

Technical Note

The Software with a Graphical User Interface for GAMOS: Basic Training and an Educational Tool for Medical Physicists

Recep KANDEMİR ^{1,ABCDEF,*}, İsmail Özsoykal ^{1,D}, Kadir AKGÜNGÖR ^{2,EF}

¹Department of Medical Physics, Institute of Health Sciences, Dokuz Eylül University, Turkey

²Department of Physics, Faculty of Science, Dokuz Eylül University, Turkey

*Corresponding author: Recep Kandemir; recepkandemir@medikalfizikci.com

(received 7 August 2022; revised 10 January 2023; accepted 17 January 2023)

Abstract

Introduction: It is necessary to have special experience to perform the Monte Carlo calculation, commonly used in medical physics and accepted as the gold standard. In this study, we developed software to teach basic steps to medical physicists who were inexperienced in the medical linear accelerator Monte Carlo simulation.

Material and methods: For the design interface, a software called GamosLinacGUI was developed using Gnome Builder, Python, and GTK. The user, who wants to learn the basics of GAMOS and simulate a linear accelerator, can enter the values in the software, select some options and quickly create geometry and physics files.

Results: For proof that the software generates the correct inputs for GAMOS simulation in the same conditions for the measurements and calculations. Required files for GAMOS have been created and tested and run the simulation accordingly. This software was tested with Centos Linux.

Conclusions: GamosLinacGUI has been successfully developed, which creates the geometry and physics files required for the simulation with GAMOS as a training and learning tool.

Keywords: educational software; graphical user interface; Monte Carlo simulation; GAMOS.

Introduction

External beam therapy, or radiotherapy, is one of the common methods used for cancer treatment. The main principle of radiotherapy is to provide the maximum dose to the tumor volume while, on the other hand, keeping it under certain threshold levels for healthy tissues and organs at risk. This principle helps avoid normal tissue complications and take the tumor under control.¹

It is not possible to make in-vivo measurements of patient dose during radiotherapy. However, it is possible to make a series of complex calculations before the treatment and obtain the dose distributions within the body of the patient. This goal is achieved by a set of computer-based algorithms that build up a treatment planning system (TPS).²

The Monte Carlo method is commonly used in many fields related to mathematics and physics. It simply aims to estimate the values of unknown quantities by following the principles of inferential statistics. In medical physics, particularly, the Monte Carlo method is used to determine the probability distributions of radiation-matter interactions based on a number of variables (particle type, particle energy, physical properties of the target

material, etc.). This is followed by an accurate prediction of different interactions, which occur in different frequencies and yield unique outcomes. Finally, these estimates are converted into absorbed dose distributions within the medium of interest.³

Advances in computer technology have enabled complex equations such as electron and photon transport to be solved in shorter periods and with higher accuracy. Moreover, computers provided with the necessary technical equipment for executing these Monte Carlo based calculations can be accessed much more easily today, compared to the early periods corresponding to the middle of the 1900s. As a result, the number of studies on Monte Carlo techniques has increased day by day, and the use of this method in medical physics has become widespread.⁴⁻⁶

From the 1970s on, there have been several Monte Carlo based codes developed focusing on radiation dosimetry in medical applications, especially radiotherapy. Mainly known among these are EGS, MCNP, PENELOPE, and GEANT4.⁷⁻¹⁰ Each code requires a certain language of programming to be known by the user. For example, GEANT4 code is based on C++ as a programming language. As a toolkit, the GEANT4 is installed with several different tools. These tools are used to construct several files with their headers, which in turn need to be

compiled with C++ code to introduce the simulation configuration to the kernel and eventually run it. The files, such as primary generator, detector construction, run action, event action, etc. are constructed in GEANT4 to define the physics and geometry of the model of the desired simulation. In this way, the user can create complex geometries and obtain a high-precision output of very detailed tracking of particles and interactions. However, as mentioned, for medical physicists who are not experienced with C++, it would be very challenging to take these steps.

A code interface called GAMOS, GEANT4-based Architecture for Medical-Oriented Simulations, has been developed to eliminate this limitation between the user and GEANT4.¹¹ With the GAMOS the user can create the files with extensions "geom" for geometry and "in" for physics inputs in ASCII format without having knowledge of the C++ programming language. In this way, the user gets the opportunity to run simulations in a simplified manner. Besides that, GAMOS has readily available libraries (e.g., scoring, filtering, classifier, verbosity) and tutorials that provide more detailed information on calculations and data analysis regarding applications specific to medical physics. However, the medical physicist must remain loyal to a specific syntax and algorithm framework required by GAMOS when creating the geometry and physics files. This could be treated as a challenging task for the medical physicist who is not familiar with constructing algorithms.

This study introduces a new graphical user interface (GUI), which is designed to eliminate all the limitations coming up from the challenges that a user cannot overcome while building his or her own simple geometry and physics files in GAMOS. This GUI offers the user to construct and edit geometry and physics files required by GAMOS for the simulation of water phantom setup, which has a very important role in the acceptance and quality control tests for the medical linear accelerators (linac). It can be used as a teaching, learning, and training tool for GAMOS and a simulation setup for inexperienced medical physicists. Input parameters such as particle type, beam energy, radiation field size, detector configuration, the choice of phase space files, and GAMOS output format will be practically defined by the user and registered to the geometry and physics files by the GUI. Also, further use of Monte Carlo based simulation programs in the field of medical physics is to be promoted.

Material and Methods

Monte Carlo Model

GAMOS (v6.1.0) simulation code has been used to design a model of TrueBeam™ linear accelerator operating at different beam energies as a sample to test the software. This model excludes certain parts that are located in the linear accelerator head, such as the target, ion chamber, flattening filter, and primary collimator. This is mainly because the geometric data regarding these related parts are not shared by the manufacturer. Alternatively, even if the data had been available, it would not be practical to model the whole system geometry, since there would be a considerable amount of particles being lost along the way to the linac surface (e.g.: absorbed by the flattening filter) and leading to weak statistical results. In other words, it is not efficient by means of computational time to produce electrons and direct them to a target material to yield a spectrum of high-energy photons. Instead, it can be a more rational approach to use a data-set having accurate momentum, energy and direction information of a beam of particles that have emerged from the target and penetrated the flattening filter. In this study, it was chosen Varian's TrueBeam™ phase space file as a sample that is available on IAEA's website in order to test the features of the software.¹² This phase space file is defined for a plane of interest located at $z=26.7$ cm below the source, just above the secondary collimators. Secondary collimators (X and Y jaws) and the base plate have been included in the model as defined by the geometric data (z-axis position, dimension, material composition, etc.) provided in the same package. These volume structures are illustrated in **Figure 1** together with the water phantom and various detector geometries, and all of which could be designed and edited by the program user.

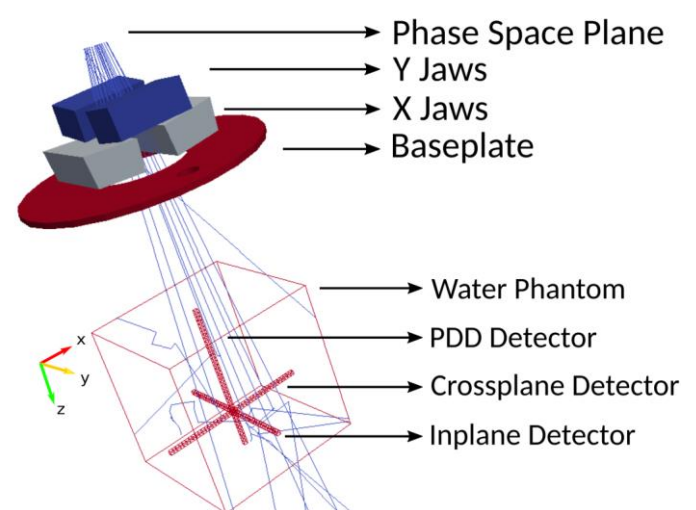


Figure 1. A sample Monte Carlo simulation geometry in GAMOS.

Graphical User Interface

The software named GamosLinacGUI was developed by using Python (v.3.6) high-level programming language, and GTK is a multi-platform toolkit for creating graphical user interfaces.^{13,14} The software graphical elements and items have been designed with the help of Gnome Builder IDE (v3.34).¹⁵ Basically, the code structure of GamosLinacGUI consists of a combination of two files with the extensions ".ui" and ".py". Moreover, all the components necessary for packaging the program are ready for the user.

GAMOS simulation software asks the user to provide at least one geometry file with ".geom" extension and one physics file with ".in" extension; both are written in ASCII format. Geometry files are embedded in the physics file, which is composed as to be read line by line and interpreted by the kernel to initialize the required configuration. In the geometry file, the user is supposed to define the media of interest by material types, dimensions, and positions as input parameters. Parametrization of the input variables enables their definition externally. GamosLinacGUI is designed on the basis of GAMOS syntax as a sample shown in **Figure 2**. Numerical inputs which are defined by the user are assigned as parameters (Lines 1–8), which are then addressed to the related lines of code (Lines 9–10) to construct and place the geometric volume structures.

The Features of GamosLinacGUI

GamosLinacGUI offers to the user the opportunity to create necessary geometry and physics files equipped with various user actions. Upon the definition of dimensions, positions, and materials of volume structures, GamosLinacGUI creates related geometry files and embeds the predefined input parameters in these files. The user can define the geometry and material properties of the outermost volume structure, named as a world, along with other volume structures, such as the container, which includes the jaws and baseplate. Note that the aperture of X and Y jaws are automatically calculated and recorded in the related geometry file as soon as the user defines the desired field size at SSD = 100 cm. Besides, the visibility and color properties of every single volume structure can be defined visually before obtaining the ".wrl" file, which can be visualized with an external VRML, the Virtual Reality Modeling Language, viewer shown as a sample in **Figure 3**. Moreover, for the detector geometry constructed within a water phantom, the user has the opportunity to select various detector volumes named as in-plane, cross-plane, 2D, and PDD (percentage depth dose) detectors, each of which can be defined as an array with the desired container and voxel size parameters (**Figure 4**). With all these capabilities, this software is a teaching and learning tool for medical physicists who want to use GAMOS.

Gamos Syntax

```
Line 1 :P halfSizeXOfMedium 10*m
Line 2 :P halfSizeYOfMedium 10*m
Line 3 :P halfSizeZOfMedium 10*m
Line 4 :PS materialOfMedium G4_AIR
Line 5 :P xPositionOfMedium 0*cm
Line 6 :P yPositionOfMedium 0*cm
Line 7 :P zPositionOfMedium 0*cm
Line 8 :ROTM R0 0*deg 0*deg 0*deg
Line 9 :VOLUME Medium BOX $halfSizeXOfWorld $halfSizeYOfWorld
$halfSizeZOfWorld $materialOfWorld
Line 10 :PLACE Medium 1 MotherVolume R0 $xPositionOfMedium
$yPositionOfMedium $zPositionOfMedium
```

Figure 2. As a sample, the syntax of basic volume definition in sample GAMOS geometry file with parameters.

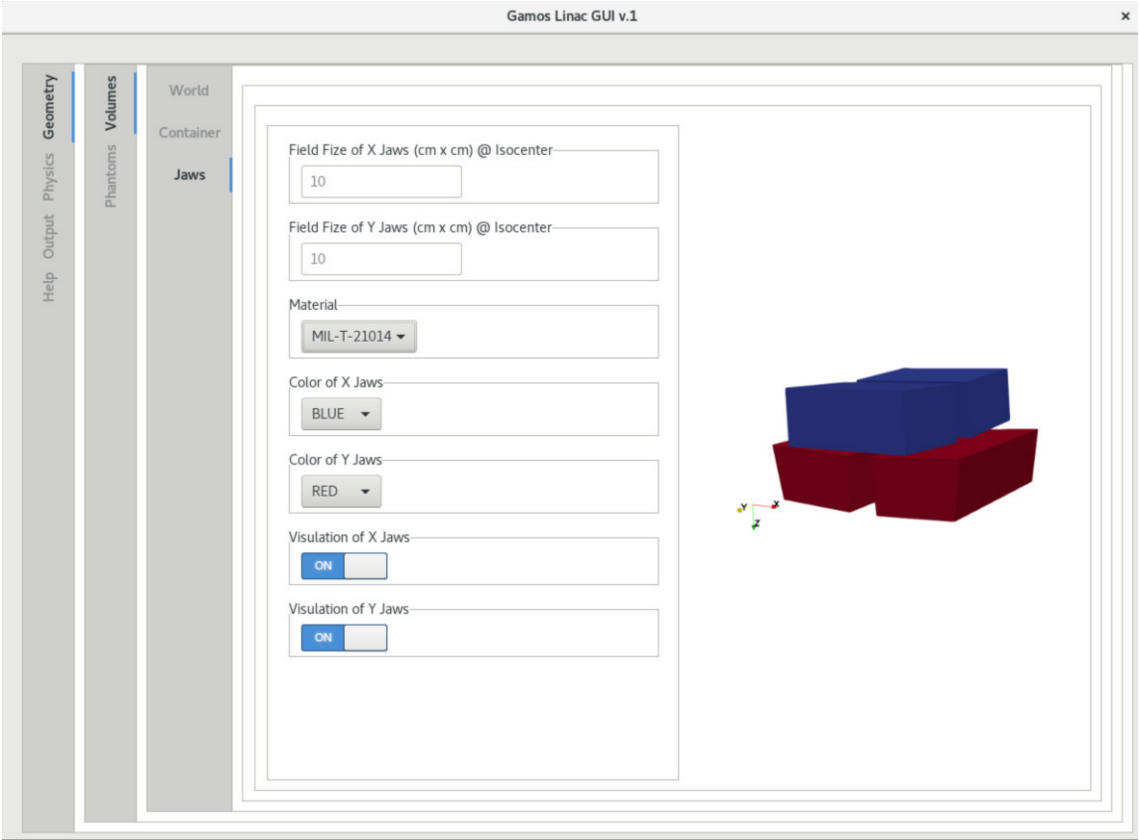


Figure 3. GamosLinacGUI interface determines the field size at SSD = 100 cm.

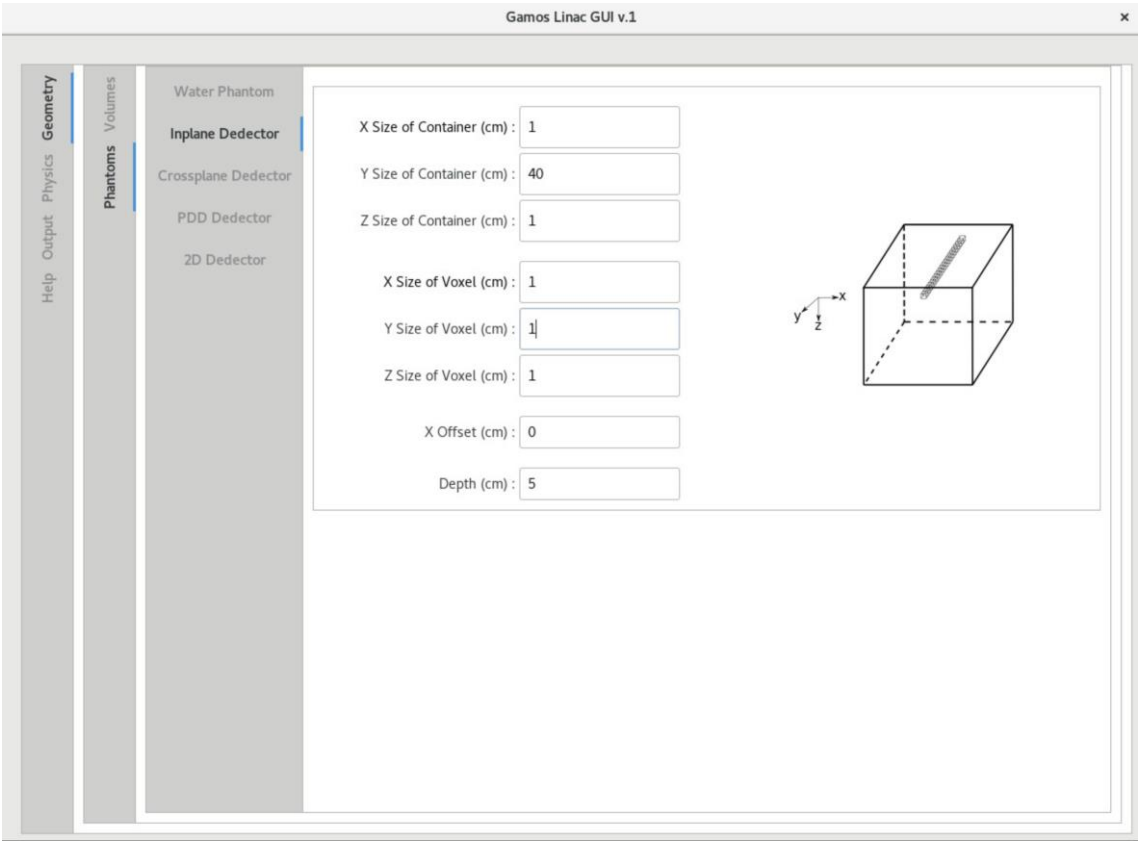


Figure 4. GamosLinacGUI interface determines voxel sizes of detectors.

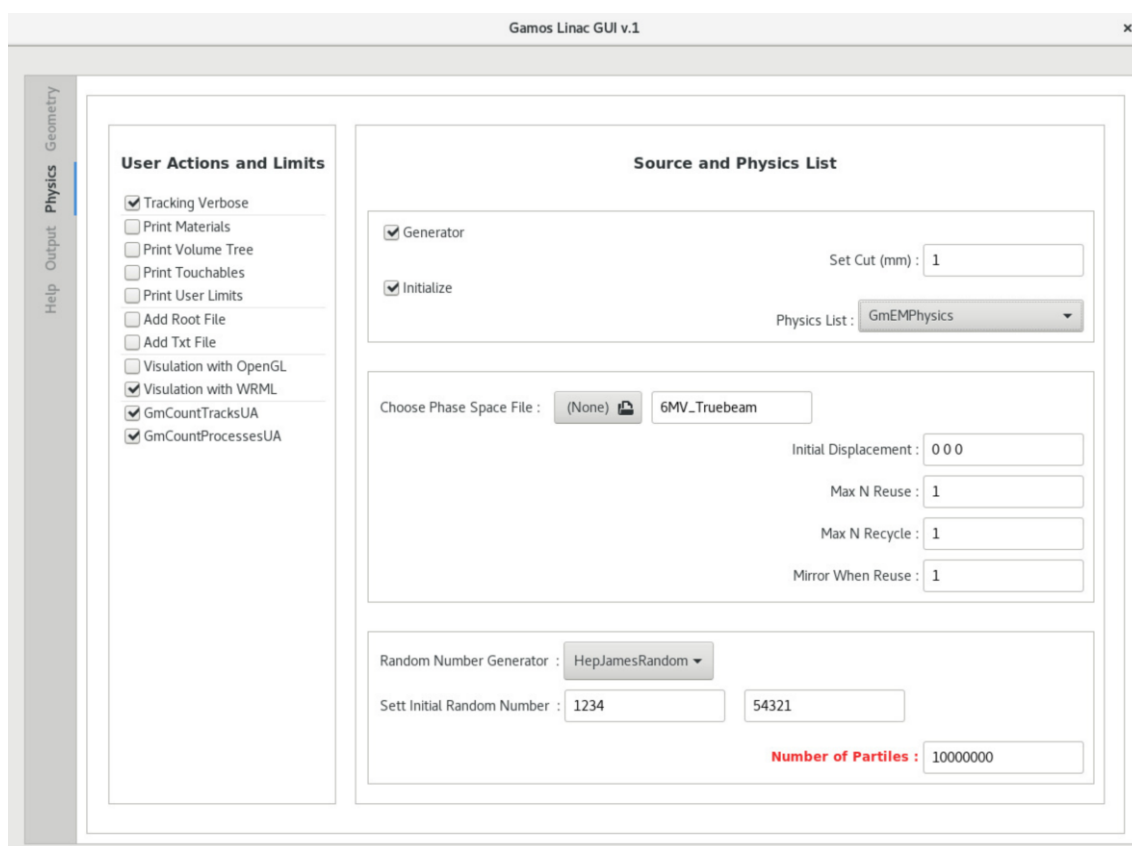


Figure 5. Definition and selection of particle generation and detection mechanisms, physics list, user limits, and user actions with GamosLinacGUI.

Table 1. The geometry, physics, and other files created by GamosLinacGUI for linear accelerator Monte Carlo simulation.

Input Type	File Name	Content of File
Geometry	world.geom	Dimensions and material information of world volume.
	material_param.geom	Material definitions are created in this file.
	linachead_param.geom	Dimensions and material information of container volume.
	jaw_param.geom	The file that contains trigonometric calculations for jaw module.
	rotation_param.geom	Rotation matrixes are defined in this file.
	phantoms_param.geom	In-plane, Cross-plane, and PDD detector array voxels are defined in this file.
	vis_param.geom	The colors and visualization options for volumes for wrml and opengl are defined in this file.
Physics	linac.geom	The main geometry file which includes all other geometry and parameter files.
	scorer.in	This is the file that contains the physics commands written for the detectors.
Others	linac.in	This is the main physics file where all physics commands are written and also required file for GAMOS to start simulation.
	visOpenGL.in	The file that contains the commands required to represent OpenGL viewer.
	visVRML.in	This is the file that writes the commands for creating the VRML file.

To ensure the program runs correctly, two different physics files are created via GamosLinacGUI as sample runs. One of them, named linac.in, is equipped so that the user can define the phase space file path and its features, set-cut value, physics list selection (GmEMPhysics and GmEMExtendedPhysics), type of random number generator and the number of particles, selection of user action, and other functionalities. The other one, named scorer.in, is designed for the user to select GAMOS scorer classes of detectors which are created to collect data in the water phantom. Furthermore, there are selections to visualize the

geometry with OpenGL or VRML format and some of the user actions offered by GAMOS (**Figure 5**).

After following all the steps suggested by the software, the user can export geometry and physics files to be used in GAMOS simulation program into the folder where the software is executed. The name of the created geometry and physics files by the software are shown in **Table 1**.

As the final step, one needs to open an operating system command prompt in the folder of geometry and physics files which are created by software, and run the command of GAMOS to start the Monte Carlo simulation.

Results

GamosLinacGUI software developed using Python and GTK has been tested on a GAMOS-installed Centos 8 Linux computer.

GamosLinacGUI is based on a model of Varian's TrueBeamTM linear accelerator. In this part of the study, a dosimetric test has been conducted to validate the linear accelerator model and GamosLinacGUI. The standard quality control procedure of a linear accelerator was designed as an experimental setup for the test. The water phantom was irradiated by a 6 MV photon beam, with a field size of $10 \times 10 \text{ cm}^2$ at SSD = 100 cm. In-plane and cross-plane dose measurements have been made at a depth of 10 cm, while measurements for PDD were recorded along the central axis. All measurements are taken by a 0.125 cm^3 Semiflex (PTW 31010, PTW, Freiburg, Germany) ion chamber. On the other hand, a similar setup was modeled via GamosLinacGUI. Detector arrays were constructed as $5 \times 5 \times 5 \text{ mm}^3$ voxels for in-plane, cross-plane and PDD calculations. Different phase-space files for 6 MV photon beams were chosen with different seed numbers to diminish the statistical error. 1×10^9 number of particles was used in this simulation run. **Figure 6** shows the calculated and measured dose profiles at a depth of 10 cm and PDD for the field size of $10 \times 10 \text{ cm}^2$ defined at SSD = 100 cm. To determine percentage differences between calculated and measured cross-plane, in-plane, and PDD profiles, the Trapezoidal Rule approach was utilized. The percent differences for crossplane, inplane, and PDD were determined to be 1.53%, 0.79%, and 0.74%, respectively, when the Monte Carlo calculations and measurement data were compared. Cross-plane, in-plane, and PDD profiles measured by Semiflex ion chamber in the 3D water phantom were agreed well with profiles calculated by GAMOS.

Discussion

In medical physics, especially in the field of radiation therapy, the absorbed dose to the patient has vital importance. Research have been focused extensively on computational methods for estimating the absorbed dose in radiotherapy treatments. Many software programs are available that operate based on the Monte Carlo technique. However, the use of these programs requires a solid grasp of complex programming languages. Even for experienced users, it may cost a valuable period of time to construct the required volume structures (e.g.: jaw modules, detector arrays) and define physics-related parameters (e.g.: set-cut, user actions, user limits). The complex nature of these programs could therefore be a significant limitation for some researchers who are not familiar with programming.

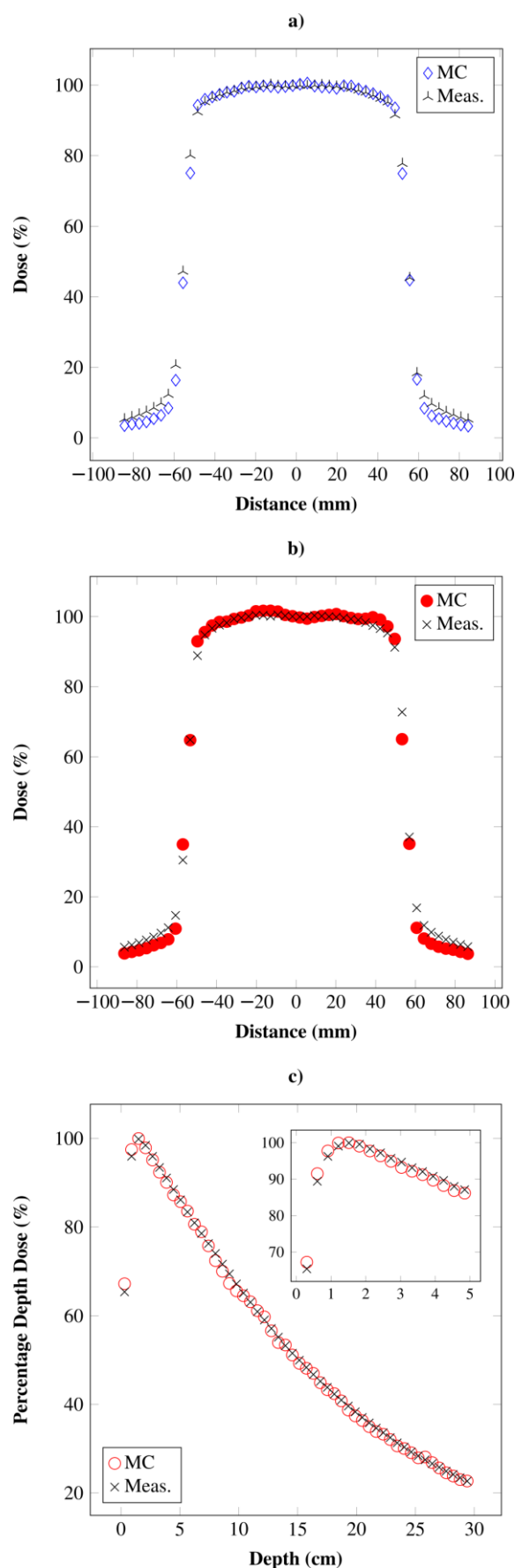


Figure 6. The comparison of measured cross-plane (a) and in-plane profiles (b) and PDD (c) with Monte Carlo calculation.

In this study, GamosLinacGUI has been introduced as an interface between the user and the GAMOS simulation software to teach and learn the basics of the simulation. This goal is achieved using generating required geometry and physics files with the user specifications via a user-friendly graphical interface.

The sample design of GamosLinacGUI is based on a model of the Varian TrueBeam™ linear accelerator with some simplifications made over the geometry of the head component. The phase space files, which are provided by the manufacturer, have been used to cover these simplifications accurately. Validation tests have shown good agreement between the measured and computed values of in-plane and cross-plane dose profiles and PDD in **Figure 6**. These results indicate that both the GAMOS model of the linear accelerator and the GamosLinacGUI can be used as reasonable tools for the characterization of a simulation setup.

GamosLinacGUI provides medical physicists with selective options to rapidly construct different water phantom setup geometries. These include the selection of different phase space files which are defined above X and Y jaws and correspond to different beam energies. On the other side, the user can define radiation field size together with the dimensions of the water phantom and detector array structures. Moreover, the user can build the physics file in such a way that he/she can obtain and print a number of simulation outputs.

GamosLinacGUI can be handled as educational material for medical physicists who want to get familiar with simulation with GAMOS. The user has the chance to create and compare different geometry and physics files, observe the effects of his/her definitions/selections on these files, and get used to the syntax of the GAMOS files.

There are some limitations regarding the capabilities of GamosLinacGUI. First, it is designed to include the jaw geometry module for only Varian TrueBeam™ linear accelerator. However, since this software is an educational and training tool, the user can also model another manufacturer's jaw module or other parts like a multi-leaf collimator by modifying the codes produced by the software.

Conclusion

An educational and training software with a graphical user interface called GamosLinacGUI is successfully developed for medical physicists who want to model medical linear accelerators by using GAMOS Monte Carlo simulation framework and have no experience in simulation and programming languages. The geometry and physics files were created for the Varian TrueBeam™ machine, and the measurements made in the water phantom were tested by comparing the simulation results. This software will speed up the process of learning the simulation algorithm and syntax of GAMOS by the users. The software will be free of charge who want to use it.

Acknowledgments

We are thankful to The Department of Radiation Oncology, Dokuz Eylul University, for giving us a chance to use The DEURADON Laboratory, Metropolis Beowulf Workstation throughout this study.

References

1. Baskar R, Lee KA, Yeo R, Yeoh K. Cancer and radiation therapy: current advances and future directions. *Int J Med Sci*. 2012; 9(3):193-199. <https://doi.org/10.7150/ijms.3635>
2. Hussain A, Muhammad W. Treatment Planning in Radiation Therapy. In: Maqbool M., eds. *An Introduction to Medical Physics*. Germany: Springer, Inc; 2017: 63-129. <https://doi.org/10.1007/978-3-319-61540-0>
3. Oelfke U, Scholz C. Dose Calculation Algorithms. In: Schlegel W., Bortfeld T., Grosu AL., eds. *New Technologies in Radiation Oncology*. Berlin: Springer, Inc; 2006:187-196. https://doi.org/10.1007/3-540-29999-8_15
4. Andreo P. Monte Carlo techniques in medical radiation physics. *Phys Med Biol*. 1991; 36(7):861-920. <https://doi.org/10.1088/0031-9155/36/7/001>
5. Konefał A, Bakoniak M, Orlef A, Maniakowski Z, Szewczuk M. Energy Spectra in water for the 6 MV X-ray therapeutic beam generated by CLINAC-2300 LINAC. *Radiation Measurements*. 2015;72:12-22. <https://doi.org/10.1016/j.radmeas.2014.11.008>
6. Adam DP, Liu T, Caracappa PF, Bednarz BP, Xu XG. New capabilities of the Monte Carlo dose engine ARCHER-RT: Clinical validation of the Varian TrueBeam machine for VMAT external beam radiotherapy. *Medical Physics*. 2020;47(6):2537-2549. <https://doi.org/10.1002/mp.14143>
7. High Energy Accelerator Research Organisation. Electron Gamma Shower (EGS) Web Page <http://rcwww.kek.jp/research/egs/>. Accessed December 16, 2019.
8. Los Alamos National Laboratory. A General Monte Carlo N-Particle (MCNP) Transport Code. <https://mcnp.lanl.gov/>. Accessed December 16, 2019.

9. Nuclear Energy Agency. PENELOPE2014, A Code System for Monte-Carlo Simulation of Electron and Photon Transport. <https://www.oecd-neo.org/tools/abstract/detail/nea-1525>. Accessed December 16, 2019.
10. Agostinelli S, Allison J, Amako K, et al. GEANT4--a simulation toolkit. Nucl Instrum Meth A. 2003;506(3):250-303. [https://doi.org/10.1016/S0168-9002\(03\)01368-8](https://doi.org/10.1016/S0168-9002(03)01368-8)
11. Arce P, Lagares JJ, Harkness L, et al. Gamos: A framework to do Geant4 simulations in different physics fields with an user-friendly interface. Nucl Instrum Meth A. 2014;735:304-313. <https://doi.org/10.1016/j.nima.2013.09.036>
12. IAEA's Phase Space Website. <https://www-nds.iaea.org/phsp/phsp.htmlx>. Accessed November 15,2019.
13. Python High Level Programming Language. <https://www.python.org>. Accessed November 15, 2019.
14. The GTK Project. <https://www.gtk.org>. Accessed November 15, 2019.
15. Builder. <https://wiki.gnome.org/Apps/Builder>. Accessed November 15, 2019.