FUSION FOR ENERGY

Memorandum

idm@F4E Ref:	F4E_D_2LWY3V 2.0	Date	2020-06-12

Subject: **DoseMap calculator – Presentation of the program**

From: **Á. Cubí**

Department: A&C / Engineering Unit / ITER Department

Contact alvaro.cubi@ext.f4e.europa.eu,

To: **Everybody interested**

A new practical tool called DoseMap calculator has been developed to obtain the doses to a person or component stationed or travelling through a radiation field.
The input data provided to the program has the form of a VTK file (it supports also VTR/VTS files) containing the dose rate map and a csv file containing the position and trajectory information (positions, velocities and waiting times), as well as the dimensions of a prism that represents the person or component. The tool provides visual representations of the data and generates a csv file with information of the dose rates and integrated doses received at each position and step of the trajectory.
The tool is a python 3.6 program with its own integrated GUI. The program has been compiled into a Windows executable and therefore it can run without the need to install Python or any other third party program for internal use. The program can be used directly in Python.

What the tool does

This tool provides an easy and graphical way to extract information from a mesh based dose rate map (it can be used with other types of meshes like heating or neutron flux data). The user provides the program with a VTK/VTR/VTS file containing the mesh data of a dose rate map and a csv containing the information of a trajectory. This data can then be visualized in 3D. The average value inside a prism (of custom dimensions) or a point can be obtained at each point of the trajectory and also the integrated value along the trajectory

License

Installation

The program has been compiled as an executable for Windows machines. This way it can run without the need of installing the Python language or any third party module or software. Alternatively, the program can be executed with Python for example from an Anaconda command prompt. This report is going to explain the simpler way to execute the program which is through the compiled executable.

Download and unzip the DoseMap calculator folder. Inside 'DoseMap calculator/target/DoseMap calculator' find the 'DoseMap calculator.exe', it has the F4E logo. Double click the .exe and the program will start. Additionaly a shortcut of the .exe can be created by right-clicking it and selecting the option 'Create shortcut'. This shortcut can be placed anywhere (e.g. the desktop) and be renamed to simply 'DoseMap calculator'. Double click the shortcut to start the program.

How to compile a new version

In case there is the need to compile a new version in the future, some recommendations are given.

Compiling into an executable a Python script is often troublesome, the problems are solved in a case-by-case basis.

For the compilation the 'fbs' module was used. This module eases the compilation of programs that make use of the 'PyQt5' GUI module, one of the main modules employed in this program. The 'fbs' documentation includes tutorials on how to prepare the code for a first time compilation. To compile the DoseMap calculator after some changes have been applied is more straightforward. The changes in the code are applied in the 'src/main/python' folder inside the DoseMap calculator folder. After that from an Anaconda prompt (in the parent folder of 'src') the command: **fbs clean** will remove the previous compiled files. Then **fbs freeze --debug** will compile the new code.

There are a few things to note before finishing the compiling process. The python version that makes use of fbs should be 3.6 as the latest 3.7 version is giving more troubles. There is a problem with the compilation that I have not managed to solve but that I have been able to locate. The PyVista module makes a call to the Snoopy module. This module defines certain functions (see the debugging console) that cannot be compiled correctly. The workaround I have been able to come up is to not use **fbs freeze** but use instead the **fbs freeze --debug** option. Once the program is compiled trying to execute it will result in an error (see debug console) due to the lack of the folder 'Lib' inside 'target/DoseMap calculator', this folder should have been generated automatically. Manually creating the folder in the correct position will solve the error (mind the starting capital letter in 'Lib').

How to use it

When starting the program two windows will appear: the debugging console and the main GUI.

The debugging window can be minimized and ignored by the user. It displays information about the actions being performed by the code which may be useful for debugging or understanding errors.

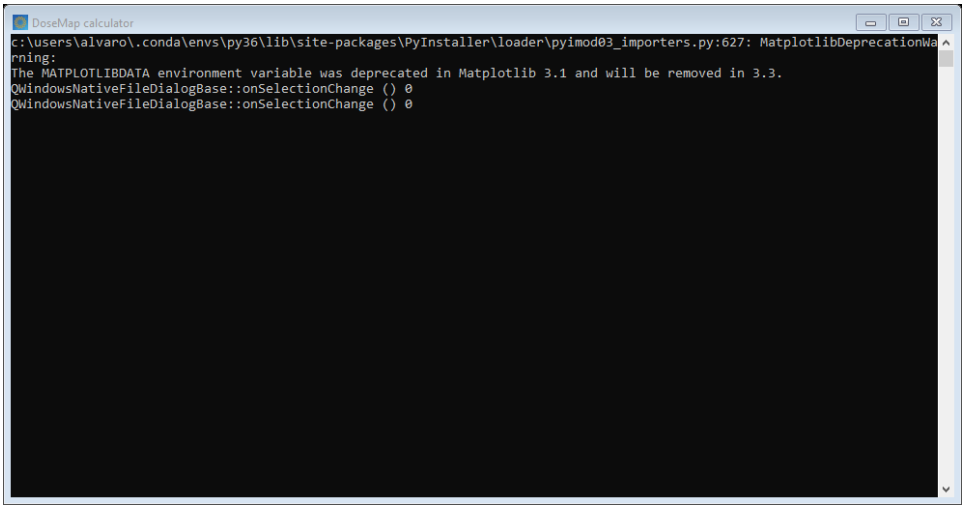


Figure 1 - Debugging window

The main window is where the user will interact with the program. Its size can be maximized or arbitrarily adjusted. The main window consists from top to bottom in a menu bar, a ComboBox bar (an interactive bar that allows the selection of several instances) and an interactive display.

The menu bar contains five menus: File, Plot, Parameters, Calculate and About us.

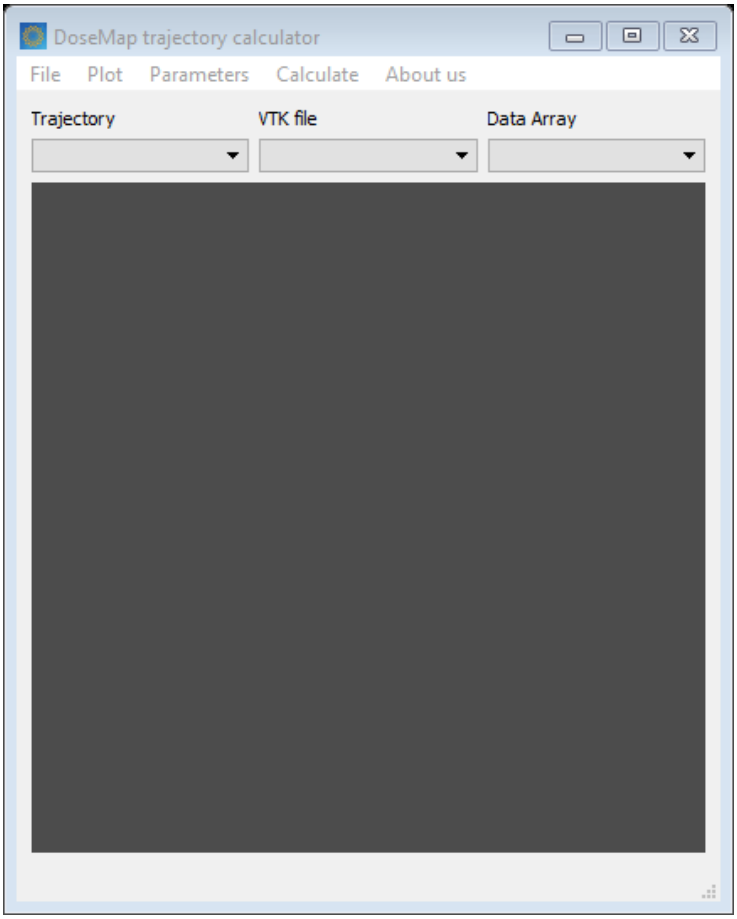


Figure 2 - Main window

File menu

- Load a trajectory csv file. Select a csv file with trajectory information as input for the program.
- Load a VTK file. Select a VTK file with the dose information as input for the program. It also allows the loading of .stl CAD files for visualization of geometries.

- Exit (shortcut: Ctrl + Q). Closes the program.

Plot menu

- Plot trajectory. Displays the trajectory selected in the ComboBox below as a group of black lines.
- Plot mesh. Displays the mesh or .stl file found in the ComboBox below. If the mesh has associated data arrays the array selected on the ComboBox will be plotted in a logarithmic 10 colours scale. Whenever a mesh is plotted it will have a slicing plane. This plane can be translated and reoriented by the user through the interactive display to generate an effective view of the desired geometry.
- Show prism. Displays a prism representing the boundaries of the person/component that will be used for the dose calculation. The prism can be displaced through the selected trajectory interacting with a sliding bar widget.
- Plot sampled trajectory. Displays the selected trajectory sampled over the data of the selected data array.
- Clear slicing planes. Removes the slicing planes from the display to obtain a clearer view.
- Clear plots (shortcut: Ctrl + C). Removes all the plots, slicing planes and widgets from the interactive display.

Parameters menu

- Input XYZ bounds. Modifies the lengths in the three dimensions of the prism that will be used in the calculation. The dose at every position is calculated as the average of all the doses found inside of the prism boundaries.
- Set step distance. Modifies the step value. This value is the maximum distance that will be computed between two points of the trajectory (i.e. the step in the dose integration).
- Display trajectory table. Displays the list with the points of the selected trajectory and the waiting times, velocities and comments but it does not allow modifications.

Calculate menu

- Calculate doses. Performs the calculation that produces the csv output file according to the trajectory, VTK file and data array selected in the ComBox bar.

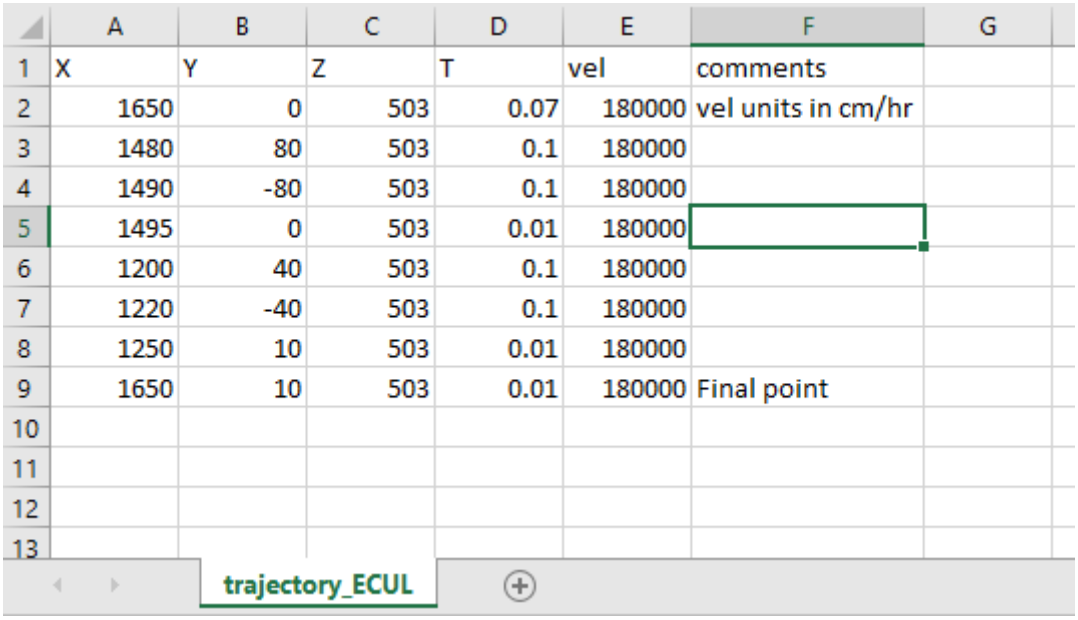
About us menu

This menu contains information about Fusion for Energy and the members of the neutronics team.

Input and output format

The VTK dose rate map file should contain the mesh and data arrays that will be used for the calculation. A VTK file can contain two types of data arrays: cell data arrays and point data arrays. The calculation is going to be performed on cell data arrays. If the loaded VTK file contains point data arrays they will automatically be converted to cell data arrays in order to be able to use them. It is recommended to provide VTK files with cell data arrays only.

The trajectory file that is loaded into the program is a csv file of the form seen in the ECUL_traj.csv file and in Figure 3. '*X*', '*Y*', '*Z*' columns show the positions of the trajectory. Column '*T*' shows the waiting time the component will spend at each position. Column '*vel*' shows the velocity of the component when traveling from the current point to the next one. The last row shows the last point and therefore the '*vel*' value at the last row will not be used. It is possible to add text comments in the right-most column of the file.



	A	B	C	D	E	F	G
1	X	Y	Z	T	vel	comments	
2	1650	0	503	0.07	180000	vel units in cm/hr	
3	1480	80	503	0.1	180000		
4	1490	-80	503	0.1	180000		
5	1495	0	503	0.01	180000		
6	1200	40	503	0.1	180000		
7	1220	-40	503	0.1	180000		
8	1250	10	503	0.01	180000		
9	1650	10	503	0.01	180000	Final point	
10							
11							
12							
13							

Figure 3 - Example of csv input

It is important to note that there is no indication of the units that should be used when inputting the positions, times and velocities. The reason for this is that the VTK file holding the dose data could be in different units. It is the responsibility of the user to ensure that the units employed in the csv file, the VTK file and the step distance are consistent with each other.

The output csv file generated with the 'Calculate doses' button includes the same unmodified columns from the input file and four new columns, see Figure 4. The four new columns are:

- 'Instant dose rate' column displays the average value inside the prism boundaries at the different points of the trajectory.
- 'Wait dose' show the values of the 'Instant dose rate' column multiplied by the time.
- 'Move dose' shows the dose accumulated during the move from the point to the next one (last point will have zero move dose).
- 'Integral dose' is the cumulative sum of all the previous wait and moving doses.

	A	B	C	D	E	F	G	H	I	J	K
1	X	Y	Z	T	vel	comments	instant dose rate	wait dose	move dose	integral dose	
2	1650	0	503	7.00E-02	180000	vel units in cm/hr	4.66E-06	3.26E-07	2.55E-07	5.81E-07	
3	1480	80	503	1.00E-01	180000		3.64E-04	3.64E-05	3.58E-07	3.73E-05	
4	1490	-80	503	1.00E-01	180000		3.69E-04	3.69E-05	1.69E-07	7.44E-05	
5	1495	0	503	1.00E-02	180000		3.32E-04	3.32E-06	9.03E-07	7.87E-05	
6	1200	40	503	1.00E-01	180000		7.29E-04	7.29E-05	3.51E-07	1.52E-04	
7	1220	-40	503	1.00E-01	180000		7.30E-04	7.30E-05	2.33E-07	2.25E-04	
8	1250	10	503	1.00E-02	180000		6.92E-04	6.92E-06	9.12E-07	2.33E-04	
9	1650	10	503	1.00E-02	180000	Final point	4.55E-06	4.55E-08	0.00E+00	2.33E-04	
10											
11											

Figure 4 - Example of csv output

How the time integral calculation is performed

This code is mainly based on the PyVista Python module. This module allows read and write meshes in VTK formats and operate them. The idea behind the dose time integral calculation is to divide the trajectory into a set of midpoints and obtain the average mesh data value inside a prism at each of those midpoints.

The program uses the step value as the maximum distance that two midpoints are allowed to have. For example if two trajectory points are (0, 0, 0) and (12, 0, 0) and the step value is set to 5 the resulting midpoints would be: (0, 0, 0), (5, 0, 0), (10, 0, 0) and (12, 0, 0). The distances between them are 5, 5 and 2. If two points from the input csv file are closer than the step value no additional midpoint would be created between them.

The step value should be lower than the length of a cell in the VTK mesh. If the step value is higher there will be cells in the trajectory that may not be included in the calculation; this may lead to unphysical results in the presence of large dose field gradients. The code is sufficiently fast to allow small step distances (0.1 times the cell length) for typical cell sizes used in ITER calculations (5 to 50 cm), but if the trajectory is very long or the prism boundaries are very large the dose calculation may require a significant time to compute (several minutes). The user could then try with bigger step values like 0.5 or 0.8 cell lengths.

Example of usage

This section serves as a tutorial where all the functions of the program are used. A demonstration video where the same actions are performed is also available.

The program is initialized by double clicking in the shortcut. Through the File menu the trajectory ECUL_traj.csv file is loaded (see Figure 3). Note that the trajectory name is now displayed in the ComboBox. The C-LITE.stl geometry file is now loaded through the '*Load a VTK file*' option in the File menu. Note that the C-LITE mesh name is now displayed in the ComboBox. With the '*Plot mesh*' and '*Plot trajectory*' options the interactive display now shows that the trajectory is placed where it was intended, see Figure 6. The '*Clear slicing planes*' option is used for clarity of the plot.

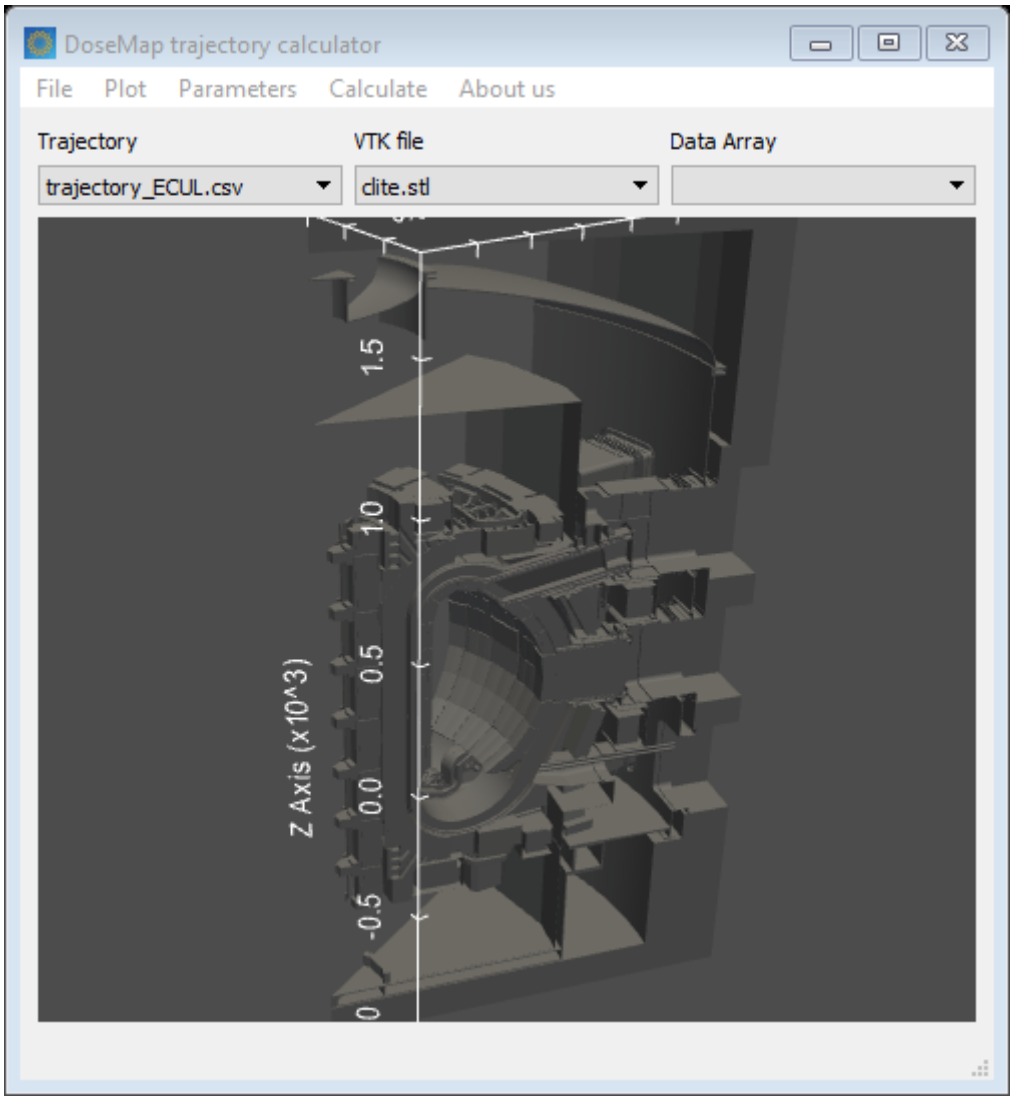


Figure 5 - Plotting a .stl CAD file

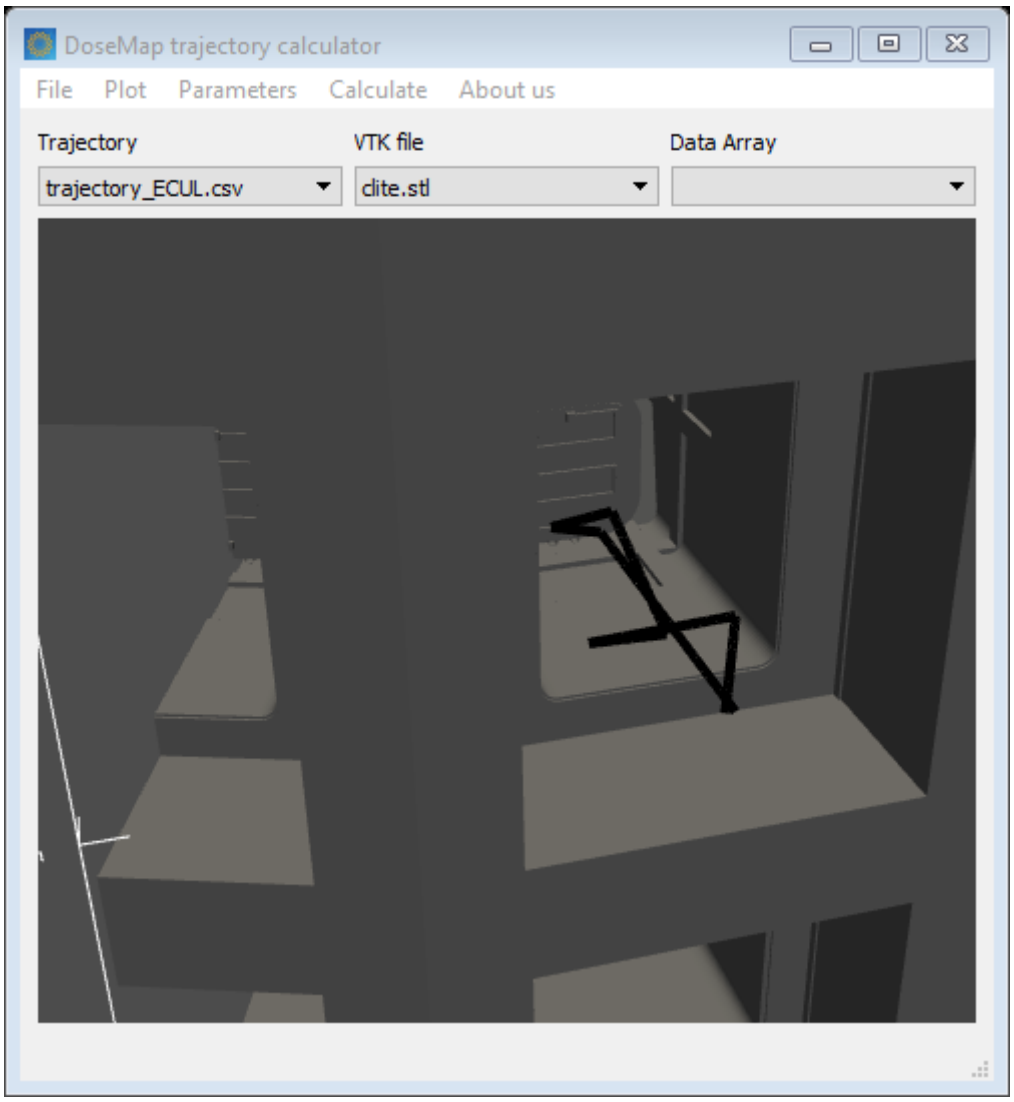


Figure 6 - Detail of the geometry where the trajectory is located

Now the VTK file ECUL_dose.vtk containing the dose map information is loaded the same way the C-LITE.stl was loaded. The option '*Clear plots*' is used, the ECUL_dose mesh is selected in the ComboBox along with the data array of name '*scalars*'. This data array shows neutron dose rate in Sv/h.

By selecting the '*Plot sampled trajectory*' and '*Plot mesh*' the dose rate map and the dose rate map information sampled over the trajectory can be seen in the display, see Figure 7.

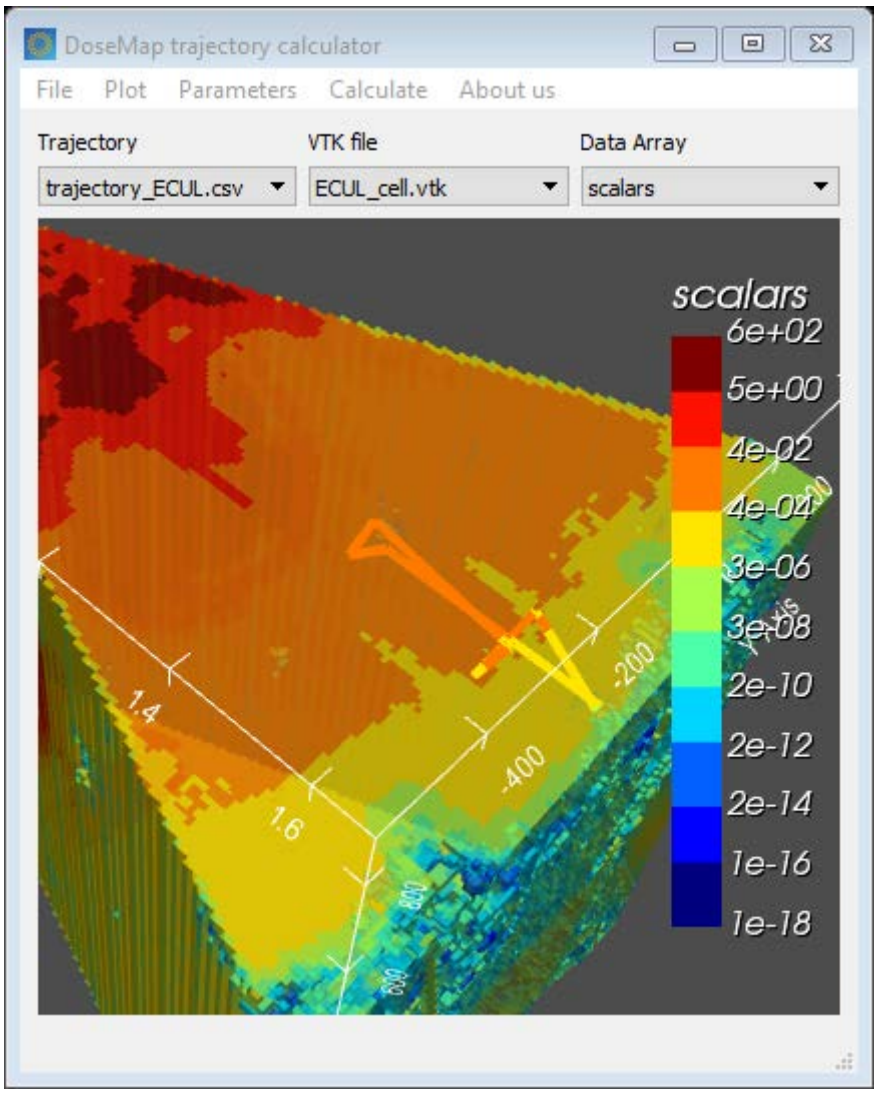


Figure 7 - Sampled trajectory and dose mesh data detail

In the Parameters menu the '*Input XYZ boundaries*' option is clicked. The next values are inputed: X=15, Y=15, Z=25. In the Plot menu select the '*Show prism*' option to display those boundaries along the trajectory, see Figure 8.

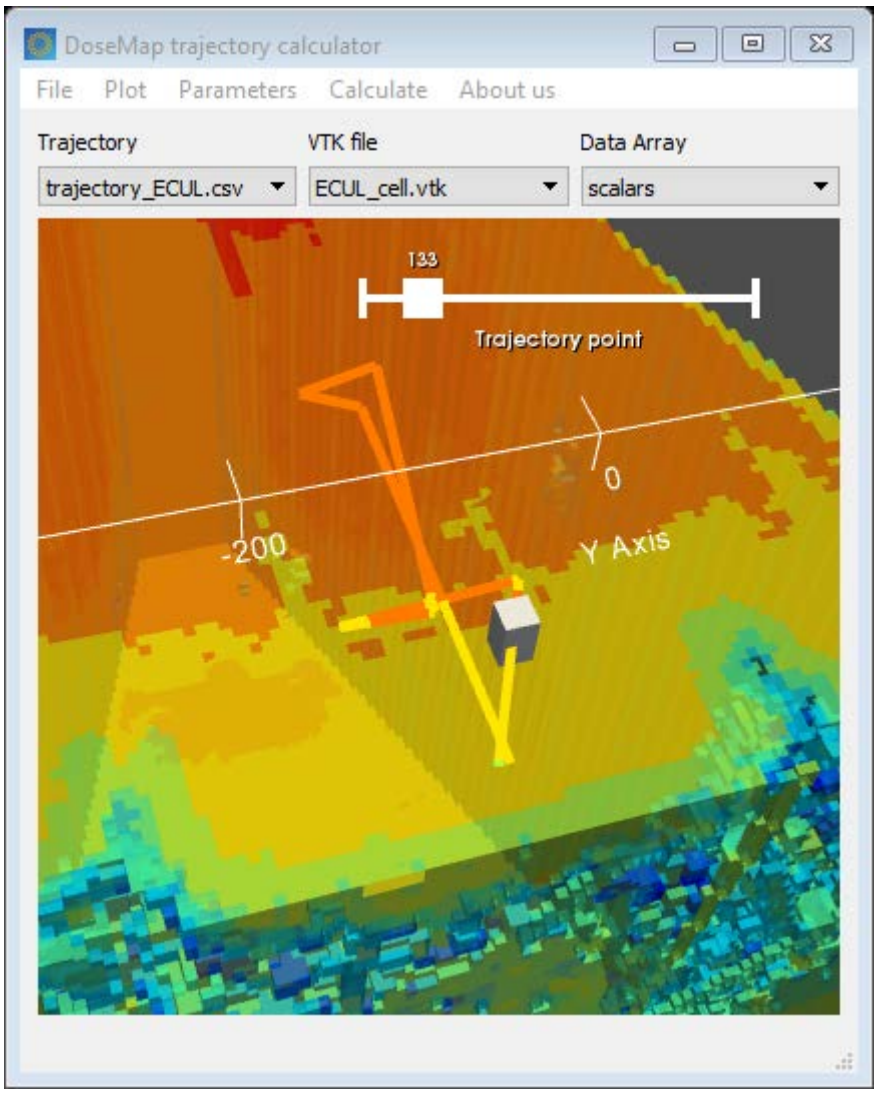


Figure 8 - Prism boundaries plotted over the trajectory

The size of the cells of this mesh (20 cm each side) is much higher than the default 1 step value (same units as the mesh, centimetres). Therefore we can increase the step value to 5 with the '*Set step distance*' option found in the Parameters menu.

Now that all the Parameters have been set clicking the '*Calculate doses*' in the Calculate menu will produce the output file with all the dose information (see Figure 4), which is stored in the same folder as the trajectory file.

Recommendations and warnings

The program gives no indication of the units that are being used. The units should be consistent among all the data inputs: mesh file, mesh values, csv trajectory information and in-program user parameters.

It is recommended to use a step value for the calculation lower than the length of the cells in the data mesh to avoid jumps between groups of cells. The initial recommended value is 0.1 times the length of the cell. If the computation time is too high (several minutes) the user can increase the step value.

It is recommended to provide the program with a mesh data file that has the data arrays of the type 'cell data'. Any 'point data' array will be automatically converted to cell data.

While the GUI of this tool is very useful for the visualization of meshes and geometries it does not intend to substitute other programs like Paraview or CATIA. The points of the trajectory should be created in one of those third party codes and use this tool's GUI for verification and visualization purposes.

Further developments

When using the program, the users will surely make suggestions for improvement that may be implemented in the future. Further developments could include the possibility to use a mesh grid instead of a prism for the

calculations and the introduction of an option to calculate doses at single points without the need to input velocities.