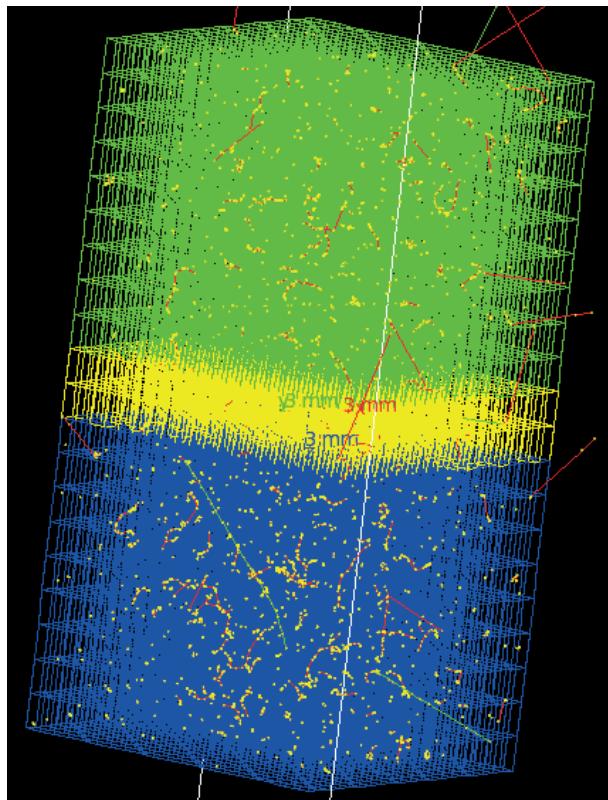


Loïc Martin & Norbert Mercier

DosiVox



Manual

Manual

DosiVox

Loïc Martin & Norbert Mercier

IRAMAT-CRP2A
UMR 5060, CNRS - Université Bordeaux Montaigne
Maison de l'Archéologie, Esplanade des Antilles
33607 Pessac Cedex, France

This document has been created using L^AT_EX.

Initial: March 25, 2015
Last update: February 8, 2019

Contents

1 Principles and functionality	8
1.1 Scope and notes	8
1.2 Detectors	8
1.3 Controlling the simulation	10
2 The Pilot Text File	12
2.1 How to interpret and edit a Pilot Text File	14
2.2 General parameters (header)	14
2.3 Material definition	16
2.3.1 Creating a new component	16
2.3.2 Creating a new material	17
2.4 Geometry and radioactivity of the model	19
3 Running DosiVox	21
3.1 Getting started with DosiVox	21
3.1.1 Open a terminal	21
3.1.2 Basic DosiVox terminal commands	22
3.1.3 Visualization tool commands	22
3.2 A DosiVox run	23
3.2.1 Internals	24
4 Results and data processing	26
4.1 Error file (_Errors suffix)	26
4.2 Probe results (_Probe suffix)	26
4.3 Sub-voxelised voxel results (_Detector1 and results/DoseMapping)	27
4.4 Single and successive grain packing results (_Detector2, _Detector3 and _Grains)	27
5 DosiVox interface for PTF creation and editing	29
5.1 Material creation	29
5.1.1 Filling with materials	29
5.1.2 The case of sediments	30
5.2 Grid size and navigation	31
5.2.1 Defining own geometries	31
5.2.2 Radioactive sources	32
5.3 Detectors and other parameters	32
5.3.1 Detectors	32
5.3.2 3D-Scan loading	33
5.3.3 Save a project and create PTFs	35
5.3.4 Files associated with the radioactive sources	36
5.4 Creating a model using <i>ImageJ</i>	36

6 Technical support	41
7 Acknowledgements	42
References	43
Index	44

List of Figures

1	Example of voxel grids of different sizes	8
2	Example of voxel grid with particle interactions.	9
3	Voxel grid with the probe detector.	10
4	The different solid detectors available	11
5	Screenshot file browser and terminal window	21
6	The visualization tool	23
7	The <i>DosiVox</i> interface.	29
8	The <i>DosiVox</i> interface, Periodic Table	30
9	The sediment information window	31
10	Adjusting the probe area	32
11	Adjusting the solid detector parameters	33
12	The load 3D-scan menu	34
13	Scan 3D-file menu	34
14	The <i>ImageJ</i> menu bar.	37
15	Creating new image in <i>ImageJ</i>	37
16	Selecting tools and changing colors in <i>ImageJ</i>	39
17	3D projection tool in <i>ImageJ</i>	40
18	ASCII file of the resulting 3D model	40

List of Tables

1	Basic component list	17
2	Probe results example	26

Genesis of this software

DosiVox and everything in this manual based on research initiated a long time ago to which contributed, among others and for different aspects of this research, C. Tribolo (Tribolo et al., 2002; Tribolo 2003), S. Incerti (Incerti et al., 2009), G. Guérin (Guérin, 2011; Guérin & Mercier, 2012; Guérin et al., 2012), and Alan Guitard (for the graphical Java interface).

- Tribolo C., 2003. Apport des méthodes de la luminescence à la chronologie de technofaciès du Middle Stone Age associés aux premiers Hommes Modernes du sud de l'Afrique. Thèse de doctorat, Université Bordeaux 1, pp. 239. <http://www.theses.fr/2003BOR12725>
- Tribolo C., Mercier N., Valladas H., Selo M., Joron J-L., Reyss J-L., Henshilwood C., Sealy J., Yates R., 2002. Internal dose rate in quartzites: the case of Blombos cave, South Africa. LED 2002 - 10th International conference on Luminescence and Electron Spin Resonance Dating, June 24–28, 2002, Reno, USA.
- Incerti, S., Seznec, H., Simon, M., Barberet, Ph., Habchi, C., Moretto, Ph. 2009. Monte Carlo dosimetry for targeted irradiation of individual cells using a microbeam facility, Radiation Protection Dosimetry 133 (1), 2–11. doi:10.1093/rpd/ncp003
- Guérin, G., 2011. Modélisation et simulations numériques des effets dosimétriques dans les sédiments quaternaires: application aux méthodes de datation par luminescence. Thèse de doctorat en Physique des archéomatériaux, Bordeaux, Université Michel de Montaigne Bordeaux 3, pp. 242. <http://www.theses.fr/2011BOR30051>
- Guérin, G., Mercier, N., 2012. Preliminary insight into dose deposition processes in sedimentary media on a scale of single grains: Monte Carlo modelling of the effect of water on the gamma dose rate. Radiation Measurement 47 (7), 541–547. doi:10.1016/j.radmeas.2012.05.004
- Guérin, G., Mercier, N., Nathan, R. , Adamiec, G. and Lefrais, Y., 2012. On the use of the infinite matrix assumption and associated concepts: a critical review. Radiation Measurements 47 (9), 778–785. doi:10.1016/j.radmeas.2012.04.004

1 Principles and functionality

1.1 Scope and notes

DosiVox (Martin et al., 2015a,b) is a software based on the *Geant4* (Agostinelli et al., 2003; Allison et al., 2006, 2016) libraries (<http://www.geant4.org>). Because *DosiVox* utilizes these open source libraries, it incorporates their licencing rules (cf. *DosiVox* license). The purpose of *DosiVox* is to allow the modelling of dose distributions in any object or series of objects in a 3D grid, whose axes are hereafter defined as x, y and z. In the following, each element of this grid is called a 'voxel' (Fig. 1).

For a simulation, each voxel must contain information on the material it is constituted of, and on the radio-elements it contains. Once this 3D geometry is filled with materials defined by the user, and contains at least one distribution of radio-elements (usually from the U- and Th-series or from ^{40}K , or any other user-defined radioactive sources – see below), simulations are performed following the physical rules of particle-matter interactions: *Geant4* simulates the passage of particles through matter using a Monte-Carlo algorithm, enabling a calculation of successive interactions and the production of secondary particles whose interactions are also simulated subsequently (Fig. 2). As detailed in Martin (2015): the α -particle spectra (U_α , Th_α) are constructed from the NIST online database (Kramida et al., 2018); the β -particle spectra (U_β , Th_β , K_β) and the γ -particle spectra (U_γ , Th_γ) are based on data from the NNDC (Brookhaven National Laboratory, USA, <http://www.nndc.bnl.gov/>) online database NuDat (version nov 2009) (Kinsey et al., 1996). In addition, the K γ -spectra (K_γ) is constructed from NNDC online database NuDat (version jan 2013) (Kinsey et al., 1996).

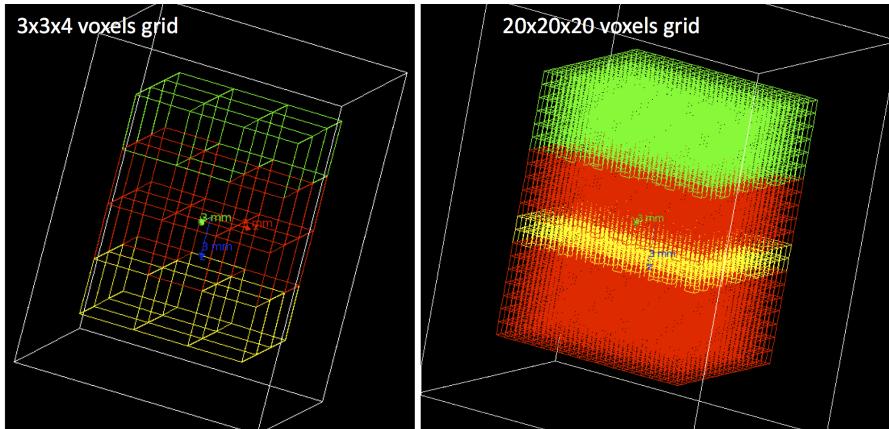


Figure 1: Example of voxel grids of different sizes.

1.2 Detectors

In *DosiVox* a detector is a virtual instrument or a solid material designed to detect the presence of particles and subsequently the emitted energy and deposited dose. *DosiVox*

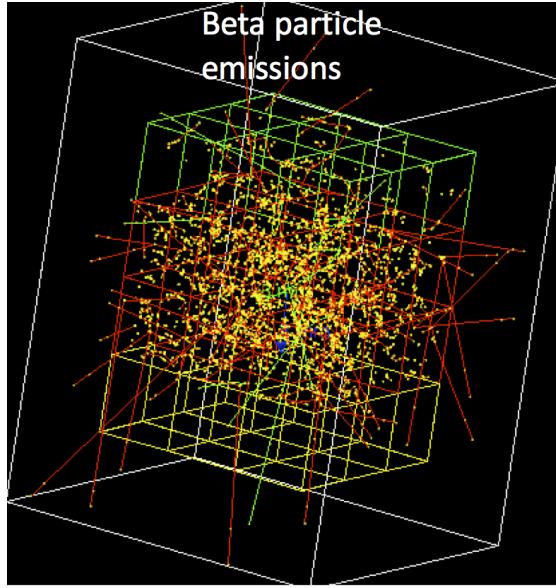


Figure 2: Example of voxel grid with particle interactions.

allows the virtual usage of different types of detectors: (1) simple detector (probe), (2) enhanced detector (a sub-voxelized voxel, a single grain packing, a successive grain packing).

1. **Probe:** The doses deposited during simulations are recorded in a always present cylindrical volume called the 'probe' (Fig. 3). The probe is divided in segments of equal size (each segment independently records a dose) and oriented along the z-axis, where its length equals the height of the model. Its diameter and the number of divisions can be chosen by the user. This detector **does not** interact with particles, it just defines the part of the geometry where the dose deposition is recorded. Additionally, the amount of energy emitted by mass of material (noted '*EmMass*') is recorded in each segment in order to allow a normalization of the dose results.

In addition to the probe, the user can add a solid detector which, in comparison to the probe, will be sensitive to the particles and will record doses and *EmMass*¹. There are 3 types of solid detectors: (1) a sub-voxelized voxel, (2) a single grain packing, (3) a succession of grain packings. Only one type can be defined in each simulation (Fig. 4).

2. **Sub-voxelized voxel (detector 1)** allows to divide the user-chosen detector voxel in a 3D grid in order to create a deeper level of geometric definition. This sub-voxels grid is defined exactly as the main voxels grid, but the doses and *EmMass* are recorded in each sub-voxel. This detector allows reconstructing objects in their environment (the main voxels grid) and mapping the deposited doses. A function in the interface allows loading 3D images in text format to construct the sub-voxelization (cf. Sec. 5.3).

¹Both can be expressed in Gy, but note the difference between emission and deposition.

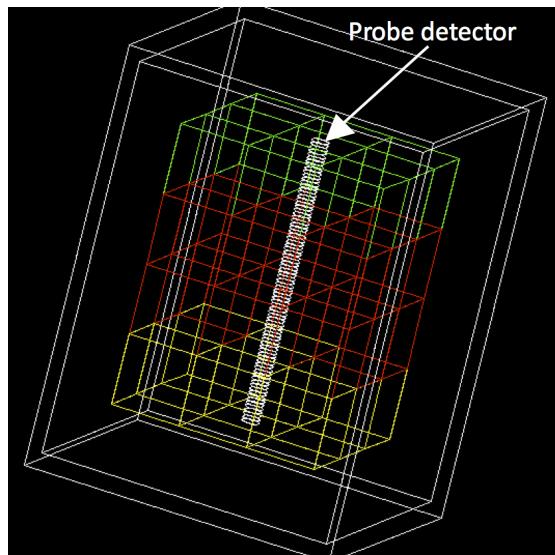


Figure 3: Voxel grid with the probe detector.

3. **The single grain packing (detector 2)** allows the user to define in a chosen voxel a box (with maximum size is the voxel size) and create a packing of mono-mineral spherical grains. Data concerning the grains come from the material chosen for filling this voxel. Each grain that does not overlap the packing box boundaries is considered as an independent detector recording the dose during the simulation. The *EmMass* is recorded in the entire packing box.
4. **The successive grain packings (detector 3)** is similar to the single grain packing (detector 2), but differs as it creates several packing boxes constituting a column along the z-axis. The X and Y positions of the column are defined by the position of the user-chosen voxel detector. Afterwards, each packing box contains a packing of grains created with the grain data of the material filling the voxel in which it is defined. In order to reduce the calculation time of the packings, only one packing geometry is defined by material: this geometry is copied in all the packing boxes filled with this material. Each grain that does not overlap the packing box boundaries is considered as independent detector recording the dose during the simulation. The *EmMass* is recorded in each packing box.

For the detector definition in the software compare Sec. 2 and Sec. 5.3. For the detector result reading, see the part 4.

1.3 Controlling the simulation

For making *DosiVox* usable by any person without special skills in programming, it can be controlled by using an unique ASCII-file, called 'Pilot Text File' (PTF). Examples of PTFs are given below and with the software and even though they look different, they all follow

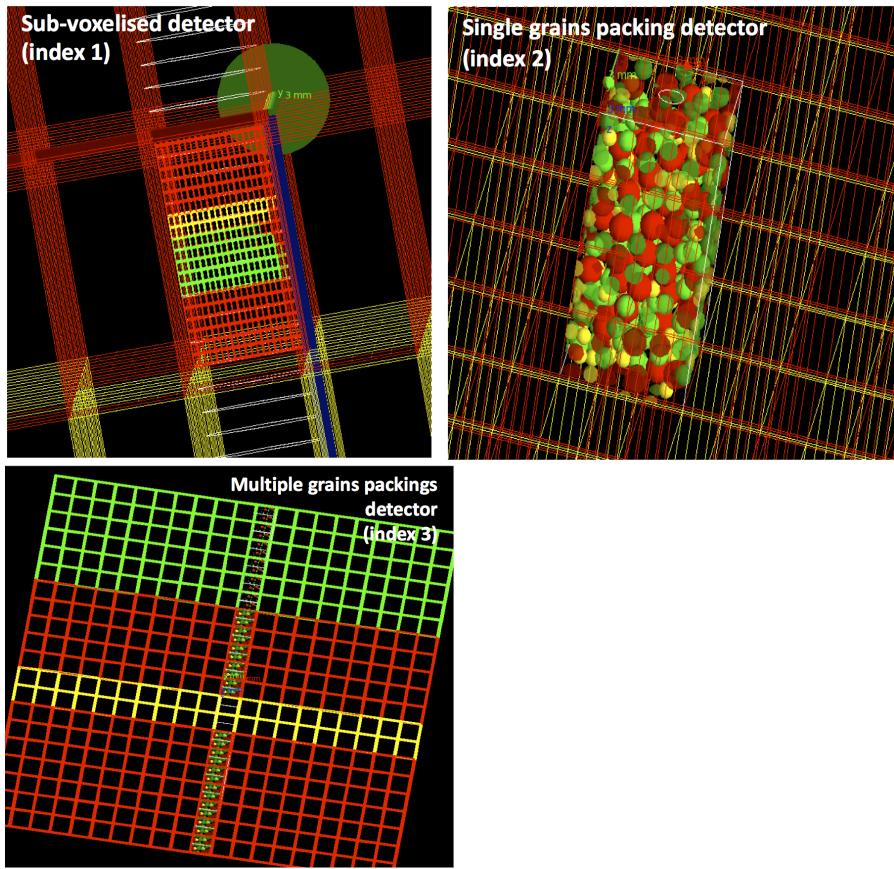


Figure 4: The different solid detectors available

the same rules that cannot be altered. A PTF defines the information necessary for totally defining the 3D geometry of the 'world' (the volume in which simulations will take place), the distribution of the radio-elements and many other parameters required for the simulation itself (type of particle α , β , γ ; size of the 3D-voxels in directions X, Y, and Z ; number of primary particles created during the simulation, etc.). In order to easily create and edit a PTF, an interface for PTF edition is available (cf. Sec. 5).

2 The Pilot Text File

The Pilot Text File (PTF) is the central element for controlling a simulation in *DosiVox* and an example is given below. This original PTF can be also found in the *DosiVox* data folder, named as XPL1:

Listing 1: Example PTF

```
1 XPL1_U_b                      # result file name
2
3 100. 1.                         # ( x 1000 ) emitted particles , clock value (%)
4 2 0 0 0                          # particle emitted: 1 for alpha , 2 for beta , 3 for
      gamma ; momentum of emission in X,Y,Z directions (0 0 0 for random
      direction)
5 1 0.005                          # element emitter (1 for uranium series , 2 for
      thorium series , 3 for potassium , 0 for User Defined) , cut in range
      value (mm)
6 0 13 -1 0                        # index for detector definition , detector voxel ,
      material for mapping , emission from grains
7 80 1 50. 50.                      # number of probe cells , probe diam (mm) , probe offset in X
      and Y (in % of voxel X,Y sizes)
8 1                                  # number of new components defined
9 3                                  # number of materials used
10
11 18 _NO2 1 2 N 1 O 2
12
13 0                                # material index
14 Clay                             # material name
15 1.8                             # dry density (g/cm3)
16 10                               # water content , % of dry mass
17 4                                # number of components
18 1 60 2 30 9 5 18 5              # component index , % of dry mass
19 ##### granulometry #####
20 2 2.65 5 2                      # grain component index , density , compacity (%) ,
      number of granulometric fractions
21 ##### diameter in mm, volumetric fraction (%) of the total grains
      mass (sum must be equal to 100) #####
22 0.15 70
23 0.05 30
24
25 1                                # material index
26 Sand                            # material name
27 2                                # dry density (g/cm3)
28 12.6                            # water content , % of dry mass
29 1                                # number of components
30 1 100                            # component index , % of dry mass
31 ##### granulometry #####
32 0 0 0 0                          # grain component index , density , compacity (%) ,
      number of granulometric fractions
33 ##### diameter in mm, volumetric fraction (%) of the total grains
      mass (sum must be equal to 100) #####
34
35 2                                # material index
```

```

36      _Limestone      # material name
37      2.7              # dry density (g/cm3)
38      3.5              # water content , % of dry mass
39      2                  # number of components
40      3 70 4 30         # component index , % of dry mass
41      ##### granulometry #####
42      0 0 0 0            # grain component index , density , compacity (%) ,
   number of granulometric fractions
43      ##### diameter in mm, volumetric fraction (%) of the total grains
   mass (sum must be equal to 100) #####
44
45      3      3      3  # voxels number along X,Y and Z axes
46      4      4      4  # voxels size along X,Y and Z axes (mm)
47
48      ##### MEDIUM COMPOSITION – cartography of materials with their index
49
50      1 1 1
51      1 1 1
52      1 1 1
53
54      0 0 0
55      0 0 0
56      0 0 0
57
58      0 0 0
59      0 0 0
60      0 0 0
61
62      2 2 2
63      2 2 2
64      2 2 2
65
66      ##### Cartography of U, Th or K for the external medium :
67
68      2 2 2
69      2 2 2
70      2 2 2
71
72      10 10 10
73      10 10 10
74      10 10 10
75
76      10 10 10
77      10 10 10
78      10 10 10
79
80      0 0 0
81      0 0 0
82      0 0 0

```

2.1 How to interpret and edit a Pilot Text File

The utilization of the PTF follows some simple rules:

- All information in a line after the symbol '#' are comments and are not interpreted by the software,
- different data on a line are separated by at least one space (adding more space does not affect the PTF interpretation anyway), successive comments respectively refer to the successive data on the same line,
- provide metric values (e.g., length) in mm (millimeter), densities in g cm^{-3} and proportions in % (percent),
- a '.' (dot) is used as decimal separator, **DO NOT** use comma.
- **DO NOT** use scientific notation for numbers (e.g., 10E03, 10^3 , 10^3),
- **DO NOT** add supplementary lines, this breaks the PTF. If you feel a need to add supplementary data (for a new material, or for modifying the model map), make sure you respect the layout of the data blocks shown on the example above.

Nota bene: several data have to be in agreement. For example, if the number of materials is '3' (line 9 in the example), 3 materials have to be defined below (Sec. 2.3). In the same way, make sure that the geometry and radio-element mapping are in agreement with the number of voxels defined in each axis (Sec. 2.4), and that the voxel number defining the detector position is correct.

2.2 General parameters (header)

The first block of data in every PTF (lines 1–9, header) refers to general parameters piloting the simulation:

- Line 1: the result file name. Further, the result file name is the prefix that will be added to every text file containing simulation results.
- Line 3: the first number represents the number of particles (in thousand of particles) emitted during a simulation run. The second number is the update frequency of the simulation progress bar in the terminal where *DosiVox* is running. It is given in % of the total number of particles emitted. In the example above, the pair 100. 1. indicates that 10^5 particles (100×1000) will be emitted, and the simulation progress will be updated every 1% of the 10^5 particles (ca. one update every 1,000 particles).
- Line 4: the first index defines the particle type (α - (1), β - (2) or γ -particles (3) can be simulated using *DosiVox*, but only one particle type in a simulation run). The three following numbers are the cartesian coordinates of the row vector defining the emission directions of the primary particles (0 0 0 means a random vector of emission for each particle). This vector can be used to define a unidirectional particle flux.

- Line 5: the first index defines the radioactive element that is considered in the simulation. The energy spectrum of the emitted particles is set with this index.

In the example above, the index 2 is used for the particle type (line 4) and the index 1 for the radioactive elements (line 5) mean that β -particles will be generated during the simulation with the energy spectrum of the β -emission of the U-series in secular equilibrium.

The β - and γ -emissions spectra for ^{40}K , U-series (^{238}U and ^{235}U in natural proportions) and Th-series – these last two being in secular equilibrium –, as well as the α -spectra for the U-series and Th-series, are available in the *DosiVox* folder data/spectra as text files. Another file named *user_defined* allows the user to define any other spectrum that can be chosen in the PTF by using the index '0' for the element emitter. To define a new spectrum, the user has to fill data in two columns (after the first line in the '*user_defined*' file): the first one is the energy of the particle (in keV), and the second the cumulative probability, for each user defined energy (the last value of this 2nd column must then be 1).

The 2nd number in Line 5 is the cut in range value for the production of secondary particles, in mm. If the emission of a secondary particle is calculated during the simulation, and its range in the material is superior to the cut value, the particle is simulated. If the particle range is inferior to the cut value, the simulation of this secondary particle is replaced by a local deposit of the kinetic energy.

- Line 6: The first index defined the type of detector used in the simulation: 0 for the cylindrical probe only, 1 for the sub-voxelised voxel, 2 for a single grains packing, or 3 for a succession of grains packings (cf. Sec. 1.2). The 2nd index is the number of the voxel defining the position of the detector (termed here $N_{detector}$).

To obtain the voxel position in cartesian coordinates (unit is number of voxels), use the formula:

$$X_{voxel} = N_{detector} \% N_{voxelsX} \quad (1)$$

$$Y_{voxel} = \left(\frac{N_{detector} - X_{voxel}}{N_{voxelsX}} \right) \% N_{voxelsY} \quad (2)$$

$$Z_{voxel} = \frac{\frac{N_{detector} - X_{voxel}}{N_{voxelsX}} - Y_{voxel}}{N_{voxelsY}} \quad (3)$$

where % is the modulo. X_{voxel} , Y_{voxel} and Z_{voxel} are the x-, y-, z-coordinates in voxels respectively, and $N_{voxelsX}$ and $N_{voxelsY}$ are the numbers of voxels of the model in the x- and y-axis, respectively. The reverse formula (to

calculate the detector voxel number from x-,y,z-coordinates in the voxel) therefore becomes:

$$N_{detector} = X_{voxel} + Y_{voxel} * N_{voxelsX} + Z_{voxel} * N_{voxelsX} * N_{voxelsY} \quad (4)$$

The next index defines the material where the dose is mapped in the case a sub-voxelised detector is chosen (the -1 value indicates that the dose is mapped in all the materials of the detector). The last index of line 6 refers to the two grains packing type detector: the value 0 indicates that in the case of particle emissions from this detector, the particles will be emitted from the matrix surrounding the grain. The value 1 would define that the particles will be emitted from the grains.

- Line 7: This line defines the probe detector characteristics: the 1st value is the number of segments of the probe, the 2nd value defines its diameter (in mm), and the next two values are the offsets in x- and y-direction of the detector voxel (in % of the voxel). Here 50. and 50. means that the central axis of the probe corresponds to the center of the voxel.
- Line 8: the value corresponds to the number of new components created for the simulation (Sec. 2.3).
- Line 9): the value corresponds to the number of materials created in the simulation (Sec. 2.3).

2.3 Material definition

The materials used to fill the voxels in *DosiVox* are made of components. The mass proportions of these components have to be defined. Several components can be found in a predefined list (18 basic components referenced by their index, from 0 to 17, are currently available; see below, or see the `basic_components_list` in the data folder of *DosiVox*), or can be created using a chemical formula. The number of new components created and the number of materials defined are indicated in the PTF, respectively in lines 8 and 9.

2.3.1 Creating a new component

To define a new component by a chemical formula follow the layout of line 11 in the example above, e.g., 18 _NO2 1 2 N 1 0 2.

The first number is the index of the new component, starting at 18 (following to the basic components indexes). The next data (_NO2) is the name given to this component. Make sure not to use a name already assigned to a component or a material defined in the simulation: this would create an error during the simulation. The next two numbers (1 and 2) are the density of this component and the number of atom types, respectively. The chemical formula is defined using the element abbreviation, followed by the number of atoms. In this example, the component is made of 1 atom of nitrogen (N 1) and 2 atoms

Table 1: Basic component list

Index n°	Name of component
0	H ₂ O
1	SiO ₂
2	Al ₂ O ₃
3	CaCO ₃
4	MgCO ₃
5	Limestone*
6	ZrSiO ₄
7	KF
8	H _{apatite} **
9	Fe ₂ O ₃
10	FeO
11	MgO
12	CaO
13	Na ₂ O
14	K ₂ O
15	air
16	vacuum
17	Pb

* CaCO₃ 70 % in mass, MgCO₃ 30 % in mass

** Hydroxyapatite: Ca₅(PO₄)₃(OH)

of oxygen (0 2). The user can add as many new components as wanted by following this layout and respecting the index numbering. In the PTF, an empty line must separate each new component definition.

2.3.2 Creating a new material

To create a new material, the following layout has to be respected (example from line 13 to 23):

Listing 2: Creating a new material in the PTF

```

13 0          # material index
14 Clay       # material name
15 1.8        # dry density (g/cm3)
16 10         # water content, % of dry mass
17 4          # number of components
18 1 60 2 30 9 5 18 5  # component index, % of dry mass
19 #### granulometry ####
20 2 2.65 5 2 # grain component index, density, compacity (%), number of
               granulometric fractions

```

```

21 ##### diameter in mm, volumetric fraction (%) of the total grains mass (sum
22   must be equal to 100) #####
22   0.15 70
23   0.05 30

```

- Line 13 – 16: The index of the material starts from 0. This index is used to define the geometry of the simulation (Sec. 2.4). Make sure not to use a material name already assigned to a component or a material defined in the simulation as this would cause an error.
- Line 17: the number of components used to create the material.
- Line 18: this line indicates the component indexes (found in the basic components list or assigned to the new components defined in the PTF) followed by the mass proportion of this component. In the example, the material consists of SiO₂ (60 % in mass: 1 60), Al₂O₃ (30 % in mass: 2 30), Fe₂O₃ (5 % in mass: 9 5) and _NO2 (5% in mass: 18 5). The values have to be separated by at least a space, and the user have to make sure that the sum of the mass proportions equals 100 %.
- Line 20: here the user can indicate the grain characteristics that will be used to create the grains packings (detector 2 or 3). If no grain packing is used in the simulation, these data will be read but not used. If the material does not contain any grain (in the facts or for the modelling), the 4 numbers of this line must be 0 (see line 32 in the PTF example).

The first value corresponds to the component index defining the grain material. It must appear in the list of components selected by the user (line 18). In the example, index 2 is the index for Al₂O₃ (Table 1). The next value of the line is the grain density (in g cm⁻³), followed by the compacity of the grains in the material (in % of the volume of the packing box). The component mass used to create the grains is subtracted from the mass defined for the material (line 18). This ensures that the voxel of this material contains the same average composition; a grains packing is created or not. The user has to make sure that the mass of the grains does not exceed the component mass defined for this material. In the example, the grain mass can be calculated as $2.65 \cdot 0.05 \cdot V_{voxel} = 0.1325 \cdot V_{voxel}$, where V_{voxel} is the volume of a voxel. The mass of component 2 is $1.8 \cdot 30 \% \cdot V_{voxel} = 0.54 \cdot V_{voxel}$. In this case, the component mass is sufficient to create the requested grain packing. The last value in this line is the number of grain sizes defined for the packing (following lines)

- Line 22 – 23: Grain sizes are defined in lines 22 and 23 (make sure that the number of lines corresponds to the number of grain sizes defined in line 20): the first value of each line corresponds to the diameter of the spheres representing the grains (in mm), and the second value is the volumic fraction of this grain sizes given in % of the total volume of the grains. The sum of the volumic fraction must be 100 %. Two other material definition blocks can be found in lines 25 to 33 and 35 to 43 (cf. Listing 1).

The number of material definition blocks must be identical to the number written in line 9. The material definition blocks must be separated by an empty line.

2.4 Geometry and radioactivity of the model

The line 45 (cf. Listing 1) in the example defines the numbers of voxels in the x-, y- and z-direction, respectively. The next line gives the dimensions of a voxel, in the x-, y- and z-direction, in mm (all the voxel shapes are identical). These two lines allow the user to define the total size of the grid and the level of voxelization.

The data from line 50 to 64 and from line 68 to 82 allow filling the grid with the material information and radioactive elements concentration for defining the geometry and radioactivity of the 3D model. They consist of a series of N_z blocks of N_y lines, each line containing N_x indexes (considering that the grid contains $N_x \times N_y \times N_z$ voxels). The user has to make sure that this layout is in accordance to the data defining the number of voxels in each direction of the grid (line 45 in the example). The different blocks represent the different layers of the model in the xy-plan, the line represents the x-direction and the column the y-direction. Notice that the z-axis is oriented to the down in order to represent the depth. The voxels are numbered from 0 to $(N_x \times N_y \times N_z - 1)$, beginning at the top left position of the block series and ending at the bottom right position. For example again here, the two blocks from line 50 to 56:

Listing 3: Repeated section on the example PTF above

```

50 1 1 1
51 1 1 1
52 1 1 1
53
54 0 0 0
55 0 0 0
56 0 0 0

```

are numbered as follows:

```

50 0 1 2
51 3 4 5
52 6 7 8
53
54 9 10 11
55 12 13 14
56 15 16 17

```

These numbers are used in *DosiVox* to identify all the voxels. They are used in line 6, second datum, to identify the detector voxel, and in the different result files (Sec. 4). The series of blocks (lines 50 to 64) can be considered as a 3D material mapping. Each index of the map represents the material of the corresponding voxel. These are the material indexes defined at the beginning of each material definition block (lines 13, 25 and 35 in the example). The data from line 68 to 82 correspond to the radio-elements concentration of the voxels. Its layout follows the same rules as the material map. Each value represents the mass proportion of the radio-elements in the corresponding voxel. During the simulation,

each of these values is multiplied by the density of the material to obtain the activity in the voxel. This activity allows to calculate the number of particles to be emitted from this voxel. The default unit for the mass proportion values is the ppm, but any unit (even arbitrary) can be used as long as the results are normalized by a dose or an *EmMass* recorded during the simulation. The user has just to take care about the proportions between the radio-elements concentration in the different voxels. For example, if one considers a model made of 4 voxels arranged in a square, the radio-element concentration mapping

1	2
5	8

is equivalent to:

2	4
10	16

as the proportions between the different voxels are the same. The resulting simulations will be equivalent in both cases. In the case where a sub-voxelized detector is defined, these two maps (materials and radio-elements concentration) are followed by a line defining the number of sub-voxels in the x-, y- and z-direction of the detector voxel. The dimensions of each sub-voxel are calculated in each direction by dividing the size of the voxel by the number of sub-voxels in this direction, in order that the sub-voxelized grid entirely fills in the detector voxel. According to this line, two 3D maps (the first for the material, the second for the radio-elements concentration in the sub-voxels) are defined with the same layout that is valid for the main voxel grid, but with the corresponding number of sub-voxels in each direction (see the PTF XPL4 in the data folder). In the case where single or successive packings of grains are defined, the two maps of the main voxel grid are followed by lines similar to the ones that can be found at the end of the PTF XPL3 in the data folder:

Listing 4: Example for detector size definition

Define detector size
100 75 50 #box size inside the internal voxel , for X,Y,Z (in % of X,Y,Z voxel sizes)
50 50 50 #offset of box location inside the internal voxel , for X,Y,Z in % of voxel dimensions

The first data line defines the size of the packing box, in % of the x-, y- and z-dimensions of the voxel (the maximum value for each dimension is 100%). Note that in the case of successive grain packings, the z-size will be rounded to a integer number of boxes matching the voxel size in the z-dimension. The second data line corresponds to the offsets in the x-, y- and z-direction of the packing box or boxes compared to the detector voxel position (in % of the voxel size). The default values 50 50 50 indicate that the center of the box matches the center of the voxel. In the case of successive grain packings, as the successive packing boxes form a column over the entire height of the model, only the two first data of the line are considered, to set the offsets in the x- and y-plans. In both cases, if the packing boxes exceed the voxel boundaries, an error message will be displayed at the simulation launch and DosiVox will end.

3 Running DosiVox

3.1 Getting started with DosiVox

The commands given in the following section can also be found in the `basic_commands` text file in the *DosiVox* folder.

3.1.1 Open a terminal

DosiVox runs on Linux system using the terminal window and terminal commands. First, power up the Linux system, for example by opening the recommended virtual machine on a virtual machine player (see the *DosiVox* installation guide). Afterwards:

→ *DosiVox* folder →  → select *open in terminal*.

A terminal window will be open at the folder location (Fig. 5).

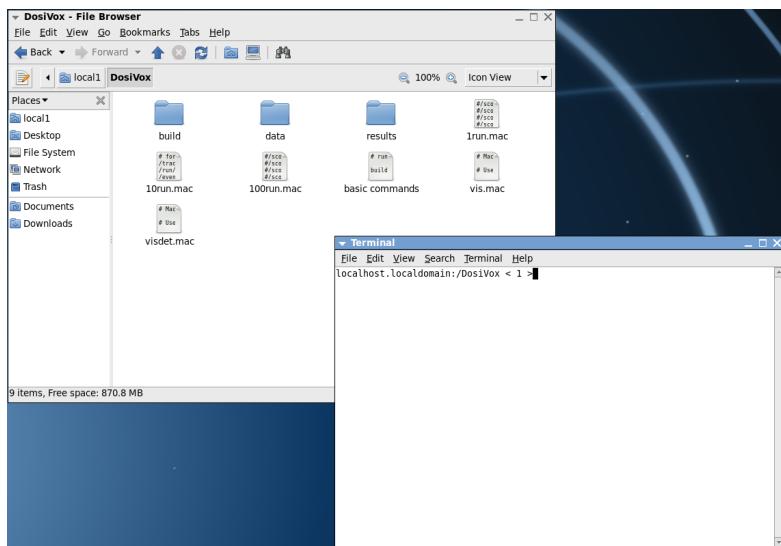


Figure 5: Screenshot file browser and terminal window

If the terminal is not open at the *DosiVox* folder location, you have to navigate in the virtual machine folder with the terminal to set it at the correct path:

→ terminal command line → `cd /[path to your DosiVox folder]` 

The command “`cd`” allows to return to the home folder, and the command “`cd ..`” allows to return one folder back. For example, if the terminal is open at the home folder location “`/local1`” and the *DosiVox* folder is on the desktop of the virtual machine, use the command “`cd /Desktop/DosiVox`” in the terminal command line to set the terminal in the *DosiVox* folder.

3.1.2 Basic DosiVox terminal commands

Once the terminal is located at the `/DosiVox` folder, you can use the following commands in the terminal command line to run *DosiVox*:

`build/DosiVox` run *DosiVox*. A PTF name is requested by the software to run the corresponding simulation.

`build/DosiVox vis.mac` runs the *DosiVox* visualization tool (QT *Geant4* tool), cf. Sec. 3.1.3 for further explanations. This command disables visualization of the detectors in order to reduce the computer memory and calculation power consumption.

`build/DosiVox visdet.mac` runs the *DosiVox* visualization tool and displays the detectors (probe, sub-voxels, grains). This displaying may be time consuming, especially when the detectors are composed of a lot of geometrical elements.

Press `CTRL` + `C` in the terminal window to abort a *DosiVox* session or a running simulation.

3.1.3 Visualization tool commands

The visualization tool of *DosiVox* allows the user to check the geometry and the particle emission of the model, by using a 3D representation. Running simulations using the visualization tool requires a lot of memory, therefore the number of particles emitted is limited to 1,000 when the tool is activated.

When the visualization tool of *DosiVox* is open, you can navigate in the visualization window by clicking in and then using the mouse left button and scroll, and the arrow keys of the keyboard. You can enter the following commands in the *session box* of the tool to run the simulation or exit the tool:

`control/execute 1run.mac` run a simulation with max. 1,000 particles. The particle tracks will be displayed at the end of the run.

`exit` exit the visualization tool. This will also exit *DosiVox* and returning to the terminal.

Color code

- The voxel and sub-voxel colors refer to the material filling in the voxel. Only 8 colors are available, so in the case of a larger number of defined materials, the same color will be attributed to different materials.

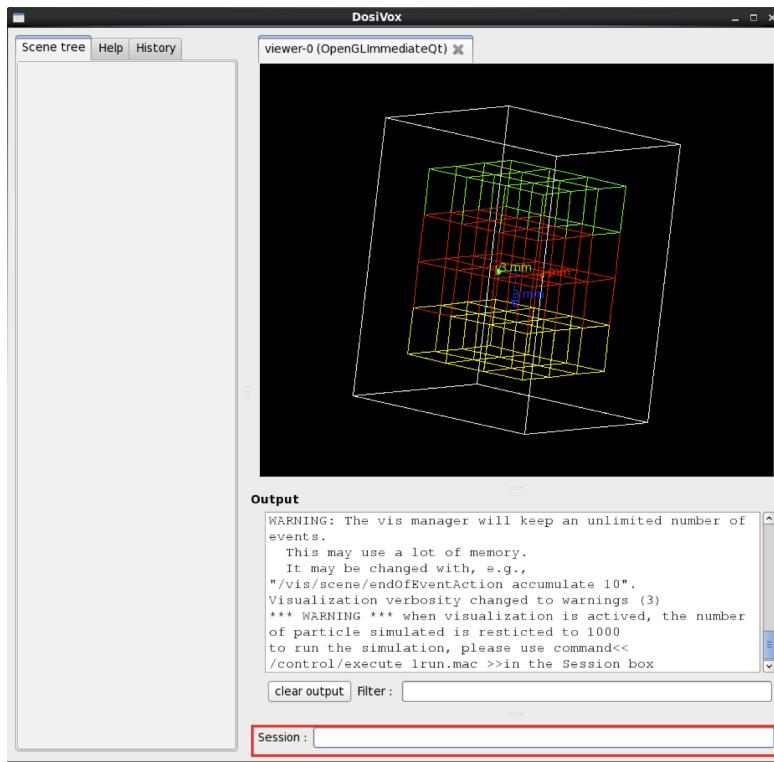


Figure 6: The visualization tool

- The grains colors refer to the diameter of the grains. As for the voxel color, if the number of defined diameters is larger than the number of available colors, the same color will be attributed to different grain diameters. The grains overlapping the packing box boundaries are displayed transparently.
- The particle track colors indicate the electric charge carried by the particles, using a Geant4-convention: **red** for the negatively charged particles (usually β -particles), **blue** for the positively charged particles (usually α -particles) and **green** for the neutral particles (usually γ - and X-rays). The **yellow** points indicate interaction points or a change of geometrical volume. In the case of particles with strong probability of interaction by unit of length, the user may see only successive yellow points of interaction instead of lines with the color attributed to the particle charge.

3.2 A DosiVox run

When starting, *DosiVox* calculates the number of particles to emit from each voxel, considering the total number of particles to emit during the simulation and the quantity of radioactive element in each voxel (calculated by radioactive element content \times material density). In the case where the sub-voxelized detector is set, the total number of particles to emit from the

detector voxel and the number of particles to emit from each sub-voxel are calculated using the sub-voxel maps.

When the simulation starts, the particles are emitted from the first voxel. When the number of particles to emit from this one is reached, the particles are emitted from the second voxel and so on. The progression of the number of particles emitted is periodically displayed, with a frequency depending on the clock value defined in the PTF in line 3. As the time necessary to simulate a particle and its interactions depends on the material and the geometry where it passes through, this progression may not exactly correspond to the time progression of the simulation.

The path and interaction of each particle (primary and secondary) are calculated with a Monte Carlo approach, selecting step by step the next most probable interaction considering the particle and the current material properties. If secondary particles are resulting from an interaction, they are explicitly simulated if their range in the current material is superior to the 'cut in range' value defined on the PTF at line 5 (Listing 1).

If not, they are replaced by a local deposit of the energy that they should have carried.

At each step, *DosiVox* checks if some energy has been deposited in a detector and recorded it. The energy of the primary particle is also recorded to allow calculating the EmMass values. At the end of the simulation, the energy recorded in each detector is normalized by the mass of the corresponding detector part (considering the volume of this part and the material density of the material filling in).

It may occur that the calculation of a particle path ends in an infinite mathematical loop, which leads to the calculation of an infinity of particle steps of zero-length. In this case, *DosiVox* 'kills' the particle, disappearing at this point in the simulation. In order to evaluate the impact of this process on accuracy of the results, the killed particle data are recorded in a separat RTF file with the *_Errors* suffix, as total quantity of energy lost by killing these particles. If this quantity is not negligible in regard of the total energy of the particles emitted during the simulation, insufficient accuracy of the results should be expected. Please contact the *DosiVox* technical support for help in such cases.

3.2.1 Internals

A *DosiVox* simulation runs on one computer core at a time; consequently, the simulation speed depends on the core frequency. *DosiVox* is not compatible with multi-threading, but you can run several parallel simulations by launching them one by one. The speed of different simulations may differ, depending on parameters like: the particle type and the energetic spectrum, the number of geometrical elements composing the model, the materials involved, etc. The 'cut in range' value has a huge impact on the simulation speed: you can speed up the calculation by increasing this value, but it can result in low accuracy in deposited dose values, because a part of the secondary particles will not be simulated explicitly. By default, we suggest to keep the 'cut in range' value much lower than the simulated particle average range, and smaller than the smallest geometric element composing the model.

If the user wants to increase simulation speed by increasing this value, he has to check for potential inaccuracy in the dose results. The memory consumption of *DosiVox* itself mostly

depends of the number of geometrical elements composing the model. In particular, the user has to be careful when a 3D image is used to create the sub-voxelized geometry: high number of sub-voxels can overload the available memory of the computer or of the virtual machine. The system monitor tool available on the scientific linux virtual machine allows to access to the computer resources consumption. It can be found in the linux task bar:

Applications→*System Tools*→*System Monitor*

4 Results and data processing

The results of any simulation are stored in the folders

- DosiVox/results/
- DosiVox/results/DoseMapping

They are provided as TXT-files and can be easily imported by e.g., *ExcelTM* or **R** for further analyses. At the end of a simulation, Result Text Files (RTF) are created with the result file name defined in the PTF used for the simulation and different suffixes are added depending on the detectors created by the user. Here is a list of the possible types of RTF-files that can be found:

4.1 Error file (*_Errors suffix*)

List of particles virtually 'killed' because they entered into a infinite loop of path calculation. The ratio of energy lost is reported in this file. A significant ratio of energy lost by killing particles on the total emitted energy may indicate inaccuracy in the simulation results (the user must decide if it is too severe). To try to reduce the problem, check the PTF and try to change simulation parameters, in particular the cut value, the resolution of the voxelised grid and the material parameters.

4.2 Probe results (*_Probe suffix*)

The name of the RTF for the cylindrical probe. It contains information about the position of the probe segment, the material at the center of the segment, and the dosimetry in all segments reported in a table form (cf. Table 2).

Table 2: Probe results example

depth(mm)	Material_Num	Material_Name	Density(g/cm3)	Water_content(%)	EmMass(Gray)	Dose_absorbed(Gray)
0.1	0	quartz	2	0	2.54994e-07	8.37316e-08

EmMass refers to the sum of the energies carried by the emitted particles by unit of mass in a volume (here each probe segment). It is expressed in Gy and would correspond to the infinite matrix dose in the case of an infinite and homogeneous medium. Note that the material name and the corresponding density (used to calculate the dose) correspond to the material at the point of the center of each segment (in the voxels or sub-voxels grid if it is the case) and **not** to the averaged material or density in this segment. The latter one would bias the dose calculation in cases where several materials are contained in the volume defined by the probe segment.

4.3 Sub-voxelised voxel results (_Detector1 and results/DoseMapping)

In the case the detector is 'voxelized', a RTF file named ResultFilename_Detector1 is created containing data about the voxelized voxel, data for each material in all the sub-voxels and information for the dose mapping.

Dose mapping: The folder DoseMapping contains the dose maps of the sub-voxelized voxel, layer by layer. A succession of text format images of the dose deposited in each sub-voxel (or only in the sub-voxels of the requested material in the PTF) can be find here. Each image represents a slice oriented on the xy-plan, and the suffix number refers to the position on the z-axis. To read these images the function import/text [image] from the free available software *ImageJ* (<http://imagej.nih.gov/ij/>) can be used. To create a stack containing all the images, use the following *ImageJ JavaTM* macro to open the images folder (you can find it on the *DosiVox* website named ImportTextImageSequence):

Listing 5: *ImageJ* macro

```
dir = getDirectory("Choose_directory");
list = getFileList(dir);
setBatchMode(true);
for (i=0; i<list.length; i++) {
    file = dir + list[i];
    run("Text_Image ...", "open=&file");
}
run("Images_to_Stack", "use");
setBatchMode(false);
```

Various functions and plugins for *ImageJ* allow reconstruction of a 3D representation using a succession of these 2D images and to calibrate them (3D viewer for example). To convert grey values in the image into doses in Gy, please use the equation in the precedent RTF. Example:

Listing 6: Example for color to dose conversion

```
DoseMapping information:
Dose maximum: Dmax = 2.5554e-05 gray
Dose minimum: Dmin = 0 gray
Equation for grey value conversion into dose in DoseMapping: dose(Gray)= (grey_value -1) * (Dmax(Gray)-Dmin(Gray))/65534 + Dmin(Gray)
```

The corresponding equation for grey value conversion in to dose in DoseMapping corresponds to:

$$dose[Gy] = (grey_value - 1) * (D_{max}[Gy] - D_{min}[Gy]) / 65534 + D_{min}[Gy] \quad (5)$$

4.4 Single and successive grain packing results (_Detector2, _Detector3 and _Grains)

In cases that the detector is 'Single-Voxel-Grain Distribution' or 'Successive-Voxel-Grain Distributions', the RTF files named PTFname_Detector2 and PTFname_Detector3 are created

respectively, but contain data about the grain averaged by packing boxes and diameters. The index ZdivNo (in case of the detector 3: 'Successive-Voxel-Grain Distributions') refers to the packing sequence number in the z-axis. The indicated *EmMass* is recorded for each packing box. The RTF with the _Grains suffix contains the details of doses recorded for each grain detector created in the model. The density column refers to the density of the grain material.

5 DosiVox interface for PTF creation and editing

The objective of the *DosiVox* interface is to provide the user an easy way to create PTFs. This application has been developed with *Microsoft Visual BasicTM* to work in the *MS WindowsTM* environment.

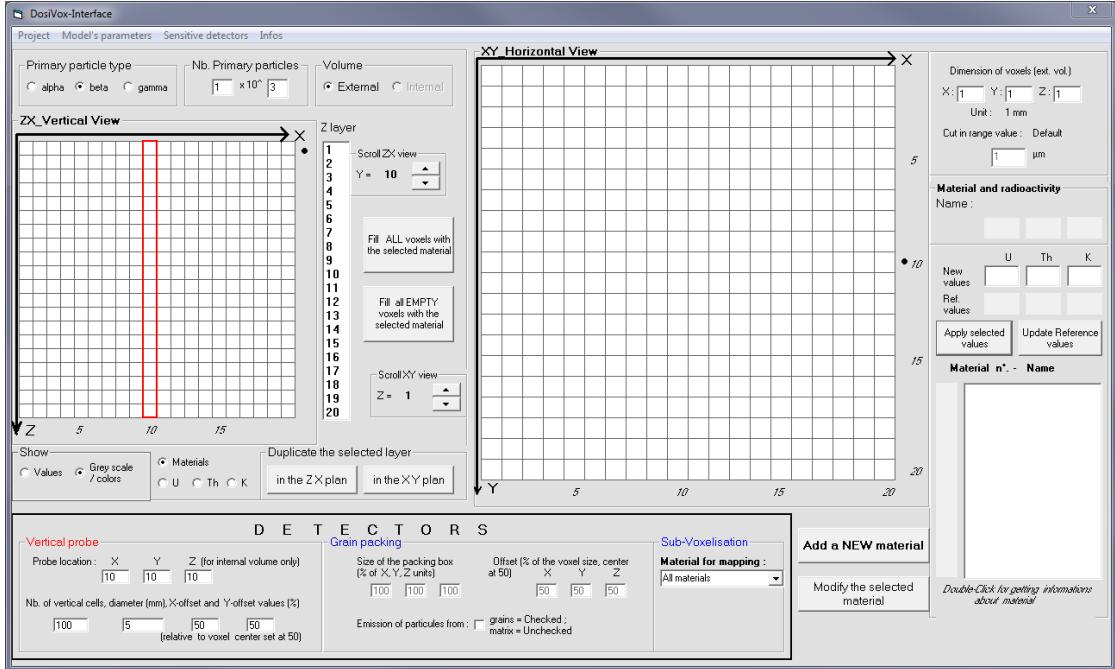


Figure 7: The *DosiVox* interface.

5.1 Material creation

5.1.1 Filling with materials

Any object defined in the world must be filled with a material. The materials are defined by clicking on the *Add a new material* button. The opened window (*Make a new material*) contains a list of the primary components already defined in *DosiVox* (but we will see that their characteristics can be changed). New components can be created and added to this list in using the Periodic Table (Fig. 8): just click on the selected element, indicate the number of associated atoms, and continue by defining the formula of interest. The density of the created component must be defined before adding it to the list of available primary components. A material used to fill voxels must have a name. Each material is made of an unlimited number of primary and/or user defined components whose proportions are defined as mass percentages. The 'dry' density can be adjusted manually, as well as the water content. If the user chooses a non zero value, water is added in the material composition in order to represent the moisture and to considerate the attenuation during the simulation.

The simulation results will be normalized by the 'dry' density of the materials.

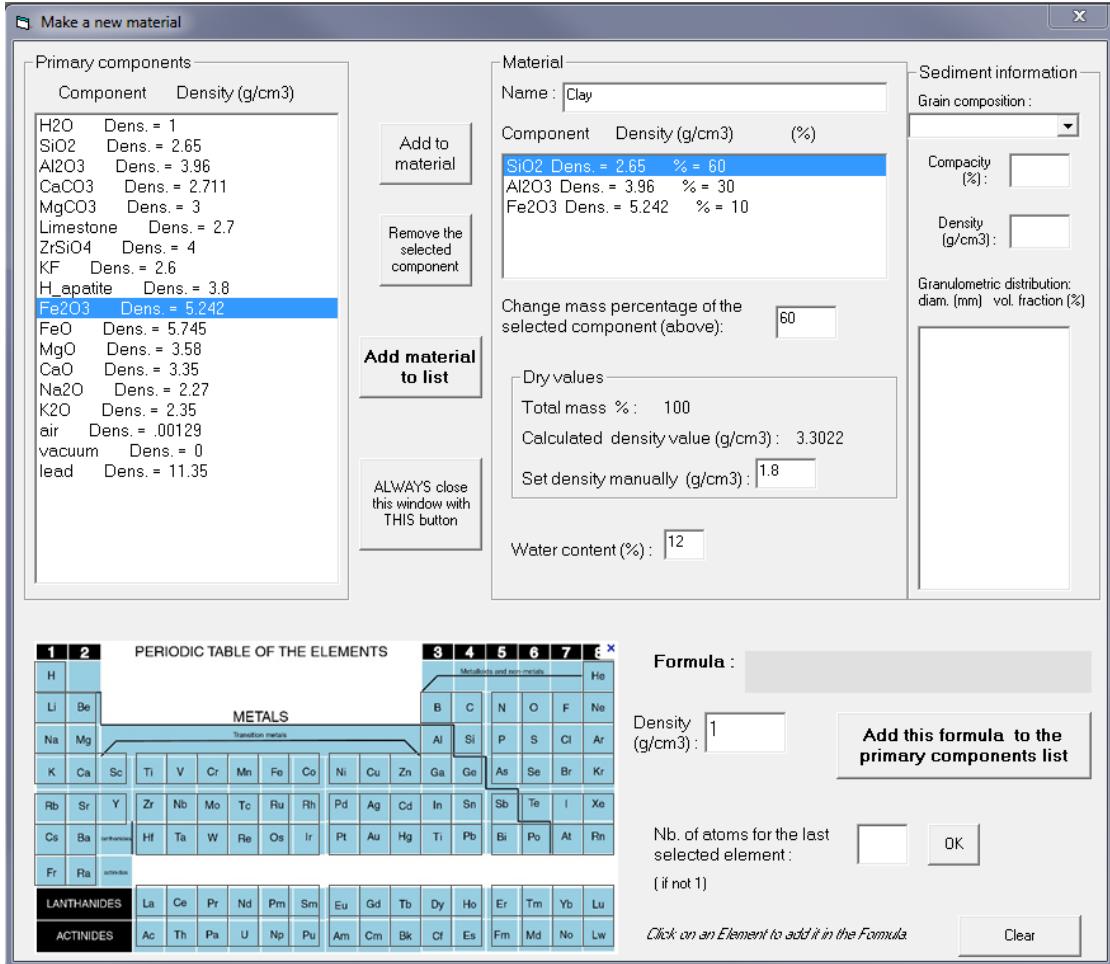


Figure 8: Click on the *Add material to list*- button for making the new material available; it then appears on the right side of the main window and is now usable to fill voxels.

5.1.2 The case of sediments

In the case where grains are explicitly created by *DosiVox*, their composition and granulometry must be defined in the *sediment information* box in the material creation window. The user has to select a component defined for the material and set the grain density and compacity to create them (the grains mass are created from this components content in the material) (Sec. 2.3.2). It is advised not to exceed 60 % since the creation of the geometry could take hours or even more when running *DosiVox*. To set the different grain sizes and their proportions, the user has to fill the 'granulometric distribution' box with lines containing the grains diameter (in mm) followed (after at least a space) by the percent of volume

of grains of this diameter on the total volume of grains. Only one pair of values 'diameter proportion' has to be set by line, and the diameter must be indicated in decreasing order. The user has to make sure that the sum of the different grain size proportions is '100'. Data example:

1.	25
0.2	50
0.05	25

These data indicate that the grain size distribution is composed by 3 diameters of 1 mm, 200 μm and 50 μm in the volumic percentages of 25, 50 and 25 respectively.

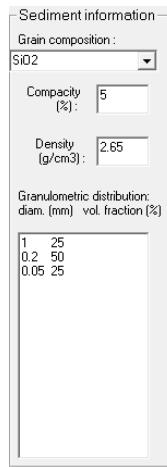


Figure 9: The sediment information window

5.2 Grid size and navigation

5.2.1 Defining own geometries

The model created by the interface is divided in a $20 \times 20 \times 20$ grid (i.e. 8,000 voxels) whose dimensions are defined in the right-upper part of the main window, and their unit in the menu :

Model's parameters → *Voxel dimension unit*

A voxel can be cubic or non-cubic. Filling of a voxel with a material is done by simply selecting a material in the list, and clicking on a voxel: by doing this, the color of the voxel becomes the color associated with this material. The colored voxel appears on the *XY_Horizontal View* of the world and in the *ZX_Vertical View*. Use the *Scroll_ZX View* button or the *Scroll_XY View*-button to display the other slices of the *XY_Horizontal* or the *ZX_Vertical* views. Since a minimum of 8,000 voxels must be filled, tools are provided to speed up this task. Click directly on the numbers (from 1 to 20) of the z-layer, or use the

button *Duplicate the selected layer* in the ZX_plan or in the XY_plan. All voxels need to be filled: use the *Fill all EMPTY with the selected material* button if necessary. Combinations of the selections values or Grey scale/colors with Material or U or Th or K allow many different displays of the information related to the distribution of the materials and their radioactivity in the world.

5.2.2 Radioactive sources

To associate U, Th, K values to a material, select the material, type in the U, Th, K values at the right hand side and click on the **Update Reference values** button. If this material has already been used to fill in voxels, all the previous U, Th, K values of these voxels will be replaced by the new ones. If you want to change the values yourself manually:

Apply selected values → [select voxels]

You can either display numerical values or colors in voxels by using the selections on the left side in the 'Show' area.

5.3 Detectors and other parameters

5.3.1 Detectors

Four types of detector are available in *DosiVox* (→*Internal Volume*).

1. If you choose 'None' in this menu, no specific detector is selected, but the probe detector will automatically be activated. Notice that this detector is always created when a simulation starts (whatever is your choice in this menu). The probe is a vertical detector (cf. Sec. 1.2) and, as seen previously, is divided in cells of equal height. Its diameter and location are also defined on the main grid (the external grid). You may also adjust its location in a voxel using the two offset values. All these parameters are adjustable in the Vertical Probe area (bottom-left corner of the main window, Fig. 10).

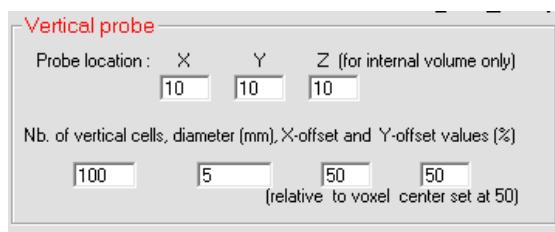


Figure 10: Adjusting the probe area

2. →*Voxelixed* in the Menu creates a second $20 \times 20 \times 20$ grid in the voxel located in the detector voxel defined by the X,Y,Z probe location. This internal grid allows definition of the internal volume. Switching from the external to the internal volumes is done with the selection options Volume:External/Internal. The information displayed on the

screen (i.e. on the visible grid) depends on this selection. You may fill in the voxels of the internal volume exactly as you did for the external volume. The material for the dose mapping results can be selected with the ‘Material for mapping’ list (select “All materials” for a dose mapping in all the sub-voxel grid).

3. In the →*Single-Voxel-Grain Distribution* option, the detector voxel defined by the X,Y,Z probe location is used to describe a sediment sample, i.e. a series of compacted grains, whose information related to the granulometry, density, compacity and grain composition were defined earlier in this guide (Sec. 1.2). This series of grains will be explicitly created in this voxel, and only in this one.
4. In the →*Successive-Voxel-Grain Distribution* option, all the voxels composing the Probe are sediment-filled with characteristics discussed just above in section 3-. The grains are explicitly created in all the voxels in which the probe is created, but differ in their character layer-by-layer in accordance with the material filling in the voxel and its own grain characteristics (Sec. 1.2). Moreover the other voxels will only contain the homogeneous defined material, without explicitly simulated grains.

If you wish this internal volume to be equal to the volume of one voxel of the external grid, do not change the 3 parameters named Size of the internal volume. If you reduce the size of the internal volume (Fig. 11), you may want to move it relatively to the center of the voxel in which it is created: use the parameters ‘Offset’ in ‘X’, ‘Y’, ‘Z’ to do that. All these parameters are adjustable in the Internal Volume Parameters area (in the middle-bottom of the main window). In this area, you can if you wish the particles to be emitted from the grains or from the matrix surrounding them. Note: If the grain packing volumes (or boxes) overlap the voxel boundaries, the model and simulation could suffer some inaccuracy. In this case, a warning message will be displayed by DosiVox.

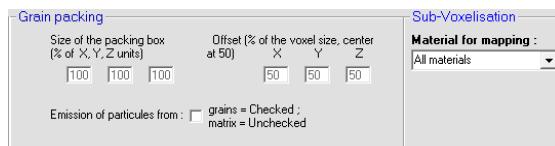


Figure 11: Adjusting the solid detector parameters

5.3.2 3D-Scan loading

DosiVox offers the possibility to load a 3D scan (Fig. 12, 13) in raster format to create the sub-voxelized geometry (in the case of the use of the sub-voxelized detector). This type of image can be obtained for example with a 3D scanner or made by the user with the ImageJ software (see part 5.4 for instructions on how to do that with this software). Any object will be represented by a text images sequence, meaning a series of superposed slices, each one being recorded as a Text file (*ImageJ* can convert many 2D format image stacks in a succession of Text file format image, like Tiff, jpeg, png, ...). Each file looks like an $N_x \times N_y$

matrix of numbers which represent different grey colors that are necessary for describing the different parts of the object:

→ Internal volume → Voxellixed → Project → Load a 3D-Scan → [search list of text files describing the object] → [select one of the text files] → OK

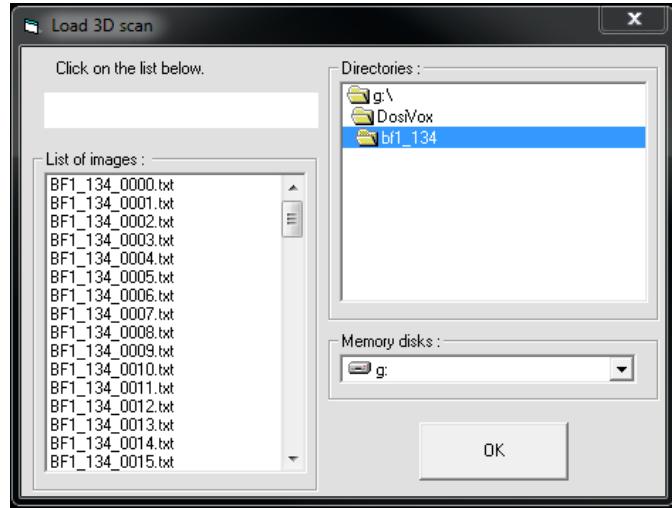


Figure 12: The load 3D-scan menu

In the new window (notice that it may take time for your computer to read all the text files and look for all the distinct numbers - representing the distinct grey colours), the user is asked to associate all the different numbers found by the software with the materials he or she previously defined. Before any association, make sure to enter the U, Th and K contents (0 if none) with the same unit as for the main voxel grid, that will be associated for the selected grey level. The result of your association will be displayed on the right side of the window. Notice that a single material can be associated with several grey level numbers (with or without different U, Th, K values). After creating all associations: → **Apply associations**.

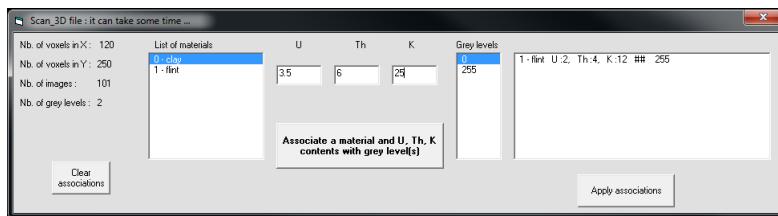


Figure 13: Scan 3D-file menu

On the left side of the 3D-scan window (Fig. 13) the dimensions of the $n \times m$ matrix (i.e. the number of voxels in the X and Y directions) are given, as well as the number of

images and the number of grey levels detected.

Warning: if you had chosen a grid larger than $20 \times 20 \times 20$, the image in the Interface window will not faithfully reproduce larger grids. However, the full grid representation will have been faithfully modelled inside the software. The user will have to make a tradeoff between computing time and resolution of the object, and hence of the dosimetric results. You might consider trying $100 \times 100 \times 100$ grid size as a first attempt.

5.3.3 Save a project and create PTFs

To save a project and create PTFs:

$\rightarrow \text{Project} \rightarrow \text{Save project} \rightarrow [\text{name your project}]^2 \rightarrow \text{Save}$

This action will create 5 distinct files. **Four files will comprise your project name.** For example, if you decided to simulate primary γ -rays:

- ProjectName_K_g
- ProjectName_U_g
- ProjectName_Th_g
- ProjectName_ud_g

These are Pilot Text Files (PTF) that will later be used to run *DosiVox* in the Terminal of the Virtual Machine. If you had chosen to simulate β -particles, you would obtain these files:

- ProjectName_K_b
- ProjectName_U_b
- ProjectName_Th_b
- ProjectName_ud_b

and these files if you had chosen to simulate primary α -particles.

- ProjectName_K_a
- ProjectName_U_a
- ProjectName_Th_a
- ProjectName_ud_a

²Hereafter called 'ProjectName'

K, U, and Th refer to the spectra available in the folder spectra and ud refers to the spectrum UserDef also available in this folder. Note that in this latter case, the distribution of the radio-elements follows the distribution of the U created by the user.

The fifth file has the name of the project (with an extension ".dvit"). It is also a text file used by the interface for reloading a project that had been already saved. The user can then load an 'old' project, change parameters, save it after the name is modified. Note that in case a 3D image was loaded when creating a project, this image was not saved and then will not be reloaded. Now you can proceed to use the PTF files to run your calculation using *DosiVox*.

5.3.4 Files associated with the radioactive sources

DosiVox contains a series of 8 spectra: 6 for the U- and Th-series (α , β and γ), and 2 for ^{40}K (β and γ) given as text files in the folder (DosiVox/data/spectra/). These spectra are based on informations obtained from the NIST (Kramida et al., 2018) and assume secular equilibrium in the U- and Th-series. Each spectrum is composed of two columns: the energy of the emitted particle (in keV), and its associated cumulative probability (*ProbaCum*). It was chosen to cumulate the probabilities starting from the highest energy of the considered spectrum, and finishing with the lowest energy and a cumulative probability of 1. Note that the first line of the text file is reserved and contains the information: 'Energy (keV)' 'ProbaCum.' A *UserDef* spectrum is also given below as an example and allows the user to define another source of radiation. Any other spectrum can then be built by following the rules defined in the previous paragraph (for instance: U-series with disequilibrium, artificial source, ...). Notice that *DosiVox* works only with the files whose names are: Kbeta, Kgamma, Ualpha, Ubeta, Ugamma, Thalpha, Thbeta, Thgamma, UserDef. For instance, opening the file to simulate ^{137}Cs photons can be done as follow:

Listing 7: Simulating ^{137}Cs photons

Energy (keV)	Cumul_Proba
662	1

Note that at least one free space must be set between the energy value and the cumulative probability.

5.4 Creating a model using ImageJ

ImageJ is a free software for scientific image processing, analysis and editing (Rasband, 1997-2012; Schneider et al., 2012). It is available at <http://imagej.nih.gov/>. A lot of plugins can be found on the internet and on the *ImageJ* website to add functionalities to the software. This part describes a way to obtain 3D images that can be used for creating sub-voxelized model in *DosiVox*, using the graphical interface.

Firstly, the user must open or create a 3D image in a 8-bit raster format (Fig. 15). This type of image is composed by a sequence of 2D 8-bit images (tiff, jpeg, png,...), each image representing a slice of the 3D image. These successive slices represent the successive data

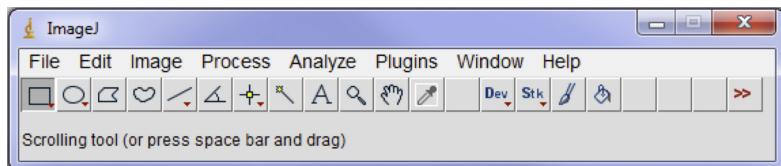


Figure 14: The *ImageJ* menu bar.

blocks composing the material and radioactivity maps in the PTF (Sec. 2.4), each pixel representing a voxel. Each pixel value (in 8-bit grey levels) will be associated with a material and a radio-element content during PTF creation. It is noticeable that *ImageJ* can import the raw image formats in order to create the 3D image (this process requiring few pieces of information about the final image size). If the user has to use a 3D image in a vector format, he has to convert it to a raster first to obtain a format compatible with *ImageJ*. The software *Binvox*³ is able to convert the most common 3D image formats and create raw format images, which can be imported in *ImageJ*.

To open a 3D image in *ImageJ* in a raster format:

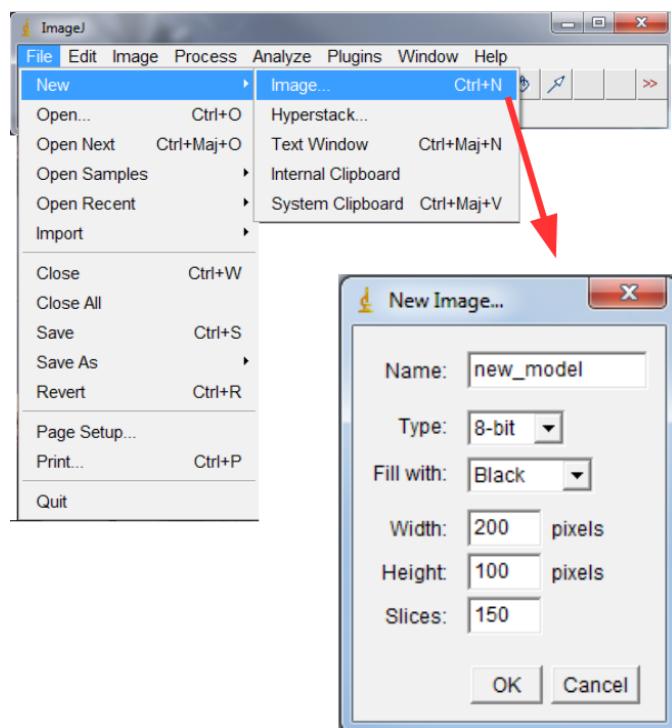


Figure 15: Creating new image in *ImageJ*

- 3D image is TIF format: → *File* → *Open*

³<http://www.cs.princeton.edu/~min/binvox/index.php>

- 3D image is a set of 2D images: → *File* → *Import* → *Image sequence*
- Other formats: → *File* → *Import* → *raw*

In case of opening text image sequence format, the plugin described in Sec. 4.3 (Listing 5) can be used. If the images sequence opened isn't in a 8-bit formats, it has to be converted by using the function in the menu → *Image* → *type* → *8-bit*. This will convert the images in 8-bit grey level images. Each grey level represented on the image sequence will be associated with a 'material and radioactive element content' pair for the PTF creation. To create a 3D image in a raster format apply:

→ *File* → *new* → *image* → [select a 8-bit image type] → [set image size]

Note: Images with a higher tonal resolution than 8-bit are not required to create *Dosivox* models. The user does not have neither to fill the 256 tones (also called "grey values") available: only a number corresponding to the number of pair "material/radioactive element content" needed to represent the sample are required.

Image size: the number of pixels in width and height will be the number of sub-voxels in the x- and y- axis, respectively, the number of slices will become the number of sub-voxels in the z-axis. All the slices must have the same size in the x- and y- axis.

ImageJ brings various tools for modifying the images. To create a simple shape, the user can select a zone in the image with the selection tools (*Image*). Adjustments can be made using → *Edit* → *selection* → *specify*, and then fill in the zone with a grey level selected in the color picker tool with the function → *Fill* in the → *Edit*. It is furthermore possible to copy and paste selections from one slice to another. Each grey level represented on the image sequence will be associated with a 'material and radioactive element content' pair for the PTF creation. Many other functions in *ImageJ* allow to further modify the image size and slices, or the selections, cf. Rasband (1997-2012) and Schneider et al. (2012).

Most relevant functions are listed below (see also Fig. 16):

- Zoom (+):  + 
- Zoom (-):  + 
- Duplicate a slice: → *Edit* → *Selection* → *Select All* → *Edit* → *Copy* → [change slice number by pressing the cursor] → *Edit* → *Paste*
- Select a single object: 
- Define thickness of voxels in z (slice spacing) for 3D visualization on *ImageJ*: → *Image* → *Stacks* → *3D Project...*

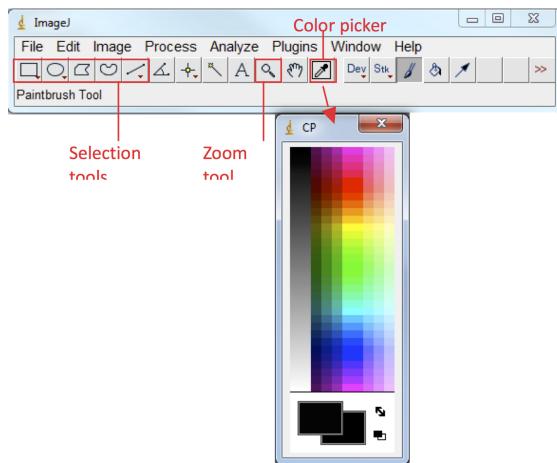


Figure 16: Selecting tools and changing colors in *ImageJ*

Once the 3D image (in 8-bit format) is created on the slice sequence, it is possible to visualize the result by using the function → *Image* → *Stack* → *3D Project* (Fig. 17) For a better visualisation of the 3D images the *ImageJ* plugin *3D viewer* (see *ImageJ* website) is recommended.

When the drawing of 3D image is completed, the image must be saved as a text image sequence:

→ *File* → *Save as ...* → *Image sequence*

Required options:

- Format = Text
- Digits = 4 (max. = 10,000 slices)
- **DO NOT** check *Use slice labels*: this option can bring some incoherences in the registration of the slice names.

Afterwards the stack of images is created (Fig. 18) as a series of z-text files. Each text file contains a data block of grey level values composing the corresponding slice whose dimensions are $x \times y$. This text image sequence can be used to create the sub-voxelised detector in the interface. To reload the 3D image in *ImageJ*, the plugin described in Sec. 5.4 can be used, but it is recommended to additionally save a 3D image in a TIFF format in *ImageJ* for modifications in the future.

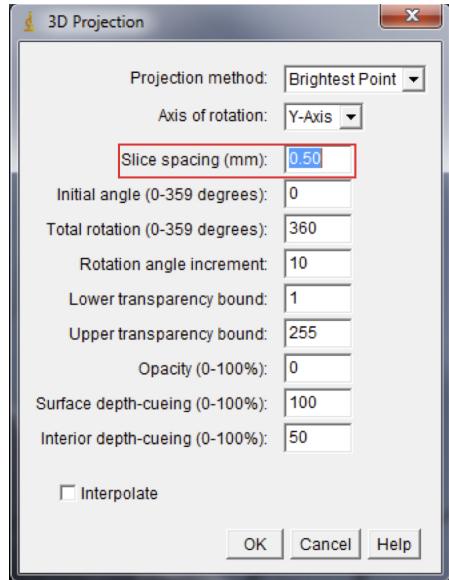


Figure 17: 3D projection tool in *ImageJ*

Fichier	Édition	Format	Affichage	?
dosiMtxLR0000	0	0	0	0
dosiMtxLR0001	0	0	0	0
dosiMtxLR0002	0	0	0	0
dosiMtxLR0003	0	0	0	0
dosiMtxLR0004	0	0	0	0
dosiMtxLR0005	0	0	0	0
dosiMtxLR0006	0	0	0	0
dosiMtxLR0007	0	0	0	0
dosiMtxLR0008	0	0	0	0
dosiMtxLR0009	0	0	0	102
dosiMtxLR0010	0	0	0	102
dosiMtxLR0011	0	0	0	102
dosiMtxLR0012	0	0	0	102
dosiMtxLR0013	0	0	0	102
dosiMtxLR0014	0	0	0	102
dosiMtxLR0015	0	0	0	102
dosiMtxLR0016	0	0	0	102
dosiMtxLR0017	0	0	0	102
dosiMtxLR0018	0	0	0	102
dosiMtxLR0019	0	0	0	102
dosiMtxLR0020	0	0	0	102
dosiMtxLR0021	0	0	0	102
dosiMtxLR0022	0	0	0	102
dosiMtxLR0023	0	0	0	102
dosiMtxLR0024	0	0	0	102
dosiMtxLR0025	0	0	0	102
dosiMtxLR0026	0	0	0	102

Figure 18: ASCII file of the resulting 3D model

6 Technical support

For questions on *DosiVox*, comments and improvements please do not hesitate to contact us:

Loïc MARTIN: loic.martin@u-bordeaux-montaigne.fr

Norbert MERCIER: norbert.mercier@u-bordeaux-montaigne.fr

DosiVox and DosiVox manuals for installation and use are available at :

<http://www.iramat-crp2a.cnrs.fr/spip/spip.php?article144>

7 Acknowledgements

The development of *DosiVox* has been supported by the Conseil Régional d'Aquitaine (France) through the program “DOSI-ART: Dosimetric Reconstruction in Archeological Artefacts and Sediments” conducted by N. Mercier (convention n°. 20121401005). The authors are grateful to J. Rink, S. Kreutzer and S. Incerti for improving the english and the form of this manual.

References

- Agostinelli, S., Allison, J., Amako, K., Apostolakis, J., Araujo, H., Arce, P., Asai, M., Axen, D., Banerjee, S., Barrand, G., Behner, F., Bellagamba, L., Boudreau, J., Broglia, L., Brunengo, A., Burkhardt, H., Chauvie, S., Chuma, J., Chytracek, R., Cooperman, G., Cosmo, G., Degtyarenko, P., Dell'Acqua, A., Depaola, G., Dietrich, D., Enami, R., Feliciello, A., Ferguson, C., Fesefeldt, H., Folger, G., Foppiano, F., Forti, A., Garelli, S., Giani, S., Giannitrapani, R., Gibin, D., Gómez Cadenas, J.J., González, I., Gracia Abril, G., Greeniaus, G., Greiner, W., Grichine, V., Grossheim, A., Guatelli, S., Gumplinger, P., Hamatsu, R., Hashimoto, K., Hasui, H., Heikkinen, A., Howard, A., Ivanchenko, V., Johnson, A., Jones, F.W., Kallenbach, J., Kanaya, N., Kawabata, M., Kawabata, Y., Kawaguti, M., Kelner, S., Kent, P., Kimura, A., Kodama, T., Kokoulin, R., Kossov, M., Kurashige, H., Lamanna, E., Lampén, T., Lara, V., Lefebure, V., Lei, F., Liendl, M., Lockman, W., Longo, F., Magni, S., Maire, M., Medernach, E., Minamimoto, K., Mora de Freitas, P., Morita, Y., Murakami, K., Nagamatu, M., Nartallo, R., Nieminen, P., Nishimura, T., Ohtsubo, K., Okamura, M., O'Neale, S., Oohata, Y., Paech, K., Perl, J., Pfeiffer, A., Pia, M.G., Ranjard, F., Rybin, A., Sadilov, S., Di Salvo, E., Santin, G., Sasaki, T., Savvas, N., Sawada, Y., Scherer, S., Sei, S., Sirotenko, V., Smith, D., Starkov, N., Stoecker, H., Sulkimo, J., Takahata, M., Tanaka, S., Tcherniaev, E., Safai Tehrani, E., Tropeano, M., Truscott, P., Uno, H., Urban, L., Urban, P., Verderi, M., Walkden, A., Wander, W., Weber, H., Wellisch, J.P., Wenaus, T., Williams, D.C., Wright, D., Yamada, T., Yoshida, H., Zschiesche, D., 2003. Geant4—a simulation toolkit. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 506 (3), 250–303. doi:doi:10.1109/TNS.2006.869826.
- Allison, J., Amako, K., Apostolakis, J., Araujo, H., Dubois, P.A., Asai, M., Barrand, G., Capra, R., Chauvie, S., Chytracek, R., Cirrone, G.A.P., Cooperman, G., Cosmo, G., Cuttone, G., Daquino, G.G., Donszelmann, M., Dressel, M., Folger, G., Foppiano, F., Generowicz, J., Grichine, V., Guatelli, S., Gumplinger, P., Heikkinen, A., Hrivnacova, I., Howard, A., Incerti, S., Ivanchenko, V., Johnson, T., Jones, F., Koi, T., Kokoulin, R., Kossov, M., Kurashige, H., Lara, V., Larsson, S., Lei, F., Link, O., Longo, F., Maire, M., Mantero, A., Mascialino, B., McLaren, I., Lorenzo, P.M., Minamimoto, K., Murakami, K., Nieminen, P., Pandola, L., Parlati, S., Peralta, L., Perl, J., Pfeiffer, A., Pia, M.G., Ribon, A., Rodrigues, P., Russo, G., Sadilov, S., Santin, G., Sasaki, T., Smith, D., Starkov, N., Tanaka, S., Tcherniaev, E., Tome, B., Trindade, A., Truscott, P., Urban, L., Verderi, M., Walkden, A., Wellisch, J.P., Williams, D.C., Wright, D., Yoshida, H., 2006. Geant4 developments and applications. IEEE Transactions on Nuclear Science 53 (1), 270–278. doi:doi:10.1109/TNS.2006.869826.
- Allison, J., Amako, K., Apostolakis, J., Arce, P., Asai, M., Aso, T., Bagli, E., Bagulya, A., Banerjee, S., Barrand, G., Beck, B.R., Bogdanov, A.G., Brandt, D., Brown, J.M.C., Burkhardt, H., Canal, P., Cano-Ott, D., Chauvie, S., Cho, K., Cirrone, G.A.P., Cooperman, G., Cortés-Giraldo, M.A., Cosmo, G., Cuttone, G., Depaola, G., Desorgher, L., Dong, X., Dotti, A., Elvira, V.D., Folger, G., Francis, Z., Galoyan, A., Garnier, L., Gayer,

M., Genser, K.L., Grichine, V.M., Guatelli, S., Guèye, P., Gumplinger, P., Howard, A.S., Hrivnacova, I., Hwang, S., Incerti, S., Ivanchenko, A., Ivanchenko, V.N., Jones, F.W., Jun, S.Y., Kaitaniemi, P., Karakatsanis, N., Karamitros, M., Kelsey, M., Kimura, A., Koi, T., Kurashige, H., Lechner, A., Lee, S.B., Longo, F., Maire, M., Mancusi, D., Mantero, A., Mendoza, E., Morgan, B., Murakami, K., Nikitina, T., Pandola, L., Paprocki, P., Perl, J., Petrović, I., Pia, M.G., Pokorski, W., Quesada, J.M., Raine, M., Reis, M.A., Ribon, A., Ristić Fira, A., Romano, F., Russo, G., Santin, G., Sasaki, T., Sawkey, D., Shin, J.I., Strakovský, I.I., Taborda, A., Tanaka, S., Tome, B., Toshito, T., Tran, H.N., Truscott, P.R., Urban, L., Uzhinsky, V., Verbeke, J.M., Verderi, M., Wendt, B.L., Wenzel, H., Wright, D.H., Wright, D.M., Yamashita, T., Yarba, J., Yoshida, H., 2016. Recent developments in Geant 4. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 835, 186–225. doi:doi:10.1016/j.nima.2016.06.125.

Kinsey, R.R., Dunford, C.L., Tuli, J.K., Burrows, T.W., 1996. The NUDAT/PCNUDAT program for nuclear data. 9. international symposium on capture gamma-ray spectroscopy and related topics, Budapest (Hungary), 8-12 Oct 1996 , 1–8URL: <https://digital.library.unt.edu/ark:/67531/metadc674863/>.

Kramida, A., Ralchenko, Y., Reader, J., NIST ASD Team, 2018. Nist atomic spectra database (version 5.0.0). National Institute of Standards and Technology, Gaithersburg, MD URL: <https://physics.nist.gov/asd>, doi:doi:10.18434/T4W30F.

Martin, L., 2015. Caractérisation et modélisation d'objets archéologiques en vue de leur datation par des méthodes paléo-dosimétriques. Simulation des paramètres dosimétriques sous Geant4. Ph.D. thesis. Université Bordeaux Montaigne. URL: <http://www.theses.fr/2015BOR30055>.

Martin, L., Incerti, S., Mercier, N., 2015a. DosiVox: Implementing Geant 4-based software for dosimetry simulations relevant to luminescence and ESR dating techniques . Ancient TL 33 (1), 1–10. URL: <http://author.ecu.edu/cs-cas/physics/Ancient-Timeline/upload/Martinetal-AncientTL-33-1-2015.pdf>.

Martin, L., Mercier, N., Incerti, S., Lefrais, Y., Pecheyran, C., Guerin, G., Jarry, M., Bruxelles, L., Bon, F., Pallier, C., 2015b. Dosimetric study of sediments at the Beta dose rate scale: characterization and modelization with the DosiVox software. Radiation Measurements 81, 134–141. doi:doi:10.1016/j.radmeas.2015.02.008.

Rasband, W., 1997-2012. ImageJ. U.S. National Institutes of Health, Bethesda, Maryland, USA URL: <https://imagej.nih.gov/ij/>.

Schneider, C., Rasband, W., Eliceiri, K., 2012. NIH Image to ImageJ: 25 years of image analysis. Nature Methods 671, 2089. doi:doi:10.1038/nmeth.2089.

Index

3D grid, 8
detector, 7, 9
EmMass, 8
Monte-Carlo algorithm, 7
Pilot Text File, 9
probe, 8
PTF, 9
PTF header, 13
solid detector, 8
voxel, 7
world, 10, 28