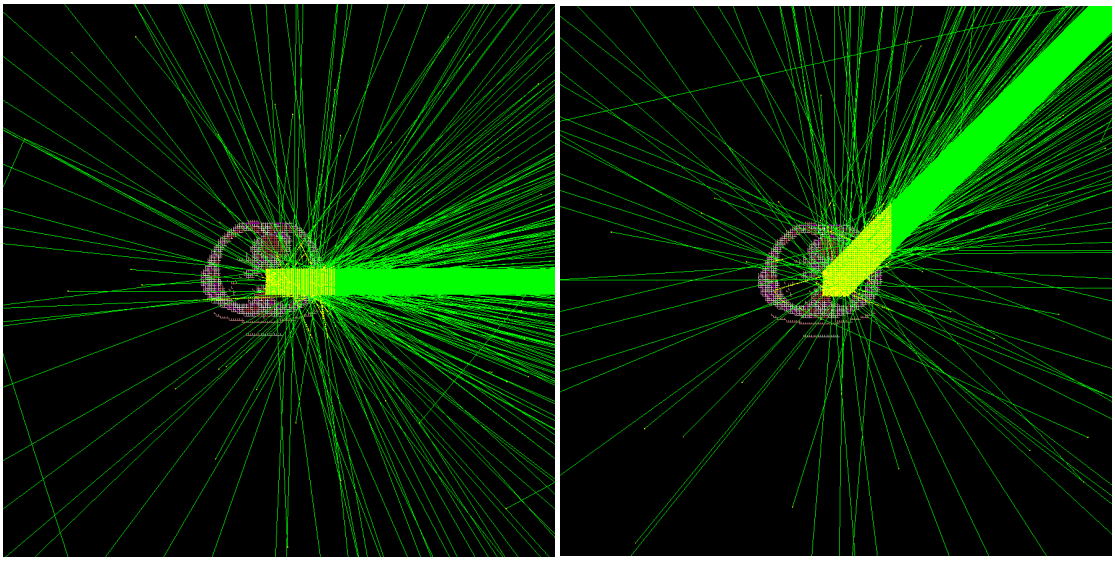
The purpose of the simulation of the DICOM voxelized phantom was to determine the feasibility to perform a Monte Carlo simulation for lung cancer. Large efforts were made regarding the implementation of a DICOM phantom in MCNP. Due to the incredibly long storage of the CT information, along with the phase space information, the amount of computer power required overcomes the possibility of a realistic simulation. The simulation in Geant4 was possible in a reasonable amount of time with the phase space used.

The geometry input information simulated was a public domain CT lung images. The beams simulated followed most of the dosimetric requirements for an SBRT for an early stage lung cancer. The minimum field size of 3.5 cm^2 was fulfilled by simulating a 4×4 cm^2 field. The beams were simulated as non-opposing and coplanar beams. Eight beams were simulated, fulfilling the requirement of a minimum of seven beams. The direction of the beams was simulated to point towards a centrally located tumor in the right lung.

Table 5.1: Angular direction of the simulated beam

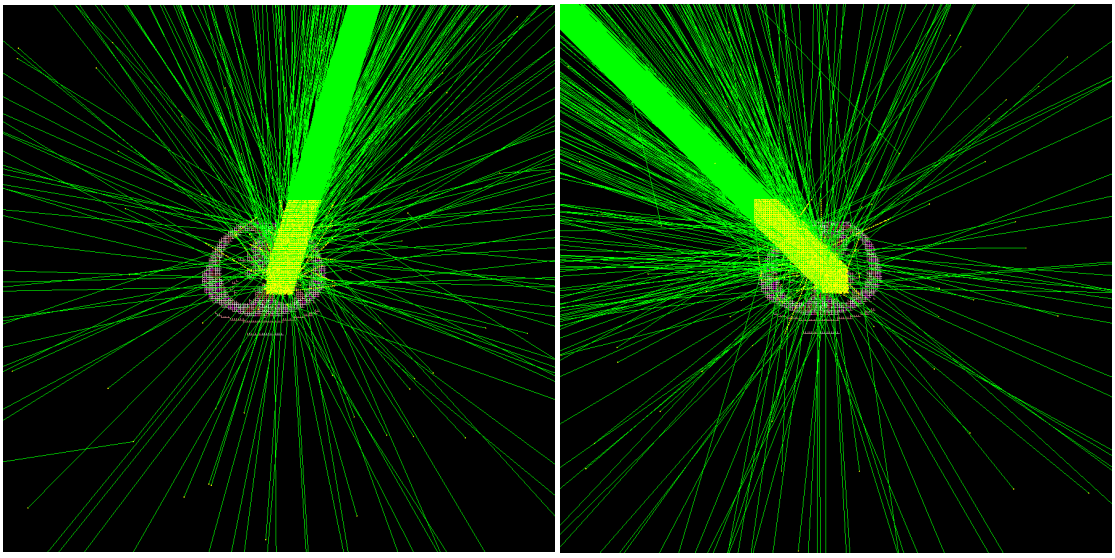
Number of beam	Beam angle
1	0
2	45
3	73
4	135
5	180
6	225
7	270
8	315

Similar to a clinical treatment planning system, the resulting dose distribution was plotted using the sum of all the absorbed dose of every beam. The beam directions used are shown in Table 5.1. The geometric representation of the simulated beam angles and the final dose distribution is shown in Figures 5.12 - 5.17. The results, although not corresponding with a real clinical simulation, indicate that the simulation of a stereotactic body radiation therapy is possible if the right parameters are introduced into the Geant4 code. Normalized dose information can be obtained by the dose distribution computed in Geant4 for every slice of the CT, making possible to compare dose the Monte Carlo dose distribution with a deterministic algorithm. However, new techniques to obtain the approximation of Monitor Units in Monte Carlo are required for Monte Carlo to serve as a code for deterministic dose-modifications in clinical practice. This first step could serve as an analysis tool for dose distribution comparisons against deterministic algorithms in cases containing severe heterogeneities and were unclear results from the deterministic calculation are expected.



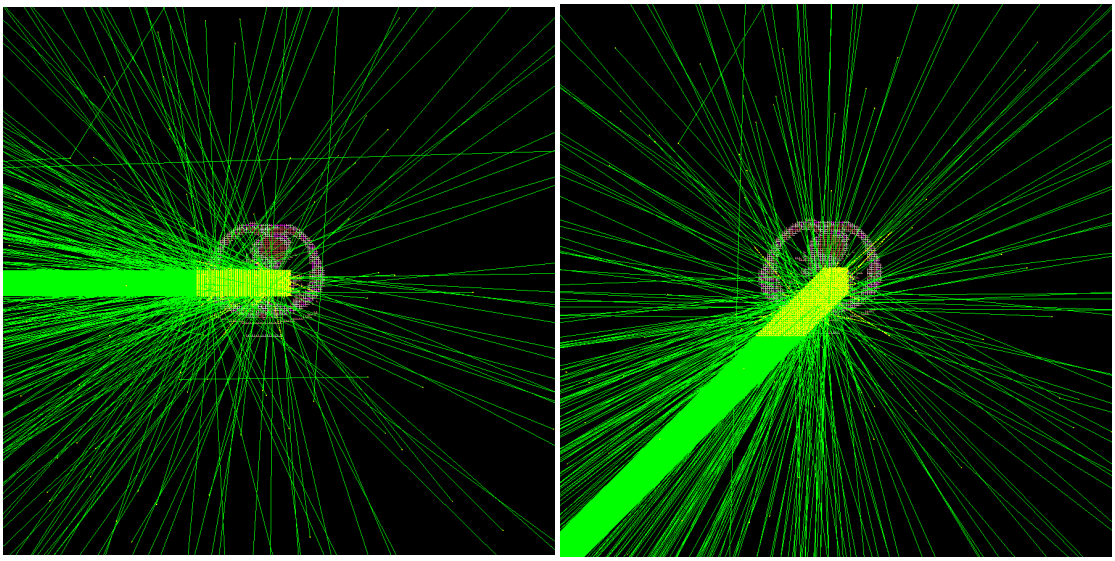
(a) (b)

Figure 5.12 DICOM voxelized image simulation in Geant4 for a a) 0 degrees beam angle and b) 45 degrees beam angle.



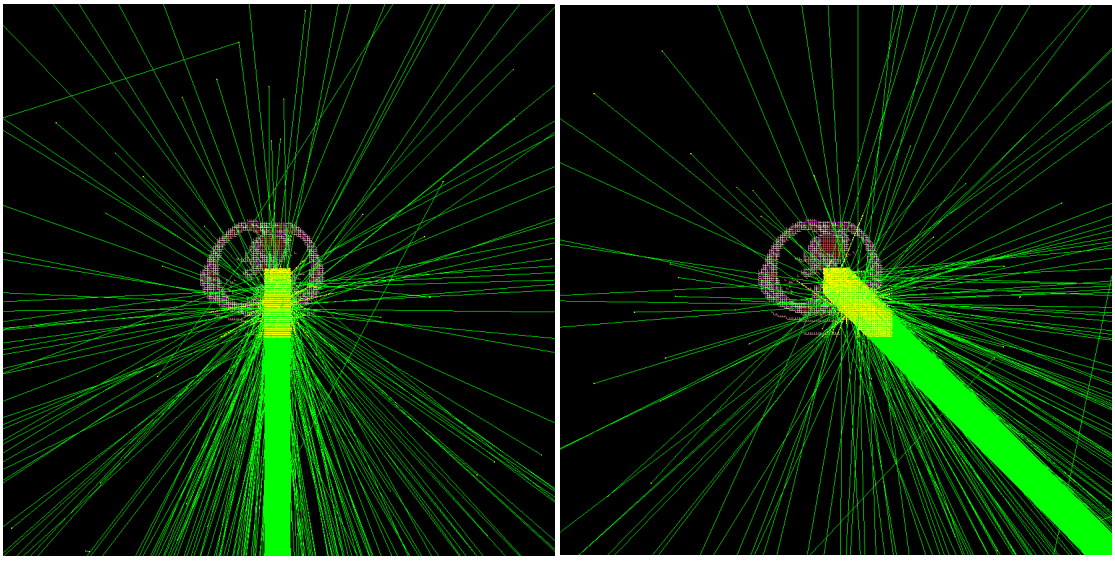
(a) (b)

Figure 5.13 DICOM voxelized image simulation in Geant4 for a a) 73 degrees beam angle and b) 135 degrees beam angle.



(a) (b)

Figure 5.14 DICOM voxelized image simulation in Geant4 for a a) 180 degrees beam angle and b) 225 degrees beam angle.



(a) (b)

Figure 5.15 DICOM voxelized image simulation in Geant4 for a a) 270 degrees beam angle and b) 315 degrees beam angle.

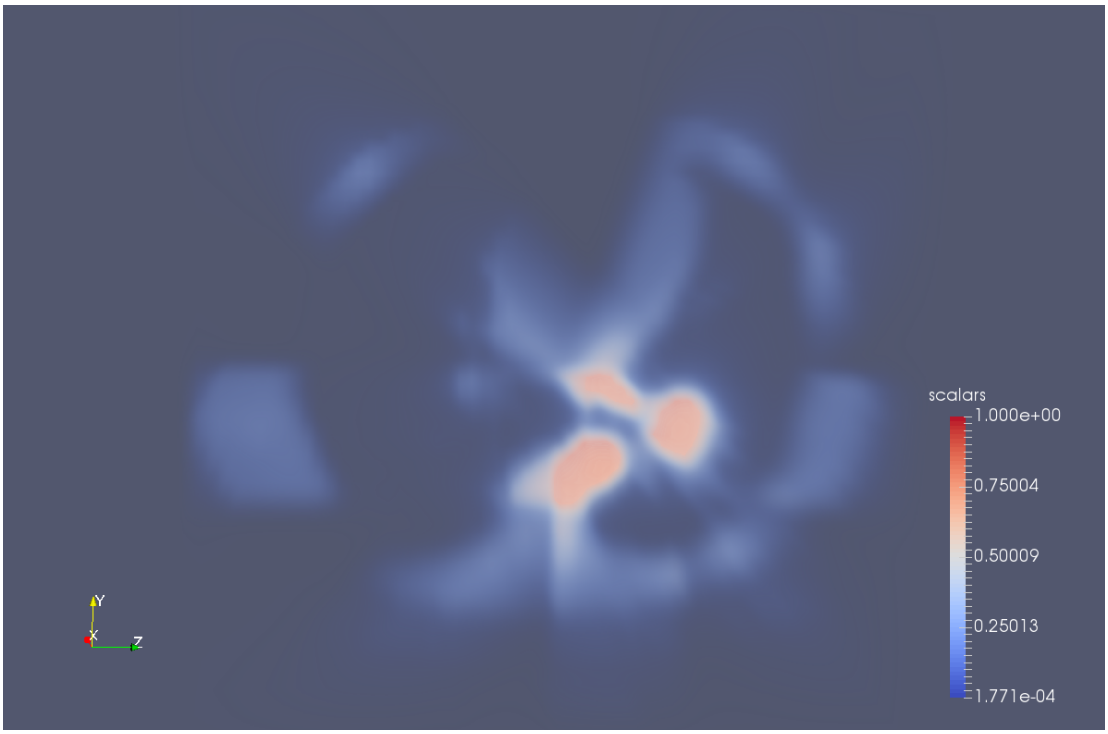


Figure 5.16 Simulated dose distribution for the DICOM voxelized phantom in Geant4

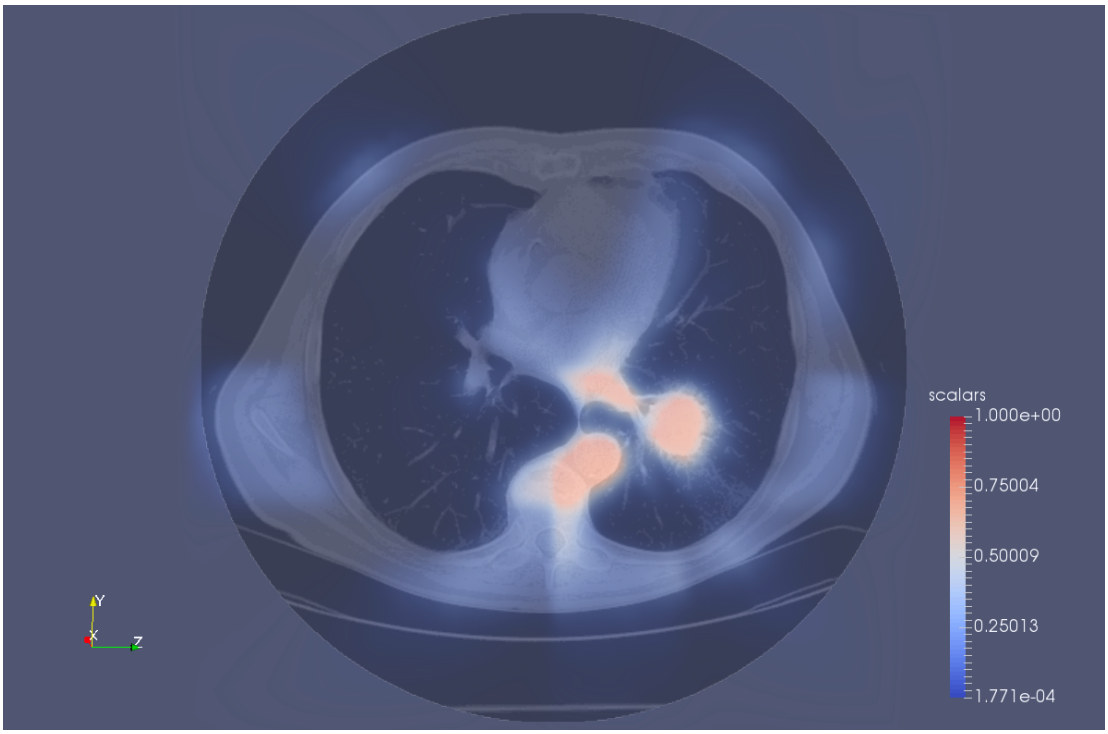


Figure 5.17 Simulated dose distribution for the DICOM voxelized phantom merged with the CT image used in Geant4

6

Conclusions and future work

The most implemented algorithms in the commercially available treatment planning systems are the deterministic algorithms. Nevertheless, the implementation of Monte Carlo dose calculations has taken place in many research institutions, especially if an unclear result of the deterministic calculation is expected. Radiotherapy cases containing severe heterogeneities are expected to benefit the most from Monte Carlo verifications. According to the Task Group from the American Association of Physicist in Medicine, the strongest motivation for a Monte Carlo verification comes from the treatment for lung cancer. This becomes evident as there are percentage differences between the calculated and the measured dose as big as 30% in some cases.

The present work aims to determine the feasibility of applying a Monte Carlo simulation for lung cancer as an analysis and research tool for clinical comparisons. The proper implementation demands that the clinical physicist understand, on some level, the fundamentals of the algorithm as well as the possible pitfalls associated with its clinical application, much as one would for any dose algorithm. This system is intended serve as an analysis tool for clinical comparisons.

The Monte Carlo simulations were performed for the MCNP and Geant4 codes. The characteristics of the simulations were the same, except in the beam phase space input information. MCNP input phsp beam only simulates initial photons, and Geant4 simulated all the phsp particles extracted from the phase space. The exclusion of other particles, other than photons, a small percentage of the particles from the provided phase space file were not simulated in MCNP. For all the measurements 50 million particles were simulated. Variance reduction techniques and specific beam devices were not simulated. The mean time simulation for both codes was of 385 *hr* for 1 processor core. Because it is possible to run the codes using parallelization, for an average high-performance computer the time is reduced to 48 *hr* for 8 processor cores for each measurement.

The verifications measurements as Percentage Depth Dose and beam profiles were simulated for both codes. The geometry construction consisted of a water tank of $675 \times 645 \times 560\ mm$, the acrylic wall thickness of 15 mm equivalent to a real phantom were also simulated. For all the measurements the source-surface-distance was 98.5 cm. The isocenter was simulated coinciding with the z_{max} of the energy of 6 MeV used.

For PDD verifications a maximum percentage difference of 2.6% was found at the 22.5 cm depth. For beam profile measurements for the $10 \times 10\ cm^2$ field a maximum percentage difference of 3.7% and 4.8% was found in the length $x = -5\ cm$ and $y = 1.5\ cm$, for the x and y beam profiles, measured from the center of the beam. The calculated flatness for MCNP and Geant4 were for the x axis 1.03%, and 4.3%, and for the y axis 1.6%, and 3.7%, respectively. For the $4 \times 4\ cm^2$ field, the characteristic bell shape of the profile curves is clearly reached showing a good agreement within the statistical uncertainties. The results are within the range recommended for both fields. Along the x axis, for most of the measurements, Geant4 predicts a slightly greater absorbed dose than MCNP. This result is expected for the absence because of the lack of electron contamination in the MCNP simulation.

The comparison of the dose distributions in the QUASAR phantom, for MCNP and Geant4, gives a maximum percentage difference of 4.32% in the percentage depth dose calculations and 3.89% in the dose distribution calculations. Because of the detector size in MCNP, the gamma index evaluation was impossible to measure.

The success of the implementation in Geant4 of the beam phase space model and the visualization of the first dose distribution in a DICOM voxelized phantom, proves the feasibility to perform a Monte Carlo simulation for lung cancer. Although it was only possible to simulate real CT images in Geant4, MCNP has also proved to be a valuable tool for heterogeneous phantom verification treatments, for commissioning purposes and or when the simulation of a heterogeneous phantom is required.

The complementation of this work comes along more measurements to provide an accurate statistics, and the simulation of at least the 300 million particles recommended instead of the 50 million simulated. Further work seems to be necessary towards the implementation of beam modifiers, it is recommended to provide a more accurate and realistic simulation.

Bibliography

[1] B. Fraass, K. Doppke, and M. Hunt, "American association of physicist in medicine radiation therapy committee task group 53: Quality assurance for clinical radiotherapy treatment planning," *The International Journal of Medical Physics Research and Practice*, vol. 25, no. 10, pp. 1773–1829, 1998.

[2] J. Keivan, *Development of a fast Monte Carlo code for dose calculation in treatment planning and feasibility study of high contrast portal imaging.* PhD thesis, McGill University, Montreal, 2008.

[3] S. Martin, *Concepts for the efficient Monte Carlo-based treatment plan optimization in radiotherapy.* PhD thesis, Combined Faculties for the Natural Sciences and for Mathematics of the Ruperto-Carola University of Heidelberg, Heidelberg, 2012.

[4] L. K. Gérard, "Evaluation of eclipse monte carlo dose calculation for clinical electron beams using heterogeneous phantoms," Master's thesis, McGill University, Montreal, Québec, Canada, 2009.

[5] J. J. DeMarco, T. D. Solberg, and J. B. Smathers, "A ct-based monte carlo simulation tool for dosimetry planning and analysis," *The International Journal of Medical Physics Research and Practice*, vol. 25, no. 1, pp. 1–11, 1998.

[6] J. Van Dyk, *The modern technology of radiation oncology.* Medical Physics Publishing, 2006.

[7] H. Wyckoff, A. Allisy, and K. Lidén, "Dose specification for reporting external beam therapy with photons and electrons," tech. rep., International Commission on Radiation Units and Measurements, 1978.

[8] C. M. Ma, E. Mok, A. Kapur, and T. Pawlicki, "Clinical implementation of a monte carlo treatment planning system," *The International Journal of Medical Physics Research and Practice*, vol. 26, no. 10, pp. 2133–2143, 1999.

[9] I. J. Chetty, B. Curran, and J. E. Cygler, "Report of the aapm task group no. 105: Issues associated with clinical implementation of monte carlo-based photon and electron external beam treatment planning.," *The International Journal of Medical Physics Research and Practice*, vol. 34, no. 12, pp. 4818–4853, 2007.

[10] T. Knöös, A. Anhesjö, P. Nilson, and L. Weber, "Limitations of a pencil beam approach to photon dose calculations in lung tissue," *The International Journal of Medical Physics Research and Practice*, vol. 40, no. 10, pp. 1411–1420, 1995.

[11] L. W. Brady, H. P. Heilmann, M. Molls, and C. Nieder, *Stereotactic Body Radiation Therapy*. Schefter Editors, Springer, 2012.

[12] J. J. DeMarco, T. D. Solberg, and J. B. Smathers, "An electron density calibration phantom for ct-based treatment computers," *The International Journal of Medical Physics Research and Practice*, vol. 19, no. 2, pp. 325–327, 1992.

[13] J. Sempau, S. Wilderman, and A. Bielajew, "Dpm, a fast, accurate monte carlo code optimized for photon and electron radiotherapy treatment planning dose calculations," *The International Journal of Medical Physics Research and Practice*, vol. 45, pp. 2263 – 2291, 2000.

[14] L. Wang, C. Chui, and M. Lovelock, "A patient-specific monte carlo dose-calculation method for photon beams," *The International Journal of Medical Physics Research and Practice*, vol. 25, pp. 867 – 878, 1998.

[15] C. M. Ma, J. S. Li, and T. Pawlicki, "A monte carlo dose calculation tool for radiotherapy treatment planning," *Physics in Medicine and Biology*, vol. 47, pp. 1671 – 1689, 2002.

[16] J. V. Siebers, P. J. Keall, B. Libby, and R. Mohan, "A method for photon beam monte carlo multileaf collimator particle transport," *Physics in Medicine and Biology*, vol. 47, pp. 3225 – 3249, 2002.

[17] I. J. Chetty, P. M. Charlan, and N. Tyagi, "Photom beam relative dose validation of the dpm monte carlo code in lung-equivalent media," *The International Journal of Medical Physics Research and Practice*, vol. 30, pp. 563 – 573, 2003.

[18] A. Alexander, F. Deblois, and G. Stroian, "Mctp: A radio-specific research enviroment for monte carlo and patient-specific treatment planning," *Physics in Medicine and Biology*, vol. 52, pp. 297 – 308, 2007.

[19] S. M. de Oncologia A.C., "Historia." `https://www.smeo.org.mx/index.php/quienes-somos/historia`, Online, accessed 14-06-2017.

[20] A. Rogers and F. Bielajew, *Monte Carlo Techniques of Electron and Photon Transport for Radiation Dosimetry*, ch. 5. Academic Press, 2014.

[21] D. W. O. Rogers, "Fifty years of monte carlo simulations for medical physics," *Physics in Medicine and Biology*, vol. 51, pp. 281 – 301, 2006.

[22] J. Seco and V. Frank, *Monte Carlo Techniques in Radiation Therapy.* CRC Press, A Taylor and Francis Book, 2013.

[23] P. Andreo, "Monte carlo techniques in medical radiation physics," *Physics in Medicine and Biology*, vol. 36, no. 7, pp. 861–920, 1991.

[24] I. Kawrakow, "Accurate condensed history monte carlo simulation of electron transport, i. egsnrc, the new egs4 version.," *The International Journal of Medical Physics Research and Practice*, vol. 27, pp. 485 – 498, 2000.

[25] F. Brown, *MCNP—A general Monte Carlo particle transport code, Version 5. Report LA-UR-03 1987.* Los Alamos National Laboratory, Los Alamos, NM., 2003.

[26] I. Kawrakow, "Geant4—a simulation toolkit.," *Nuclear Instruments and Methods in Physics*, vol. A506, pp. 250 – 303, 2003.

[27] J. Baro, J. Sempau, J. M. Fernandez-Varea, and F. Salvat, "Penelope—an algorithm for monte carlo simulation of the penetration and energy loss of electrons and positrons in matter.," *Nuclear Instruments and Methods in Physics*, vol. A100, pp. 32 – 46, 1995.

[28] N. Reynaert, S. C. van der Marck, D. R. Schaart, W. Van der Zee, V. Vliet-Vroegindeweij, M. Tomsej, J. Jansen, B. Heijmen, M. Coghe, and C. De Wagter, "Monte carlo treatment planning for photon and electron beams," *Radiation Physics and Chemestry*, vol. 76, pp. 643–686, 2007.

[29] F. H. Attix, *Introduction to Radiological Physics and Radiation Dosimetry.* Wiley-VCH Verlag GmbH and Co., 2004.

[30] J. Turner, *Atoms, Radiation, and Radiation Protection.* Wiley-VCH Verlag GmbH and Co., 2007.

[31] C. Roberto, "Phase-space database for external beam radiotherapy." `https://www-nds.iaea.org/phsp/phsp.htmlx`, Online, accessed 28-07-2017.

[32] F. M. Khan, *The physics of radiation therapy.* Lippincott Williams and Wilking, 1993.

[33] E. B. Podgorsak, *Radiation oncology physics: A handbook for teachers and students*. International Atomic Energy Agency, 2005.

[34] C. M. Ma, S. B. Jiang, T. Pawlicki, Y. Chen, J. S. Li, J. Deng, and A. L. Boyer, "A quality assurance phantom for imrt dose verification," *Physics in Medicine and Biology*, vol. 48, pp. 561 – 572, 2003.

[35] B. Walters, I. Kawrakow, and D. Rogers, "History by history statistical estimators in the beam code system," *The International Journal of Medical Physics Research and Practice*, vol. 29, pp. 2745 – 2752, 2002.

[36] B. Dobler, N. Streck, E. Klein, R. Loeschel, P. Haertl, and O. Koelbl, "Hybrid plan verification for intensity-modulated radiation therapy (imrt) using the 2d ionization chamber array i'mrt matrixx—a feasibility study," *Physics in Medicine and Biology*, vol. 55, pp. N39 – 55, 2010.

[37] E. E. Grein, R. Lee, , and K. Luchka, "An investigation of a new amorphous silicon electronic portal imaging device for transit dosimetry," *The International Journal of Medical Physics Research and Practice*, vol. 29, pp. 2262 – 2268, 2002.

[38] P. B. Greer and C. C. Popescu, "Dosimetric properties of an amorphous silicon electronic portal imaging device for verification of dynamic intensity modulated radiation therapy.," *The International Journal of Medical Physics Research and Practice*, vol. 30, pp. 1618 – 1627, 2003.

[39] M. Iori, E. Cagni, A. E. Nahum, , and G. Borasi, "A new method for the clinical dosimetry of intensity-modulated arc therapy (imat)," *The International Journal of Medical Physics Research and Practice*, vol. 34, pp. 2759 – 2773, 2007.

[40] C. M. Ma, T. Pawlicki, S. B. Jiang, J. S. Li, J. Deng, E. Mok, A. Kapur, L. Xing, L. Ma, and A. L. Boyer, "Monte carlo verification of imrt dose distributions from a commercial treatment planning optimization system," *The International Journal of Medical Physics Research and Practice*, vol. 45, pp. 2483 – 2495, 2000.

[41] K. Bush, R. Townson, and S. Zavgorodni, "Monte carlo simulation of rapidarc radiotherapy delivery," *Physics in Medicine and Biology*, vol. 53, pp. N359 – 370, 2008.

[42] F. A. Bielajew, *Fundamentals of Radiation Dosimetry and Radiological Physics*. The University of Michigan, 2005.

[43] T. R. S. no. 430, *Commissioning and quality assurance of computerized planning systems for radiation treatment of cancer.* International Atomic Energy Agency, 2014.

[44] B. G. Carleson, *Radiation therapy planning.* McGraw Hill, 1996.

[45] W. Laub, M. Alber, M. Birkner, and F. Nusslin, "Monte carlo dose computation for imrt optimization," *Physics in Medicine and Biology*, vol. 45, pp. 1741 – 1754, 2000.

[46] R. A. Brooks, L. G. Mitchell, M. O. Christopher, and G. DiChiro, "On the relationship betenn computed tomography numbers and specific gravity," *Physics in Medicine and Biology*, vol. 26, pp. 141 – 147, 1981.

[47] I. N. d. C. Unidad funcional de oncología torácita y laboratorio de medicina personalizada, "Cáncer de pulmón." `http://cancerdepulmon.com.mx/seccion-educativa`, Online, accessed 28-07-2017.

[48] R. T. O. G. 0813, *Seamless phase I/II study of stereotactic lung radiotherapy (SBRT) for early stage, centrally located, non-small lung cancer (NSCLC) in medically inoperable patients.* AAPM, 2012.

[49] L. A. N. Securuty, "Monte carlo methods, codes, and applications group." `https://mcnp.lanl.gov/`, Online, accessed 28-07-2017.

[50] O. O. systems limited, "Multi purpose body phantom." `http://osl.uk.com/rte.asp?id=241`, Online, accessed 28-07-2017.

[51] J. K. Shultis and R. E. Faw, *An MCNP primer.* Department of Mechanical and Nuclear Engineering, 2011.

[52] V. Hubert-Tremblay, L. Archambault, C. Jarlskog, and L. Beaulier, "Geant4 dicom documentation." `http://www.nucleaire.phy.ulaval.ca/phys_med/DICOM/`, Online, accessed 28-07-2017.

[53] G. collaboration team, "Medical applicartions." `http://geant4.cern.ch/applications/medapp.shtml`, Online, accessed 28-07-2017.

[54] D. R. White, R. V. Griffith, and I. J. Wilson, *Photon, electron, proton and neutron interaction data for body tissues.* Oxford Academic, 2016.

A

Geant4 phase space classes

Besides the *G4IAEAphspReader*, the classes *iaea_config.h*, *iaea_header.h*, and *iaea_phsp.h* are also included. The class for the *iaea_phsp.h* was downloded from the IAEA phase space documentation page. The code for this classes is shown below.

A.1 iaea config file

```
// int *     signed, signed int System dependent
// unsigned int    *     unsigned    System dependent
//__int8     1     char, signed char  -128 to 127
//__int16    2     short, short int, signed short int -32,768 to 32,767
//__int32    4     signed, signed int -2,147,483,648 to 2,147,483,647
//__int64    8     none  -9,223,372,036,854,775,808 to
   9,223,372,036,854,775,807
//char 1     signed char  -128 to 127
//unsigned char    1     none  0 to 255
//short      2     short int, signed short int -32,768 to 32,767
//unsigned short   2     unsigned short int 0 to 65,535
//long 4     long int, signed long int -2,147,483,648 to 2,147,483,647
//unsigned long    4     unsigned long int  0 to 4,294,967,295
//enum *     none  Same as int
//float      4     none  3.4E +/- 38 (7 digits)
//double     8     none  1.7E +/- 308 (15 digits)
//long double 10   none  1.2E +/- 4932 (19 digits)

#ifndef IAEA_CONFIG
#define IAEA_CONFIG
```

```
#ifdef DOUBLE
typedef double IAEA_Float;
#else
typedef float IAEA_Float;
#endif

typedef short IAEA_I16;
typedef long IAEA_I32; // RCN changed to long to allow storage of EGS
    LATCH, Dec. 2006
#ifdef WIN32
typedef __int64 IAEA_I64;
#else
#if defined NO_LONG_LONG || defined LONG_IS_64
typedef long IAEA_I64;
#else
typedef long long IAEA_I64;
#endif
#endif

#ifdef __cplusplus
#define IAEA_EXTERN_C extern "C"
#else
#define IAEA_EXTERN_C extern
#endif

#ifdef WIN32

#ifdef BUILD_DLL
#define IAEA_EXPORT __declspec(dllexport)
#elif defined USE_DLL
#define IAEA_EXPORT __declspec(dllimport)
#else
#define IAEA_EXPORT
#endif
#define IAEA_LOCAL

#else

#ifdef HAVE_VISIBILITY
```

```
#define IAEA_EXPORT __attribute__ ((visibility ("default")))
#define IAEA_LOCAL __attribute__ ((visibility ("hidden")))
#else
#define IAEA_EXPORT
#define IAEA_LOCAL
#endif

#endif



#endif
```

A.2 iaea header file

```
#ifndef IAEA_HEADER
#define IAEA_HEADER

/* ************************************************************************
   */
#include "iaea_record.h"

// defines
#define SEGMENT_BEG_TOKEN '$'
#define SEGMENT_END_TOKEN ':'
#ifndef MAX_STR_LEN
  #define MAX_STR_LEN 512 /* maximum length of a string */
#endif
#define MAX_NUMB_LINES 30 /* maximum number of lines in a block */

#define MAX_NUMB_EXTRALONG_TYPES 3 /* maximum number of extra long allowed
   */
// 0: User defined generic type
// 1: Incremental history number n_hist
//    n_hist = 0 if previous primary particle scored
//    n_hist > 0 indicates how many primary particle read before the
   current one
// 2: LATCH (EGS)
// 3: ILB (PENELOPE)
// more to be defined

#define MAX_NUMB_EXTRAFLOAT_TYPES 3 /* maximum number of extra float
   allowed */
 // 0: User defined generic type
 // 1: XLAST (x coord. of the last interaction)
 // 2: YLAST (y coord. of the last interaction)
 // 3: ZLAST (z coord. of the last interaction)
 // more to be defined


struct iaea_header_type
{
```

```
FILE *fheader;
//
    ***************************************************************************
// 1. PHSP format

int file_type;          // 0 = phsp file ; 1 = phsp generator
int byte_order;         // as defined by get_byte_order routine
int record_contents[9]; // record_contents[i] = 1 or 0 (variable or
    constant)
                        // correspond to the following logical variables :
                        //            ix,iy,iz,iu.iv,iw;
                        //      iweight,iextrafloat,iextralong;

float record_constant[7]; // if record_contents[i<7] = 0
                        // then record_constant[i] contents the constant
                           value
                        // extra floats and longs are always variable
                        // so no need to store them

// contains the keyword describing each stored extrafloat
int extrafloat_contents[NUM_EXTRA_FLOAT];

// contains the keyword describing each stored extralong
int extralong_contents[NUM_EXTRA_LONG];

int record_length;
//  record_length = 1 +                                (particle)
//                4 +                                  (energy)
//                SUM(i=0;i<3) {record_contents[i]*4} (ix,iy,iz)
//                SUM(i=3;i<6) {record_contents[i]*4} (iu,iv,iw)
//                record_contents[6]*4 +              (iweigth)
//                record_contents[7]*4 +              (iextrafloat)
//                record_contents[8]*4 +              (iextralong)
IAEA_I64 checksum;

//
    ***************************************************************************
// 2. Mandatory description of the phsp
```

```
char coordinate_system_description[MAX_STR_LEN*MAX_NUMB_LINES+1];

// Counters for phsp file
IAEA_I64 orig_histories;
IAEA_I64 nParticles;
IAEA_I64 particle_number[MAX_NUM_PARTICLES];

// Event generator input file
char input_file_for_event_generator[MAX_STR_LEN*MAX_NUMB_LINES+1];

//
    ****************************************************************************
// 3. Mandatory additional information

unsigned int iaea_index; // Agency ID
char title[MAX_STR_LEN*MAX_NUMB_LINES+1];

char machine_type[MAX_STR_LEN*MAX_NUMB_LINES+1];

char MC_code_and_version[MAX_STR_LEN*MAX_NUMB_LINES+1];

float global_photon_energy_cutoff;

float global_particle_energy_cutoff;

char transport_parameters[MAX_STR_LEN*MAX_NUMB_LINES+1];

//
    ****************************************************************************
// 4. Optional description

char beam_name[MAX_STR_LEN*MAX_NUMB_LINES+1];

char field_size[MAX_STR_LEN*MAX_NUMB_LINES+1];

char nominal_SSD[MAX_STR_LEN*MAX_NUMB_LINES+1];

char variance_reduction_techniques[MAX_STR_LEN*MAX_NUMB_LINES+1];
```

```
  char initial_source_description[MAX_STR_LEN*MAX_NUMB_LINES+1];

  // Documentation sub-section
  char MC_input_filename[MAX_STR_LEN*MAX_NUMB_LINES+1];

  // Assumed to be the preferred citation
  char published_reference[MAX_STR_LEN*MAX_NUMB_LINES+1];
  char authors[MAX_STR_LEN*MAX_NUMB_LINES+1];

  char institution[MAX_STR_LEN*MAX_NUMB_LINES+1];

  char link_validation[MAX_STR_LEN*MAX_NUMB_LINES+1];

  char additional_notes[MAX_STR_LEN*MAX_NUMB_LINES+1];

  //
      ****************************************************************************

  IAEA_I64 read_indep_histories;

// CLASS FUNCTIONS

public:
      int read_header();
      int write_header();
      int print_header();
      int set_record_contents(iaea_record_type *p_iaea_record);
      int get_record_contents(iaea_record_type *p_iaea_record);
      void initialize_counters();
      void update_counters(iaea_record_type *p_iaea_record);

private:
      int read_block(char *lineread,char *blockname);
      int get_block(char *lineread);
      int get_blockname(char *line, char *blockname);
      int write_blockname(char *blockname);

      int check_byte_order();
      void print_statistics();
```

```
};

#endif
```

A.3 iaea phsp file

```
/*
 * INTERFACE FOR IAEA PHSP ROUTINES (CONTAINS ONLY DECLARATIONS)
 *
 * Copyright (C) 2006 International Atomic Energy Agency
 *
    ---------------------------------------------------------------------------
 *
 * Permission is hereby granted, free of charge, to any person obtaining a
    copy
 * of this software and associated documentation files (the "Software"), to
    deal
 * in the Software without restriction, including without limitation the
    rights
 * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
 * copies of the Software, and to permit persons to whom the Software is
    furnished
 * to do so, subject to the following conditions:
 *
 * The above copyright notice and this permission notice shall be included
    in all
 * copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
    OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
    THE
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
    FROM,
 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
 * THE SOFTWARE.
 *
 *-----------------------------------------------------------------------------
 *
 *   AUTHORS:
```

```
 *
 *   Roberto Capote Noy, PhD
 *   e-mail: R.CapoteNoy@iaea.org (rcapotenoy@yahoo.com)
 *   International Atomic Energy Agency
 *   Nuclear Data Section, P.O.Box 100
 *   Wagramerstrasse 5, Vienna A-1400, AUSTRIA
 *   Phone: +431-260021713; Fax: +431-26007
 *
 *   Iwan Kawrakow, PhD
 *   e-mail iwan@irs.phy.nrc.ca
 *   Ionizing Radiation Standards
 *   Institute for National Measurement Standards
 *   National Research Council of Canada Ottawa, ON, K1A 0R6 Canada
 *   Phone: +1-613-993 2197, ext.241; Fax: +1-613-952 9865
 *
 ******************************************************************************
 * For documentation
 * see
    http://www-nds.iaea.org/reports-new/indc-reports/indc-nds/indc-nds-0484.pdf
 ******************************************************************************/

#ifndef IAEA_PHSP
#define IAEA_PHSP

#include "iaea_config.h"

/*********************************************************************
* Initialization
*
* Given a file name header_file of length hf_length, initialize a
* new IAEA particle source, assign an unique source_ID to it and return
* this Id in result. Dont assume header_file is null-terminated as the
* function may be called from a Fortran program.
* The need for an Id arises from the fact that some applications may
* want to use several IAEA sources at once. The implementation must
    therefore
* maintain a list of already initialized sources.
* If an error occures (e.g. header file does not exist, there are errors
* in the header file, etc.), assign a negative number to result.
```

```
* (one may want to specify a list of error codes so that the application
* knows what went wrong). This function *must* be called before using
* any of the following functions for a given source id.
*
* access = 1 => opening read-only file
* access = 2 => opening file for writing
* access = 3 => opening file for appending/updating
*
**********************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_new_source(IAEA_I32 *source_ID, char *header_file,
                     const IAEA_I32 *access, IAEA_I32 *result,
                     int hf_length);

/**********************************************************************
* Maximum number of particles
*
* Set n_particle to the maximum number of particle of type type the
* source with Id id can return. If type<0, set n_particle to the
* total number of all particles. For event generators, this function
* should set n_particle to the maximum integer that can be stored in a
* signed 64 bit integer. If the source with Id id does not exist,
* set n_particle to a negative number.
**********************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_get_max_particles(const IAEA_I32 *id, const IAEA_I32 *type,
                            IAEA_I64 *n_particle);

/**********************************************************************
* Maximum energy
*
* Return the maximum energy of an initialized IAEA source with Id id
* in Emax. Set Emax to negative if a source with that Id does not exist.
**********************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_get_maximum_energy(const IAEA_I32 *id, IAEA_Float *Emax);

/**********************************************************************
* Number of additional floats and integers returned by the source
```

```
*
* Return the number of additional floats in n_extra_float and the number
* of additional integers in n_extra_integer for the source with Id id.
* Set n_extra_integer and/or n_extra_float to be negative if such a
* source does not exist.
**********************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_get_extra_numbers(const IAEA_I32 *id, IAEA_I32 *n_extra_float,
                            IAEA_I32 *n_extra_int);

/**********************************************************************
* Number of additional floats and integers to be stored
*
* Set the number of additional floats in n_extra_float and the number
* of additional integers in n_extra_integer for the source with Id id
* to be stored in the corresponding file.
**********************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_set_extra_numbers(const IAEA_I32 *id, IAEA_I32 *n_extra_float,
                                IAEA_I32 *n_extra_int);

/***********************************************************************************
* Set a type type of the extra long variable corresponding to the "index"
   number
* for a corresponding header of the phsp "id". Index is running from zero.
*
* The current list of types for extra long variables is:
*   0: User defined generic type
*   1: Incremental history number (EGS,PENELOPE)
*      = 0 indicates a nonprimary particle event
*      > 0 indicates a primary particle. The value is equal to the number of
*          primaries particles employed to get to this history after the last
*          primary event was recorded.
*   2: LATCH (EGS)
*   3: ILB (PENELOPE)
*   more to be defined
*
* Usually called before writing phsp header to set the type of extra long
```

```
* variables to be stored. It must be called once for every extralong
   variable.
*
* type = -1 means the source's header file does not exist
*          or source was not properly initialized (call iaea_new_...)
* type = -2 means the index is out of range ( 0 <= index < NUM_EXTRA_LONG )
* type = -3 means the type to be set is out of range
*          ( 1 <= type < MAX_NUMB_EXTRALONG_TYPES )
*****************************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_set_type_extralong_variable(const IAEA_I32 *id,
                                      const IAEA_I32 *index,
                                            IAEA_I32 *type);

/*****************************************************************************
* Set a type type of the extra float variable corresponding to the "index"
   number
* for a corresponding header of the phsp "id". Index is running from zero.
*
* The current list of types for extra float variables is:
*   1: XLAST (x coord. of the last interaction)
*   2: YLAST (y coord. of the last interaction)
*   3: ZLAST (z coord. of the last interaction)
*   more to be defined
*
* Usually called before writing phsp header to set the type of extra float
* variables to be stored. It must be called once for every extra float
   variable.
*
* type = -1 means the source's header file does not exist
*          or source was not properly initialized (call iaea_new_...)
* type = -2 means the index is out of range ( 0 <= index < NUM_EXTRA_FLOAT )
* type = -3 means the type to be set is out of range
*          ( 1 <= type < MAX_NUMB_EXTRAFLOAT_TYPES )
*****************************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_set_type_extrafloat_variable(const IAEA_I32 *id,
                                       const IAEA_I32 *index,
                                             IAEA_I32 *type);
```

```
/*****************************************************************************
* Get a type type of all extra variables from a header of the phsp "id".
*
* extralong_types[] AND extrafloat_types[] must have a dimension bigger than
* MAX_NUMB_EXTRALONG_TYPES and MAX_NUMB_EXTRAFLOAT_TYPES correspondingly
*
* The current list of types for extra long variables is:
*   0: User defined generic type
*   1: Incremental history number (EGS,PENELOPE)
*      = 0 indicates a nonprimary particle event
*      > 0 indicates a primary particle. The value is equal to the number of
*          primaries particles employed to get to this history after the last
*          primary event was recorded.
*   2: LATCH (EGS)
*   3: ILB (PENELOPE)
*   more to be defined
*
* The current list of types for extra float variables is:
*   1: XLAST (x coord. of the last interaction)
*   2: YLAST (y coord. of the last interaction)
*   3: ZLAST (z coord. of the last interaction)
*   more to be defined
*
* Usually called before reading phsp header to know the type of extra long
* variables to be read. It must be called once for every extra float
   variable.
*
* result = -1 means the source's header file does not exist
*            or source was not properly initialized (call iaea_new_...)
*****************************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_get_type_extra_variables(const IAEA_I32 *id, IAEA_I32 *result,
     IAEA_I32 extralong_types[], IAEA_I32 extrafloat_types[]);

/*****************************************************************************
* Set variable corresponding to the "index" number to a "constant" value
* for a corresponding header of the phsp "id". Index is running from zero.
*
```

```
*       (Usually called as needed before MC loop started)
*
*               index =  0 1 2 3 4 5 6
*          corresponds to x,y,z,u,v,w,wt
*
* Usually called before writing phsp files to set those variables which
* are not going to be stored. It must be called once for every variable
*
* constant = -1 means the source's header file does not exist
*              or source was not properly initialized (call iaea_new_...)
* constant = -2 means the index is out of range ( 0 <= index < 7 )
**************************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_set_constant_variable(const IAEA_I32 *id, const IAEA_I32 *index,
                                IAEA_Float *constant);

/**************************************************************************
* Get value of constant corresponding to the "index" number
* for a corresponding header of the phsp "id". Index is running from zero.
*
*               index =  0 1 2 3 4 5 6
*          corresponds to x,y,z,u,v,w,wt
*
* Usually called when reading phsp header info.
* It must be called once for every variable
*
*  result = -1 means the source's header file does not exist
*              or source was not properly initialized (call iaea_new_...)
*  result = -2 means the index is out of range ( 0 <= index < 7 )
*  result = -3 means that the parameter indicated by index is not a constant
**************************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_get_constant_variable(const IAEA_I32 *id, const IAEA_I32 *index,
                                IAEA_Float *constant, IAEA_I32 *result);

/***************************************************************************
* Get n_indep_particles number of statistically independent particles read
* so far from the Source with Id id.
*
```

```
* Set n_indep_particles to negative if such source does not exist.
*****************************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_get_used_original_particles(const IAEA_I32 *id,
                                      IAEA_I64 *n_indep_particles);

/*****************************************************************************
* Get Total Number of Original Particles from the Source with Id id.
*
* For a typical linac it should be equal to the total number of electrons
* incident on the primary target.
*
* Set number_of_original_particles to negative if such source does not
   exist.
*****************************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_get_total_original_particles(const IAEA_I32 *id,
                                       IAEA_I64
                                          *number_of_original_particles);

/*****************************************************************************
* Set Total Number of Original Particles for the Source with Id id.
*
* For a typical linac it should be equal to the total number of electrons
* incident on the primary target.
*
* Set number_of_original_particles to negative if such source does not
   exist.
*****************************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_set_total_original_particles(const IAEA_I32 *id,
                                       IAEA_I64 *number_of_original_particles);

/*****************************************************************************
* Partitioning for parallel runs
*
* i_parallel is the job number, i_chunk the calculation chunk,
* n_chunk the total number of calculation chunks. This function
* should divide the available phase space of source with Id id
```

```
* into n_chunk equal portions and from now on deliver particles
* from the i_chunk th portion. (i_chunk must be between 1 and n_chunk)
* The extra parameter i_parallel is needed
* for the cases where the source is an event generator and should
* be used to adjust the random number sequence.
* The variable is_ok should be set to 0 if everything went smoothly,
* or to some error code if it didnt.
***************************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_set_parallel(const IAEA_I32 *id, const IAEA_I32 *i_parallel,
                       const IAEA_I32 *i_chunk, const IAEA_I32 *n_chunk,
                       IAEA_I32 *is_ok);

/***************************************************************************
* setting the pointer to a user-specified record no. in the file
*
* record_num is the user-specified record number passed to the function.
* id is the phase space file identifier.
* The variable result should be set to 0 if everything went smoothly,
* or to some error code if it didnt.
***************************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_set_record(const IAEA_I32 *id, const IAEA_I64 *record_num,
                         IAEA_I32 *result);

/***************************************************************************
* check that the file size equals the value of checksum in the header
*
* id is the phase space file identifier. If the size of the phase space
* file is not equal to checksum, then result returns -1, otherwise result
* is set to 0.
***************************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_check_file_size_byte_order(const IAEA_I32 *id, IAEA_I32 *result);

/***************************************************************************
* Get a particle
*
* Return the next particle from the sequence of particles from source
```

```
* with Id id. Set n_stat to the number of statistically independent
* events since the last call to this function (i.e. n_stat = 0, if
* the particle resulted from the same incident electron, n_stat = 377
* if there were 377 statistically independent events sinc the last particle
* returned, etc.). If this information is not available,
* simply set n_stat to 1 if the particle belongs to a new statistically
* independent event. Set n_stat to -1, if a source with Id id does not
* exist. Set n_stat to -2, if end of file of the phase space source reached
***************************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_get_particle(const IAEA_I32 *id, IAEA_I32 *n_stat,
IAEA_I32 *type, /* particle type */
IAEA_Float *E, /* kinetic energy in MeV */
IAEA_Float *wt, /* statistical weight */
IAEA_Float *x,
IAEA_Float *y,
IAEA_Float *z, /* position in cartesian coordinates*/
IAEA_Float *u,
IAEA_Float *v,
IAEA_Float *w, /* direction in cartesian coordinates*/
IAEA_Float *extra_floats,
IAEA_I32 *extra_ints);

/***************************************************************************
* Write a particle
* n_stat = 0 for a secondary particle
* n_stat > 0 for an independent particle
*
* Write a particle to the source with Id id.
* Set n_stat to -1, if ERROR (source with Id id does not exist).
***************************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_write_particle(const IAEA_I32 *id, IAEA_I32 *n_stat,
const IAEA_I32 *type, /* particle type */
const IAEA_Float *E, /* kinetic energy in MeV */
const IAEA_Float *wt, /* statistical weight */
const IAEA_Float *x,
const IAEA_Float *y,
const IAEA_Float *z, /* position in cartesian coordinates*/
```

```
const IAEA_Float *u,
const IAEA_Float *v,
const IAEA_Float *w, /* direction in cartesian coordinates*/
const IAEA_Float *extra_floats,
const IAEA_I32 *extra_ints);

/*************************************************************************
* Destroy a source
*
* This function de-initializes the source with Id id, closing all open
* files, deallocating memory, etc. Nothing happens if a source with that
* id does not exist. Header is updated.
*************************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_destroy_source(const IAEA_I32 *source_ID, IAEA_I32 *result);

/*************************************************************************
* Print the current header associated to source id
*
* result is set to negative if phsp source does not exist.
*************************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_print_header(const IAEA_I32 *source_ID, IAEA_I32 *result);

/*************************************************************************
* Copy header of the source_id to the header of the destiny_id
*************************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_copy_header(const IAEA_I32 *source_ID, const IAEA_I32 *destiny_ID,
                      IAEA_I32 *result);

/*************************************************************************
* Update header of the source_id
*************************************************************************/
IAEA_EXTERN_C IAEA_EXPORT
void iaea_update_header(const IAEA_I32 *source_ID, IAEA_I32 *result);

#endif
```

B

MCNP Codes

Due to the excessive size of the phase space used in MCNP simulations presented in here, the beam simulation has been replaced by a monoenergetic beam. The geometry of the phantoms, detectors, and materials used are the same used in the phase space simulations.

We presented the simulations of the percentage depth dose for the water phantom simulation and the $QUASAR^{TM}$ Multi-Purpose Body Phantom following the size and real geometry of the QUASAR. The simulation of the beam profiles follows the same structure from the water phantom except for the position of the detectors.

B.1 Percentage depth dose simulation

```
Percentage Depth Dose
C Water phanton simulation with MCNP6
C (2017) coded by the authors at the
C Instituto Politecnico Nacional and
C Instituto Nacional de Investigaciones Nucleares
C If you wish to use this simulation, the authors would fully
C appreciate you mentioning the origin of this file in your article
C or work.
C Please, consider that the input must be used only for the purpose
C of non-profit research.
C If you need additional information, please contact us.
C E-mail: fis.ivonsanchez@gmail.com, arturo.delfin@inin.gob.mx
C ==========================================
C CELL CARDS
```

```
C ========================================
10  10   0.1001 -10 +100 +101 +102 +103 +104 +105 +106 +107 +108
            +109 +110 +111 +112 +113 +114 +115 +116 +117 +118
            +119 +120 +121 +122 +123 +124 +125 +126 +127 +128
            +129 +130 +131 +132 +133 +134 +135 +136 +137 +138
            +139 +140 +141 +142 +143 +144 +145 +146 +147 +148
            +149 +150 +151 +152 +153 +154 +155 +156 +157 +158
            +159 +160 +161                                imp:p=1
20  20     0.1074 -20  +10                                imp:p=1
30  30 0.00004988 -30                                     imp:p=1
40  30 0.00004988 -40 +20  +30                            imp:p=1
100 10 0.1001 -100                                        imp:p=1
101 10 0.1001 -101                                        imp:p=1
102 10 0.1001 -102                                        imp:p=1
103 10 0.1001 -103                                        imp:p=1
104 10 0.1001 -104                                        imp:p=1
105 10 0.1001 -105                                        imp:p=1
106 10 0.1001 -106                                        imp:p=1
107 10 0.1001 -107                                        imp:p=1
108 10 0.1001 -108                                        imp:p=1
109 10 0.1001 -109                                        imp:p=1
110 10 0.1001 -110                                        imp:p=1
111 10 0.1001 -111                                        imp:p=1
112 10 0.1001 -112                                        imp:p=1
113 10 0.1001 -113                                        imp:p=1
114 10 0.1001 -114                                        imp:p=1
115 10 0.1001 -115                                        imp:p=1
116 10 0.1001 -116                                        imp:p=1
117 10 0.1001 -117                                        imp:p=1
118 10 0.1001 -118                                        imp:p=1
119 10 0.1001 -119                                        imp:p=1
120 10 0.1001 -120                                        imp:p=1
121 10 0.1001 -121                                        imp:p=1
122 10 0.1001 -122                                        imp:p=1
123 10 0.1001 -123                                        imp:p=1
124 10 0.1001 -124                                        imp:p=1
125 10 0.1001 -125                                        imp:p=1
126 10 0.1001 -126                                        imp:p=1
127 10 0.1001 -127                                        imp:p=1
```

```
128 10 0.1001 -128                                          imp:p=1
129 10 0.1001 -129                                          imp:p=1
130 10 0.1001 -130                                          imp:p=1
131 10 0.1001 -131                                          imp:p=1
132 10 0.1001 -132                                          imp:p=1
133 10 0.1001 -133                                          imp:p=1
134 10 0.1001 -134                                          imp:p=1
135 10 0.1001 -135                                          imp:p=1
136 10 0.1001 -136                                          imp:p=1
137 10 0.1001 -137                                          imp:p=1
138 10 0.1001 -138                                          imp:p=1
139 10 0.1001 -139                                          imp:p=1
140 10 0.1001 -140                                          imp:p=1
141 10 0.1001 -141                                          imp:p=1
142 10 0.1001 -142                                          imp:p=1
143 10 0.1001 -143                                          imp:p=1
144 10 0.1001 -144                                          imp:p=1
145 10 0.1001 -145                                          imp:p=1
146 10 0.1001 -146                                          imp:p=1
147 10 0.1001 -147                                          imp:p=1
148 10 0.1001 -148                                          imp:p=1
149 10 0.1001 -149                                          imp:p=1
150 10 0.1001 -150                                          imp:p=1
151 10 0.1001 -151                                          imp:p=1
152 10 0.1001 -152                                          imp:p=1
153 10 0.1001 -153                                          imp:p=1
154 10 0.1001 -154                                          imp:p=1
155 10 0.1001 -155                                          imp:p=1
156 10 0.1001 -156                                          imp:p=1
157 10 0.1001 -157                                          imp:p=1
158 10 0.1001 -158                                          imp:p=1
159 10 0.1001 -159                                          imp:p=1
160 10 0.1001 -160                                          imp:p=1
161 10 0.1001 -161                                          imp:p=1
800  0          +40                                         imp:p=0

C ==========================================
C SURFACE CARDS
C ==========================================
```

```
C WATER
10   RPP -33.75 33.75 -32.25 32.25 -128   -72.0
C WALLS OF THE PHANTOM OF 15 mm
20   RPP -35.25 35.25 -33.75 33.75 -129.5 -72.0
C REMOVES THE UPPER WALL
30   RPP -35.25 35.25 -33.75 33.75 -72.0 -71.0
c AIR OUTSIDE THE PHANTOM
40   RPP -37.75 37.75 -35.25 35.25 -130.0 +10.0
c DETECTORS
100 RPP -0.5 0.5 -0.5 0.5 -103.0 -102.5
101 RPP -0.5 0.5 -0.5 0.5 -102.5 -102.0
102 RPP -0.5 0.5 -0.5 0.5 -102.0 -101.5
103 RPP -0.5 0.5 -0.5 0.5 -101.5 -101.0
104 RPP -0.5 0.5 -0.5 0.5 -101.0 -100.5
105 RPP -0.5 0.5 -0.5 0.5 -100.5 -100.0
106 RPP -0.5 0.5 -0.5 0.5 -100.0 -99.5
107 RPP -0.5 0.5 -0.5 0.5 -99.5 -99.0
108 RPP -0.5 0.5 -0.5 0.5 -99.0 -98.5
109 RPP -0.5 0.5 -0.5 0.5 -98.5 -98.0
110 RPP -0.5 0.5 -0.5 0.5 -98.0 -97.5
111 RPP -0.5 0.5 -0.5 0.5 -97.5 -97.0
112 RPP -0.5 0.5 -0.5 0.5 -97.0 -96.5
113 RPP -0.5 0.5 -0.5 0.5 -96.5 -96.0
114 RPP -0.5 0.5 -0.5 0.5 -96.0 -95.5
115 RPP -0.5 0.5 -0.5 0.5 -95.5 -95.0
116 RPP -0.5 0.5 -0.5 0.5 -95.0 -94.5
117 RPP -0.5 0.5 -0.5 0.5 -94.5 -94.0
118 RPP -0.5 0.5 -0.5 0.5 -94.0 -93.5
119 RPP -0.5 0.5 -0.5 0.5 -93.5 -93.0
120 RPP -0.5 0.5 -0.5 0.5 -93.0 -92.5
121 RPP -0.5 0.5 -0.5 0.5 -92.5 -92.0
122 RPP -0.5 0.5 -0.5 0.5 -92.0 -91.5
123 RPP -0.5 0.5 -0.5 0.5 -91.5 -91.0
124 RPP -0.5 0.5 -0.5 0.5 -91.0 -90.5
125 RPP -0.5 0.5 -0.5 0.5 -90.5 -90.0
126 RPP -0.5 0.5 -0.5 0.5 -90.0 -89.5
127 RPP -0.5 0.5 -0.5 0.5 -89.5 -89.0
128 RPP -0.5 0.5 -0.5 0.5 -89.0 -88.5
129 RPP -0.5 0.5 -0.5 0.5 -88.5 -88.0
```

```
130 RPP -0.5 0.5 -0.5 0.5 -88.0 -87.5
131 RPP -0.5 0.5 -0.5 0.5 -87.5 -87.0
132 RPP -0.5 0.5 -0.5 0.5 -87.0 -86.5
133 RPP -0.5 0.5 -0.5 0.5 -86.5 -86.0
134 RPP -0.5 0.5 -0.5 0.5 -86.0 -85.5
135 RPP -0.5 0.5 -0.5 0.5 -85.5 -85.0
136 RPP -0.5 0.5 -0.5 0.5 -85.0 -84.5
137 RPP -0.5 0.5 -0.5 0.5 -84.5 -84.0
138 RPP -0.5 0.5 -0.5 0.5 -84.0 -83.5
139 RPP -0.5 0.5 -0.5 0.5 -83.5 -83.0
140 RPP -0.5 0.5 -0.5 0.5 -83.0 -82.5
141 RPP -0.5 0.5 -0.5 0.5 -82.5 -82.0
142 RPP -0.5 0.5 -0.5 0.5 -82.0 -81.5
143 RPP -0.5 0.5 -0.5 0.5 -81.5 -81.0
144 RPP -0.5 0.5 -0.5 0.5 -81.0 -80.5
145 RPP -0.5 0.5 -0.5 0.5 -80.5 -80.0
146 RPP -0.5 0.5 -0.5 0.5 -80.0 -79.5
147 RPP -0.5 0.5 -0.5 0.5 -79.5 -79.0
148 RPP -0.5 0.5 -0.5 0.5 -79.0 -78.5
149 RPP -0.5 0.5 -0.5 0.5 -78.5 -78.0
150 RPP -0.5 0.5 -0.5 0.5 -78.0 -77.5
151 RPP -0.5 0.5 -0.5 0.5 -77.5 -77.0
152 RPP -0.5 0.5 -0.5 0.5 -77.0 -76.5
153 RPP -0.5 0.5 -0.5 0.5 -76.5 -76.0
154 RPP -0.5 0.5 -0.5 0.5 -76.0 -75.5
155 RPP -0.5 0.5 -0.5 0.5 -75.5 -75.0
156 RPP -0.5 0.5 -0.5 0.5 -75.0 -74.5
157 RPP -0.5 0.5 -0.5 0.5 -74.5 -74.0
158 RPP -0.5 0.5 -0.5 0.5 -74.0 -73.5
159 RPP -0.5 0.5 -0.5 0.5 -73.5 -73.0
160 RPP -0.5 0.5 -0.5 0.5 -73.0 -72.5
161 RPP -0.5 0.5 -0.5 0.5 -72.5 -72.0

C ==========================================
C DATA CARDS
C ==========================================
C MONOENERGETIC BEAM OF 6MV IN A 10X10 FIELD
MODE  P E
SDEF  POS=0 0 0 X=d1 Y=d2 Z=0 PAR=2 ERG=6.00 VEC=0 0 -1 DIR=1
```

```
SI1  -5    5
SP1   0    1
SI2  -5    5
SP2   0    1
*F08:p 100 101 102 103 104 105 106 107 108 109 110
*F18:p 111 112 113 114 115 116 117 118 119 120 121
*F28:p 122 123 124 125 126 127 128 129 130 131 132
*F38:p 133 134 135 136 137 138 139 140 141 142 143
*F48:p 144 145 146 147 148 149 150 151 152 153 154
*F58:p 155 156 157 158 159 160 161
C WATER
m10
       1001   0.666657
       8016   0.333343
C LUCIT
m20
       1000   0.533320
       6000   0.333345
       8000   0.133335
C AIR
m30
       6000   0.000150
       7000   0.784431
       8000   0.210748
      18000   0.004671
C
NPS    50 000 000
print 30
```

B.2 QUASAR phantom geometry simulation

```
Quasar phantom simulation
C Water phanton simulation with MCNP6
C (2017) coded by the authors at the
C Instituto Politecnico Nacional and
C Instituto Nacional de Investigaciones Nucleares
C If you wish to use this simulation, the authors would fully
C appreciate you mentioning the origin of this file in your article
C or work.
C Please, consider that the input must be used only for the purpose
C of non-profit research.
C If you need additional information, please contact us.
C E-mail: fis.ivonsanchez@gmail.com, arturo.delfin@inin.gob.mx
C ========================================
C CELL CARDS
C ========================================
10   10  -0.296000 -10 +50 +270 +260 +250 +240 +120 +130 +140 +150 imp:p=1
20   10  -0.296000 -20                                          imp:p=1
30   20  -1.127000 -30 +10 +20 +290 +280 +100 +110               imp:p=1
40   30  -0.001205 -40 +30                                      imp:p=1
50   20  -1.127000 -50 +160 +170 +180 +190 +200 +210 +220 230 imp:p=1
100  10  -0.296000 -100                                         imp:p=1
110  10  -0.296000 -110                                         imp:p=1
120  10  -0.296000 -120                                         imp:p=1
130  10  -0.296000 -130                                         imp:p=1
140  10  -0.296000 -140                                         imp:p=1
150  10  -0.296000 -150                                         imp:p=1
160  10  -0.296000 -160                                         imp:p=1
170  10  -0.296000 -170                                         imp:p=1
180  10  -0.296000 -180                                         imp:p=1
190  10  -0.296000 -190                                         imp:p=1
200  10  -0.296000 -200                                         imp:p=1
210  10  -0.296000 -210                                         imp:p=1
220  10  -0.296000 -220                                         imp:p=1
230  10  -0.296000 -230                                         imp:p=1
240  10  -0.296000 -240                                         imp:p=1
250  10  -0.296000 -250                                         imp:p=1
```

```
260  10  -0.296000 -260                                        imp:p=1
270  10  -0.296000 -270                                        imp:p=1
280  10  -0.296000 -280                                        imp:p=1
290  10  -0.296000 -290                                        imp:p=1
1000   0         +40                                           imp:p=0

C ==========================================
C SURFACE CARDS
C ==========================================
C RIGHT LUNG (CYLINDER)
10  rcc  +0.000 -6.00 -100.000 0.000  12.000 +00.000 +04.00
C LEFT LUNG (CYLINDER)
20  rcc  +19.00 -6.00 -100.000 0.000  12.000 +00.000 +04.00
C QUASAR PMMA
30  rec  +9.5  -6.000 -100.000 00.000 12.000 +00.000
         +15.000 00.000 +00.000 +10.000
C AIR OUTSIDE THE QUASAR
40  rpp  -30.00  +30.00 -30.00 +30.00 -130.00 +10.00
C TUMOR SIMULATION
50  rcc  +0.0  -2.00 -100.0 0.00 +4.00 +0.00 +2.0
C DETECTORS
100 rpp -0.25 +0.25 -0.25 +0.25 -105 -104.5
110 rpp -0.25 +0.25 -0.25 +0.25 -104.5 -104.05
120 rpp -0.25 +0.25 -0.25 +0.25 -103.95 -103.5
130 rpp -0.25 +0.25 -0.25 +0.25 -103.5 -103
140 rpp -0.25 +0.25 -0.25 +0.25 -103 -102.5
150 rpp -0.25 +0.25 -0.25 +0.25 -102.5 -102.05
160 rpp -0.25 +0.25 -0.25 +0.25 -101.95 -101.5
170 rpp -0.25 +0.25 -0.25 +0.25 -101.5 -101
180 rpp -0.25 +0.25 -0.25 +0.25 -101 -100.5
190 rpp -0.25 +0.25 -0.25 +0.25 -100.5 -100
200 rpp -0.25 +0.25 -0.25 +0.25 -100 -99.5
210 rpp -0.25 +0.25 -0.25 +0.25 -99.5 -99
220 rpp -0.25 +0.25 -0.25 +0.25 -99 -98.5
230 rpp -0.25 +0.25 -0.25 +0.25 -98.5 -98.05
240 rpp -0.25 +0.25 -0.25 +0.25 -97.95 -97.5
250 rpp -0.25 +0.25 -0.25 +0.25 -97.5 -97
260 rpp -0.25 +0.25 -0.25 +0.25 -97 -96.5
270 rpp -0.25 +0.25 -0.25 +0.25 -96.5 -96.05
```

```
280 rpp -0.25 +0.25 -0.25 +0.25 -95.95 -95.5
290 rpp -0.25 +0.25 -0.25 +0.25 -95.5 -95

C ========================================
C DATA CARDS
C ========================================
C MONOENERGETIC BEAM OF 6MV IN A 10X10 FIELD
MODE  P E
SDEF  POS=0 0 0 X=d1 Y=d2 Z=0 PAR=2 ERG=6.00 VEC=0 0 -1 DIR=1
SI1  -5    5
SP1   0    1
SI2  -5    5
SP2   0    1
*F08:p 100 110 120 130 140 150 160 170 180 190
*F98:p 200 210 220 230 240 250 260 270 280 290
C MATERIALS
C WATER
m10
       1001   0.666657 $ atm/bar-cm
       8016   0.333343 $ atm/bar-cm
C LUCIT
m20
       1000   0.533320 $ atm/bar-cm
       6000   0.333345 $ atm/bar-cm
       8000   0.133335 $ atm/bar-cm
C AIR
m30
       6000   0.000150 $ atm/bar-cm
       7000   0.784431 $ atm/bar-cm
       8000   0.210748 $ atm/bar-cm
      18000   0.004671 $ atm/bar-cm
C
nps 50 000 000
print 30
```

C

Geant4 water phantom input file

We presented the simulations of the water phantom for the measurements of the contruction for the percentage depth dose, x and y profiles (off-axis ratio) measurements.

C.1 Water Phantom simuation

The water phantom construction is simulated in the Geant4 class of DetectorConstruction.cc. The used code is enlisted below. The implemented code is a modification of the code provided by the scientific and technical work of the Geant4 collaboration.

```
#include "B1DetectorConstruction.hh"

#include "G4RunManager.hh"
#include "G4NistManager.hh"
#include "G4Box.hh"
#include "G4Cons.hh"
#include "G4Orb.hh"
#include "G4Sphere.hh"
#include "G4Trd.hh"
#include "G4LogicalVolume.hh"
#include "G4PVPlacement.hh"
#include "G4SystemOfUnits.hh"

B1DetectorConstruction::B1DetectorConstruction()
: G4VUserDetectorConstruction(),
  fScoringVolume(0)
```

```
{ }

B1DetectorConstruction::~B1DetectorConstruction()
{ }

G4VPhysicalVolume* B1DetectorConstruction::Construct()
{
  // Get nist material manager
  G4NistManager* nist = G4NistManager::Instance();

  // Envelope parameters MEASUREMENT CHANGE
  //
  G4double env_sizeXY = 200*cm, env_sizeZ = 300*cm;
  G4Material* env_mat = nist->FindOrBuildMaterial("G4_AIR");

  // Option to switch on/off checking of volumes overlaps
  //
  G4bool checkOverlaps = true;

  //
  G4double world_sizeXY = 1.2*env_sizeXY;
  G4double world_sizeZ = 1.2*env_sizeZ;
  G4Material* world_mat = nist->FindOrBuildMaterial("G4_AIR");

  G4Box * solidWorld =
    new G4Box("World",                    //its name
       0.5*world_sizeXY, 0.5*world_sizeXY, 0.5*world_sizeZ); //its size

  G4LogicalVolume * logicWorld =
    new G4LogicalVolume(solidWorld,       //its solid
                        world_mat,        //its material
                        "World");         //its name

  G4VPhysicalVolume * physWorld =
    new G4PVPlacement(0,                  //no rotation
                      G4ThreeVector(0,0,0), //at (0,0,0)
                      logicWorld,         //its logical volume
                      "World",            //its name
                      0,                  //its mother volume
```

```
                  false,              //no boolean operation
                  0,                  //copy number
                  checkOverlaps);     //overlaps checking

//
// Envelope
//
G4Box * solidEnv =
  new G4Box("Envelope",               //its name
      0.5*env_sizeXY, 0.5*env_sizeXY, 0.5*env_sizeZ); //its size

G4LogicalVolume * logicEnv =
  new G4LogicalVolume(solidEnv,       //its solid
                    env_mat,          //its material
                    "Envelope");      //its name

new G4PVPlacement(0,                  //no rotation
                G4ThreeVector(),      //at (0,0,0)
                logicEnv,             //its logical volume
                "Envelope",           //its name
                logicWorld,           //its mother volume
                false,                //no boolean operation
                0,                    //copy number
                checkOverlaps);       //overlaps checking
//
// Shape 1
//

  G4Material* shape1_mat = nist->FindOrBuildMaterial("G4_A-150_TISSUE");
  G4Material* agua = nist->FindOrBuildMaterial("G4_WATER");

// ========================================================
// THE MEASUREMENTS ARE HALF THE SIZE OF THE BOX AS SPECIFY
// ========================================================

G4VSolid* pBoxSolid =
 new G4Box ("aBoxSolid",
       33.0*cm, 31.5*cm, 27.25*cm);
```

```
  G4LogicalVolume* pBoxLogic=
        new G4LogicalVolume( pBoxSolid,
            shape1_mat,
      "aBoxSolid",
      0,0,0);

  G4VPhysicalVolume* aBoxPhys =
   new G4PVPlacement(0,
      G4ThreeVector(0,0,0),
      "aBoxSolid",
      pBoxLogic,
      physWorld,
      false,
      0);

 // AAPM WALLS Z

  G4VSolid* wBoxSolid =
        new G4Box ("bBoxSolid",
                  33.0*cm, 31.5*cm, 0.75*cm);

  G4LogicalVolume* wBoxLogic=
        new G4LogicalVolume(wBoxSolid,
               agua,
               "bBoxSolid",
               0,0,0);

G4VPhysicalVolume* wBoxPhys =
        new G4PVPlacement(0,
               G4ThreeVector(0,0,-28*cm),
               "bBoxSolid",
               wBoxLogic,
       physWorld,
               false,
               0);

// APPM WALL 1

  G4VSolid* w1BoxSolid =
```

```
        new G4Box ("cBoxSolid",
                  33.0*cm, 0.75*cm, 27.25*cm);

  G4LogicalVolume* w1BoxLogic=
        new G4LogicalVolume(w1BoxSolid,
               agua,
               "cBoxSolid",
               0,0,0);

G4VPhysicalVolume* w1BoxPhys =
        new G4PVPlacement(0,
               G4ThreeVector(0,32.25*cm,0),
               "cBoxSolid",
               w1BoxLogic,
      physWorld,
               false,
      0);


G4VPhysicalVolume* w2BoxPhys =
        new G4PVPlacement(0,
               G4ThreeVector(0,-32.25*cm,0),
               "cBoxSolid",
               w1BoxLogic,
      physWorld,
               false,
      0);


//AAPM WALL 2


  G4VSolid* w3BoxSolid =
        new G4Box ("dBoxSolid",
                  0.75*cm, 31.5*cm, 27.5*cm);

  G4LogicalVolume* w3BoxLogic=
        new G4LogicalVolume(w3BoxSolid,
               agua,
```

```
                "dBoxSolid",
                0,0,0);

G4VPhysicalVolume* w3BoxPhys =
        new G4PVPlacement(0,
                G4ThreeVector(33.75*cm,0,0),
                "dBoxSolid",
                w3BoxLogic,
      physWorld,
                false,
      0);


G4VPhysicalVolume* w4BoxPhys =
        new G4PVPlacement(0,
                G4ThreeVector(-33.75*cm,0,0),
                "dBoxSolid",
                w3BoxLogic,
      physWorld,
                false,
      0);
  return physWorld;
}
```