MONTE CARLO SIMULATION OF A MEDICAL LINEAR ACCELERATOR FOR GENERATION OF PHASE SPACES

Alex C. H. Oliveira$^{1,2}$, José W. Vieira$^{1,3,4}$, Marcelo G. Santana$^2$, Fernando R. A. Lima$^{1,2}$

$^1$ Universidade Federal de Pernambuco (UFPE)  
Av. Prof. Moraes Rego, 1235  
50670-901 Recife, PE  
oliveira_ach@yahoo.com

$^2$ Centro Regional de Ciências Nucleares (CRCN-NE / CNEN)  
Av. Prof. Luiz Freire, 200  
50740-540 Recife, PE

$^3$ Instituto Federal de Educação, Ciência e Tecnologia (IFPE)  
Av. Prof Luiz Freire, 500  
50740-540 Recife, PE

$^4$ Escola Politécnica de Pernambuco (UPE)  
Rua Benfica, 455  
50750-470 Recife, PE

ABSTRACT

Radiotherapy uses various techniques and equipment for local treatment of cancer. The equipment most often used in radiotherapy to the patient irradiation are linear accelerators (Linacs) which produce beams of X-rays in the range 5-30 MeV. Among the many algorithms developed over recent years for evaluation of dose distributions in radiotherapy planning, the algorithms based on Monte Carlo methods have proven to be very promising in terms of accuracy by providing more realistic results. The MC methods allow simulating the transport of ionizing radiation in complex configurations, such as detectors, Linacs, phantoms, etc. The MC simulations for applications in radiotherapy are divided into two parts. In the first, the simulation of the production of the radiation beam by the Linac is performed and then the phase space is generated. The phase space contains information such as energy, position, direction, etc. of millions of particles (photons, electrons, positrons). In the second part the simulation of the transport of particles (sampled phase space) in certain configurations of irradiation field is performed to assess the dose distribution in the patient (or phantom). The objective of this work is to create a computational model of a 6 MeV Linac using the MC code Geant4 for generation of phase spaces. From the phase space, information was obtained to assess beam quality (photon and electron spectra and two-dimensional distribution of energy) and analyze the physical processes involved in producing the beam.

1. INTRODUCTION

Cancer is responsible for over 12% of all causes of death in the world: more than 7 million people die of this disease annually. As life expectancy has improved gradually, cancer incidence, estimated at 12.7 million new cases in 2008, will reach almost 22.2 million till 2030, according to the latest estimates from the International Agency for Research on Cancer (IARC). Therefore, the cancer treatment will continue to be one of the greatest health challenges of the 21st century. One of the most important and commonly used techniques is radiotherapy [1]. Approximately 52% of patients are undergoing radiotherapy at least once during their cancer treatment [2,3].
Radiotherapy is local treatment of cancer with ionizing radiation. It uses a variety of techniques and equipment to irradiate the affected areas of the human body [4]. The equipment more often used in radiotherapy is the linear accelerator (Linac) which produce X-ray beams in the range from 5 to 30 MeV [5,6].

The current radiotherapy planning systems use images to model the geometry and internal structures of the patient. Furthermore, provide some method of heterogeneity correction in way that the changes in terms of density can be considered during the evaluation of the dose. The accuracy of the method of heterogeneity correction is nowadays the main feature of a radiotherapy planning system. Among many algorithms developed over the last years, methods based on Monte Carlo (MC) have proven to be very promising in terms of accuracy, providing more realistic results [7]. Thus, computational methods based on MC are becoming the most advanced technique to evaluate dose distributions. Adding the fact that the computational power of processors has increased dizzyly, it has become possible to use MC methods in radiotherapy planning systems in a plausible time for clinical practice [2,6].

The dosimetric evaluation procedure in radiotherapy is normally divided into two stages (Fig. 1). The first refers to simulation of the production of the radiation beam from Linac and the generation of phase space. The phase space is a set of information about the particle state (energy, position, direction, etc.) [5]. In the second stage, the particles which constitute the phase space are transported in the object representation that will be irradiated (patient, physical phantom, etc.) and dose distribution is evaluated. Conventionally, the object representation is obtained from the computed tomography images thereof. The focus of this work is the first stage. The goal is to create a computational model of a 6 MV Linac photon beam, using the Geant4 MC code to generate phase spaces.

Figure 1: Representation of the MC simulation model commonly used for applications in radiotherapy divided into two stages.
2. MONTE CARLO SIMULATIONS

The MC methods perform numerical simulation of problems using essentially a sequence of random numbers. These statistical methods may be used to simulate the behavior of physical systems, mathematical, biological, etc. which can be described by random sampling of probability density functions (PDFs).

2.1. Simulation of Radiation Transport

In the simulation of radiation transport using MC methods, the history of a particle is defined as a sequence of tracks where each track ends with an interaction event where the particle can change its direction, lose energy and occasionally produce secondary particles. The history ends when it leaves the region of interest or when its energy is lower than the predefined cutoff energy. In the latter case, the remaining energy is deposited at the point where the transport of the particle was stopped [8-10].

Due to the importance given to the application of MC methods to transport radiation, various software or code systems were developed and made available, mostly for free [11]. The Geant4 is one of the main MC codes used in medical physics, including innovative areas such as radiotherapy with heavy particles and image-guided radiation therapy [12,13].

2.2. Simulation for Parallel Computing

MC codes frequently allow evaluate the radiation effects with great accuracy. However, the accuracy of the simulation is a direct function of the number of histories and, consequently, of the simulation time. A long simulation time is inappropriate for some applications that would benefit much from its accuracy, but require a quick response. Parallel processing have been widely exploited as a suitable technique to reduce the simulation time [14]. Parallel computing is a computing way which various processes are executed simultaneously using hardware resources to solve a particular problem [15].

The G4MPI is a native interface of the Geant4 for parallel computing that uses the libraries MPI (Message-Passing Interface) and implements the parallelism for simulation of separated histories on remote processors. Using this interface, user applications can be parallelized with different MPI libraries such as LAM/MPI, MPICH2, OpenMPI, etc. [12].

2.3. Simulation of Linear Accelerators

To simulate the production of the photon beam of a Linac, it is necessary to define in detail the components of the head which have influence in the output beam. Commonly, these components are the target, the primary collimator, flattening filter, the ionization chamber, the mirror and the secondary collimator (Fig. 2a). The target, the primary collimator and flattening filter are the components of the Linac that have the most influence on the shape of the photon spectrum. Information about the geometry and the materials of the components of a Linac are obtained from its manufacturer [16-20].
Figure 2: Components typical of a Linac head (a). Beam of the Linac without (b) and with (c) the flattening filter.

In a Linac designed to radiotherapy, electrons gain energy by interacting with an electromagnetic field of synchronized radio frequency rather than accelerating by the potential difference. The accelerator structure consists of a long cylindrical tube having a series of deflectors circular. [5] When the electron beam leaves the accelerator tube, it is bent by magnetic fields, so it forms a collimated beam approximately 2 to 3 mm in diameter and hit the target normally [21-23]. Thus, the primary electron beam can be determined by two main characteristics: the size (diameter) of the focal point and the energy average. These parameters are typically determined by trial and error, by performing comparisons between calculations and measurements of dose distributions [17,24,25]. In MC simulations, these characteristics are modeled by Gaussian distributions, by adopting values for the standard deviation (σ) or the FWHM (Full Width at Half Maximum) [20,23,25-28].

The targets used in Linacs are usually tungsten or a laminate of copper-tungsten thin enough to completely stop the primary electrons [5,17,21]. The energy lost by the electrons is converted into X-ray photons whose main emission is directed forward (Fig. 2b). To compensate for the lack of scattering at the edge of the field, it is used a flattening filter that projects deliberately a beam profile which increases toward the edges (Fig. 2c). The simplest filters have a conical shape and are generally circularly symmetric [5,29]. The material should be made of medium atomic number, such as steel and copper [17,20].

The head of a Linac has two collimators to ensure that only the treatment area is irradiated. The primary collimator is fixed and the secondary one (also called "jaws") is adjustable. The material chosen for both is usually tungsten. The primary collimator field defines the maximum beam of X-rays, typically 40 × 40 cm to 100 cm from the source. The opening therethrough has the shape of a truncated cone or four sides pyramid. The secondary collimators consist of two pairs of jaws, one above the other. Conventionally, the jaws are labeled X1, X2 and Y1, Y2. Its edges often define arcs such that their inner faces are approximately tangential to the beam of radiation, thereby reducing the penumbra [5,20,29].
In order to enable the location of the X-ray beam, a light beam is generated to match the limits of the useful radiation beam. This is achieved by means of reflection of a high intensity light source located outside of the radiation field [5].

A detector of ionization chamber type is used to obtain measurements of the amount of radiation provided by the accelerator and also to control the uniformity and symmetry of the beam [5,21]. The ionization chambers are of transmission type of parallel plate with large diameter for monitoring the total field [29]. In some Linacs, the signal of the ionization chamber is affected by the position of the secondary collimator. Backscattered particles of collimators can generate additional charges in the chamber. In small fields, will occur more backscattering. The magnitude of this effect is normally limited to a few percent, but for some types of Linacs, great effects have been reported [17].

To assess the dose distribution in a particular geometry, it is necessary that the state of particles in the incident beam is accurately known, i.e., energies, directions and positions of the photons, electrons, and positrons. This data set is called phase space [11]. The obtaining of the phase space in MC simulations of Linacs is accomplished by defining a sensitive volume that stores the information of the particles that pass through it. Generally, this volume is a thin circular cylinder located just above the secondary collimator. In this case the phase space becomes a virtual Linac and can be used in different simulations. However, one should pay attention to the amount of particles stored in the phase space so will be sufficient sample to obtain the expected accuracy [2,17,26].

The Geant4 has several fully coded examples that demonstrate the implementation of the classes needed to build a custom simulation [12]. The example MedLinac2 (examples/advanced/medical_linac) of Geant4, developed by Caccia et al (2010), simulates a typical Linac used in radiotherapy. In this example, the electron beam is generated with Gaussian energy and directional distribution. Furthermore, it allows the generation of phase space. The MedLinac2 was subjected to various tests which resulted in a good agreement with the experimental data [2].

### 3. METHODOLOGY AND RESULTS

#### 3.1. Monte Carlo Simulation

To perform a simulation in Geant4 is needed basically write the codes describing the geometry, the materials used, the particles of interest, the physical processes of interaction and the function main that indicates where the program will start and manage its execution.

The MC simulation of the head of a 6 MeV Linac was performed in Geant4 (version 9.6) installed on a computer running the Linux operating system (Ubuntu). This application has been called SSLinacs (Simulation System of Linacs). The physical model that was used the G4EmStandardPhysics_option4. The limit to produce secondary particles was set equal to 0.5 mm [23].

From the literature review, were set default values for the parameters that characterize the main components of a 6 MeV Linac. These values are the standard for the simulations and...
can be modified with the purpose of adjustments of results or to simulate new models of Linac head.

### 3.1.1. Electron Beam

The primary electron beam is defined as having a Gaussian energy and spatial distribution. For the spatial distribution, the used value of the parameter FWHM was 2 mm [17,22,30] and the energy distribution, 3% of the mean energy (6 MeV) [23,25-27]. Figure 3 shows how was performed the implementation of the initial state of electrons. The first and second argument of the function shoot are, respectively, the mean value and the standard deviation of the Gaussian distribution.

```cpp
ParticleGun->SetParticlePosition(G4ThreeVector(CLHEP::RandGauss::shoot(0, 0.84932),
                                              CLHEP::RandGauss::shoot(0, 0.84932), z0));
ParticleGun->SetParticleEnergy(CLHEP::RandGauss::shoot(6*MeV, 0.01274*6*MeV));
```

**Figure 3:** Implementation of the initial states of the electrons.

### 3.1.2. Geometry and Materials

The target of the Linac was defined as two cylinders, one made of tungsten and the other made of copper with 3 mm and 1.5 mm in height, respectively, and with 10 mm in diameter [18]. The primary collimator was modeled as a cylinder of tungsten with 50 mm in height and 120 mm in diameter with conical opening. The dimensions of the truncated cone were defined properly to form a circular irradiation field with $240\,\text{cm}$ in diameter, at $100\,\text{cm}$ from the source. The flattening filter was modeled as two truncated cones and a cylinder (all of copper): the first truncated cone has $14.5\,\text{mm}$ high, $1.0\,\text{mm}$ superior radius and $8.5\,\text{mm}$ bottom radius; the second truncated cone has $27.0\,\text{mm}$ high, $8.5\,\text{mm}$ superior radius and $34.0\,\text{mm}$ bottom radius; and the cylinder has $0.5\,\text{mm}$ height and $38.5\,\text{mm}$ radius. The modelings for the ionization chamber and the mirror were obtained from the codes of MedLinac2. Basically, the ionization chamber is six layers of polyimide interspersed by layers of air and the mirror is Mylar with 12 degrees inclination. The secondary collimator was set as four parallelepipeds of tungsten whose dimensions were established in such a way to block the X-ray beam for a field irradiating of $40 \times 40\,\text{cm}$ at $100\,\text{cm}$ from the source. Figure 4 illustrates the SSLinacs running. The image was taken in view mode of the Geant4.

### 3.1.3. Phase Space

The volume of detection for obtaining the phase space was modeled as a cylinder of helium with 0.1 mm in height and diameter adjusted to the opening of the beam. It was positioned just below the ionization chamber for estimating the intensity of the beam that passed through. The information stored in the phase space is: identification number (ID) of the particle, position and direction, energy, name of the particle (gamma, e-, e+), the name of the volume where the particle was created and the name of the physical process that the created. Such information is obtained through the native function ProcessHits (Figure 5). In user
function SetParticulas, the information is stored in a generic array. After a certain number of events, the data array is saved in a text file. This process repeats until the end of the run. Thus, it avoids excessive memory usage. The number of particles whose information is stored in the phase space should vary from $2 \times 10^6$ to $1.7 \times 10^8$ to obtain a good precision [19,25,31,32].

Figure 4: SSLinaes running.

```cpp
ProcessHits(G4Step* aStep, G4TouchableHistory*) { 
G4Track* Track = aStep->GetTrack(); 
G4double EnergCin = Track->GetKineticEnergy(); 
if (EFAtivo && (EnergCin > 0.)) { 
  Pos = aStep->GetPreStepPoint()->GetPosition(); 
  Dir = aStep->GetPreStepPoint()->GetMomentumDirection(); 
  NomePart = Track->GetDefinition()->GetParticleName(); 
  Energ = EnergCin; 
  if(Track->GetlogicalVolumeAtVertex()) 
    NomeVol = Track->GetlogicalVolumeAtVertex()->GetName(); 
  else NomeVol = "xxx"; 
  if(Track->GetCreatorProcess()) 
    NomeProces = Track->GetCreatorProcess()->GetProcessName(); 
  else NomeProces = "xxx"; 
  (PhaseSpace::GetPointer())->SetParticulas(Pos,Dir,Energ,NomePart,NomeVol,NomeProces); 
} return true; }
```

Figure 5: Code for obtaining the information contained in the phase space.

3.1.4. Parallelization

To make the faster simulations, the simulation was parallelized using G4MPI and OpenMPI (version 1.6.4). It were added some lines of code in the function main to initialize and finalize MPI processes (Figure 6).
It was simulated $4.2 \times 10^8$ histories (primary electrons), using 7 cores of a computer with Intel Core i7 1.73 GHz and 8 GB RAM. The simulation lasted approximately 8 hours. The resulting phase space has approximately $2.1 \times 10^7$ particles saved to a text file of about 1.8 GB.

```
#include "G4MPImanager.hh"
#include "G4MPIsession.hh"
// (...)
CLHEP::Ranlux64Engine randomEngine;
CLHEP::HepRandom::setTheEngine(&randomEngine);
// (...)
G4MPImanager* g4MPI = new G4MPImanager(argc, argv);
G4MPIsession* session = g4MPI->GetMPIsession();
// (...)
session->SessionStart();
// (...)
delete g4MPI;
```

**Figure 6: Lines of code that were added to the main function for parallelization.**

Before starting the simulation, the separation of the jaws of the secondary collimator was defined in such a way as to form a radiation field of $10 \times 10$ cm at 100 cm from the source. This was done to analyze the backscattering of particles to the ionization chamber.

### 3.2. Analysis of Phase Space

The analysis of the phase space was performed in two ways:

1. By energy distribution, analyzing the spectra of photons, electrons and positrons and two-dimensional (2D) distribution of energy of the particles in the color chart;

2. By physical processes, analyzing the correlation between particle types, physical processes of generation and source volumes.

For this, it was developed software, called PSAnalysis, to read the file from the phase space, organize your data and display color graphics. The current version of PSAnalysis returns text files with information that can be analyzed in software of data analysis, with the exception of the 2D distribution of energy. In this work, Microsoft Office Excel® 2007 was used.

The PSAnalysis was developed in Microsoft Visual Studio® 2010 using the type of project Windows Forms Application and the C# programming language [33]. Figure 7 shows the main window of the PSAnalysis.

**3.2.1. Energy Distribution**

Figure 8 shows the code implemented to obtain a text file containing the histogram of energy (number of particles per energy interval) of photons, electrons or positrons. In Figure 9 are
shown the spectra of photons, electrons and positrons. The interval of energy was set at 0.065 MeV. The frequencies of particles in the intervals were normalized by the total number of particles of each type. Note that the spectrum of photons extends above the average energy of the electron beam. This is due to the FWHM of the energy distribution of the beam.

![Figure 7: Main window of the PSAnalysis, software for analysis of phase spaces.](image)

```csharp
// SPECTRA OF PHOTONS (0) ELECTRONS (1) OR POSITRONS (2)
if (ii == 0 || ii == 1 || ii == 2) {
    linha = 0;
    string partic, nome = "gamma";
    if (ii == 1) nome = "e-";
    if (ii == 2) nome = "e+";
    double Inter = 0.065; int NIInter = 100;
    int[] mtFreq = new int[NIInter];
    while (true) {
        strLinhaAtual = sr.ReadLine(); // sr -> StreamReader()
        if (strLinhaAtual != null) {
            strsDadosLinha = strLinhaAtual.Split('t');
            partic = strsDadosLinha[8];
            if (partic == nome) {
                Temp = Convert.ToDouble(strsDadosLinha[7].Replace('.', ','));
                for (int j = 0; j < NIInter; j++)
                    if (Temp > j * Inter && Temp <= (j + 1) * Inter) mtFreq[j]++;
            }
        }
    }
}
```

![Figure 8: Code to obtain a text file with an energy histogram.](image)

To generate the 2D distribution of energy, it was used the implementation of the XY Color Chart, developed by Xu [34]. The XY Color Chart is a rectangle mesh grid in the X-Y plane with colors determined by the values of a data array. The maximum and minimum values are assigned to the first and last colors of the color map. The colors for the remaining values are
determined linearly between this range [34]. The code of implementation of the XY Color Chart, available at: www.drxudotnet.com, were added to PSAnalysis. The data array is obtained from the file with the phase space. Figure 10 shows the code implemented for obtaining this data array.

![Energy spectrum of the electrons (red), positrons (green) and photons (blue).](image)

```csharp
linha = 0; double x, y; float Raio = 38.5f; float Inter = 0.14f;
int i, j, NInter = 550; double[,] mtEnerg = new double[NInter, NInter];
while (true) {
    strLinhaAtual = sr.ReadLine();
    if (strLinhaAtual != null) {
        if (linha >= 3) {
            strsDadosLinha = strLinhaAtual.Split(	);
            x = Convert.ToDouble(strsDadosLinha[1].Replace('.', ',')) + Raio;
            y = Convert.ToDouble(strsDadosLinha[2].Replace('.', ',')) + Raio;
            Temp = Convert.ToDouble(strsDadosLinha[7].Replace('.', ','));
            i = (int)((x - Inter / 2) / Inter + 0.5);
            j = (int)((y - Inter / 2) / Inter + 0.5);
            mtEnerg[i, j] += Temp; } linha++; }
    else break; }
fstream.Close(); sr.Close(); double EnergMax = 0;
for (i = 0; i < NInter; i++)
    for (j = 0; j < NInter; j++) {
        Temp = mtEnerg[i, j];
        if (Temp > EnergMax) EnergMax = Temp; }
for (i = 0; i < NInter; i++)
    for (j = 0; j < NInter; j++) mtEnerg[i, j] = 100 * (mtEnerg[i, j] / EnergMax);
Form forma = new Form();
forma = new FillContour(ref mtEnerg, NInter); forma.Show();
```

![Figure 10: Code for obtaining the data array to generate the 2D distribution of energy.](image)
The method used to obtain the data array was similar to the method used in generating a digital image. It was established the pixel size (mm) with respect to the value of radius (38.5 mm) of the detection volume (phase space). Taking the value of 0.14 mm for the pixel, one gets a "picture" of 550 x 550. The 2D distribution of energy with these dimensions is shown in Figure 11. The percentage is relative to the maximum energy area. The distribution it is reasonably homogeneous.

![Figure 11: 2D distribution of energy of the particles contained in the phase space.](image)

### 3.2.1. Physical Processes

Figure 12 shows the code implemented to obtain a text file containing the number of particles related to the physical processes of generation and volumes where these processes occurred (Table 1). Similarly, one obtains the relation of source volumes and types of particle (Table 2). Table 3 shows the names of the physics processes and volumes represented in the previous tables.

Tables 1 and 2 present a quantitative description of the relative importance of the components of the head of a Linac. It is clearly seen that most of the particles are X-rays produced by bremsstrahlung in the tungsten target (95.5%). The scattered photons originate mainly from the primary collimator, secondary collimator and flattening filter. The secondary collimator is responsible for the production of particles backscattered into the ionization chamber that in this simulation represented 0.5% of total particles.
It is known that the electrons present in the output beam of a Linac are responsible for a significant fraction of the surface dose in the patient [17]. Analyzing the values of Table 2, one notes that electrons and positrons represent less than 1% of the total particles in phase space, and that the major sources of these particles are the ionization chamber and the flattening filter.

```
linha = 0; int NLinhas = 6, NCol = 8; int i, j; string Process, Vol;
int[,] mtFreq = new int[NLinhas, NCol];
string[] strsProcess = new string[NLinhas];
string[] strsVol = new string[NCol];

strsProcess[0] = "eBrem";
strsProcess[1] = "annihil";
strsProcess[4] = "eIoni";
strsVol[0] = "v1";
strsVol[1] = "v2";
strsVol[2] = "v3";
strsVol[3] = "v4";
strsVol[4] = "v5";
strsVol[5] = "v6";
strsVol[6] = "vX";
strsVol[7] = "vY";

while (true) {
    strLinhaAtual = sr.ReadLine();
    if (strLinhaAtual != null) {
        if (linha >= 3) {
            strsDadosLinha = strLinhaAtual.Split(\'\t\');
            Process = strsDadosLinha[10];
            Vol = strsDadosLinha[9];
            for (i = 0; i < NLinhas; i++)
                for (j = 0; j < NCol; j++)
                    if (Process == strsProcess[i] && Vol == strsVol[j]) {
                        mtFreq[i,j]++;
                    break; }
        linha++; }
    } else
        break; }
fstream.Close(); sr.Close();

// (...) Saves the contents of the array mtFreq in a text file (*.txt)
```

Figure 12: Code to obtain the text file containing the number of particles related to the physical processes and the source volumes.

Table 1: Physical Processes x Volumes. The values are in percentage relative to the total number of particles.

<table>
<thead>
<tr>
<th>Physical Processes</th>
<th>v1</th>
<th>v2</th>
<th>v3</th>
<th>v4</th>
<th>v5</th>
<th>v6</th>
<th>vX</th>
<th>vY</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>eBrem</td>
<td>95.5305</td>
<td>0.0252</td>
<td>0.1864</td>
<td>1.0386</td>
<td>0.0537</td>
<td>0.0035</td>
<td>0.1431</td>
<td>0.0050</td>
<td>96.9860</td>
</tr>
<tr>
<td>annihil</td>
<td>0.3125</td>
<td>0.0263</td>
<td>0.2228</td>
<td>1.1431</td>
<td>0.0288</td>
<td>0.0047</td>
<td>0.3378</td>
<td>0.0130</td>
<td>2.0889</td>
</tr>
<tr>
<td>compt</td>
<td>0.0001</td>
<td>0.0028</td>
<td>0.0034</td>
<td>0.0995</td>
<td>0.7277</td>
<td>0.0184</td>
<td>0.0131</td>
<td>0.0005</td>
<td>0.8654</td>
</tr>
<tr>
<td>conv</td>
<td>0.0000</td>
<td>0.0003</td>
<td>0.0009</td>
<td>0.0126</td>
<td>0.0156</td>
<td>0.0005</td>
<td>0.0041</td>
<td>0.0002</td>
<td>0.0341</td>
</tr>
<tr>
<td>eloni</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0001</td>
<td>0.0159</td>
<td>0.0028</td>
<td>0.0002</td>
<td>0.0001</td>
<td>0.0190</td>
</tr>
<tr>
<td>phot</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0019</td>
<td>0.0007</td>
<td>0.0001</td>
<td>0.0000</td>
<td>0.0037</td>
<td>0.0001</td>
<td>0.0066</td>
</tr>
<tr>
<td>TOTAL</td>
<td>95.8431</td>
<td>0.0546</td>
<td>0.4156</td>
<td>2.2945</td>
<td>0.8417</td>
<td>0.0299</td>
<td>0.5020</td>
<td>0.0187</td>
<td>...</td>
</tr>
</tbody>
</table>
Table 2: Volumes x Particles. The values are in percentage relative to the total number of particles.

<table>
<thead>
<tr>
<th>Particles</th>
<th>v1</th>
<th>v2</th>
<th>v3</th>
<th>v4</th>
<th>v5</th>
<th>v6</th>
<th>vX</th>
<th>vY</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>gamma</td>
<td>95.8430</td>
<td>0.0515</td>
<td>0.4093</td>
<td>2.1817</td>
<td>0.0824</td>
<td>0.0082</td>
<td>0.4834</td>
<td>0.0180</td>
<td>99.0776</td>
</tr>
<tr>
<td>e-</td>
<td>0.0000</td>
<td>0.0030</td>
<td>0.0058</td>
<td>0.1063</td>
<td>0.7515</td>
<td>0.0215</td>
<td>0.0165</td>
<td>0.0006</td>
<td>0.9053</td>
</tr>
<tr>
<td>e+</td>
<td>0.0000</td>
<td>0.0001</td>
<td>0.0004</td>
<td>0.0064</td>
<td>0.0078</td>
<td>0.0002</td>
<td>0.0021</td>
<td>0.0001</td>
<td>0.0172</td>
</tr>
<tr>
<td>TOTAL</td>
<td>95.8431</td>
<td>0.0546</td>
<td>0.4156</td>
<td>2.2945</td>
<td>0.8417</td>
<td>0.0299</td>
<td>0.5020</td>
<td>0.0187</td>
<td>...</td>
</tr>
</tbody>
</table>

Table 3: Representation of names of volumes and physical processes.

<table>
<thead>
<tr>
<th>Volumes</th>
<th>Physical Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>Tungsten Target</td>
</tr>
<tr>
<td>v2</td>
<td>Copper Target</td>
</tr>
<tr>
<td>v3</td>
<td>Primary Collimator</td>
</tr>
<tr>
<td>v4</td>
<td>Flattening Filter</td>
</tr>
<tr>
<td>v5</td>
<td>Ionization Chamber</td>
</tr>
<tr>
<td>v6</td>
<td>Mirror</td>
</tr>
<tr>
<td>vX</td>
<td>Jaws X</td>
</tr>
<tr>
<td>vY</td>
<td>Jaws Y</td>
</tr>
<tr>
<td>eBrem</td>
<td>Bremsstrahlung</td>
</tr>
<tr>
<td>annihil</td>
<td>Electron-Positron Annihilation</td>
</tr>
<tr>
<td>compt</td>
<td>Compton Scattering</td>
</tr>
<tr>
<td>conv</td>
<td>Pair Production</td>
</tr>
<tr>
<td>eloni</td>
<td>Ionization</td>
</tr>
<tr>
<td>phot</td>
<td>Photoelectric Effect</td>
</tr>
</tbody>
</table>

4. CONCLUSIONS

The Linac is the most commonly used equipment in radiotherapy for treatment of patients. The modeling of its structure influences the degree of accuracy of dose assessment. This work aimed to create a computer model of the head of a 6 MeV Linac using the MC code Geant4. The methodology used to analyze the phase space helps to understand the characteristics of beam of a Linac and with this one can improve the accuracy of clinical dosimetry, providing more realistic data. However, it is also necessary to implement methods of error analysis to make the analysis more reliable. This is what is intended to do for publication of the next jobs.

ACKNOWLEDGMENTS

The authors would like to thank CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) for financing the research.
REFERENCES


