GAMMORA
User Guide

Gammora was written by:

J. LESTE, T. YOUNES, M. CHAUVIN & L. SIMON

Institut Universitaire du Cancer de Toulouse, France
Centre de Recherches en Cancerologie de Toulouse, France

Abstract

This document describes how to install and how to use the GAMMORA package. GAMMORA is a tool that produces macro scripts for Monte-Carlo simulations with GATE v9.0 (or later) of the VARIAN TrueBeam STx linac. It includes the sources (6FF, 6FFF, 10FF and 10FFF), the geometry of the head, the ability to insert a CT, etc. GAMMORA is able to read DICOM RT PLAN to produce dynamic macros including the gantry and MLC motion.

Unfortunately, we tried to make this document as short as possible but it is not possible to start GAMMORA without reading it. Moreover we try to make the manual very clear even if you do not know GATE, but it is recommended to know the basis of the linux commands (bash) and it is better to know some basis of GATE.

GaMMoRa

Corresponding authors:

- Luc SIMON: simon.luc@iuct-oncopole.fr
- Jeremy LESTE: jeremy.leste@gmail.com

Contents

1. What is GAMMORA?

GAMMORA (GAte Monte-Carlo MOdel for RAdiotherapy) is a free and open-source tool under LGPL licence, available on Github [2].

GAMMORA is a comprehensive tool that automatically creates macro scripts for GATE v9.0 (with Geant4 version 10.6.1) for any kind of Monte-Carlo simulation of the most common radiotherapy device: the TrueBeam STx (Varian Medical Systems, Palo Alto, CA). It includes the different head elements (MLC, Jaws, ...) and the sources of photons (6FF, 6FFF, 10FF and 10FFF) and electrons as phase space files (pre-trained GAN that generates the particles). GAMMORA allows creating macro from a user configuration file or automatically from a DICOM-RT PLAN file. This includes the ability to model different plans such as VMAT, DCA, electron beams...

GAMMORA was implemented and validated in Toulouse (France) at CRCT (Cancer Research Center of Toulouse). Several PhD Thesis [12, 3, 8] and articles [6, 4, 14, 15, 13, 10, 9] were published about the validation and the applications of GAMMORA. We can cite the following clinical or research applications for which GATE and GAMMORA could be useful:

- 3D Second check of radiotherapy clinical plans
- Assessment of the interplay effect, and other influences of breathing motion
- Assessment of a new algorithm
- Assessment of output correction factor for new detectors

- Reference computation in complex conditions: small and very small fields, heterogeneous regions (bone, lung, metal, nanoparticles), motion.

- In a nutshell, every time that a reference computation is needed in radiotherapy or if the user wants to control or modify something in the computation process.

2. Installation

2.1. Installation of vGATE

The best way to start GATE [7] and GAMMORA for any kind of operating system is to use vGATE. vGATE is a virtual machine (VM) including Ubuntu, Geant4, GATE, code, etc. All is already compiled and installed properly (but sources are still inside!). vGate can be run on any OS (Mac, Windows and Linux) and is similar to a computer in your computer. Of course, advanced users can download and compile GATE themselves.

- Install virtual box on your system (or any other VM tool):
  `https://www.virtualbox.org/`

- You also have to install from the Download page of the Virtual Box website the **VirtualBox 6.1.16 Oracle VM VirtualBox Extension Pack**

- Download vGate v9.1 or v9.0 on the Download page of the openGate website:
  `http://www.opengatecollaboration.org`

Note: actually it is recommended to use Gate v9.1. If you use Gate v9.0 there is a bug when using multi-slice simulation.

The .ova file is quite large (8 Gb), place it in a safe directory. It can be deleted, once the VM is imported in Virtual Box.

- Launch virtual box
- In the main menu: **File** → **Import a Virtual machine**
- Choose the downloaded .ova file and click **Open**, then **Continue**
- Click **Import** and wait several minutes
- When the import is over, select the VM in the left part, and click on **Settings**
- Several parameters can be changed but here are the two main ones.
  - Under the **System** tab, select the quantity of RAM that you need for your VM. Select a value a least equal to 4000 Mo and no more than 80% of the maximum of your machine. Recommended value 10000 Mo.
  - Under the **Shared folders** tab, add a folder that will be shared by your real machine and your VM. Enable the option: **Auto-Mount**. In the VM, the directory will be mounted to the path: /media/sf_shareVB/
- Close the **settings** windows and click on the green arrow to start your vGate machine (if needed, password is the word: *virtual*)

- **Congratulations**, you are probably in a Ubuntu environment. Right click on the desktop to open a terminal. In the terminal if you type

  ```
  Gate
  ```

  you should be in the Gate environment. Then to quit type:

  ```
  exit
  ```

2.2. Installation of some needed packages

Install the following needed Python3 packages by typing in a terminal:

```
pip3 install stl
pip3 install mesh
pip3 install numpy-stl
pip3 install pandas
pip3 install panda
```

2.3. Two small tips

Maybe you will encounter the two following classic issues.

- For french user in particular, it may be possible that the keyboard is not properly selected. To select your keyboard: **Menu** (bottom left corner) → **control Center** → **keyboard**. Then select the keyboard under the **Layouts** tab

- For macOS users, the following operation must also be done. In the top menu of macOS (with the apple icon) Select **Devices** → **insert guest addition cd images**. Then, in a terminal type the following commands:
  - `cd /media/gate/VBox_GAs_5.2.18/`
  - `sudo sh VBoxLinuxAdditions.run`

    (password: virtual)
  - `sudo reboot`

    (the VM restarts...)

2.4. Installation of GAMMORA

GAMMORA is a github repository [2]. To clone it in your VM is very easy. Go in your home directory:

```
cd
```

Then, clone the package with the following command:

```
git clone https://github.com/uhqd/GAMMORA.git
```

It creates a GAMMORA/ directory on your computer, in the directory where you ran this last command.

To be completely ready you may have to compile two small C codes: In the GAMMORA/ directory type the following commands:

```
cd utils/C/
chmod 755 compile.sh
./compile.sh
```

This will create two binaries in /C/bin/

2.5. What's in the GAMMORA repository?

2.5.1. gaga-phsp/

Varian propose on its website [1] the sources of TrueBeam STx in IAEAphsp format. But, unfortunately, these files are not in open acces and are only for VARIAN customers. GAMMORA can create macro for these native files. For the users who do not have access to these IAEAphsp files, we propose in GAMMORA the phase spaces for photon sources for all energies as a pre trained Generative Adversarial Neural (GAN) network. These GAN were trained using the *gaga* tool (GAn fo GAte) [11].

This directory *gaga-phsp/* contains these *gaga* phase space (phsp) files of the TrueBeam for four photon energy modes: 6FF, 10FF, 6FFF and 10FFF.

The training has been performed by the GAMMORA team, using the IAEAphsp files, provided by Varian for its customer [5, 1].

These pre-trained GAN can generate any number of particles for your simulation.

Since Gate v9.0, it is possible to compile Gate with the libtorch library (already done in the VM). Thus, Gate is able to read any particle, generated with these gaga-phsp.

Each energy is composed of two files:

- .json: this contains the pytorch parameters
- .pt: the pre-trained GAN

This *gaga* package is freely available **here** on github. and was developed by the CREATIS team (Lyon, France) [11].

2.5.2. my_first_GATE_macro/

GAMMORA creates GATE macros. This is a simple example of GATE macro that uses our gaga-phsp files. See the section 3.1 to try this example.

2.5.3. macroMaker/

This is the GAMMORA project (see section 4).

3. Getting started

3.1. Hello world!

Here we are going to run a first example of GATE macro without GAMMORA but very similar to those created by GAMMORA.

For further details on GATE, please report to the openGate website.

In the GAMMORA directory, in your terminal, type:

```
cd my_first_GATE_macro/
Gate --qt mac/main.mac
```

The Gate QT visualization should appear and a simple Monte-Carlo simulation is running.

It consists in a gaga-phsp source of photon that turns and irradiate a phantom. There is not yet any elements of the linac head (jaws, MLC,...).

Close it when it is over and let's inspect the code using:

```
code mac/main.mac
```

In GATE, the commented line started with a #

Look at these commands and the commented lines to understand at least the structure of the macro.

The simulation is dynamic. It is made of several *run*. Each run can be seen as a new simulation.

The duration of each individual run are in the file:

```
data/myTime.timeslices
```

The number of primary particles for each run is in the file:

```
data/primary.dat
```

The position of the gantry at each run is in the file:

```
data/gantryMovement.placements
```

It is not possible to prepare these files manually for a real case, where gantry, collimator, MLC and even the patient can move during the simulation. This is the role of GAMMORA.

4. Hello world! ...but with GAMMORA!

4.1. Example 1: a full VMAT treatment with a patient CT

Let's see now the creation of a simple TrueBeam macro using GAMMORA.

But before this, the separation in twos steps must be explained. To speed up the computation, GAMMORA separates the simulation in two parts (see fig 1). The simulation in the head of the linac to create an intermediate phsp file (this will be in a *phsp/* directory) and the simulation in the patient (in *clinic/* directory). The user will have to execute both simulations to obtain some output (i.e. dose distribution in the patient).

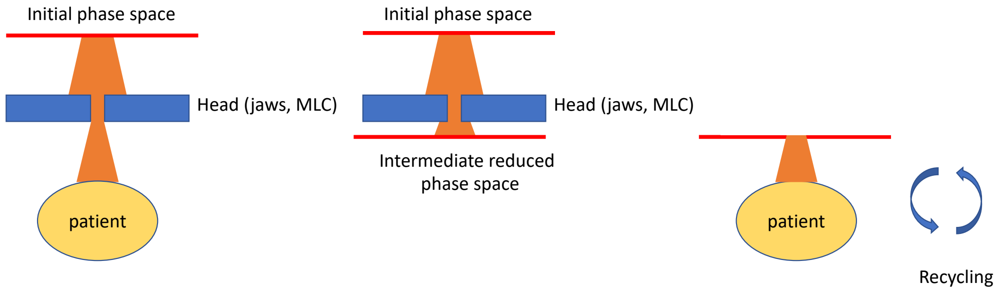

Figure 1: The complete simulation (left) needed by the user is separated in two sub-simulations. First (center) the 'head part' simulates the head and then the output particles are strored in plane at the exit of the head. Then (right) a second simulation is achieved (right) without the head, using as an input the phase space file created in the previous simulation

Here is a first example of the creation of macros in automatic mode, based on a clinical patient (with not much explanations, details are in the next section).

Images, and DICOM-RT data are in the directory:

```
/input/patient/MrGAMMORA/.
```

From the GAMMORA main directory, go in the macroMaker/ directory.

First, generate all the macros:

```
python3 ex1.py
```

This should read the configuration file (*config_file_ex1.txt*) and creates the two sets of macros in the *output/* directory (phsp and clinic)

In phsp/ you will find a directory (study name) that contains two directories (one for each field of the RT plan) that contains two directories: :

- The head step in /phsp/
- The patient step in /clinic/

First, we need to execute the GATE phsp simulation (head step). Go in the phsp/ dir:

```
cd output/myFirstSimu/myFirstSimu_beam_0/phsp/
```

The phsp directory contains the typical three GATE directories (*mac/ output/ data/*).

In this simple example there is no job splitting, thus mac/ contains only one sub-directory: 0/

Execute the GATE simulation:

```
Gate -a [out,output/0] mac/0/main.mac
```

This command executes the GATE simulation (main.mac) and the output data are stored in output/0/. You should see a dynamic simulation with gantry and MLC motion, but no patient at this step. After a few seconds, the simulation is over. The secondary phase space that will be used for the second step is: *output/0/myIAEA.root.*

Thus to execute the second GATE simualtion, go in the clinic dir/ and execute the script that prepares the second step:

```
cd ../clinic/
chmod 755 set_nb_reducued_phsp.bash
./set_nb_reducued_phsp.bash
```

Then execute the second step with the same command:

```
Gate -a [out,output/0] mac/0/main.mac
```

The final output is in *output/0/*

4.2. Example 2: another simple example in manual mode

The section 4.1 showed us how to use GAMMORA with a DICOM file (for a complex VMAT treatment and a patient CT). Now we are going to execute a second example in manual mode (static gantry, no MLC, SSD=100 cm, field 20x20 cm^2 a water phantom).

Let's go in *macroMaker/* directory and execute the following commands.

First to generate the two sets of macros

```
python3 ex2.py
```

Then to execute the head simulation:

```
cd output/watertank/waterTk/phsp/
Gate -a [out,output/0] mac/0/main.mac
```

Finally to execute the 'patient' simulation (in a water phantom):

```
cd ../clinic
chmod 755 set_nb_reducued_phsp.bash
./set_nb_reducued_phsp.bash
Gate -a [out,output/0] mac/0/main.mac
```

Then you will find the output in *clinic/output/0/*

5. The basics of use of the configuration file

5.1. In a nutshell

We saw in these last two examples that using a configuration file edited by the user, GAMMORA generates two sets of GATE macros for your TrueBeam simulations.

The main command to use GAMMORA and to create all your macros is:

```
python3 <main file.py>
```

This main file file (e.g. draft.py) is a one line python code :

```
Gammora_study.GammoraStudy("config_file.txt",
   study_type="manual", study_name="manual_test")
```

that defines 3 parameters that can be changed by the user:

- Name of the configuration file, here config_file.txt
- Study type: ***manual*** or ***patient***. Manual study type is used for simple macros (like example 2 in section 4.2). Patient study type is for automatic simulation based on DICOM files (see example 1 in section 4.1).
- Study name: set the name of the *study*. This will be the name of the sub-directory created in the output/ directory. The generated macros of GATE will be in the directory: output/<study name>/<simulation name>/The simulation name is set in the config file (see section 5.2).

Of course it is possible to add several lines like this one to generate more GATE macros. To execute a macro in Gate, the user must go in the created study directory and type:

```
Gate -a [out,output] --qt mac/0/main.mac
```

In this command:

- mac/0/main.mac is the GATE main macro file
- –qt is an option to enable the qt environment (only for simulation with visualization)
- -a [out,mypath] is a GATE option to replace {out} by mypath in the main macro

Let's inspect the configuration file: config_file.txt

5.2. The configuration file: ***general*** *section*

In this configuration file the user can set several parameters. Each parameter is located the line after a tag. The tag and the user input are in the following format:

```
=PARAMETER_NAME=
value
```

All other lines in the configuration file are ignored. GAMMORA only consider the lines located just AFTER a tag. In the following lines the gantry rotation is set to 90 degrees:

```
=GANTRY_ROTATION=

90

54

whatever
```

Most of tags can be set at the value *auto*. This is a default value defined in the file *Gammora_manual_beam.py*.

In our examples (sections 4.1 and 4.2) the configuration files called by GAMMORA were config_file_ex1.txt and config_file_ex2.txt resp.

This configuration file is read in both study types: manual and patient. But some parameters are ignored, depending on this study type. See the details hereafter.

- **=SIMULATION_NAME=** Must be a string with no space. This will be the name of the sub-directory created in **output/<study name>/**. If set to **auto** the simulation name will be the same than the study name (that is set in draft.py).

- **=VISU=** In Gate it is recommended to enable the visualization for testing and developping the macros and to disable it for data production. Set this parameter to:

  - **0** Visualization disabled
  - **1** Visualization enabled
  - **2** Visualization enabled (MLC view)
  - **auto** Default

- **=PHYSICS=** It is possible to generate macro without any physics list (for testing purpose)
  - **0** No physics list
  - **1** Physics list enabled. The template file */utils/mac/mac_template/Xphysics.macX* will be used as a physics list.
  - **file** The user gives a file name of the physics list that will be inserted in the GATE macros. This file must be in */utils/mac/mac_template/*
  - **auto** Default value is 1
- **=HEAD_SIMULATION=** Enables the creation of the set of macros for the linac head. Executing theses macros in GATE will simulate the particles transport in the linac head and create an intermediate phsp file at the output of the head. This phsp file will be used in the patient simulation.
  - **0** disabled
  - **1** enabled
  - **auto** Default is 1
- **=PATIENT_SIMULATION=** Enables the creation of the set of macros for the patient/phantom part. Executing these macros in GATE will simulate the particle transport in the patient. But, to execute these macros you will need the intermediate phsp, file previously created in the head simulation.
  - **0** disabled

- **1** enabled
- **auto** Default is 1

- **=INDEX=** Must be an integer (or **auto**) Gives the number of jobs when splitting the simulation.
  - **1** no splitting
  - If the dynamic split type is selected, the Index value gives the number of jobs created to simulate the total number of particles. Each job will simulate the whole field (e.g. a full arc in case of VMAT).
  - If the static split type is selected, it will divide the simulation in separate jobs that will simulate a part of the field (e.g. in case of VMAT). If the selected INDEX value is lower than the number of Control Points (CP) this value will be ignored and the number of jobs will be the number of CP. If the selected INDEX value, N, is higher than the total number of Control Points (CP), then the simulation will be divided in N jobs and the trajectory of the arc will be divided in more points than the CP (over sampling). Between two CPs, artificial CPs are created with an intermediate position of gantry and MLC.

- **=SPLIT_TYPE=** To set the splitting type. It can be set to **dyn** (dynamic) or **stat** (static) (or **auto**). See INDEX tag for the splitting number. Dynamic splitting is recommended. Static splitting can be used for users that want to study interplay effect (the whole arc is simulated for each target motion).

5.3. The configuration file: ***geometry*** *section*

These parameters are ignored if *STUDY TYPE* is **patient**. In this case they are read in DICOM file.

- **=GANTRY=** Must be a float between 0 and 360. Set Gantry angle (in deg). For arcs, set the start angle (auto = 0)
- **=GANTRY_STOP=** Must be **X** or a float between 0 and 360. To create arcs, set stop angle (in deg) else set to **X** for static field.
- **=ROTATION_DIRECTION=** Set rotation direction for arcs. **0** clockwise. **1** counter clockwise. No **auto** value.
- **=COLLI=** Must be a value between -180 and 180 Set Collimator angle (in deg). **auto** is 0.
- **=X1=** Must be an integer < 210. Set X1 jaw opening (mm).
- **=X2=** Must be an integer < 210. Set X2 jaw opening (mm).
- **=Y1=** Must be an integer < 210. Set Y1 jaw opening (mm).
- **=Y2=** Must be an integer < 210. Set Y2 jaw opening (mm).
- **=MLC=** Set the use of MLC 120HD. Possible values are:
  - **0** No MLC
  - **.mlc** Looking for a static MLC file (next tag) in *utils/mlc/*. This MLC file can be exported from Varian Eclipse. This option is under development and is not available in manual mode. To insert

a MLC, use the the patient study type (thus, this tag will be ignored).

- **=MLC_FILE=** Set the name of the MLC file if MLC tag is set to .mlc. The file must be in *utils/mlc/*. This option is under development and is not available. To insert a MLC, use the the patient study type (thus, this tag will be ignored).

- **=PHANTOM=** Set the phantom or patient model

  - **0** no patient or phantom
  - <**.mac filename**> A phantom as .mac file will be used. The .mac must be in *utils/mac/phantom/* For example, try **waterbox300.mac**
  - **auto** Set this tag to **auto** in patient study type to insert a patient CT. Insert a patient scanner is not possible in manual mode.

*5.4. The configuration file: **actor** section*

The actor in GATE defines a region where any output data must be collected. Thus, it is the association of a geometry (box, cylinder) and an output data type (phase space, dose distribution, counter...)

- **=PHSP_ACTOR=** To define a phsp actor. A PhSp actor in GATE, is an actor that store the incident particles in a file that can be use as a source for another simulation.

  - **0** no phsp actor
  - **1** use the default phsp actor *utils/mac/actor/iaea_make_phspR_cyl.mac*

- **auto** is 1
- **.mac filename** An actor as .mac file will be used. The .mac must be in *utils/mac/actor/* For example, try **waterActor.mac** or **portalActor.mac**

. Notice that the IAEA format for phsp can not be recycled for the patient simulation in dynamic splitting mode.

- **=PATIENT_ACTOR=** To define an actor in the patient or phantom. This actor can for example store dose distribution.
  - **0** no patient actor
  - **1** Use the default actor *utils/mac/actor/dose_actor.mac*. The next two tags MUST be specified.
  - **auto** is 1
  - **.mac filename** An actor as .mac file will be used. The .mac must be in *utils/mac/actor/* For example, try **dose_actor_octa.mac** The next two tags will be ignored if this option is used.
- **=PATIENT_ACTOR_SIZE=** If the default Patient Actor is used, you must define the actor size. This tag is ignored if a file was used as a patient actor (tag PATIENT_ACTOR).
  - **auto** 300 x 300 x 300 mm^3
  - <**x y z**> (mm) (must be 3 float). Ex. **300.0 125.0 425.0** will create an actor centered on the patient or phantom with the size 300.0 x 125.0 x 425.0 mm^3

- **=PATIENT_ACTOR_RESOLUTION=** If the default Patient Actor is used, you must define the actor resolution (number of voxels). This tag is ignored if a file was used as a patient actor (tag PATIENT_ACTOR).
  - **auto** 300 x 300 x 300 voxels.
  - <**x y z**> (voxels) (must be int) ex: **300 150 450**. The actor with the size defined with the previous tag will be divide in 300 x 150 x 450 voxels.

5.5. The configuration file: ***source*** *section*

This section describes the sources that will be included in the GATE macros.

- **=SOURCE=** It describes the format of the files used as a phase space source.
  - **0** no source
  - **auto** default is IAEAphsp
  - **iaea** Will use IAEAphsp provided by Varian for its customer only. Not included in GAMMORA
  - **gaga** Recommended value: it will use our pre trained GAN that generates particles..
- **=ENERGY=** Will select the energy (user must check that files are available). User can select:
  - **6X**

  - **10X**
  - **6FFF**
  - **10FFF**
  - **auto** (6X)

- **=NB_PART=** Must be an integer. Select the total number of particles of all generated macros. **auto** is 1000000000. Ignored in patient mode.

- **=RECYCLING=** Must be an integer. Select the number of recycling value (i.e. number of times that the same particles will be used). This is only for the patient part of the simulation using an intermediate phsp source (not the head simulation). **auto** is 100

5.6. The configuration file: ***computing*** *section*

This section is only for one option.

- **=EXECUTE_ON_CLUSTER=** This option is to prepare a script for splitted simulation to be computed on a super computer (or a grid) that accept SLURM scripts.
  - **0** Local simulations
  - **3** Prepare the SLURM scripts for the head and patient simulation. Must be configured for your needs.

6. Bibliography

[1] MyVarian webpage. URL: https://varian.force.com/.

[2] GAMMORA Repository. 2020. URL: https://github.com/uhqd.

[3] Beilla, S.. [Modélisation Monte-Carlo d'un accelérateur linéaire pour la prise en compte des densités pulmonaires dans le calcul de la dose absorbée en Radiothérapie Stéréotaxique], PhD Thesis, Université de Toulouse. Ph.D. thesis; Université de Toulouse; 2016.

[4] Beilla, S., Younes, T., Vieillevigne, L., Bardies, M., Franceries, X., Simon, L.. Monte Carlo dose calculation in presence of low-density media: Application to lung SBRT treated during DIBH. Physica Medica 2017;41:46–52. doi:10.1016/j.ejmp.2017.04.009.

[5] Constantin, M., Perl, J., LoSasso, T., Salop, A., Whittum, D., Narula, A., Svatos, M., Keall, P.J.. Modeling the truebeam linac using a CAD to Geant4 geometry implementation: dose and IAEA-compliant phase space calculations. Medical physics 2011;38(7):4018–24. URL: http://www.ncbi.nlm.nih.gov/pubmed/21858999.

[6] Delbaere, A., Younes, T., Vieillevigne, L.. On the conversion from dose-to-medium to dose-to-water in heterogeneous phantoms with Acuros XB and Monte Carlo calculations. Physics in Medicine and Biology 2019;64(19). doi:10.1088/1361-6560/ab3df3.

[7] Jan, S., Santin, G., Strul, D., Staelens, S., Assié, K., Autret, D., Avner, S., Barbier, R., Bardiès, M., Bloomfield, P.M., Brasse, D.,

Breton, V., Bruyndonckx, P., Buvat, I., Chatziioannou, A.F., Choi, Y., Chung, Y.H., Comtat, C., Donnarieix, D., Ferrer, L., Glick, S.J., Groiselle, C.J., Guez, D., Honore, P.F., Kerhoas-Cavata, S., Kirov, A.S., Kohli, V., Koole, M., Krieguer, M., van der Laan, D.J., Lamare, F., Largeron, G., Lartizien, C., Lazaro, D., Maas, M.C., Maigne, L., Mayet, F., Melot, F., Merheb, C., Pennacchio, E., Perez, J., Pietrzyk, U., Rannou, F.R., Rey, M., Schaart, D.R., Schmidtlein, C.R., Simon, L., Song, T.Y., Vieira, J.M., Visvikis, D., Van de Walle, R., Wieërs, E., Morel, C.. GATE: a simulation toolkit for PET and SPECT. Physics in medicine and biology 2004;49:4543–4561. doi:10.1088/0031-9155/49/19/007. arXiv:0408109.

[8] Leste, J.. Implementation and clinic application of a Monte Carlo model of an external radiotherapy linear accelerator, PhD Thesis, Université de Toulouse. Ph.D. thesis; Université de Toulouse; 2020.

[9] Leste, J., Medjahed, I., Arnaud, F.X., Ferrand, R., Franceries, X., Bardies, M., Simon, L.. A study of the interplay effect for VMAT SBRT using a four-axes motion phantom. Journal of Applied Clinical Medical Physics 2020;21(8):208–215. doi:10.1002/acm2.12947.

[10] Leste, J., Younes, T., Chauvin, M., Vieillevigne, L., Bardies, M., Franceries, X., Nalis, J., Simon, L.. 36 Monte Carlo simulation of absorbed dose distribution for electron beam using GATE/GEANT4. Physica Medica 2018;56:21. URL: https://linkinghub.elsevier.com/retrieve/pii/S1120179718312110. doi:10.1016/j.ejmp.2018.09.049.

[11] Sarrut, D., Krah, N., Létang, J.M.. Generative adversarial networks (GAN) for compact beam source modelling in Monte Carlo simulations. Physics in Medicine and Biology 2019;64(21). URL: https://pubmed.ncbi.nlm.nih.gov/31470418/. doi:10.1088/1361-6560/ab3fc1. arXiv:1907.13324.

[12] Younes, T.. [Méthodologie pour la détermination de la dose absorbée dans le cas des petits champs avec et sans hétérogénéités pour des faisceaux de photons de haute énergie], PhD Thesis, Université de Toulouse. Ph.D. thesis; Université de Toulouse; 2018.

[13] Younes, T., Beilla, S., Simon, L., Fares, G., Vieillevigne, L.. PO-0804: Relative dosimetry evaluation for small multileaf collimator fields on a TrueBeam linear accelerator. Radiotherapy and Oncology 2017;doi:10.1016/s0167-8140(17)31241-0.

[14] Younes, T., Chauvin, M., Delbaere, A., Labour, J., Fonteny, V., Simon, L., Fares, G., Vieillevigne, L.. Towards the standardization of the absorbed dose report mode in high energy photon beams. Physics in Medicine & Biology ;:PMB–110997 (accepted, under revision).

[15] Younes, T., Vieillevigne, L.. Validation of the dose calculation algorithm Acuros XB and impact of its usage in clinical stereotactic treatments. Physica Medica 2015;doi:10.1016/j.ejmp.2015.10.029.