

# **MCNScript**

Open-source shell program for running Monte Carlo simulations with MCNelectron

**v1.0.5**

User's Manual

by Andrius Poškus

(Vilnius University, Department of Solid State Electronics)

2017-03-10

Copyright © 2016 – 2017 by Andrius Poškus

E-mail: [andrius.poskus@ff.vu.lt](mailto:andrius.poskus@ff.vu.lt) or [andrius.poskus@live.com](mailto:andrius.poskus@live.com)

Web: <http://web.vu.lt/ff/a.poskus/mcnelectron/>

## Contents

1. Introduction .....	1
2. Introduction to the user interface of MCNScript.....	1
3. Editing the MCNelectron command-line arguments and running Monte Carlo simulations .....	3
4. Parameters and controls related to particle track statistics and track filtering .....	5
4.1. Track statistics .....	7
4.2. Track filtering.....	7
4.3. Track segment selection .....	10
4.4. Examples of particle track display in the MGED Graphics Window.....	12
5. RT script file and MGED startup .....	16
6. Graphical parameters.....	19
6.1. Viewing cube and eye position parameters .....	19
6.2. Graphical parameters controlling the size limits, display of the source, and background color .....	20
6.3. Using a cut-away plane .....	20
6.4. Specifying the amount of data to show in MGED.....	20
6.5. Properties of materials.....	24
7. Manipulating the view in the MGED environment.....	25



## 1. Introduction

MCNScript is an open-source dialog-based “shell” program for running Monte Carlo simulations with MCNelectron, as well as for organizing MCNelectron input and output files and creating graphics rendering scripts for visualization of the simulation geometry and particle tracks. The raytracing scripts (with the file name extension “.RT”) created by MCNScript must be loaded by MGED (Multi-Device Geometry Editor – a part of the open-source software package BRL-CAD), which generates raytraced or wireframe images from 3D models using the constructive solid geometry technique.

MCNScript is a Windows application (the oldest supported version of Windows is XP SP2; all newer versions of Windows are supported). Both 32-bit and 64-bit versions of MCNScript are available. MCNScript is written in C++, using Microsoft Foundation Classes (MFC). MCNScript is developed by Andrius Poškus (Vilnius University, Faculty of Physics, Department of Solid State Electronics). The user interface of MCNScript makes extensive use of the MFC Grid Control v2.26 written by Chris Maunder. MCNScript is distributed under the GNU General Public License.

MCNScript and MCNelectron are available from <http://web.vu.lt/ff/a.poskus/mcnelectron/>. BRL-CAD can be downloaded from <http://brl-cad.org/>.

## 2. Introduction to the user interface of MCNScript

MCNScript is started by running the executable file MCNScript.exe or MCNScript\_x64.exe (the former is 32-bit, the latter is 64-bit). The main window of MCNScript shows the names of MCNelectron input and output files, the MCNelectron command line and various graphical parameters. All displayed parameters form a “parameter set”. Parameter sets can be selected, created, deleted or renamed using the top row of controls (see Fig. 2.1).

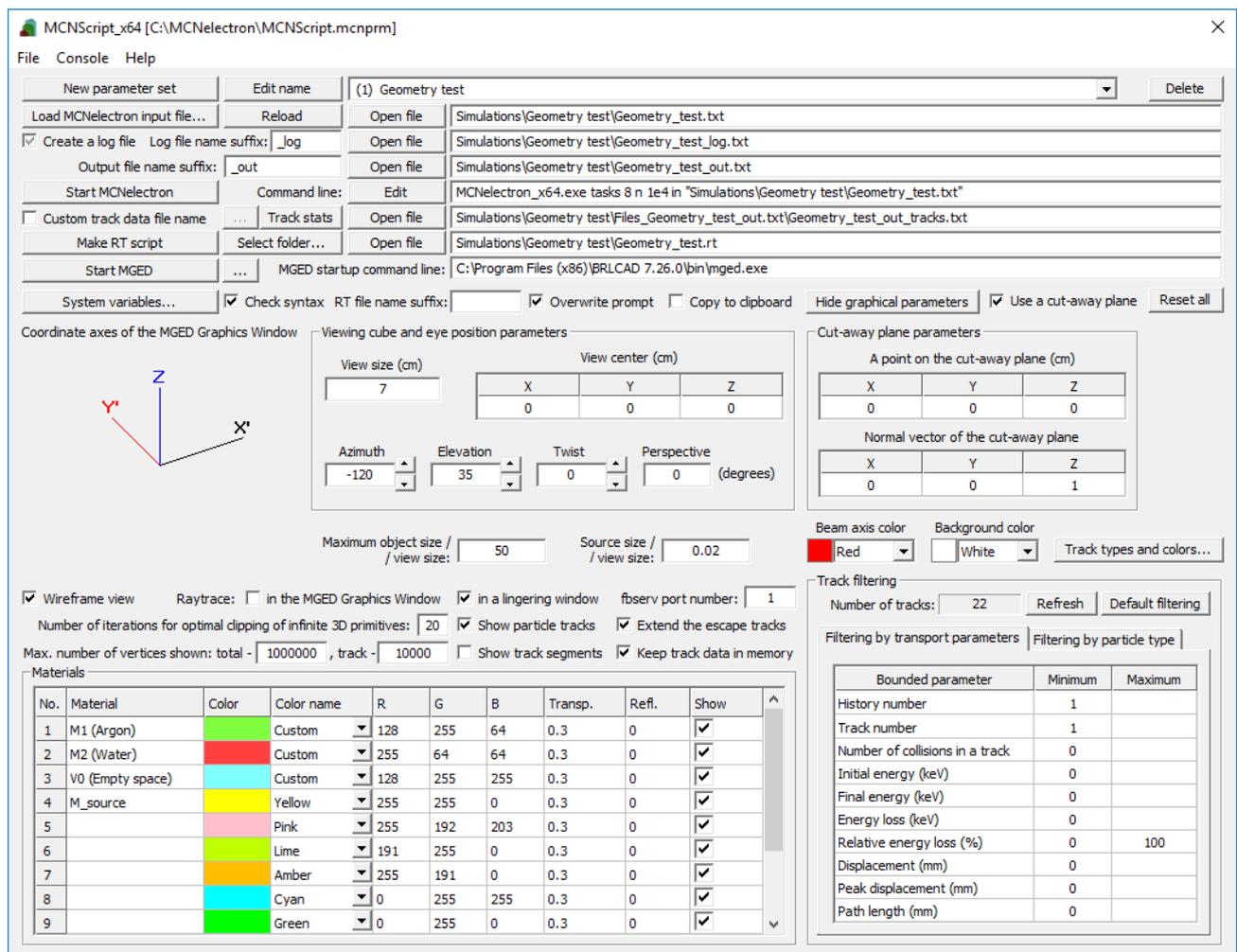


Fig. 2.1. An example of the MCNScript main window.

The data of all parameter sets are stored in an ASCII file with extension “.MCNPRM”. Further on, this file will be called “parameter file”. The name of the current parameter file is shown in the title bar of the main window (see Fig. 2.1). Parameter files can be opened or saved using the “File” menu. The name of the parameter file can be specified on the command line used to start MCNScript. If MCNScript is started without any command-line arguments, the parameter file will be assigned the default name “MCNScript.mcnprm” (if there is a file with that name in the folder where the executable file “MCNScript.exe” is located, then it will be opened automatically, otherwise it will be created).

Although the main function of MCNScript is running the simulations with MCNelectron and setting up the initial view of the simulated system geometry and particle tracks in the MGED window, it can also perform elementary statistical analysis of the particle track data calculated by MCNelectron, as well as filter the particle tracks according to various criteria. The calculated track statistics are shown in a dialog window that is opened by clicking the button “Track stats”, which is in the sixth row of controls of the MCNScript main window (see Fig. 2.1). The same dialog window contains two tabs with the data used for track filtering (those tabs are duplicated at the bottom right of the MCNScript main window). The tab “Filtering by transport parameters” contains a grid control with transport parameters (this tab is the active one by default; it is visible in Fig. 2.1), and the tab “Filtering by particle type” contains a set of controls for selecting the types of secondary particles for calculation of the statistics and for display in the MGED Graphics Window.

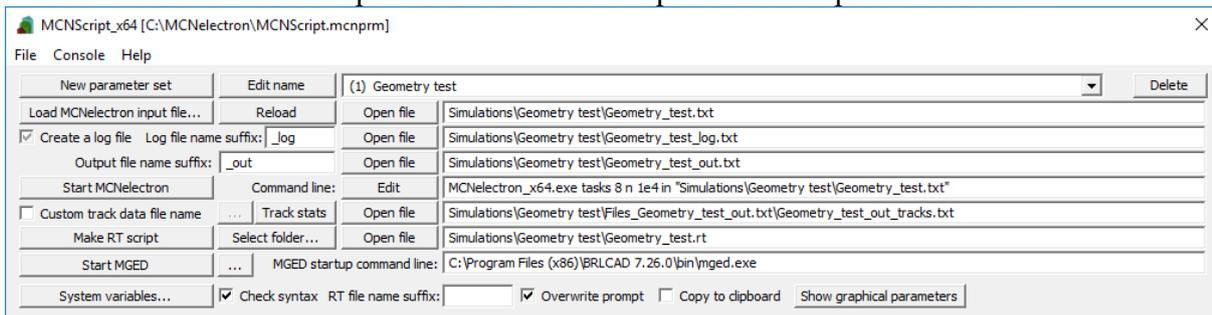
All the parameters shown in the MCNScript main window can be grouped into three categories according to the time of their modification in a single simulation cycle:

- 1) the MCNelectron command-line arguments,
- 2) the name of the track data file and the RT script file, as well as the MGED command line and track filtering data,
- 3) the graphical parameters.

The parameters of the first group are needed to run a simulation with MCNelectron (they are set in rows 2 – 5 of controls of the MCNScript main window). The parameters of the second group are used to tell the program where the data of the tracks that have to be displayed in the MGED Graphics Window are located, what are the criteria to be used for selecting the tracks that should be shown, where the output RT script should be placed and how MGED should be started (those parameters are set in rows 6 – 8 of controls of the MCNScript main window).

The graphical parameters control the initial view of the system geometry and particle tracks after loading the RT script created by MCNScript. The view may be modified afterwards using various MGED commands or by re-creating the RT script with a different set of graphical parameters and then reloading it in MGED. The graphical parameters occupy the bottom two thirds of the MCNScript main window (see Fig. 2.1). They can be hidden using the button “Hide graphical parameters”. After clicking this button, the text on it changes to “Show graphical parameters” (see Fig. 2.2). After clicking this button again, the MCNScript main window will be restored to its initial size (with the graphical parameters shown). If an invalid value is entered in any of the numerical text boxes, the previous value is restored. The button “Reset all” (see Fig. 2.1) is used to set all graphical parameters to their default values.

A more detailed description of the mentioned parameters is presented below.



**Fig. 2.2.** The reduced version of the MCNScript main window (without the graphical parameters).

### 3. Editing the MCNelectron command-line arguments and running Monte Carlo simulations

The names of the MCNelectron input file, log file and output file are shown in the three text boxes that are at the end of the second, third and fourth row of controls of the MCNScript main window (see Fig. 2.1). The remaining command-line arguments can be specified in the dialog window “MCNelectron command-line arguments”, which is opened by clicking the button “Edit”, which is in the fifth row of controls (see Fig. 2.1). An example of that dialog window is shown in Fig. 3.1.

By default, the name of the MCNelectron output file is formed by appending the suffix “\_out” to the name of the MCNelectron input file. If this suffix is specified, then the name of the output file will be omitted from the command line (see Fig. 2.1). The mentioned suffix can be modified in the text box “Output file name suffix” (see Fig. 2.1 and Fig. 3.1). If that text box is cleared of any text, then it becomes possible to enter an arbitrary name of the MCNelectron output file directly in the text box where that file name is shown (in this case, the output file name will be specified on the command line, even if that name is the same as the default name). The file name suffix may include the two-character string “%d”, which has a special meaning: if it is present, then it will be replaced by the sequence number of the current parameter set (this number is shown in parentheses before the name of the parameter set in the dropdown list, which is at the top of the MCNScript main window).

By default, the log file name is formed by appending the suffix “\_log” to the name of the MCNelectron input file. This suffix can be modified in the text box “Log file name suffix” (see Fig. 2.1 and Fig. 3.1). In all other respects, the name of the log file is treated in the same way as the name of the output file (see the previous paragraph). The log file name may be specified on the command line only when the tri-state checkbox “Create a log file” is checked or when it is indeterminate (as in Fig. 2.1). This checkbox corresponds to the MCNelectron command-line argument “writeLog”. If that checkbox is indeterminate, this keyword will be omitted from the command line, i.e., the corresponding parameter will be assigned the default value. The checkbox “Create a log file” is duplicated in the MCNScript main window (see Fig. 2.1).

The tri-state checkbox “Check the simulation setup for errors” and the checkbox “Quit after checking” correspond to the MCNelectron command-line argument “check”. If the first of the mentioned checkboxes is indeterminate and the second mentioned checkbox is unchecked (as in

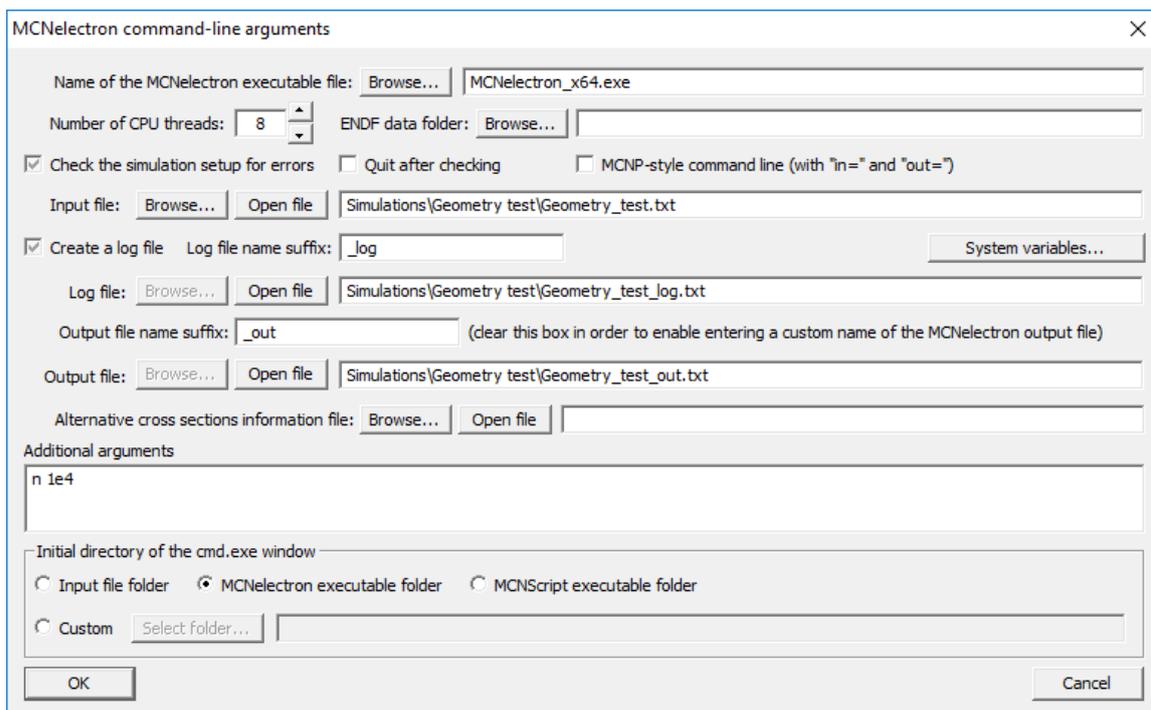
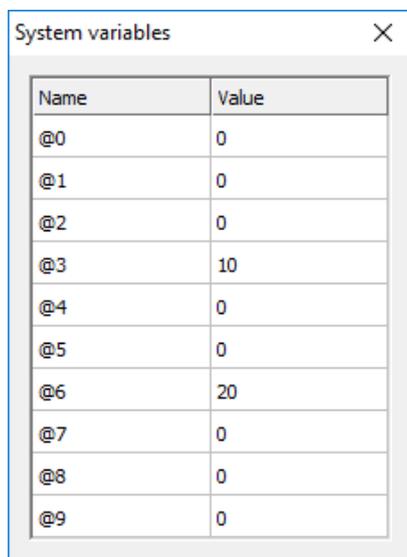


Fig. 3.1. An example of the dialog window with command-line arguments of MCNelectron.

Fig. 3.1), then the keyword “check” will not be specified on the command line and MCNelectron will use the default setting “check 1”. If the checkbox “Check the simulation setup for errors” is not checked, then the command line will contain “check 0” (in this case, the checkbox “Quit after checking” will have no effect and MCNelectron will quit after processing the input file). If the checkbox “Check the simulation setup for errors” is checked, then checking or un-checking the checkbox “Quit after checking” corresponds to the option “check -1” or “check 1”, respectively. The command-line option “check -1” is also used if the checkbox “Check the simulation setup for errors” is indeterminate and the checkbox “Quit after checking” is checked.



Name	Value
@0	0
@1	0
@2	0
@3	10
@4	0
@5	0
@6	20
@7	0
@8	0
@9	0

**Fig. 3.2.** An example of the dialog window with values of the MCNelectron system variables.

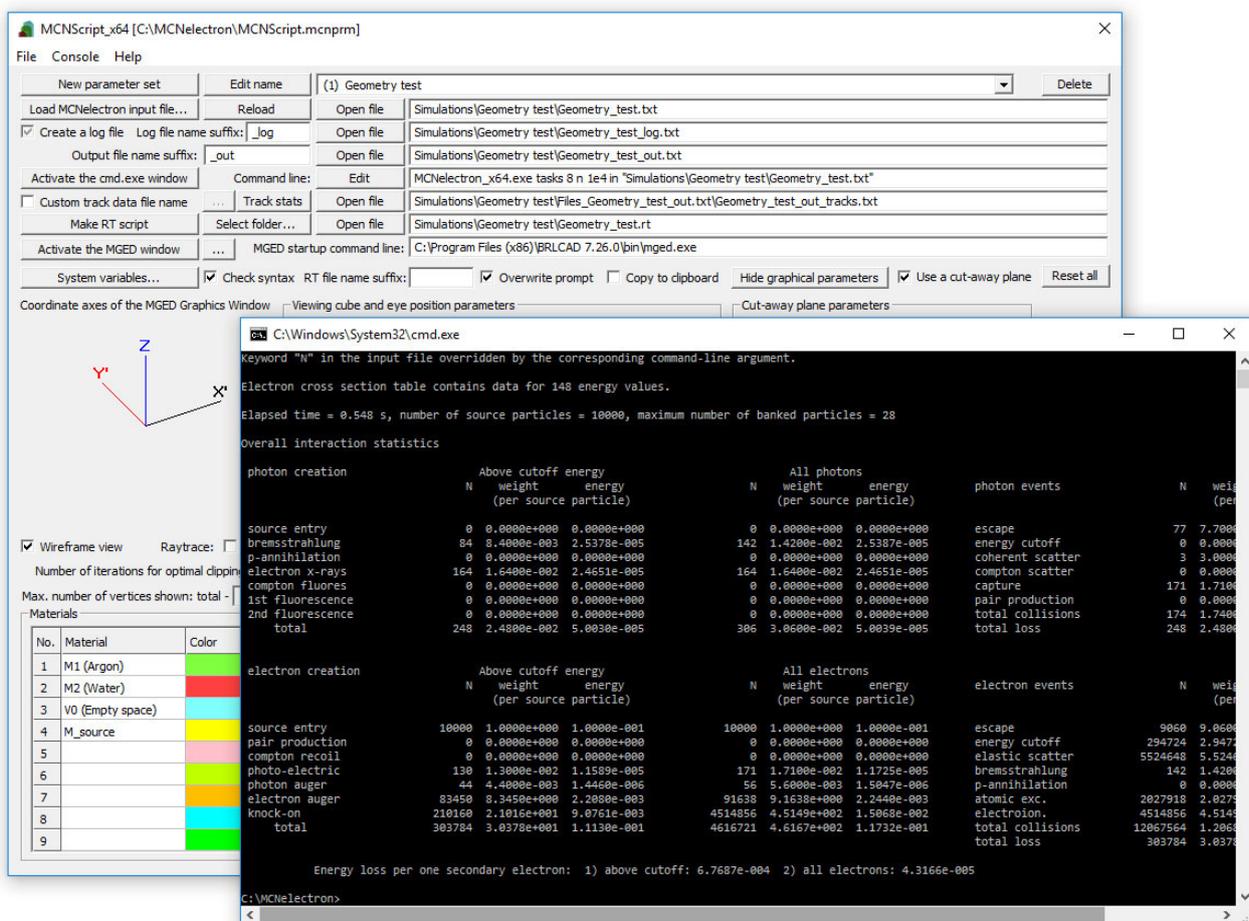
The button “System variables...” (see Fig. 3.1) is used to open the dialog window where the values of the MCNelectron system variables may be specified (see Fig. 3.2). Those variables may be used in the user programs embedded in the MCNelectron input file (see the MCNelectron user manual for details). The “System variables” dialog window is modeless, which means that it may be kept visible while interacting with the other windows of the MCNScript graphical user interface. The button “System variables...” is also present in the MCNScript main window (see Fig. 2.1).

The remaining MCNelectron command-line arguments, which can be modified in the dialog window shown in Fig. 3.1, include the number of CPU threads, the name of the folder with the ENDF/B cross section data, and all other command-line arguments that are understood by MCNelectron or MCNelectron\_CUDA (they are described in the MCNelectron user manual).

The entire MCNelectron command line is shown in the text box at the end of the fifth row of controls of the MCNScript main window (see Fig. 2.1).

The specified MCNelectron input and output files can be opened by clicking the buttons “Open file”, which are to the left of the text boxes with the file names (see Fig. 2.1 and Fig. 3.1). This action is equivalent to double-clicking the file name in the Windows File Explorer.

After specifying the names of input and output files and entering all other necessary command-line arguments, a Monte Carlo simulation can be started. This is done by clicking the button “Start MCNelectron”, which is at the beginning of the fifth row of controls of the MCNScript main window (see Fig. 2.1). After clicking this button, the Windows command-line interpreter “cmd.exe” window is opened, where all messages of MCNelectron are output (see Fig. 3.3). After finishing the simulation run, the corresponding cmd.exe window is not closed automatically (it must be closed manually). While the cmd.exe window is open, the text on the button “Start MCNelectron” is changed to “Activate the cmd.exe window” (see Fig. 3.3). Afterwards, if the cmd.exe window is minimized or covered by other windows, this button can be used to bring it to the foreground. Each created cmd.exe window is “attached” to the corresponding parameter set. If there are several cmd.exe windows, each of them can be activated by selecting the corresponding parameter set and then clicking the button “Activate the cmd.exe window”. After closing the cmd.exe window, the text on the mentioned button becomes “Start MCNelectron” again. It is possible to “detach” a cmd.exe window from the current parameter set. This is done by right-clicking the button “Activate the cmd.exe window” and selecting “Reset this button” from the context menu. Then the text on that button becomes “Start MCNelectron” again. Afterwards, it will be possible to use this button for starting another simulation without closing the “detached” cmd.exe window.



**Fig. 3.3.** An example of the cmd.exe window created by clicking “Start MCNelectron” in the MCNScript main window. When the mentioned button is clicked, it is replaced by “Activate the cmd.exe window”.

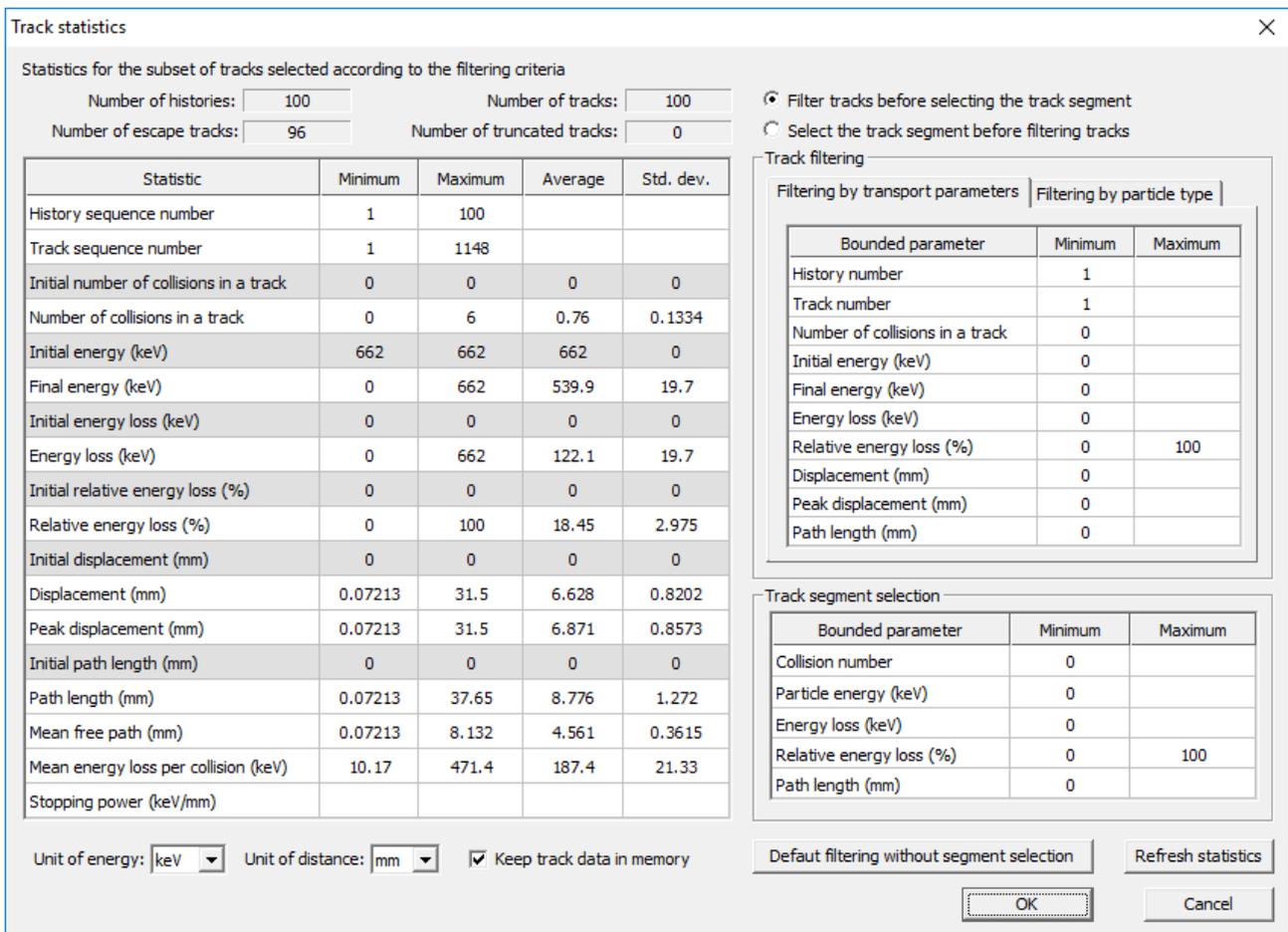
#### 4. Parameters and controls related to particle track statistics and track filtering

The sixth row of controls of the MCNScript main window contains the controls related to particle track data calculated by MCNelectron (see the MCNelectron user manual for the description of the format of the track data files and the MCNelectron parameters needed for output of track data). The name of the track data file created by MCNelectron is uniquely defined by the output file name: MCNelectron forms the name of the track data file by appending “\_tracks.txt” to the output file name, and puts the track data file in the subfolder with the name “Files\_<output\_file\_name>”. Consequently, this is the name that is shown in the text box at the end of the sixth row of controls of the MCNScript main window, if the checkbox “Custom track data file name” is not checked (see Fig. 2.1). If the latter checkbox is checked, then it is possible to enter an arbitrary file name in the mentioned text box, or to browse for a track data file using the button “...”, which is in the same row of controls (see Fig. 2.1).

The specified file with track data can be opened by clicking the button “Open file”, which is to the left of the corresponding text box with the file name (see Fig. 2.1). This action is equivalent to double-clicking the file name in the Windows File Explorer.

The button “Track stats”, which is in the sixth row of controls of the MCNScript main window (see Fig. 2.1), is used for selecting a subset of particle tracks according to various criteria and for elementary statistical analysis of that subset. An example of the “Track statistics” dialog window, which is opened by clicking the button “Track stats”, is shown in Fig. 4.1.

The top two of the four text boxes at the top of the “Track statistics” dialog window contain the total number of histories and tracks satisfying the filtering criteria. One “history” is the sequence of all interaction events started by a single source particle. It consists of the source particle track and any number of secondary particle tracks. Consequently, the number of tracks is in general greater



**Fig. 4.1.** An example of the dialog window with particle track statistics. In this example, the incident particles are photons with energy 662 keV, which are incident normally on a layer of iron with thickness 1 cm and density 7.874 g/cm<sup>3</sup>. The statistics have been calculated for the source photons only.

than the number of histories. The filtering criteria are applied to individual tracks (not histories). The text box “Number of histories” indicates the number of histories containing at least one track that satisfies the filtering criteria. The next two text boxes contain the numbers of escape tracks and truncated tracks. “Escape tracks” are the tracks that end with an escape event (i.e., a collision that is followed by escape of the particle from the system). “Truncated tracks” are the tracks whose last collision is not the last collision of the complete track. A track can be truncated either due to track segment selection (see Section 4.3), or by MCNelectron during the Monte Carlo simulation: as explained in the MCNelectron user manual, one of the parameters controlling the output of track data is the maximum number of collisions in a history (if that number is exceeded, then all unfinished tracks of the current history are truncated).

The grid control on the left shows several descriptive statistics characterizing the selected subset of particle tracks. The other controls are used for defining the mentioned subset of tracks and for refreshing the statistics. The two dropdown lists at the bottom left of the “Track statistics” dialog window are used for selecting the units of energy and distance in the three grid controls visible in Fig. 4.1. The checkbox “Keep track data in memory” is used to find the balance between the speed of calculating the track statistics and the amount of random access memory (RAM) used. If it is unchecked, then track data will not be kept in RAM; instead, they will be loaded from file every time when they are needed. If the checkbox “Keep track data in memory” is checked, then the entire dataset stored in the track data file will be kept in memory for subsequent use. In this case, the track data will be loaded from file only for the initial calculation, or after the track data file is modified. If the track dataset is very large (millions of collisions), then re-calculating the track statistics with the latter option takes much less time, but requires much more RAM.

A more detailed description of the controls related to track statistics and track filtering is presented below.

#### 4.1. Track statistics

The grid control with the track statistics contains the minimum, maximum and average values of various physical quantities characterizing a track, as well as the standard deviations of the mentioned average values. The mentioned quantities are the following:

- 1) history sequence number,
- 2) track sequence number,
- 3) initial number of collisions in a track (can be non-zero only for track segments; see Section 4.3);
- 4) total number of collisions in a track,
- 5) initial energy of the track (i.e., the energy of the particle immediately after its creation),
- 6) final energy of the track (i.e., the particle energy after the last collision of the track),
- 7) initial energy loss (can be non-zero only for track segments; see Section 4.3),
- 8) total energy loss (i.e., the difference of the initial and final energies),
- 9) initial relative energy loss (can be non-zero only for track segments; see Section 4.3),
- 10) relative energy loss (i.e., the ratio of the energy loss and the initial energy),
- 11) initial displacement (can be non-zero only for track segments; see Section 4.3),
- 12) displacement (i.e., the distance between the initial point and the last collision point),
- 13) peak displacement (i.e., the maximum distance between a collision point and the initial point),
- 14) initial path length (can be non-zero only for track segments; see Section 4.3),
- 15) path length (the total length of the track, i.e., the total distance traveled from the initial point to the last collision, regardless of the direction of motion),
- 16) mean free path (the ratio of the path length and the number of collisions),
- 17) mean energy loss per collision (the ratio of the energy loss and the number of collisions),
- 18) stopping power (the ratio of the average energy loss and the specified path length).

Standard deviations of the average values are calculated as follows:

$$\sigma = \sqrt{\frac{1}{n(n-1)} \sum_{i=1}^n (x_i - \bar{x})^2}, \quad (4.1)$$

where  $n$  is the number of tracks in the selected subset,  $x_i$  is the value of the given quantity for track No.  $i$  ( $i = 1, 2, \dots, n$ ), and  $\bar{x}$  is the average value of the same quantity, calculated over the same subset of tracks. The stopping power is calculated only after specifying the minimum and maximum path length in the bottom row of the grid control “Track segment selection” (see Section 4.3).

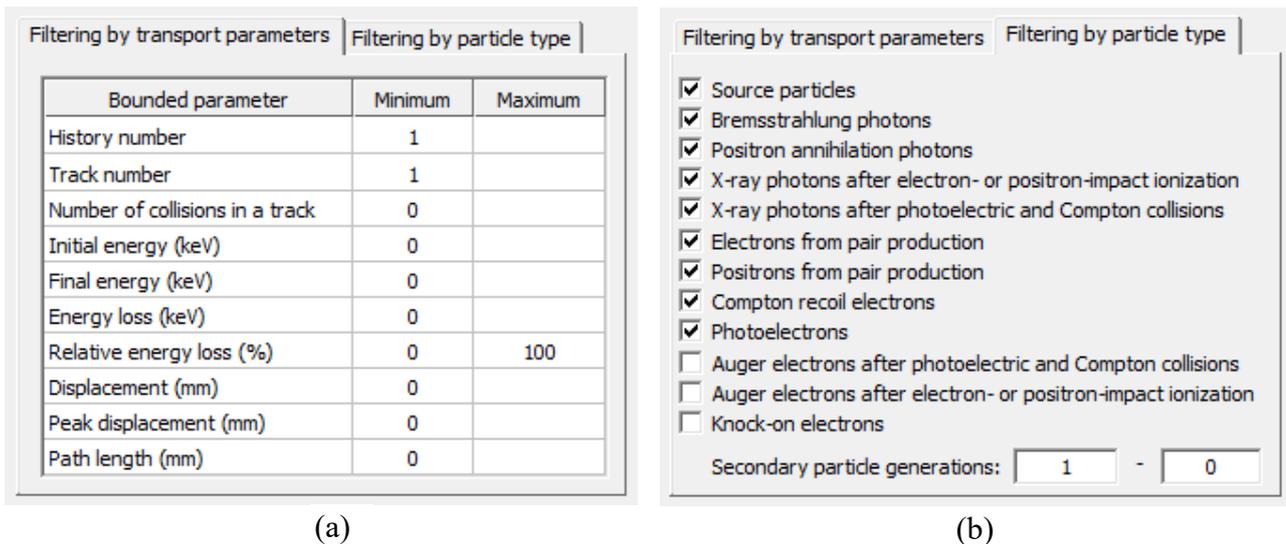
The average value and the standard deviation are not calculated only for the first two of the mentioned quantities (the sequence numbers of histories and tracks), because in the case of the latter quantities the mentioned two statistics are meaningless.

Two of the mentioned 18 statistics (mean free path and mean energy loss per collision) are defined only for the histories with collisions. Consequently, averaging of those two quantities is done over the histories with collisions, and  $n$  in the expression for the standard deviation (4.1) of each of those quantities has the meaning of the total number of tracks in those histories. It should be noted that the initial point of a track is never included in the number of collisions.

#### 4.2. Track filtering

The group box “Track filtering” (see Fig. 4.1) contains two tabs: “Filtering by transport parameters” and “Filtering by particle type”. Those tabs contain controls used to define the various track parameters used as criteria for selecting a subset of particle tracks (further on, this selection will be called “track filtering”). The contents of those two tabs are shown in Fig. 4.2.

The first of the mentioned tabs contains a grid control with the minimum and maximum values of the track sequence number, number of collisions in a track and several parameters characterizing the track geometry and particle energy (see Fig. 4.2a). The parameters whose limits can be set in this grid control are the following:



**Fig. 4.2.** The two tabs containing the track filtering controls: (a) filtering by track sequence number, particle energy and track geometry, (b) filtering by the type and origin of the primary particle of the track. In this example, all secondary particles are excluded, because the lower bound of the range of secondary particle generation numbers (1) is greater than the upper bound (0). The same result would be achieved by unchecking all checkboxes except “Source particles” (in such a case, the numbers entered in the two text boxes at the bottom would have no effect).

- 1) sequence number of the history containing the track,
- 2) track sequence number in the overall set of tracks (including all tracks in all histories),
- 3) total number of collisions in a track,
- 4) initial energy of the track (i.e., the energy of the particle immediately after its creation),
- 5) final energy of the track (i.e., the particle energy after the last collision of the track),
- 6) total energy loss (i.e., the difference of the initial and final energies),
- 7) relative energy loss (i.e., the ratio of the energy loss and the initial energy),
- 8) displacement (i.e., the distance between the initial point and the last collision point),
- 9) peak displacement (i.e., the maximum distance between a collision point and the initial point),
- 10) path length (the total length of the track, i.e., the total distance traveled from the initial point to the last collision, regardless of the direction of motion).

By default, each of the mentioned quantities is not constrained (apart from the “natural” constraints, which follow from the meaning of that quantity). In such a case, the lower bound of the corresponding filtering parameter is equal to its minimum possible value, and the upper bound is equal to the maximum possible value. If the parameter does not have a “natural” upper bound (for example, the path length, which can acquire all positive values), then the corresponding cell of the third column of the grid control in the tab “Filtering by transport parameters” is empty (the same effect is achieved by entering “-1” in that cell). It should be noted that filtering by path length is not applied to the particles that did not interact with any atom after their creation.

The second of the track filtering tabs contains the controls for selecting any combination of 12 types of tracks and the range of the secondary particle generation numbers. The track type identifies both the type of the particle and the type of the event where that particle was created. The generation number is the number of particle’s “ancestors”. The source particles belong to generation No. 0. Every secondary particle created by a source particle belongs to generation No. 1. Every secondary particle created by a first-generation particle belongs to generation No. 2, etc. In order to remove the upper bound of the generation number, the second text box at the bottom of Fig. 4.2b should be cleared of any text (the same result is achieved by entering “-1” in this text box). The smallest number that can be entered in the first text box is 1, because filtering by generation number is applied to secondary particles only (not to source particles). In order to include or exclude the source particles, the top checkbox should be checked or unchecked (see Fig. 4.2b).

By default, a minimal filtering is applied. Fig. 4.2a and Fig. 4.2b correspond to the default filtering (with the exception of the secondary particle generation number, which is unlimited by default). It is obvious that in the case of the default filtering the only constraint is exclusion of the low-energy electrons (Auger electrons and knock-on electrons). The default filtering can be restored by clicking the button “Default filtering without segment selection” (see Fig. 4.1).

After modifying the filtering parameters and clicking the button “Refresh statistics” (see Fig. 4.1), the track statistics will be updated. If the filtering parameters have been modified, but the statistics have not been refreshed yet, the asterisk is appended to the numbers in the four text boxes at the top of the “Track statistics” dialog window to indicate that the displayed set of statistics may be no longer current.

The two tabs of the “Track filtering” group of the “Track statistics” dialog window are duplicated in the lower right of the main MCNScript window (see Fig. 2.1). The changes of the filtering parameters done in the “Track statistics” dialog window will be carried over to the main MCNScript window only after clicking “OK” in the “Track statistics” dialog window.

The functionality of filtering by particle type is also provided by the dialog window “Track types and colors”, which is opened by clicking the button “Track types and colors...” in the MCNScript main window (see Fig. 2.1). An example of the dialog window “Track types and colors” is shown in Fig. 4.3. It has the same checkboxes and text boxes as the tab “Filtering by particle type” (see Fig. 4.2b), as well as the controls for selection of track colors for different track types. Below is the explanation of the meanings of all columns of the grid control that is in the dialog window “Track types and colors”:

- 1) The column “No.” contains the number of the particle type (those are the same type numbers that are assigned to the tracks in the track data file created by MCNelectron).
- 2) The column “Particle type” contains the name of the particle type and the type of the event where that particle was created.
- 3) The column “Include” contains checkboxes for selecting the track types that should be included during track filtering and shown in the MGED Graphics Window. If a checkbox is checked, then all tracks of the corresponding type will be included, otherwise they will be excluded.
- 4) The column “Color” shows the track colors. Those colors are used for drawing the tracks in the MGED Graphics window (examples are presented in Section 4.4).
- 5) The column “Color name” contains the dropdown list controls for selecting the color of each type of tracks. The user may select one of 15 standard colors, or a custom color. In order to set a

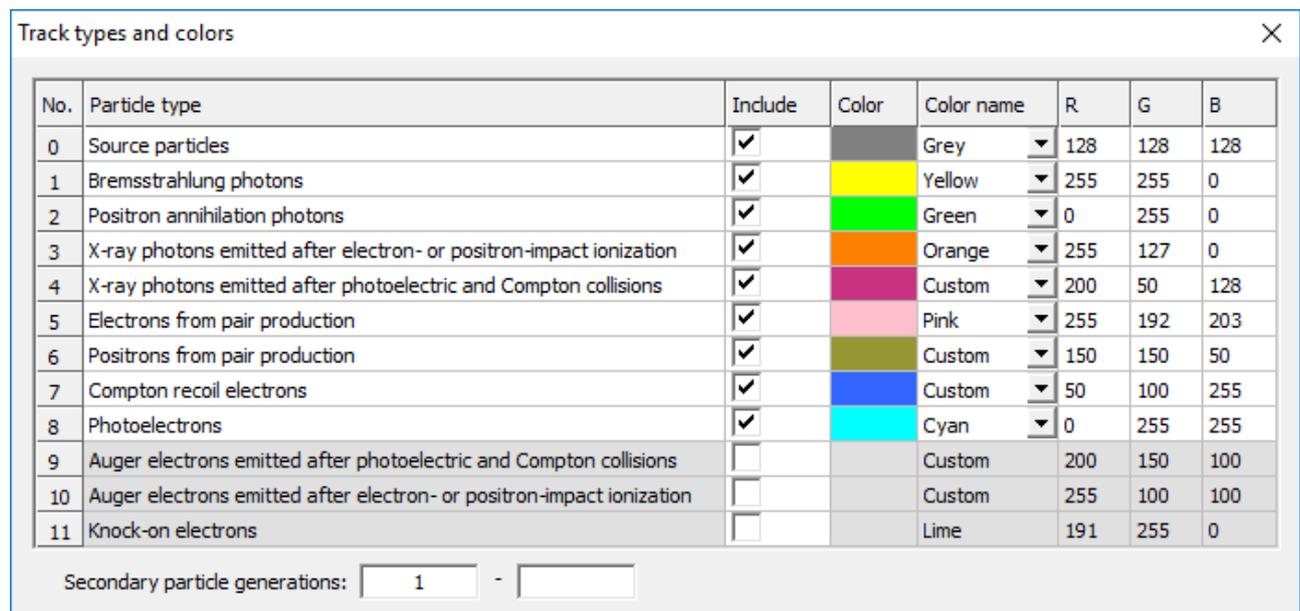


Fig. 4.3. An example of the dialog window “Track types and colors”.

custom color, the last entry of the dropdown list (“Custom”) must be selected. Then the standard Windows color selection dialog is opened.

- 6) The columns “R”, “G”, “B” contain the three color components (red, green and blue) of the corresponding color. They can be edited manually, copied and pasted. The allowed values of each color component are from 0 to 255. A change of any of these three numbers causes an update of the grid cells in columns “Color” and “Color name” of the same row.

The two text boxes at the bottom of the dialog window “Track types and colors” mirror the text boxes of the tab “Filtering by particle type” (see above).

A change of the state of any of the checkboxes in the column “Include”, or a change of a number in one of the two text boxes at the bottom of the dialog window “Track types and colors” is immediately mirrored by a corresponding change in the tab “Filtering by particle type” (see Fig. 4.2b), and vice-versa. Consequently, the dialog window “Track types and colors” can be used instead of the tab “Filtering by particle type”. While this dialog is visible, the interaction with the main MCNScript window (or with the other dialogs created by MCNScript) is not blocked. I.e., the dialog window “Track types and colors” is a so-called “modeless” dialog box (as opposed to a “modal” dialog, which blocks the interaction with the window that created the dialog). Thus, it is possible to keep the dialog window “Track types and colors” visible all the time while working with MCNScript.

The tracks shown in the MGED Graphics Window are the same ones that satisfy the filtering criteria. The tracks may be optionally replaced by their segments as described below.

### **4.3. Track segment selection**

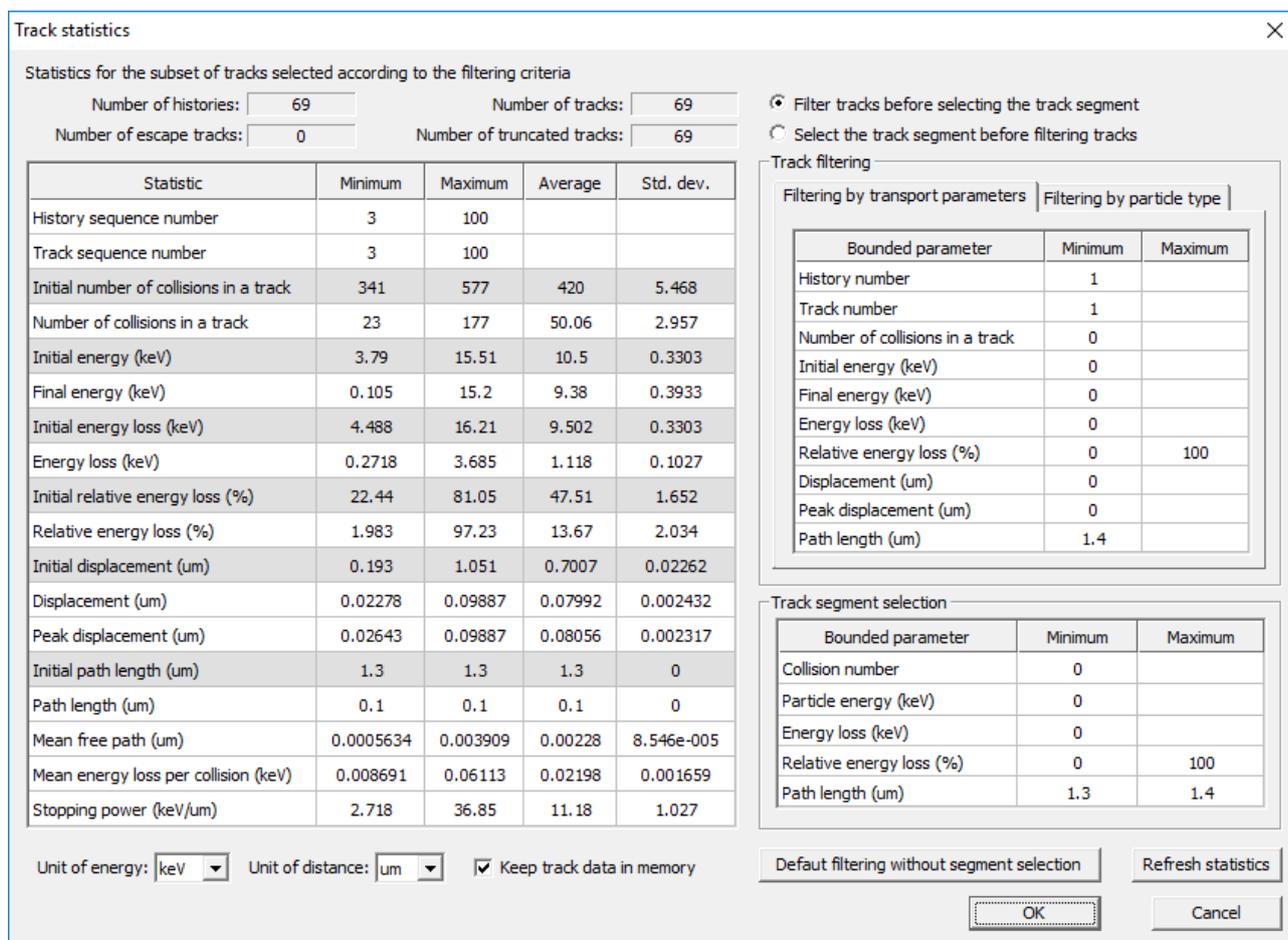
The grid control “Track segment selection” (see Fig. 4.4) is used to replace each track by its shorter segment, selected according to the constraints specified in that grid control. If the track statistics are re-calculated after such a replacement, then those segments will be treated in the same way as the complete tracks (for example, the initial point of each segment will not be included in the total number of collisions, and the energy corresponding to that point will be treated as the initial energy for calculating the energy loss). The quantities defining a track segment are the following:

- 1) collision sequence number in the complete track (the value 0 corresponds to the initial point of the complete track),
- 2) particle energy,
- 3) energy loss (i.e., the difference of the initial and current energies),
- 4) relative energy loss (i.e., the ratio of the energy loss and the initial energy),
- 5) path length (the length of the track, i.e., the distance traveled from the initial point to the current collision, regardless of direction of motion).

In order to re-calculate the track statistics for the selected track segments, the button “Refresh statistics” must be clicked.

The two radio buttons that are at the top right of the “Track statistics” dialog window (see Fig. 4.4) control the order in which the track filtering and the segment selection are applied. In general, this order affects the track statistics shown in the grid control on the left, because the track segments are treated exactly in the same way as complete tracks (both during calculation of the track statistics and during track filtering, if the latter is applied after the segment selection).

If track segments have been defined in the “Track segment selection” grid control, then the values in the six rows with grey background in the track statistics grid control (see Fig. 4.4) correspond to the initial point of the track segment. If this point does not coincide with the initial point of the complete track, then all those values are in general non-zero (otherwise, only the initial energy is non-zero). For example, the initial number of collisions is equal to the sequence number of the starting vertex of the track segment in the complete track (the number ‘0’ corresponds to the initial point of the complete track). Four of those statistics that have the meaning of the initial change of energy or distance (initial energy loss, initial relative energy loss, initial displacement and



**Fig. 4.4.** An example of the dialog window with particle track statistics, using both track filtering and track segment selection. In this case, the incident particles are electrons with energy 20 keV, which are incident normally on a thick layer of copper with density 8.96 g/cm<sup>3</sup>. The stopping power given in the bottom row of the grid control on the left approximately corresponds to electron energy 10 keV (see the values of the initial and final energy in this grid control). The secondary particles are excluded.

initial path length) indicate the change of the corresponding quantity in the initial part of the complete track, i.e., the part that precedes the selected track segment.

The example of Fig. 4.4 shows how to apply both track filtering and track segment selection simultaneously. In this example, only the source particle track segments corresponding to the path length values from 1.3 μm to 1.4 μm are analyzed (see the “Track segment selection” grid control in Fig. 4.4). In order to exclude the tracks shorter than 1.4 μm, the minimum path length is set to 1.4 μm in the “Filtering by transport parameters” grid control. The average values of the resulting initial and final energies (see the grid control on the left in Fig. 4.4) are sufficiently close to each other to make the estimate of the stopping power approximately equal to its value corresponding to the arithmetic mean of those two average energies, i.e., 10 keV. Consequently, the last row of the grid control on the left shows the approximate stopping power of electrons with energy 10 keV (in this example, the target material is copper with density 8.96 g/cm<sup>3</sup>). As it is evident in Fig. 4.4, this stopping power is (11.18 ± 1.03) keV/μm. This value has been obtained by dividing the average energy loss (1.118 keV) by the specified path length (1.4 – 1.3 = 0.1 μm).

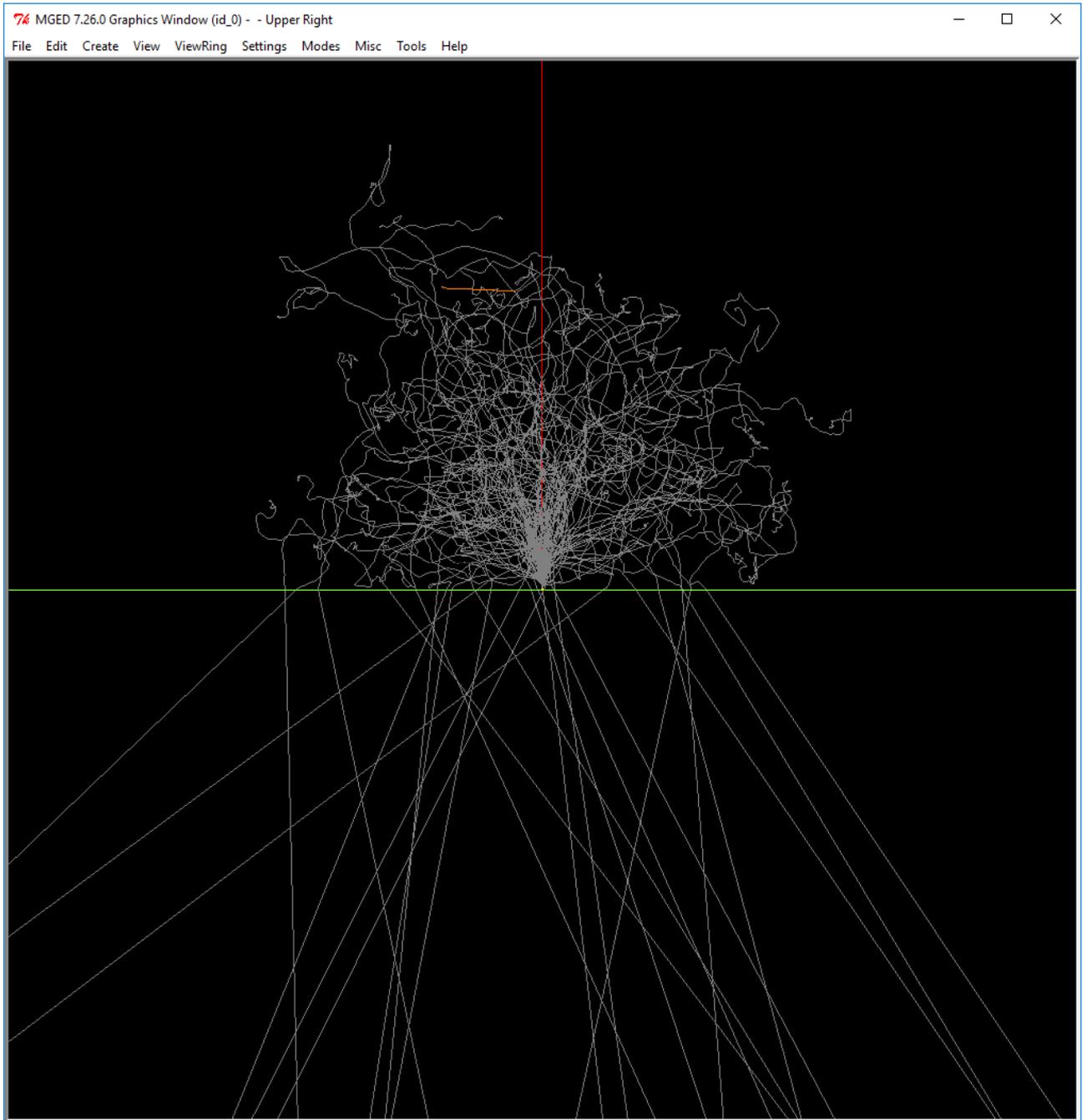
If the lower bound of the path length specified in the bottom row of the “Track segment selection” grid control is zero, then the initial point of the track segment is either the initial point of the complete track or one of the vertices of the complete track (depending on the numbers entered in the other rows of this grid control). Otherwise, it is defined as the point in the particle trajectory where the total distance traveled from the initial point of the complete track is exactly equal to the

specified lower bound of the path length. Similarly, if the upper bound of the path length has not been specified in the bottom row of the “Track segment selection” grid control, then the final point of the track segment coincides with one of the collision points of the complete track. Otherwise, it is defined as the point in the particle trajectory where the total distance traveled from the initial point of the complete track is exactly equal to the specified upper bound of the path length. Consequently, if in absence of the constraints on the length of track segments there are no tracks (or track segments) with the total length between those two bounds, then after specifying the lower and upper bounds of the path length its value for all track segments that are taken into account will be exactly equal to the difference of the mentioned bounds (as in Fig. 4.4). Otherwise, some of the track segments will be shorter, which will be obvious from the statistics given in the “Path length” row of the track statistics grid control (in the case of the example of Fig. 4.4, this can be seen after removing the constraint on the minimum path length in the “Filtering by transport parameters” grid control, i.e., after setting it to 0, and then clicking the button “Refresh statistics”). In such cases, the stopping power is not calculated.

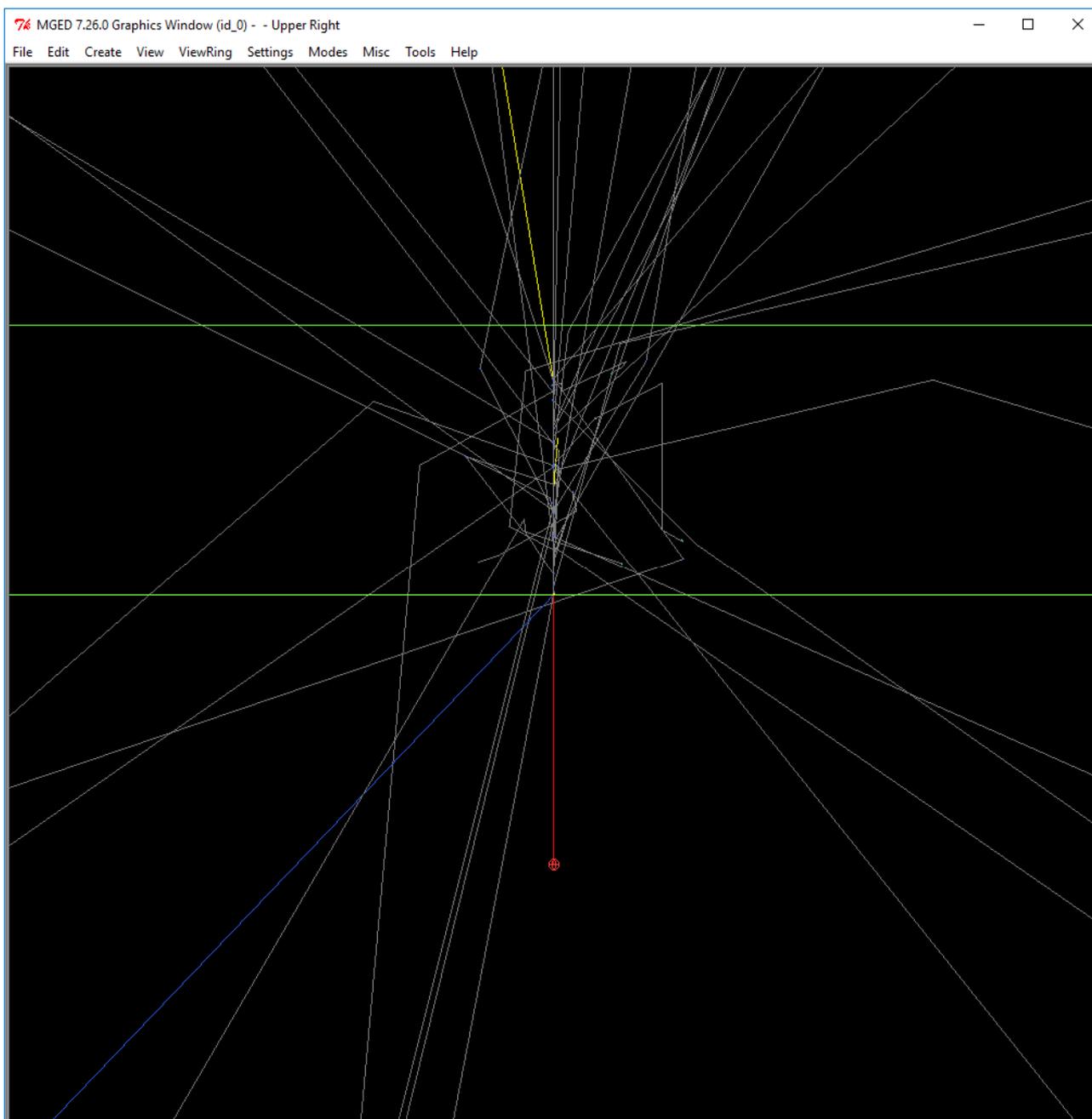
For escape tracks, the path length does not include the last straight part of the track (i.e., the part after the last interaction event). If the last interaction corresponds to the value of the path length that is between the bounds specified in the “Track segment selection” grid control, then the last straight part of the escape track will be shown in the MGED Graphics Window.

#### ***4.4. Examples of particle track display in the MGED Graphics Window***

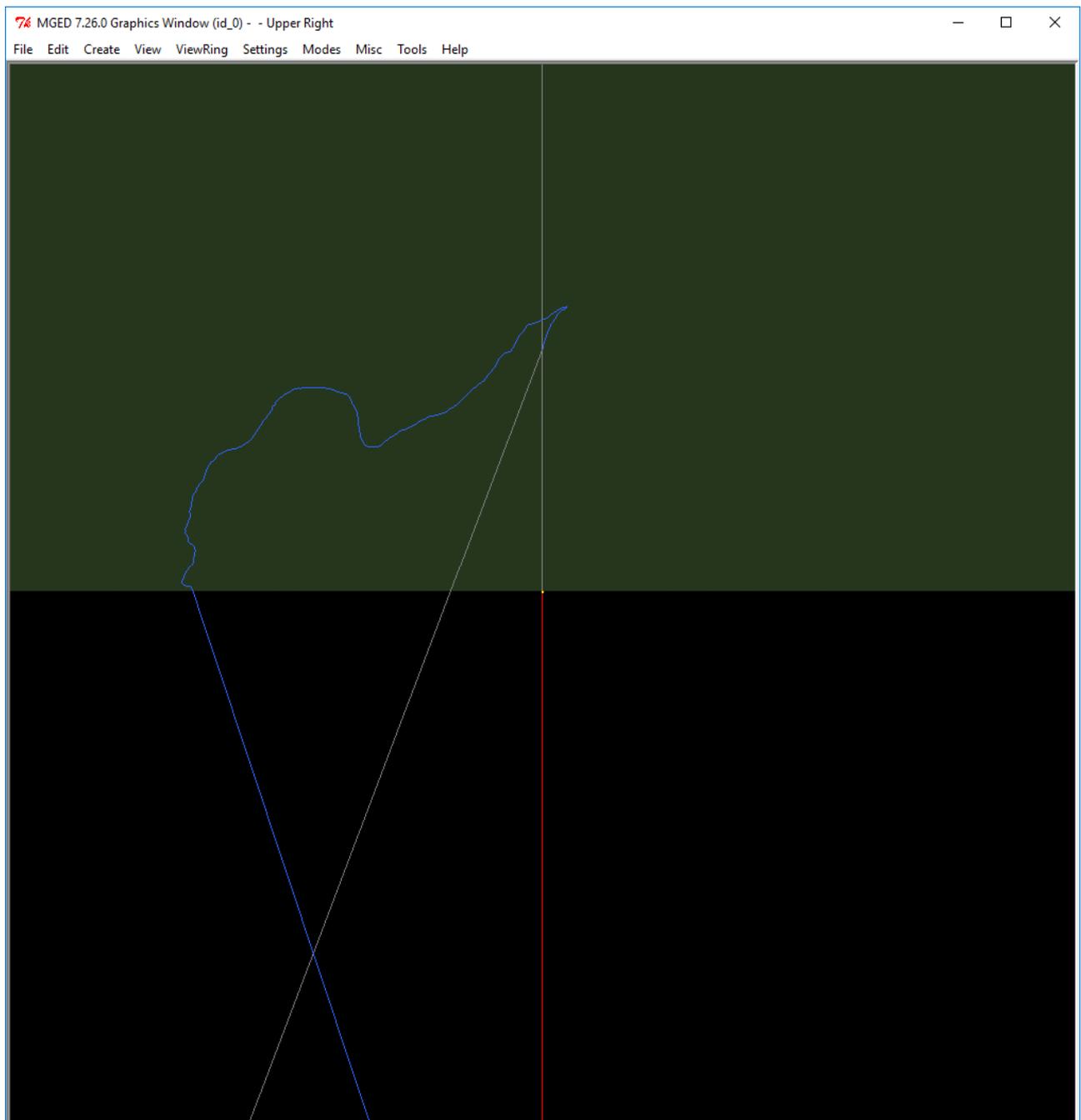
Fig. 4.5, Fig. 4.6 and Fig. 4.7 show examples of electron and photon tracks as seen in the MGED Graphics Window. In the case of escape tracks, the final straight segment of the track is extended beyond the clipping cube (defined in Section 6.2). The captions of Fig. 4.5, Fig. 4.6 and Fig. 4.7 provide additional information about these images.



**Fig. 4.5.** An example of electron tracks shown in the MGED Graphics Window. The electrons with energy 20 keV are incident normally on a thick layer of copper with density  $8.96 \text{ g/cm}^3$ . The direction of incidence is upwards (it is shown by the red line), and the point of incidence is in the center of the window. The target material occupies the top half of the window. The view size is  $3 \text{ }\mu\text{m}$ . The number of source tracks (grey lines) shown is 100. The 25 straight lines that are seen in the bottom half of this image represent the backscattered electrons. The orange line near the top represents the track of an X-ray photon emitted after an impact-ionization event.



**Fig. 4.6.** An example of photon tracks shown in the MGED Graphics Window. The photons with energy 662 keV are incident normally on a 1-centimeter-thick layer of iron with density  $7.874 \text{ g/cm}^3$ . The view size is 4 cm. The source is shown as a small red sphere at a distance of 1 cm from the layer, the direction of incidence is upwards (it is shown by the red line), and the point of incidence is in the center of the window. The two horizontal green lines represent the two surfaces of the target layer. The number of source tracks (grey lines) is 100. However, 61 of those tracks have no collisions. Those 61 tracks are represented by a straight grey line starting at the point of incidence and pointing upwards (it obscures the red line showing the direction of the incident beam). The yellow lines represent bremsstrahlung photons, and the blue lines are Compton recoil electrons. One of the Compton recoil electrons escaped from the layer, as it is evident from the straight blue line in the bottom half of the image (the magnified track of that electron in the target material before its escape is shown in Fig. 4.7).



**Fig. 4.7.** A magnified and rotated portion of the view of Fig. 4.6. The view size is 0.031 cm. The image shown in Fig. 4.6 has been rotated around the vertical axis, so that the trajectory of the scattered photon (grey line) lies in the projection plane (the MGED commands for manipulating the view, including its magnification and rotation, are described in Section 7). The track of the Compton recoil electron before its escape from the layer (blue line) is clearly seen.

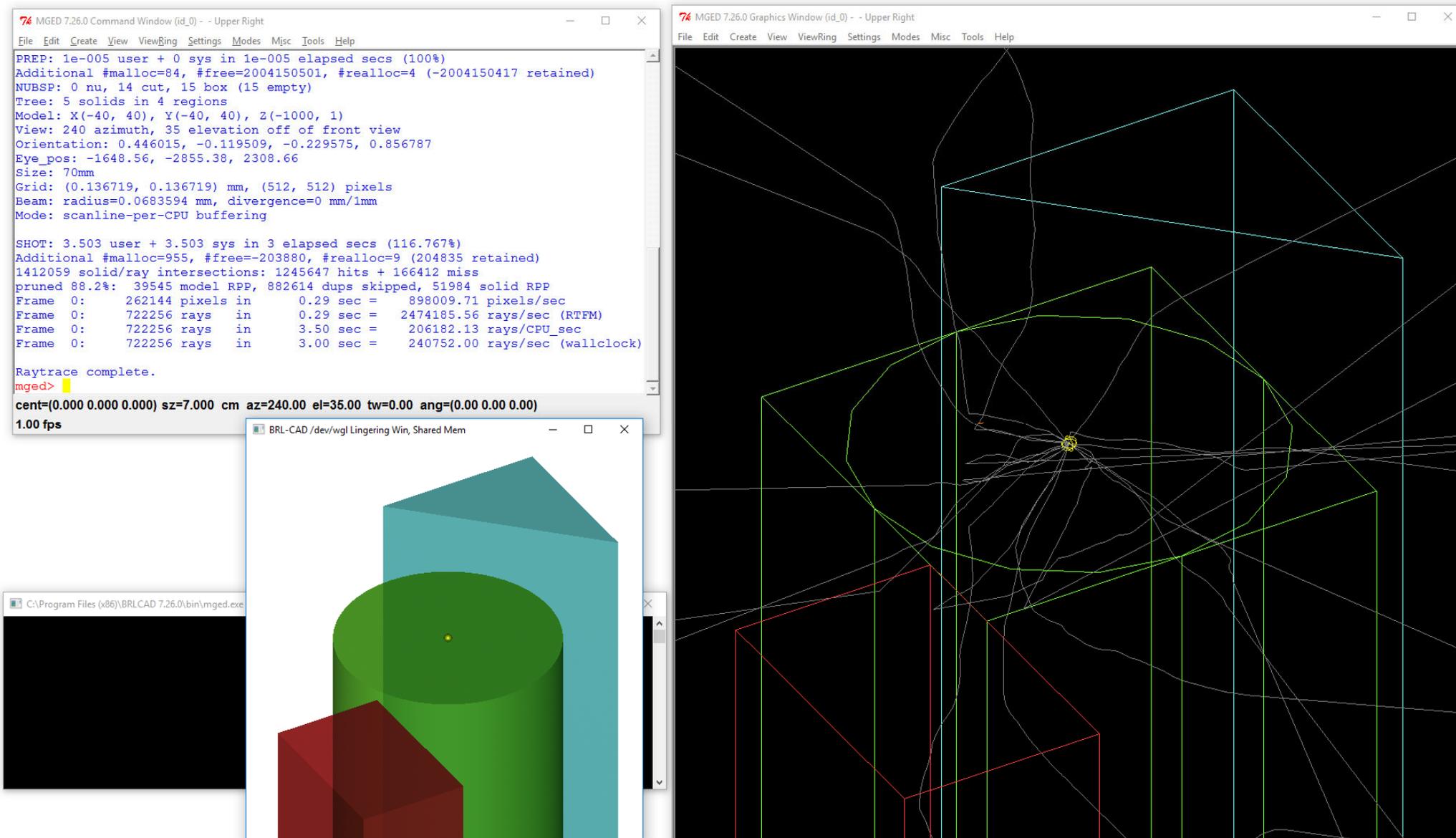
## 5. RT script file and MGED startup

In order to create an RT script using the data of the specified MCNelectron input file and the graphical parameters described in Section 6, the button “Make RT script” must be clicked (see Fig. 2.1). After clicking that button, a script file with the name specified in the text box of the seventh row of controls of the MCNScript main window will be created. If the checkbox “Copy to clipboard” is checked (see Fig. 2.1), then the script will be simultaneously copied to the clipboard. The folder where the script file must be created can be selected by clicking the button “Select folder...”, which is also in the seventh row of controls (see Fig. 2.1). The name of the script file is formed by appending the suffix specified in the text box “Script file name suffix” to the name of the MCNelectron input file and adding the extension “.RT”. The mentioned file name suffix may include the two-character string “%d”, which has a special meaning: if it is present, then it will be replaced by the sequence number of the current parameter set (this number is shown in parentheses before the name of the parameter set in the dropdown list, which is at the top of the MCNScript main window). The script is created only if the checkbox “Check syntax” is checked (see Fig. 2.1).

If particle tracks have to be displayed in the MGED Graphics Window, then the MGED commands for drawing each type of tracks will be placed into a separate script file. The initial part of the name of each of those files is the same as the name of the main script file (i.e., it is the name of the MCNelectron input file with the appended suffix), and the last part of the track script file name has the format “\_tracks<n>\_<string>.txt”, where “<n>” is the track type number, and “<string>” is a short sequence of characters describing the track type. For example, the name of the file with photoelectron tracks ends with “\_tracks8\_photoel.rt”. The type numbers are shown in the first column of the grid control in the dialog window “Track types and colors” (see Fig. 4.3). Each of those track scripts will be loaded automatically when executing the main script, because the main script contains the MGED commands for loading each of the created track script files.

In order to execute the created script, MGED must be started. MGED can be started either via the Windows user interface, or by clicking the button “Start MGED” of the MCNScript main window (see Fig. 2.1). The latter method is more convenient, because it sets the working directory of MGED to the folder where the script files are placed (consequently, navigating to the script file after selecting the MGED menu command “File / Load Script...” is less time-consuming). In order to start MGED by clicking “Start MGED”, the full name (including the installation folder) of the MGED executable file (mged.exe) must be specified in the text box of the eighth row of controls of the MCNScript main window (see Fig. 2.1). The MGED executable file can be located using the button “...”, which is in the same row of controls. That file is usually located in the subfolder “bin” of the BRL-CAD installation folder. If the BRL-CAD installer was run as administrator, then the default installation folder of the 32-bit BRL-CAD software system (assuming BRL-CAD version number 7.26.0) would be “C:\Program Files (x86)\BRLCAD 7.26.0” on 64-bit Windows, or “C:\Program Files\BRLCAD 7.26.0” on 32-bit Windows. If the BRL-CAD installer was run without administrator privileges, then the default installation folder of BRL-CAD 7.26.0 would be “C:\Users\<User Name> \Documents\BRLCAD 7.26.0”.

Normally, an MGED startup would create three windows – a blank window with the name of the MGED executable file in the title bar, the “Command Window” and the “Graphics Window”. Both of the latter two windows have a menu bar at the top (see Fig. 5.1). Those two menus are identical. Additional windows may be opened by selecting various menu commands or by entering commands into the Command Window, which is also used by MGED for displaying various messages. This type of MGED graphical user interface (GUI) will be called “GUI mode”. On some Windows configurations, the GUI mode of MGED may be not available. In such a case, MGED would start in so-called “classic mode”, where the menus are not available. In classic mode, the MGED commands must be entered in the console window with the name of the MGED executable file in the title bar, and the MGED messages are displayed in the same window (in GUI mode, this window is blank). If the MGED executable file “mged.exe” was started without any command-line arguments, then the failure to initialize the GUI mode would be signaled by the following message:



**Fig. 5.1.** An example of the MGED user interface in “GUI mode” (i.e., with menu bars present). This view is obtained after loading the script corresponding to the example of Fig. 2.1. The square window at the bottom left is the so-called “lingering window”, which contains a raytraced image with a cut-away view of the simulation setup (the small yellow sphere in the center of the raytraced image represents an isotropic point source of electrons with energy 100 keV). The MGED Graphics Window on the right contains the wireframe view of the simulation setup (colored lines) and 20 electron tracks (they are shown by grey lines).

```
couldn't connect to display "localhost:0"  
MGED unable to initialize gui, reverting to classic mode.  
attach (nu|wgl)[nu]?
```

After this prompt, the three characters “wgl” must be typed and the “Enter” key must be pressed. Then MGED will be started in classic mode. In order to start MGED in classic mode immediately (without the mentioned prompt), the MGED startup command line must be modified by enclosing the name of the MGED executable file with double quotes and by appending the following string with command-line arguments: “-c -a wgl”. For example, the string entered in the text box “MGED startup command line” of the MCNScript main window could be the following:

```
"C:\Program Files\BRLCAD 7.26.0\bin\mged.exe" -c -a wgl
```

After starting MGED, the script must be either loaded from file (MGED menu command “File / Load Script...”) or pasted into the MGED Command Window (in GUI mode, the script can be pasted using the keyboard shortcut “Ctrl+V”, followed by the key “Enter”). Alternatively, the script file can be executed using the MGED command “source”. This keyword must be followed by the full name of the script file, with all backslash characters ‘\’ replaced by the forward slash ‘/’ (in addition, if the file name includes spaces, then it must be enclosed by double quotes). In classic mode, an RT script file can be executed only by the latter method. For example, if the RT script is in the file “C:\MCNelectron\script.rt”, then it can be executed by typing “source C:/MCNelectron/script.rt” after the prompt “mged>” (see Fig. 5.1) and pressing the key “Enter”. If the script file is located in the folder designated as the working directory of MGED, then it is sufficient to specify only the file name without the folder name (for example, “source script.rt”). In classic mode, the script may be also executed by pasting it into the console window after the “mged>” prompt, but then execution of a large script may take a long time, because MGED would scroll through every line of the script in the console window.

When the RT script created by MCNScript is executed, the corresponding MGED database file is created. It has the same name as the RT script file, but with a different extension (“.G”), and is located in the same folder as the RT script file. The database file contains information about the system geometry.

In order to close all windows of the MGED graphical user interface when MGED is in GUI mode, the menu command “File / Exit” must be selected (it is equivalent to entering the “exit” command in the Command Window). This command does not close the “lingering” windows with raytraced images of 3D objects. In order to close *all* windows created by the current MGED instance (both the GUI windows and the lingering windows) simultaneously, the console window with the name of the MGED executable in the title bar must be closed.

After an instance of MGED is started by clicking the button “Start MGED”, the text on the latter button becomes “Activate the MGED window” (see Fig. 3.3). Afterwards, if any of the windows corresponding to that instance of MGED are minimized or covered by other windows, this button can be used to bring them to the foreground. Each instance of MGED is “attached” to the corresponding parameter set. If there are several instances of MGED, each of them can be activated by selecting the corresponding parameter set and then clicking “Activate the MGED window”. After closing that instance of MGED, the text on the mentioned button becomes “Start MGED” again. This functionality is also available when the executable file name specified in the text box at the end of the eighth row of controls of the MCNScript main window is not “mged.exe”. Regardless of the file name, a new instance of the specified program is started by clicking “Start MGED”, and afterwards that instance can be activated by clicking “Activate the MGED window”. One possible alternative to mged.exe is archer.exe, which is usually located in the same folder as mged.exe (Archer has the same capabilities as MGED, but a more convenient GUI). It is possible to “detach” a MGED window from the current parameter set. This is done by right-clicking the button “Activate the MGED window” and selecting “Reset this button” from the context menu. Then the text on that button becomes “Start MGED” again. Afterwards, it will be possible to use this button for starting another instance of MGED without closing the “detached” MGED window.

## 6. Graphical parameters

The graphical parameters are grouped according to their function. The different groups of graphical parameters are described below.

### 6.1. Viewing cube and eye position parameters

The term “viewing cube” is used in the MGED documentation when referring to the square MGED display. It represents the part of the 3D “world” seen by the user. The size of the viewing cube and the position of its center are controlled by the text box “View size” and the three-cell grid control “View center” (see Fig. 6.1). All sizes and coordinates are specified in centimeters. Since the edges of the viewing cube are aligned with the viewing direction (or “eye position”), three additional angles (azimuth, elevation and twist) must be provided for complete definition of the viewing cube in terms of the Cartesian coordinate system used for defining dimensions and positions of all 3D objects composing the simulation setup. The meaning of those three angles is illustrated in Fig. 6.2. They must be entered in three text boxes titled “Azimuth”, “Elevation” and “Twist” (see Fig. 6.1) in degrees (either manually, or in 5-degree increments using the nearby spin controls). The mentioned three angles determine the directions of the coordinate axes as seen in the MGED Graphics Window. Those directions are shown in the picture that is to the left of the group

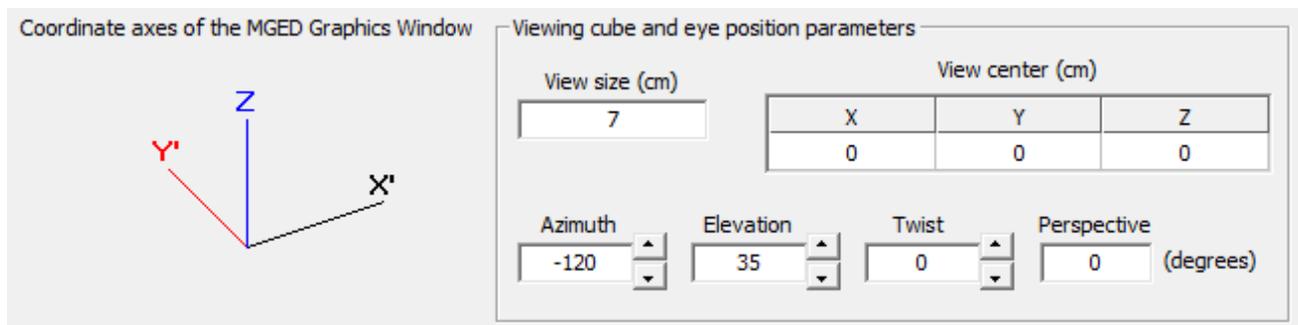


Fig. 6.1. Viewing cube and eye position parameters.

“Viewing cube and eye position parameters” (see Fig. 6.1). If an axis title is shown with the prime symbol, this means that the positive half-axis is below the projection plane.

The text box “Perspective” (see Fig. 6.1) is used to set the perspective angle. Perspective makes the 3D image more realistic by making the receding parallel lines appear to be non-parallel. The possible values of the perspective angle are from 0 to 90°. The default value is 0 (no perspective). This value is recommended for easiest interpretation of complex images. **Note:** The “perspective” parameter is the only graphical parameter that is not applied when the RT script created by MCNScript is loaded using the MGED menu command “File / Load Script...”. It is applied only when the script is loaded using the MGED command “source”, or pasted into the MGED Command Window, or when the perspective angle is set manually using the MGED command “set perspective”.

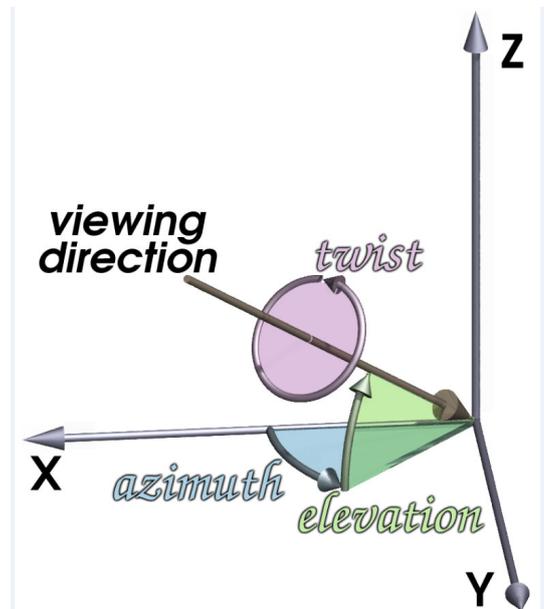


Fig. 6.2. The meaning of azimuth, elevation and twist angles (this diagram was found at <http://brlcad.org/gallery/diagrams/diagram>).

## 6.2. Graphical parameters controlling the size limits, display of the source, and background color



Fig. 6.3. Size limits and display of the source.

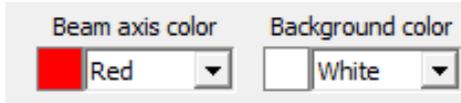


Fig. 6.4. Beam axis color and background color.

If the MCNelectron simulation setup includes a point source (MCNelectron input file directive “SOURCE\_POSITION 1”) or a radially symmetrical distribution of source points on a plane (directive “SOURCE\_POSITION 2”), then the source center can be shown in MGED as a spherical region. The ratio of the diameter of that sphere to the initial view size is entered in the text box “Source size / view size” (see Fig. 6.3). If zero ‘0’ has been entered in this text box, then the source center will not be shown. The example of Fig. 5.1 corresponds to the value of this parameter equal to 0.02 (the source is shown in this example as a small yellow sphere at the center). In the case of a non-volume non-isotropic source (SOURCE\_POSITION 1 or 2, SOURCE\_DIRECTION 1 or 2), the symmetry axis of the angular distribution of source radiation is shown in the MGED Graphics Window as a ray with the initial point at the source center. The color of that line is set using the dropdown list box “Beam axis color” (see Fig. 6.4). If the last entry (“None”) is selected in that list, then the beam axis will not be shown.

The background color is set using the dropdown list box “Background color” (see Fig. 6.4). However, the selected background color will be automatically applied in the MGED Graphics Window only after a successful raytracing. If raytracing in the MGED Graphics Window is not required or if it fails, the background color can be applied in the MGED Graphics Window only manually, using the MGED Raytrace Control Panel, which is opened by selecting the MGED menu command “File / Raytrace” (see also Section 7).

## 6.3. Using a cut-away plane

A cut-away plane can be used to uncover obscured regions of the system geometry by removing the part of the system that intersects with a specified half-space. In order to enable usage of the cut-away plane, the checkbox “Use a cut-away plane” must be checked (see Fig. 6.5). After that, the mentioned half-space is defined using the group of controls “Cut-away plane parameters” (see Fig. 6.5). The cut-away plane is defined by specifying the coordinates of a single point belonging to that plane (in centimeters) and the components of the normal vector.

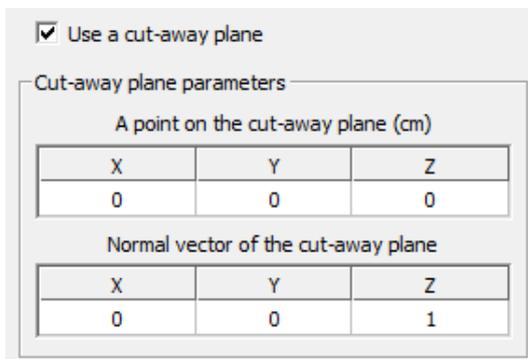
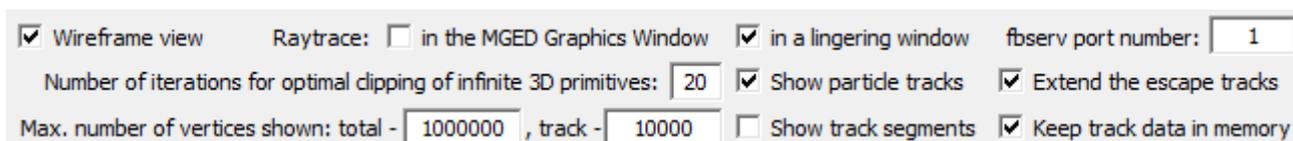


Fig. 6.5. The cut-away plane parameters.

The normal vector may be of any non-zero length. The part of the simulation setup that is removed is inside the half-space where the normal vector is directed to. An example of a raytraced image obtained using a cut-away plane is shown in Fig. 5.1 and Fig. 6.7. In this example, the simulation setup corresponds to the sample MCNelectron input file given in Section 4.2 of the MCNelectron user manual (the cross-sectional view of this simulation setup is shown in Fig. 4.1 of the MCNelectron user manual); the cut-away plane is the XY plane of the coordinate system, and the source is at the point (0, 0, 0).

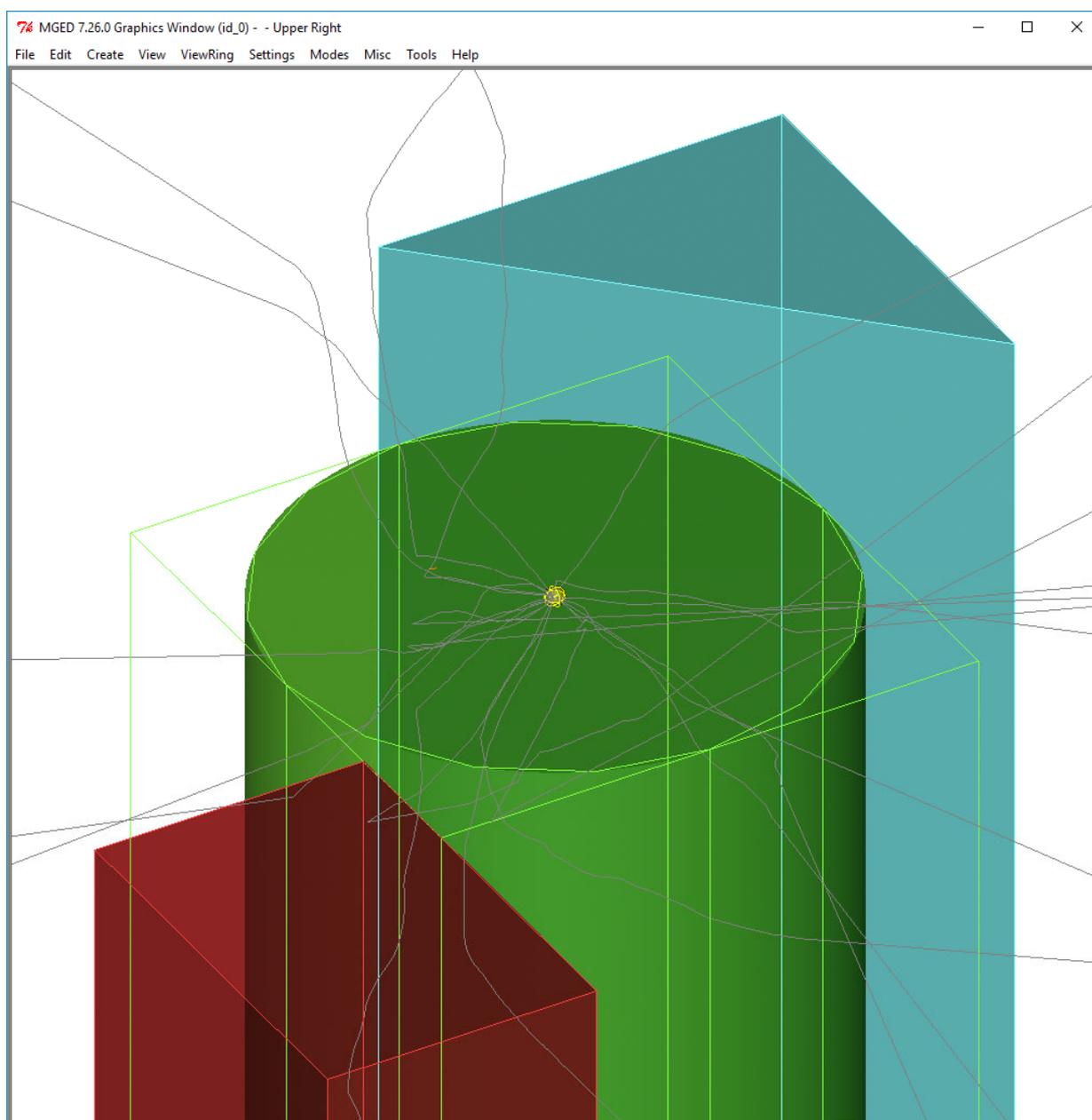
## 6.4. Specifying the amount of data to show in MGED

The group of text boxes and checkboxes above the grid control “Materials” (see Fig. 6.6), excluding the checkbox “Keep track data in memory”, which is duplicated in the “Track statistics” dialog (see Section 4), are used for specifying the amount of data that should be shown in MGED:



**Fig. 6.6.** The controls for specifying the amount of data to show in MGED.

- If the checkbox “Wireframe view” is checked, then the wireframe view of the surfaces used to construct the 3D objects of the simulation setup will be shown in the MGED Graphics Window. If raytracing has been done in the MGED Graphics Window, then the raytraced image will be overlaid by the wireframe view (see Fig. 6.7).
- If the checkbox “Raytrace in the MGED Graphics Window” is checked, then the raytraced image will be shown in the MGED Graphics Window (see Fig. 6.7). However, if the MGED framebuffer is not active, then raytracing in the MGED Graphics Window will fail. In order to activate the MGED framebuffer, the MGED menu command “Modes / Framebuffer active” must be selected. In addition, if several instances of MGED are active, then the Graphics Window of only one of them will be used for raytracing. This is because the RT scripts created by



**Fig. 6.7.** An example of a raytraced image of 3D objects in the MGED Graphics Window (using a cut-away plane). The raytraced image is overlaid by the wireframe view (colored lines) and particle tracks (grey lines).

MCNScript use a fixed framebuffer server port number “5559” for raytracing in the MGED Graphics Window (in contrast to raytracing in lingering windows, as explained below). Consequently, in such a case it is better to use the MGED Raytrace Control Panel for raytracing in the MGED Graphics Window (see Section 7).

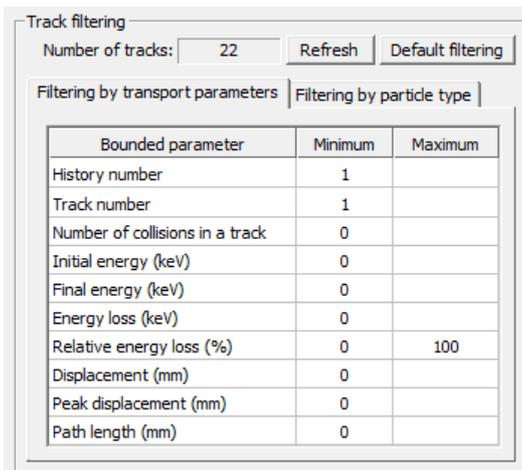
- If the checkbox “Raytrace in a lingering window” is checked, then a so-called lingering window with the raytraced image will be created (this does not require the MGED framebuffer to be active). Unlike the MGED Graphics Window, the lingering window shows *only* the raytraced image of the 3D objects of the simulation setup, and does not show the wireframe view or any of the supplemental lines (i.e., the particle tracks and the beam axis). An example of a lingering window with a raytraced image is shown in Fig. 5.1. Each lingering window has an associated framebuffer server port number, which must be entered in the text box “fbserve port number” (see Fig. 6.6). In order to open a new lingering window without closing the previous one, both the port number and the name of the MGED database (saved in a file with extension “.G”) must be changed. The name of the MGED database file is the same as the RT script file name (the latter can be changed by changing the MCNelectron input file name or the suffix used to form the name of the RT script file).
- The text box “Number of iterations for optimal clipping of infinite 3D primitives” is used for optional adjustment of the positions of the planar surfaces that are added by MCNScript in order to truncate (“clip”) infinite objects, i.e., the positions of the flat faces of truncated cylinders and cones, and the clipping planes of open polyhedra. Their initial positions are such that they are either beyond the clipping cube (defined in Section 6.2), or coincide with the faces of the clipping cube. However, those positions may be iteratively adjusted until each clipping plane touches the surface of the cell containing a given 3D primitive. This adjustment increases the amount of detail in the wireframe view (the raytraced image is not affected). The adjustment is done in steps, whose length is decreased by half in each iteration. The maximum number of iterations (steps) can be specified in this text box. If zero is entered in this text box, then no adjustments will be made. The maximum allowed number of iterations is 50.

Open polyhedra may arise as a result of the method used by MCNScript for drawing planar surfaces that do not belong to macrobodies (the term “macrobody” is defined in the next paragraph). In order to avoid drawing of half-spaces (which is in MGED slower than drawing of finite 3D primitives), the set of flat bounding surfaces of each 3D object (excluding the surfaces that belong to macrobodies) is replaced by a polyhedron bounded by those planes. The 3D object is then defined as the intersection of this polyhedron and all other 3D primitives used in the definition of the object. If this polyhedron is open, then it is “closed” by adding the clipping planes to it (the clipping planes are normal to the coordinate axes). The edges of all polyhedra are visible in the wireframe view (see Fig. 5.1 and Fig. 6.7). The cylinder shown Fig. 5.1 and Fig. 6.7 is surrounded by four clipping planes, which are required to close the set of planar bounding surfaces of this 3D object (initially, this set contained only the top and bottom planes). This 3D object is not replaced automatically by a right circular cylinder (which is one of the standard 3D primitive shapes in MGED), because it was originally defined as a region of space bounded by three independently defined surfaces – two planes and one cylinder, and because MCNScript does not “know” if those planes are normal to the symmetry axis of the cylinder.

In addition to the four types of “regular” surfaces (planes, spheres, circular cylinders and circular one-sheet cones), MCNelectron and MCNScript support the so-called “macrobody”, i.e., the sets of two or more regular surfaces defining various 3D shapes, which can be used in definitions of the cells of the geometry along with the regular surfaces. There are 8 “standard” macrobodies (right circular cylinder, truncated right-angle cone and 6 types of polyhedra) and a “general” macrobody (a shorthand notation for an arbitrary set of user-defined regular surfaces and their “senses”). If the definition of a 3D object includes standard macrobodies, then the positions of the flat faces of those macrobodies are never adjusted, and those macrobodies are shown in the wireframe view separately from the other 3D primitives. Since all the standard macrobodies recognized by MCNelectron have counterparts in MGED, no additional clipping is

needed when displaying them in the wireframe view. For example, if the cylinder shown in Fig. 5.1 and Fig. 6.7 was defined as a macrobody (surface type “RCC”, meaning “right circular cylinder”), then the wireframe view would not include the four clipping planes parallel to the cylinder’s symmetry axis that are visible in Fig. 5.1 and Fig. 6.7.

- If the checkbox “Show particle tracks” is checked, then the particle tracks satisfying the filtering criteria will be shown in the MGED Graphics Window. Each particle track is shown as a piecewise-linear curve starting at the point of entry of a source particle into a cell of the geometry, or at the point of creation of a secondary particle, with each linear segment representing free flight of the particle between two collisions. The color of those lines is specified in the dialog window “Track types and colors” (see Fig. 4.3 in Section 4.2). In general, not all tracks are shown, but only the ones that satisfy the filtering criteria defined in the two tabs, which are at the bottom right of the MCNScript main window (see Fig. 6.8). Those tabs are duplicated in the dialog window “Track statistics” (see Section 4.2). The number of the tracks satisfying the filtering criteria is shown in the text box “Number of tracks”, which is above the mentioned tabs, after clicking the button “Refresh” (see Fig. 6.8). In order to apply the default (minimal) filtering, the button “Default filtering” must be clicked (see Fig. 6.8). This would remove any constraints on transport parameters (as shown in Fig. 6.8) and most of the constraints on the included track types (as shown in Fig. 4.3 in Section 4.2). For more information about track filtering, see Section 4.2.



**Fig. 6.8.** The track filtering controls of the MCNScript main window.

- If the checkbox “Extend the escape tracks” is checked, then the final straight segment of each escape track will be extended beyond the clipping cube (defined in Section 6.2). For example, the images shown in Fig. 4.5, Fig. 4.6 and Fig. 4.7 were obtained using this option. If the checkbox “Extend the escape tracks” is not checked, then the final point of each escape track will coincide with the escape point (i.e., the point where the particle crossed the boundary of a cell of the geometry for the last time on its way out of the system).
- The two text boxes at the bottom of this group of controls (see Fig. 6.6) are used to limit the number of track vertices shown in the MGED Graphics Window (here, the term “vertices” includes not only the internal nodes, but also both the initial and final points of the piecewise-linear curve representing each track). The number entered in the first text box (“total”) is the maximum allowed total number of vertices (summed over all tracks shown). If that number is exceeded, then some of the tracks will not be shown (those are the last tracks of the current set of tracks). If the text box “total” is empty, then all tracks will be shown (the same effect is achieved by entering “-1” in this text box). The number entered in the second text box (“track”) is the maximum number of vertices allowed in a single track. If the number of vertices in a track exceeds the latter number, then some of the vertices will be skipped when drawing the track in MGED. The initial and final points of a track and the vertices corresponding to the first and last collisions are never skipped. Consequently, the minimum number allowed in the text box “track” is 4. The difference of the sequence numbers of each skipped vertex and the previous skipped vertex is approximately constant. If the text box “track” is empty, then all vertices of each track will be shown (the same effect is achieved by entering “-1” in this text box).

The mentioned skipping of tracks and vertices does not affect the track statistics shown in the “Track statistics” dialog window (see Section 4) and the number of tracks shown above the track filtering controls of the MCNScript main window (see Fig. 6.8). The track statistics are always calculated for all tracks and all vertices in the current set of tracks.

- If the checkbox “Show track segments” is checked, then the tracks segments defined in the “Track statistics” dialog window (see Section 4.3) will be shown instead of the complete tracks. In this case, the number displayed in the text box “Number of tracks” after clicking “Refresh” (see Fig. 6.8) is the number of track segments obtained after applying both the track filtering and the track segment selection (the order in which these two operations are applied is also defined in the “Track statistics” dialog window).

### 6.5. Properties of materials

The grid control “Materials” at the bottom of the MCNScript main window (see Fig. 6.9) is used for specifying colors, transparencies and reflectivities of the materials defined in the MCNelectron input file, as well as for selecting the materials that should be shown in MGED. This grid control is updated after loading an MCNelectron input file (i.e., after clicking the buttons “Load MCNelectron input file...”, “Reload”, or “Make RT script”), but only if the checkbox “Check syntax” is checked (see Fig. 2.1). If necessary, new rows will be added to this grid control automatically in order to accommodate all materials. Below is the explanation of the meanings of all columns of this grid control:

- 1) The column “Material” contains the material identifiers. The format of a material identifier is “M<n>”, where “<n>” is a positive integer number. These are the same identifiers that are specified in the MCNelectron input file (see Section 4.2 of the MCNelectron user manual). If the simulation setup includes a cell that is not filled with any material (i.e., with a zero or negative number specified instead of the material identifier), then the corresponding material identifier in column “Material” would be replaced by the string “V<n>”, where “V” stands for “Void” and “<n>” is either the number 0 or a positive integer number, which is the opposite of the number written after the cell identifier in the definition of the corresponding cell of the geometry.

As explained in Section 4.2 of the MCNelectron user manual, materials can be optionally assigned custom names. If such a name has been assigned, it is shown in this column after the material identifier, in parentheses (see Fig. 6.9). If the sphere representing the source of particles is displayed, then the corresponding material is assigned the name “M\_source” (see Fig. 6.9).

The material identifiers and names are updated after loading an MCNelectron input file, i.e., after clicking the button “Load MCNelectron input file...” or “Make RT script” (see Fig. 2.1). The number of non-empty grid cells in the column “Material” (excluding the cell with the string “M\_source”) is equal to the number of materials defined in the MCNelectron input file.

- 2) The column “Color” shows the colors of the materials. Those colors are used both for displaying the 3D objects in the raytraced image and for the lines of the wireframe view (see Fig. 6.7).
- 3) The column “Color name” contains the dropdown list controls for selecting the color of each material. The user may select one of 14 standard colors, or a custom color. In order to set a custom color, the last entry of the dropdown list (“Custom”) must be selected. Then the standard Windows color selection dialog is opened.

No.	Material	Color	Color name	R	G	B	Transp.	Refl.	Show
1	M1 (Argon)		Custom	128	255	64	0.3	0	<input checked="" type="checkbox"/>
2	M2 (Water)		Custom	255	64	64	0.3	0	<input checked="" type="checkbox"/>
3	V0 (Empty space)		Custom	128	255	255	0.3	0	<input checked="" type="checkbox"/>
4	M_source		Yellow	255	255	0	0.3	0	<input checked="" type="checkbox"/>
5			Pink	255	192	203	0.3	0	<input checked="" type="checkbox"/>
6			Lime	191	255	0	0.3	0	<input checked="" type="checkbox"/>
7			Amber	255	191	0	0.3	0	<input checked="" type="checkbox"/>
8			Cyan	0	255	255	0.3	0	<input checked="" type="checkbox"/>
9			Green	0	255	0	0.3	0	<input checked="" type="checkbox"/>

Fig. 6.9. The grid control for specifying properties of materials and for selecting the materials to show in MGED.

- 4) The columns “R”, “G”, “B” contain the three color components (red, green and blue) of the corresponding color. They can be edited manually, copied and pasted. The allowed values of each color component are from 0 to 255. A change of any of these three numbers causes an update of the grid cells in columns “Color” and “Color name” of the same row.
- 5) The column “Transp.” contains values of transparency of each material. The allowed values of transparency are from 0 to 1, where 0 corresponds to a completely opaque material, and 1 corresponds to a perfectly transparent material. The default transparency is 0.3 (the effect of setting this value of the transparency is apparent in Fig. 5.1 and Fig. 6.7).
- 6) The column “Refl.” contains values of reflectivity of each material. The allowed values of reflectivity are from 0 to 1, where 0 corresponds to a completely non-reflecting material, and 1 corresponds to a perfectly reflecting material. The default reflectivity is 0.
- 7) The column “Show” contains checkboxes for selecting the materials that should be shown in the MGED Graphics Window and in the lingering window. If the checkbox is checked, then all 3D objects using the corresponding material will be shown, otherwise they will not be shown.

## 7. Manipulating the view in the MGED environment

The RT script created by MCNScript is a set of MGED commands. The same set of commands could be entered manually after the “mged>” prompt in the MGED Command Window (see Fig. 5.1), with exactly the same results. Below is a description of several MGED commands that could be used to change some features of the image displayed in the MGED Graphics Window without the need to re-create the entire RT script. All those commands, excluding the “rt” command, affect only the wireframe view, the display of particle tracks and the beam axis; the raytraced image, if present, is not affected. Since any change of the view parameters would make the raytraced image outdated, it should be removed from the MGED Graphics Window before making any changes. This can be done in the MGED Raytrace Control Panel, which is opened by selecting the MGED menu command “File / Raytrace”. An example of the dialog window opened after selecting the mentioned menu command is shown in Fig. 7.1. The background color can be specified either by entering its RGB color components in the text box “Background Color”, or by selecting one of the standard colors from the list that is opened by clicking the button on the right of the mentioned text box (see Fig. 7.1). The color components, if entered manually, must be separated by spaces (the black color is specified by three zeroes “0 0 0”, and the default initial background color after starting MGED is a dark blue color, specified by “0 0 50”). In order to remove the raytraced image and apply the selected background color to the MGED Graphics Window, the button “fbclear” must be clicked (see Fig. 7.1). In order to create a raytraced image of the currently selected 3D objects in the MGED Graphics Window, using the current values of graphical parameters, the button “Raytrace” must be clicked. The mentioned actions will be successful only if the checkbox “Enable Framebuffer” is checked (see Fig. 7.1). In order to suppress reporting of object overlaps in the Command Window, the button “Advanced Settings...” must be clicked and the option “-R” must be typed in the text box “Other Options” (see Fig. 7.1).

The other actions that could be performed to change the contents of the MGED Graphics Window are described below. In the instructions given below, the term “command” means typing the specified text after the “mged>” prompt in the MGED Command Window (see Fig. 5.1) and pressing the “Enter” key. While the Command Window is active, the up and down arrow keys can be used for selecting a previously executed command.

- If the left mouse button is clicked on the image shown in the MGED Graphics Window, the view size will be increased by a factor of 2. If the right mouse button is clicked, the view size will be decreased by half. In any case, the coordinates of the view center do not change.
- Pressing the left or right arrow key while the MGED Graphics Window is active would rotate the displayed lines (excluding the raytraced image, if it is shown) around the vertical axis passing through the center of the view by 1°. Pressing the up or down arrow key while the MGED Graphics Window is active would similarly rotate the view around the horizontal axis.

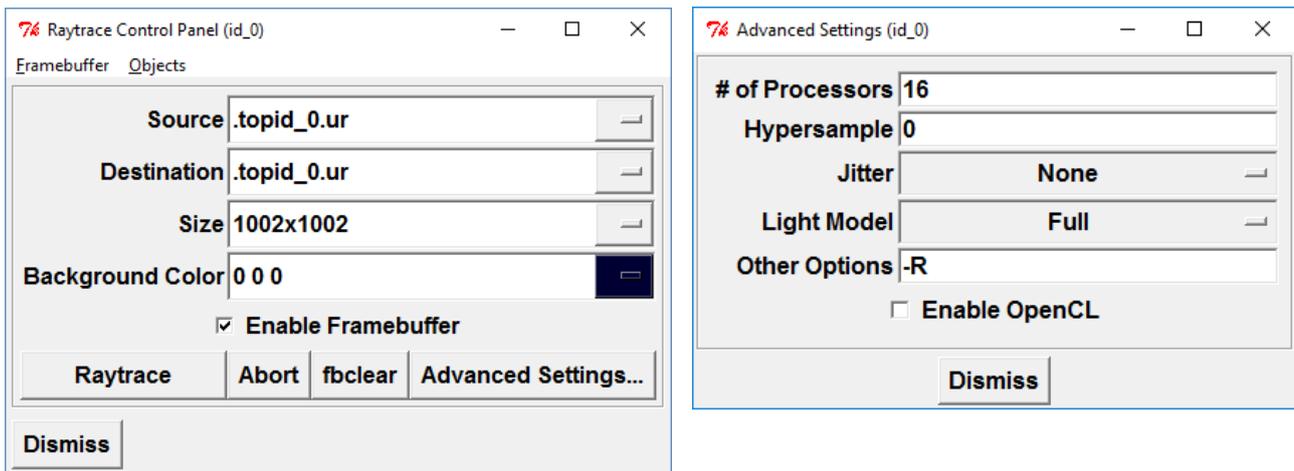


Fig. 7.1. The MGED Raytrace Control Panel (left) and the “Advanced Settings” dialog window (right).

- If the MGED Graphics Window is active, then pressing the left or right arrow key while the key “Shift” is pressed would shift the view right or left.
- If the MGED Graphics Window is active, then pressing the up or down arrow key while the keys “Ctrl” and “Shift” are pressed would shift the view down or up.
- If the MGED Graphics Window is active, then pressing the left or right arrow key while the keys “Ctrl” and “Shift” are pressed would change the “twist” angle of the view by 1° (pressing the right arrow would rotate the view clockwise, and pressing the left arrow would rotate the view counter-clockwise).
- Pressing the “X” key while the MGED Graphics Window is active would start continuous rotation of the displayed lines (excluding the raytraced image, if it is shown) around the horizontal axis passing through the center of the view. Pressing the “X” key again would speed up the rotation. Pressing “Shift+X” would slow it down or cause rotation in the opposite direction.
- Pressing the “Z” key while the MGED Graphics Window is active would start continuous counter-clockwise rotation of the displayed lines around the axis that is normal to the projection plane and passes through the center of the view. Pressing the “Z” key again would speed up the rotation. Pressing “Shift+Z” would slow it down or cause rotation in the opposite direction.
- The command “Z” is used to remove the wireframe view of the 3D objects and all supplemental lines (i.e., the particle tracks and the beam axis) from the MGED Graphics Window.
- In order to display only the 3D objects corresponding to a particular set of materials, one should use the command “draw”, followed by space-delimited identifiers of those materials. The identifiers must be constructed according to the rules described at the beginning of Section 6.5. For example, in order to display only the objects composed of the first three materials listed in the grid control of Fig. 6.9 and to hide the source of particles, the following command must be entered: “draw M1 M2 V0”. Before entering this command, the MGED Graphics Window should be cleared using the command “Z”. After the “draw” command is entered, the view is automatically resized and re-centered to accommodate all surfaces used to construct the objects composed of the listed materials.
- The command “view size” is used to get or set the view size. If this command is entered without any parameters, then the current view size will be displayed in the command window. In order to set the view size, it must be specified after “view size”. For example, in order to set the view size to 7 cm, the following command should be entered: “view size 7”. In GUI mode, the current view size is always shown at the bottom of the MGED Command Window, after “sz=” (see Fig. 5.1).

MCNScript always sets the internal MGED units of distance to centimeters and then uses an appropriate conversion factor if the actual distances are much less or much greater than 1 cm. Although there is the MGED command “**units**”, which allows setting other units of distance (for example, microns), they are not used, because it has been determined empirically that the highest accuracy of the image displayed in the MGED Graphics Window is achieved when MGED “thinks” that the view size is between 0.001 cm and 1000 cm. If the conversion factor is used, it is displayed in a message in the MGED Command Window after loading the RT script created by MCNScript. For example, if the initial view size specified in the MCNScript main window is of the order of  $10^{-4}$  cm, then the following message is displayed in the MGED Command Window after loading the RT script:

```
*****
Coordinate conversion: 1 MGED centimeter = 1 um = 0.0001 cm.
*****
```

In this example, the command “view size 7” would set the actual view size to 7  $\mu\text{m}$  (although in MGED internal units it would be 7 cm).

- The command “**view center**” is used to get or set the coordinates of the view center. If this command is entered without any parameters, then the current values of the X, Y and Z coordinates of the view center will be displayed in the command window. In order to set the coordinates of the view center, they must be listed after “view center” in the same order as mentioned above. For example, in order to set  $x$  to 1 cm,  $y$  to 0 and  $z$  to 2 cm, the following command should be entered: “view center 1 0 2”. In GUI mode, the current coordinates of the view center are always shown at the bottom of the MGED Command Window, after “cent=” (see Fig. 5.1).
- The command “**ae**” is used to get or set the values of the azimuth, elevation and twist angles (see Fig. 6.2 in Section 6.1). If this command is entered without any parameters, then the current values of the mentioned three angles will be displayed in the command window (in degrees). In order to set the angles, they must be listed after “ae” in the same order as mentioned above. For example, in order to set azimuth to  $90^\circ$ , elevation to  $35^\circ$  and twist to  $20^\circ$ , the following command should be entered: “ae 90 35 20”. In GUI mode, the current values of the azimuth, elevation and twist angles are always shown at the bottom of the MGED Command Window, after “az=”, “el=” and “tw=”, respectively (see Fig. 5.1).
- The command “**set perspective**” is used to get or set the perspective angle. If this command is entered without any parameters, then the current perspective angle will be displayed in the command window (in degrees). In order to set the perspective angle, it must be entered after “set perspective” (the allowed values are from 0 to 90). For example, in order to set the perspective angle to  $15^\circ$ , the following command should be entered: “set perspective 15”. The default value of the perspective angle is 0 (no perspective). This value is recommended for easiest interpretation of complex images.
- The command “**vdraw**” allows drawing of lines and polygons in the MGED Graphics Window. In MCNScript, this command is used for drawing the particle tracks and the symmetry axis of the incident beam (“beam axis”). Each of these two applications of the command “draw” requires building the corresponding named list of drawing commands. After the list is built, the corresponding set of lines can be drawn by entering two commands: “vdraw open <list\_name>” and “vdraw send”. The list name corresponding to the set of particle tracks is “tracks<n>”, where “<n>” is the track type number, and the list name corresponding to the beam axis is “beamAxis”. The track type numbers are shown in the first column of the track color grid control, which is in the dialog window “Track types and colors” (see Fig. 4.3). For example, in order to draw the tracks of the Compton recoil electrons, the commands “**vdraw open tracks7**” and “**vdraw send**” have to be entered. In order to draw the beam axis, the commands “**vdraw open beamAxis**” and “**vdraw send**” have to be entered. The lines are drawn using the current values of the view size,

view center, eye position and perspective (see above). It should be noted that the tracks of each type can also be drawn by loading the corresponding track script file (see Section 5).

- The command “**rt**” is used to open a “lingering window” with the raytraced image of the objects that were previously specified using the “draw” command (see above). The raytraced image is drawn using the current values of the view size, view center, eye position and perspective (see above). If the “rt” command is entered without any parameters, then the objects will be drawn on a black background, and all occurrences of object overlaps will be reported in the command window. In order to suppress reporting of the overlaps, the option “-R” must be specified after “rt”, i.e., “rt -R” (there must a space before “-”). In order to set a custom background color, its RGB components must be specified using the format “-C<R>/<G>/<B>”, where “<R>”, “<G>”, “<B>” are integer numbers from 0 to 255, which specify the red, green and blue components of the background color, respectively. For example, in order to raytrace on the white background, the following command should be entered: “rt -R -C255/255/255”. The lingering window created using the “rt” command will be automatically closed after clicking the left or right mouse button anywhere on the raytraced image, or after pressing any key on the keyboard while the lingering window is active. Repeated usage of the “rt” command creates additional lingering windows (the lingering windows that were created earlier are not closed automatically). **Note:** The use of the “rt” command in the RT scripts created by MCNScript is slightly more advanced than described above: the lingering window is not closed after clicking a mouse button or pressing a key on the keyboard, and repeated execution of the same script will close the lingering window that was created earlier, replacing it with a new lingering window. All lingering windows created by entering “rt” or “rt -R” in the Command Window should be closed before executing an RT script created by MCNScript, because otherwise MGED may crash.