Institute for Analytical Instrumentation,

Russian Academy of Sciences

Koltsov S.

Manual

Saint Petersburg
2006

Contents

Introduction

Virtual Device is an Add-on software to SIMION v.7. *Virtual Device* covers several features which are not fully developed in SIMION, such as fast creation of 3D geometry (CAD program), fast analysis of results of simulation in SIMION and fast creation of different ions distributions.

CAD program. The main idea of CAD program allowing fast creation of 3D geometry is following. It is known that ion optic often consists of typical electrodes like lenses or rods, therefore it is possible to create specific users' ion optics only by using standard components. However, SIMION itself provides possibility to create any desirable shape of electrodes, because there is special language and any user can develop his/her own geometry by using this language. On the one hand this gives the user a very powerful tool, but on other hand specific skills are required.

CAD program is based on the idea that a user can use finite number of electrodes to recreate specific geometry and each typical electrode can be represented as a bit of program in SIMION language. The CAD program has a number of well defined 3D geometrical objects like sphere, cone or bar, and each geometrical project coupled with bits of program in SIMION language. User can graphically recreate specific 3D geometry as a combination of standard electrodes by setting geometrical parameters of each electrode. Therefore total geometry can be transformed into a program in SIMION language (GEM file). The advantage of this way of creation is the following: 1. Very quick creation of 3D geometry. 2. Very simple way of rearranging of geometry (shifting moving electrodes relatively to each other). 3. Automatic transformation of 3D geometry into geometrical file (GEM file). The disadvantage of this way of creation is only one: not any 3D geometry can be created into *Virtual Device* due to finite number of electrodes. But this limitation can be overcome by increasing the number of typical electrodes. At this moment total number of typical electrodes is 14, which covers about 90-95 % of typical ion optics schemes.

Analysis of results of simulation. SIMION gives possibility to save results of simulation into a "txt" file and format of this file is open (see SIMION manual). Therefore user can write his/her own software to analyze results of simulation or use some well known mathematical software. Actually there are several standard procedures often used in ion optics: building ion distribution in different planes, phase distribution in different planes, creation of histograms of specific ion optics parameters (for example time of flight), calculation of resolution of energy analyzers and mass-spectrometers.

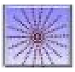
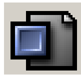
Virtual Device gives possibility to make such standard procedures as ion distribution, phase distribution in different planes, calculation of histograms of exit parameters (therefore definition of shape of peak), transformation of time of flight into mass/charge terms and calculation of mass resolution. It is also possible to transform data to Excel format.

Initial ion distribution. SIMION provides creation of initial ion distribution, but until recently some key features have not been realized in SIMION. At this moment SIS has developed new format of creation of ion distribution. But this feature is available only online (). *Virtual Device* gives possibility to create initial ion distribution as it was realized in version 7 and to create some specific ion distribution (for example Gauss distribution of energy).

Main menu of Virtual Device

The main menu looks as following:

Each pictogram in menu presents different parts of software.

 -close software,  - run program to create 3D model,  - create ions distribution,  - analyze SIMION data,  - create special ion distribution (disc),  - create special ion distribution (point source),  - about author.

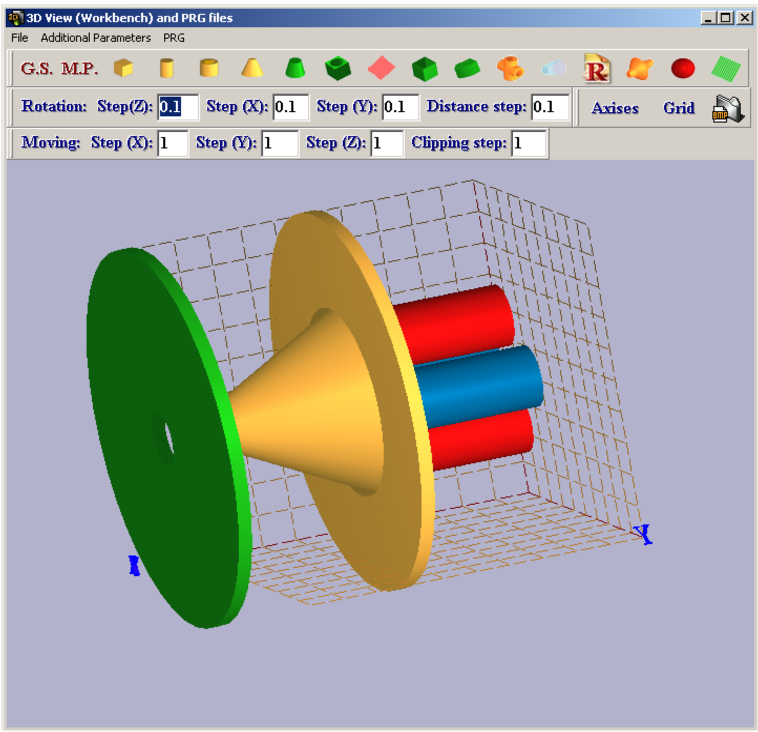
To run any part of software it is necessary just to click the appropriate button by mouse (left mouse button).

Chapter 1. CAD program.

Chapter 1.1 Menu.

This part of software covers creation of 3D project and transformation it to GEM file. It consists of three windows.

Workbench. The first one is workbench, where user can see 3D geometry. Below there is an example of 3D geometry (picture 1). You can rotate 3D geometry by moving mouse with left button pressed. The grid shows the size of the project. All parts, that are beyond the grid, will be cut by SIMION.

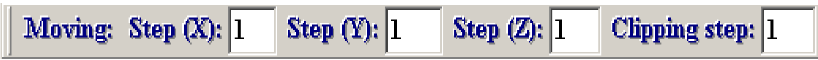


Picture 1. Workbench in *Virtual Device*.

This toolbar has a number of geometrical objects. To add a new object into project user can just press on one of those pictograms (but there is another way of adding of geometrical objects to project). The picture on pictogram presents geometrical type of geometrical object (except any axial object, see chapter 1.3).
G.S. – setting up geometrical size of project and all necessary parameters for SIMION.
M.P. – setting up mobile point.

This toolbar allows changing step of rotation of the project and step of moving of the project in space.

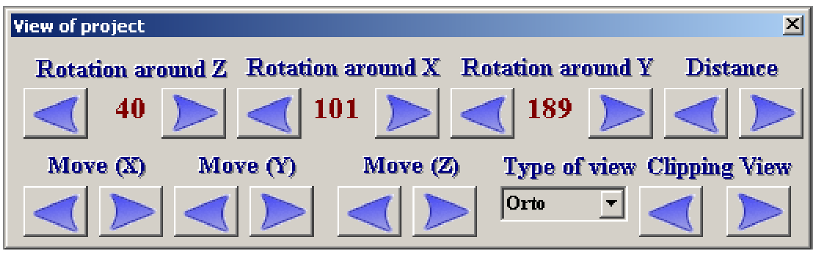
This toolbar allows switching on/of drawing of axis and grid, and also saving picture from workbench window into file (bmp format).



This toolbar allows changing the step of moving of the project in space.

The next window (toolbar) is intended for rotating and moving 3D geometry (picture 2). To avoid numerical pressing on mouse button during rotation of project user can only place cursor on one of buttons of rotation, press left mouse button and project starts rotate around axis with some step. Steps of rotation can be settled up in toolbar is described above.

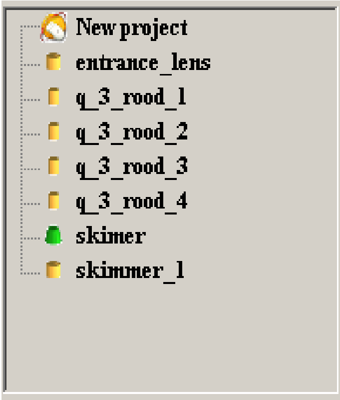
This toolbar also gives possibility of to change point of view (Type of view). *Clipping view* allows cutting off a part of 3D geometry and seeing geometry from the inside. Clipping plane is invisible, which is always parallel to the screen. By pressing mouse button on buttons of *Clipping view* user can move clipping plane forward/back compared with the plane of screen. When the clipping plane crosses a geometrical object, a part of an object, situated between the clipping plane and the plane of screen, will be cut off. Step of moving of the clipping plane can be settled up in toolbar described above (*Clipping step*).



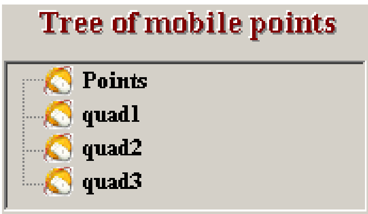
Picture 2. Toolbar for rotation, moving and clipping of project.

To move project in space (in workbench window) it is necessary to press buttons *Move X, Y, Z*. It can be done by left mouse button. Step of moving can be set up in a toolbar described above.

Working Window (picture 3). This window gives possibility to create project, to save project to file, to load project from file, to transform project to geometrical file, to add geometrical object to project, to delete geometrical object from object. This window also contains a tree of the project and a tree of mobile points.



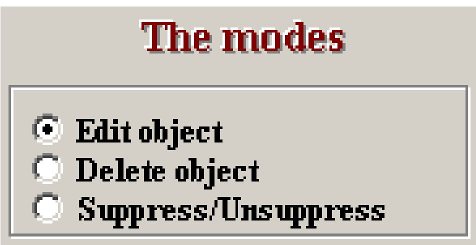
Tree of project. When user adds new geometrical object to project, a new item appears in the tree of project. The pictogram in tree shows what kind of object the user has added to project. The tree shows total number of geometrical objects in the project, type of object; the tree can also be used for editing, deleting and suppressing objects (see modes of working).

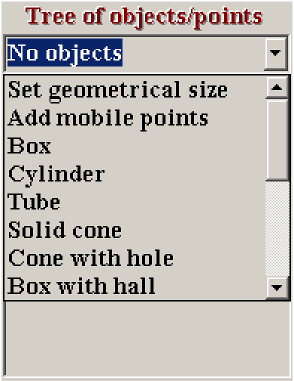


The tree of mobile points shows how many mobile points the user is using in the project. This tree can also be used for editing and deleting mobile points.

Picture 3. Working window.

The three modes of working with project. User can work with project in three modes. The first mode (**Edit object**) is the mode where user can add and edit parameters of geometrical object. To edit an object user has to pick up appropriate pictogram in the tree of project. The second mode (**Delete object**) is the mode where user can delete object from project (from tree). To delete geometrical object from project it is necessary, first, to choose mode 'Delete object', and second, to pick up appropriate pictogram in the tree of project. The third mode is '**Suppress/Unsuppress**'. It is the mode where user can make one of geometrical objects invisible without removing it from project. For this it is necessary, first, to choose mode 'Suppress/Unsuppress', and, second, to pick up appropriate pictogram in the tree of project. Then this geometrical object will become invisible in the workbench.





It is the box equivalent to the toolbox with a number of geometrical objects. This box can drop down with the list of objects. Therefore user can add new objects to project by choosing an object from the dropdown list. It is the second way of adding new objects into project.

Set geometrical size (G.S.) – setting up geometrical size of project and all necessary parameters for SIMION.

Add mobile points (M.P.) – setting up mobile point.

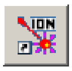
Buttons

 - (New project). This button allows creating a new project or clear old project. It means that the user deletes all geometrical objects from the project if they existed in the project.

 - Load old project from file to workbench.  - Save project to file.

 - Add a project from file to the project which already exists in the workbench.

 - Load project created in previous versions from file to workbench.

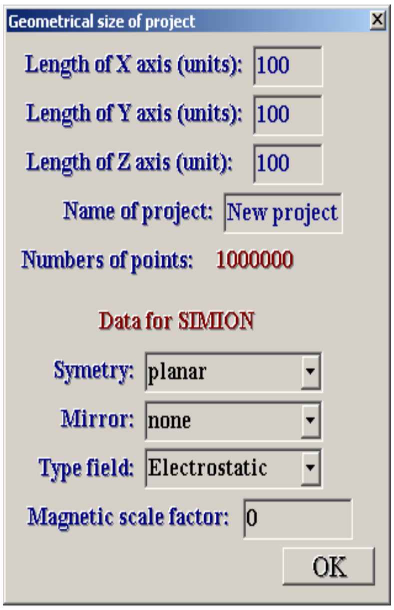
 - Close CAD program.  - convert 3D project to geometrical file (GEM). By pressing this button user expresses 3D geometry of the whole project in SIMION language.

Chapter 1.2. How to create 3D project.

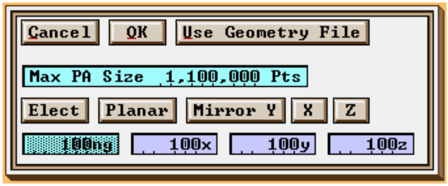
The creation of 3D project can be split into several steps. 1. Setting geometrical size of project and all necessary parameters for SIMION. 2. Creation of electrodes. 3. Transformation of 3D geometry into GEM file.

Step 1. Setting initial parameters of project can be done in the following way: Press button  or choose '**Set geometrical size**' from dropdown list of 'tree of objects/points'.
You will see a window (picture 4) which is equivalent to window in SIMION (picture 5).
Length of X axis – length of project along X axis. **Length of Y axis** – length of project along Y axis. **Length of Z axis** – length of project along Z axis. **Name of project** – this name is just a name of project and it is not used in SIMION. **Symmetry** –defines the type of symmetry. **Planar** means 3D volume (see SIMION manual). **Mirror**: here user can set up type of mirror. For example, **none** means 3D volume does not have any symmetry. The details of types of symmetry are discussed in chapter 1.4. **Type field** –defines what kind of field can be used (electrostatic or magnetic field). **Magnetic scale factor** – default value: zero. When user has settled up geometrical sizes along the axes, *Virtual Device* calculates total number of cells. If total number of cells exceeds the value 50 000000 then *Virtual Device* warns you about it (see picture 6). This

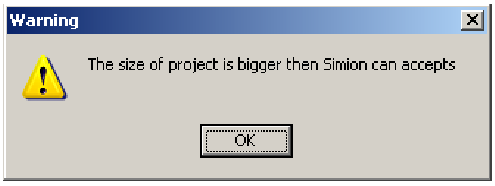
warning does not play any role for creation of 3D geometry in *Virtual Device*, but SIMION does not accept geometry file with total number of cell more than 50 000000.



Picture 4. Geometrical size of project in *Virtual Device*.



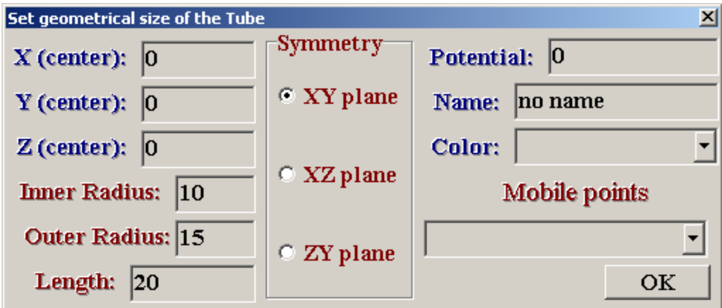
Picture 5. Geometrical size of project in SIMION.



Picture 6.

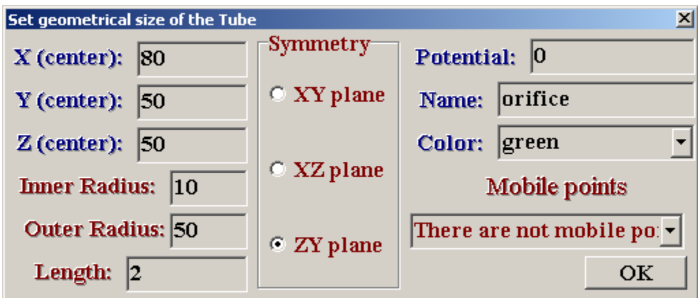
Step 2. To create/add geometrical objects to project user can choose appropriate pictogram from the toolbar or from the dropdown list.

For example, let us build a system which consists of an orifice, a conical skimmer and a quadrupole (four rods). Set up geometrical size of project as is shown on picture 4. Orifice is a circular plate with hole. Pick up pictogram tube . You will see the following window (picture 7):



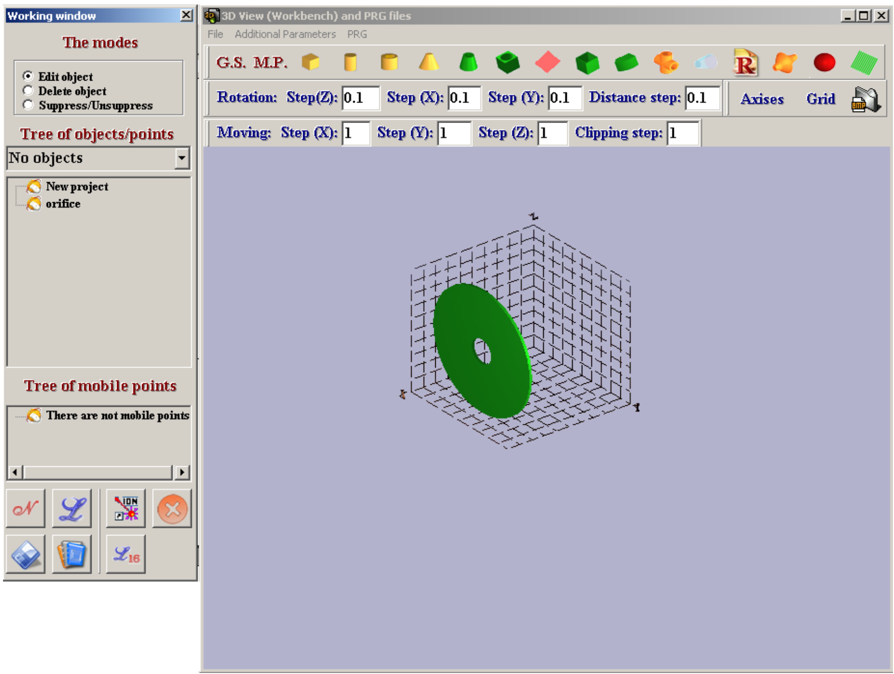
Picture 7. Setting geometrical parameters of tube.

Specify all parameters as is shown on picture 8. The details of all parameters of different geometrical objects are described in chapter 1.3.



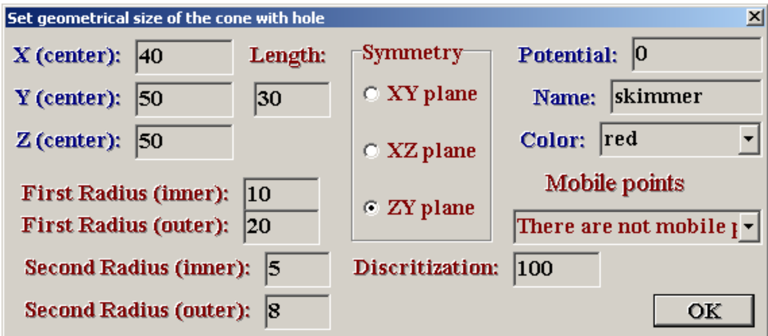
Picture 8.

Now press **OK** and you will see the following picture in workbench (picture 9).



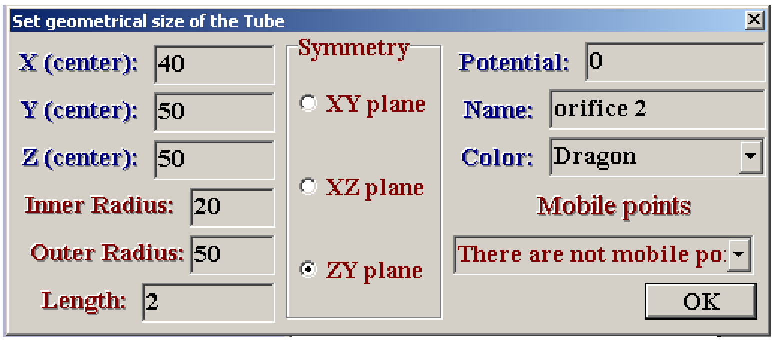
Picture 9.

Now we will add conical skimmer. Choose  or **cone with hole** from dropdown toolbar and specify geometrical parameters as following (picture 10):



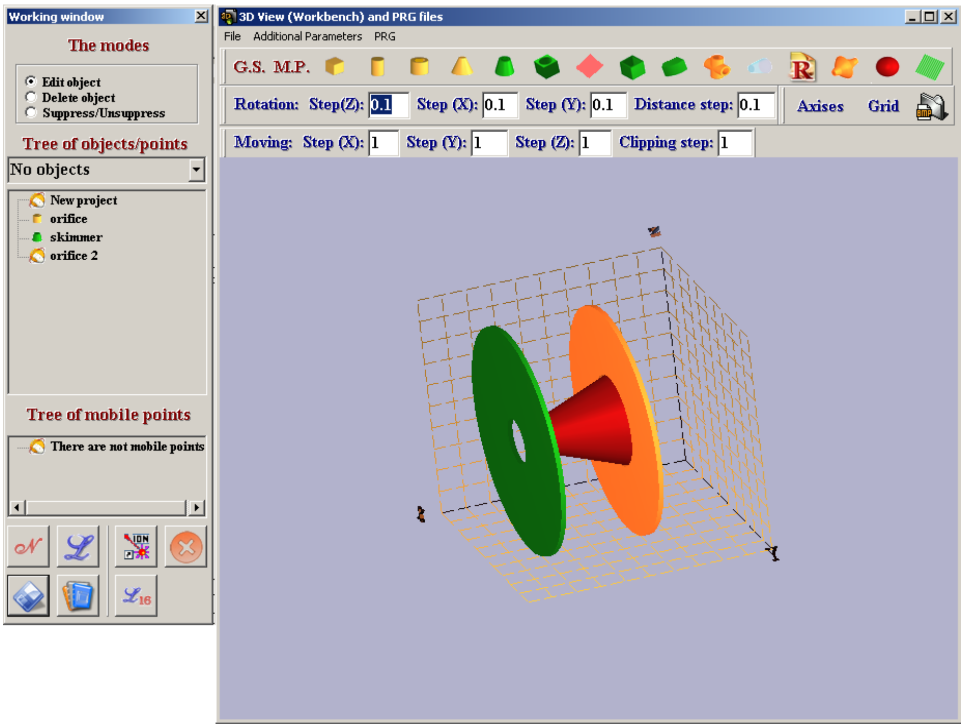
Picture 10. Setting parameters of cone with hole.

Now we will add an electrode which looks like orifice. Repeat instructions for orifice, and set up the following parameters (picture 11):



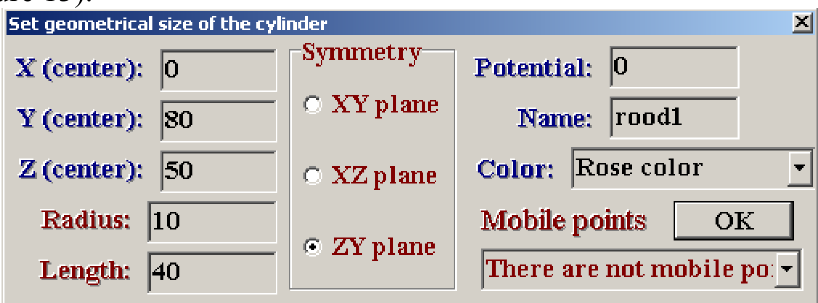
Picture 11.

Now you can see the following picture in workbench (picture 12).

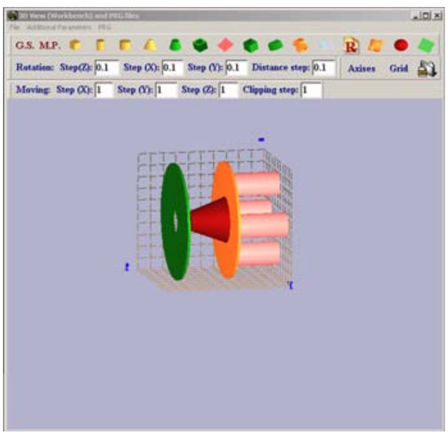


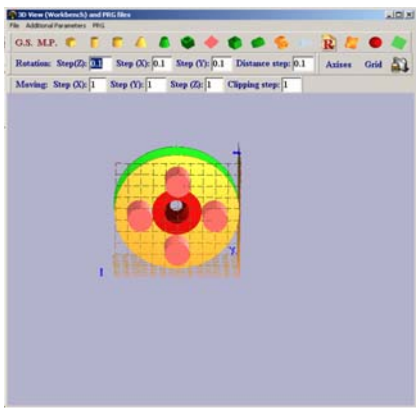
Picture 12.

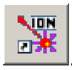
Now we will add four rods (a quadrupole) to our project. Pick up  and specify the following parameters (picture 13).

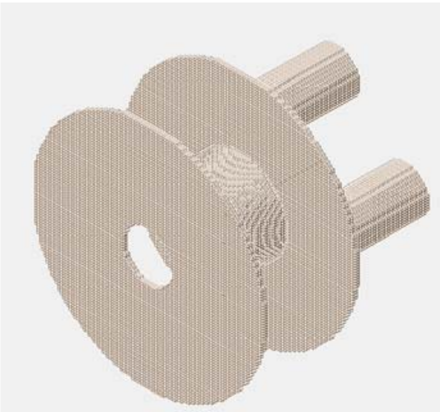


Repeat adding rods 3 times with the following parameters:
Y (center): 20, Name: rod2,
Z (center): 20, Name: rod3,
Z (center): *0, Name: rod4,
At last you will get the following project (pictures 14, 15):

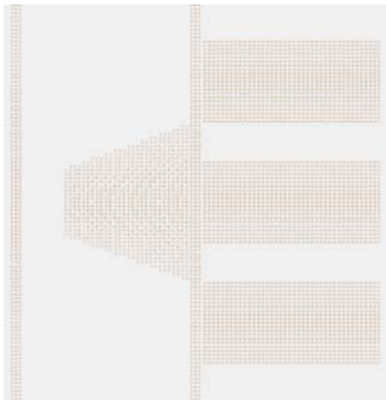

Picture 14.


Picture 15.

Step 3. Transformation of project to SIMION. To transform a 3D project into geometry file (GEM file) you need just topress button . Then specify the folder and the file name (for example, my_skimmer). *Virtual Device* will create file my_skimmer.gem.
Run SIMION, open geometry file and you will see the following picture in SIMION workbench (pictures 16, 17):



Picture 16.

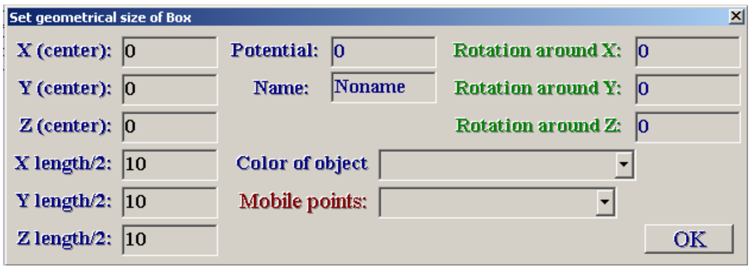


Picture 17.

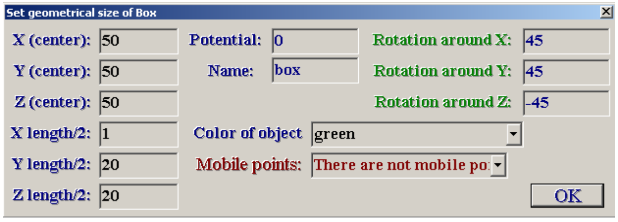
To change parameters of geometrical object it is necessary just to pick up appropriate pictogram from tree of object (mode: Edit object).
Now you can judge how easy it is to create a complicated 3D project for SIMION investigations.

Chapter 1.3. Description of geometrical objects.

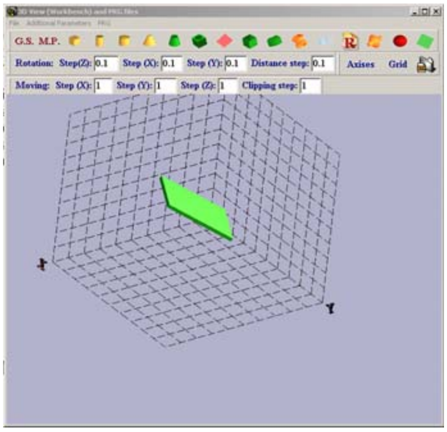
Object: BOX. This object presents itself box with the following parameters:



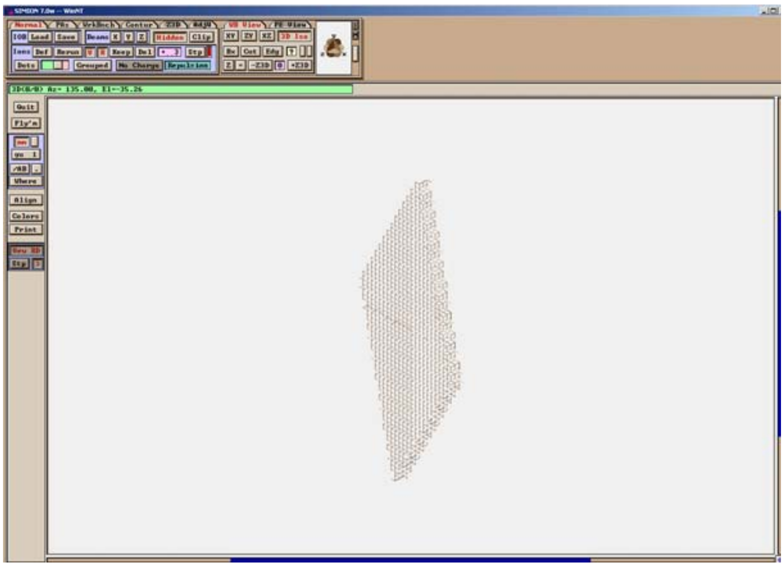
X,Y, Z (center) – position of center of box. **X, Y, Z length/2** – half a width along the axes. **Potential**: it is potential in volts. **Name**: it is just the name of the object. It is better to give different names to different electrodes (this is a suggestion). Name does not play any role. **Color of object**: here user can specify color of object. Color of object is not used in SIMION. **Rotation around X, Y, Z**: here user can specify angle of rotation around the axes. Angle is in degrees. **Mobile point**: here user can set connection between the geometrical object and the mobile point (if a mobile point was added to the project). Details of using mobile points see in chapter 1.4.
Example:



Results

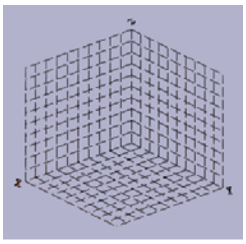


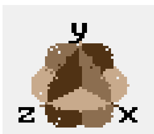
Workbench in *Virtual Device*.



Workbench in SIMION.

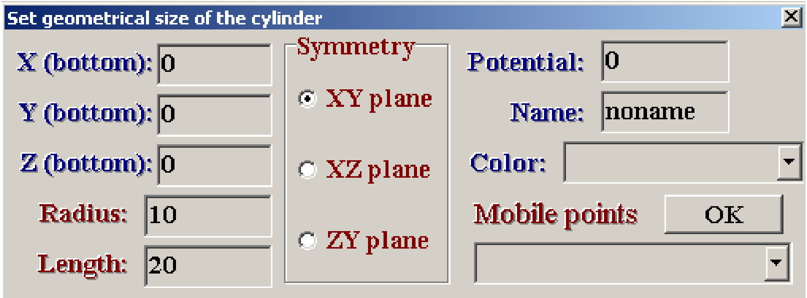
It is necessary to notice that default direction of axes in *Virtual Device* is different from direction of axes in SIMION.





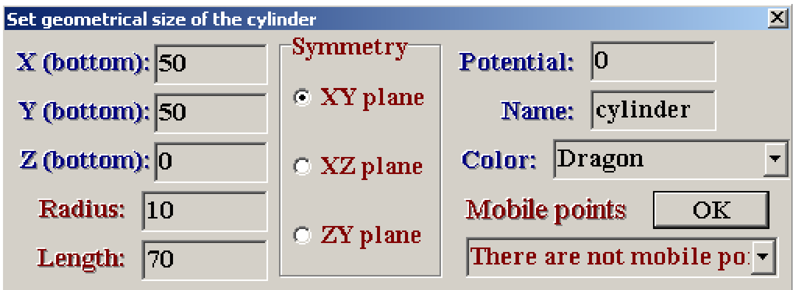
User can rotate axes in *Virtual Device* to get the same direction of axes in SIMION.

 **Object: Cylinder.** This object presents itself as a solid cylinder with the following parameters:

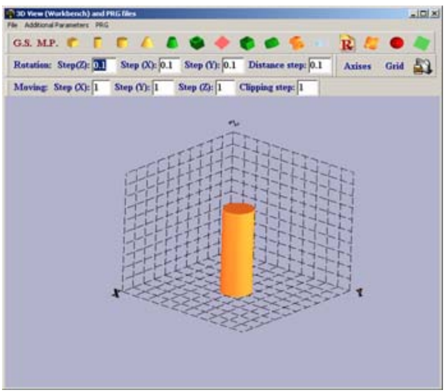


X,Y, Z (bottom) – position of bottom of cylinder. **Radius** – radius of cylinder. **Length** – length of cylinder. **Symmetry** – this option defines plane of bottom of cylinder. For example, if ZY plane was choosen than cylinder will be along X axis. **Potential**: it is potential in volts. **Name**: it is just a name of object. **Color of object**: here user can specify color of object. Color of object is not used in SIMION. **Mobile point**: here user can set connection between geometrical object and mobile point (if a mobile point was added to the project).

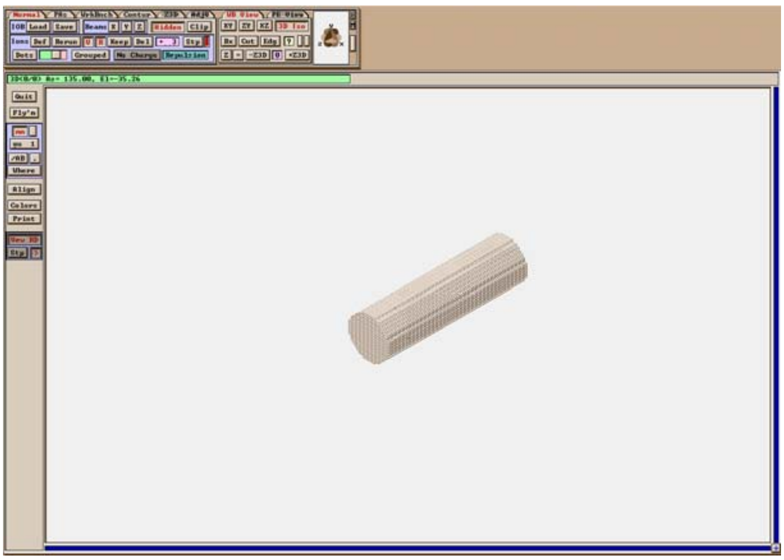
For example:



Results:

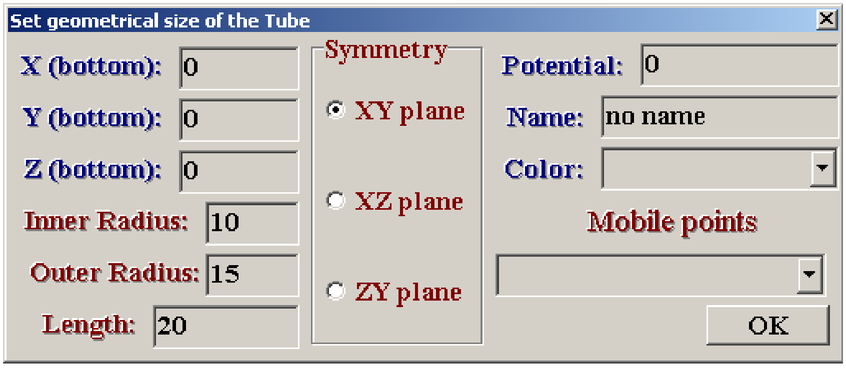


Workbench in *Virtual Device*.



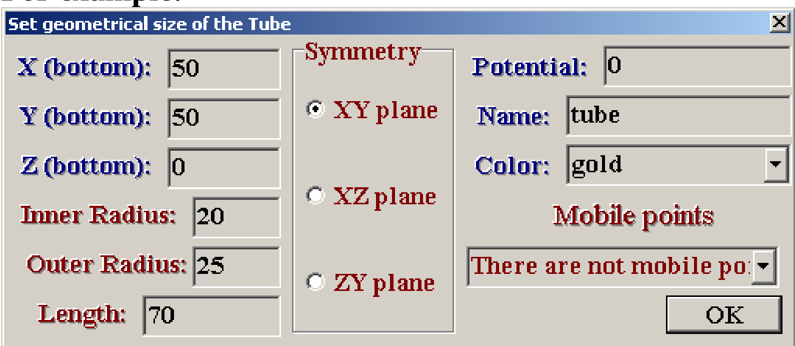
Workbench in SIMION.

 **Object:Tube.** This object presents itself as a tube with the following parameters:

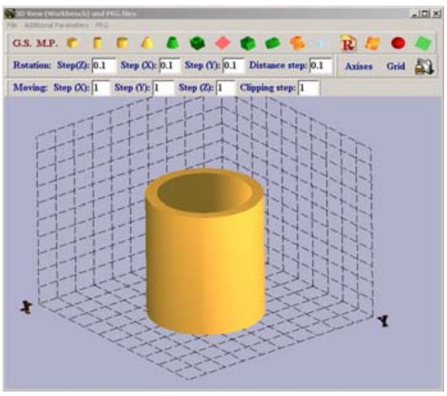


X, Y, Z (bottom) – position of bottom of tube. **Inner Radius** – inner radius of tube. **Outer Radius** – outer radius of tube. The difference between outer radius and inner radius is thickness of tube. **Length** – length of tube. **Symmetry** – this option defines plane of bottom of tube. **Potential**: it is potential in volts. **Name**: it is just a name of object. **Color of object**: here user can specify color of object. Color of object is not used in SIMION. **Mobile point**: here user can set connection between the geometrical object and the mobile point (if a mobile point was added to the project).

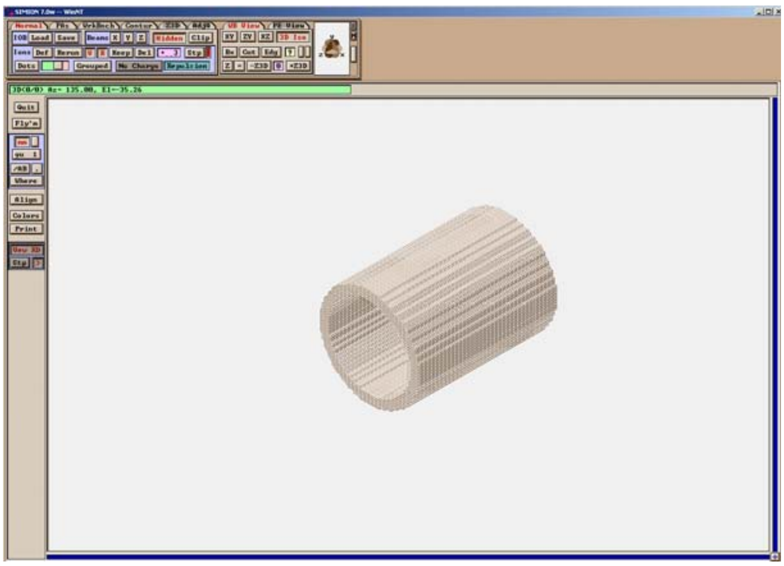
For example:



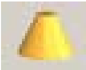
Results:

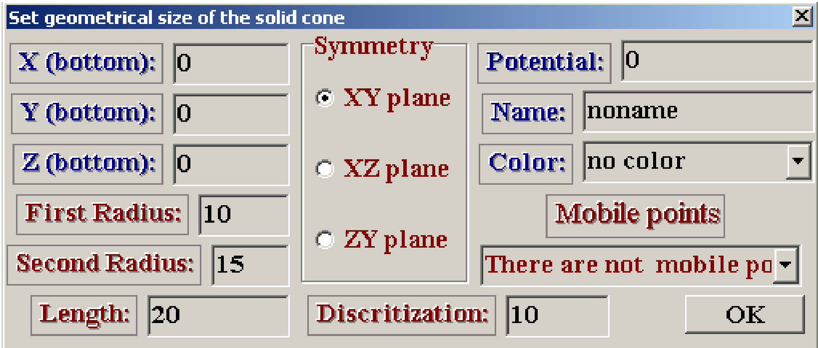


Workbench in *Virtual Device*.



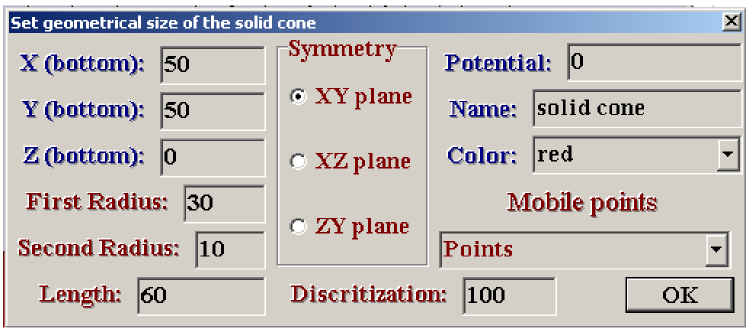
Workbench in SIMION.

 **Object: Solid cone.** This object presents itself as a solid cone with the following parameters:

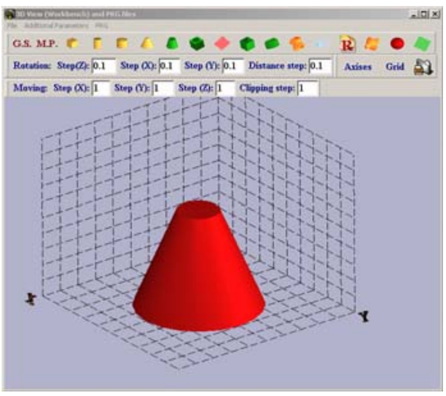


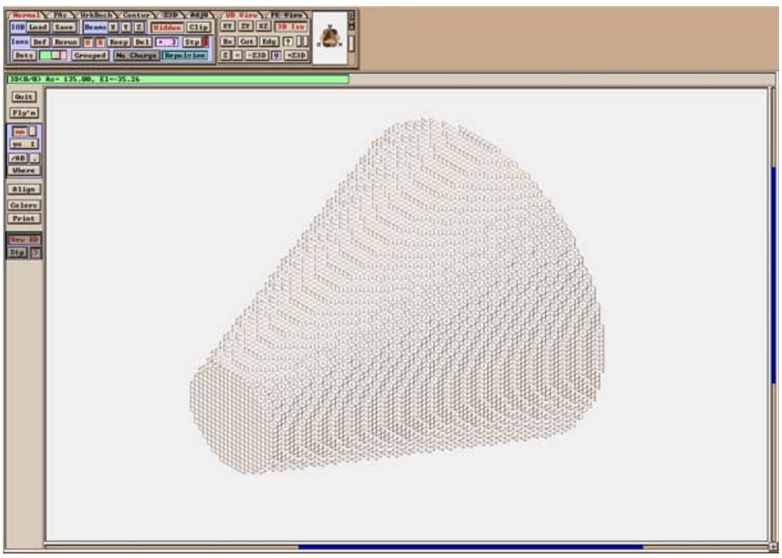
X, Y, Z (bottom) – position of bottom of solid cone. **First Radius, Second Radius** – radiuses of both ends of the cone. **Length** – length of cone. **Symmetry** – this option defines plane of bottom of cone. **Potential**: it is potential in volts. **Name**: it is just a name of object. **Color of object**: here user can specify color of object. Color of object is not used in SIMION. **Mobile point**: here user can set connection between the geometrical object and the mobile point (if a mobile point was added to the project). **Discritization**: this parameter defines how a solid cone can be made in SIMION. Actually a solid cone as a geometrical primitive does not exist in SIMION language. But it can be build as a number of slices with different radiuses. Therefore discritization is the number of slices. This parameter influences only 3D view in SIMION.

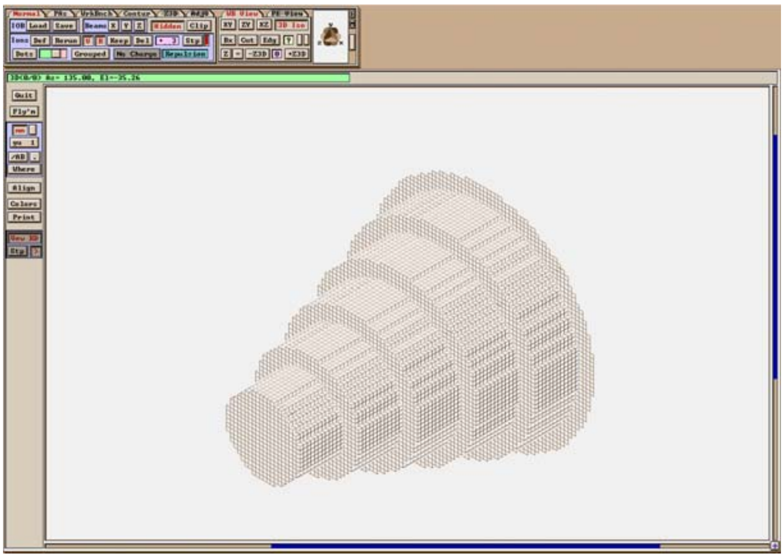
For example:



Results:


Workbench in *Virtual Device*.


Workbench in SIMION. Discritization:100

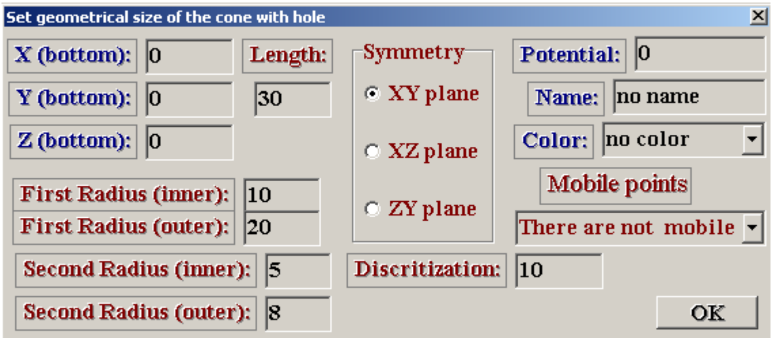


Workbench in SIMION. Discritization:5

So user can see the difference in pictures (SIMION workbench) for different values of discritization.

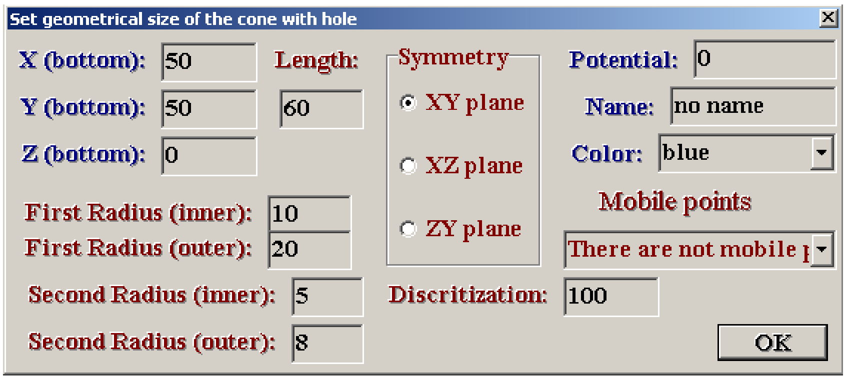
Therefore user has to remember about this parameter. When user defines value of discritization, it means that user defines the thickness of slice. The thickness of slice has to be less than or equal to length of cell. It is because roughness of geometry is completely defined by length of cell.

 **Object: Cone with hole.** This object presents itself as a cone with hole with the following parameters:

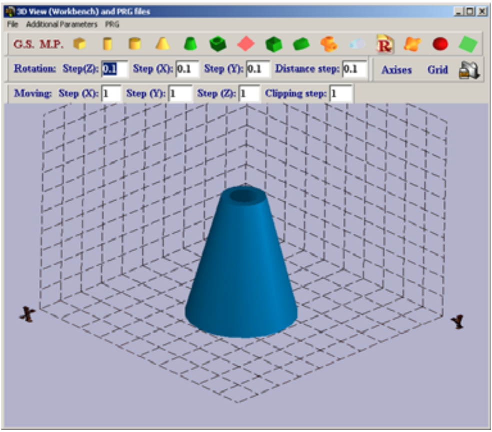


X,Y, Z (bottom) – position of bottom of cone. **First Radius (inner), First Radius (outer)** – outer radius defines radius of cone at one end of cone, inner radius defines radius of hole at the same end of cone. **Second Radius (inner), Second Radius (outer)** – outer radius defines radius of cone at the second end of cone, inner radius defines radius of hole at the second end of cone. **Length** – length of cone. **Symmetry** – this option defines plane of bottom of cone. **Potential**: it is potential in volts. **Name**: it is just a name of object. **Color of object**: here user can specify color of object. Color of object is not used in SIMION. **Mobile point**: here user can set connection between the geometrical object and the mobile point (if a mobile point was added to the project). **Discritization**: this parameter defines how cone can be made in SIMION. Actually cone as a geometrical primitive does not exist in SIMION language. But it can be built as a number of slices with different radiuses. Therefore discritization is number of slices. This parameter influences only 3D view in SIMION.

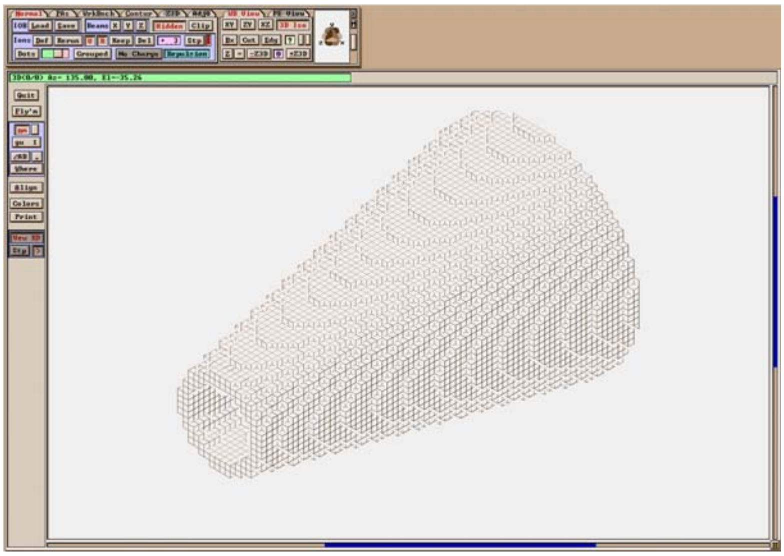
For example:



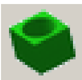
Results

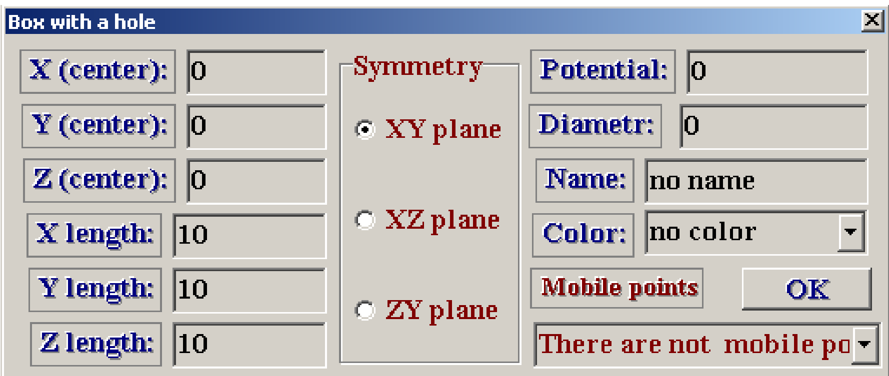


Workbench in *Virtual Device*.



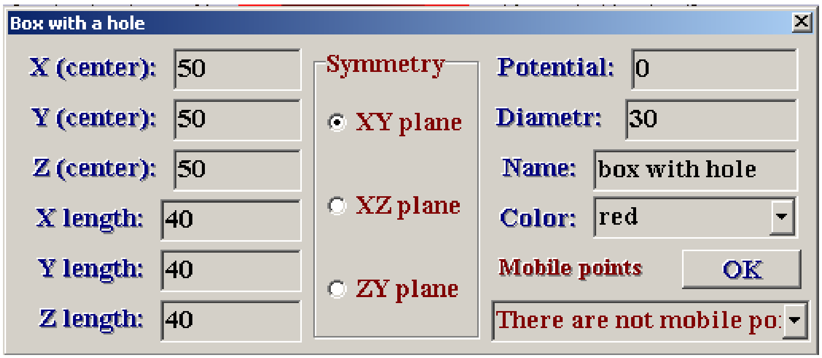
Workbench in SIMION. Discritization:100

 **Object: Box with hole.** This object presents itself as a box with hole with the following parameters:

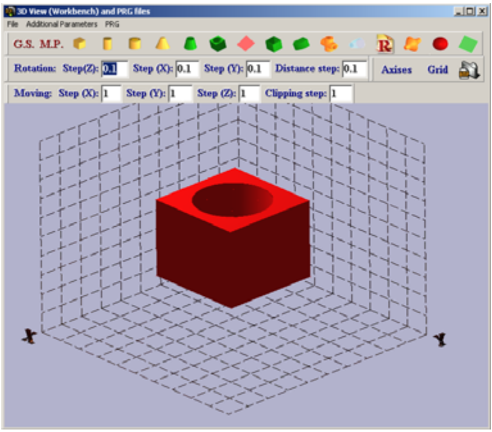


X, Y, Z (center) – position of box center. **X, Y, Z length** – length of box along axes. **Symmetry** – this option defines plane of bottom of box. **Potential**: it is potential in volts. **Diameter**: this parameter defines diameter of hole. If diameter exceeds length of box then diameter is automatically reduced. **Name**: it is just a name of object. **Color of object**: here user can specify color of object. Color of object a not used in SIMION. **Mobile point**: here user can set connection between the geometrical object and the mobile point (if a mobile point was added to the project).

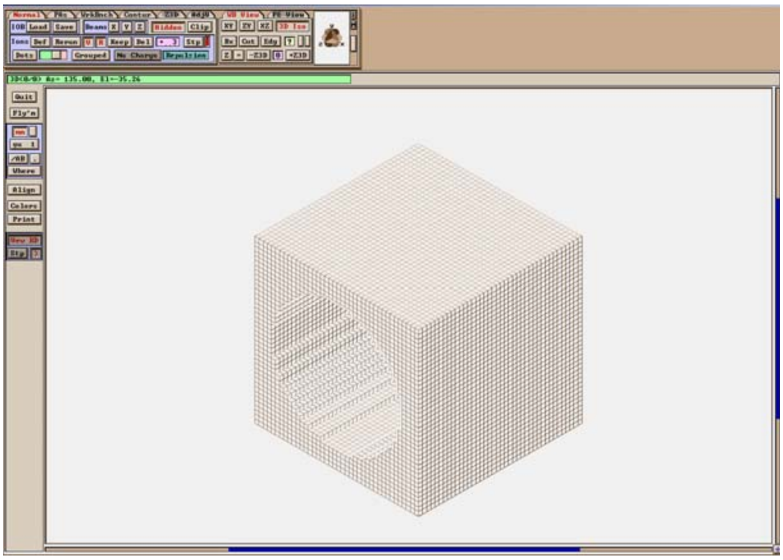
For example:



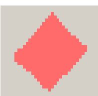
Results:

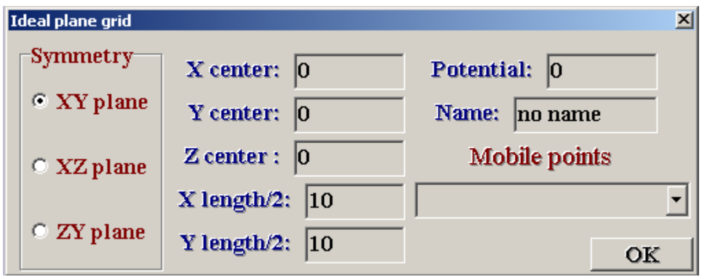


Workbench in *Virtual Device*.



Workbench in SIMION.

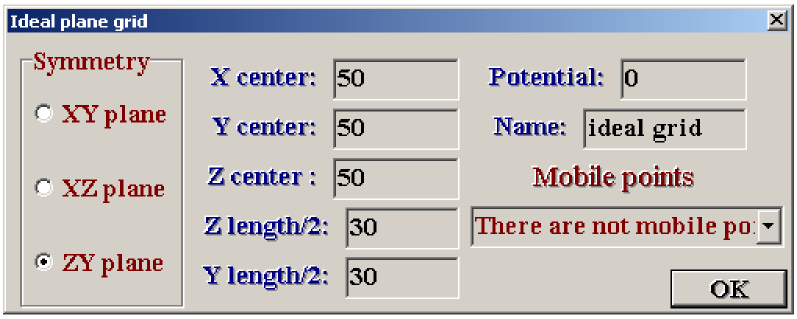
 **Object: Ideal plane grid.** This object presents itself as a box with thickness of one cell. According to SIMION rules any object with thickness about one cell will be regarded as ideal grid. Ideal grid is described by the following parameters:



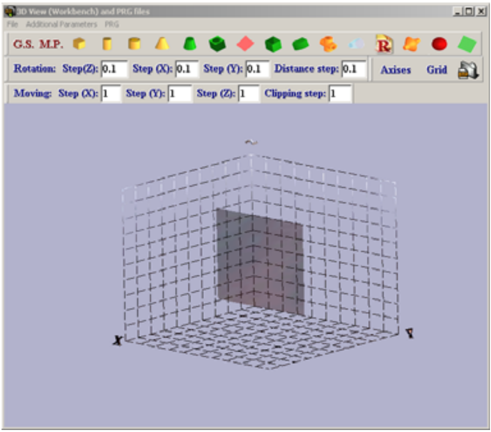
Symmetry – this option defines plane of grid. If XY plane is chosen then user has to specify length along X and Y axes. If XZ plane is chosen then user has to specify length along X and Z axes. If ZY plane is chosen then user has to specify length along Z and Y axes. **X, Y, Z (center)** – position of grid center. **Potential**: it is potential in volts. **Name**: it is name of object. **Mobile**

point: here user can set connection between the geometrical object and the mobile point (if a mobile point was added to the project).

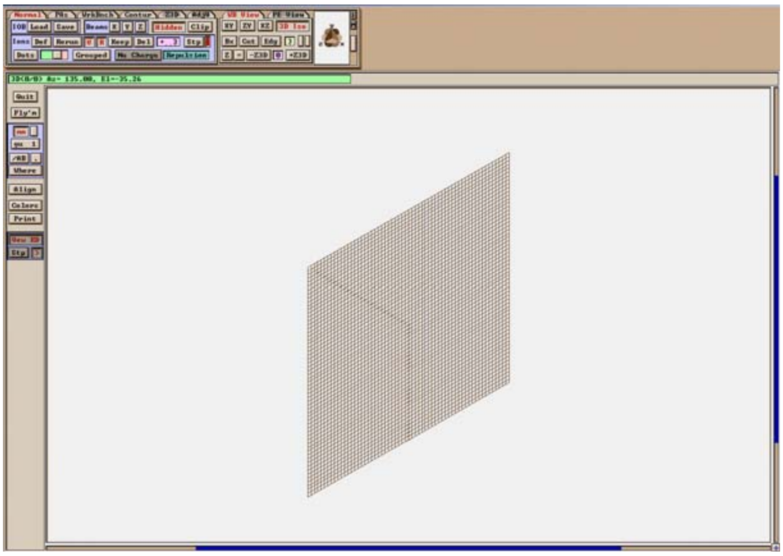
For example:



Results:

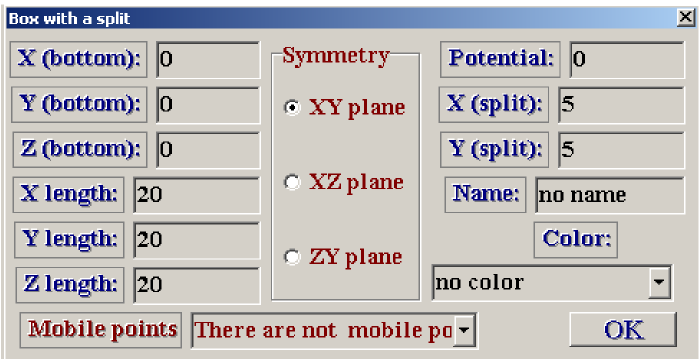


Workbench in *Virtual Device*.



Workbench in SIMION.

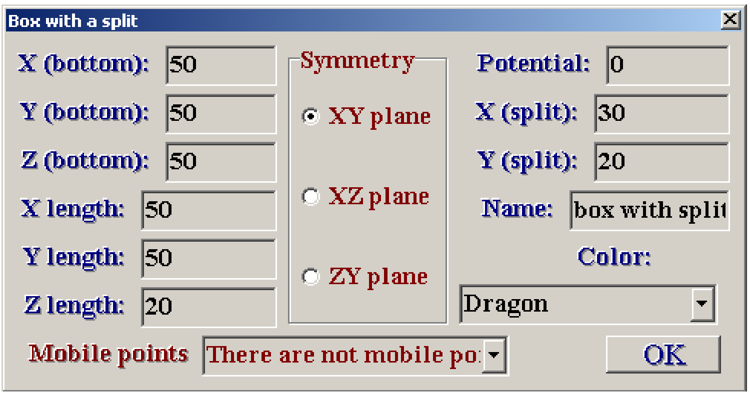
Object: Box with split. This object presents itself as a box with split with the following parameters:



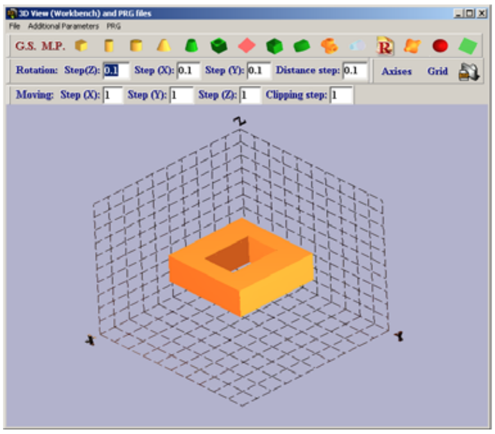
X, Y, Z (bottom) – position of bottom of box. **X, Y, Z length** – length of box along axes. **Symmetry** – this option defines plane of bottom of box. **Potential**: it is potential in volts.

Split: those parameters define radius of splits along appropriate axes. If radius exceeds length then radius is automatically reduced. **Name**: it is just a name of object. **Color of object**: here user can specify color of object. Color of object is not used in SIMION. **Mobile point**: here user can set connection between the geometrical object and the mobile point (if a mobile point was added to the project).

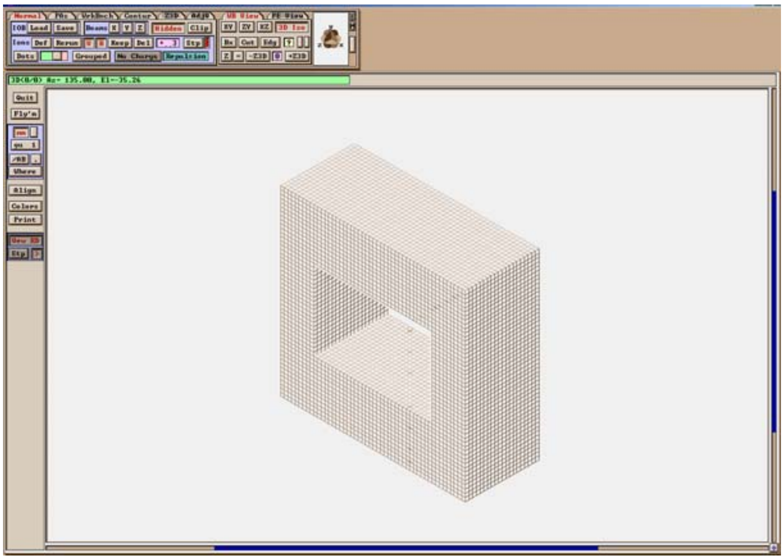
For example:



Results:

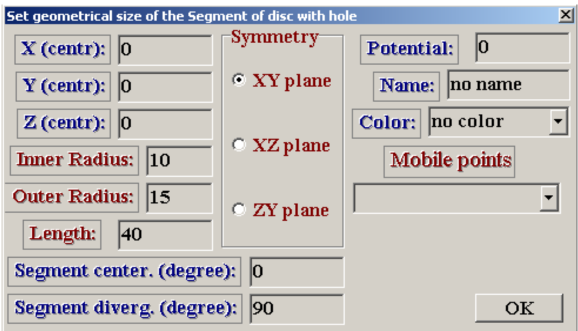


Workbench in *Virtual Device*.



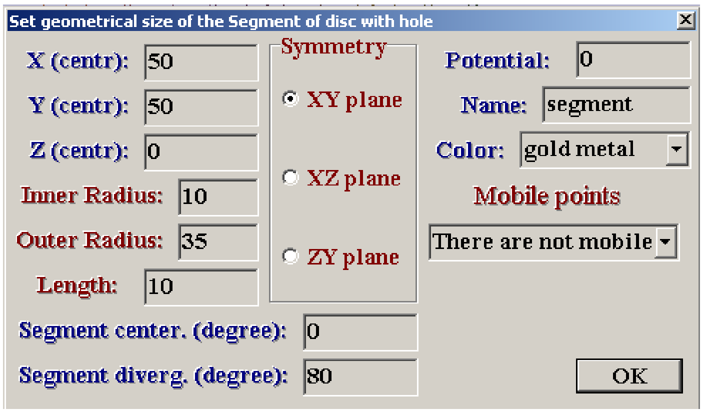
Workbench in SIMION.

with the following parameters:

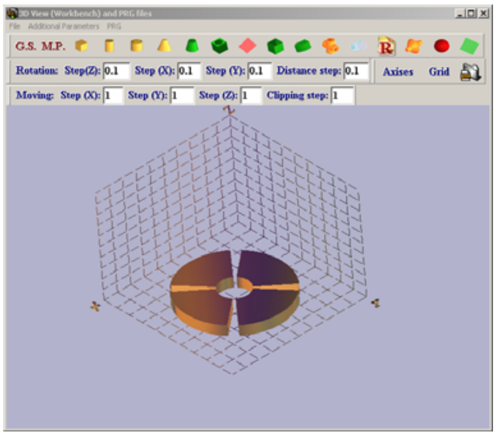


X, Y, Z (bottom) – position of bottom of segment. **Inner Radius** – radius of hole. **Outer Radius** – radius of segment. **Symmetry** – defines plane of segment. **Length** – defines thickness of segment in the plane of symmetry. **Potential** – potential in volts. **Name** – name of object. **Color of object**: here user can specify color of object. Color of object is not used in SIMION. **Mobile point**: here user can set connection between the geometrical object and the mobile point (if a mobile point was added to the project). **Segment center (degrees)**: this parameter defines position of segment center. This position is in degrees. **Segment diverg. (degrees)**: this parameter defines divergence angle of segment.

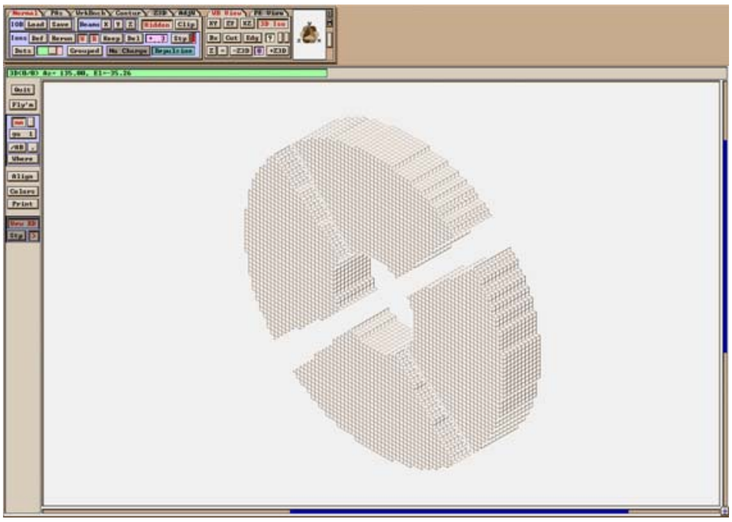
For example:

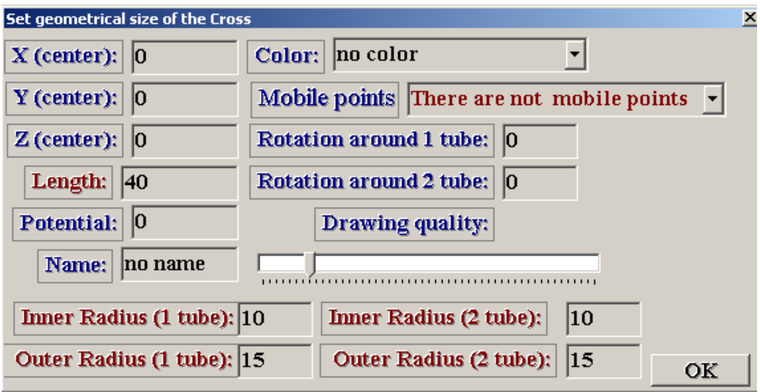


If you add four segments to a project with parameters as in example (above) and the following parameters:
first segment **Segment center (degree): 0,** second segment **Segment center (degree): 90,** third segment **Segment center (degree): 180,** fourth segment **Segment center (degree): 270,** then you will get the following project:
Results:



Workbench in *Virtual Device*.

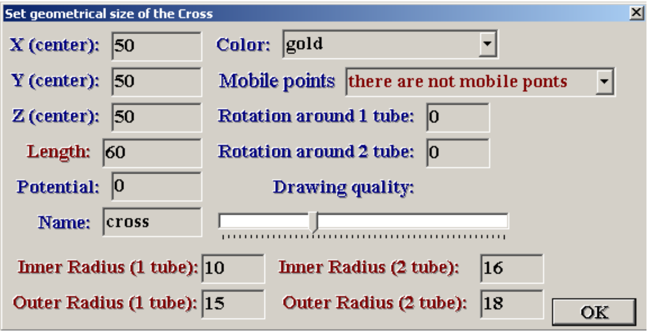




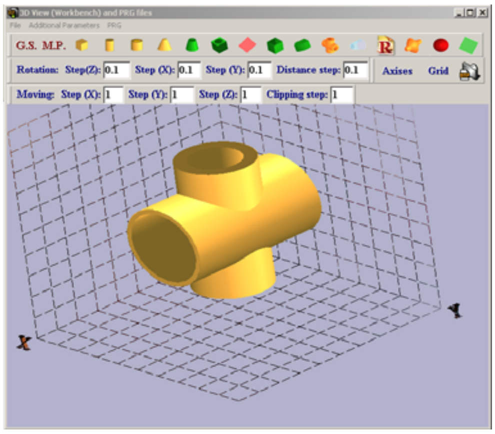
X, Y, Z (center) – center of crossing of tube. **Length** – length of both tubes. **Potential**: it is potential in volts. **Name** – name of object. **Color of object**: here user can specify color of object. Color of object is not used in SIMION. **Mobile point**: here user can set connection between geometrical object and mobile point (if mobile point was added to project).

Inner Radius (Tube 1), Outer Radius (Tube 1) – radiuses of the first tube. The difference between outer and inner radiuses is thickness of the first tube. **Inner Radius (Tube 2), Outer Radius (Tube 2)** – radiuses of the second tube. The difference between outer and inner radiuses is thickness of the second tube. **Rotation around Tube 1, Rotation around Tube 2** – those parameters define angle of rotation around tubes. **Potential** – potential in volts.

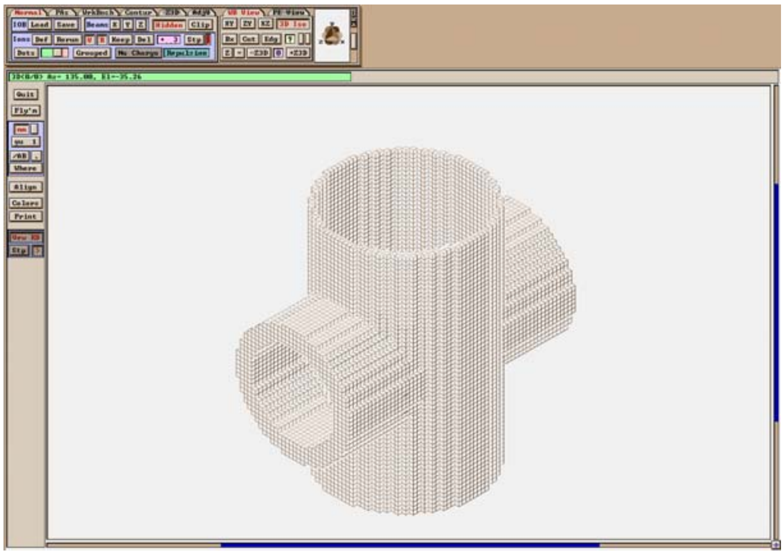
Example:



Results:

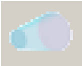


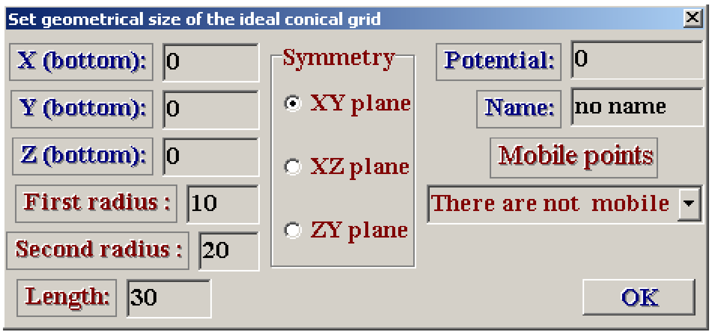
Workbench in *Virtual Device*.



Workbench in SIMION.

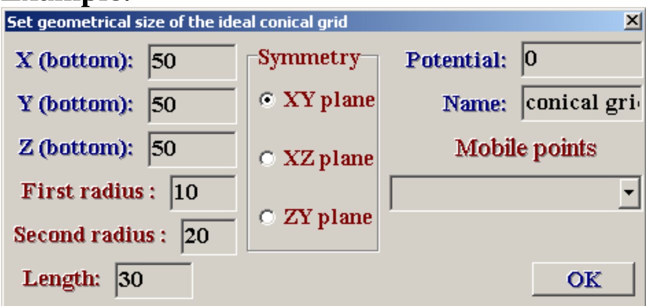
Drawing quality – this parameter defines quality of drawing of cross in *Virtual Device* only and it does not influence results of transformation of cross to SIMION. User can set up drawing quality of cross according of users' video card. The high value of drawing quality leads to decreasing of velocity of redrawing of cross during rotation.

 **Object: Ideal conical grid.** This object presents itself as an ideal conical grid with the following parameters:

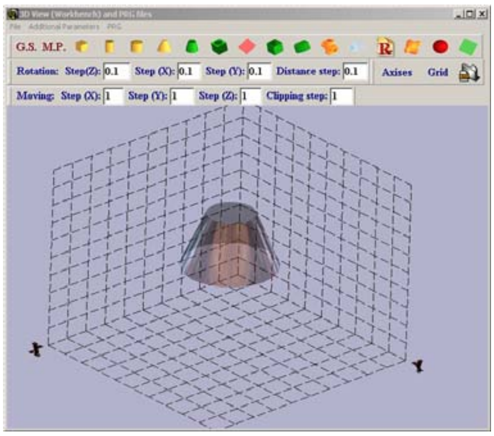


X, Y, Z (bottom) – these parameters define position of bottom of conical grid. **First, second radiuses** – radiuses of both ends of conical grid. **Length** – length of conical grid. **Symmetry** – this parameter defines plane of bottom. **Potential** – potential in volts. **Name** – name of object. **Mobile point**: here user can set connection between geometrical object and mobile point (if mobile point was added to project).

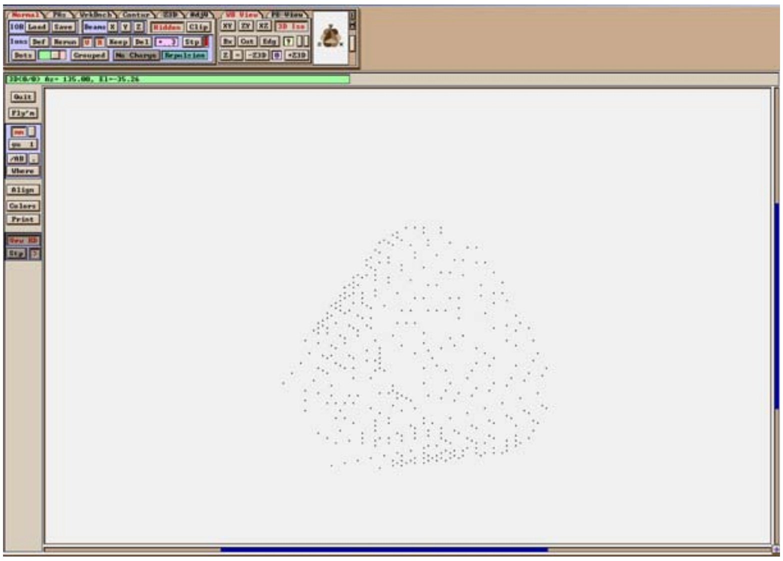
Example:



Results:

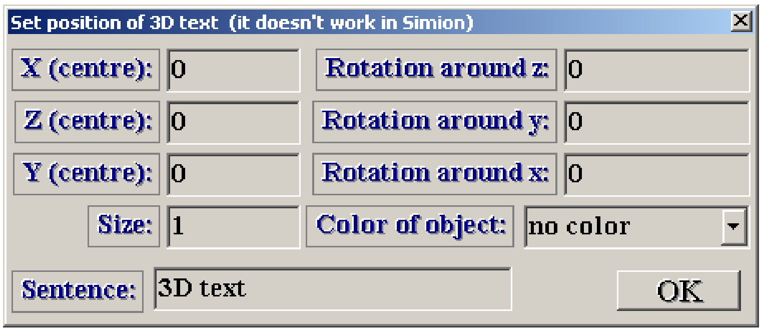


Workbench in *Virtual Device*.



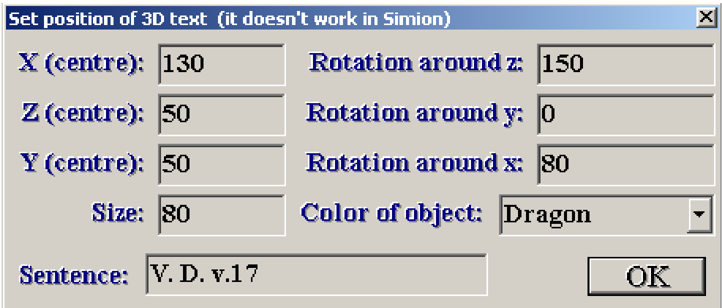
Workbench in SIMION. This grid is almost invisible in SIMION

 **Object: 3D sentence.** This object allows adding 3D letters to workbench. This object is not supported in SIMION:

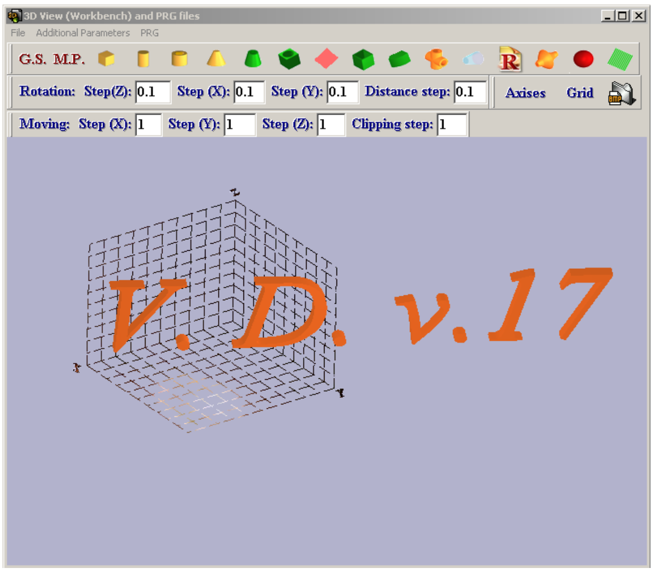


X, Y, Z (center) – these parameters define position of first letter in sentence. **Size** – this parameter defines the size of sentence. **Sentence** – here user can type letters. **Rotation around Z, Y, X**: those parameters define angles of rotation around axes (angles are in degrees). **Color**: this parameter defines color of object.

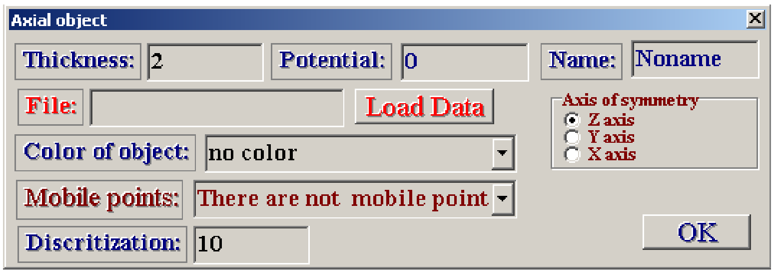
For example:



Results:



 **Object: Any axial object.** This object gives possibility to create a geometrical object with axial symmetry where shape / profile of electrode is given:



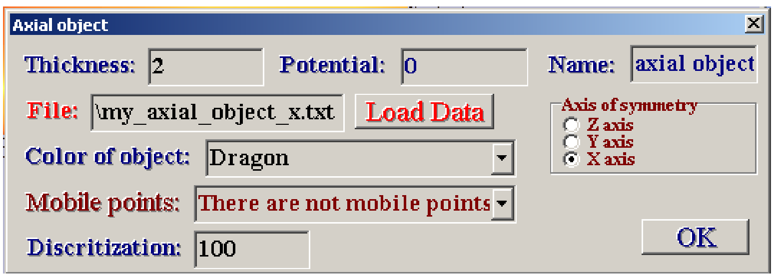
File – here user can specify the file where shape of electrode is stored. The format of file is discussed bellow. The file describes inner surface of electrode. **Thickness** – here user can set up thickness of electrode. The outer surface of electrode is defined as inner surface plus thickness. **Color of object**: here user can specify color of object. Color of object is not used in SIMION. **Mobile point**: here user can set connection between geometrical object and mobile point (if mobile point was added to project). **Potential**: potential in volts. **Name**: name of object. **Discretization**: this parameter defines how this geometrical object can be made in SIMION. Geometrical object with axial symmetry can be build as a number of slices with different radiuses. Each slice is presented as a round electrode with a hole. Radius of the hole is defined by data from file. The outer radius is radius of the hole plus thickness. Therefore parameter **discretization** is a number of slices. It influences only 3D view in SIMION. **Axis of symmetry** – here user can specify the axis of axial symmetry.

Format of file: The format of file with data consists of several columns. Examples of such files are placed in \examples_vds\axial object\
For example: my_axial_object_x.txt

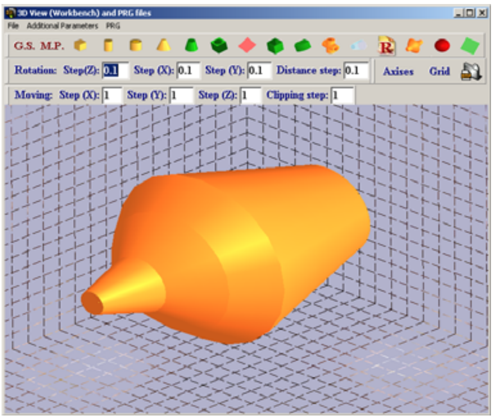
0 50 50 25
80 50 50 50
100 50 50 35
120 50 50 15
150 50 50 5

The first column is data along X axis, the second column is data along Y axis, and the third column is data along Z axis. The last column is radius.

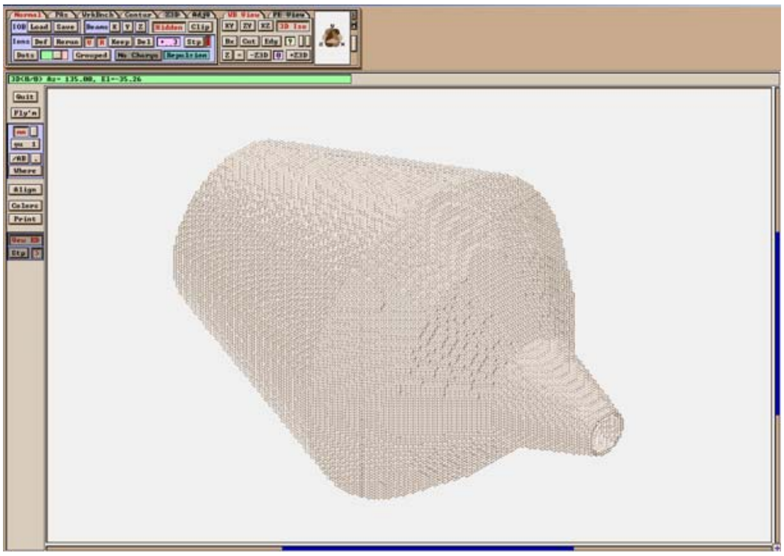
For example:



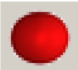
Results

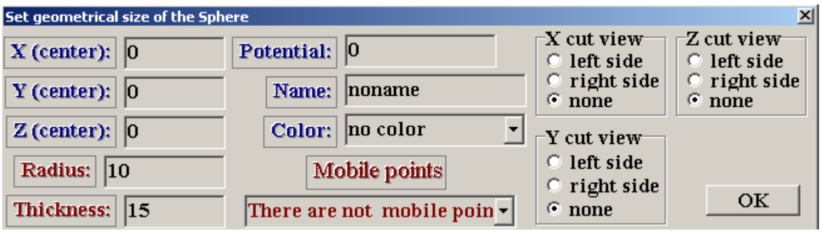


Workbench in *Virtual Device*.



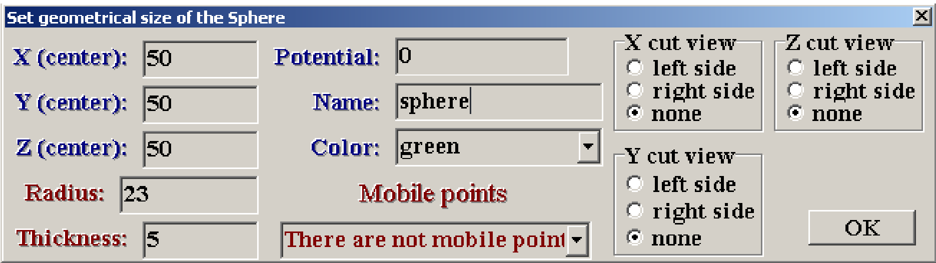
Workbench in SIMION.

 **Object: Sphere.** This object defines sphere with the following geometrical parameters:

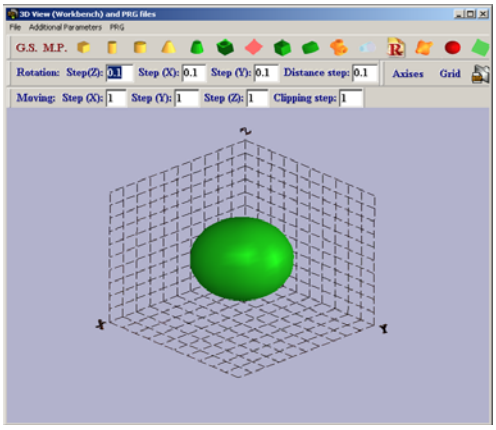


X, Y, Z (center) – those parameters define position of center of sphere. **Radius** – defines the radius of inner surface of sphere. **Thickness** – here user can set up thickness of electrode. The outer surface of sphere is defined as inner radius plus thickness. **Potential**: potential in volts. **Name**: name of object. **Color of object**: here user can specify color of object. It is not used in SIMION. **Mobile point**: here user can set connection between geometrical object and mobile point (if mobile point was added to project).
X, Y, Z cut view – here user can specify half and quarter of sphere. When *None* is selected, a whole sphere is drawn**. Left, right sides** – selects parts of sphere to be cut of (along appropriate axis).

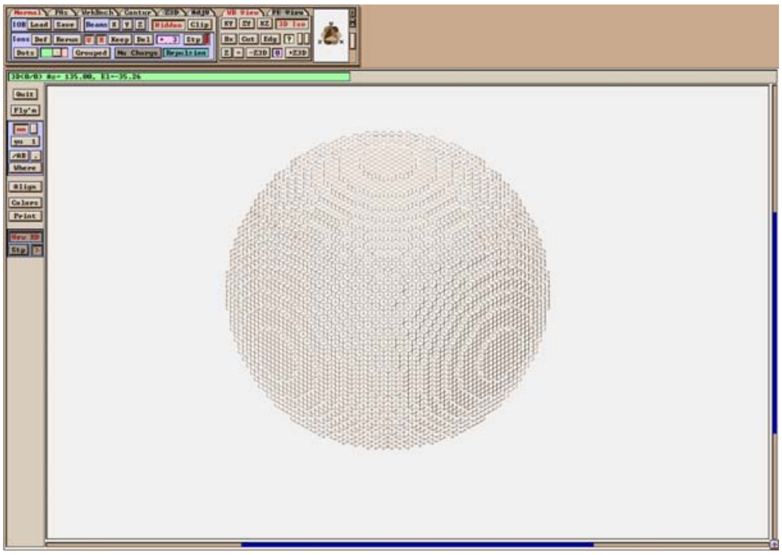
Example 1: full sphere.



Results

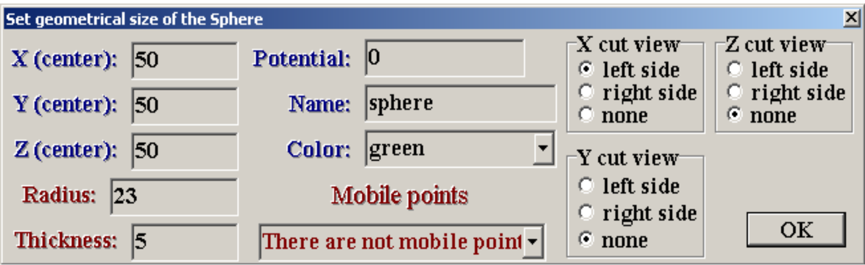


Workbench in *Virtual Device*. Sphere

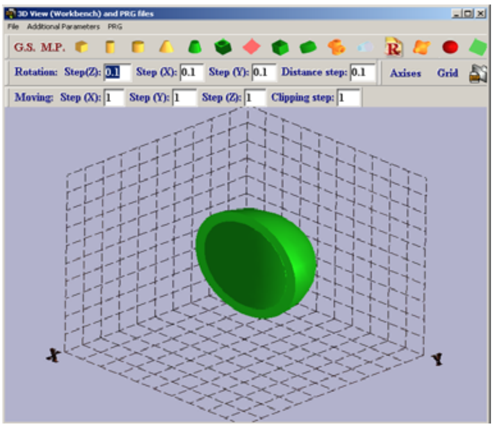


Workbench in SIMION.

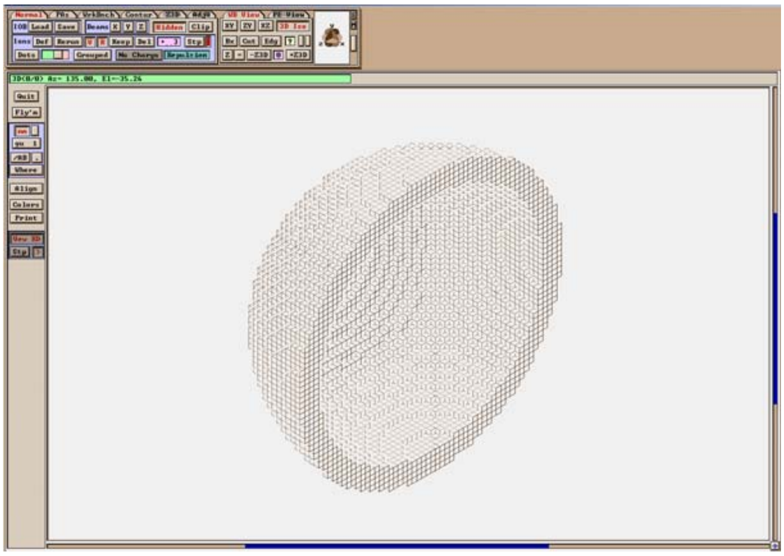
Example 2: Half sphere.



Results:

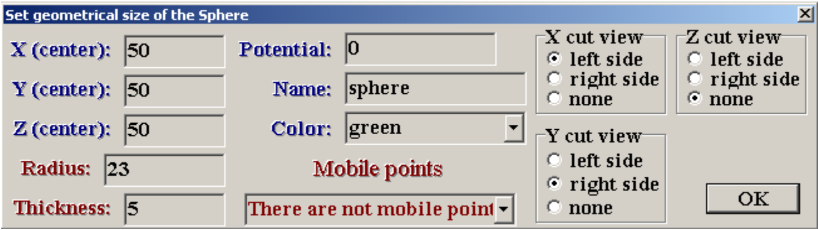


Workbench in *Virtual Device*.
Half Sphere.

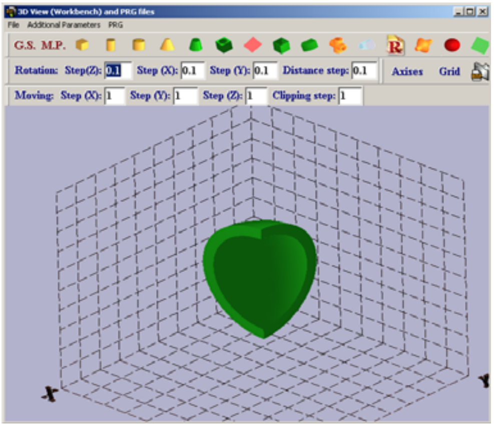


Workbench in SIMION.

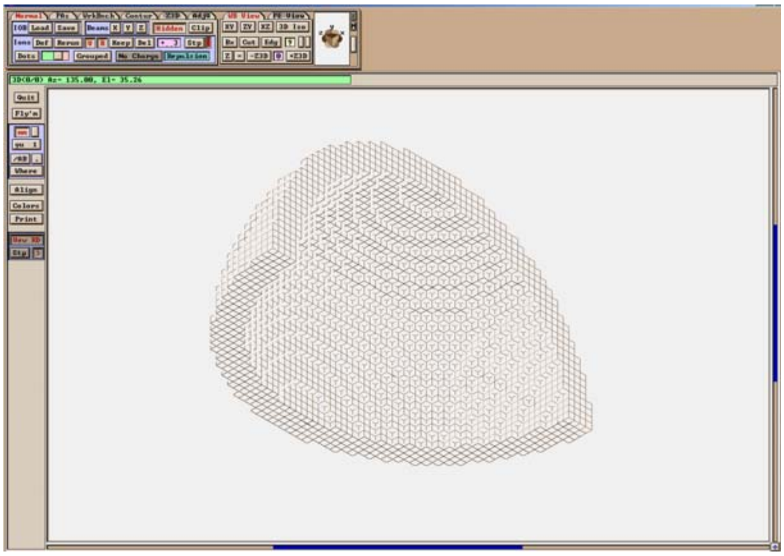
Example 3: Quarter sphere.



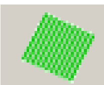
Results:

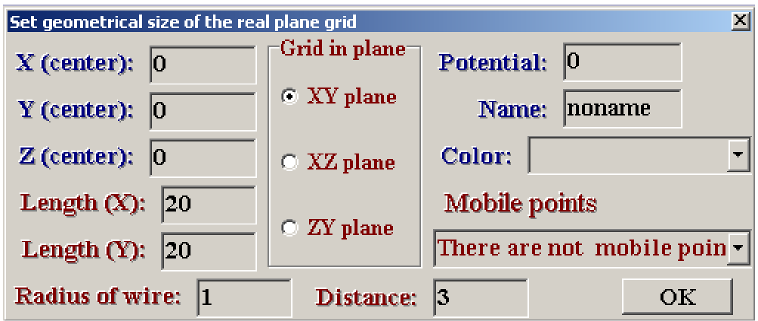


Workbench in *Virtual Device*. Quarter sphere.



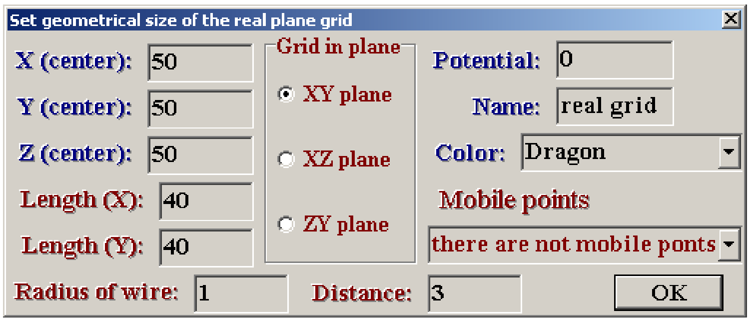
Workbench in SIMION.

 **Object: Real grid.** This object defines parameters of real grid (as a single object). The grid presents itself as a number of wires:

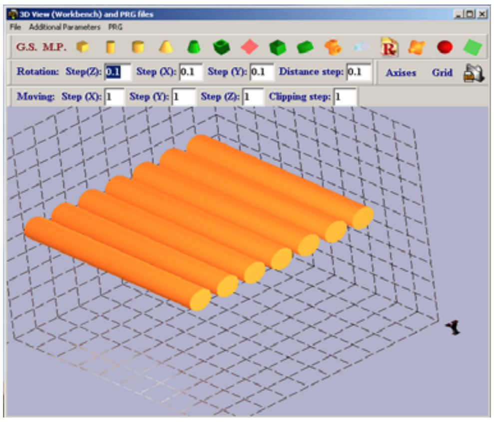


X, Y, Z (center) – position of center of grid. **Length (X, Y, Z)** – defines sizes of grid. **Radius of wire** – defines the radius of wire. **Distance** – defines the distance between wires. **Potential**: potential in volts. **Name**: name of object. **Color of object**: here user can specify color of object. It is not used in SIMION. **Mobile point**: here user can set connection between geometrical object and mobile point (if mobile point was added to project).

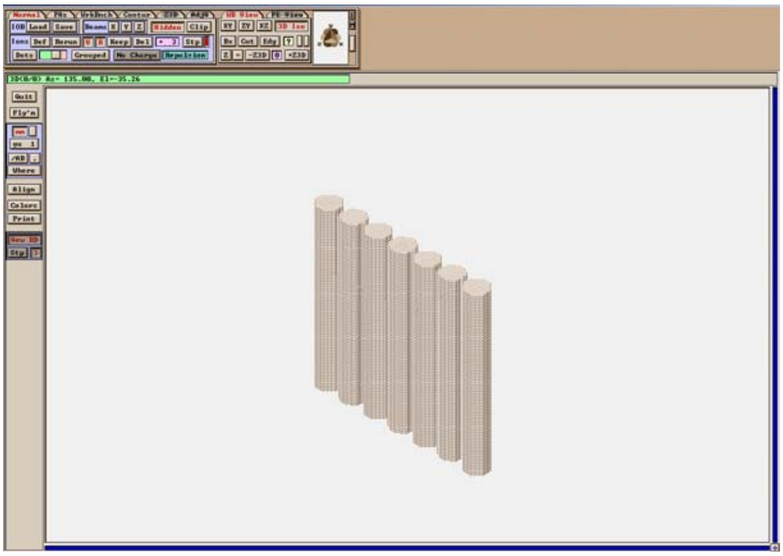
For example:



Results:



Workbench in *Virtual Device*.
Real grid.



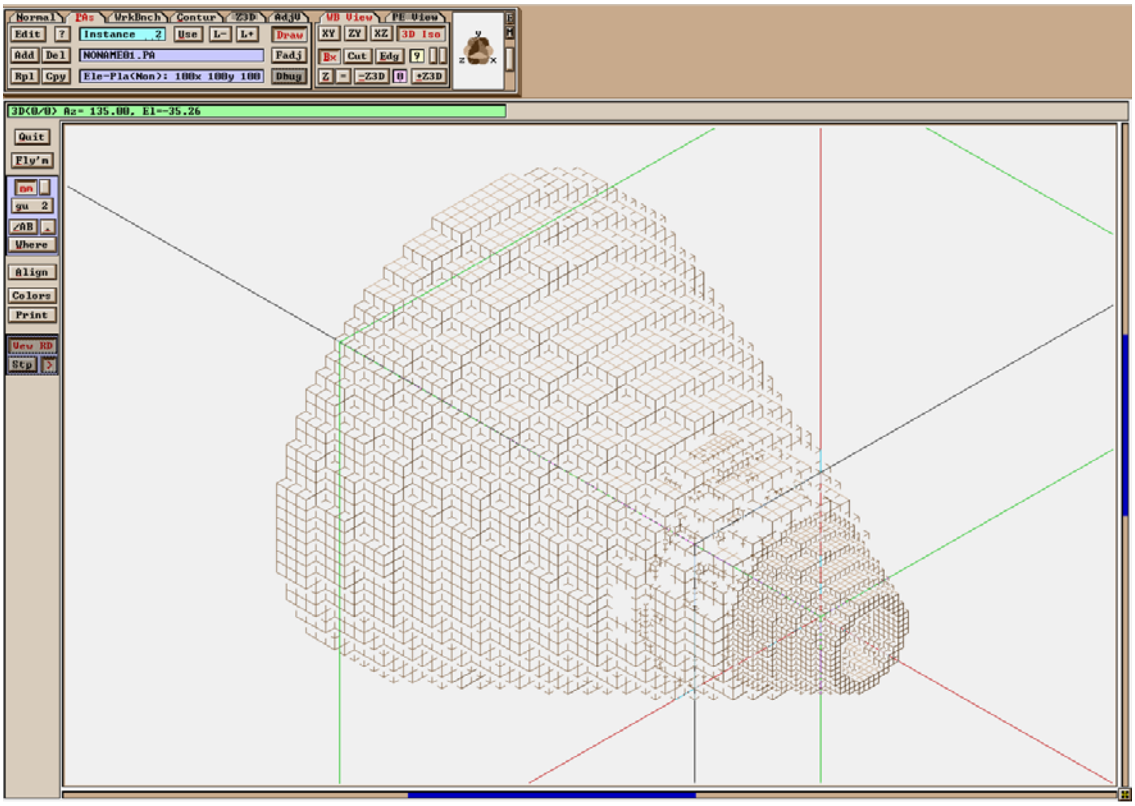
Workbench in SIMION. Real grid.

The advantage of real grid is the following. The real grid can be arranged as a single "fast adjustment electrode"; therefore user can quickly change potential on grid.

Dear User, at this moment there are 14 geometrical objects. If you feel that some important type of electrode is missing you can inform the author (kol-sergei@yandex.ru), and your electrode will be added to *Virtual Device*.

Chapter 1.4. Useful hints.

This chapter covers some useful hints and suggestions.
Hint 1. The quality of 3D geometry in SIMION is strongly dependent on the number of points per millimeter. Therefore it is better to set up as many points as possible. But there is limitation for total number of points (maximum number =50 000 000, in version 7). This limitation can be overcome in two ways. First way: User can set up a 3D project with different number of points per millimeter. It is better to first create a geometry with rough grid, and then - with precise grid within the former. Two instances will overlap each other.
For example: Two cones with different density. The cone with precise grid is placed at tip of the cone with rough grid.



Picture 18. Two cones with different density of points per millimeter.

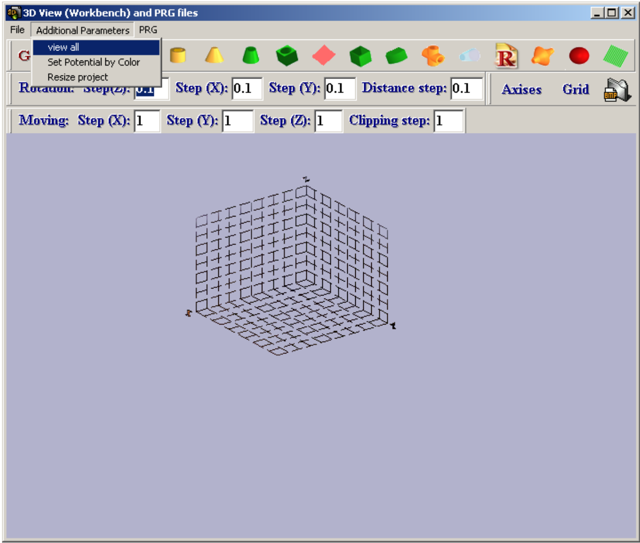
Second way: Sometimes 3D geometry can be splitted up for several instances which do not overlap. Therefore user can set up about 50 000 000 points for each instance. In this case total number of points can be much more then 50 000 000 points.

Hint 2. When setting number of total points it is necessary to take into account how many points are in the most narrow hole or split. If it is supposed that ions have to come through a hole, then field in this region has to be calculated very carefully. Therefore total number of points can be estimated from estimation of hole. For example, we would like to estimate number of points for a system of lenses.
Example A. **Symmetry: Planar, Mirror: none**: The lens of system of lenses is about 50 mm, the hole (diameter) of lenses is 2 mm, and outer diameter of lenses is 30 mm. The total volume is 30*30*50=45000 mm^2. This volume corresponds to X*Y*Z=50 000 000 points. Suppose we like to use 100 points for diameter. It means 2mm = 100 points, therefore 30 mm = 1500, and 50 =

2500. If we base on this estimation then total number of points will be 1500*1500*2500=5625 000 000. Obviously this value is much above 50 000 000. Therefore number of points which can be used in hole is a very important parameter.

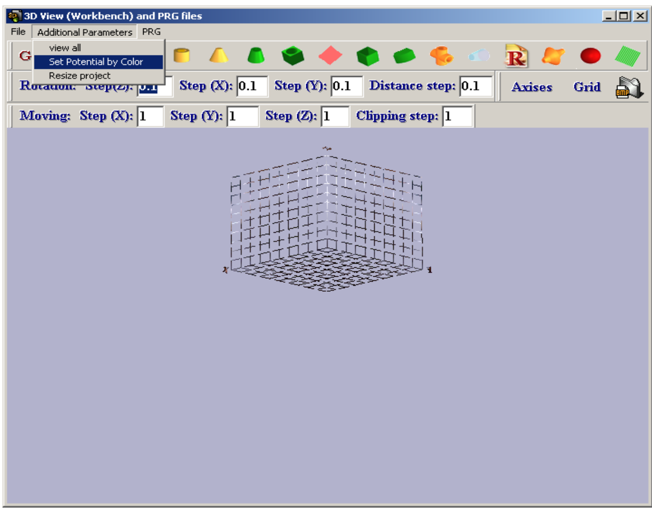
Example B. **Symmetry: Planar, Mirror: YZ**: The total number of points can be reduced if we use mirror type symmetry. For example, let us discuss the same geometrical system of lenses. We have supposed that axis of axial symmetry is X axis. "Geometry" can be made only for region X>0, Y>0, Z>0. The region Y<0, Z<0 is automatically calculated by SIMION if we specify Mirror: YZ. In that case the total number of points will be (1500/2)*(1500/2)*(2500) =1406250000. Compare this results with results in example **A** above**:** we can conclude that by using mirror we reduce total number by four times.

Hint 3. If you have unintentiously deleted 'working window' and 'view of project window', You can restore those windows by choosing 'Additional Parameters – view all'. See picture 19.



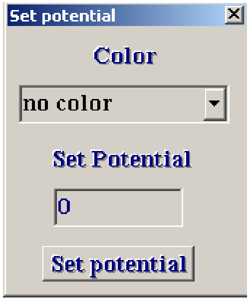
Picture 19.

Hint 4. User can automatically set up the same potential for electrodes which have the same color. For example you have 10 electrodes with some color, say – red. Then choose 'Additional Parameters – Set Potentials by color'. See picture 20.



Picture 20.

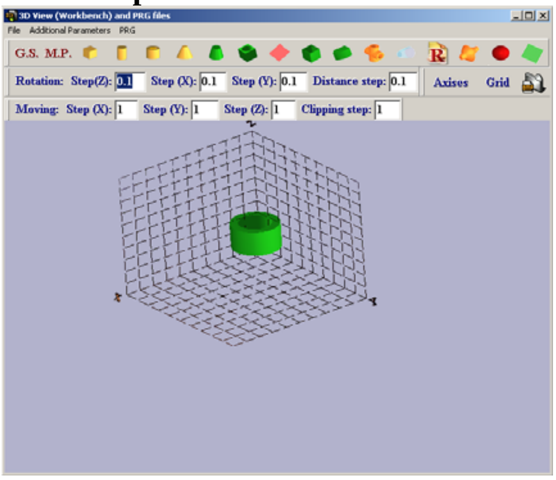
Then you will see the following window (picture 21). Specify color (for example, red) and value of potential. By pressing ‘Set potential’ user automatically sets up value for all electrodes with red color.



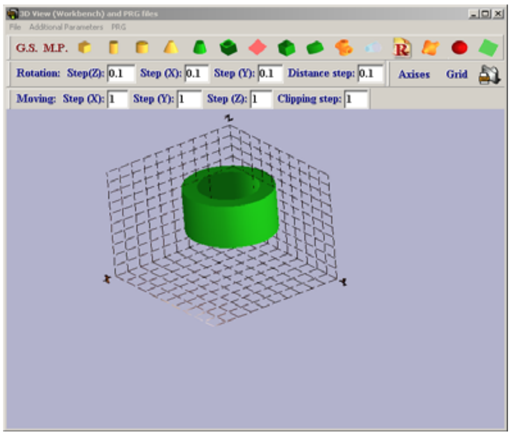
Picture 21.

Hint 5. Resize project. It is intuitively clear that it is more convenient to create a 3D project in millimeters. But SIMION deals with points. To make users’ life easier option **Resize project** was developed. User can change the size of geometrical object by setting coefficient of multiplication. Geometrical parameters (position of center/bottom, length and so on) will be changed by multiplication by this coefficient.

For example:



Results:



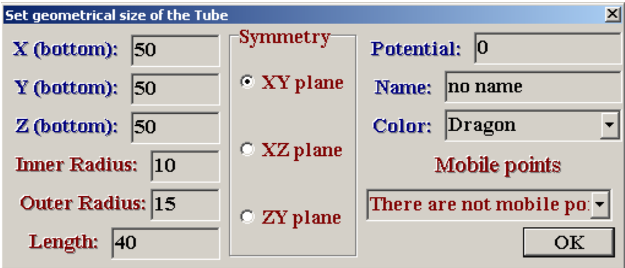
Now user can transform 3D geometry to geometry file, save as PA arrays and load PA file with coefficient 0.5. User will get project in SIMION in millimeters, however, with large number of points per millimeter.

Hint 6. How are millimeters in *Virtual Device* related to points in SIMION?

The relation between millimeters in *Virtual Device* and points/millimeters in SIMION is quite simple.

For example: We would like to create a tube in millimeters in *Virtual Device*, and we like to have this tube in SIMION with the millimeters/points ratio = 0.1. It means 1 millimeter is presented by 10 points (cells) in SIMION. We can do it in several steps.

A. Let us make the following tube:



The length is 40 mm (we supposed that is in millimeters). We would like to have this distance also in SIMION.

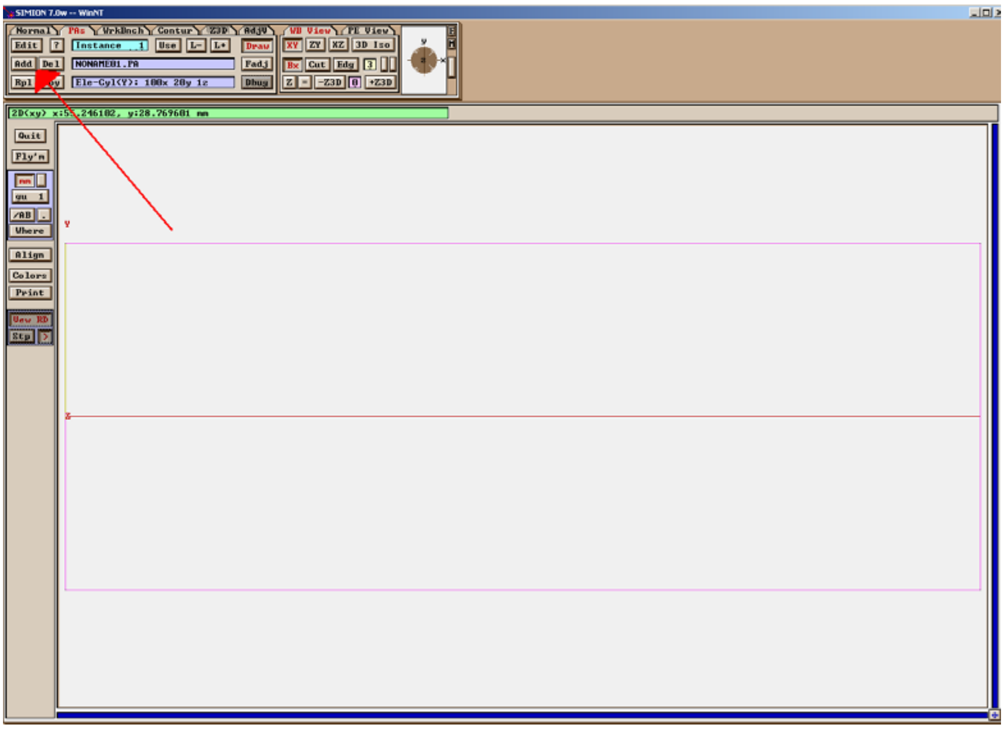
B. Then we have to enlarge 3D project by resizing (**coefficient of resizing 10/1**) or we can set up parameters of tube by hand.

C. After resizing it is necessary transform our 3D project to a GEM file.

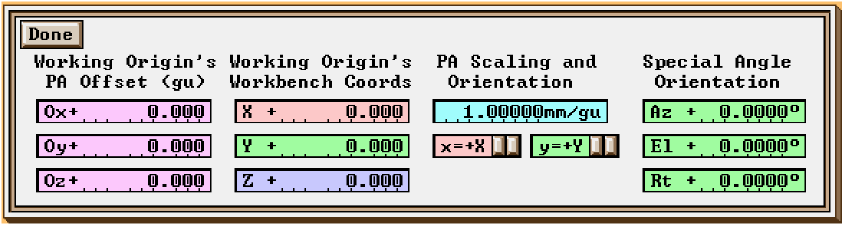
D. Create new project in SIMION by loading GEM file and save result as tube.PA#.

I. Empty memory in SIMION and create a new project in SIMION.

F. Add tube.PA# to the new project. You can do by pressing button '**Add'** in '**Pas'.** See the following picture.



G. To add the tube to the new project user has to fill in following data:



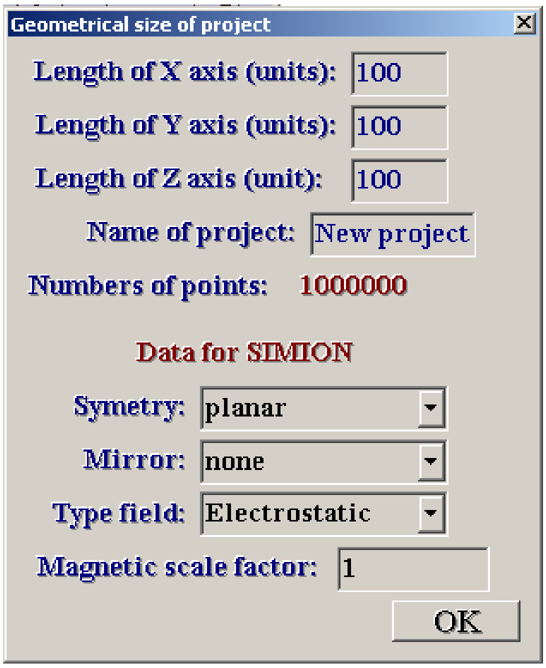
Here user has to set up coefficient mm/gu as 0.1. This coefficient is exactly reverse to coefficient of resizing. Now you have the tube with length 40 mm and the scale is 10 points per one millimeter.

Chapter 1.5. Symmetry and mirror of 3D project.

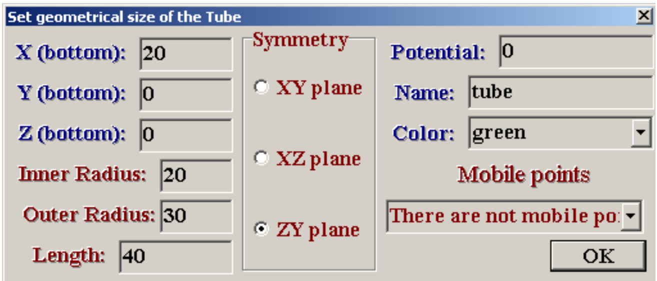
Here we shall discuss the problem of symmetry. As it was shown above, there is a possibility to set up a type of symmetry and of mirror in SIMION. Using of those parameters allows reducing total number of points per millimeter and reducing the number of electrodes actually created in *Virtual Device*.

Example. Let us create a simple electrode – short tube (some kind of a lens).

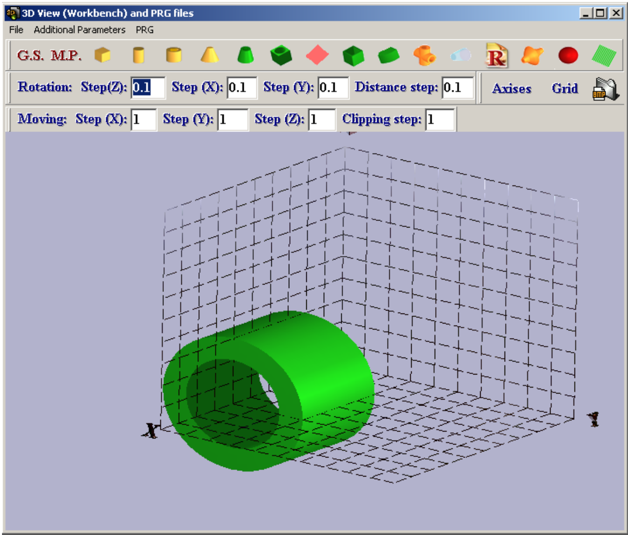
We will set up the following parameters of the project and the tube (21, 22).



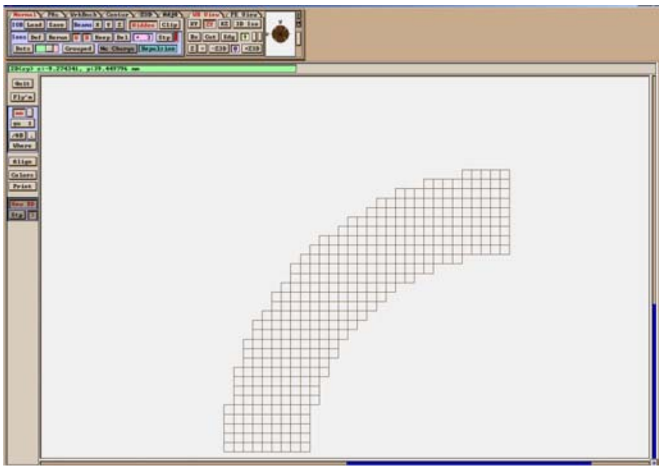
Picture 21.



Picture 22.

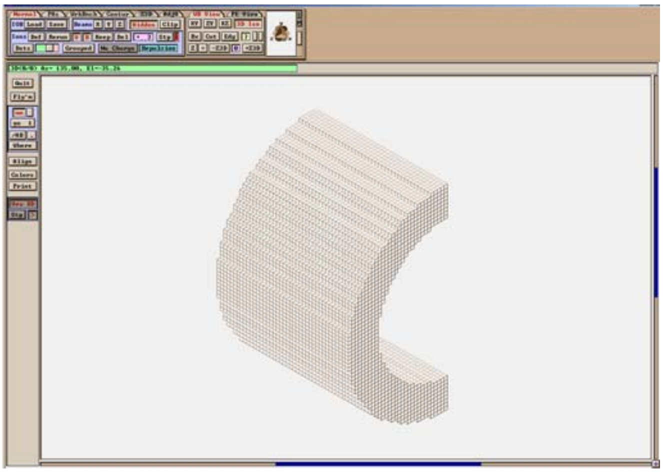


Picture 23. *Virtual Device* workbench

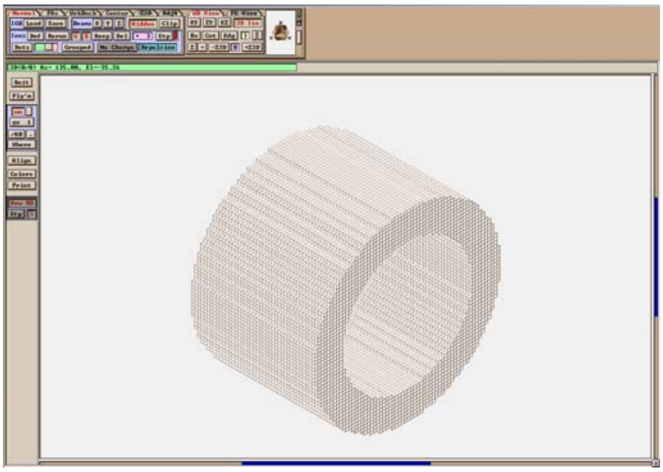


Picture 24. SIMION workbench,
Mirror: none

The part of the tube stretching beyond the boundary of the grid is cut off by SIMION. Now we will change only the mirror parameter. Let us set it up the following value: **Mirror: Y.** it does not influence the picture in *Virtual Device* workbench, but the picture in SIMION workbench will be different.

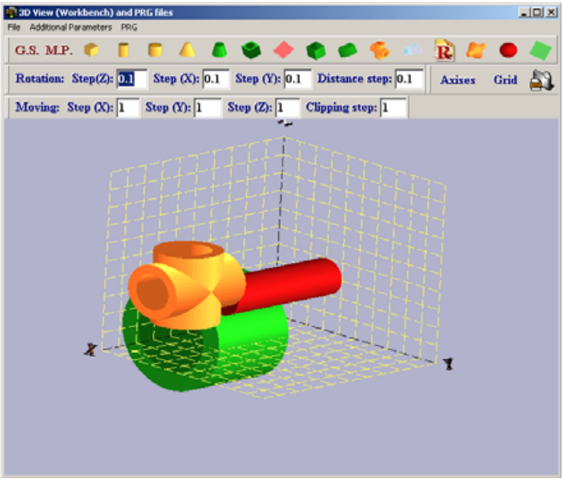


Picture 24. SIMION workbench,
Mirror: Y.

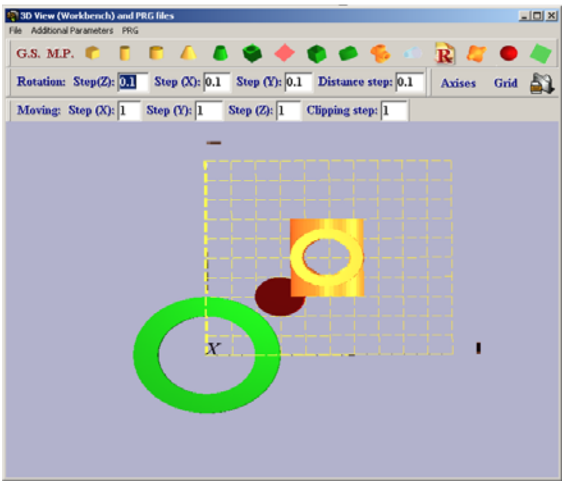


Picture 25. SIMION workbench,
Mirror: YZ.

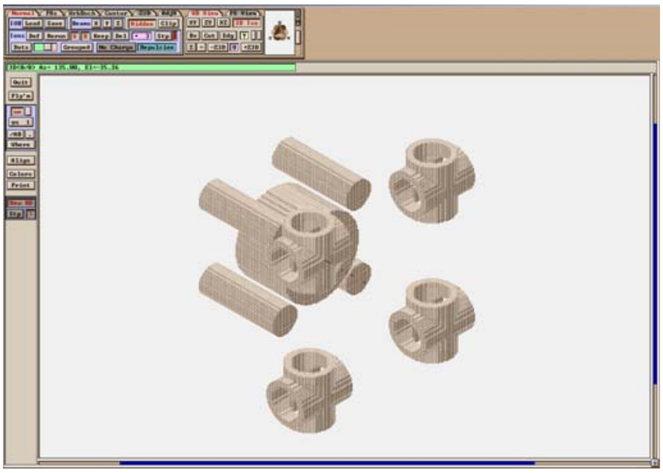
Therefore, user can create a complicated geometry just by changing the mirror parameter. Here is another example of such geometry. Size of project: X: 200, Y: 100, Z: 100.



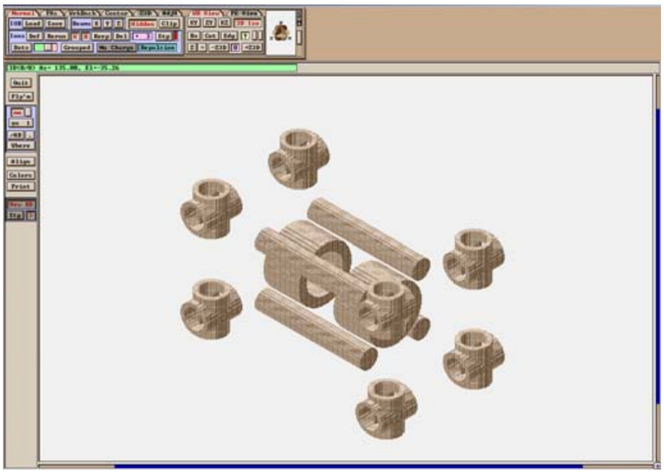
Picture 26. *Virtual Device* workbench



Picture 27. *Virtual Device* workbench



Picture 28. SIMION workbench,
Mirror: YZ.

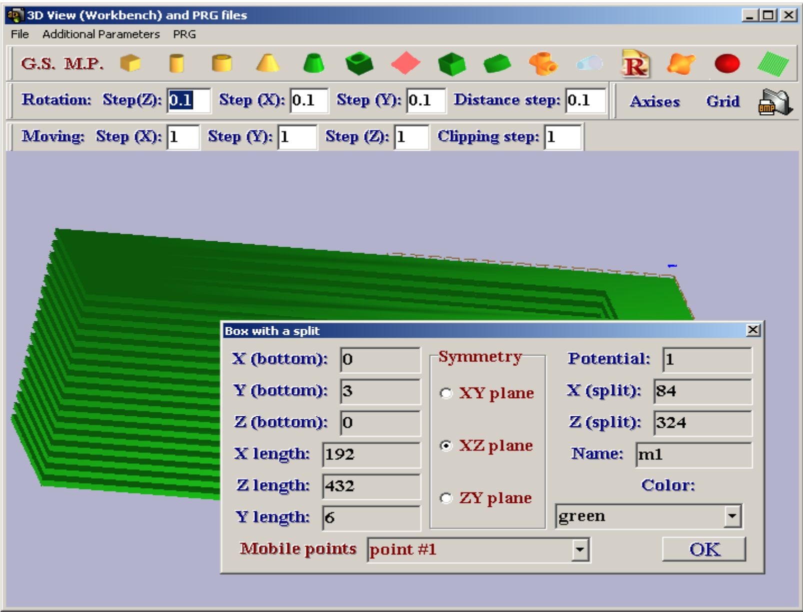


Picture 29. SIMION workbench,
Mirror: XYZ.

Now you can see how the mirror parameter influences 3D geometry in SIMION.

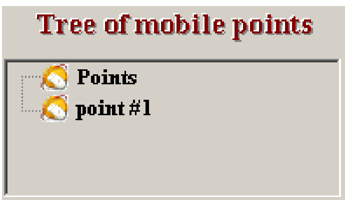
Chapter 1.6. Mobile points.

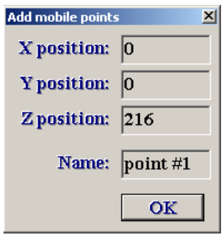
The main purpose of mobile points is simplification of working with very complicated projects. How mobile points can make life easier? Actually position of geometrical object (center or bottom) is defined relatively to zero of system of coordinates. If user adds mobile point to project then user can specify position of geometrical object according to that mobile point, while position of mobile point is automatically settled up according to zero of system of coordinates.
Example: load project '\tof_parts\mirrors.vds'. All electrodes in this project are related to mobile points 'point #1'. See picture 30.



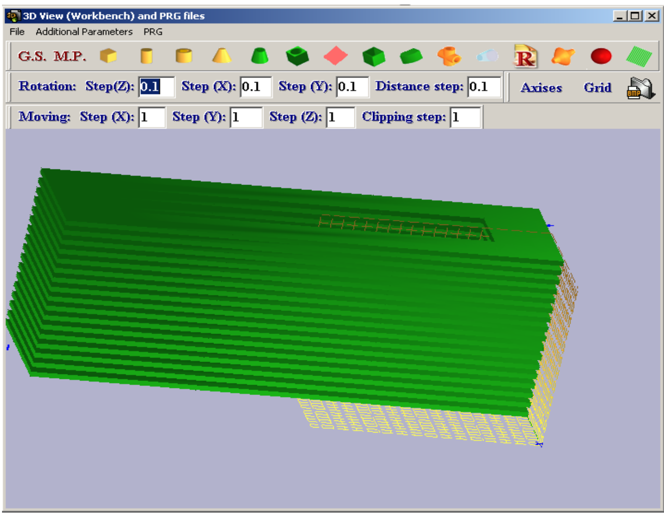
Picture 30.

Now choose pictogram 'points#1' and change X position to zero.





You will see that all electrodes will be automatically shifted. Therefore mobile point allows changing positions of all electrodes associated with mobile point at the same time. If you do not use mobile points then you have to change positions of all electrodes (step by step) by hand.

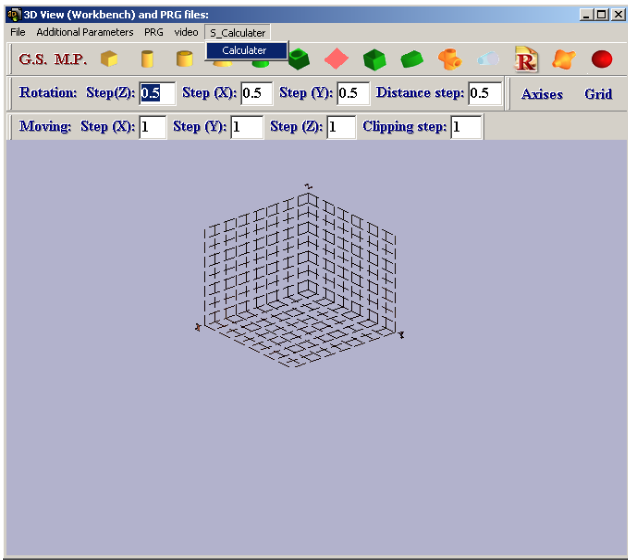


Therefore you can evaluate how useful mobile points are.

Chapter 1.7. Scientific calculator.

Introduction: The main purpose of scientific calculator is to increase speed of development of electrodes with axial symmetry. Some electrodes (profiles) can be described by analytical expressions. *Virtual Device* provides possibility of fast building of such electrodes. The creation of electrodes is based on translation of arithmetic expression (which is just a text) to code, which can be executed by computer. The are many possibilities of translation, but we base on so-called 'reverse Polish writing'. We shall not go into details of how this algorithm works. The details of algorithm used in *Virtual Device* are published in [5].

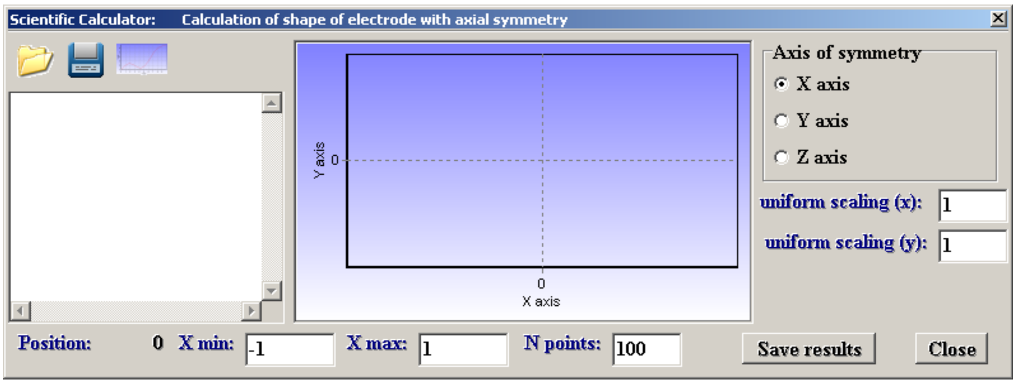
To run calculator pick up 'S_Calculator/Calculator' as is shown on the following picture.

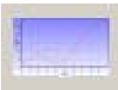


You will get a window of calculator. It is not just an ordinary calculator, here user can type complicated formulas and quickly create graphs. The results of calculation can be stored in text files and hereinafter used in creation of electrodes with axial symmetry (axial objects). Actually shape of graphs is used as profile of an electrode with axial symmetry.

The calculator window has the following parts: 1. on the left there is white space where user can type formulas (examples of formulas are in the directory ***Virtual***

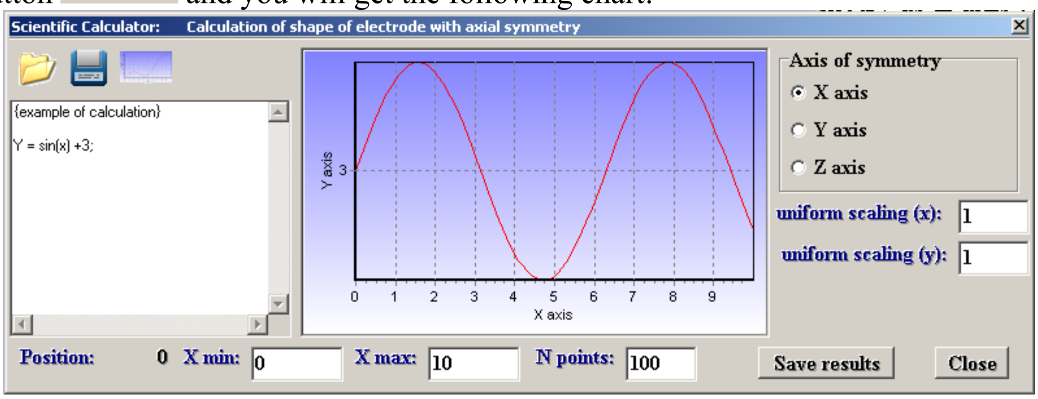
***Device*\examples_scientific_calculator**\). 2. A blue-white chart in the middle is chart where user can see results of calculation (as a graph), 3. to the right there is option to save results in text format. This format is one used for creation of objects with axial symmetry.



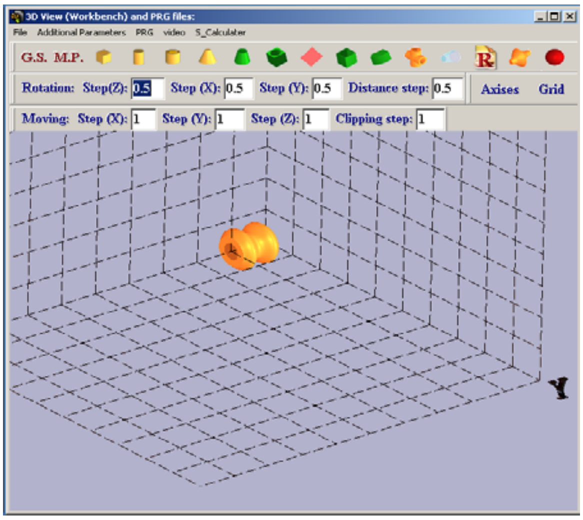
 - button to load formulas from text file,  - button to save formulas in text file.  - button to run calculation. User has to type function in the following format: y=f(x), therefore **X min** and **X max** define the range of x. **N points** – defines how many points will be used in calculation in the given range. Button '**Close**' closes the window. **Position** shows the position of cursor in area of formulas. That can be very useful when user has made mistakes: *Scientific calculator* reminds about error and names the position where error occurs.

It is necessary to note that user has to use function y=f(x) because this part of code (interpreter of code) was imported to *Virtual Device*. According to axial symmetry it means X axis is axis of symmetry, Y (Z) - axes of radius. To have access to other axes of symmetry we just add possibility of changing axis of symmetry when data will be written to file. For example: user types a function y=f(x) but chooses Z axis as axis of symmetry (to the right of calculator window). Therefore X axis will be substituted by Z axis. The example of that is shown below. The best way to learn how to use *Scientific calculator* is just to repeat several examples.

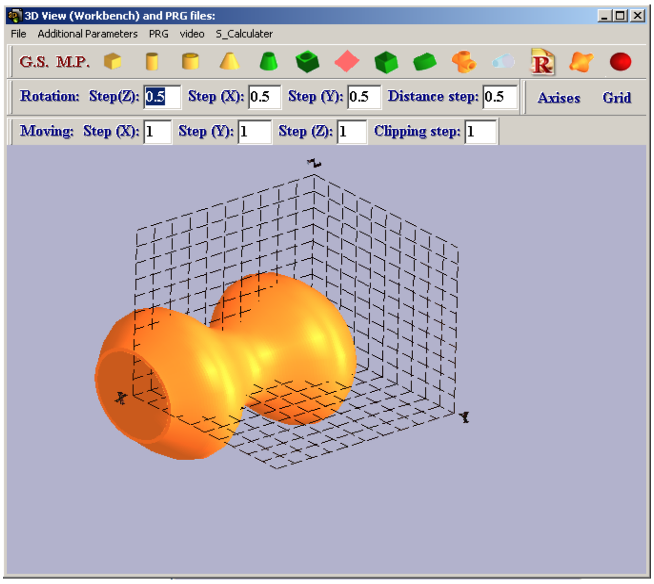
Example 1: type the following expression: Y=sin(x) +3, and set up range of X [0:10]. Press button  and you will get the following chart:

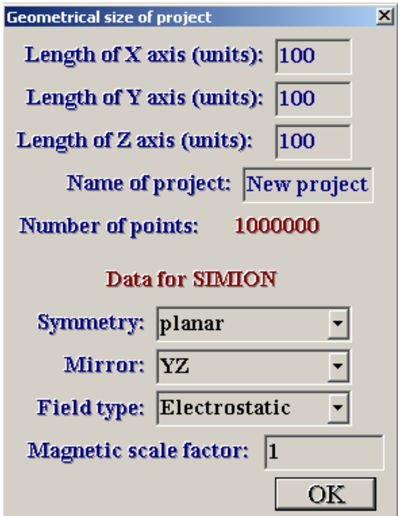


Save data: press button 'Save results' and specify file name (for example: my_shape.txt). Now create a new axial object (see any axial object) and load data from fresh baked file (my_shape.txt); you will get axial object with the following surface:

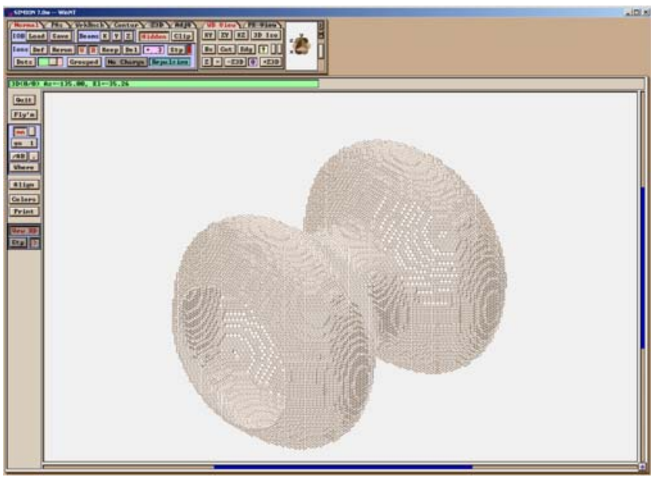


This electrode is quite small because sin(x) is range [-1; 1]. To increase all data proportionally just change '**uniform scaling (x)** and **(y)**' to 10 and save results (you do not need to recalculate). Load the new data into your axial object and you will get the following results (left picture):

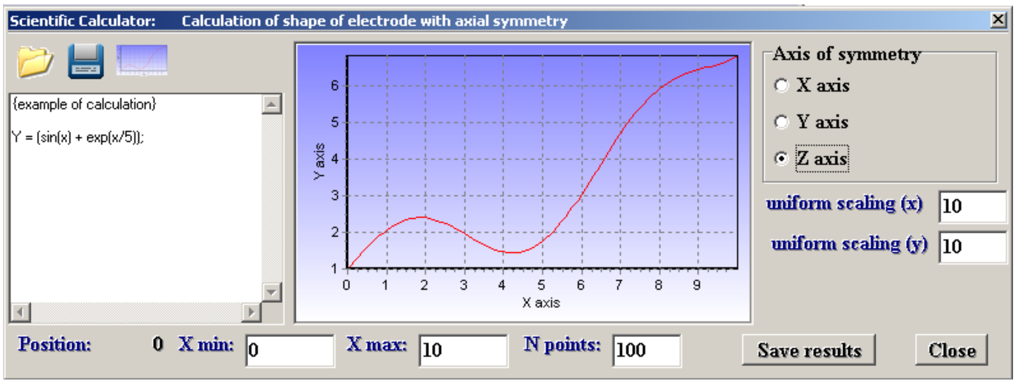




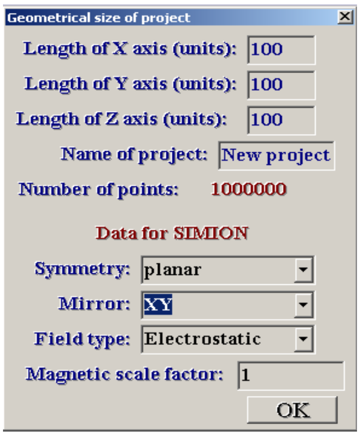
As you can see, the "uniform scaling" coefficients are just scaling coefficients. Now specify parameters for SIMION project as is shown on the right picture above, then create gem file and load this file to SIMION. As a result you will get the following picture:



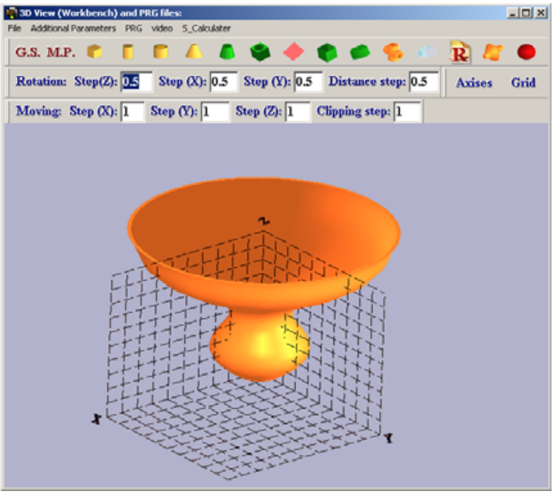
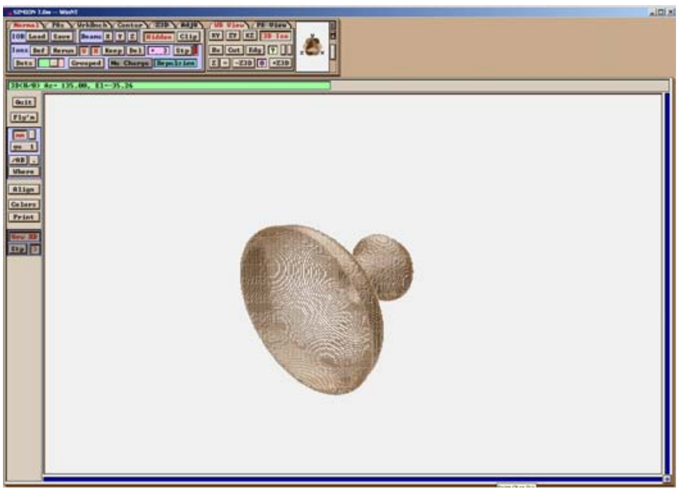
Example 2: Now load formulas from file my_example.txt. Specify X range as [0; 10], specify axis of symmetry as Z axis, uniform scaling coefficients as 10. You will get the following results:



Now save results in some txt file, launch a new axial object and load data from your txt file. Remember that you have to specify axial symmetry for axial object as Z axis, because you have saved results of calculation in *Scientific calculator* with Z axis of symmetry. Do not forget to set up correct parameters for SIMION project as is shown on the following picture:



If you do all correctly, you will get the following picture in *Virtual Device* and it is equivalent in SIMION:

To create such electrodes I spend about several minutes. How many hours do you spend to do the same?

There are more examples in directory *Virtual Device*\examples_scientific_calculater\.

Syntax: Operation and functions.
At the moment user can use the following functions and operations:

notation	Operation/functions
abs(x)	absolute value
arccos(x)	Arccosines (results in radians)
arctg(x)	Arctangents (results in radians)
cos(x)	Cosine (argument in radians)
exp(x)	exponent
ln(x)	natural logarithm
sin(x)	Sine (argument in radians)
sqrt(x)	square root
tg(x)	Tangents (argument in radians)
(	Opening bracket
)	Closing bracket
+	addition
-	subtraction
*	multiplication
/	division
=	assignation

Constants: Pi, E (base of natural logarithm).

Formal rules:

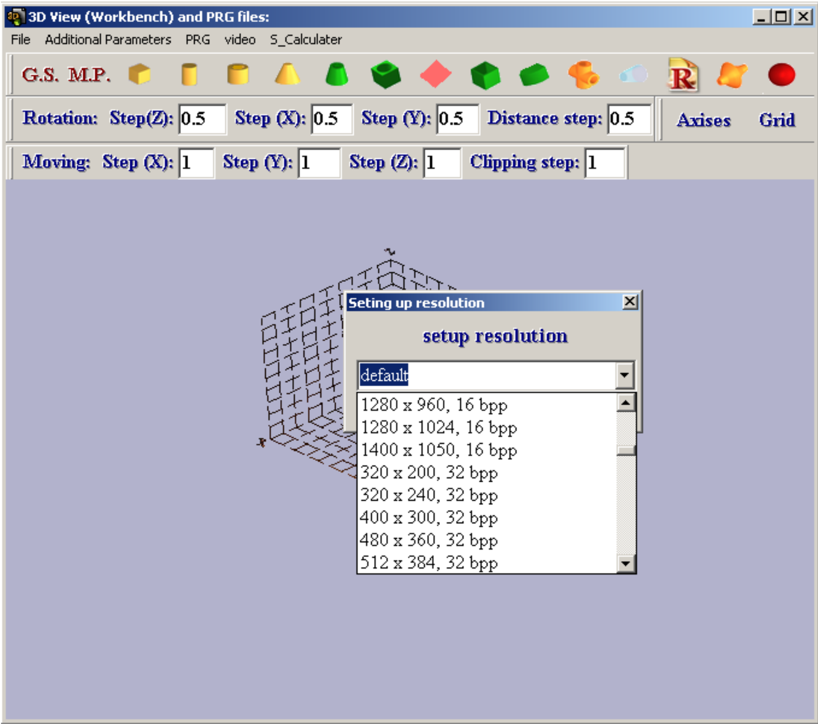
1. There is no difference between upper and lower cases, for example: Sin, sin, SIN are equivalent.
2. The description of formulas can occupy as many lines as is necessary. User can continue description to a new line at any position. Numbers have to be unbroken.
3. User can use identifier. For example Pi is one of identifiers. It is possible to set up user's identifiers.
4. Comments have to be placed in brackets {} at any position of description but not inside a number or inside an identifier.
5. If description of very complicated formulas consists of many expressions, then those expressions have to end by semicolon **;** (see examples in the catalogue described above).
6. User can use variables. If variable is not defined in description, then it is supposed that this variable is zero.
7. Numerical constant can be written in the following formats: 5,5.1, 4E10, -4.2E-3. Please take attention that separator for numerical number is a dot.
8. Usual mathematical restrictions are present. For example user can not calculate sqrt(-1).

If user makes a syntax error in description, translator shows the list of errors and defines where those errors occur (position).

Zooming chart: to zoom in a chart, locate mouse on some part of the chart, press left mouse button, and move mouse to the left and down. It will indicate a rectangle, then release left mouse button. The area inside the rectangle will be zoomed in. To zoom out, locate mouse on some part of the chart, press left mouse button, and move mouse to the right and up.

Chapter 1.8. Screen resolution.

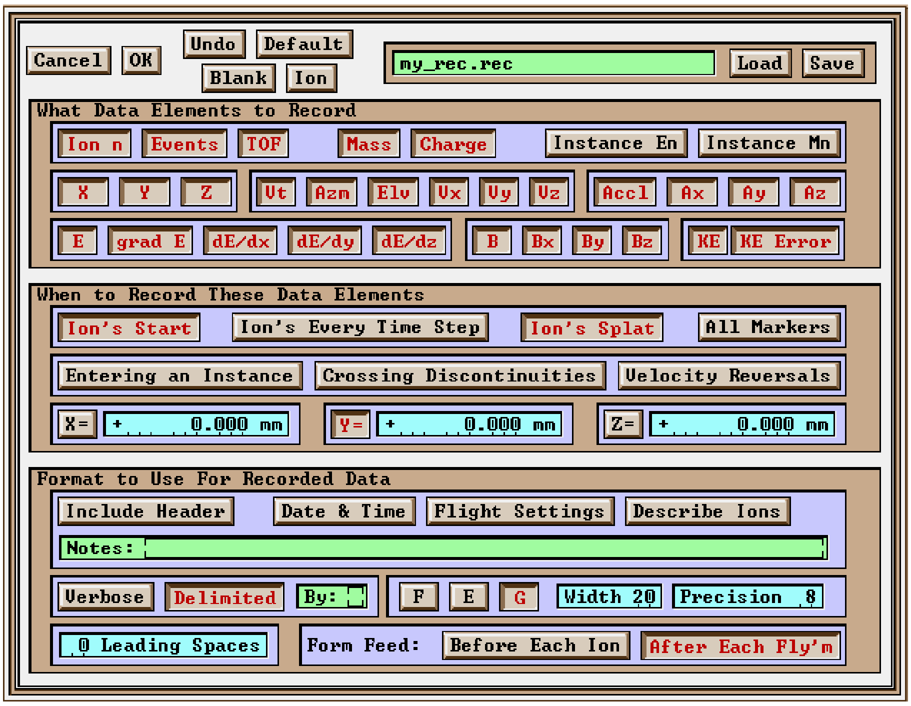
There is possibility to tune resolution of screen inside *Virtual Device*. To do it, just pick up **video** from main menu. '**Setting resolution'** inside it allows changing resolution. '**Restore default mode'** restores default resolution of your computer. When user picks up **'Setting resolution'**, *Virtual Device* checks all possible variations of videos of your computer and creates a list of modes. So user can choose an appropriate mode from the list. The length of list varies from computer to computer. An example of such list is presented on the following picture:



Chapter 2. SIMION data Analysis.

Chapter 2.1. Load and Save data.

SIMION allows saving results of simulation in txt file. Therefore there is possibility to develop users' software for analysis. *Virtual Device* gives such possibility to user. It allows unfolding SIMION data into a table, if SIMION data are saved in txt file in the following format (example of this format is in file my_rec.rec):



Picture 31.

Important note: User has to remember that only two events are allowed. The first event is event of ion birth. The second event is event of ion death. For example: **Ion's Start** and **Ion's Splat**. It is not important what kind of death is. User can change **Width** and **Precision**.

Buttons:
To load data from file in SIMION format (see format above) it is necessary to press button '**Open SIMION data**'.
The each row in table (picture 32) presents one ion with two events. The first event (birth) takes columns from 1 to 28, and second event (death) takes columns from 29 to 57.

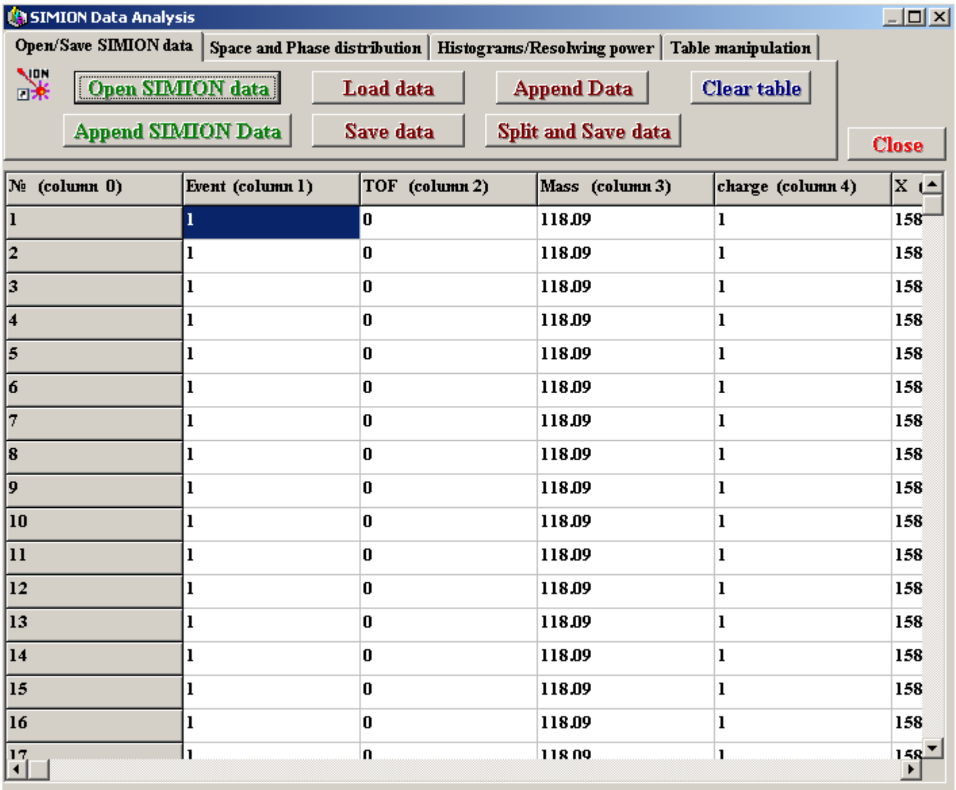
Button '**Append SIMION data**' – this button allows to add data from SIMION file to existing data.

Button '**Save data**' – this option allows to save data from table to txt file.

Button '**Load data**' – this option allows loading data from txt file to table.

Button '**Append data**' – this option allows adding data from txt file to data in table.

Button '**Split and Save data**' – this option allows splitting data from table and save splits into different txt files.
Button '**Clear table**' – this button clear table.



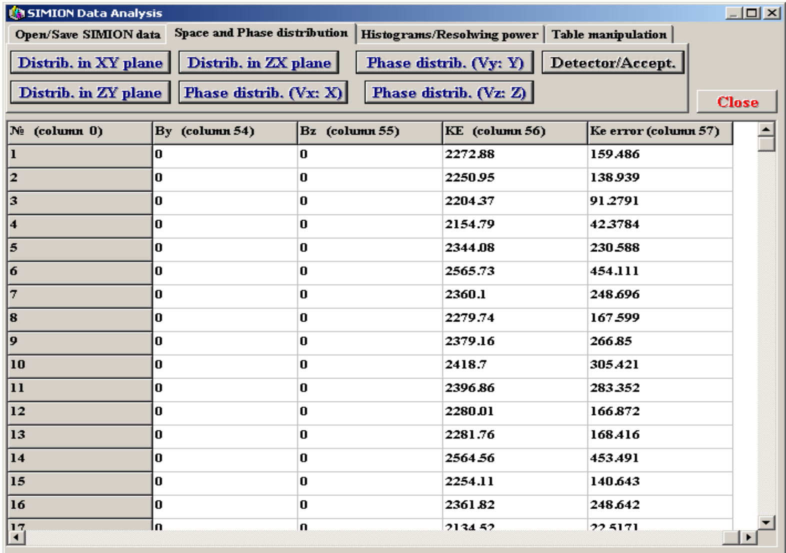
Picture 32. Open/Save data.

Chapter 2.2..

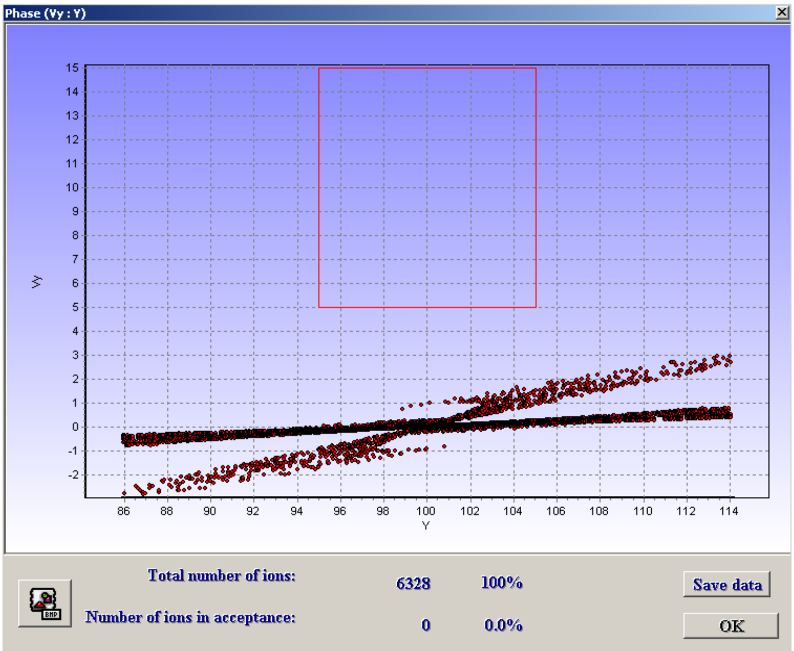
This part of *Virtual Device* gives possibility quickly makes plots of ion distribution in different planes and make plots of phase distribution of ions along axes (see picture 33).
Button '**Distrib. In XY plane**' – this option allows making plot of ions in XY plane.
Button '**Distrib. In ZY plane**' – this option allows making plot of ions in ZY plane.
Button '**Distrib. In ZX plane**' – this option allows making plot of ions in ZX plane.
Button '**Phase distrib. (Vx: X)**' – this option allows making plot of ions phases along X axis.
Button '**Phase distrib. (Vy: Y)**' – this option allows to make plot of ions phases along Y axis.
Button '**Phase distrib. (Vz: Z)**' – this option allows to make plot of ions phases along Z axis.

Example of ions phase distribution is presented on picture 34. The red rectangle on picture 34 is window, parameters of which are defined in window of detector/acceptance (see picture 35). User can easily evaluate how many ions inside of detector/acceptance. *Virtual Device* automatically calculates total number of ions in distribution and how many ions inside of detector/acceptance.

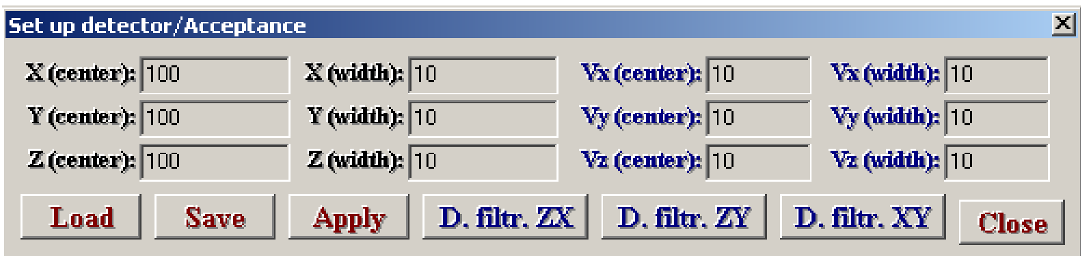
Detector/ Accept: This option allows set up rectangle which can be used for estimation number of ions in different planes and number of ions in phase diagrams (see picture 35). User has possibility to save data of detector/acceptance to 'txt' file.
Buttons **D. filtr. ZX**, **D. filtr. ZY, D. filtr. XY** allows deleting all ions which are out side of rectangle. It is filtration inside of window.



Picture 33. Menu for Space and Phase Distribution.



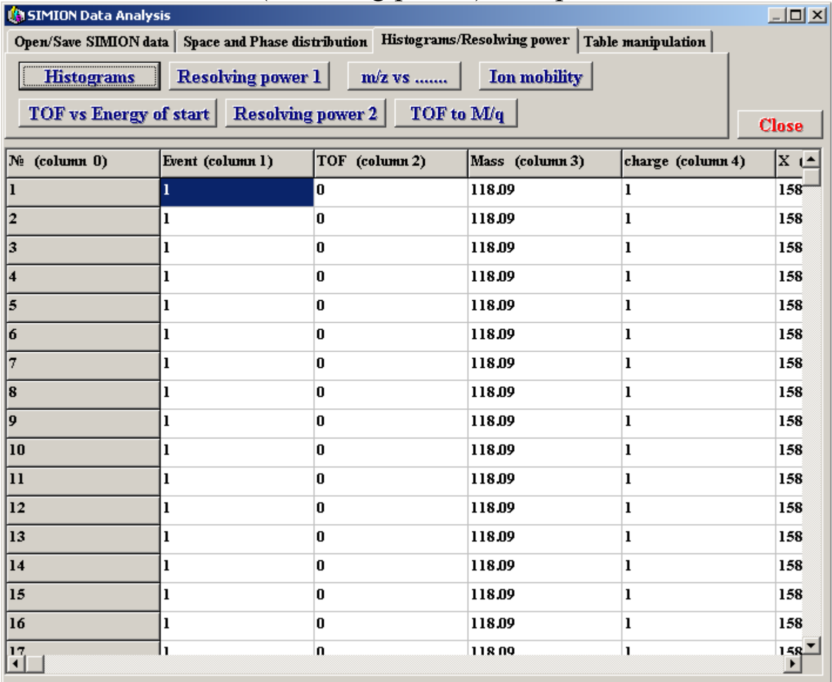
Picture 34. Example Phase Distribution.



Picture 35. Size parameters of Detector/Acceptance.

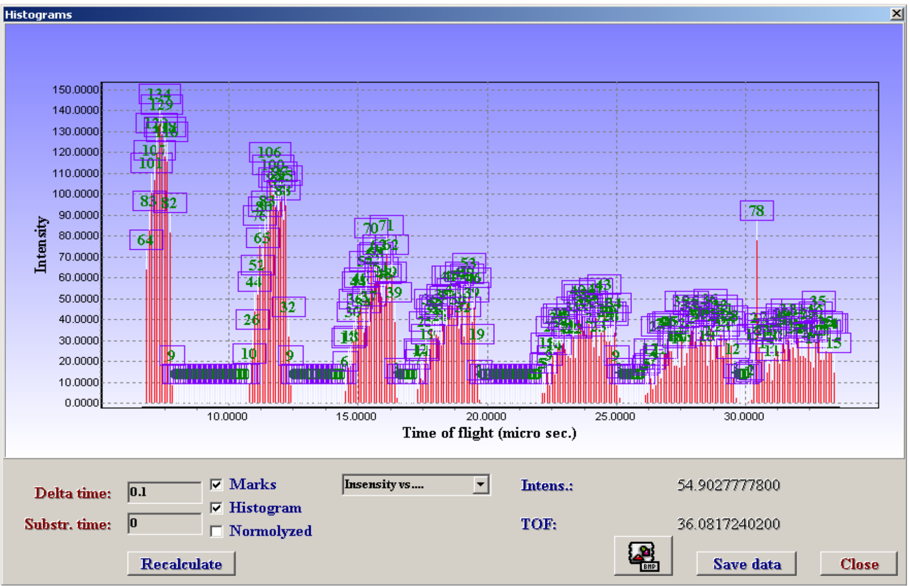
Chapter 2.3. Histograms/Resolving power.

This part of *Virtual Device* allows quickly create Histograms of ion exit parameters (event o death) and calculate resolution (resolving power). See picture 36.



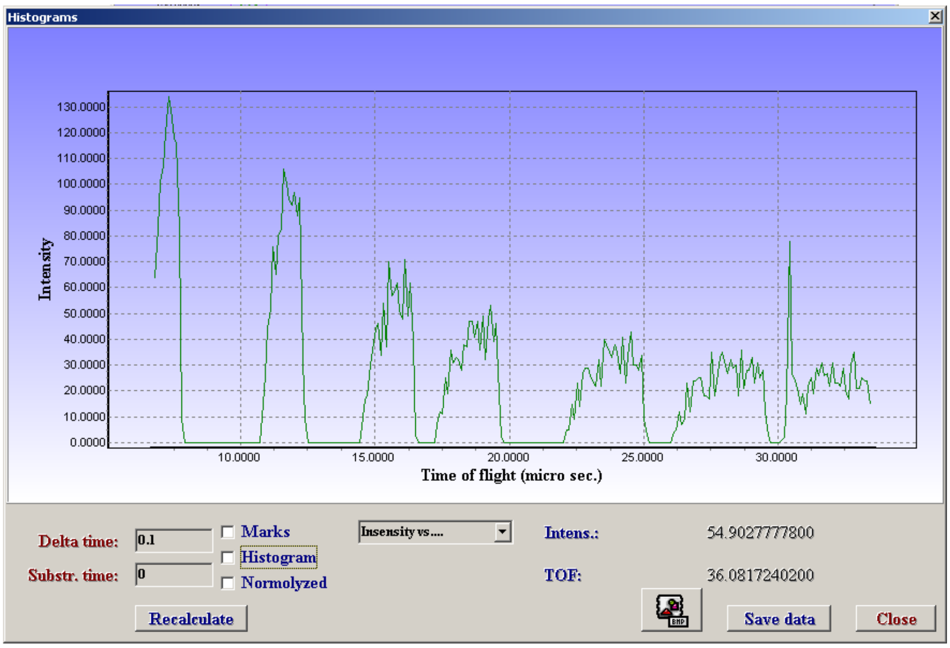
Picture 36. Histogram/Resolving power window.

Histograms: This button run chart where user can build histogram of exit parameter (see picture 37).



Picture 37. Creation of histograms.

User can set up exit parameter from dropdown list '**intensity vs….**'. As definition (as it show on picture 36) *Virtual Device* calculate intensity of time of flight. **Marks** – this allows switching on/off labels (grin numbers in blue rectangle). **Histogram** allows to switch between histograms (red lines) and envelope curve of histogram (grin lines) (see picture 38).



Picture 38. Creation of envelop of histograms.

Button **Normolyzed** makes chart where intensity is in range [0; 1].
Button **Recalculate**: to refresh drawing after choosing new exit parameter just press this button.
Button '**Save data'** allows saving chart as txt file with two columns (intensity and exit parameter).

Button  allows save picture in file (bmp format).

Button: **Substr.** (time, energy, X, Y, Z and so on) allows subtract delta value from appropriate exit parameters. This feature can be useful for example for estimation of time of flight.

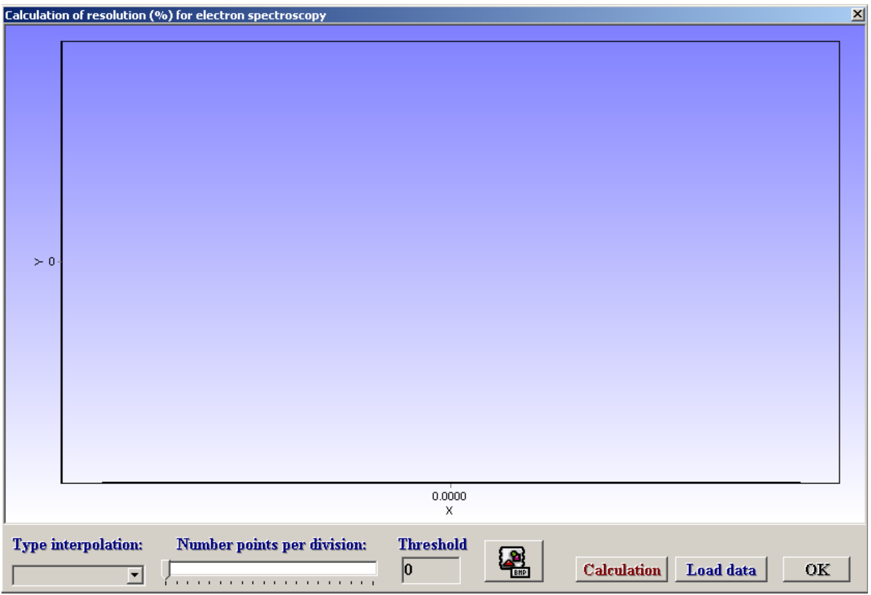
Button: **Delta** (time, energy, X, Y, Z and so on). This value specifies quality of histogram. With this big value you will get a very rough histogram.

When user moves mouse over window, *Virtual Device* calculate position of mouse and show the intensity and value of exit parameter at mouse position.

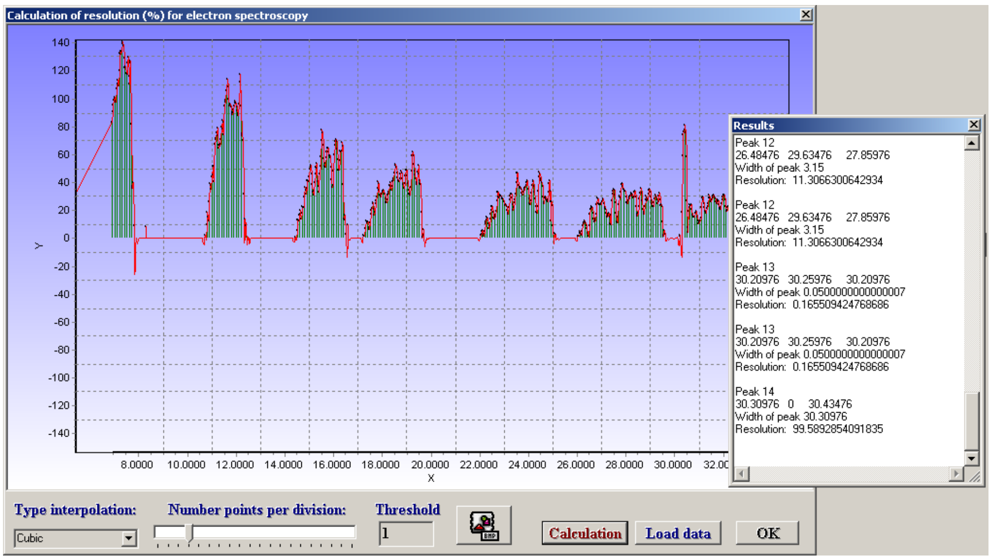
Button '**Resolution (100%)**' this button allows to calculate resolution of peaks in specter.
The way of calculation of resolution is described in chapter '**Main parameters of mass-spectrometers, Resolution in electron spectroscopy'**. The calculation of resolution can be done in following steps (see picture 39). **1.** Load data from file. To do it press button '**Load data**'. This data can be created by saving data in histogram window. **2.** Set up type of interpolation. User can set up **linear** or **cubic** interpolation. **3.** In case of small number of points per peak user can add additional points between points from file. To add points user has to set up number points in '**number points per division**'. **4.** Specify level of **threshold**. This parameter specified level where resolution will be calculated. **5.** To calculate resolution under above discussed parameters just press button 'Calculation'. The results of calculation presented in picture 40. The red color in specter is approximation of specter (linear or cubic approximation). The black points are points which inserted in specter between data points. *Virtual Device* finds all peaks, calculate the width of peaks at threshold (left, right, center of peak), and calculate resolution of each peak

according with formula $R = \frac{\Delta E}{E} \cdot 100\%$. Calculated data presented in window **Results**. Button

allows save picture in file (bmp format).



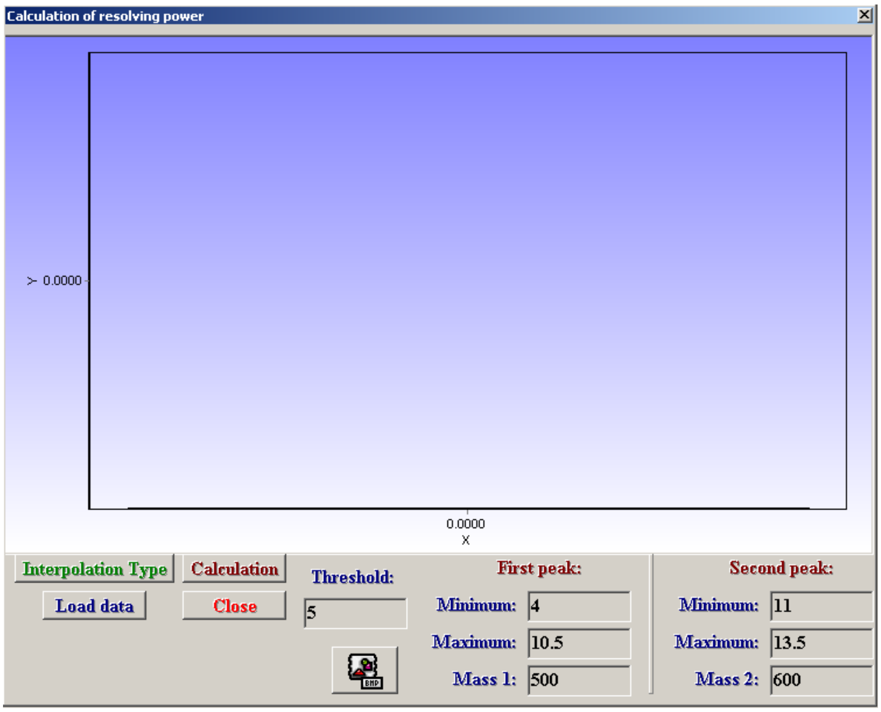
Picture 39. Window for calculation of resolution.



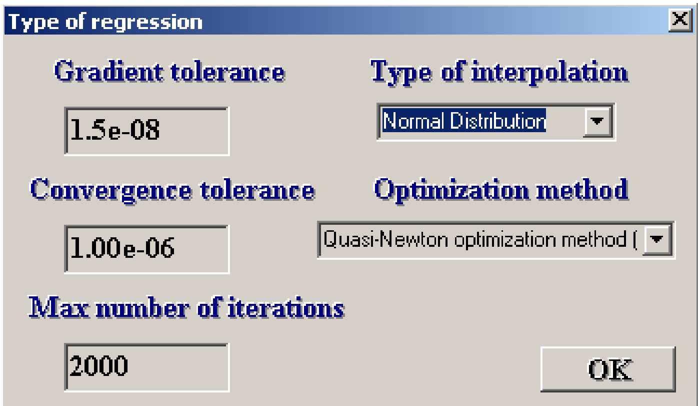
Picture 40. Results of calculation of resolution.

Resolving power. This option allows to calculate resolving power for mass specters (Formula: $R = \frac{m_1 + m_2}{m_2 - m_1} \cdot \frac{\Delta x}{d_1 + d_2}$). Here user has to specify approximation of both peaks, and masses which corresponds to peaks. Calculation of resolving power can be done in next steps. **1.** Load data from file. To do it just press button '**Load data**'. This data can be created by saving data in histogram window. **2.** Specification of type of interpolation (regression, see picture 42). **Type of interpolation** (Normal distribution or Eckerly function). Here user cans chouse type of interpolation for peak. The difference between Normal distribution and Eckerly function is just constant. The parameters: **Gradient tolerance, convergence tolerance, Max number of**

iteration, Optimization method – defines quality of approximation. It is necessary to notice that optimization method strongly influence on results of approximation.



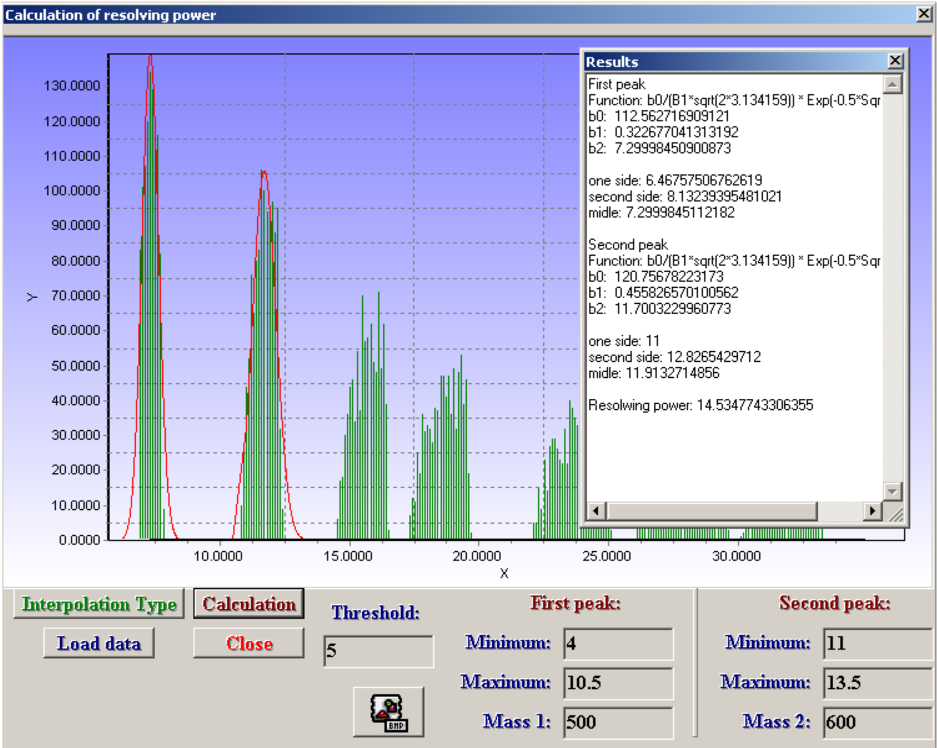
Picture 41. Window for calculation of resolving power.



Picture 42. Window for specification of type of regression.

3. Set up threshold. This parameter defines the level where resolving power calculates. **4.** Set up peaks which will be used for calculation of resolving power. **Minimum**, **Maximum** defines left and right wings of peak. **Mass 1, 2** parameters defines masses which correspond to both peaks. 5. Calculation of resolving power. To calculate resolving power just press button '**Calculation'**. The results of calculation are presented on picture 43. The red color is approximation of peaks.

The results of calculation are in window 'Results'. Button  allows save picture in file (bmp format).



Picture 43. Results of calculation of resolving power.

Example of results of resolving power calculation.

First peak
Function: y := B0/B1 * Exp(-0.5*Sqr((X-B2)/B1))
b0: 44.959248464339
b1: 0.322677006022181
b2: 7.29998453731304

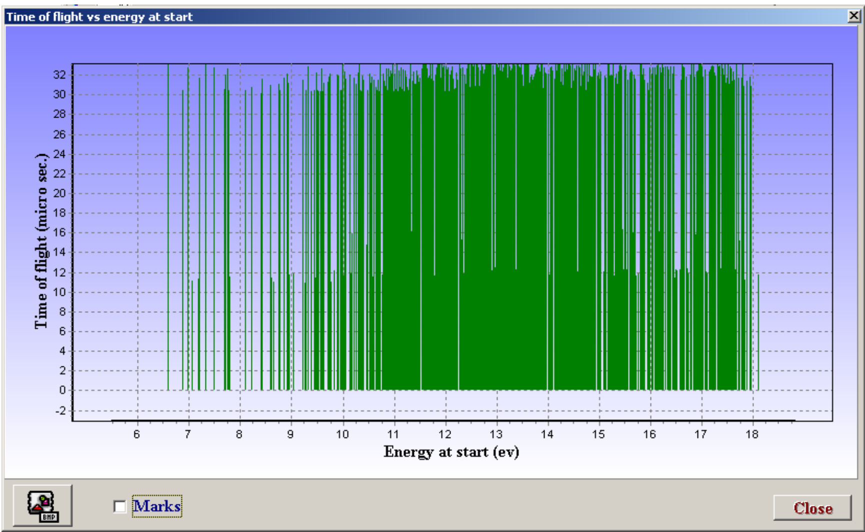
One side: 6.46757515636707
Second side: 8.13239391194106
Middle: 7.29998453415406

Second peak
Function: y := B0/B1 * Exp(-0.5*Sqr((X-B2)/B1))
b0: 48.232083486332
b1: 0.455826569442327
b2: 11.7003229921036

One side: 11
Second side: 12.8265429712
Middle: 11.9132714856

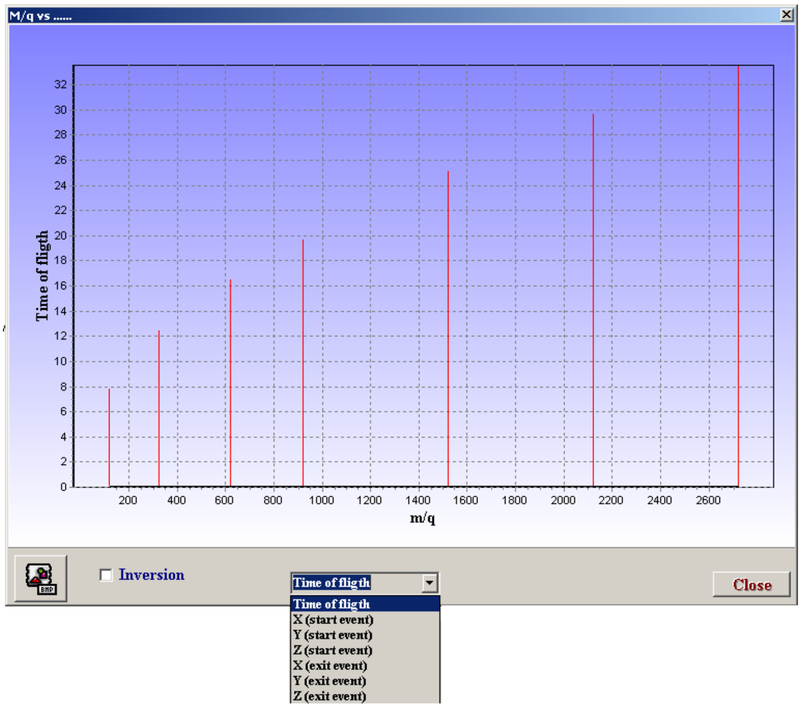
Resolving power: 14.5347748062745

Tof vs. Energy of start: This option allows creating chart with time of flight versus energy of start. Example of such chart is presented on picture 44. **Marks**. This option allows switching off/on marks. Button  allows save picture in file (bmp format).



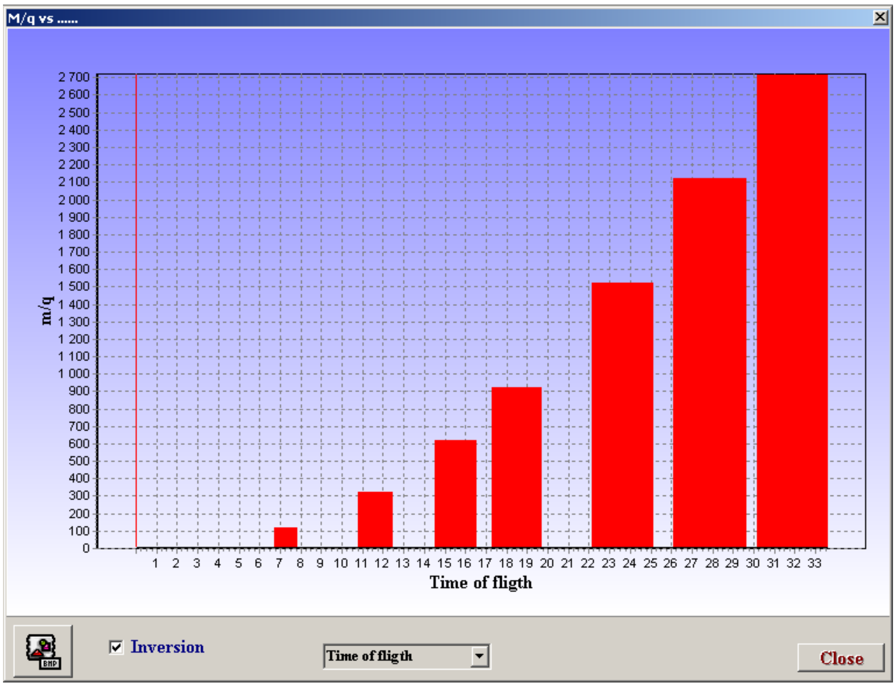
Picture 44. Chart of ion distribution: TOF versus energy of start.

m/z vs…..: This option allows estimate mass/charge ratio (first parameter) versus different parameters (second parameter). See picture 45. Dropdown list allows specifying the second parameter. Button  allows save picture in file (bmp format).



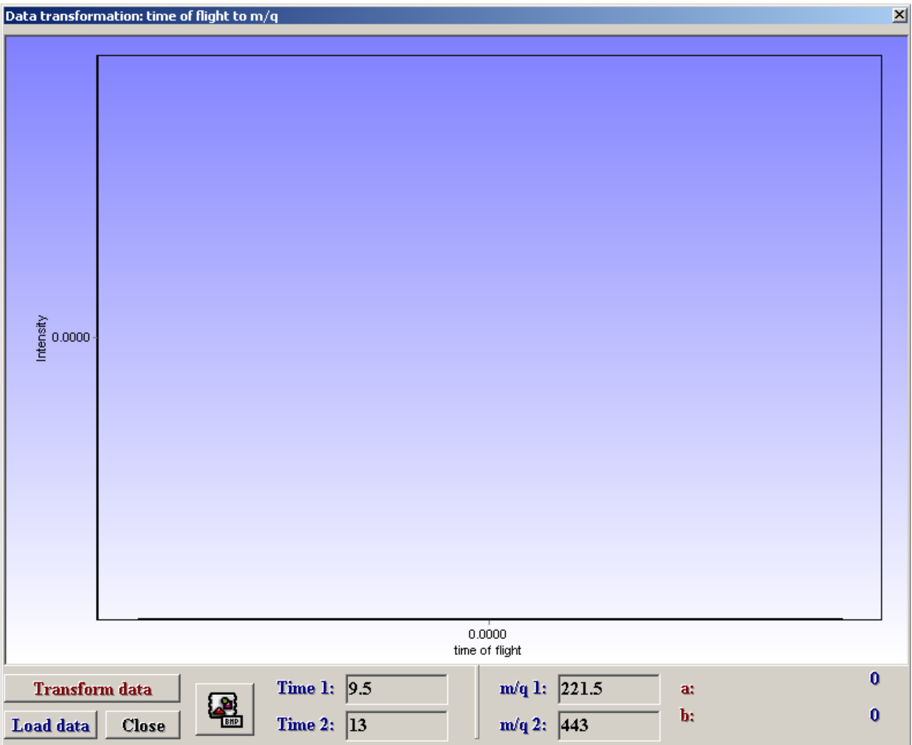
Picture 45. Chart of m/z versus…

The Inversion allows change axes. For example see picture 46.

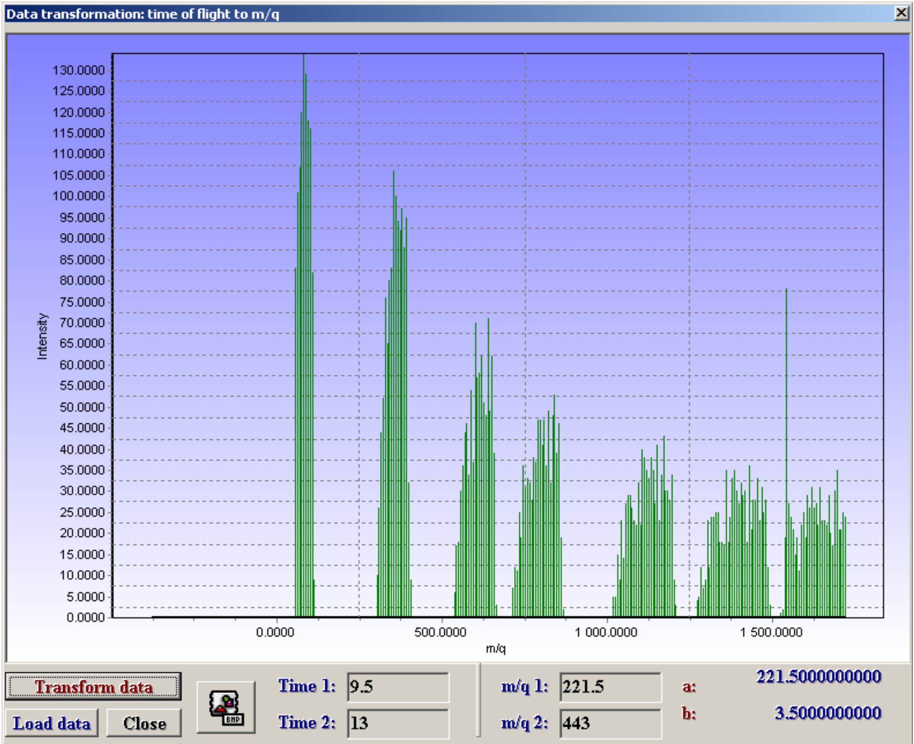


Picture 46. Chart of m/z versus… Inversion of axes.

TOF to m/z: this option allows transforms specter of time of flight (TOF) to ratio mass/charge (see picture 47). Transformation can be done in following step. **1.** Load data from file. This data can be done by saving data in histogram window. **2.** Set up parameters of transformation. This transformation based on following relation between mass/charge ratio and TOF: $m/z = a \cdot TOF + b$. User has to specify two points in TOF specter and related points in m/z terms. 3. Press button '**Transformation data**' and *Virtual Device* calculate coefficient **a** and **b** and make transformation of data (see picture 48).



Picture 47. Window of transformation of data in time to data in m/z.

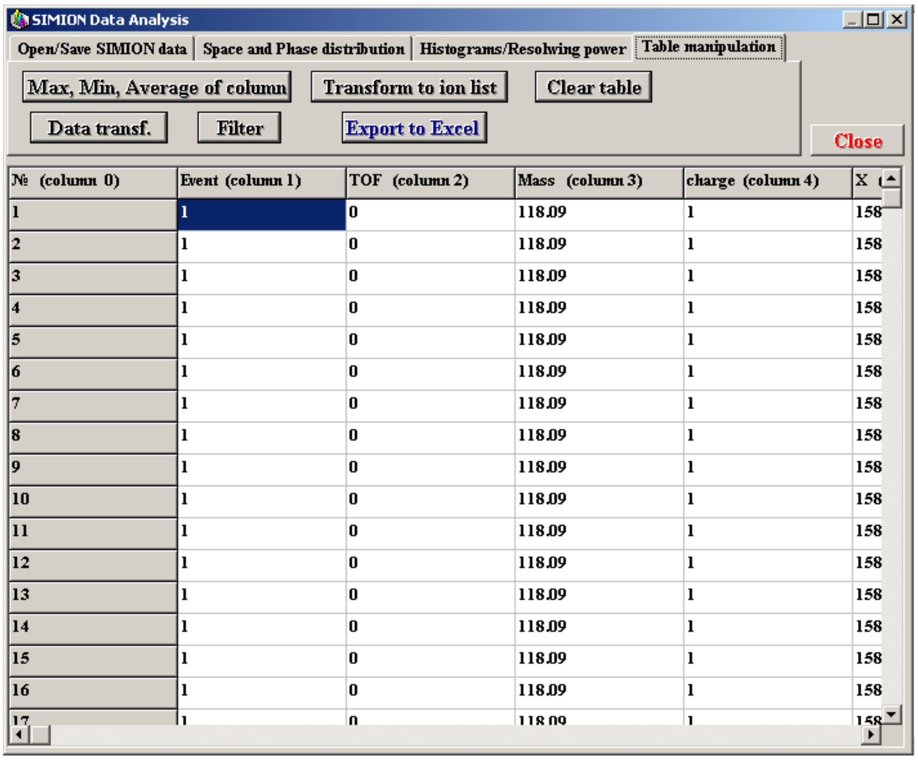


Picture 48. Result of transformation from TOF to m/q.

Ion mobility: This option is discussed in 'Simulation of ion movement in gas and related problems'.

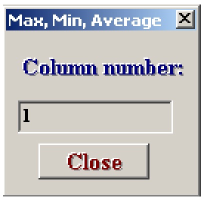
Chapter 2.4. Table manipulation.

This chapter covered discussion about manipulation of data in table (picture 49).

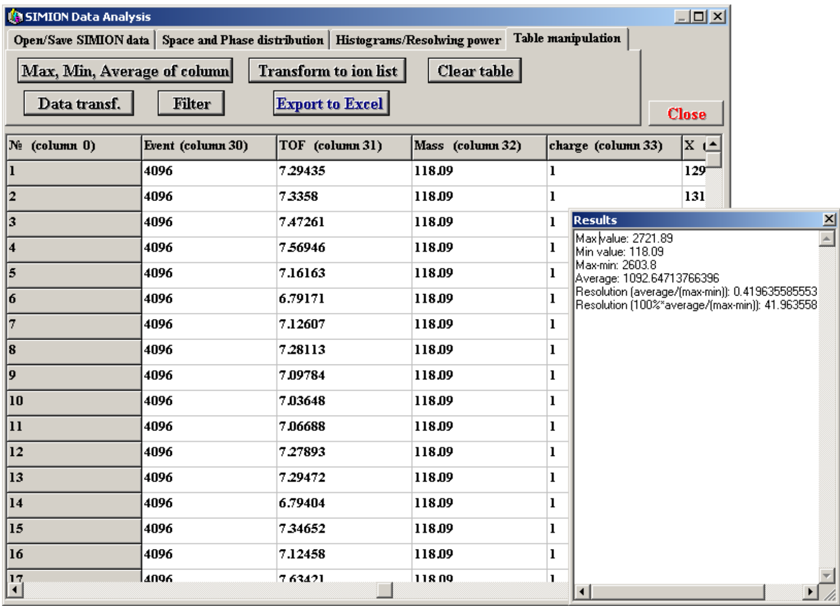


Picture 49. Table manipulation window.

Max, Min, Average of column: To calculate maximum, minimum and average value of data in column user has specify the number of column. Press this button and type number of column in following window:



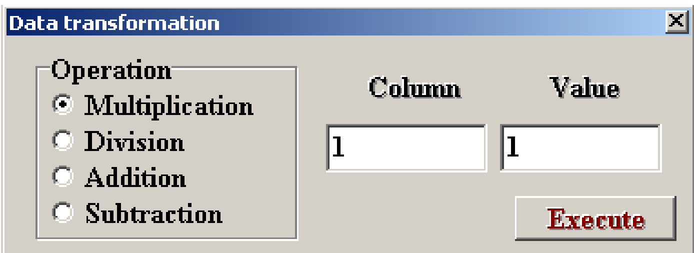
The results will be presented in next window (Results):



Window '**Results**' gives following data (example):
Max value: 2721.89
Min value: 118.09
Max-min: 2603.8
Average: 1092.64713766396
Resolution (average/(max-min)): 0.419635585553408
Resolution (100%*average/(max-min)): 41.9635585553408%

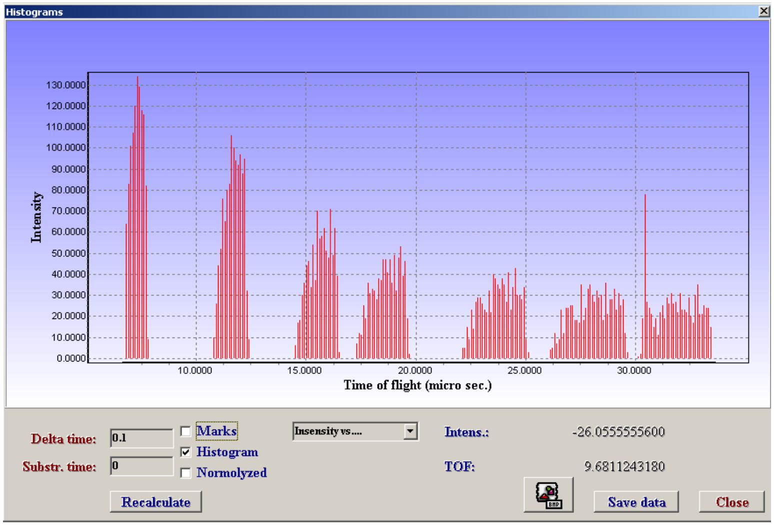
This window also gives possibility allows quickly estimate resolution, which calculated from different formulas.

Data transf.: This option allows changing the value in table. It can be done by following operation: multiplication, division, addition, subtraction. To change data user has to fill out following option:



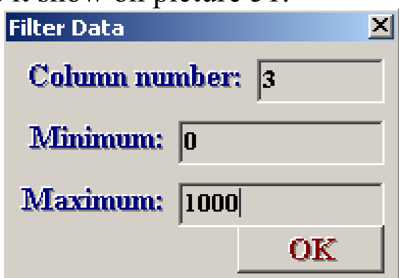
For example, you like to recalculate time of flight (column 31) from microsecond to second. 1. chouse **Multiplication**, set up **Column**: 31, set up **Value**: 0.000001, Press **Execute**.

Filter. This option allows deleting data from table, which does not satisfied to some criteria. Therefore it is some kind of data filtration. For example: Load data from file '\Agilent\Agilent7.txt'. To do it use Button '**Load data**'. This file contains data of 7 peaks, which corresponds to different time of flight (different mass/charge ratio). Histograms of those peaks are presented on picture 50.



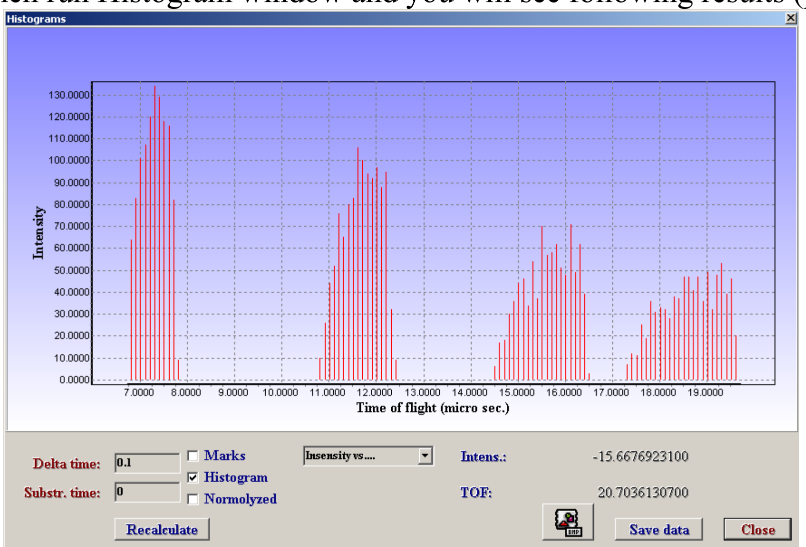
Picture 50.

Let us delete ions from table with mass/ charge ratio more then 1000 amu. Do to it press button '**Filter.**' and set up parameters as it show on picture 51.



Picture 51. Filtration data window.

Now press OK, then run Histogram window and you will see following results (picture 52):



Picture 52. Histograms after filtration.

Compare specter on pictures 50 and 52 you can estimate the effect of filtration.

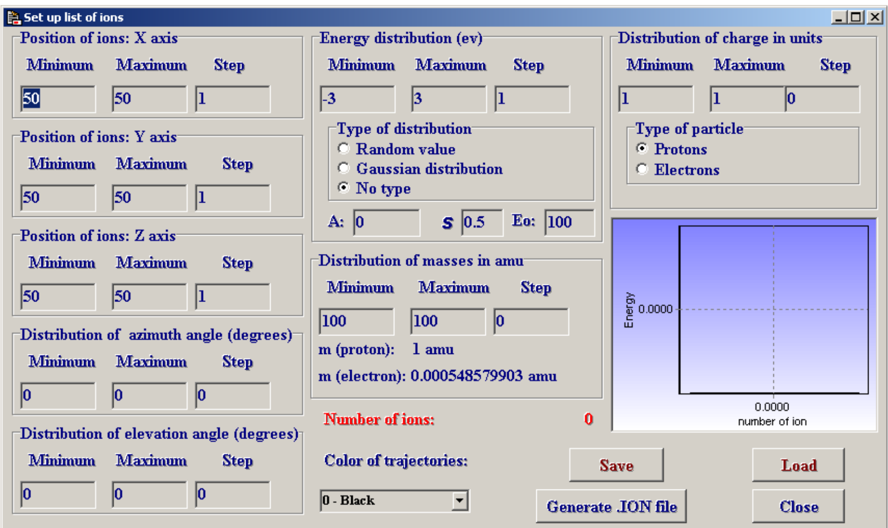
Tranform to ion list: This option allows transforming exit parameters of ions (death event) to list of ion. This list of ions user can use as initial ion distribution in SIMION simulation.

Export to Excel: this option can be useful for those users who like to use Microsoft Excel software for data analysis.

Chapter 3. Creation of ions list.

This part of *Virtual Device* allows creating the list of ions. This part is almost similar to the corresponding part of SIMION except energy distribution.
Parameters: **position of ions X, Y, Z, azimuth angle, elevation angle, masses, charge** are appropriate parameters of ions (see picture 53).
Energy distribution:

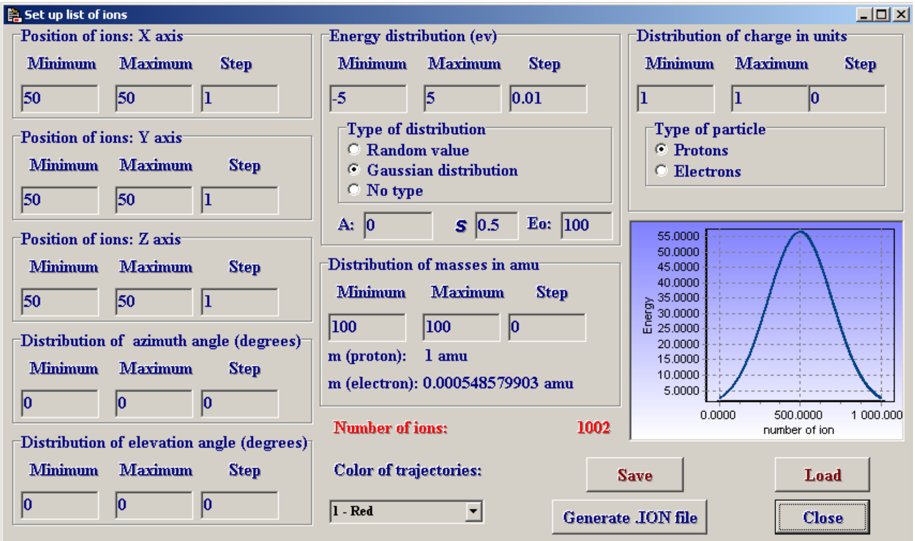
1. **No type**: You have to specify max, min and the step of energy. In that case you will produce a number of ions n=(max-min)/step. Energy will be in range min to max.
2. **Random value**: You have to specify max, min (e_max, e_min). The program will calculate linear function f=a*e+b, where b=e_min, a=(e_max-e_min)/1000. Then the program will run a cycle from e_min to e_max, and energy of ions will be E=a*random(1000)+b. Total number of ions will be 1000.
3. **Gauss distribution**. User has to specify A, sigma, E0, max, min, and step. In this case max and min means maximum and minimum of Gauss wings in terms of energy (energy axis). Value A means center of Gauss distribution, and sigm (σ) – dispersion. As you know, maximum of Gauss distribution is about 1. To get Gauss distribution on a higher level E_0 it is necessary to multiply Gauss by E_0.
   Program will run a cycle from min to max with step, ei – current energy in circle on energy axis. Energy distribution will be $E = \frac{E_0 e^{\frac{(ei-A)^2}{2\sigma}}}{\sqrt{2\pi\sigma}}$. Number of ions is (max-min)/step.



Picture 53. Creation of list of ions in *Virtual Device*.

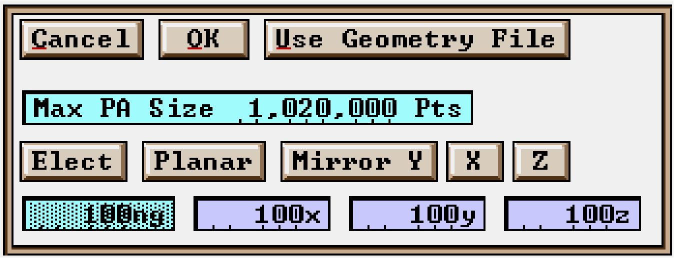
Save: this option stores current parameters in txt file. **Load:** this option loads parameters from txt file to window '**Set up list of ions**'. **Generation .ION file:** this option generates list of ions based on parameters and stores it to file in SIMION format.

For example (Gauss distribution): Set up parameters as is shown on picture 54.



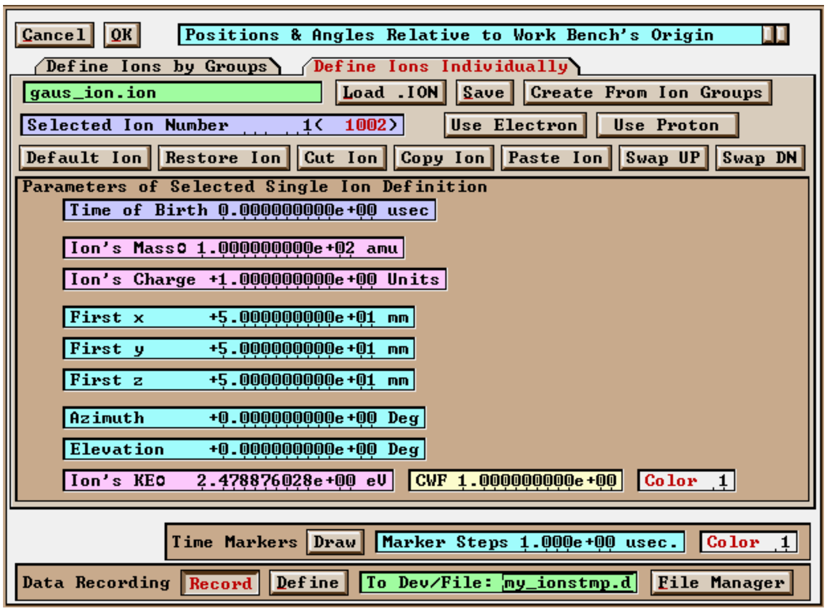
Picture 54. Generation of Gauss distribution.

1. Press button **Generation .ION file** and specify file where ions have to be stored (for example 'gaus_ion.ion'). **2.** Now you have to create an empty project (without electrodes), (see picture 55).



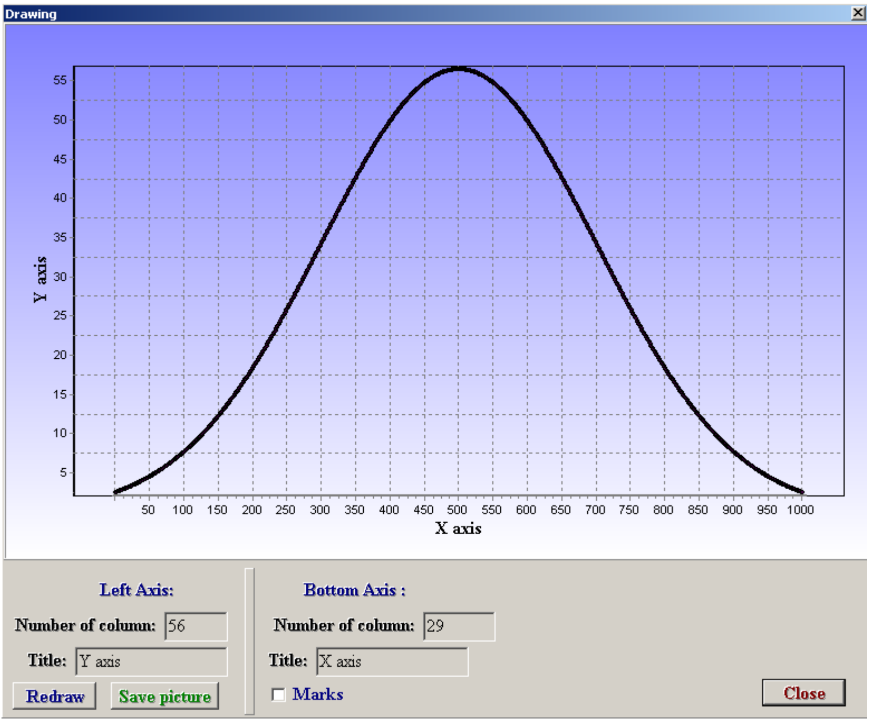
Picture 55. Creation of new project in SIMION.

4. Load list of ions from file 'gaus_ion.ion'. You can do it by pressing button '**Load .ION**' in **Defines Ions Individually**. See picture 56.



Picture 56. Load list of ions.

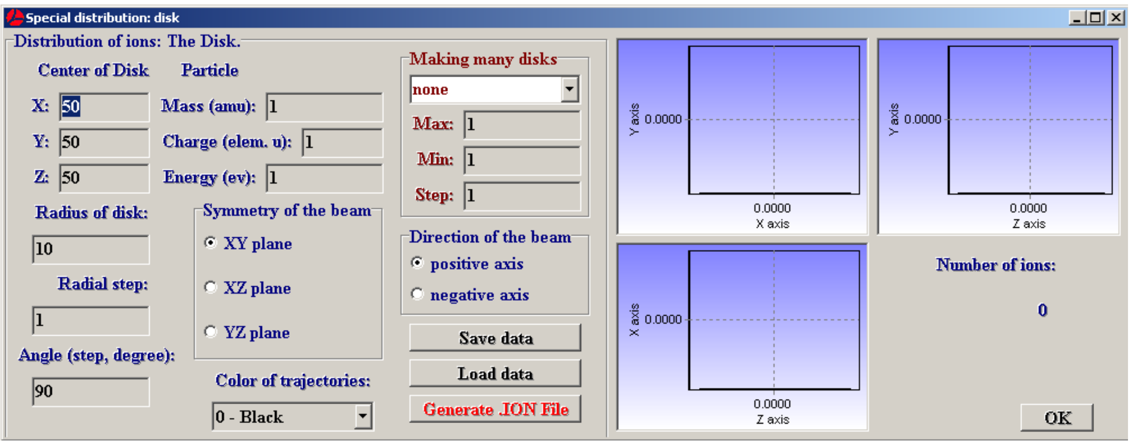
4. Specify files where results of simulation have to be stored. Do not forget about format of saving (see picture 31). Run ions. Since our ions will fly in an empty space, the initial energy distribution will not change.
5. Load results of simulation into *Virtual Device*. To see the energy distribution it is necessary to create a graph of energy versus number of ions. To do this, just point out mouse on the table and twice press quickly left mouse button. You will see a window where user can make charts with any column versus any column. Set up the number of the column for the bottom (**column 29** – number of ions) and the left axes (**column 56** – exit energy), and press redraw. You will see the following Gauss distribution (see picture 57).



Picture 57. Results of simulation of Gauss energy distribution.

Chapter 4. Creation of special Ion distribution disk.

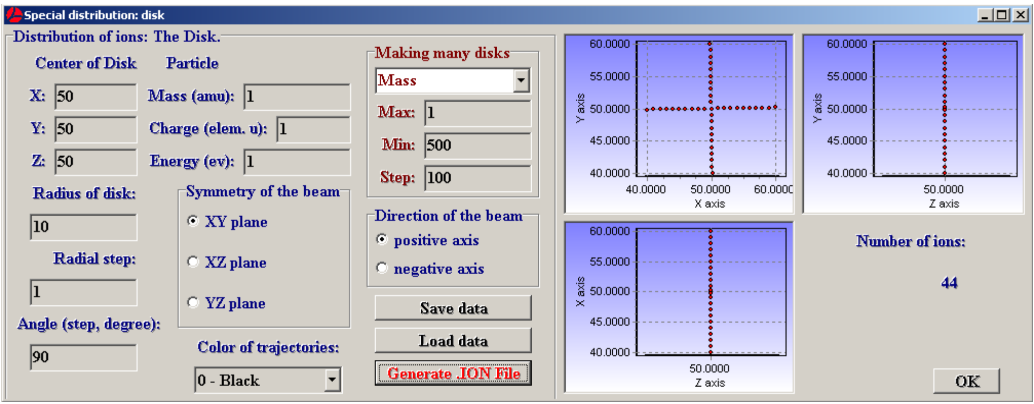
This part of *Virtual Device* allows creating ion distribution which presents itself as crossing of many lines (see 58).



Picture 58. Parameters of disk.

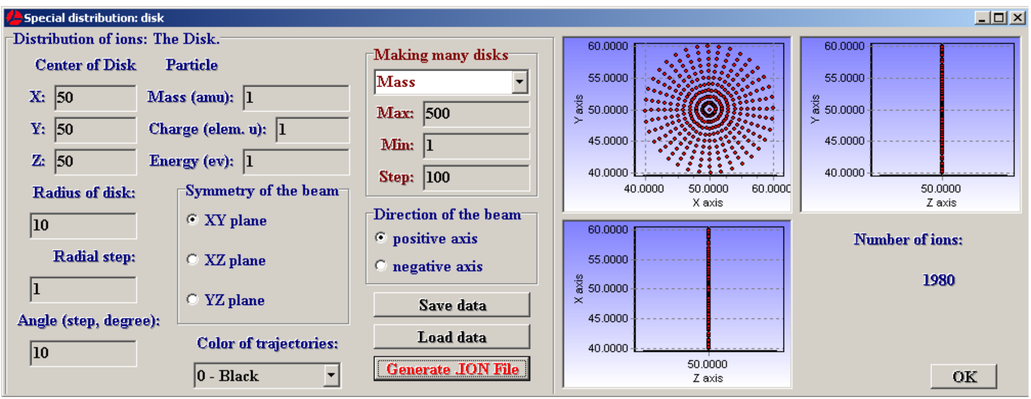
X, Y, Z – those parameters define center of the disk. **Radius of disk:** defines radius of disk. **Radial step:** defines step along radius. **Angle (step, degree):** this parameter defines the number of lines that can be drawn. For example, 90 means that lines will rotate with step 90 degree, therefore total number of lines in disk will be 4. **Mass, charge, Energy:** parameters which define type of ions. **Symmetry of the beam:** this parameter defines directions of beam. Actually it is setting up azimuth and elevation angles. **Color of trajectories:** this parameter defines color of trajectories in SIMION. **Direction of the beam:** this parameter defines direction of axis along which ions will fly.
Save data: this button allows saving current parameters into txt file.
Load data: this button allows loading parameters from txt file to window.
Generate .ION File: this button generates list of ion based on current parameters in window.

Making many disks: this parameter allows creating many disks where one of parameters is changing from disk to disk. **Max, Min, Step:** those parameters define the number of disks that will be created and which parameter will be changing from disk to disk.
Example 1. Set up the following parameters and press **Generate .ION File.** *Virtual Device* will generate a distribution which presents itself as a series of crosses where ion mass varies from 1 to 500 with a step 100 amu.



Picture 59. Ion distribution – cross.

Example 2. Set up parameter **Angle (step, degree):** 10 and press **Generate .ION File.** *Virtual Device* will generate the following distribution as disk (see picture 60). This distribution is 5 disks where the first disk is ions with mass 1 amu, the last disk is ions with mass 500 amu.



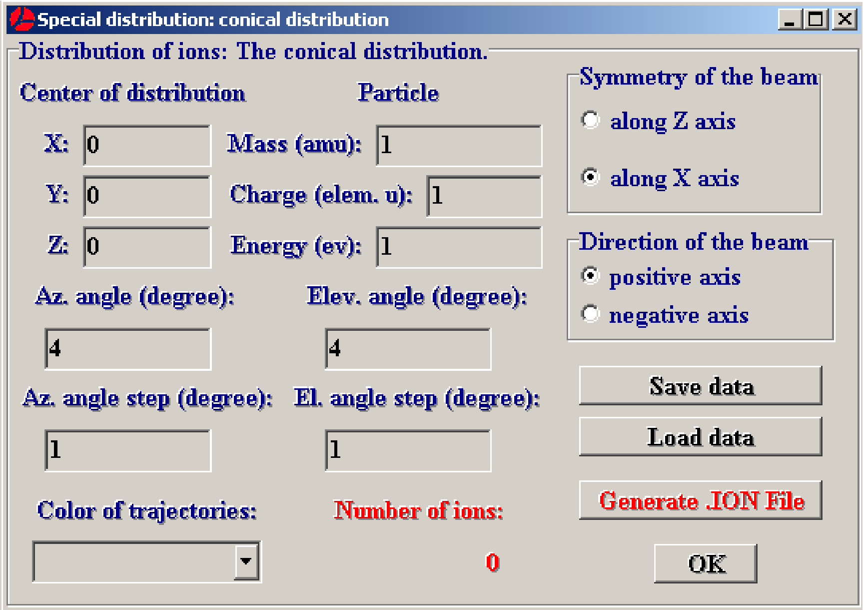
Picture 60. Ion distribution – disk.

Picture inside the window shows ion distribution as it looks in SIMION.

This kind of distribution can be very useful in ion optics investigation where user would like to investigate how first image transforms to second image. If ions distribution during the transformation in ion optics does not change, then this ion optics is ideal.

Chapter 5. Creation of special ion distribution - conical distribution.

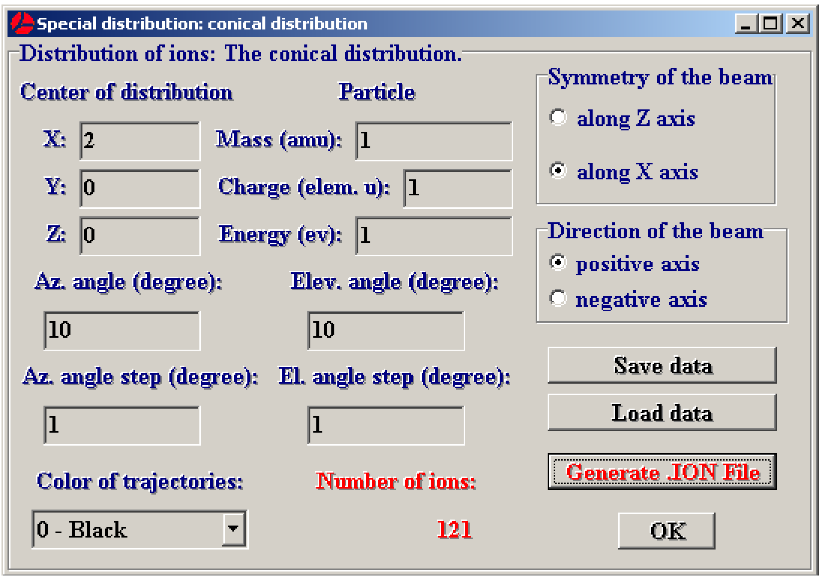
This part of *Virtual Device* allows creating ion distribution which presents itself as conical distribution (see picture 61). Ions start from a point source and fly under two angles according to axis.



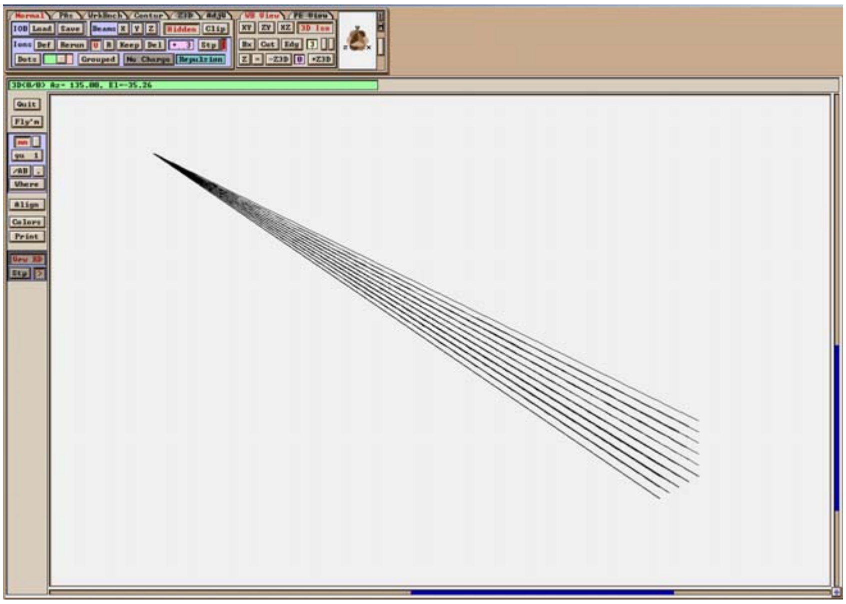
Picture 61. Parameters of conical distribution.

X, Y, Z – these parameters define position of the point source in space. **Az. Angle (degree), El angle (degree)** define divergence of azimuth and elevation angles. **Az. Angle step (degree), El angle step (degree)** define steps of changing of azimuth and elevation angles. **Mass (amu), Charge (elem. u), Energy (ev):** these parameters define type of ions. **Symmetry of beam, Direction of beam** define direction of ions in space. **Save data, Load data** allow storing (loading) current parameters in txt file (from txt file). **Generate .ION file:** this option generates list of ions based on parameters in window. **Number of ion**: it just shows how many ions were stored in file .ION.

Example: 1. Set up parameters as is shown on picture 62. 2. Create an empty project in SIMION. 3. Load ions to SIMION and run ions. You will see the following picture (see picture 63).



Picture 62.



Picture 63.

It is necessary to note that creation of ions along Y axis is complicated due to an internal problem of SIMION v.7 This problem was resolved in version 8.

Chapter 6. Main parameters of mass-spectrometers / energy analyzers.

Let us discuss briefly the main mass-spectrometers' parameters relevant to different ways of ion separation.

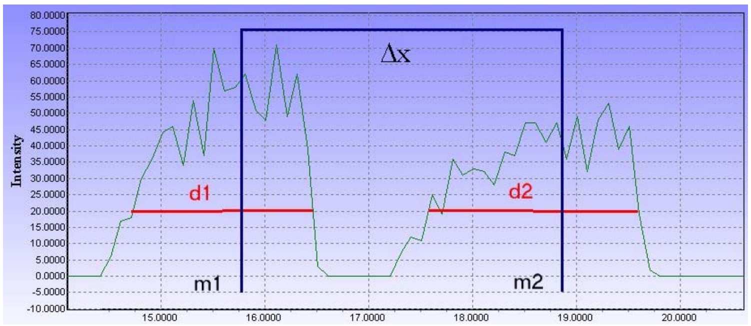
Dispersion. This parameter defines how mass-spectrometer can effectively separate ions according to mass/charge ratio. Suppose we have mass specter which was got by scanning a parameter ω. For example, parameter ω can be magnetic field or accelerating voltage. The increment of ω as function of mass can be presented as the following series (we suppose charge =1):

$$\Delta\omega = \lambda_1 \frac{\Delta m}{m} + \lambda_2 \left(\frac{\Delta m}{m}\right)^2 + ...$$

where λ_1 is often called 'dispersion' when $\Delta\omega$ is a distance between two peaks (m and Δm) [2]. If we take the first term of the series then dispersion can be expressed as follows: $D_m = m\frac{\Delta x}{\Delta m}$, where Δx is distance between two peaks. It is necessary to note that some scientists regard dispersion as Δx only. From our point of view is not a good definition because Δx becomes function of $\frac{\Delta m}{m}$, therefore it is necessary to specify ratio $\frac{\Delta m}{m}$ each time when dispersion is discussed. Our definition is more convenient because the value of dispersion depends only on geometrical and physical parameters of mass-spectrometers.

Resolution (resolving power) in mass spectrometry. It is supposed that ion with mass m corresponds to position x and (m+Δm) corresponds to (x+Δx). Δx can be expressed in terms of dispersion: $\Delta x = D_m \frac{\Delta m}{m}$. We can resolve two peaks if $\Delta x \geq$d, where d is width of peak at base of peak (we suppose that both peaks are equal). Therefore resolution of the mass-spectrometer can be expressed in the following way:

$R = m\frac{\Delta x}{\Delta m \cdot d}$. Actually assumption about equality of peaks is not true. Therefore we have to take into account the difference in width of peaks. Resolution in this case can be expressed as follows: $R = \frac{m_1 + m_2}{m_2 - m_1} \cdot \frac{\Delta x}{d_1 + d_2}$, where m_1, m_2 - masses of ions which correspond to two peaks, d_1, d_2 - the widths of peaks at base [2]. Resolution is often defined at level 50% or 10% of intensity of peak. Therefore d_1, d_2 are widths of peaks at those levels (see picture 39).



Picture 39.

Resolution in electron spectroscopy. The definition of resolution in electron spectroscopy is slightly different from definition in mass spectrometry. Resolution can be expressed as follows (in percent): $R = \frac{\Delta E}{E} \cdot 100\%$, where ΔE is width of peak at level 50% of intensity of peak, E is energy of peak (center of peak) [3,4]. But in principle it is possible to use other threshold (level). It is necessary to note that in electron spectroscopy resolution ($R = \frac{\Delta E}{E} \cdot 100\%$) is used, but in mass spectrometry resolving power is used ($R = \frac{m}{\Delta m}$).

Terminology

1. Geometrical object – It is single electrode with certain number geometrical parameters, which completely define size of electrode and its’ position in space.
2. Project - Project presences itself a combination of geometrical objects inside of 3D volume.
3. Mobile point – When user set up geometrical object it is necessary to set up position of center of geometrical object. This position can be settled relatively to crossing of axis (x=0, y=0, z=0) relatively to some point (x, y, z). This point has name mobile point.

Refences

1. Dahl David A. SIMION version 7, user’s manual, 2000.
2. Кузема А.С., Савин О. Р., Чертков И. Я., Анализирующие системы магнитных масс-спектрометров, Киев, Наукова думка, 1987.
3. Koltsov S, High transmition electron optics with quasi mirror intermediate stage, Ph thesis, 2000, Sank- Petersburg.
4. Голиков Ю.К., Уткин К.Г, Чепарухин В.В., Расчет элементов электростатических электронно-оптических систем, учебное пособие, Ленинград, 1984.
5. Archangelsky A, Examples of programming in Delphi, version 5-7, Delphi for professionals, Moscow, publisher ‘Binom’, 2003.