Winter 2018

Modelling of a Megavoltage Linear Accelerator Using EGSnrc Software to Perform Monte Carlo Simulations

Daniel McKinney
danielmckinney1980@gmail.com

MODELLING OF A MEGAVOLTAGE LINEAR ACCELERATOR USING EGSnrc SOFTWARE TO PERFORM MONTE CARLO SIMULATIONS

by

Daniel McKinney

B.S., Utah Valley University, 2007

A Research Paper
Submitted in Partial Fulfillment of the Requirements for the
Master of Science

Department of Physics
in the Graduate School
Southern Illinois University Carbondale
December 2018

RESEARCH PAPER APPROVAL

MODELLING OF A MEGAVOLTAGE LINEAR ACCELERATOR USING EGSnrc SOFTWARE TO PERFORM MONTE CARLO SIMULATIONS

by

Daniel McKinney

A Research Paper Submitted in Partial

Fulfillment of the Requirements

for the Degree of

Master of Science

in the field of Physics

Approved by:

Saikat Talapatra Ph.D., Chair

Poopalasingam Sivakumar Ph.D.

Thushari Jayasekera Ph. D.

Graduate School
Southern Illinois University Carbondale
Nov 9, 2018

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In modern cancer treatment, radiation therapy is often used for tumor ablation in patients that are expected respond well to this form of therapy. In today's radiation oncology clinics, linear accelerators produce beams of photons or electrons with energies as high as 25 MeV (McGinley, 2002, p. 6). When radiation is used for the ablation of cancerous tissue it must also interact with healthy tissue as it passes through the body; therefore, it is essential for medical physicists to accurately model a beam of radiation to minimize absorbed dose to healthy tissue during treatment planning.

There are many treatment planning systems (TPSs) in use today that can make 3D dose calculations quick and convenient. TPSs are commercially available software systems that allow the calculation of absorbed dose, delivered by a linear accelerator, on patient specific CT data sets. Popular TPSs in use today include; Eclipse (Eclipse^{TM} Treatment Planning, 2018), Pinnacle (Radiation Oncology, 2018), and Monaco® (Elekta, 2018). TPSs aim to give medical staff the most accurate dose delivered to the patient. "Monte Carlo techniques are regarded as the dose calculation technique which offers the best accuracy; nevertheless, its use in clinical practice is still prevented by long calculation times" (Francescon, 2000, p. 1579). Since the time of this quote some advancement has been made, allowing Monaco to become the first and only commercially available TPS to offer photon calculations using Monte Carlo techniques.

The major goal of this research project was to gain a better understanding of how Monte Carlo dose calculations are performed. The project also aims to create a basic model of a commercially available linear accelerator to gain understanding of the factors involved in safely directing beams of radiation while treating patients. A major hurdle of modeling radiation beams is learning the computer programming language, specific to the Monte Carlo software kit used to perform these simulations, and using it to create objects in the beam path that affect the shape of the beam.

The Monte Carlo software package used for this project was first developed at the SLAC (Stanford Linear Accelerator Center) sometime during the late 80s and into the early 90s. It was named EGS (Electron Gamma Shower) in its earlier versions, but after many years of collaboration with the Nuclear Regulatory Commission (NRC) the current version used here is called EGSnrc. "The EGS (Electron–Gamma–Shower) system of computer codes is a general-purpose package for the Monte Carlo simulation of the coupled transport of electrons and photons in an arbitrary geometry for particles with energies above a few keV up to several hundreds of GeV" (Kawrakow, 2017, p.1-4, 15). This project will further restrict the particle energies to a range on the order of 10 MeV currently used in radiotherapy clinics.

CHAPTER 2

BEAM CREATION IN LINEAR ACCELERATORS

The creation of beam using any linear accelerator (linac) starts at one end of an accelerator tube where low energy electrons are injected into the tube. High frequency microwaves are simultaneously injected into the tube via the waveguide. The inside of the tube is kept near vacuum and consists of copper sections (the walls are near 0 electric potential) designed to create a standing, or traveling, wave depending on the design. The injected electrons "gain energy from the sinusoidal electric field by an acceleration process analogous to that of a surf rider" (Kahn, 2003, p. 42-43). While the complete acceleration process detail is outside the scope of this project, a more detailed description can be found at the above reference. For this purpose, it is enough to say that the result is a pencil beam of electrons, with energy on the order of 6-20 MeV, created incident on the beam path which must be modelled for Monte Carlo simulation.

At this time, the person operating the linac must determine what form of radiation beam is desired for treatment. If the desired outcome is a photon beam, a slab of material is electronically inserted in the beam path (usually a combination of W and/or Cu) forcing the high-speed electrons to interact with the nuclei via a process known as bremsstrahlung resulting in the creation of photons. These slabs of material are known as "transmission-type targets...in which the electrons bombard the target from one side and the x-ray beam is obtained on the other side" (Kahn, 2003, p.33). If an electron beam is required, the target simply needs to be removed to let the electron beam pass unobstructed.

Both photon and electron beams must pass through a primary collimator. After the electron pencil-beam interacts with the target, the direction of the photons is uncertain. The purpose of the primary collimator is to restrict unwanted radiation from leaving the treatment head in directions other than that of the primary beam. After exiting the primary collimator, the beam is then incident on the carousel. The carousel contains a flattening filter specific to each photon energy available, as well as a scattering filter to be used for electron beams. There is also an open slot available in the carousel for a photon beam delivered in flattening filter free (FFF) mode. Following the carousel, the beam passes through an ionization chamber that monitors the symmetry of the beam as well as the radiation output from the treatment head. The beam then passes to the secondary collimators, also known as Jaws. There are two sets of jaws perpendicular to the primary beam axis known as the X and Y jaws. Unlike the primary collimator, these jaws are adjustable and allow the beam to be further collimated into square or

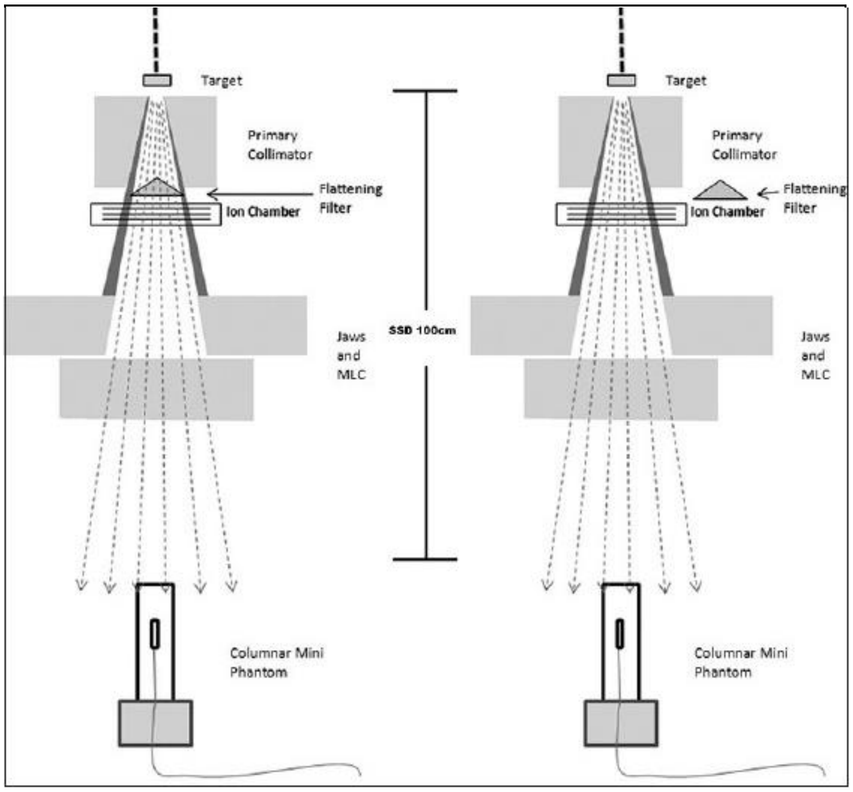


Figure 1: Basic photon beam design for flattened and FFF beams (Ashokkumar, 2014). Left: flattening filter is inserted attenuate central parts of the beam and equalize fluence across the whole beam. Right: Flattening filter removed making beam similar to an isotropic point source.

rectangular fields before they exit the treatment head. A basic representation of the photon beam path, flattened and FFF, can be seen in figure 1.

Patient specific beam shaping happens after the secondary collimators in all available beam types. Photon beams are shaped using multi-leaf collimators (MLCs) that vary in dimension depending on the system design. MLCs can be described as sets of 5 mm. or 1 cm. thick leafs that can be extended into the beam after the jaws to create more complex beam shapes. Alternatively, compensators are used to block unwanted parts of photon beams from reaching the patient. When delivering electron beams, electron cones are attached to the treatment head. Patient specific blocks of Cerrobend are poured in the desired shape and attached to the end of the electron cone. None of these types of patient specific objects will be modelled in this project and will be considered for modelling in future projects. A list of objects in the primary beam path of photons and electrons can be seen in Table 1 below.

Photon Beam	Electron Beam
x-ray transmission target	no target
Primary Collimator	Primary Collimator
Flattening Filter	Scattering Foil
Monitor Ionization Chamber	Monitor Ionization Chamber
Secondary Collimator	Secondary Collimator
Multi-Leaf Collimator	Electron Cone
Physical Wedge or Compensator	Electron Block (cerrobend: bizmuth, lead tin, and cadmium)

Table 1: Summary of objects in the beam path for photon beams (left) and electron beams (right).

CHAPTER 3

MODELLING OF MC SIMULATIONS IN EGSNRC AND BEAMNRC

EGSnrc software offers a couple of different options for creating beam geometries when using the EGSnrc framework. The first option is to create a beam simulation using an egs++ application, and the second is to use the software BEAMnrc. The egs++ language is specific to this program and essentially gives the user access to a "c++ class library that was built to allow for c++ applications to directly interface with the MORTRAN3 EGSnrc core" (Townson, 2018, p. 11). The class library described in Report PIRS-898 give the user the ability to create virtually any possible geometry if they use the objects in the library creatively. Every egs++ application will take 6 different kinds of input definitions; geometry definition, media definition, particle source definition, ausgab definition, run control definition, and Monte Carlo transport parameters. Within each of these definitions, there are many available classes that can be used to simulate the interaction of sources with matter (Rogers, 2018) . Each of these definitions is combined into an egs input file that will be read in when the application is run. The required variables for each run are a simulation geometry, a simulation source, ncase (the number of particles histories to be run), a density correction file for each medium used, and the Monte Carlo transport parameters, and an ausgab (output) object (Kawrakow, 2018). Coding for egs++ applications can be hard to learn and tedious for the user, but can also allow the user the freedom to create complex geometries. A simple coding example from Report PIRS-898 will be included in the appendix to help bring clarity to the reader on egs++ coding.

BEAMnrc is a software kit that is included in the current EGSnrc installation package. It is a little easier to use than egs++ applications to model linear accelerators. It is a Monte Carlo simulation system "for modelling radiotherapy sources which was developed as part of the

OMEGA project to develop 3-D treatment planning for radiotherapy (with the University of Wisconsin)" (Rogers, 2018, p. 2). Some of the usability that is gained when creating egs input files with BEAMnrc comes from the option of using a GUI. All the same variables must be defined for the EGS simulation to run; however, they can be selected from a list of options presented to the user inside the GUI. Three main files are needed to run a simulation using the BEAMnrc GUI; and accelerator, a PEGS4 file, and an egs input file (*.egsinp).

The accelerator file is essentially a list of objects that the user intends to define to put in the beam path. Once a new accelerator is created the user is asked to select from a list of component modules (CMs) and save them. The CMs will be defined later in the process and will make up the simulation geometry. The accelerator is saved to a file with the file extension *.module. It is helpful to think of this CM list as objects the beam will interact with in order along the primary beam path. New users could simply refer to Table 1 above and decide which CMs would easily create each of those objects in order (e.g. a target; a primary collimator; flattening filter; ionization chamber; Jaws; MLCs). The PEGS4 file is located somewhere inside the EGSnrc file structure called the HEN HOUSE, and it is the file that defines the "density effect corrections in ICRU report 37" (Rogers, 2018, p. 248). After the accelerator and the PEGS4 file are defined, the user simply needs to select each CM from the list created by the software and input the dimensions that will create the object they want. It is highly recommended, by this student, that the BEAMnrc User's Manual (Rogers, 2018) is used to understand what each CM does and its variables are prior to use.

CHAPTER 4

EGS++ CODING FOR 10MV PHOTON BEAM

An example of the egs++ coding that I created for a 10MV photon beam will be presented here. As described in the previous chapter, the objects that needed to be defined are; particle source, geometry, Ausgab object, MC transport parameters, media, and run control. The particle source used in the coding assumes that electron pencil-beam exiting the waveguide is a parallel beam of electrons incident on the target. Within the EGSnrc c++ class library, there is a specific library designed for this purpose named egs_parallel_beam. Once called, this library expects certain inputs that must be defined, as seen in figure 2 below:

```
:start source definition:
    :start source:
        library = egs_parallel_beam
        name = my_source
        :start shape:
            type = cylinder
            radius   = 0.15
            height   = 2
            axis     = 0 0 1
            midpoint = 0
        :stop shape:
        direction = 0 0 1
        charge = -1
        :start spectrum:
            type = monoenergetic
            energy = 10
        :stop spectrum:
    :stop source:

    simulation source = my_source

:stop source definition:
```

Figure 2: Coding used in this project to define a cylindrical 10MeV particle source.

From the shape definition, you can see that the source is a 2 cm long cylinder with a 0.15 cm radius centered on the origin. The direction variable defines a vector of motion along the +z axis. A charge of -1 indicates that the particles used are electrons. And the spectrum definition indicates that all electrons have an initial energy of 10 MeV traveling along the z-axis at the beginning of the simulation.

An ausgab object defines what output the is desired following the simulation. The available ausgab objects are egs_track_scoring, egs_dose_scoring, and beam_dose_scoring (Kawrakow, 2018). The second two objects define regions of dose that scored within the simulation. Simulations done for this project used the egs_track_scoring object which allowed the visualization of the particle tracks within a 3D viewing window using egs_view (egs_view will be demonstrated in the results section later). Coding to define the ausgab object is seen below in Figure 3. When the track scoring object is chosen, an extra file is generated inside the application folder with the file extension *.ptracks following the simulation. This file is used do display the tracks of the simulated particles within the egs_view application.

```
      :start ausgab object definition:
               :start ausgab object:
                       name = tracks
                       library = egs_track_scoring
              :stop ausgab object:
      :stop ausgab object definition:
```

Figure 3: Coding used in this project to define the ausgab object.

Media definitions are done in either a PEGS4 or a pegsless mode within the EGS framework. PEGS4 mode can be used for materials which have had their "interaction cross sections calculated a-priori using the PEGS4 code" (Kawrakow, 2017, p. 268). Pegsless mode may be used for any other materials; however, parameters for the cross-section calculation must be included for each medium used in the geometry definition (Kawrakow, 2017, p. 268). When using pegless mode, a set of density correction files for each element and many compounds can be found inside the EGS file structure. Refer to the reference above for the file location. An abbreviated version of the coding used is shown below in figure 4. Factors ae and ap are the lower threshold for electron and photon energy, respectively, below which a particle in the simulation will be disregarded; similarly, ue and up define the upper energy threshold above which particles will be ignored. In my coding below, air and copper are examples of a compound and an element which have density correction files already defined. A density correction file must exist in the proper location with these filenames, or the application will error out. The other compounds and elements used in this simulation were; tungsten, beryllium, kapton, and vacuum (Kawrakow, 2017, p. 268).

```
:start media definition:

        ae = 0.521
        ap = 0.01
        ue = 50.51
        up = 50

        :start air:
        density correction file = air_dry_nearsealevel
        :stop air:

        :start copper:
                density correction file = copper
        :stop copper:

:stop ausgab object definition:
```

Figure 4: Abbreviated coding used in this project to define the media used in the simulation.

Geometry definitions are the most time consuming factor in the creation of beam simulations using egs++ language. For this section of the modelling, information was obtained from a commercial manufacturer of linear accelerators. As such, much of the geometry information used in my simulation is proprietary and will not be displayed here for the reader. What is displayed is a much abbreviated version of the geometry definition to give the reader insight into the manner in which geometries were created by the student for simulations. The particle source is defined as a pencil beam of electrons travelling through the origin and along the +z axis. So the objects used to shape the primary beam are built around the z-axis making it the central ray of the beam. Referring to table 1, the objects I modelled for the photon beam are a target, a primary collimator, the flattening filter, an ionization chamber, and a secondary collimator. The one main difference to this ordering in the model created was that the pencil beam passes through a small aperture in the primary collimator prior to hitting the target, so the beam passes through vacuum at the beginning of my geometry definition prior to target interaction.

```
      :start geometry definition:
               :start geometry:
                       library = egs_cones
                       type = EGS_ConeStack
                       name = Base_particle_track_10x
                       axis = 0,0,0,0,0,1
                       :start layer:
                               thickness = z
                               top radii = tr1 tr2 tr3
                               bottom radii = br1 br2 br3
                               medium = vacuum tungsten air
                       :stop layer

      #### All layers created in this manner to define the Primary
########## Collimator, flattening filter and ion chamber (on the order of 80
########## layers defined)
               :stop geometry:

      ################# Begin Y Jaw definition #####################
               :start geometry:
                       name = Y_Slab
                       library = egs_ndgeometry
                       type = EGS_XYZGeometry
                       x-planes = x1 x2 x3 x4
                       y-planes = y1 y2 y3 y4
                       zplanes = z1 z2
                       :start media input:
                               media = tungsten air
                               set medium = 4 1
                       :stop medium input:
               :stop geometry:

      #### X Jaw definition created in the same manner as Y ######
                       name = X_Slab

      ############# Begin air box definition #####################
               :start geometry:
                       library = egs_box
                       name = my_box
                       box size = 40 40 268
```

```
                        :start media input:
                                media = air
                        :stop media input:
                :stop geometry:

        ######### Combine all other geometries inside my_box #########

         :start geometry:
                name = simulation_box
                library = egs_genvelope
                base geometry = my_box
                inscribed geometries = Base_particle_track_10x Y_Slab X_Slab
         :stop geometry:

         simulation geometry = simulation_box

        :stop geometry definition:
```

Figure 5: Abbreviated coding for the geometry definition.

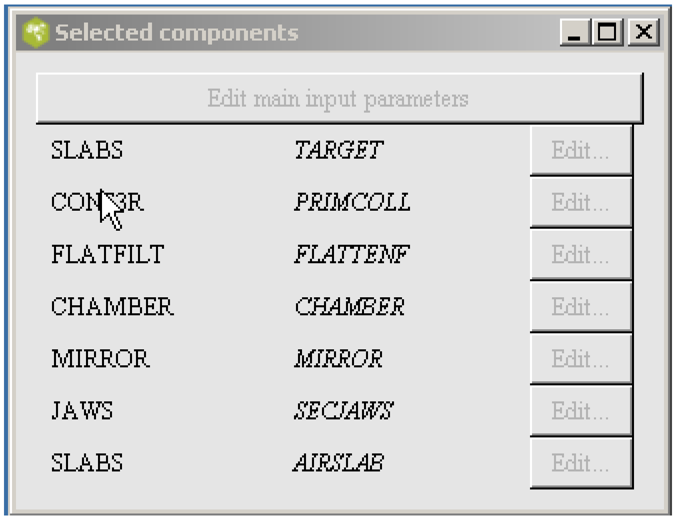
Only one simulation can be defined at any given time to run a simulation, although there are many geometries defined above. In this case, I defined the geometry "my_box" to be a large box of air with dimensions 40 cm x 40 cm x 268 cm. Creation of a box this size serves as a base space that will contain all the other geometries so that they can be inscribed inside it by the egs_genvelope library. The geometry that envelopes all the others, named "simulation_box", is then used as the simulation geometry so that all other geometries are included in the simulation.

The code used for the MC transport parameters is not described in this section because all the default variables were used in this simulation. There is a list of these default settings inside the EGSnrc C++ class library (Kawrakow, 2018) if the reader is interested; However, if the user wishes to apply default transport settings this section of coding can be omitted from the input file and the default settings will be applied.

CHAPTER 5

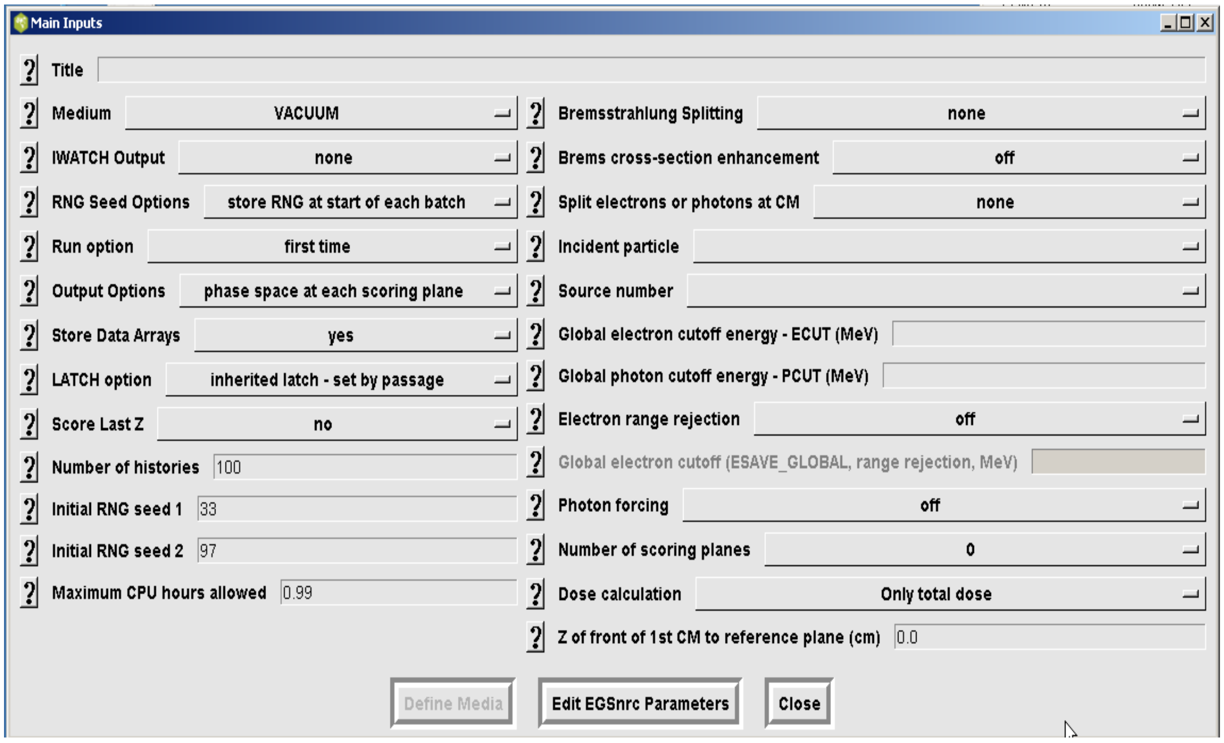
BEAMNRC PARAMETERS FOR 10MV PHOTON BEAM

BEAMnrc was used to re-create the same geometry described above in the egs++ coding chapter. There are differences when using CMs rather than egs++ classes; however, and exact match was not achieved. I created a new linear accelerator, modelled after the example accelerator BEAM_EX16MVp, using the BEAMnrc GUI. Upon creation of a new accelerator, I was required to select CMs that will define the primary beam path. The list selected for this accelerator is displayed in Figure 6 Below. Once the accelerator is defined the CMs appear in the "selected components" window and can be edited through the GUI. Each CM is designed differently and the description of each one can be found in the BEAMnrc user's manual (Rogers, 2018, p. 140). Defining each component in the list completes the simulation geometry for this application. By selecting the "Edit main input parameters" button the user then can input the info needed to define the particle source, MC transport parameters, the number of histories to be run, and the desired output from the simulation. There are also many options that are not available in egs++ that can increase the accuracy or efficiency of the calculation. The main input parameters window is displayed in Figure 7 at the top of the next page. None of these advanced options were used during the example simulation presented here. As in the last example, the particle source used was a parallel circular beam with a radius of 0.15 cm incident on the target. The number of histories used in this simulation was 1e6. The global electron cutoff energy was 0.7 MeV while the Global photon cutoff energy was 0.01 MeV. The medium used in the simulation was air aside from those defined in the CMs. Finally, a phase space output file was selected to be generated at a plane perpendicular to the z axis at a distance of 100 cm from the target.



SLABS	*TARGET*	Edit...
CONESR	*PRIMCOLL*	Edit...
FLATFILT	*FLATTENF*	Edit...
CHAMBER	*CHAMBER*	Edit...
MIRROR	*MIRROR*	Edit...
JAWS	*SECJAWS*	Edit...
SLABS	*AIRSLAB*	Edit...

Figure 6: List of my Component Modules characterizing the linear accelerator in BEAMnrc



Title			
Medium	VACUUM	Bremsstrahlung Splitting	none
IWATCH Output	none	Brems cross-section enhancement	off
RNG Seed Options	store RNG at start of each batch	Split electrons or photons at CM	none
Run option	first time	Incident particle	
Output Options	phase space at each scoring plane	Source number	
Store Data Arrays	yes	Global electron cutoff energy - ECUT (MeV)	
LATCH option	inherited latch - set by passage	Global photon cutoff energy - PCUT (MeV)	
Score Last Z	no	Electron range rejection	off
Number of histories	100	Global electron cutoff (ESAVE_GLOBAL, range rejection, MeV)	
Initial RNG seed 1	33	Photon forcing	off
Initial RNG seed 2	97	Number of scoring planes	0
Maximum CPU hours allowed	0.99	Dose calculation	Only total dose
		Z of front of 1st CM to reference plane (cm)	0.0

Figure7: BEAMnrc main input parameters window

CHAPTER 6

DISCUSSION AND FUTURE WORK

Each of these simulation types have some advantages that deserve some attention here. When modelling geometries with egs++ it can be very advantageous to view your geometry in 3D while it is being created. Also I found it very informative to use different geometries as the simulation source so that I could visualize the effects each beam component had. In Figure 8, I have created a series of simulations that illustrate the coning down effect that the primary and secondary collimators achieve. Figure 8a) shows a simulation of the pencil beam interacting with only a solid slab of copper representing the target. It is easy to visualize why photon sources can be modelled as point sources at times. As the figures progress, I add the primary collimator around the target first, and then I add the remaining beam path. In the final Image, the beam's interactions with a slab of water can be seen. This helps to visualize what might happen inside of

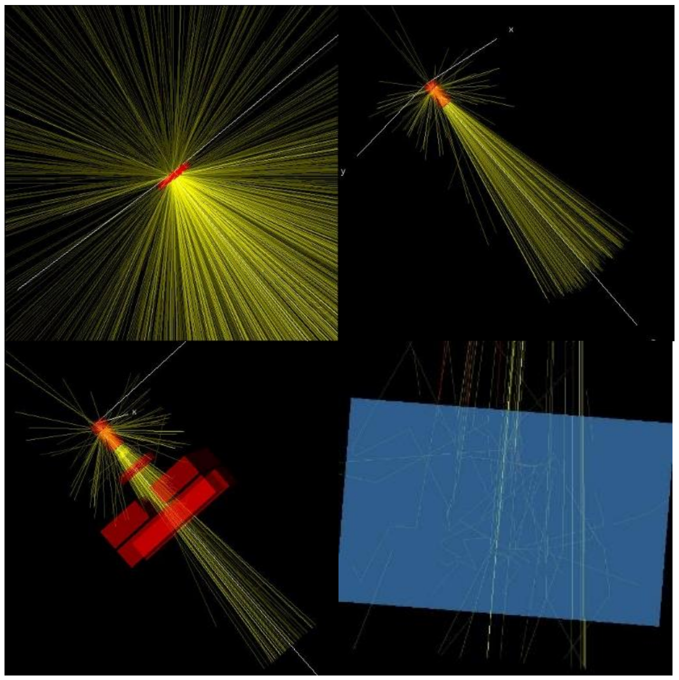


Figure 8: collimation of the beam path a) pencil beam striking copper slab b) surround copper slab by primary collimator c) cone down the beam using the jaws d) beam interacting with water slab at 100 cm SSD

the human body as patients are treated. It should be noted that the yellow lines represent the photon “tracks” that were generated earlier in the *.ptracks file.

BEAMnrc simulation advantages are observed in the data that can be generated in phase space file outputs. Phase space files obtained from a scoring plane contain information about each particles location, energy, and direction as it passes the scoring plane. These files are powerful because they contain enough information to be used in other applications (e.g., DOSXYXnrc) as particle sources. This application would allow absorbed dose to be calculated inside a CT dataset, for instance. This early simulation example is not near the level of sophistication used to treat patients today. But, it is a good start at building the basis needed to make more complicated simulations possible in the future.

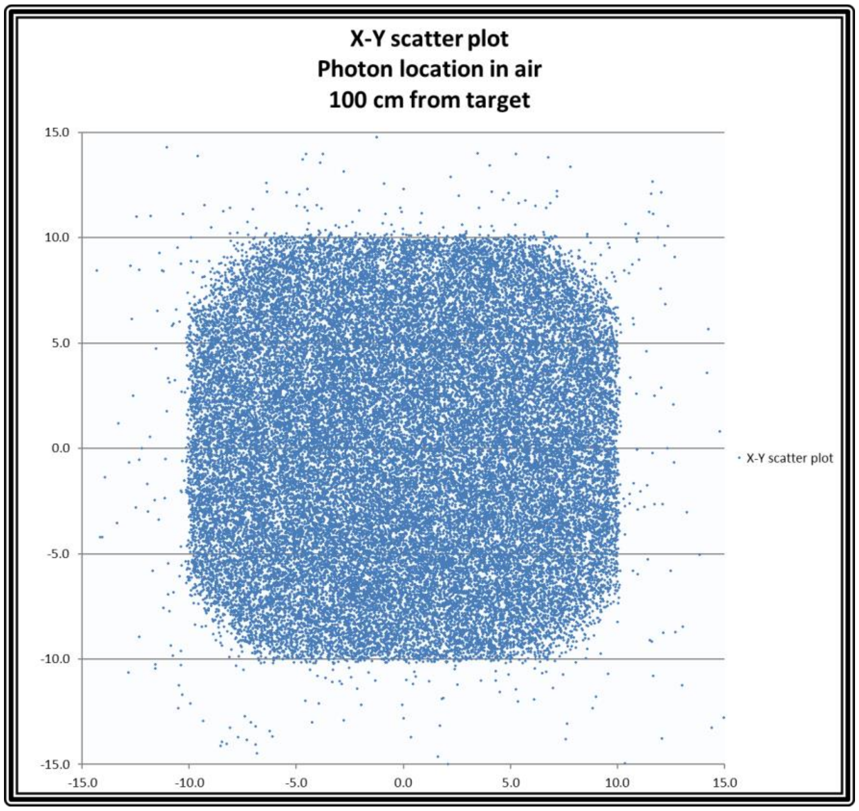


Figure 9: X-Y scatter plot of photons crossing a scoring plane at a distance of 100 cm from the target. Secondary collimators were designed to create a 10x10 cm beam at this distance away from the target in this simulation. X and Y are both perpendicular to the primary beam axis

Further work needs to be done in regards to validation of the current beam geometries. Validation of this data will require measured data of the linear accelerator that was modelled for comparison with calculations. Future works will also include calculating 3D dose in slabs of water using the DOSXYZnrc application, and testing of beam properties when irradiating more complicated geometries and differing density objects. Work can be done to improve the model for other beam shaping devices in the beam path referred to previously; such as, MLCs, electron cones, electron blocks, and physical wedges.

REFERENCES

1. Francescon, P., Cavedon, C., Reccanello, S., & Cora, S. (2000). Photon dose calculation of a three-dimensional treatment planning system compared to the Monte Carlo code BEAM. MEDICAL PHYSICS -LANCASTER PA-, (7), 1579. Retrieved from https://services.readcube.com/reader/pdf?ticket= ecdb97af-ba2e-460f-8085-a3094a8144bd

2. Khan, F. M. (2003). *The physics of radiation therapy*. Philadelphia : Lippincott Williams & Wilkins, c2003.

3. Ashokkumar, S., Nambi, Raj N. A., Sinha, S.N., Yadav, G., Thiyagarajan, R., Raman, K., Mishra. M.B. (2014) Comparison of Head Scatter Factor for 6MV and 10MV flattened (FB) and Unflattened (FFF) Photon Beam using indigenously Designed Columnar Mini Phantom. J Med Phys [serial online] Retrieved from: http://www.jmp.org.in/text.asp?2014/39/3/184/139010

4. Townson, R., Tessier, F., Mainegra, E., and Walters, B. (2018, June 1). Getting Started with EGSnrc Retrieved from: https://nrc-cnrc.github.io/EGSnrc/doc/getting-started.pdf

5. Kawrakow, I., Mainegra-Hing, E., Tessier, F., Townson, R., & Walters, B. (2018). EGSnrc C++ class library Report PIRS-898. Retrieved November 1, 2018, from https://nrc-cnrc.github.io/EGSnrc/doc/pirs898/index.html

6. Rogers, D.W.O., Walters, B., Kawrakow, I. (2018). BEAMnrc Users Manual NRCC Report PIRS-0509(A)revL Retrieved from: https://nrc-cnrc.github.io/EGSnrc/doc/pirs509a-beamnrc.pdf

7. Kawrakow, I., Mainegra-Hing, E., Rogers, D.W.O., Tessier, F., Walters, B.R.B. (2017). The EGSnrc Code System: Monte Carlo simulation of electron and photon transport. Technical Report PIRS-701, National Research Council Canada.

8. McGinley, P. H. (2002) *Shielding techniques for radiation oncology facilities* (2nd ed.). Madison, WI: Medical Physics Publ.

9. Eclipse™ Treatment Planning System. (n.d.). Retrieved November 01, 2018, from https://www.varian.com/oncology/products/software/treatment-planning/eclipse-treatment-planning-system

10. Radiation oncology treatment planning systems | Philips Healthcare. (n.d.). Retrieved November 01, 2018, from https://www.usa.philips.com/healthcare/solutions/radiation-oncology/radiation-treatment-planning

11. Elekta. (n.d.). Monaco®. Retrieved November 01, 2018, from https://www.elekta.com/software-solutions/treatment-management/external-beam-planning/monaco/

VITA

Graduate School
Southern Illinois University

Daniel J Mckinney

Danielmckinney1980@gmail.com

Utah Valley University
Bachelor of Science, Physics, May 2007

Research Paper Title:
Modelling of a Megavoltage Linear Accelerator Using EGSnrc to Perform Monte Carlo Calculations

Major Professor: Saikat Talapatra Ph.D.