InterDosi User' manual

a user-friendly open source code aimed at facilitating the use of Geant4 C++ toolkit for internal dosimetry on voxelized phantoms

https://github.com/EL-Bakkali-Jaafar/InterDosi

Table Of Contents

Page

1.0 INTRODUCTION

2.0 ARCHITECTURE OF INTERDOSI

3.0 GETTING STARTED WITH INTERDOSI

4.0 EXAMPLE OF MODELING OF SOME HUMAN AND BIOTA VOXELIZED PHANTOM

5.0 TROUBLESHOOTING

6.0 FOR DEVELOPERS

Table Of Contents

1.0 INTRODUCTION

1.1 About this document

This document refers to the InterDosi open source version 1.0 and comprises some instructions and tips which directs the user of this code to easily establish internal dosimetry studies either in human or non-human biota voxelized phantoms. The code is user-friendly since it supports a well-developed Java-based GUI application whereby internal dosimetric calculation based on powerful Monte Carlo, namely Geant4 are made easy and less effort is required since for. It should be noted here, that user of this multi-threaded code does not need to have a deep knowledge of Geant4 Monte Carlo code nor to consult further documentation as this manual has been constructed in such manner that the self-consistent concept is supported well. In all cases, anyone have a question about this document, should don't hesitate to emailing me at:

> bahmedj@gmail.com

1.2 Quick Introduction to InterDosi

The word InterDosi stands for "**Inter**nal **Dosi**metry", as its name implies, this code is about of a Monte Carlo modeling tool especially oriented for performing internal dosimetry studies on computational human and non-human biota voxel-based phantoms. Dose calculation engine used implicitly by this code is Geant4 Monte Carlo code for the reason that this most famous and well-recognized tool is powerful and widely used around the world for various applications involving the simulation of ionizing radiation transport and its interaction with the matter.

With knowing that the learning of Geant4 code is not straightforward as the coding with is by using C++ programming language and requires huge time to learn a such code, this reflects negatively on the exorbitant effort required to build an advanced application based on Geant4 code. To deal with this issue, at that point in particular, this open source code has saw the light in order to provide to scientific researchers whom have a poor knowledge in Geant4/C++

programming, a user-friendly platform allowing them to perform internal dosimetric studies on voxelized phantoms in an easy manner and in a very short time too. The code offers a set of useful graphical tools helping user to quasi-automatically perform all steps required for building a complete internal dosimetry study. Briefly, this tool allows user to import a 3D model of any existing phantom having a common file format or a well-known structure like as GDML, STL, OBJ and MCNP hexahedra lattice, and it converts them to a suitable voxel-based phantom file format accepted by the code, this operation is done into few steps, almost automatically. The involved dosimetric quantities as specific absorbed fraction and absorbed fraction can be calculated in a voxel-based phantom for a given configuration made up of three parameters, namely: source organ, primary particle type and primary particle energy. The code gives an oportunity to user to choice betwenn the most particle types emitted by radioactive substances used in nuclear medicine, including electron, positron, gamma, alpha, neutron and proton. In addition, code allows user to visualize phantom geometry in two or three dimensions by implicitly calling Geant4 visualization system. Visualization of internal dose heatmap for a given voxel-based phantom may be possible either in sagittal, coronal or transversal view, which give to user a visual idea of how internal absorbed doses are two-dimensionally distributed in different phantom organs for a particular slice from phantom whole volume.

Architecturally, InterDosi is a multi threaded tool that supports parallel calculations feature which knows to be a good approach to reduce statistical errors associated to scored physics quantities as well as reducing whole time spent by a given simulation. InterDosi can be run in sequential mode that is means that one CPU is used, or in parallel mode where a multi-core architecture is exploited. In order to enhance simulation statistics, a particle recycling variance reduction technique is also available in which each primary particle is repeated for a given times.

The present open-source software has been coded by Jaafar EL Bakkali, Assistant Professor in Nuclear Physics, which is working at Royal School of Military Health Service, Rabat-Morocco.

2.0 ARCHITECTURE OF INTERDOSI

2.1 Programming languages used for the creation of InterDosi

Different components of InterDosi code have been coded by using the following programming languages:

i. Java programming language, used to code a user-friendly Java-based application which makes the work with InterDosi code more easy and a high flexibility is provided so.

ii. C++/Geant4 programming language, used specially in development of the core of InterDosi code which consists of two Geant4-based modules.

iii. Linux scripting language, used for making various scripts that ensure various aspects of connections between C++ modules and Java-based main GUI application.

2.2 Software files and folders

	Name	Description
Folder	bin	This directory contains binary files belonging to two Geant4-based codes localized into the core folder, namely *“G4InterDosi”* and *“IJKID_Builder”*.
	config	This directory contains a single file called *“InterDosi.config”* which contains a one line readable text that points out to the full path to the Geant4 installation location.
	core	This directory contains four main folders, namely: • G4InterDosi: contains source code written in C++/Geant4 language of a Geant4-based application dedicated to perform internal dose calculations. • IJKID_Builder: contains source code written in C++/Geant4 language of a Geant4-based application dedicated to voxelize GDML tessellated geometries of a given phantom. • MaterialsDB: this repository contains a list of files in a plain text format, owning a file extension of “.mat”, each file has a name of a particular human or non-human biota tissue and stores material specification owned by this

		tissue that requires value of density, number of chemical elements and a percentage of each of them. For more information about this file format please see the section *“2.3.6 MAT data format”*. • Scripts: this repository contains a number of bash-based scripts ensuring strong connections between all developed modules and tools. There is in total five scripts named as follows: - IJKID_Builder_build.sh: this script is responsible for executing compilation task for IJKID_Builder Geant4-based code. - IJKID_Builder_run.sh: the running task of IJKID_Builder Geant4-based code is implicitly handled by this script. - interdosi_build.sh: auto-compilation of G4InterDosi Geant4-based code is performed by this Linux scripts. - calculate_dosimetric_quantities.sh: this Linux script automatically run G4InterDosi Geant4-based code that calculates internal dosimetric physics quantities. - visualization.sh: this Linux script calls in a background Geant4 visualization system and passing to it a set of user defined visualization parameters.
	doc	This directory contains software user’s guide in PDF file format (the present document).
	outputs	This directory contains three main output folders: • DoseInXYZ_files: this directory contains a set of files in DoseInXYZ file format. For more information about this file format please see the section *“2.3.8 DoseInXYZ data format”*. • Dosimetric_files: this directory contains a number of produced excel-based files containing internal dosimetric results of a given calculation. • Phantinfo_files: this directory contains a set of files in Phantinfo file format, for more information about this file format please see the section *“2.3.9 Phantinfo data format”*.
	lib	This directory contains a set of libraries used by the main Java-based application.
	inputs	This directory contains four main inputs folders: • GDMLTesselatedPhantom_files: Herein, user should put GDML files. For more information about GDML file format please see the section *“2.3.5 GDML data format”*. • IJKIDPhantom_files: In this directory, is where user should put IJKID files. For more information about IJKID

		file format please see the section *“2.3.4 IJKID data format”*. • STLPhantom_files: Herein where user should put STL-based phantom files. Detailled information about STL file format can be seen on the section *“2.3.1 STL data format”*. • OBJPhantom_files: Herein, user should put OBJ files. For more information about OBJ file format please see the section *“2.3.2 OBJ data format”*. • PhantomConfiguration_files: Herein where InterDosi_GUI will create an user-defined new phantom configuration file or where the program will load an existing configuration file. For more information about VOXPH file format please see the section *“2.3.7 VOXPH data format”*. • MCNP-Lattice-basedPhantom_files: Herein, user should put MCNP lattice-based phantom files. For more information about the format of this file, please see the section *“2.3.3 MCNP Lattice data format”*.
	schema	This directory contains GDML schema used by Geant4 code to load and validate GDML tessellated geometries read from files localized at “GDMLTesselatedPhantom_files” input directory.
File	InterDosi_GUI.jar	This is the main application that should be run by user, it is written in Java language and can be launched by typing on Linux terninal: *java -jar InterDosi_GUI.jar*

2.3 Description of all supported data input formats

2.3.1 STL data format

STL stands for the word “*Stereolithography*”, it is about a file format native to the stereolithography CAD software created by 3D Systems. STL file stores information about 3D model and it is used by many software packages for its rapid prototyping since it describes only the surface geometry of a three-dimensional object without any representation of other common CAD model attributes (color, texture, ...). The STL format can be specified either in ASCII or in binary representations. It should be noted here, that the ASCII representation is the only format that is supported by the InterDosi code.

For more information about STL file format, do not hesitate to visit the following webpage:

https://all3dp.com/what-is-stl-file-format-extension-3d-printing/#pointone

2.3.2 OBJ data format

OBJ is a file format native to "Wavefront Technologies" used in its advanced visualizer animation package. It is about a simple data-format that represents 3D geometry. This file format has been widely used by various software packages. Also, the ASCII representation is the only format that is supported by the InterDosi code.

For more information about this file format, please take a look at:

http://paulbourke.net/dataformats/obj/

2.3.3 MCNP Lattice data format

As it is well-known that the default MCNP input data format is "INP" which contains the input information necessary to describe any modeled experimental setup. In MCNP paradigm, lattices are repeated structure features used to fill cells of repeated structure, it can be either hexahedra (LAT=1) or hexagona prism (LAT=2). The hexahedra lattice with a universe containing a homogeneous cube can represent a voxel in an image that is identified by a chemical composition and density corresponding to a such organ. The InterDosi code via the tool "**MCNP-Lattice-To-IJKID converter** " can only process data related to lattice structure of type 1 (namely hexahedra) and ignores other input data.

2.3.4 IJKID data format

IJKID data format is a native file format belongs to InterDosi code, in such file the data of a voxelized phantom are stored in an ASCII text, this, line flowed by another line and each line saves four data: voxel IDs along x, y, and z axis, and a material ID.

2.3.5 GDML data format

Due to its flexibility, Geant4 Monte Carlo code provides various approaches to code materials and geometries of a specific experimental setup, one of them is by using an XML markup language called GDML that providing a simple read/write mechanism allowing user to code geometry and material data in an XML-based format. It should be emphasized here, that GDML code can be flexibly modified without needing to compile again the Geant4-based application. For more information, do not hesitate to read the following document:

http://lcgapp.cern.ch/project/simu/framework/GDML/GDML2/doc/GDMLmanual1.3.pdf

2.3.6 MAT data format

MAT data format is a native file format of InterDosi code, in such file, data specifying material that composites each phantom organ are written in an ASCII text format, this is done line by line and follow this structure:

```
NameOfOrgan  Density (in g/cm3) NumberOfComponents
ChemicalElement_1  Fraction_1
……
ChemicalElement_N  Fraction_N
```

As we can see above, the first line contains three parameters: the name of tissue/organ, density of that tissue/organ and number of chemical elements composing this tissue/organ. Whereas other lines contain for each one an abbreviation of a chemical element and its massive fraction. Here a basic example of how this file look like is provided (Air material).

```
Air 0.00120479 3
Ar 0.013
N 0.755
O 0.232
```

2.3.7 VOXPH data format

VOXPH data format is a native file format that is owned by InterDosi open-source code, in such file data associated to a voxelized phantom configuration file are stored in an ASCII text format. The data format is simple and assembled by a binary of flag and value(s). The flag is a word without number and it is started by "@" symbol.

To have a better idea of the structure of this file, herein we provide a demonstrative example:

```
@VoxelizedPhantomName       NameOfThePhantom
@VoxelizedPhantomFileDir    ijkid_File_FullPath
@Voxelizedimensions   X_DIMENSION        Y_DIMENSION         Z_DIMENSION
@NumberOfVoxels       NVOXELSX    NVOXELSY    NVOXELSZ
@NumberOfPhantomOrgans      NumberOfPhantomOrgans
@BackgroundID         -1
@PhantomOrganItem     0          OrganName_0  MaterialName_0        CLR_R  CLR_G CLR_B
.
.
.
@PhantomOrganItem     N          OrganName_N  MaterialName_N        CLR_R CLR_G  CLR_B
```

As remark, a file having an extension of ".voxp" can be generated automatically by InterDosi_GUI Java-based application or can be edited or recreated via any known text editor.

2.3.8 DoseInXYZ data format

DoseInXYZ data format is a file format native to InterDosi program, which is a default file format used by the incorporated visualizer tool that lets user see a two dimensional map of internal dose distributed following a given axis for a given slice from whole phantom volume. The content of a file having this data format is as follows:

The first line is always contains full path of used voxelized phantom configuration file of an extension of *".voxph"*.

Other lines contain four raws per line, the stored data are: voxel ID along x axis (Idx), voxel ID along y axis (Idy), voxel ID along z axis (Idz) and a normalized absorbed dose absorbed by a particular voxel (Idx, Idy, Idz).

The content of this text file may look like:

```
/home/jaafar/InterDosi/inputs/PhantomConfiguration_files/DigiMousePhantom.voxph
11      385     43      109.164
11      390     43      1421.61
12      391     44      171.604
13      373     40      306.448
13      373     41      358.998
13      386     44      336.567
13      388     44      1627.6
```

2.3.9 Phantinfo data format

Phantinfo data format is an ASCII text file format native to InterDosi program which stores interested information about each organ of a simulated voxel-based phantom, including: organ name, organ mass and organ volume. This file will be produced just after calculations have done, and the created file will be taken place at *"/outputs/Phantinfo_files"* directory. Here a demonstrative example of what this file may look is presented:

```
===============================================================
ORGANE_NAME: Gonads
ORGANE_MASSE (  g): 2.160538e+01
ORGANE_VOLUME ( CM_3 ): 2.077440e+01
===============================================================
ORGANE_NAME: Gills
ORGANE_MASSE (  g): 6.850560e+01
ORGANE_VOLUME ( CM_3 ): 6.850560e+01
===============================================================
ORGANE_NAME: Hepatopancreas
ORGANE_MASSE (  g): 3.085992e+01
ORGANE_VOLUME ( CM_3 ): 2.939040e+01
```

```
======================================================================
ORGANE_NAME: Heart
ORGANE_MASSE (  g): 2.378880e+00
ORGANE_VOLUME ( CM_3 ): 2.265600e+00
======================================================================
ORGANE_NAME: Shell
ORGANE_MASSE (  g): 3.419869e+02
ORGANE_VOLUME ( CM_3 ): 3.288336e+02
```

2.4 Modules of InterDosi

2.4.1 G4InterDosi

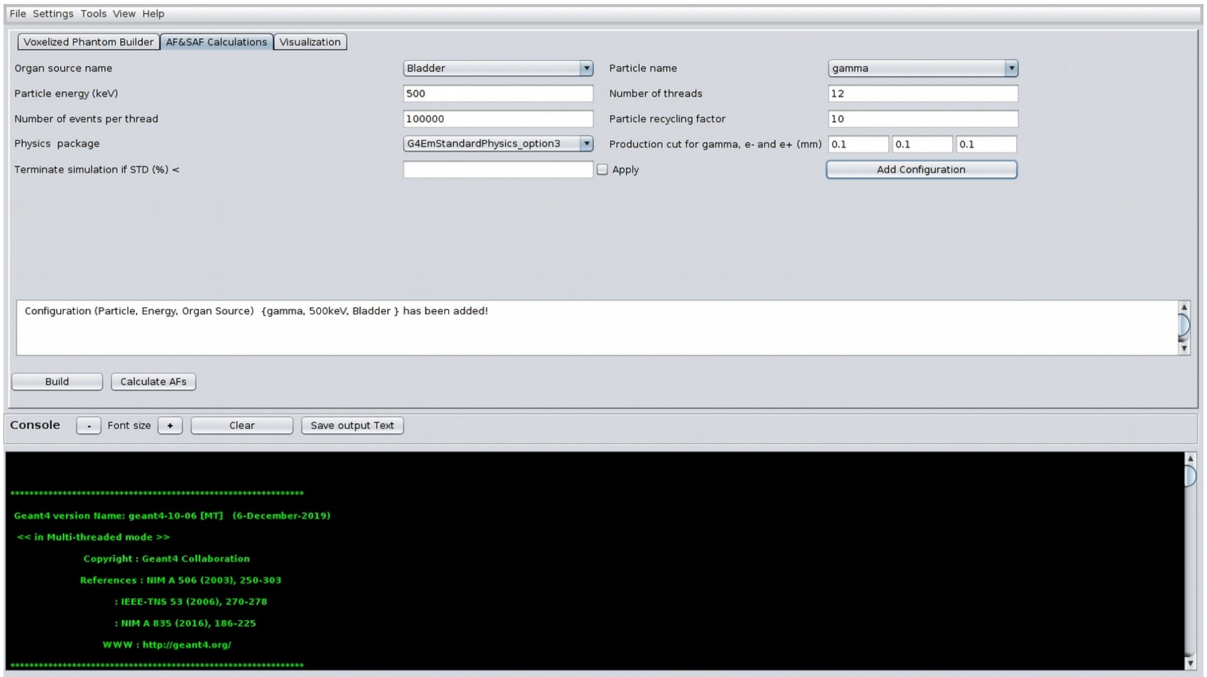
This module contains a set of C++/Geant4 classes for reading geometry setup, physics and primary generator parameters. Calculation of internal dosimetric quantities are performed in a background without user intervention. User should define parameters via InterDosi_GUI Java-based application which passes them to this C++ module.

2.4.2 IJKID_Builder

This module is responsible for the creation of a voxel-based phantom file having ".ijkid" file format from tessellated geometries that were read from a GDML file. This module is implicitly invoked by "*IJKD-to-GDML converter*" tool of the InterDosi_GUI Java-based application.

2.4.3 InterDosi_GUI Java-based application

This module is the main interface that can be manipulated by the user and it is run in foreground, which is about an user-friendly Java-based application that contains a set of well-developed GUI components helping user to setup their voxel-based phantom and run AF/SAF calculations in a fast way. In the below picture a snapshot of InterDosi_Gui is provided.



Organ source name: Bladder
Particle energy (keV): 500
Number of events per thread: 100000
Physics package: G4EmStandardPhysics_option3
Terminate simulation if STD (%) <
□ Apply
Particle name: gamma
Number of threads: 12
Particle recycling factor: 10
Production cut for gamma, e- and e+ (mm): 0.1 0.1 0.1

Configuration (Particle, Energy, Organ Source) {gamma, 500keV, Bladder } has been added!

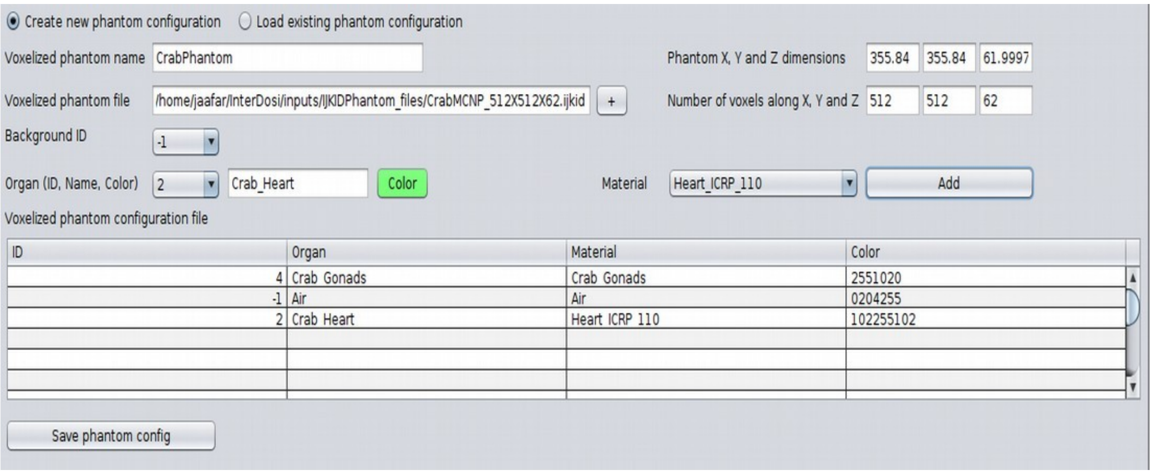
```
**************************************************************
 Geant4 version Name: geant4-10-06 [MT]   (6-December-2019)
  << in Multi-threaded mode >>
                   Copyright : Geant4 Collaboration
                  References : NIM A 506 (2003), 250-303
                             : IEEE-TNS 53 (2006), 270-278
                             : NIM A 835 (2016), 186-225
                         WWW : http://geant4.org/
**************************************************************
```

This GUI application has the following components:

2.4.3.1 Voxelized Phantom builder

This feature allows user to create a new voxel-based phantom configuration or simply load an existing configuration. If the first choice is selected, user need to set some parameters that include:

- ➔ Voxelized phantom's name.
- ➔ Full path of file describing a voxelized phantom having a file extension of ".ijkid".
- ➔ Voxelized phantom dimensions along x, y and z axis (in millimeter).
- ➔ Number of voxels along x,y and z axis.
- ➔ The ID number associated to the background medium.
- ➔ ID numbers associated to all selected organs and the associated names, materials and colors.



● Create new phantom configuration ○ Load existing phantom configuration

Voxelized phantom name: CrabPhantom

Phantom X, Y and Z dimensions: 355.84 | 355.84 | 61.9997

Voxelized phantom file: /home/jaafar/InterDosi/inputs/IJKIDPhantom_files/CrabMCNP_512X512X62.ijkid

Number of voxels along X, Y and Z: 512 | 512 | 62

Background ID: -1

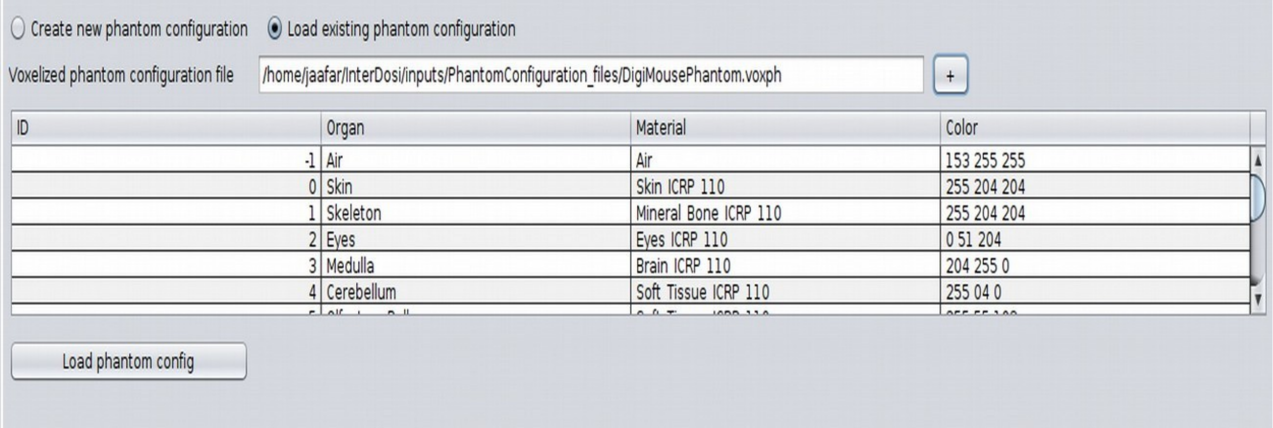
Organ (ID, Name, Color): 2 | Crab_Heart

Material: Heart_ICRP_110

Voxelized phantom configuration file

ID	Organ	Material	Color
4	Crab Gonads	Crab Gonads	2551020
-1	Air	Air	0204255
2	Crab Heart	Heart ICRP 110	102255102

If second choice is selected by user, the only what to do is to load a voxelized phantom configuration file which is previously created and having a file extension of "*.voxph*". When user clicked on "Load phantom config", as shown in below figure, all phantom data will dump to a table having four columns namely: ID of organ, name of that organ, material filled by that organ, and color value used to display geometry of that organ.



○ Create new phantom configuration ● Load existing phantom configuration

Voxelized phantom configuration file: /home/jaafar/InterDosi/inputs/PhantomConfiguration_files/DigiMousePhantom.voxph

ID	Organ	Material	Color
-1	Air	Air	153 255 255
0	Skin	Skin ICRP 110	255 204 204
1	Skeleton	Mineral Bone ICRP 110	255 204 204
2	Eyes	Eyes ICRP 110	0 51 204
3	Medulla	Brain ICRP 110	204 255 0
4	Cerebellum	Soft Tissue ICRP 110	255 04 0

2.4.3.2 AF & SAF calculations

This tab allows user to specify all parameters required by internal dosimetric calculations, including:

➔ Name of a source organ.

➔ Type of a primary particle.

➔ Kinetic energy of a selected primary particle.

➔ Number of threads.

➔ Number of primary events per thread.

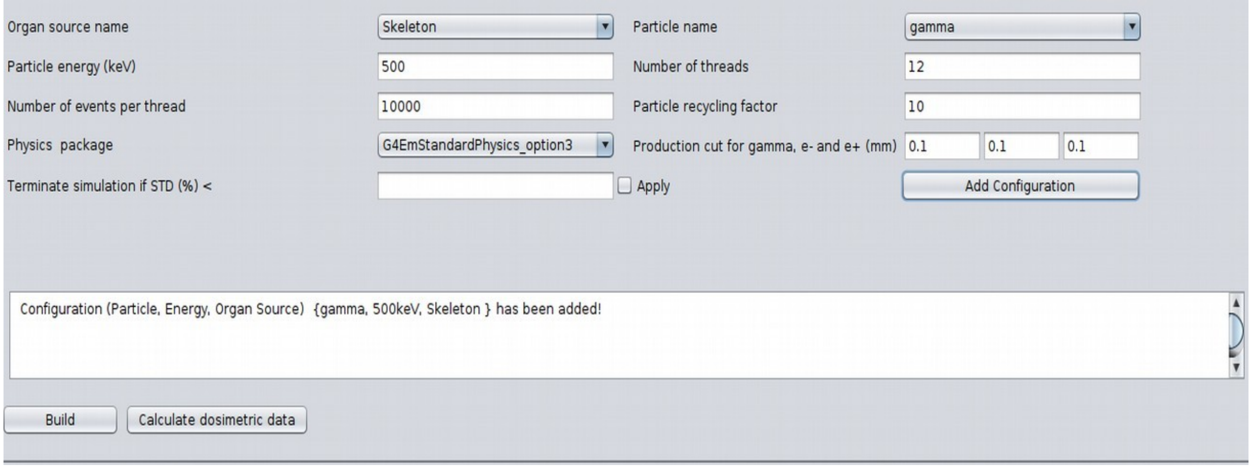
➔ Physics package.

➔ Production cuts for secondary particles (electron, positron and photon).

➔ Primary particle recycling factor (a variance reduction technique for enhancing simulation statistics).

➔ If selected, the maximum value of statistical errors corresponding to the calculated dosimetric quantities.

Here in the below figure, what may this graphical interface look like:



Organ source name: Skeleton
Particle name: gamma
Particle energy (keV): 500
Number of threads: 12
Number of events per thread: 10000
Particle recycling factor: 10
Physics package: G4EmStandardPhysics_option3
Production cut for gamma, e- and e+ (mm): 0.1 0.1 0.1
Terminate simulation if STD (%) <
□ Apply

Configuration (Particle, Energy, Organ Source) {gamma, 500keV, Skeleton } has been added!

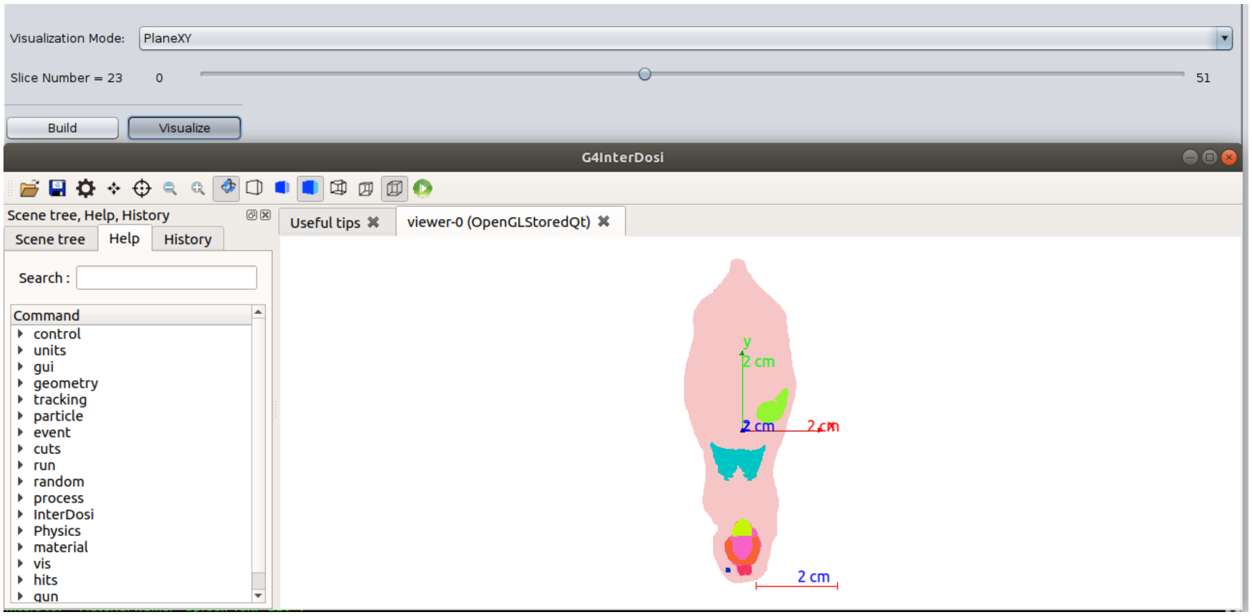
It should be emphasized here, that this code supports the multi-configuration feature. That means that user can define many configurations as he wants and the next dosimetric calculation will be started until the previous one was finished, this approach can save a lot of time, since all required configurations can be set all at once.

2.4.3.3 Visualization

This tab let user see geometrical representation of an used voxel-based phantom, this into various modes of visualization:

- ➔ Phantom geometry model in an XY plane (user should define Z slice value).
- ➔ Phantom geometry model in an XZ plane (user should define Y slice value).
- ➔ Phantom geometry model in an YZ plane (user should define X slice value).
- ➔ 3D phantom geometry model.
- ➔ Partial 3D phantom geometry model (user should define start and end values of Z slices).

User needs to define a visualization mode and also a slice number. Use should not forget to click on "*Build*|" button before calling the visualization option. Here in the below figure a snapshot of the 3D geometry visualization tool.



Visualization Mode: PlaneXY

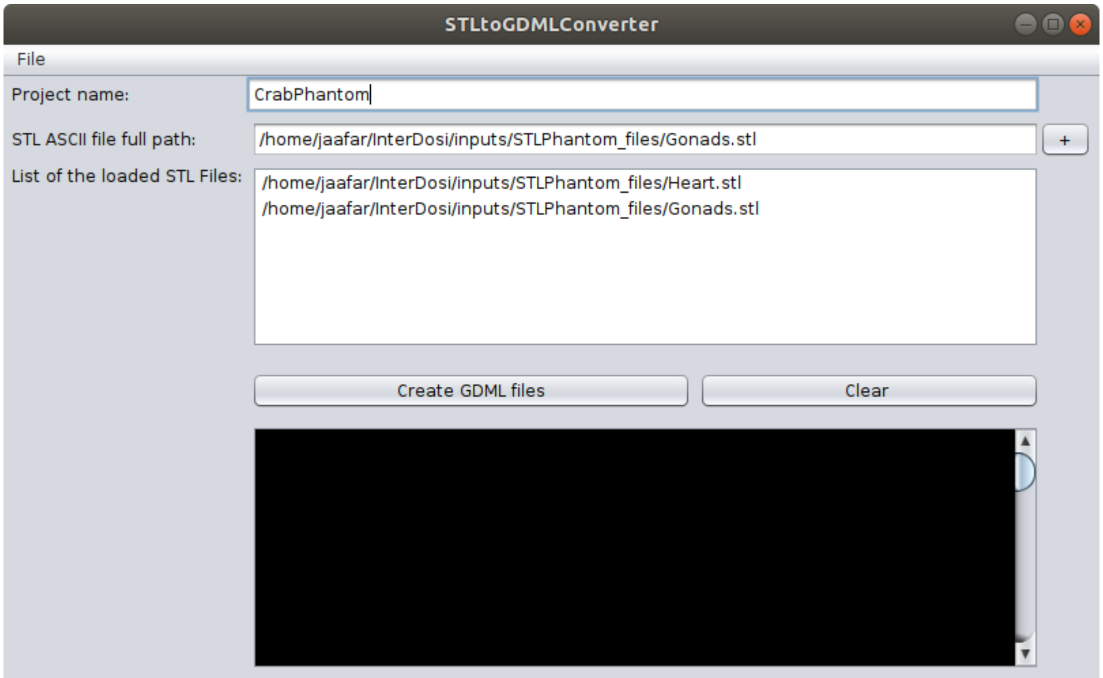
Slice Number = 23

2.4.3.4 Tools

The "tools" menu contains various converters allowing user to export voxel-based phantom files in various formats and converting them into a suitable file format recognized by the present open-source software. Furthermore, this menu contains a dose visualizer tool that rending a 2-dimensional heat map of doses deposited in a specific region of a voxel-based phantom and also a file explorer tool. In an expended detail, these tools are:

1. STL-to-GDML converter:

This tool as seen in the below figure, converts STL files to their equivalent in a GDML markup language. User must fill all parameters required for successful establishing this conversion including: name of current project and a set of full paths of where STL files (only ASCII file format is supported) are located. For more information about STL file format, please do not be lazy to read the section "*2.3.1 STL data format*". Then by clicking on "Create GDML files" the tool will start processing STL data then create suitable analogues in a GDML markup language. It should be noted here that material specification is ignored because this specification will be done during the creation of a configuration file for a voxelized phantom**.** User can get any kind of information belongs an operation that is in progress, this by take a look at the joined console (black text box component).



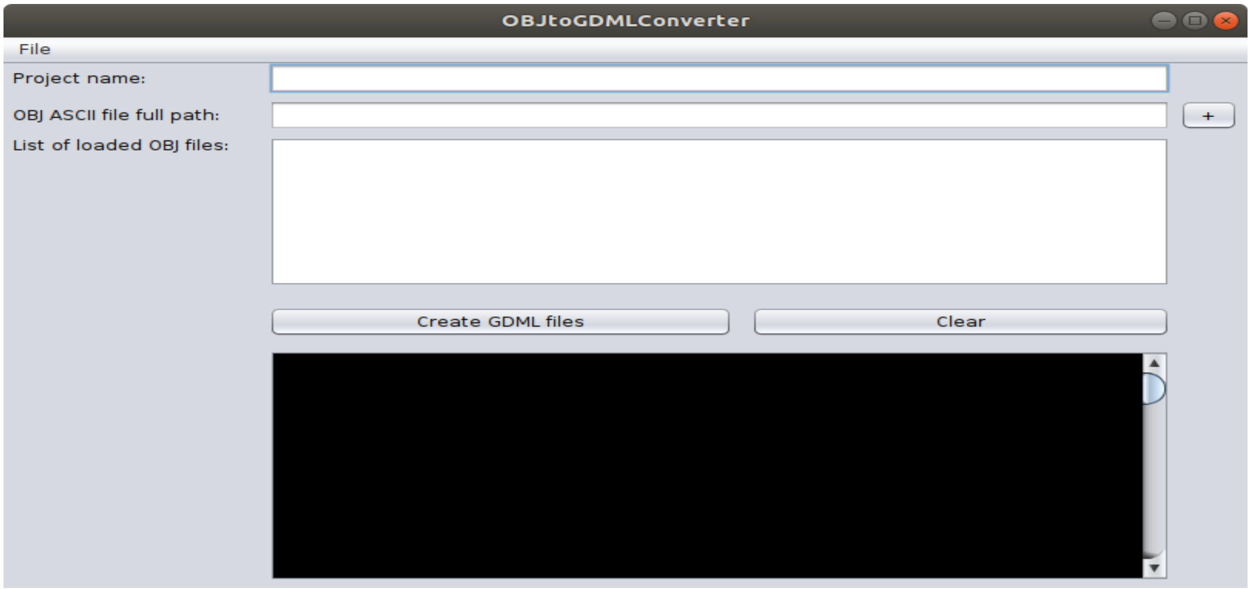
Project name: CrabPhantom

STL ASCII file full path: /home/jaafar/InterDosi/inputs/STLPhantom_files/Gonads.stl

List of the loaded STL Files:
/home/jaafar/InterDosi/inputs/STLPhantom_files/Heart.stl
/home/jaafar/InterDosi/inputs/STLPhantom_files/Gonads.stl

2. **OBJ-to-GDML converter**:

This tool allows user to obtain analogue representations in a GDML markup language, of geometries described in the loaded OBJ files.



To obtain so, user must fill all parameters required for successful establishing this conversion including: name of the current project, and a set of full paths of where OBJ files (only ASCII file format is supported) are located. For more information about OBJ file format, please do not skimp to read the section "*2.3.2 OBJ data format*". By clicking on "*Create GDML files*" the tool will start treat and extract OBJ data then create suitable analogues in a GDML markup language. It should be noted here that material specification is also ignored here because this specification will be performed at the time of creation a voxel-based phantom configuration. The joined console (black text box component) lets user to get in real time a high verbosity level of information are being delivered by any task that is in progress. The text on console can be cleaned by simply clicking on "*clean*" button.

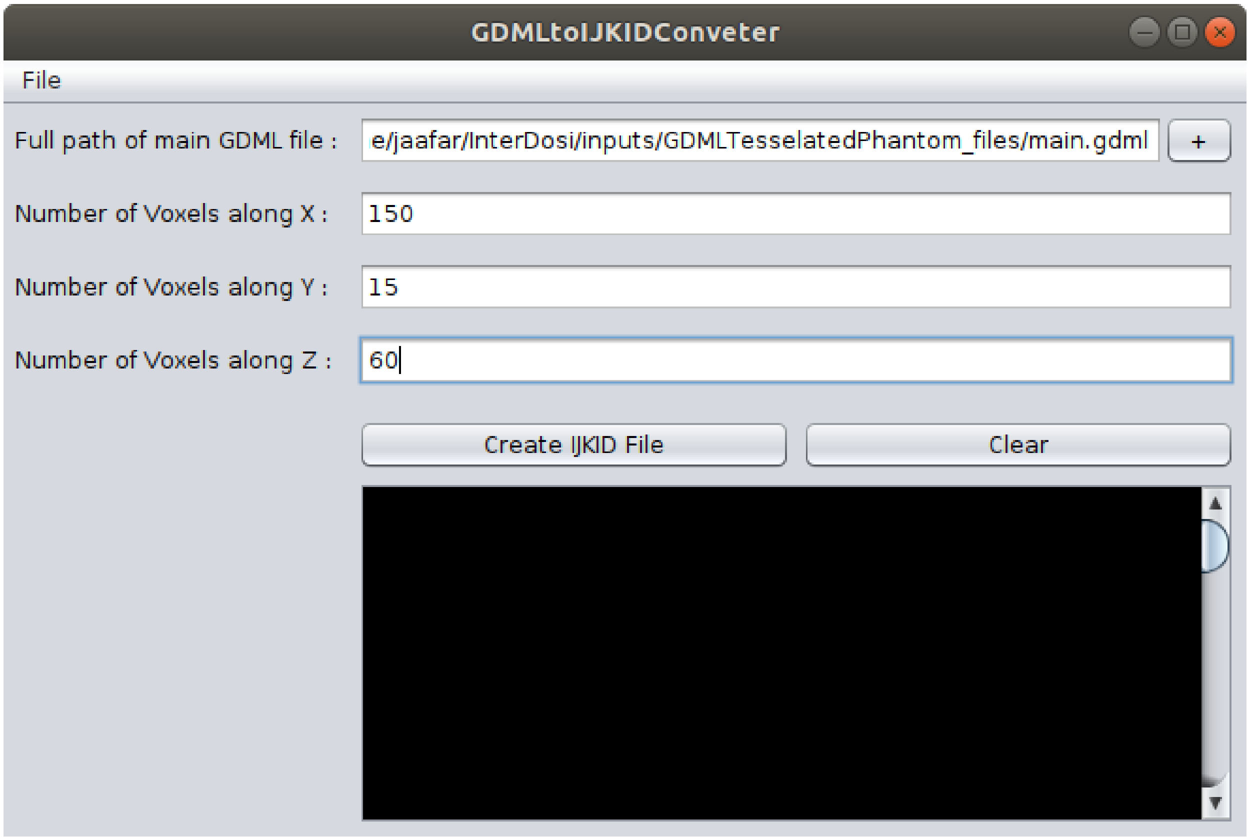
3. **GDML-to-IJKID converter:**

This tool allows user to create a voxel-based phantom file from tessellated geometries described in GDML files. All what needs user to set are:

- ➔ Full path to a main GDML file,
- ➔ Number of voxels along x-axis,

- ➔ Number of voxels along y-axis,
- ➔ Number of voxels along z-axis.

By clicking on "*Create IJKID file*" the tool will start treating GDML data then create a corresponding IJKID file having a IJKID file format. Deep information about IJKID file format can be obtained form section "*2.3.4 IJKID data format*". During the execution of this task, a console in the bottom will deliver at real time some information of what job is in progress.



Full path of main GDML file : e/jaafar/InterDosi/inputs/GDMLTesselatedPhantom_files/main.gdml

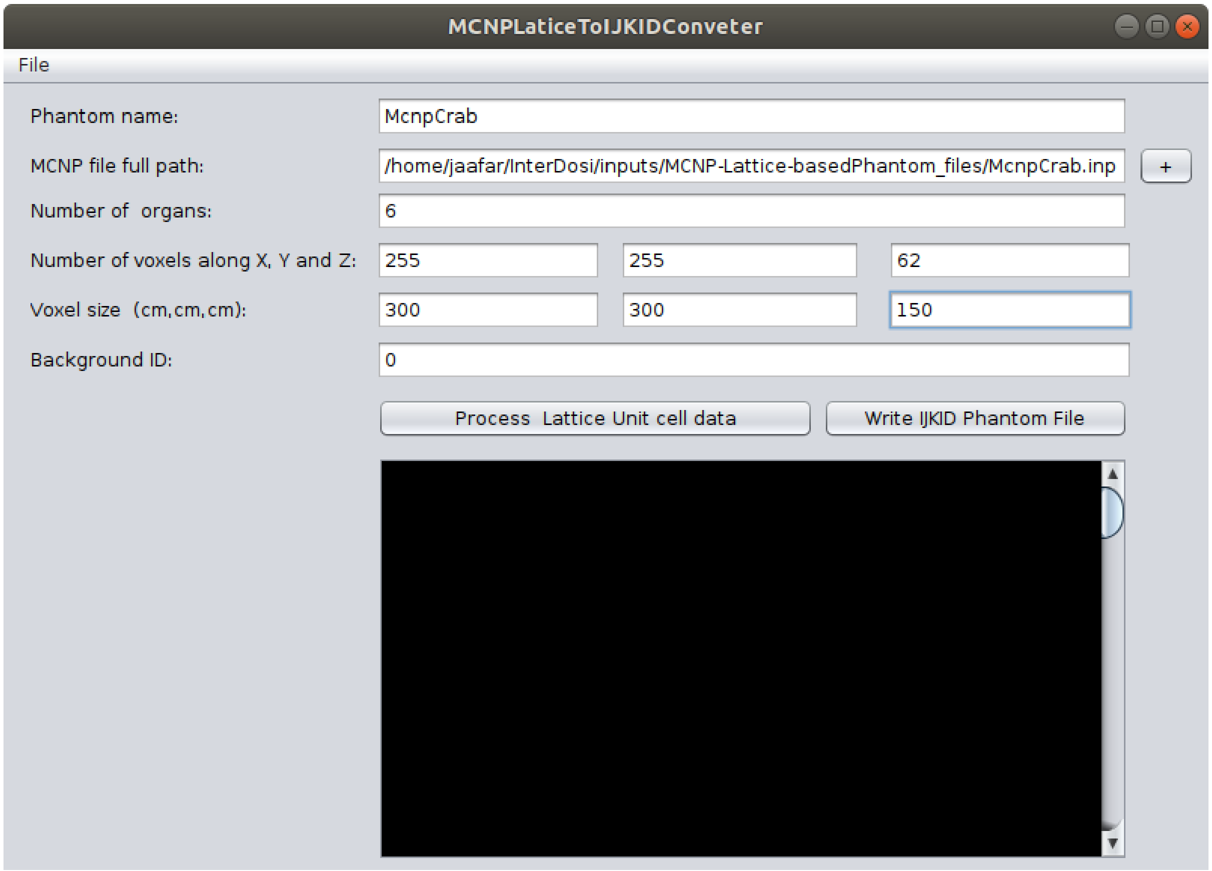
Number of Voxels along X : 150

Number of Voxels along Y : 15

Number of Voxels along Z : 60

4. **MCNP-Lattice-To-IJKID converter**:

This tool allows user to create a voxel-based phantom file from an MCNP input file. Only the hexahedral lattice is supported which can be used to create a voxel-based model assembled by a set of cubic volume filled with difference materials. User should give: a voxel-based phantom name, full path that points out to the emplacement of the used MCNP input file, total number of simulated organs, number of voxels along each axis, voxel size in centimeter, an ID number that reflects a background area (volume that not necessary to be viewed by Geant4 visualization system). By clicking on “*Process Lattice Unit Cell data*”, the tool will analysis the given MCNP input data and extract all necessary geometry data (material data are ignored). The extracted data will used to construct an IJKID phantom file this is done by clicking on “*Write IJKID Phantom files”*. This tool also provides a console in which information about current job are shown in a real time.



Phantom name: McnpCrab

MCNP file full path: /home/jaafar/InterDosi/inputs/MCNP-Lattice-basedPhantom_files/McnpCrab.inp

Number of organs: 6

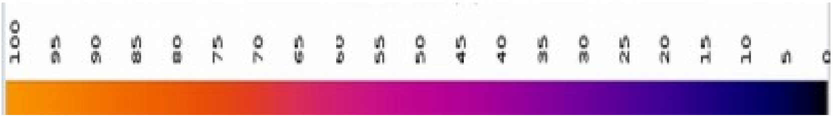
Number of voxels along X, Y and Z: 255 255 62

Voxel size (cm,cm,cm): 300 300 150

Background ID: 0

5. Dose-HeatMap Visualization:

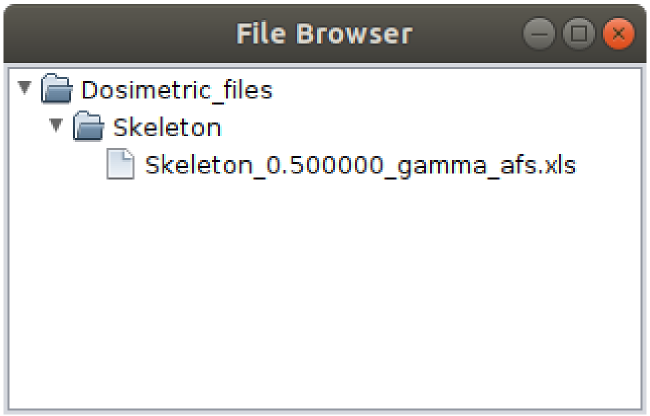
This tool allows user to visualize in a color-based technique called heatmap, absorbed doses distribution on a user-defined axial, sagittal or coronal slice. This can be useful for user to quickly analyze the obtained data by having a visual assessment. It well-known that the heatmap concept is based on representing data with different values as different colors. There are some common gradients of colors that can be used for drawing a heatmap, we cite: seven-color, five-color, mono-chrome and two-color gradients. This last one is used as default heatmap gradient by the InterDosi code. The figure below shows a two-color heatmap gradient used by the code which has two extremities: low value or colder (black color) and high value or warner (orange color).



User can save the obtained dose heatmap image, this by left clicking on a mouse button then a pop-up menu will be displayed then user should select "*Save as → PNG*".

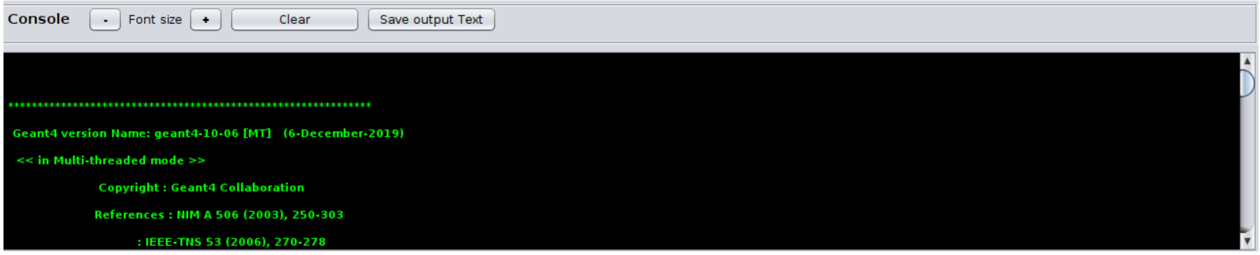
6. FileExplorer:

This tool allows user to see all dosimetric files (in an excel file format) created during Monte Carlo internal dose calculations. By clicking on a desired source organ name, an excel file having a name of a given configuration will be appeared, which can be automatically opened this by double clicking on. Here below a snapshot of this tool is given.



2.4.3.5 Console

Console component gives to user a lot of information about what is being executed in a real time in the background. Indeed, all kinds of messages related to the compilation and execution of a Geant4-based code are shown in a high verbosity level. In addition all types of messages delivered by all operations performed by the developed tools are consecutively written to this console. Data relating to the used voxelized phantom are also figured out on the console. The output text of the console can be saved to any directory determined by the user and it can be also cleaned up. For an adaptive vision, user can set the font size of the text on the console as he pleased. Here below a snapshot of this console is displayed.



```
**************************************************************

 Geant4 version Name: geant4-10-06 [MT]   (6-December-2019)
  << in Multi-threaded mode >>
                      Copyright : Geant4 Collaboration
                     References : NIM A 506 (2003), 250-303
                                : IEEE-TNS 53 (2006), 270-278
```

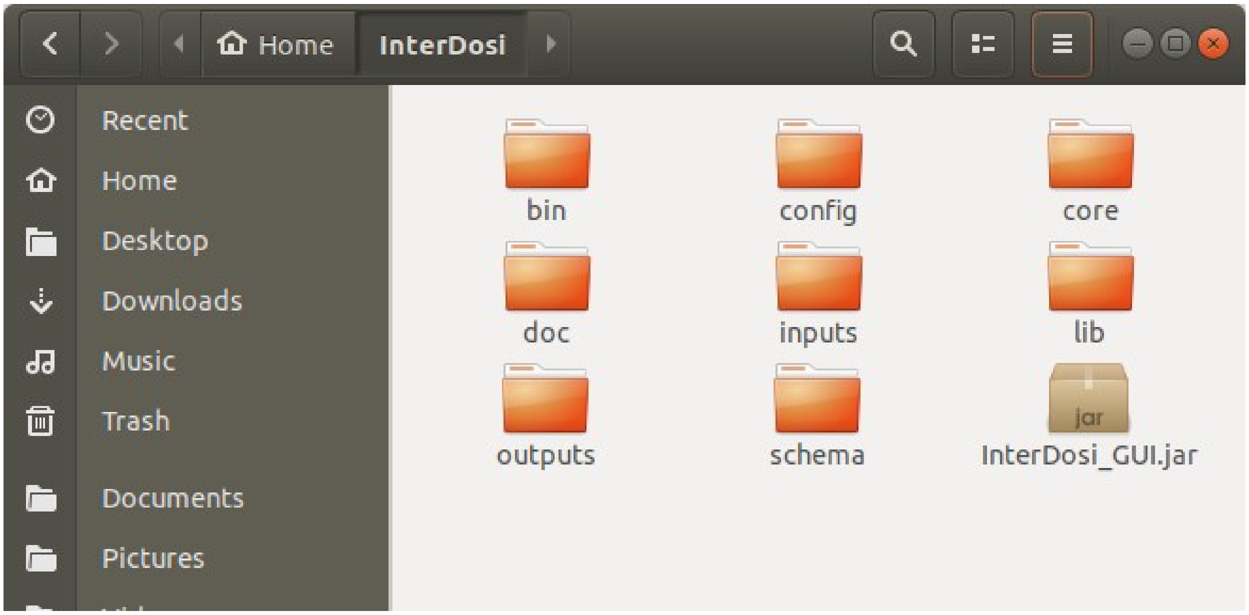
3.0 GETTING STARTED WITH INTERDOSI

3.1 Getting InterDosi

InterDosi is hosted on Github and it can be freely obtained from the following link:

> https://github.com/EL-Bakkali-Jaafar/InterDosi/releases

By clicking on *"clone or download"* button, anyone can get a source copy of InterDosi code. The compressed file should be unzipped by any known unzipper tool, then user may see some files and folders as are shown in the below figure. Full description of all of files and folders can be read from the section "*2.2 Software files and folders*".



InterDosi has been successfully tested in Ubuntu 16 and 18, but it should work with other Linux Distribution as well. InterDosi has been tested on a laptop {Intel(R) Core(TM) i3-2328M CPU @ 2.20GHz, OS TYPE : 64 bit Ubuntu 18.04 LTS}, and also on a Lenovo Workstation {Intel ® Xenon(R) CPU W3690 @ 3.47 GHZ x 12, OS TYPE : 64 bit Ubuntu 16.04 LTS}.

3.2 Prerequisite software and libraries

InterDosi was built on the top of the Geant4.10.05 code (also successfully tested on Geant4.10.06), so users must have installed this Monte Carlo package in their operating systems. The source of the Geant4.10.05 code can be freely downloaded from the following link:

```
http://geant4-data.web.cern.ch/geant4-data/releases/geant4.10.05.tar.gz
```

Users must also download all required data or tools and must follow Geant4 installation guide. Then they should compile Geant4 by using cmake tool (version 3.13), this is assured, by inserting the following command into a Linux terminal:

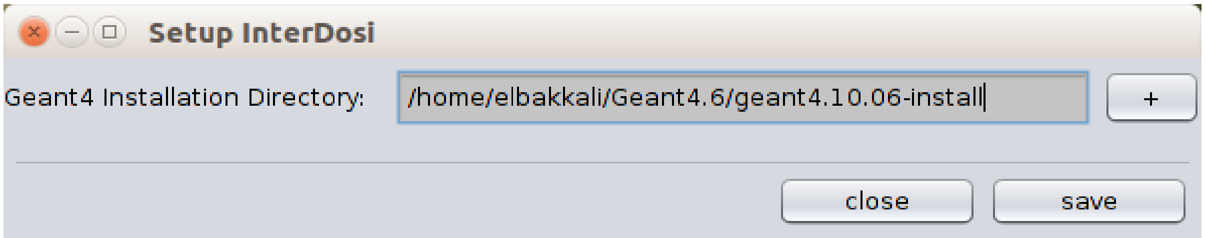
```
cmake  -DGEANT4_BUILD_MULTITHREADED=ON -
DGEANT4_USE_SYSTEM_EXPAT=OFF -DGEANT4_USE_QT=ON -
DGEANT4_USE_OPENGL_X11=ON
-DCMAKE_INSTALL_PREFIX=/home/user/Geant4/geant4.10.05-install
/home/user/Geant4/geant4.10.05
```

It should be noted here, that, for a custom installation of data files, the mandatory ones are as follows:

G4NEUTRONXS1.4: data files for evaluated neutron cross-sections on natural composition of elements.
G4EMLOW7.3 : data files for low energy electromagnetic processes.
G4PhotonEvaporation5.2 :data files for photon evaporation.
G4SAIDDATA1.1: data files from evaluated cross-sections in SAID data-base.
G4ABLA3.1: data files for nuclear shell effects in INCL/ABLA hadronic mode.
G4NEUTRONHPDATA : Neutron data files with thermal cross-sections.
G4ENSDFSTATE2.2 : data files for nuclides properties.

3.3 Linking InterDosi to an existing installation of Geant4

InterDosi don't require any installation, instead that it should be properly configured, this may done by double clicking on *"InterDosi_GUI.jar"* Java-based application and by choosing from principal menu the sub-menu called *"Settings → Setup InterDosi"*. A small Java form shown in below figure will be appeared.



Geant4 Installation Directory: /home/elbakkali/Geant4.6/geant4.10.06-install

User must set the path that points out to the the Geant4 installation location. It should be emphasized here, that InterDosi_GUI is a Java-based application and requires a Java 8 Runtime to be appropriately run, which can be easily installed from Linux terminal, this by typing the following command:

```
sudo apt-get install openjdk-8-jre
```

3.4 InterDosi phantom configuration file tips

User should pay intention when he is trying to use one of the offered phantom configuration files of an extension ".voxph", this by clicking on *"Load existing phantom configuration"*, especially for the parameter *"@VoxelizedPhantomFileDir"* the value must be pointed out to a full path directory where the voxel-based phantom file is localized. If user plans to copy InterDosi main folder to another emplacement, he must also update this parameter by giving a full path of where phantom configuration file is existed so.

3.5 Adding a new material to the InterDosi material library

As it is well-told about through this document, InterDosi offers to user a high flexibility to interact with, as many data and parameters can be changed in an easy way. Indeed, if user can not find a material in the list of provided materials, he can without doubt create a definition of this new material and save it to a text file having an extension of *".mat"*, this by following rules described in the section "*2.3.6 MAT data format*". After that, the InterDosi_GUI Java-based application will detect automatically this new material and will add it the list of available materials which can be seen by selecting "*Voxelized Phantom Builder → Create new phantom configuration*" tab. User can also change data of any existing material, this by editing the desired MAT file using any known text editor as "*gedit*" and he should not forget to respect writing rules of Mat files.

3.6 Import and configure a new voxelized phantom

In this section detailed steps on how to incorporate an existing voxel-based phantom into the InterDosi code are given for novice user. As demonstrative example, the Digimouse voxel-based phantom is taken so. The required steps are as follows:

- ✓ What should user to do firstly, is to go to the download page and download Atlas Digimouse compressed file: http://neuroimage.usc.edu/mouse_atlas/atlas.zip
- ✓ After unzipping the downloaded file, user should see three files: two analyze files: data file named "*atlas_380x992x208.img*" and header file named "*atlas_380x992x208.hdr*". The third file is an organ index library file called "*atlas_380x992x208.txt*", which contains a set of ID numbers associated to each organ assembling the whole body of Digimouse phantom. Technically, the voxelized phantom contains a matrix of 380 × 992 × 208 elements of voxels of 0.1 mm × 0.1 mm × 0.1 mm dimensions.
- ✓ After that, user should download and install ImageJ2 a Java-based image processing software and install also a plug-in called "*analyze*".
- ✓ Next, user should import the Atlas Digimouse image file (.hdr & .img) and preferably resize the contained matrix to 190 × 496 × 52 elements that will consequently increase voxel dimensions to 0.2 mm × 0.2 mm × 0.4 mm, which can in the fact reduce RAM memory that is required to be allocated for loading this voxel-based phantom. If user dispose a computer of RAM memory greater than 8 GB, then he can keep the original phantom size.

✓ The resized image file must be visualized by ImageJ2 visualization tool and exported into XY Coordinates image file format by clicking on the following ImageJ2 tool: "*Analyze →Tools → Save XY Coordinates*".

✓ Then user should rename the obtained file extension to *".ijkid"* and copy the file to the *"InterDosi/inputs/IJKIDPhantom_files"* input directory.

✓ Then, user should open InterDosi_GUI Java-based application and select *"Voxel Phantom Builder"* tab and then choice *"create new phantom configuration"* option.

✓ Next, user must provide all specific data associated to the voxelized phantom, including: 1) Voxelized phantom name. 2) Full path to a IJKID file. 3) Voxelized phantom dimensions along x, y and z axis. 4) Number of voxels along x,y and z axis. 5) The ID number associated to a background medium. 6) ID numbers of all selected organs and the associated names, materials and also colors for displaying those organs.

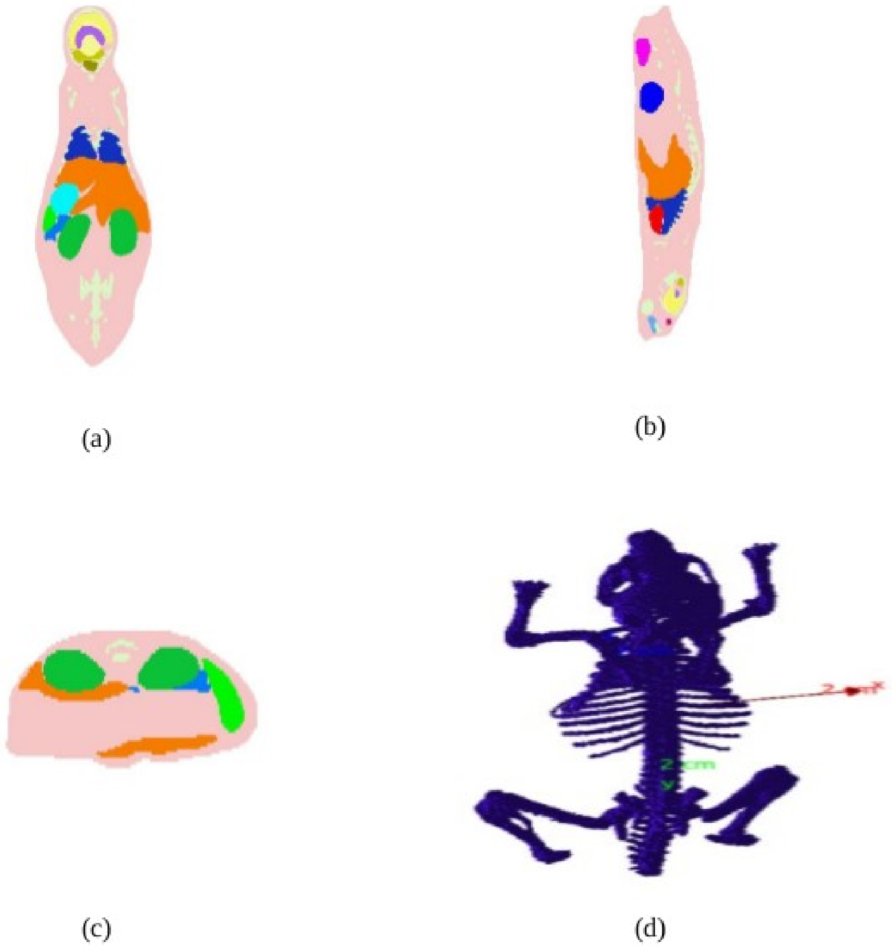
✓ Finally by clicking on "*Save phantom config"*, a configuration file for a new voxel-based phantom will be created and put into *"InterDosi/inputs/PhantomConfiguration_files"* input directory.

✓ Finally, user can use the created voxelized phantom, what he needs to do is only load the created configuration file of an extension of ".voxph", this by clicking on "*Load phantom config*". Now the voxel-based phantom is ready to be used for calculating internal dosimetric data.

4.0 EXAMPLE OF MODELING OF SOME HUMAN AND BIOTA VOXELIZED PHANTOMS

4.1 Mouse Phantom

As it is well-known, the development of new radiopharmaceuticals was majorly carried out on a mouse. From current literature, we can find a number of developed mathematical models describing various internal organs of a mouse called phantoms, with particular attention the Digimouse model which presents our interest. This modern computational voxelized phantom was developed by biomedical imaging group of the University of South California. In the below figure, coronal (a), sagittal (b), transversal (c) graphical representations mouse and 3D rending of mouse skeleton are shown.


(a) (b) (c) (d)

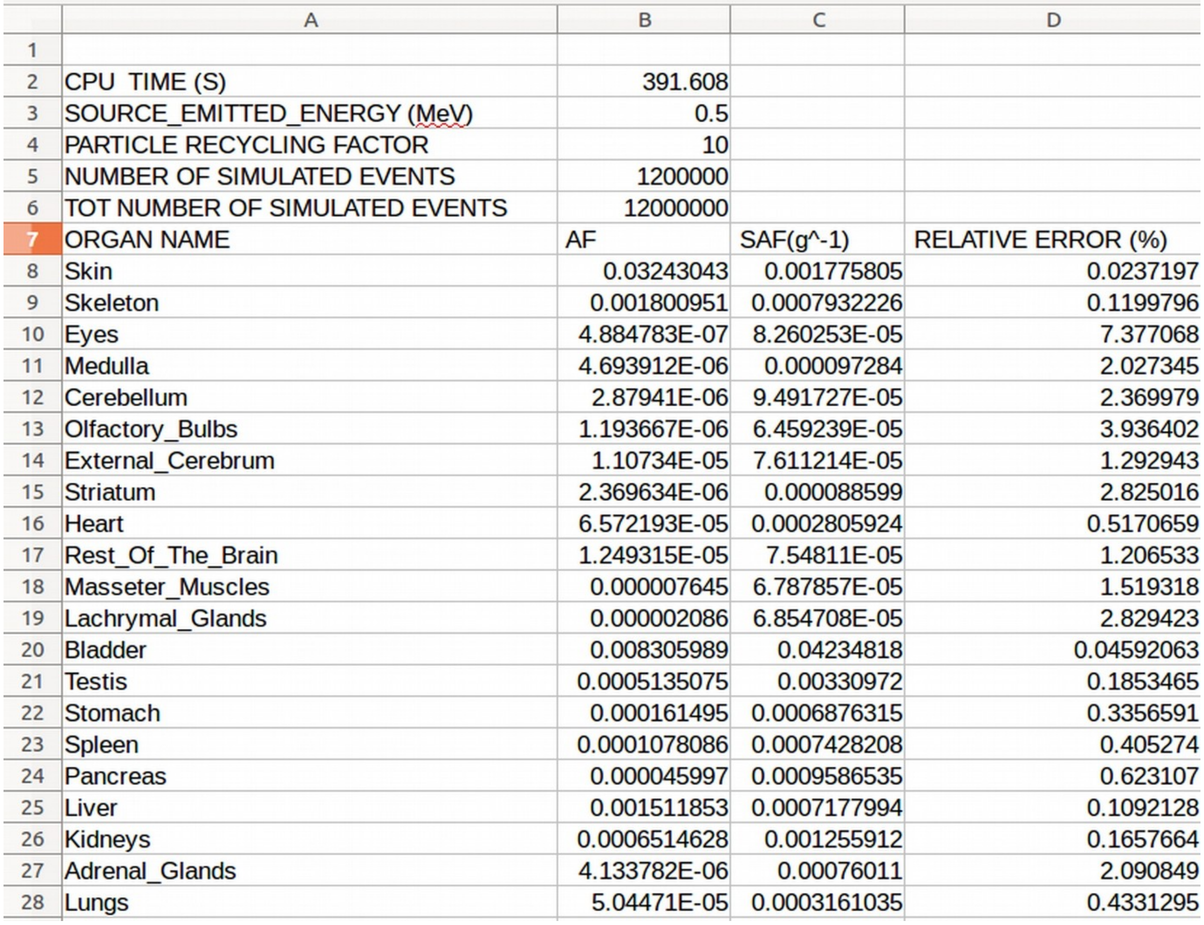
In order to show a basic example that can be performed by InterDosi code for assessing internal doses on a mouse, herein some steps required for calculating photon SAFs in various mouse organs when mouse bladder is chosen as a source of monoenergitic photons of 500 keV, emitted isotropically and distributed uniformly within bladder volume:

- ✓ In the tab named "*Voxelized Phantom Builder*" user should select "*load existing phantom configuration*" option, choice "*DigimousePhantom.voxph*" file and, click on "*load phantom config*" button.
- ✓ In the tab named "*AF & SAF Calculation*", user should fill parameters with the following values:

Parameter	***Value(s)***
Organ source name	*Bladder*
Particle energy (KeV)	*500*
Particle name	*gamma*
Particle recycling factor	*10*
Secondary Particle production cuts (mm)	*0.1 0.1 0.1*
Number of Threads	*12*
Number of Event per Thread	*100000*
Physics Package	*G4EmStandardPhysics_Option3*
Number of Event per Thread	*100000*
sigmax and sigmay (cm, cm)	*0.05 0.05*
cut gamma, cut e- and cut e+ (mm, mm, mm)	*1 1 1*

- ✓ Then, to launch internal dosimetric calculations, only to do is to click on "*build*" and " *calculate dosimetric data*" buttons.

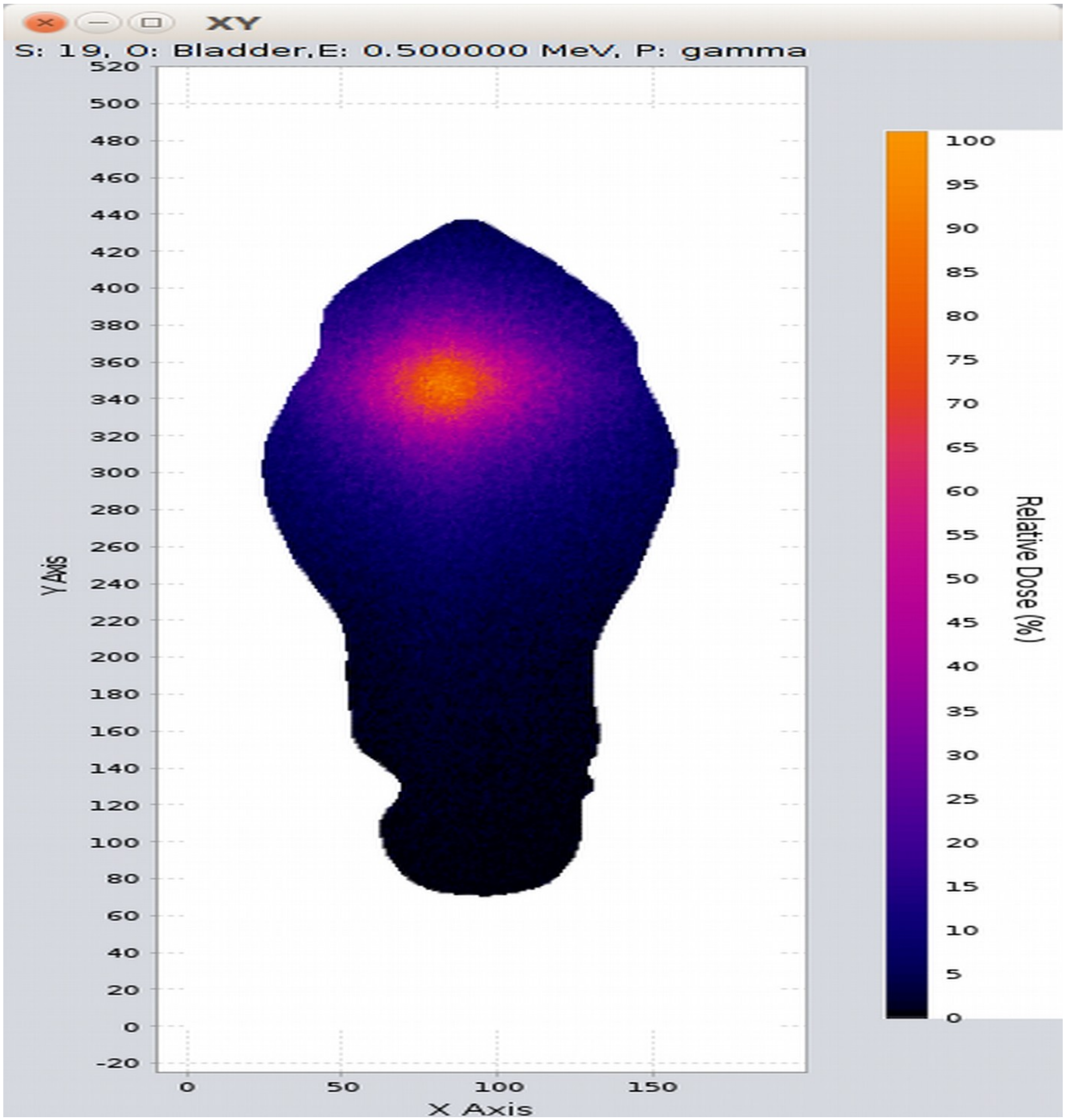
Once the calculations had been finished, an excel file called "*Bladder_0.500000_gamma_afs.xls*" will be produced which contains dosimetric data and it is located at “*/outputs/dosimetric_files/bladder*” outputs directory. Here below, an example of content of this excel file:



	A	B	C	D
1				
2	CPU TIME (S)	391.608		
3	SOURCE_EMITTED_ENERGY (MeV)	0.5		
4	PARTICLE RECYCLING FACTOR	10		
5	NUMBER OF SIMULATED EVENTS	1200000		
6	TOT NUMBER OF SIMULATED EVENTS	12000000		
7	ORGAN NAME	AF	SAF(g^-1)	RELATIVE ERROR (%)
8	Skin	0.03243043	0.001775805	0.0237197
9	Skeleton	0.001800951	0.0007932226	0.1199796
10	Eyes	4.884783E-07	8.260253E-05	7.377068
11	Medulla	4.693912E-06	0.000097284	2.027345
12	Cerebellum	2.87941E-06	9.491727E-05	2.369979
13	Olfactory_Bulbs	1.193667E-06	6.459239E-05	3.936402
14	External_Cerebrum	1.10734E-05	7.611214E-05	1.292943
15	Striatum	2.369634E-06	0.000088599	2.825016
16	Heart	6.572193E-05	0.0002805924	0.5170659
17	Rest_Of_The_Brain	1.249315E-05	7.54811E-05	1.206533
18	Masseter_Muscles	0.000007645	6.787857E-05	1.519318
19	Lachrymal_Glands	0.000002086	6.854708E-05	2.829423
20	Bladder	0.008305989	0.04234818	0.04592063
21	Testis	0.0005135075	0.00330972	0.1853465
22	Stomach	0.000161495	0.0006876315	0.3356591
23	Spleen	0.0001078086	0.0007428208	0.405274
24	Pancreas	0.000045997	0.0009586535	0.623107
25	Liver	0.001511853	0.0007177994	0.1092128
26	Kidneys	0.0006514628	0.001255912	0.1657664
27	Adrenal_Glands	4.133782E-06	0.00076011	2.090849
28	Lungs	5.04471E-05	0.0003161035	0.4331295

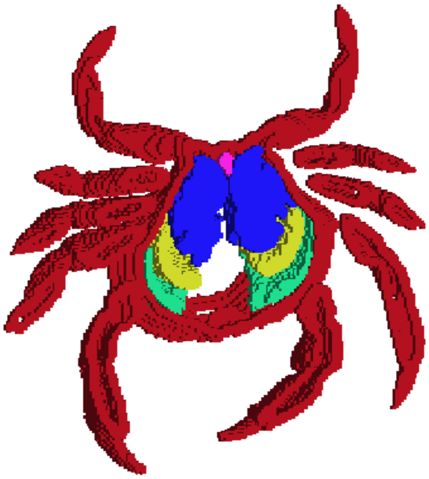
As seen above, this file keeps interesting information about dosimetric data calculation including: whole CPU time spent by the simulation, kinetic energy of source organ, particle recycling factor (PRF) if the PR technique is applied, sum of number of simulated primary events of each thread (SUM_PeThread) and total number of simulated primary events (numerically = SUM_PeThread x PRF), and four data for each simulated organ: organ name, AF, SAF and relative error.

To visualize dose heat map of a 500 keV gamma inside mouse body, user need to click on “*Tools → Dose-Heat-Map Visualizer*” menu and select “ *Bladder_0.5000_gamma.DoseInXYZ*” file, then user should select a visualization mode (for example XY) and a slice number (for example 19), by clicking on “*view*” button the program will produce the following image:



4.2 Crab Phantom

To well-understand the effects of exposures to ionizing radiations on conservation of marine biodiversity, assessment of impacts of ionizing radiations on a crab organs and also on it sensitive life stage, can be a practical approach and of great interest. Indeed, a crab is one of aquatic species used as a biological indicator to asses pollution level of the marine environment. From current literature, we can find only one voxel-based model of crab which was early developed by Caffrey and Higley. Herein a 3D model of a crab reference phantom created by InterDosi program.



To calculate internal dosimetric quantities in a voxelized crab phantom for a particular configuration, user should follow these steps:

- ✓ In the tab named "*Voxelized Phantom Builder*" user should select "*load existing phantom configuration*" option, choice "*VoxelizedCrabPhantom.voxph*" file and click on "l*oad phantom config*" button.
- ✓ In the tab named "*AF & SAF Calculation*", user should fill desired parameters and click on "*build*" and "*calculate dosimetric data*" buttons in order to start dosimetric calculations.

The program will produce a dosimetric file in an excel format named as "*OrganSourceName_KineticEnergyInMeV_ParticleName_afs.xls*" and it is localized at "*/outputs/dosimetric_files/OrganSourceName*" outputs directory. The excel file can be opened by choosing "*View → FileExplorer*" and by double clicking on the name of the desired excel file.

Similarly, user can visualize the geometry of a crab phantom or dosimetric data distribution as described in the section "*4.1 Mouse Phantom".*

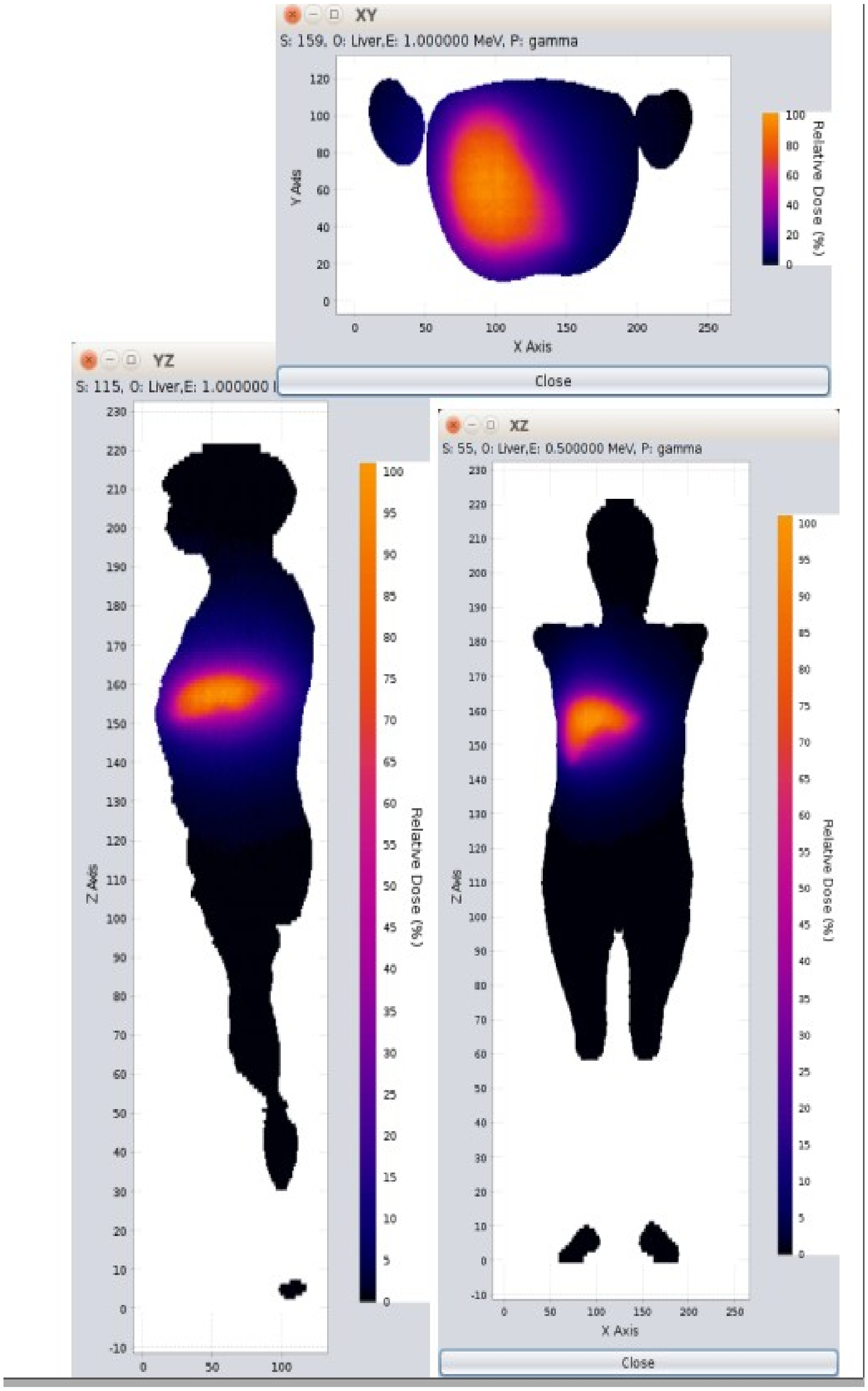
4.3 ICRP Human Male Phantom

To calculate internal dosimetric quantities in a voxelized ICRP Human Male phantom for a particular configuration, user should follow these steps:

- ✓ In the tab named "*Voxelized Phantom Builder*" user should select "load existing phantom configuration" option and choice "ICRP_adult_male.voxph" file and click on "*load phantom config*" button.
- ✓ In the tab named "*AF & SAF Calculation*", user should fill desired parameters and click on "*build*" and "*calculate dosimetric data*" buttons in order to start dosimetric calculations.

The program will produce a dosimetric file in excel format named as "*OrganSourceName_KineticEnergyInMeV_ParticleName_afs.xls*" localized at "/outputs/dosimetric_files/OrganSourceName" outputs directory.

Figure below shows transversal, coronal, and sagittal graphical representations of relative absorbed doses two-dimensional distribution (dose heat map). This example is about a configuration of the following parameters: liver (source organ), gamma (primary particle), 1 MeV (Kinetic energy).



XY

S: 159, O: Liver,E: 1.000000 MeV, P: gamma

YZ

S: 115, O: Liver,E: 1.000000 I

XZ

S: 55, O: Liver,E: 0.500000 MeV, P: gamma

5.0 TROUBLESHOOTING

5.1 Some common errors

5.1.1 Segmentation fault

Beside that InterDosi code has been built as well as it is possible to respect to the user-friendly concept, however some errors can be generated due to bad values associated to parameters passed to the code. Some common error committed by user is by entering wrong path to the IJKID phantom which needs to be written by hand. Error message can be like this one:

```
./Merge_outputsMerging of dosimetric data from threads have been successfully terminated !
./start_calculations.sh: line 1:  3465 Segmentation fault      (core dumped) ./G4InterDosi /home/
jaafar/InterDosi/bin/g4interdosi.bin/calculate_af_Shell_gamma_150.mac
```

To deal with this issue, please see the section "*3.4 InterDosi phantom configuration file tips*".

5.1.2 G4InterDosi executable does not exist

Sometime user can delete files belongs "/bin" directory, which means that G4InterDosi executable is no longer exist, this can affect the launch of dosimetric calculations and there is no program that can do that, consequently user can get like the following error message:

```
/home/jaafar/InterDosi/core/scripts/calculate_dosimetric_quantities.sh: line 9: cd: g4interdosi.bin: No
such file or directory
```

To resolve this problem, only to do is by rebuilding G4InterDosi Geant4-based code, this by clicking on "build" button belongs the tab "AF&SAF calculations".

5.1.3 Simulation fails and an exception is generated

When user try to create and fill by hand a phantom configuration file of an extension of VOXPH , he should pay a strong attention since the part of InterDosi code responsible for reading the file with that format is sensitive to blank spaces.

For example : if a blank space is detected between values of organ name and material name that belong to the key "*PhantomOrganItem*", the the simulation will stopped and shown the following message error :

```
Warning from G4Material::G4Material() define a material with density=0 is not allowed.
The material  will be constructed with the default minimal density: 1e-25g/cm3
```

To fix this error, user only to do is to remove a blank space. It should be noted here that for each key the associated values are separated by a tab character.

5.2 I need a Help

For any questions, users should contact me at bahmedj@gmail.com

6.0 FOR DEVELOPERS

6.1 How to contribute

C++ advanced programmer who want to contribute in development of this open-source code are welcome !

The C++ code source of InterDosi is available at:

https://github.com/EL-Bakkali-Jaafar/InterDosi

Users whom have some Java programming skills are welcome to improve InterDosi_GUI Java-based application. The Java source code of InterDosi_GUI is available at:

https://GitHub/EL-Bakkali-Jaafar/InterDosi_GUI

7.0 LICENSES

7.1 InterDosi c++ code

InterDosi is an open-source C++ code distributed under the GNU General Public License. Please see:

> https://www.gnu.org/licenses/gpl-3.0.txt

This code uses four external software:

1. Geant4.10.05, developed by the members of the Geant4 collaboration. Please see:

> https://cern.ch/Geant4

7.2 InterDosi_GUI Java-based application

InterDosi_GUI Java-based application was developed under Netbeans 8.0 and distributed under MIT license. Please see:

> https://opensource.org/licenses/MIT