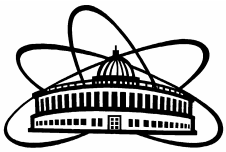


BETACOOL User Manual

based on BOLIDE interface

(since 1995)

I.Meshkov, A.Sidorin, A.Smirnov, G.Trubnikov, R.Pivin

Joint Institute for Nuclear Research
Joliot Curie, 6, Dubna, 141980 Russian Federation

http://lepta.jinr.ru/betacool

Dubna, 2007

Introduction

The user manual describes how to work with BETACOOL software developed for calculation of ion beam parameters in a storage ring taking into account peculiarity of electron, stochastic and laser cooling, intrabeam scattering processes, beam interaction with residual gas, beam interaction with internal target, interaction with colliding beam in a collider mode of the ring operation.

The version of the program includes three basic algorithms:

- RMS Dynamics simulates evolution in time of second order momentum of the ion distribution function (r.m.s. emittances) under a common action of a few heating or cooling effects which are described in terms of characteristic times of the beam r.m.s. parameter variation.
- Model Beam simulates evolution of the distribution function shape. The beam is presented as an array of modelling particles. Evolution of the particle momentum components is described in terms of Langevin equation. Each heating or cooling effect is characterized by friction and diffusion components. The friction leads to regular momentum variation, the diffusion is simulated using random number generator.
- Tracking provides a tracking of a particle array along the ring circumference with arbitrary step using Molecular dynamics technique for intrabeam scattering simulation.

The software is divided in two independent parts: physical code (BETACOOL), which is written using only standard C++ syntax and interface part, which is an executable program working under Windows environment (BOLIDE). Connection between two parts of the program is provided using three types of the files: input, output and file used for control of the calculation process. Such a structure on the one hand allows to use the program on PC, to control and analyse results during simulations. From the other hand the physical part of the program compiled for UNIX operation system can be used for calculations independently on interface.

Contents

I. Interface

Introduction

The interface part of the software consists of executable file **Bolide.exe**, files containing information about BETACOOL exterior and input files for post processing of the calculated data:

bolideNN.dfm – each file corresponds to *Form* and describes parameters of visual components;
bolide.top – position of *Forms*
bolide.grf – setting of 2D graphs
bolide.srf – setting of 3D graphs

The physical part of the software consists of the executable file **Betacool.exe** compiled for Windows or UNIX operation system and a few input and output files:

betacool.war – output file which saves of console messages
***.bld** – input files with simulation parameters
***.ela** – input files with distribution of electrons
***.err** – input files with distribution of solenoid errors
***.inj** – input files with initial ion distribution
***.lat** – input files with lattice structure of storage ring
***.lvt** – input files with table of longitudinal components of the electron cooling friction force
***.mad** – input files in the format of input MAD8 program
***.pat** – input files of painting procedure for electron beam shift
***.red** – input files of positions for the reduction of lattice structure
***.tfs** – input files with lattice structure in the format of output MAD8 program
***.tvt** – input files with table of transverse components of the electron cooling friction force
***.cur** – output files with 2D graphs (see Appendix for details)
***.srf** – output files with 3D graphs (see Appendix for details)
***.use** – output files with lattice structure

The software also includes the total set of BETACOOL source codes: *.cpp and *.h files, project files for compilators:

betacool.dsp and **benacool. dsw** - MS Visual C++ (Windows),
betacool.bpr - C++Builder (Windows),
betacool.cbx - BuilderX (Windows / LINUX),
makefile and **objects** for GCC compilator (LINUX)
#backup.bat and **#upload.bat** – command files for saving of BETACOOL package
#save.bat – command file for saving of current 2D and 3D graphs in separate folder
mad8win.bat – command file for run MAD8 program
mad8.exe – MAD8 program
mad8.dic – dictionary of MAD8 program

The first part of this guide describes structure of input and output files of the program, procedures which can be executed with the program, files for the calculation process control. This part contains mainly background information and to start simulations on PC one can start directly from the second part.

The second part describes a structure of the Windows interface and contains simplest examples of input file preparation and simulations with general procedures of the program using the interface.

1. Work without graphical interface

1.1. Starting the calculations, control of the calculation process

The simplest way to start a work with BETACOOL program is to save the **Betacool.exe** file, file of input parameters (*.bld format), other required input files (from archived Betacool kit – download from website http://lepta.jinr.ru/betacool) and #SAVE.bat file into the same folder.

To start the BETACOOL program one needs to type in the command line the following command:

<path>\betacool.exe <inputfilename> /<parameter>

The parameter <inputfilename> is obligatory. If this parameter is skipped the program generates in the output flow the warning message presented in the Fig. 1 and stops the calculation after press Enter.

Fig. 1. Betacool output in the case, when <inputfilename> parameter is skipped.

If the specified file is absent in the current program folder, the program generates the following warning

```
START - 2006 / 11 / 24 - 13 : 22 : 39
D:\sao\GSI\2006\betacool_GSI_v2\betacool.exe NE.bld /rates
Cannot open file :  NE.bld
```

, and stops the calculations.

All the messages and warnings generated by the program during calculations are saved in the current folder to the file **Betacool.war**.

Depending on <parameter> value Betacool can execute corresponding procedure and stop the work or provide simulations in infinite cycle.

In the last case to stop the work of the program one needs to create in the same folder the file **bolide.stp**. The program will delete this file and stop the work. To pause the calculations (for instance, to edit input file) one needs to create in the same folder the file **bolide.pau**. To continue the calculations one needs to delete this file. To change parameters during calculations one needs to make necessary editions in the input file and create the file **bolide.run**. The program will re-read the input file, delete the **bolide.run** file and continue the calculations. (when working with interface these files are generated by the interface tools.)

During simulations the program creates output files and saves them in the current folder. To save a total set of input and output files for further post processing one needs to execute **#SAVE.bat** file. In the current folder the subfolder with the name **#<inputfilename>** will be created and all input and output files will be copied into this subfolder. Whereupon one can rename input file and continue the simulations.

1.2. BETACOOL procedures

Parameter	Procedure
rates	Sum of the rates of active effects Output into input file
3d	3D maps of sum of the rates in the range of the beam emittances specified in and momentum spread specified in *output* – **rateeh.sur, rateev.sur, ratedp.sur**
friction	3D maps of the friction force in the range of plot of the friction force components as function of the ion velocity at a fixed angle between the velocity and electron beam axis *output:* **ftr.cur, flong.cur** plot of the friction force components as function of angle between the velocity and electron beam axis at a fixed ion velocity *output:* **fatr.cur, falong.cur**
g	Generates tables of the friction force in the range of ion velocity *output:* **fftr.sur, fflong.sur**
space	Potential distribution inside electron beam *output:* **inecool.cur, charge.cur, vdrift.cur**
lattice	Calculation of lattice functions along the ring
collision	Choice of parameters for luminosity calculation in the frame of Model Beam algorithm
injection	Visualization of the initial ion distribution when the injected beam parameters are
dynamics	RMS dynamics algorithm
model	Model beam algorithm
tracking	Tracking procedure

Note: program analyses only the first letter in the parameter name.

If <parameter> field is empty the program reads input file and generates list of parameters skipped in the input file and their default values. For instance if after program termination the **Betacool.war** contains the following warnings:

```
START - 2006 / 11 / 24 - 17 : 8 : 14
D:\sao\GSI\2006\betacool_GSI_v2\betacool.exe EDMD.bld
Input parameter [ 87 , 1 ] was initialised with default value: 0
Input parameter [ 87 , 2 ] was initialised with default value: -0.001
Input parameter [ 87 , 3 ] was initialised with default value: 0
Input parameter [ 87 , 4 ] was initialised with default value: 2
Input parameter [ 87 , 5 ] was initialised with default value: 0
Input parameter [ 87 , 6 ] was initialised with default value: 0
Input parameter [ 87 , 7 ] was initialised with default value: 0.001
Input parameter [ 87 , 8 ] was initialised with default value: 2
END - 2006 / 11 / 24 - 17 : 8 : 14
```

it means that in the file EDMD.bld the parameters [87, 1 – 8] are absent, and for calculations their values indicated in the last position of warning will be used.

1.3. Input files

General input parameters are collected in *.bld file. The format of this file is developed in a way to make easy collect and edit it in any simple editor. All the parameters are provided with comments and structure of parameters list is connected to the program algorithm.

The example of one of the parameter groups is presented below:

```
[row=1] Beam | Parameters | Emittance
1.6= Horizontal emittance, pi*mm*mrad
1.6= Vertical emittance, pi*mm*mrad
0.0016= Momentum spread
100000000= Number of particles
0= bunched(0), coasting(1)
0= Collider regime (0/1)
0.008281840936= Mean beam radius, m
10.58738= Longitudinal form factor
746.4608902= Longitudinal space charge impendances, Ohm
87519902.22= Transverse space charge impendances, Ohm/m
0.0006055379817= Peak current, A
3= Emittance definition: RMS(0),CS(1),FWHM(2),%(3)
=
35= Percents for transverse degrees of freedom
68= Percents for longitudinal degree of freedom
1; Mean Longitudinal Invariant
```

In the *.bld file names of files required for calculations in specific cases are also specified. In the example below the parameters **`nesr_p034a.tfs`**, **`r005.mad`** and **`rhic.red`** are the filenames of output MAD, input MAD files and file describing procedure of the lattice structure reduction correspondingly. If they are necessary for simulations they have to be saved in the same folder with Betacool.exe.

```
[row=13] Ring | Lattice Structure | Lattice Filename
0= Lattice File: Output MAD (0), Input MAD (1), No file (2)
2= Lattice Structure: Reduce (0), Extend (1), No changes (2)
1= Extend step, cm
nesr_p034a.tfs= Output MAD Filename
0= Auto skip of points (0/1)
r005.mad= Input MAD Filename
1=
rhic.red; Reduce Filename
```

1.4. Output files

As a result of simulations BETACOOL modifies the input file (*.bld format) and creates a few files from the following list:

The files `*.cur` as a rule contain time dependencies of the beam parameters. Example of the `*.cur` file format:

```
0     0.0003849600109
18    0.0003703453546
36    0.0003563822554
54    0.0003430464386
72    0.0003303146755
90    0.0003094731154
99    0.0002900512001
108   0.0002719605886
117   0.0002551191811
```

The first column of the files with beam parameter time dependencies contains the current time in [sec], second column – values of corresponding variable. The columns are divided by tabulator symbol. Strings are finished by the *end of line* symbol. List of graphs is described in Appendix.

The files `*.sur` contain functions of two variables. Example of `*.sur` file:

```
      0     0.000333333 0.000666666 0.001
0     -0    -0    -0    -0
0.000333333 2.98404e-12 2.98364e-12 2.98243e-12 2.9804e-12
0.000666666 2.16460e-12 2.16452e-12 2.16426e-12 2.1638e-12
0.001 1.01881e-12 1.01879e-12 1.01873e-12 1.01864e-12
```

First element in the first string of the file is empty. Other elements in the first string contain the value of first variable. The elements of the first column contain the values of second variable. Corresponding value of the function are placed at the cross of the string and column. Numbers in the string are divided by *tabulator* symbol. Strings are finished by the *end of line* symbol.

For post-processing of the output BETACOOL files one can use any graphical editor.

2. Working with windows interface

The interface part of the software consists of executable file **Bolide.exe**, ***.dfm** files containing information about BETACOOL exterior and input files for post processing of the calculated data.

The simplest way to start a work with BETACOOL program is the following:
- to save the **Interface** files, **Betacool.exe** file, file of input parameters (*.bld format) and other required input files (from archived Betacool kit – download from website http://lepta.jinr.ru/betacool) to the same folder;
- to start **Bolide.exe** file, load the input file, check its validity, to import absent parameters from correct input file (if necessary);
- to start BETACOOL program using one of the `TBrowse` components in the visual Windows of the **Interface** and click **Open** button.
- BETACOOL program is working as Win application and stops the calculations after their completion, or can be stopped using corresponding **Interface** tool,
- during the calculation the **Interface** program automatically reads the results from output files and represents them in numerical or graphical format to the corresponding windows.

The detailed description of the interface is presented in **Bolide.doc**.

2.1. Load input file, check its validity, import of absent parameters

After start the **Bolide.exe** file the *main window* of the interface is opened (Fig.2). (If the desktop was saved the windows active in previous run are opened also.) *Main window* manages the interface operation and has global menu and buttons which duplicate actions from the **Files** *menu*. The bar title indicates the name of input file. At usual setting the interface opens input file used in previous run. To open new file one needs to use *menu item* **File** *submenu item* **Open**, or corresponding button in the toolbar. It opens a new input file using the standard MS Windows dialog.

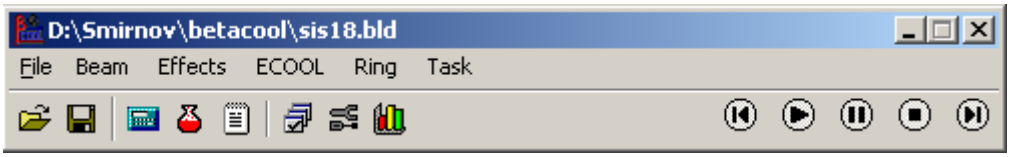


Fig. 2. Main window of the BOLIDE interface for BETACOOL program.

Main window has the following buttons for working with interface:
– open the input file with using Windows Open dialog;
– save input file;

– calculator tool;

– periodic table of chemical elements;

– text editor (by default open **betacool.war**)

– setup dialog window of the interface program;

– dialog window for elaboration and editing of exterior;

– redraw all graphics;

- restart calculation with new parameters;

- start calculation process;

- pause calculation process;

- restart with changed parameters. If some changes of parameters (in input file via interface) during calculation were made by user then program will reread initial file and will continue calculation with new parameters;

- stop calculation. To stop the program with correct OS memory cleaning we advise to use this item. In this case a special file-marker will be created in the current directory and as soon as program will find it then it immediately will shutdown with correct saving all the calculated results.

All those buttons are duplicated as menu items in Main menu.

To check a validity of the input file one needs to use *menu item* **Task|Parameters** and push the *button* **Open** at the *TBrowse* component **Check parameters** (Fig. 3). This component starts Betacool without parameter. The program reads input file and generates list of parameters skipped in the input file and their default values and saves it in the **Betacool.war** file.

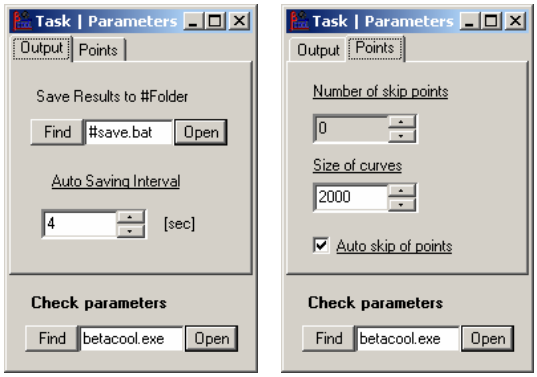


Fig. 3. Window Task | Parameters

To read the **Betacool.war** file one can use *menu item* **File|Editor** or corresponding button from the toolbar. The internal editor is generated to open the **Betacool.war** file. If the input file contains total set of input parameters the warnings are absent (Fig. 4).

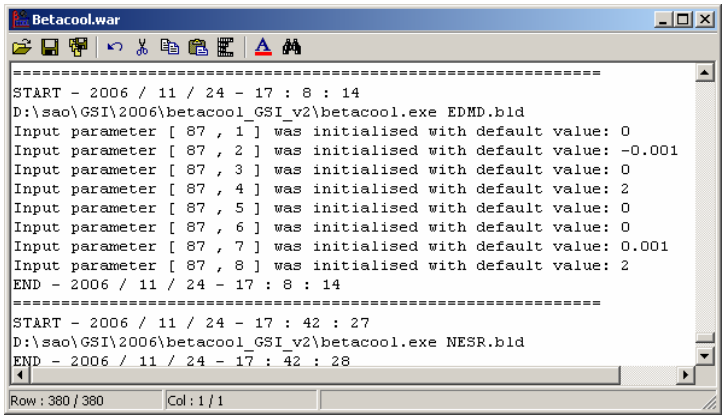


Fig. 4. **Betacool.war** file in the **Editor** *window*. The warning after two consequent runs of the **Check parameters** procedure with input file **EDM.bld** and **NESR.bld**. In the **EDM.bld** file a few parameters are skipped, the **NESR.bld** contains total set of parameters.

To import the absent parameters into input file one needs to open a file containing total set of parameter and use *menu item* **File|Import** and load required file using the standard MS Windows dialog. In the example presented in the Fig. 4 to provide simulations with **EDM.bld** file one needs to open **NESR.bld** file and import **EDM.bld** file. After, the skipped parameters in the **EDM.bld** will be substituted from **NESR.bld** and **EDM.bld** saved with total set of parameters.

2.2. Parameters of calculation for Windows Interface

Parameters of calculation can be defined on the *Form* **Task | Parameters** (Fig.3). To save results it's needed to push *Button* **Open** (**Save Results to #Folder**) on *TabSheet* **Output**. All results will be collected to current folder. The name of the folder depends of input file name (for example for input filename **ESR.BLD** the folder name will be **#ESR.BLD**). **Auto Saving interval** defines the time interval after what the Betacool code saves all output files on disk.

Number of skip points indicates how many points of 2D graphs will be saved: 0 – means all point are saved, 1 – each second points are saved, 2 – each third, etc. **Size of curves** – length of 2D graphs. If output points reach the limit of curve size then the output points will be saved on the beginning of curve. If **Auto skip point** is switched on then parameters **Number of skip point** will be automatically incremented when output points reach the limit of curve size.

Appendix. List of graphs

alfax.cur	horizontal alpha function	Ring | Lattice Structure | alfa functions
alfay.cur	vertical aplpha function	Ring | Lattice Structure | alfa functions
betax.cur	horizontal beta function	Ring | Lattice Structure | beta functions
betay.cur	vertical beta function	Ring | Lattice Structure | beta functions
bunch2t.cur	bunch length on time	Beam | Evolution | Bunch
charge.cur	space charge parabola of electron beam	ECOOL | Cooler | Space charge
dispx.cur	horizontal dispersion	Ring | Lattice Structure | beta functions
dispx_.cur	derivative of horizontal dispersion	Ring | Lattice Structure | alfa functions
dist_dp.cur	sorted longitudinal invariants	Beam | Distribution | Invariant
dist_ex.cur	sorted horizontal invariants	Beam | Distribution | Invariant
dist_ey.cur	sorted vertical invariants	Beam | Distribution | Invariant
dist_ip.cur	longitudinal beam profile	Beam | Distribution | Profile
dist_ix.cur	horizontal beam profile	Beam | Distribution | Profile
dist_iy.cur	vertical beam profile	Beam | Distribution | Profile
dist_sp.cur	distribution of momentum spread	Beam | Distribution | Coordinate
dist_sx.cur	distribution of horizontal coordinates	Beam | Distribution | Coordinate

dist_sy.cur	distribution of vertical coordinates	Beam | Distribution | Coordinate
dp2t.cur	momentum spread on time	Beam | Evolution | Momentum
dpmo2t.cur	deviation of momentum spread on time	Beam | Evolution | Momentum
ex2t.cur	horizontal emittance on time	Beam | Evolution | Emittance
ey2t.cur	vertical emittance on time	Beam | Evolution | Emittance
falong.cur	longitudinal electron cooling force	ECOOL | Draw force | 1-D force
fatr.cur	transverse electron cooling force	ECOOL | Draw force | 1-D force
flong.cur	longitudinal electron cooling force	ECOOL | Draw force | 1-D force
footprint.cur	particle invariants	Beam | Evolution | 3D Diagram
ftr.cur	transverse electron cooling force	ECOOL | Draw force | 1-D force
gamma2.cur	ordering criteria	Beam | Evolution | 3D Diagram
gamma3.cur	temperature equilibrium of beam	Beam | Evolution | 3D Diagram
inecool.cur	particles in cooler section	ECOOL | Cooler | Space charge
kappax.cur	horizontal beam-beam on time	Beam | Evolution | Beam-beam
kappay.cur	vertical beam-beam on time	Beam | Evolution | Beam-beam
laserf.cur	laser cooling force	Effects | Laser Cooling | Force
lum2t.cur	luminosity on time	Beam | Evolution | Luminosity
lumitest.cur	test calculation of luminosity	Effects | Collision point | Luminosity
nr2t.cur	particle number on time	Beam | Evolution | Number
shiftNNNcur	electron beam shifts	ECOOL | Cooler | Shifts
spaceN.cur	particles in phase space	Beam | Phase Space
th2t.cur	horizontal growth rate on time	Task | Growth Rates | Evolution
tn2t.cur	particle number growth rate on time	Task | Growth Rates | Evolution
tp2t.cur	longitudinal growth rate on time	Task | Growth Rates | Evolution
tv2t.cur	vertical growth rate on time	Task | Growth Rates | Evolution
txy2t.cur	evolution of emittance on momentum	Beam | Evolution | 3D Diagram
vdrift.cur	drift velocity of electron space charge	ECOOL | Cooler | Space charge
density.sur	distribution of electron beam density	ECOOL | Electron bunch | Array density
evolution.sur	evolution of profile on time	Beam | Distribution | Evolution
fflonf.sur	longitudinal electron cooling force	ECOOL | Draw forces | Longitudinal
fftr.sur	transverse electron cooling force	ECOOL | Draw forces | Transverse
ratedp.sur	longitudinal component of growth rates	Task | Growth Rates | Longitudinal
rateeh.sur	horizontal component of growth rates	Task | Growth Rates | Horizontal
rateev.sur	vertical component of growth rates	Task | Growth Rates | Vertical

II. Growth Rates Algorithm

Introduction

This procedure is intended for calculation of characteristic times of the beam r.m.s. parameter variation (growth rates) in accordance with few heating or cooling active effects. Characteristic times (τ_{hor}, τ_{ver}, τ_{lon}) are functions of all three emittances and particle number. The horizontal and vertical rates are determined in the program as

$$\frac{1}{\tau} = \frac{1}{\varepsilon}\frac{d\varepsilon}{dt},$$

where ε is the corresponding emittance. The longitudinal rate is determined as

$$\frac{1}{\tau} = \frac{1}{\sigma_p^2}\frac{d\sigma_p^2}{dt},$$

where σ_p is rms relative momentum spread. Rates have positive sign for a heating process and negative for cooling one. The negative sign of the lifetime (τ_{life}) corresponds to the particle loss and the sign of the lifetime can be positive in the presence of particle injection, when particle number increases.

In the program code this procedure is described in **void xDynamics::Rates()** (files xdynamic.cpp, xdynamic.h).

1. Starting the calculations, setting of parameters.

1.1. Command prompt mode

The simplest way to start a **Growth Rates** calculation **(GR)** is to put the **Betacool.exe** file, file of input parameters (*.bld format), other required files (from archived Betacool kit – download from website http://lepta.jinr.ru/betacool) and #SAVE.bat file into the same folder.

To start the **GR** one needs to type in the command line the following command:

<path>\betacool.exe <inputfilename> /rates

The parameter <inputfilename> is obligatory.

1.2. Windows interface

The interface part of the software consists of executable file Bolide.exe, *.dfm files containing information about BETACOOL exterior and input files for post processing of the calculated data.

The simplest way to start a work with **GR** is the following:
- to put the **Interface** files, Betacool.exe file, file of input parameters (*.bld format) and other required input files (from archived Betacool kit – download from website http://lepta.jinr.ru/betacool) to the same folder;
- to start Bolide.exe file, load the input file, check its validity, to import absent parameters from correct input file – see below.

- to start **GR** program using *TBrowse component* **Calculate** on the *Form* **Task | Growth Rates**.
- during the calculation the **Interface** program automatically reads the results from output files and represents them in numerical or graphical format to the corresponding windows.

The detailed description of the interface is presented in Bolide.doc.

2. Specification of the task. Calculation of sum of the rates

To provide a choice of effects acting on the ion beam distribution function one needs to specify the task using *menu item* **Task** *submenu item* **Growth Rates** (Fig.1).

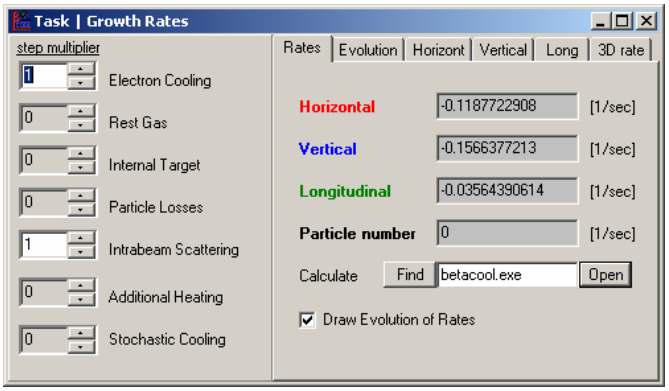


Fig.1. *Form* **Task | Growth Rates**

Here user must choose active effect by changing number in front of the effect name to arbitrary non-zero value. All the Effect classes have the same parent class **xEffect**, which has a Boolean variable `use`. When the corresponding *Counter* is non-zero this variable of the effect is **true**. A value of the step multiplier has effect for the Model Beam algorithm only. All the variables **Effect** are put in the array using self counter system. The program calculates sum of the rates in cycle calculating the rates of the effects at use = **true**. Parameters of each effect involved into calculation must be set in appropriate *Forms*. In addition user must prepare beam and ring parameters.

For the ion beam one should use the *Form* **Beam | Parameters (Fig. 2),** *TabSheet* **Emittance**.

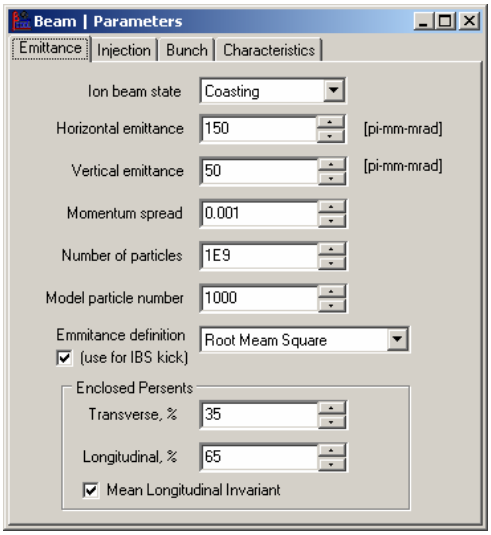


Fig. 2. **Beam | Parameters | Emittance**

Here only the following parameters must be set: **ion beam state**, **horizontal emittance**, **vertical emittance**, **momentum spread** and **number of particles**. More detailed description of this Form is given in **Beam Manual**.

For the ion ring user must set corresponding parameters in *Form* **Ring | Parameters** (fig. 3).

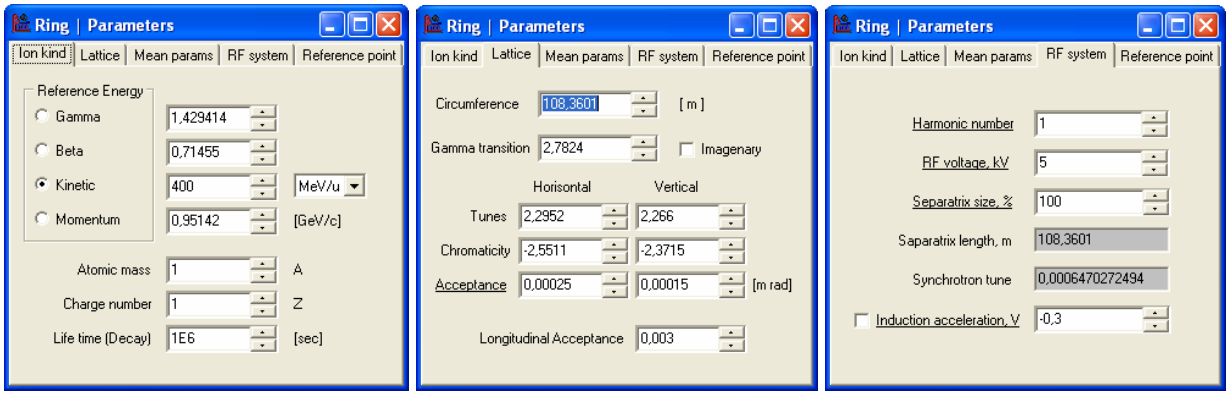


Fig. 3. **Ring | Parameters**.

Here the following parameters must be specified:
- on the *TabSheet* **Ion kind**: reference energy (depending on the presentation type), atomic mass, charge number, lifetime;
- on the *TabSheet* **Lattice**: circumference, gamma transition, horizontal and vertical tunes, chromaticities, acceptances, longitudinal acceptance;
- on the *TabSheet* **RF system**: harmonic number, RF voltage, separatrix size, induction acceleration (if needed).

More detailed description of this Form is given in **Ring Manual**.

Note: if IBS effect is chosen user must check how to specify Ring lattice. If **Piwinsky** model of IBS is chosen (*Form* **Effects | Inrabeam Scattering | IBS model**) user has nothing to do with ring lattice and setting parameters in **Ring | Parameters | Lattice** is enough. If another IBS model is chosen it is necessary to load a file with ring lattice structure. One has to call **Ring | Lattice Structure** and specify a file with lattice in *TabSheet* **Lattice Filename**, *TBrowse* **Output MAD filename**. For detailed description of IBS effect setting up please look **IBS Manual**.

By next step user must specify parameters of **GR** using *Form* **Task | Growth Rates** (Fig..1). If user wants to calculate only immediate rates – one has to use *TBrowse* component **Calculate** which starts the calculations. This *Form* contains four *edit windows* for representation of sum of the rates of corresponding rms beam parameter. In case of successive calculation the program does not generate any warning and stops the calculation process. The Betacool.war file contains only information about calculation time (Fig. 4).

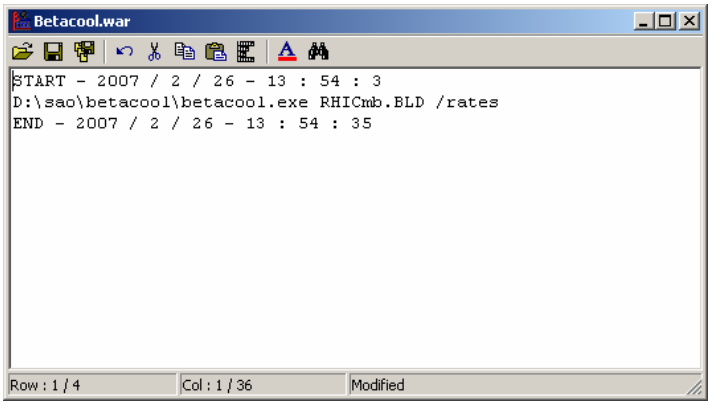


Fig. 4 Betacool.war file after successive calculation of the growth rates.

If user wants to calculate evolution of growth rates and to look over the dynamics of this process one must check *CheckBox* **Draw Evolution of Rates** (Fig.1) which switches ON/OFF drawing of **Evolution of Rates** plot (Fig.5) and run RMS dynamics calculation – **Task | RMS Dynamics**. Parameters of the evolution calculation (time step, etc) can be defined inhere.

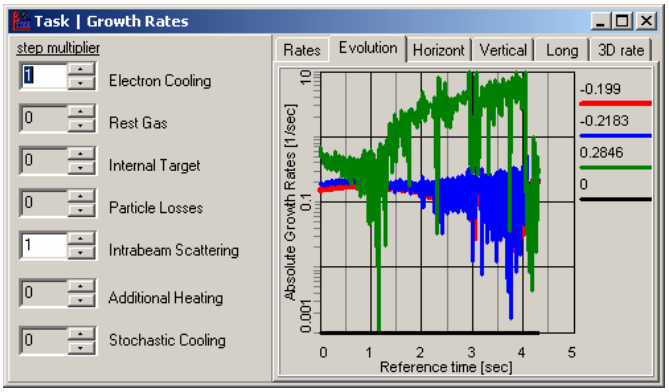


Fig. 5. **Task | Growth Rates | Evolution**.

To calculate 3D plots of the growth rates vs transverse and|or longitudinal emittances one should use *TabSheet* **3D rate** (Fig 6) which determines the parameters of the visualization. Here one can set minimal and maximal emittance value for plots and range division. The same parameters are specified for momentum spread.

To calculate and draw necessary 3D diagrams user must use *TBrowse* **Calculate 3D rate** (here Betacool is launched with 3d *parameter*).

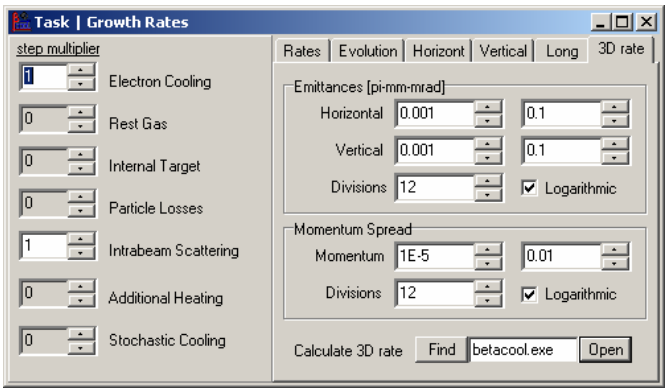


Fig.6. Window of the **Task | Growth Rates | 3D rate**.

3. Output files

As a result of **GR** Betacool modifies the input file (*.bld format) and creates a few files *.cur which contain time dependencies of the beam parameters.

Simultaneously results of **GR** are visualized to *TabSheets* of the current *Form:* 2-D plot of the function of Absolute Growth Rates over the time is on the **Evolition** *TabSheet*(Fig. 1.5)**,** and 3-D plots of the growth rates vs transverse and/or longitudinal emittances are on the *TabSheets* **Horizont, Vertical, Long** (Fig.7).

<parameter>	Procedure
rates	Sum of the rates of active effects Output into input file
3d	3D maps of sum of the rates in the range of the beam emittances specified in and momentum spread specified in Output: *rateeh.sur, rateev.sur, ratedp.sur*

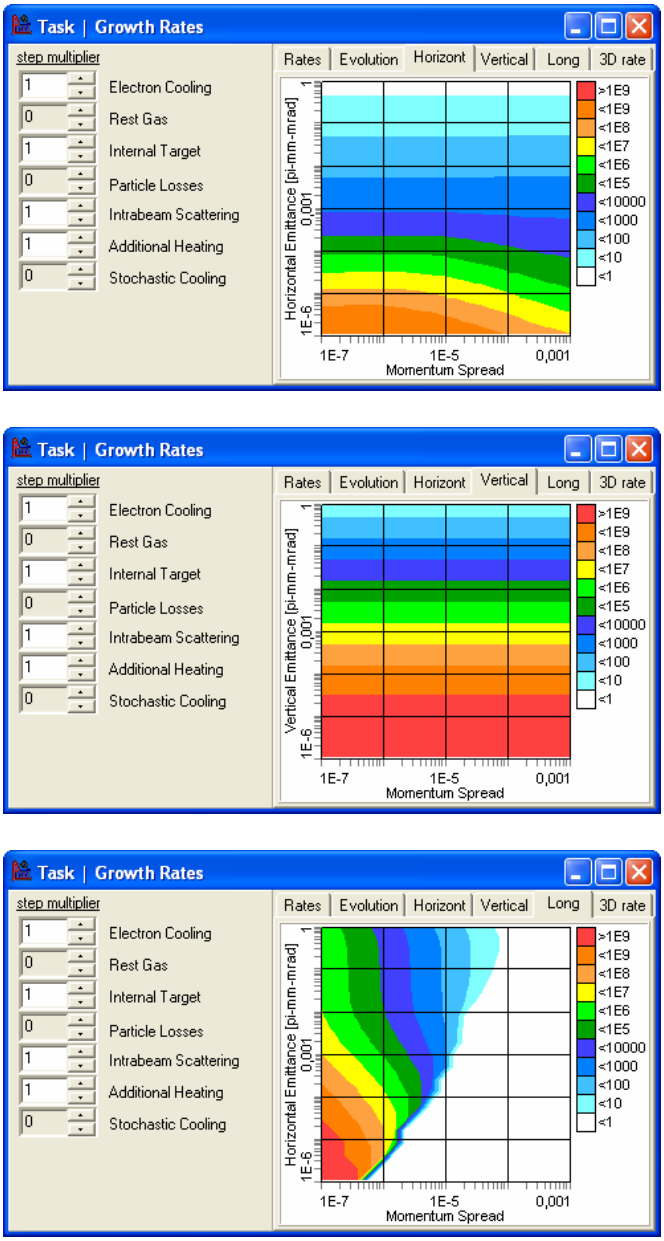


Fig. 7. Window of the Horizontal, Vertical and Longitudinal 3D Growth Rates.

4. Example of analysis of phase space diagrams of growth rates

The example of r.m.s. evolution during cooling process for HESR is presented on Fig.8. After cooling process all parameters achieved constant value and does not change for a long time. Particle loss rates are in a few orders less in comparison with cooling time and don't take into account during these simulations.

An unexpected behaviour of emittance can be explained with 3D phase space diagram (Fig.9). These diagrams presented dependence of growth rates on the momentum spread and horizontal emittance. The vertical emittance is assumed to be equal to the horizontal one.

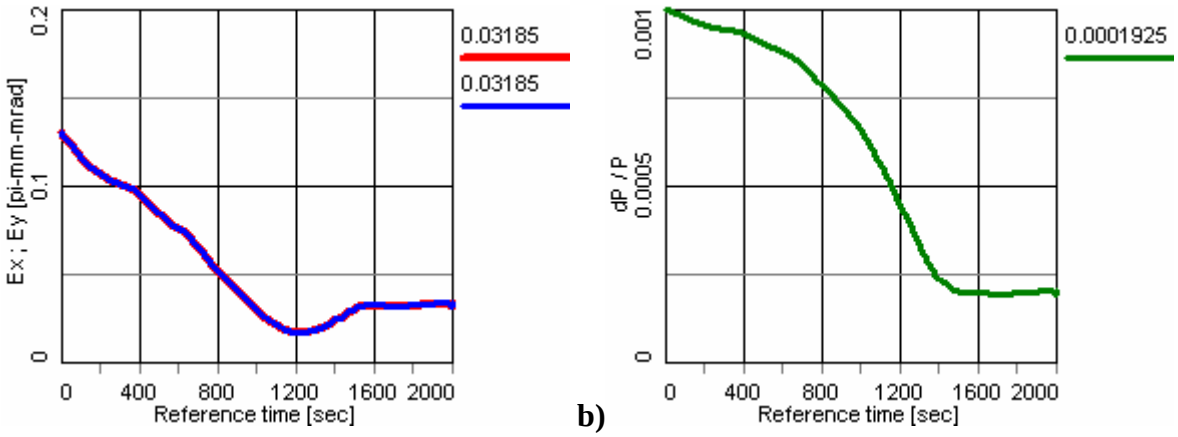


Fig.8. r.m.s. beam dynamics for HESR under action from IBS and ECOOL effects.
a) transverse emittances, b) momentum spread

IBS growth rates (Fig.9a,b) are calculated in accordance with Martini model. Colour areas indicate different values of growth rates. White area for longitudinal component means that in this region of beam parameters the momentum spread is decreased and emittances is increased. The temperature relaxation exists for large momentum spread and small emittance. Beam parameters due to IBS come to the equilibrium temperature between all degree of freedom. Cooling rates for EC (Fig.9c,d) are calculated in accordance with Parkhomchuk formula of cooling force. Transverse and longitudinal components of cooling rates have approximately the same behaviour.

Summary of cooling and heating rates are presented on Fig.9e,f. Boundaries between colour and white areas shows the equilibrium between IBS and EC for transverse and longitudinal components. Equilibrium point can be found if one overlaps these pictures each other (Fig.9g). Position of this point does not depend on initial coordinate. For very complicate pictures more then one equilibrium points can be found. In this case the equilibrium parameters can depend on initial values.

Fig.9g shows the dependence of the transverse emittance on the momentum spread during cooling process for r.m.s. dynamics on Fig.7. Initially the electron cooling force achieves the equilibrium with transverse component of IBS. During this process the emittance and momentum spread are decreased (evolution from start point on Fig.9). Then cooling process continues and beam parameters change in accordance with the equilibrium boundary of transverse component. Momentum spread continues to decrease but transverse emittance begins to increase.

When the cooling force also reaches the equilibrium with longitudinal component of IBS beam parameters achieve the equilibrium point, which does not depend on initial parameters (end point on Fig.9). r.m.s. dynamics is rather different and the cooling time can change very strong. It means that initial parameters of ion beam don't influence on the equilibrium point but they have a strong influence on the cooling time.

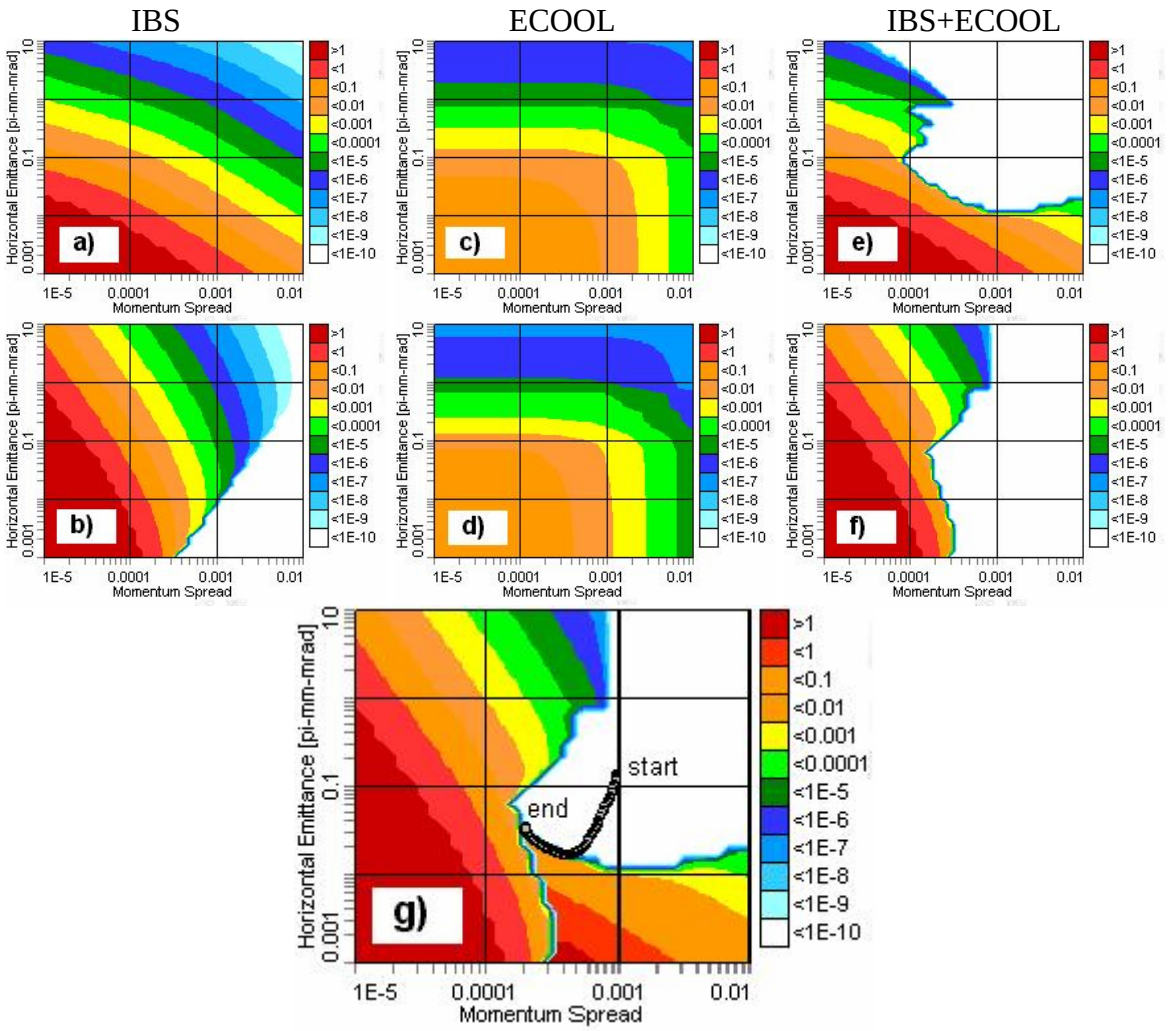


Fig.9. Phase space diagram of growth rates [sec^{-1}] for HESR.
a, b – transverse and longitudinal components of IBS growth rates, **c, d** – transverse and longitudinal cooling rates of EC, **e, f** – summary of cooling and heating rates, **g**– overlapping picture **f** over **e** and r.m.s dynamics in accordance to Fig.1.2.

III. RMS Dynamics

Introduction

RMS Dynamics is algorithm included in BETACOOL. This algorithm allows to analyze evolution in time of r.m.s. ion beam parameters under a common action of a few heating or cooling effects which are described in terms of characteristic times of the beam r.m.s. parameter variation. This model presumes Gaussian distribution of the ions in all degrees of freedom.

Step 1. Lunch Interface

Launch file bolide.exe. As a result the *MainForm* window is opened (Fig.1).

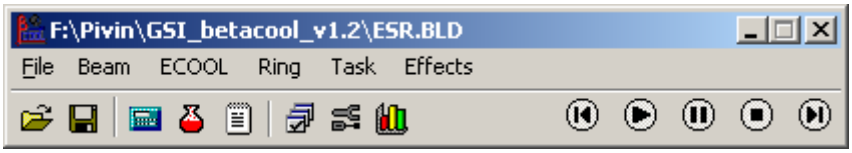


Fig.1. MainForm.

Step 2. Open file

Open file of input parameters (*.bld format).

Choose **File|Open** or  on the *MainForm* Fig.1 and choose file (Fig.2).

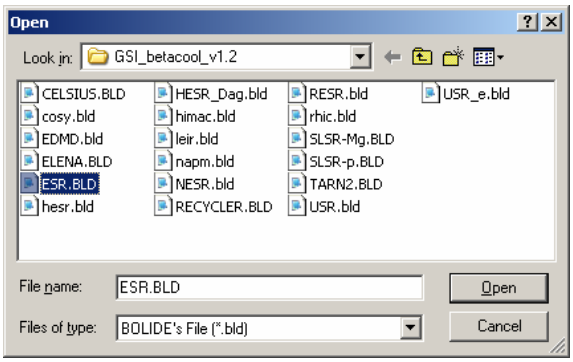


Fig.2. File open window

Step 3. Setting of the ring parameters

One needs to choose *MenuItem* **Ring|Parameters (**Fig.3).

It's necessary to set **Reference Energy** on *TabSheet* **Ion kind (**Fig.3). Energy can be set in four ways:

- Lorenz factor γ (**Gamma**);
- Particle velocity in the units of the speed of light (**Beta**);
- Kinetic energy (**Kinetic**);
- Particle momentum in GeV/c (**Momentum**).

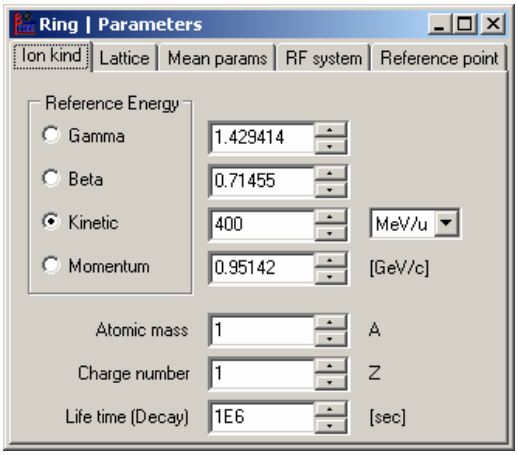
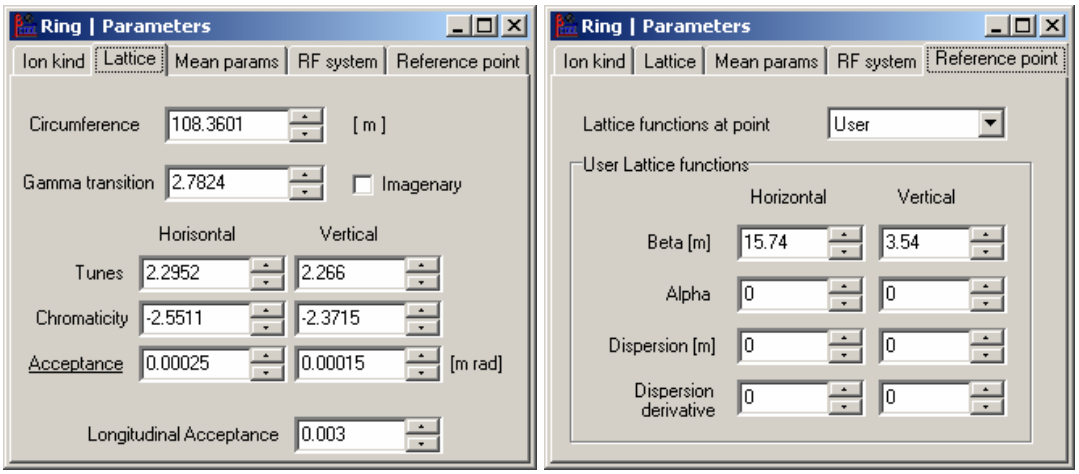


Fig.3. Ring parameters.

Also it's necessary to set up **Atomic mass**, **Charge number** and **Life time**. For references the parameters of chemical elements are collected in the periodic table – M*enuItem* **File | Periodic Table** or corresponding *Button*  in the tool bar.

Using the *TabSheet* **Lattice** (Fig.4a) set up: **Circumferensce, Gamma transition, Tunes, Chromaticity, Acceptance and Longitudinal Acceptance**.



a) b)

Fig.4. Ring parameters

Using the *TabSheet* **Reference point** Fig.4b set up: **Lattice functions at point** and **User Lattice functions**.

<u>Step 4.</u> Setting of lattice structure

If **IBS** effect is included into calculation, depending on the IBS model one has to set ring lattice structure. Piwinski model needs only mean ring lattice, so it is not necessary to set this *Form*. Other models need real (or reduced) lattice structure of the ring. The ring lattice structure can be imported from input or output MAD file. Betacool translator does not support all the possibilities of MAD input file syntax and sometime it is necessary to modify the input file manually. Therefore for IBS rate calculation it is easier to use output MAD file. To provide a choice of the lattice file name and its specification the *MenuItem* **Ring|Lattice Structure** and corresponding visual form (Fig.5) are used.

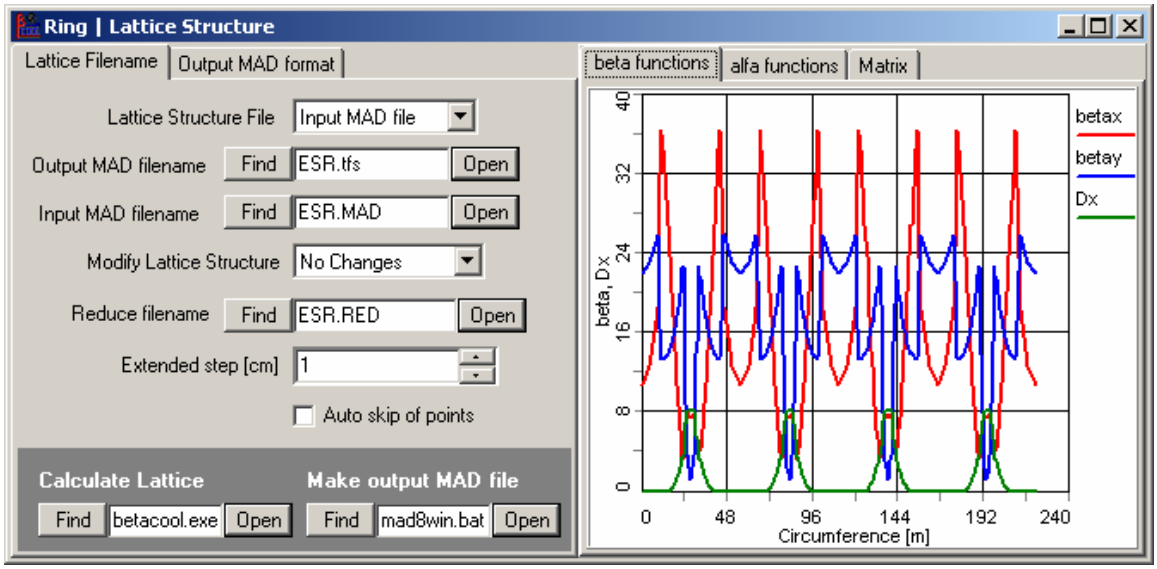


Fig.5. Ring lattice structure.

In the *TabSheet* **Lattice Filename,** *ComboBox* **Lattice Structure File** the option **Output MAD file** has to be chosen. For IBS calculations using Martini or Jei Wei models one needs to find appropriate lattice structure file. Button **Find** of the *TBrowse* component opens the file manager window. Button **Open** opens the file using internal text editor. The chosen file name is indicated in the edit window of the *TBrowse* component and saved in the input BETACOOL file. This name is used for initialization of the ring structure after start of the program. Validity of the file can be checked using *TBrowse* component **Calculate Lattice.** Button **Open** of this *TBrowse* component starts BETACOOL with the parameter /lattice. At this parameter BETACOOL reads MAD output file, transforms lattice parameters into internal format and saves them into the files BetaX.cur, BetaY.cur, DispX.cur, AlfaX.cur, AlfaY.cur, DispX_.cur. During this procedure the program checks validity of the data in all positions of lattice structure file in accordance with the description tacking from the *TabSheet* **Output MAD format.**

<u>Step 5.</u> Setting of the beam parameters

One needs to choose *MenuItem* **Beam|Parameters (**Fig.6). Use **Emittance** *TabSheet* to set up: **Ion beam state**, **Horizontal emittance**, **Vertical emittance**, **Momentum spread** and **Number of particles**. **Model particle number** is used for Model Beam algorithm only.

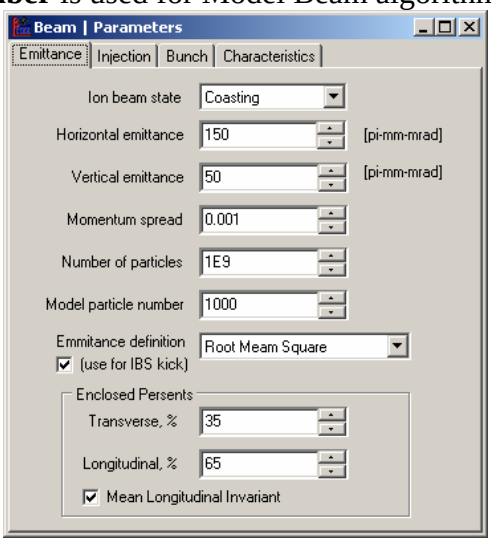


Fig.6. Beam parameters.

Step 6. Setting the active effects

To calculate sum of the rates for a few active effects the *MenuItem* **Task|Growth Rates** are used. Left part of the corresponding visual form contains the list of the effects that can be used in simulations. An effect is switched ON in simulations when the corresponding **step multiplier** parameter has value equal to 1. In the (Fig.7) an example of the task specification, when the **Electron cooling**, **Internal Target** and **Intrabeam Scattering** effects are active, is presented. For more details about each effect one has to look their description (IBS manual, Restgas manual, Target manual et all).

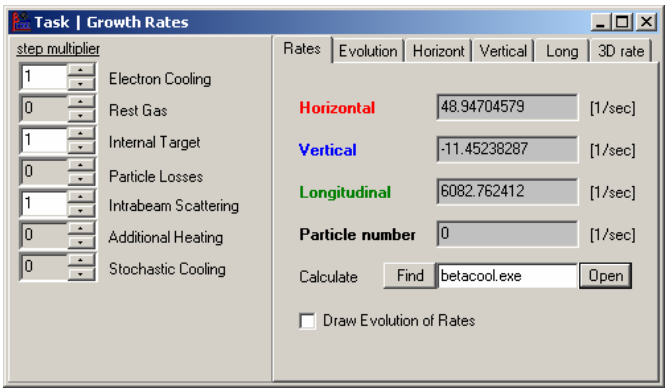


Fig. 7. Effect switching list.

Step 7. Output results

Evolution of the beam parameters such as emittances, momentum spread and particle numbers during simulation can be observed on *TabSheets* **Emittance**, **Momentum** and **Number** of *MenuItem* **Beam|Evolution** (Fig.8). Instructions for working with the 2D plot are presented in the file Bolide.doc.

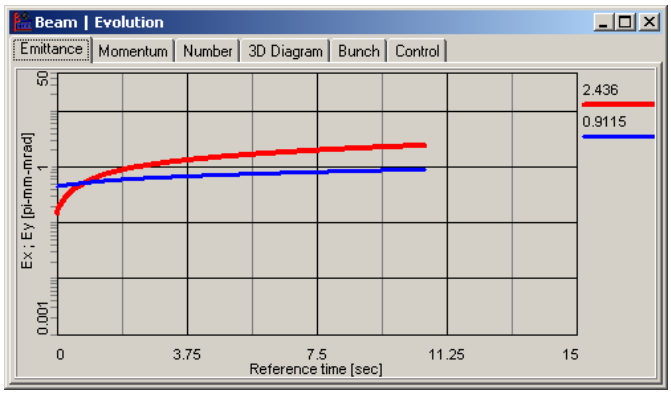


Fig.8. Beam parameters evolution.

Set up a list of observed parameters is presented on *TabSheets* **Control (**Fig.9).

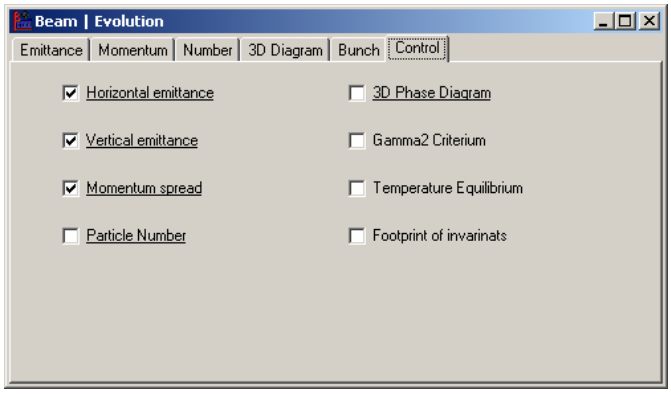


Fig.9. Control of the evolution plots.

Step 8. The starting of RMS Dynamics simulation

To start simulation one needs to choose *MenuItem* **Task|RMS Dynamics** (Fig.10)

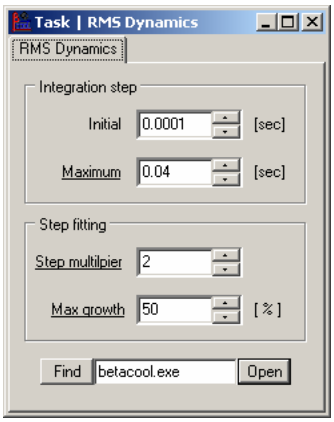


Fig.10. RMS dynamics parameters.

It's necessary to choose **Initial** and **Maximum** step in seconds on this *Form.*
Integral step determinates the initial step of simulation in time. **Maximum** step sets limit for the upper step growing. Changing of the step is defined by **Step multiplier** and **Max growth**, where:
- **Step multiplier** is the value of step increment.
- **Max growth** is the max difference between previous and next values of calculated beam r.m.s. parameters.
- Push *Button* **Open** to begin simulation.

IV. Model Beam Algorithm

Introduction

Model Beam (MB) algorithm was developed on the base of SIMCOOL code which was developed by Novosibirsk group. This algorithm uses a few thousands of test particles with arbitrary distribution. The action from IBS on the each test particle is calculated from the current distribution of test particles. This algorithm can reach a good accuracy when the distribution of test particle is closed to Gaussian. Some modification of this method was made for simulation of IBS in the case of non Gaussian distribution.

Evolution of the ion distribution function is described by the Fokker-Plank equation. Friction and diffusion terms in the general case depend on the distribution function. However in some cases, when the effects acting on the distribution function do not lead to change of its shape, the Fokker-Plank equation can be reduced to equation for the second order moments of the distribution function.

In general case the Fokker-Plank equation can be reduced to Langevin equation in invariant or momentum space. The **Model Beam** algorithm realizes solution of Langevin equation in momentum space using Monte Carlo method. In the frame of this algorithm the ion beam is presented as a particle array. Each particle is presented as a 6 co-ordinate vector: $\vec{X} = \left(x, \frac{p_x}{p}, y, \frac{p_y}{p}, s - s_0, \frac{\Delta p}{p} \right)$, where x and y are the horizontal and vertical co-ordinates, p_x and p_y are corresponding momentum components, s-s_0 is the distance from the bunch center (in the case of coasting beam – distance from a reference particle), Δp is the particle momentum deviation from momentum of reference particle p. Action of each effect is simulated as the particle momentum variation in accordance with the following equation:

$$\left(p_{x,y,s} / p\right)_{fin} = \left(p_{x,y,s} / p\right)_{in} + \Lambda_{x,y,s}\Delta T + \sqrt{D_{x,y,s}\Delta T}\,\xi_{x,y,s},$$

where p_s is the particle longitudinal momentum deviation, subscript *in* correspond to initial momentum value, subscript *fin* relates to final particle momentum after action of the effect, Λ and D are the drift and diffusion terms for corresponding degree of freedom, ΔT is step of the integration over time, ξ is Gaussian random number at unit dispersion.

Calculation scheme of the Model Beam algorithm
The basic scheme of the algorithm is the following:

- on the first stage a beam is generated with defined parameters;
- in the selected point of the ring in accordance to the current lattices so called “kicks” from active effects are calculated (coordinates and angles of every particle are changed correspondingly).
- obtained vector of coordinates is multiplied **Turn step** times by the transformation matrix of the whole ring;
- go to the first step;

Below a step-by-step instruction for setting and using this algorithm is given.

Step 1. Lunch interface

Launch file bolide.exe. As a result the *MainForm* window is opened (Fig.1).

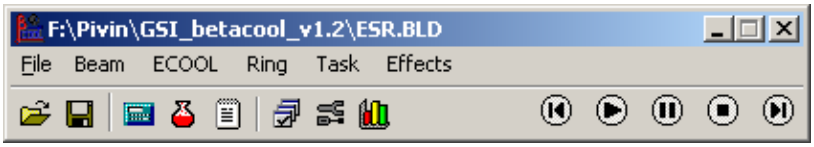


Fig.1. **MainForm**.

Step 2. Open file

Open file of input parameters (*.bld format). Choose **File|Open** or  on the *MainForm* (Fig.1) and choose file (Fig.2).

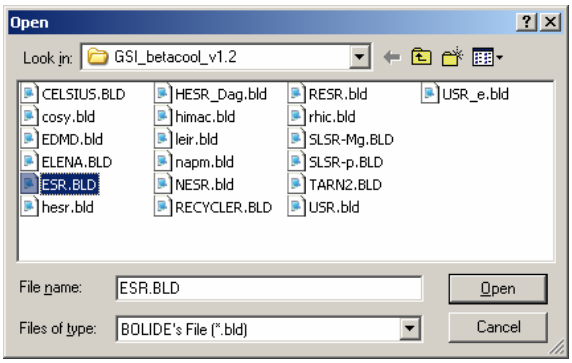


Fig.2. **File open** *window*.

Step 3. Setting of the ring parameters.

One needs to choose *MenuItem* **Ring|Parameters (**Fig.3). It is necessary to set **Reference Energy** on *TabSheet* **Ion kind (**Fig.3). Energy can be set in four ways:
- Lorenz factor γ (**Gamma**);
- Particle velocity in the units of the speed of light (**Beta**);
- Kinetic energy (**Kinetic**);
- Particle momentum in GeV/c (**Momentum**).

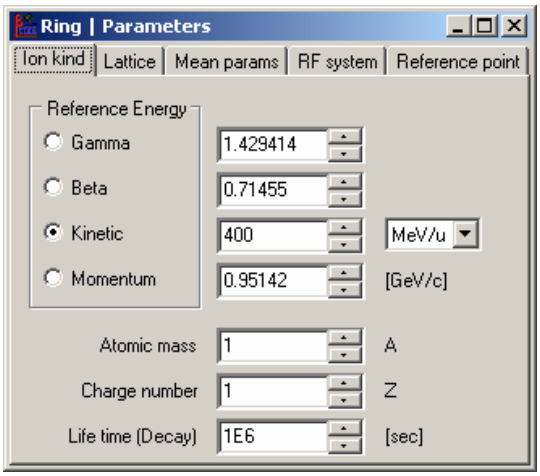


Fig.3. Ring parameters.

In addition it is necessary to set up **Atomic mass**, **Charge number** and **Life time**. For references the parameters of chemical elements are collected in the periodic table – M*enuItem* **File | Periodic**

Table. If the beam is bunched one needs to set RF **harmonic number** and **amplitude** of the RF voltage in the *TabSheet* **RF system** (Fig. 4).

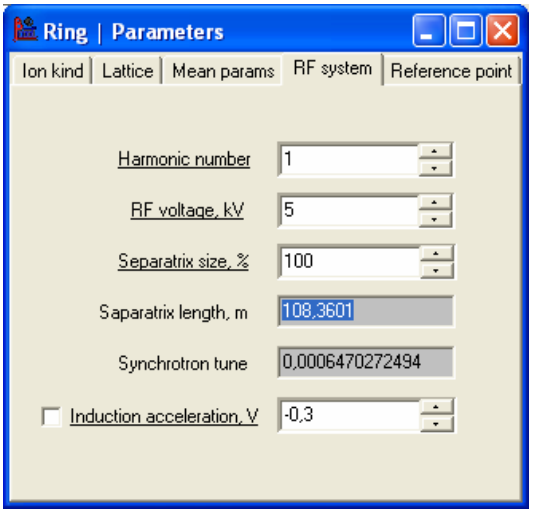
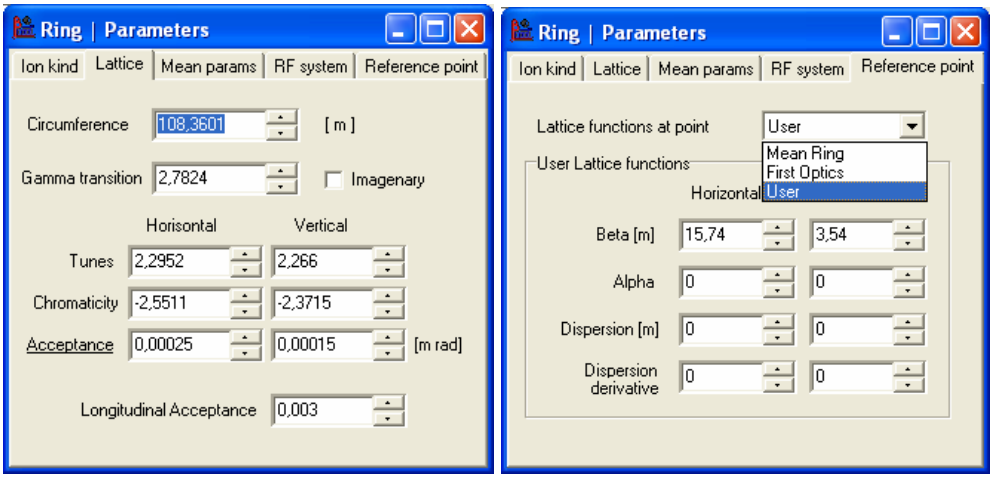


Fig. 4. *Form* **Ring | Parameters**. *TabSheet* **RF system**.

The **Separatrix size** parameter is used for the particle loss simulation and it can be arbitrary positive number. **Separatrix length** and **Synchrotron tune** are output parameters. The check box **Induction acceleration** has to be not checked.

Using the *TabSheet* **Lattice** (Fig.5a) set up: **Circumferensce, Gamma transition, Tunes, Chromaticity, Acceptance and Longitudinal Acceptance**. These characteristics of ring are necessary for numerical calculations. At Fig. 5a an example of ESR ring lattice is shown.



a) b)

Fig.5. Ring parameters.

Using the *TabSheet* **Reference point** Fig.5b set up: **Lattice functions at point** and **User Lattice functions**. This "Reference point" means the point (characterized with lattice) where turn over the ring will start for model beam simulation (matching the particle array with the ring optic structure). Depending on the user choice (*ComboBox* **Lattice functions at point**) it can be either mean ring lattice (**mean ring** – here program automatically takes averaged calculated lattice – see *TabSheet* **Mean params**) or **first optics** (here program counts that reference point is first optic element from the MAD lattice structure) or **user-defined** (here user must indicate lattices for the reference point – see *Edit windows* on the current *TabSheet* below).

Step 4. Setting of ring lattice

If **IBS** effect is included into calculation, depending on the IBS model one has to set ring lattice structure. All other effects either use lattice in their location (defined in the effect parameters) or use lattice in the **Reference point**.

IBS Piwinski model needs only mean ring lattice, so it is not necessary to set this *Form*. Other models need real (or reduced) lattice structure of the ring. The ring lattice structure can be imported from input or output MAD file. Betacool translator does not support all the possibilities of MAD input file syntax and sometime it is necessary to modify the input file manually. Therefore for IBS rate calculation it is easier to use output MAD file. To provide a choice of the lattice file name and its specification the *MenuItem* **Ring|Lattice Structure** and corresponding visual form (Fig.6) are used.

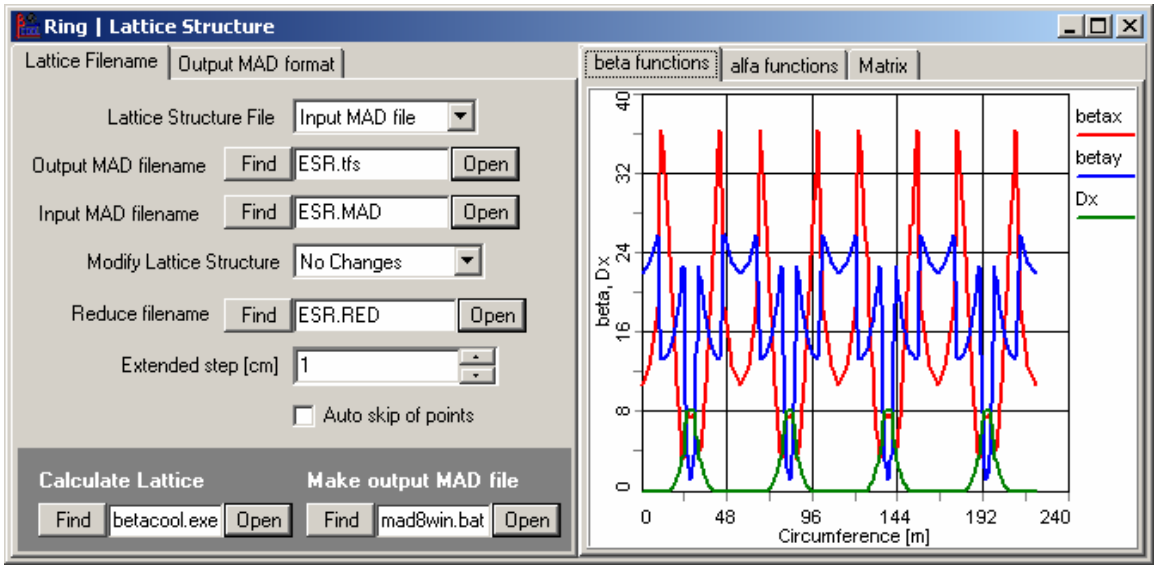


Fig.6. Ring lattice structure.

In the *TabSheet* **Lattice Filename,** *ComboBox* **Lattice Structure File** the option **Output MAD file** has to be chosen. For IBS calculations using Martini or Jei Wei models one needs to find appropriate lattice structure file. Button **Find** of the *TBrowse* component opens the file manager window. Button **Open** opens the file using internal text editor. The chosen file name is indicated in the edit window of the *TBrowse* component and saved in the input BETACOOL file. This name is used for initialization of the ring structure after start of the program. Validity of the file can be checked using *TBrowse* component **Calculate Lattice.** Button **Open** of this *TBrowse* component starts BETACOOL with the parameter /lattice. At this parameter BETACOOL reads MAD output file, transforms lattice parameters into internal format and saves them into the files BetaX.cur, BetaY.cur, DispX.cur, AlfaX.cur, AlfaY.cur, DispX_.cur. During this procedure the program checks validity of the data in all positions of lattice structure file in accordance with the description tacking from the *TabSheet* **Output MAD format.**

For more detailed information about setting Ring parameters please see [RING Manual].

Step 5. Setting beam parameters.

One needs to choose *MenuItem* **Beam|Parameters** (Fig.7). Use **Emittance** *TabSheet* to set up: **Ion beam state**, **Horizontal emittance**, **Vertical emittance**, **Momentum spread** and **Number of particles**. Here number of particles means number in the main (not modeled) beam and this number is used for particle losses calculation (evolution of beam particle number). **Model particles**

number we recommend to set it not less then 1000, but the time of calculation depends proportionally on the number of particles, both as accuracy. Model particles will be generated in concordance with indicated emittance values and type of **Emittance definition**.

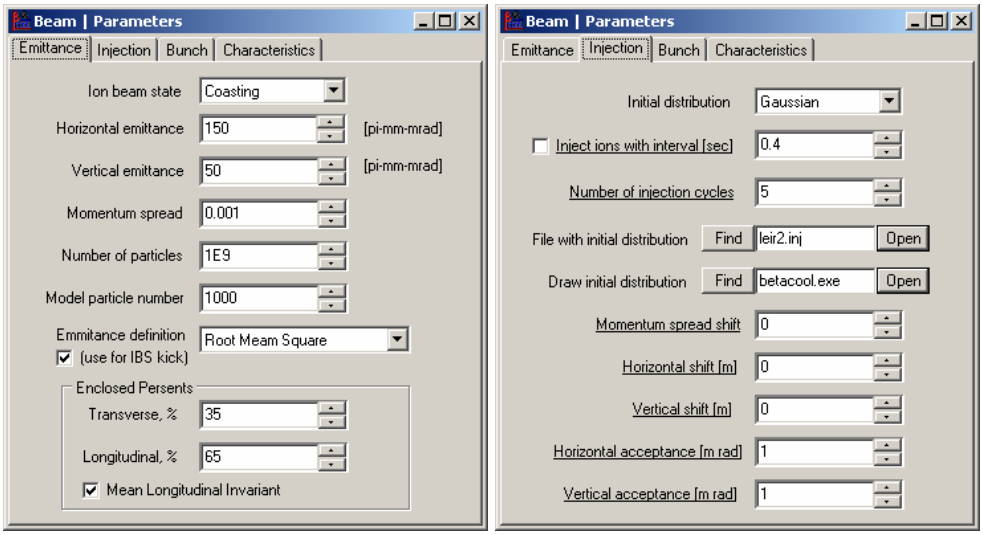


Fig.7. Beam | Parameters | Emittance & Injection.

One has to define beam emittance representation. Go to **Beam | Parameters | Emittance** and choose **Emittance definition** (Fig.7). Here 4 types of emittance definition are proposed: **Root Mean Square** (usual rms emittance – when 1 sigma of Gaussian distributed particles is presumed), **Courant Snyder** (when emittance is calculated as Courant Snyder invariant), **Full Width on Half Maximum** (emittance corresponded to particles inside full width on half maximum of distribution), **Enclosed Percents** (emittance occupied by the indicated percent of beam particles). **Use for IBS kick** – means the using the choosing emittance definition for the intrabeam scattering kick, otherwise, the **Root Mean Square** definition is used for the intrabeam scattering kick.

If one chooses **Emittance definition** as **Enclosed Percents**, it is necessary to add value of **Enclosed Percents** for **Transverse** and **Longitudina**l phase space (Fig. 7). **Mean Longitudinal Invariant** parameter has to be switched on in the case when the momentum spread has some deviation from the reference energy.

If multi-injection process is used for simulations user can use *TabSheet* **Beam | Parameters | Injection** (Fig. 7). Here necessary to switch ON the process (*Checkbox* **Inject ions with interval**) and to define injection repetition period (**Inject ions with interval**), **number of injection cycles**, indicate file with pre-generated array of injected particles, Shift of longitudinal momentum (**Momentum spread shift**), Shift of horizontal and vertical positions of the injected beam (**Horizontal shift, Vertical shift**), Horizontal and Vertical acceptances at injection (**Horizontal acceptance, Vertical acceptance**).

To adjust RF system parameters to the required bunch length (in the case of bunched beam) the TabSheet **Bunch** can be used (Fig. 8). This tab sheet contains one input parameter – **Number of bunches**. This parameter is necessary for luminosity calculation only and in IBS rate calculation it does not play a role. It can be arbitrary positive non zero number. After starting Betacool the program outputs rms bunch length corresponding to relative momentum spread of the bunch and RF system parameters. By a few subsequent runs of simulations one can adjust RF voltage (or harmonic number) to have required rms bunch length.

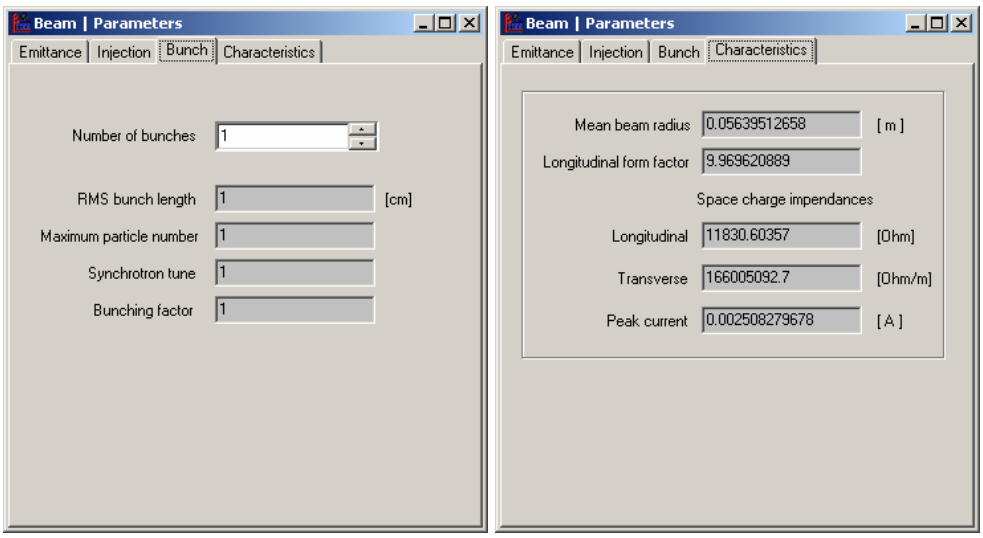


Fig.8. Beam | Parameters | Bunch & Characteristics.

The *TabSheet* **Characteristics** (Fig. 8) includes beam parameters calculated when main parameters of the beam and ring are determined.

For more detailed information about setting Beam parameters please see **Beam Manual**.

<u>Step 6.</u> Setting the active effects and starting the calculations.

To switch ON active effects into simulation the *MenuItem* **Task|Growth Rates** is used. Left part of the corresponding visual form contains the list of the effects that can be used in simulations. An effect is switched ON in simulations when the corresponding **step multiplier** parameter has integer non-zero value. If this value is integer positive it defines how many times the effect will be

In the (Fig.9) an example of the task specification, when the **Electron cooling**, **Internal Target** and **Intrabeam Scattering** effects are active, is presented. For more details about each effect one has to look their description (IBS Manual, Rest Gas Manual, Target Manual et al).

Any integer value of **step multiplier** means that this effect is active. Positive value more then unit – means that this effect skip a few integration steps in accordance with **step multiplier** and applies the kick one time per a few integration steps. Negative value – means that this effect applies a few kicks per one integration step. This possibility can be very useful if the different effects have a different calculation time or different values of kicks.

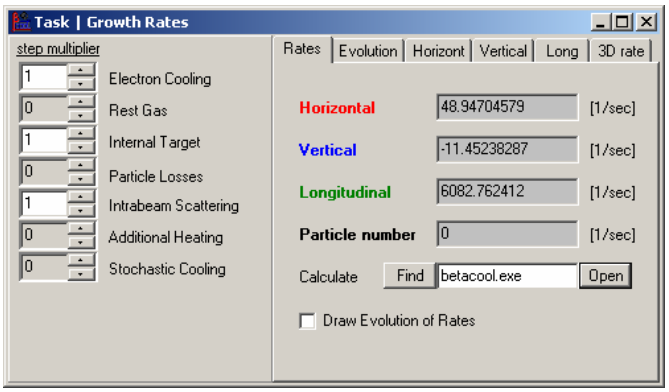


Fig. 9. Effect switching list.

Next step is setting the calculation parameters. At the *Form* **Task | Model Beam | Model Beam** (Fig. 10) one has to set up **Integration step**. There are two possibilities to do this. User can set either **Time** step in seconds or the step in number of turns (**Turn number)**. **Stop time** indicates the

reference time when the simulation will finish automatically (zero value means infinity calculation). In the case of **Common betatron tune** all model particles have the same value of the random phase advance on each integration step, otherwise, each particle gets the different phase advance. For the synchrotron tune user can choose random value (**Random synchrotron tune**) or synchrotron tune will be calculated from ring parameters.

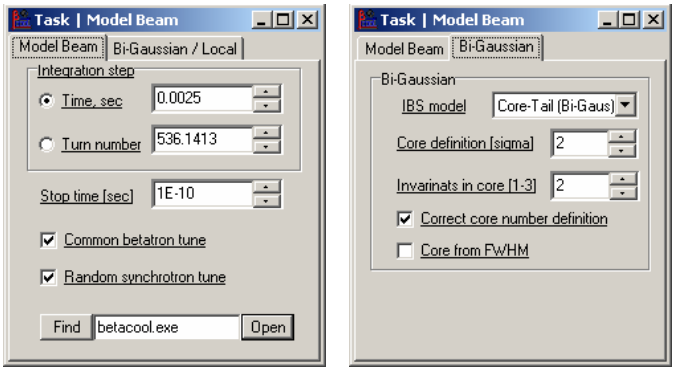


Fig. 10. *Form* **Task | Model Beam**.

After that one should choose **IBS kick model** at the *Form* **Task | Model Beam | Bi-Gaussian** (Fig. 10). *ComboBox* **IBS model** gives user a choice of model for kick applying on particles: **Gaussian** (momentum deviation is distributed on particles in accordance with Gaussian law), **Bi-Gaussian** (particle beam distribution is assumed as sum of two Gauss distributions – one for core another for tails, momentum deviation is distributed on particles in accordance with two independent "Gausses" and area where they are overlapped), **FWHM** (momentum deviation is distributed on particles assuming Gaussian distribution with determined FWHM). The detailed description of these models one can find in **IBS Physics Guide**.

Core definition describes how many particles in the rms sigma units are defined as core particles. **Invariants in core** is describes how many invariants (horizontal, vertical and longitudinal) are used for the definition of the particles in core. User can set the **Correct core number** in the accordance with **Invariants in core**, otherwise, the particle number in the core the will be used from Bi-Gaussian or FWHM interpolation. **Core from FWHM** – means that the core sigma always is defined from FWHM method, anyway the tail sigma is defined from choosing **IBS model**.

When all parameters are defined user must push "Open" at **Task | Model Beam | Model Beam** to start simulation (Fig. 12).

Step 7. Taking into account particle losses.

If one plans to use **Particle Losses effect** go to **Effects|Particle Losses** (Fig.11) and choose Generate model particles on losses with distribution. It allows to recover lost particle in the model beam. BETACOOL operates with two number of particles (beams) – the first number (major) – it contains real number of particles. The second one – is model beam – it usually contains a few thousands particles and is used for dynamics simulation in accordance with active effects. When particle is lost due to some factors we have to decrease number of particles in the major beam, but the number of particles in the model beam has to stay unchanged.

The *ComboBox* **Generate model particles on losses with distribution** allows to make a choice how lost particles will be recovered in the model beam in accordance with defined distribution from the list: **Gaussian** (particles will be regenerated by Gauss), **Real** (particle will be regenerated according to existing distribution), or **None** (model particles don't re-generated and its value decrease). If **Off** is chosen – then the model particles don't loss but anyway the number of major particle decrease (for example - decay process).

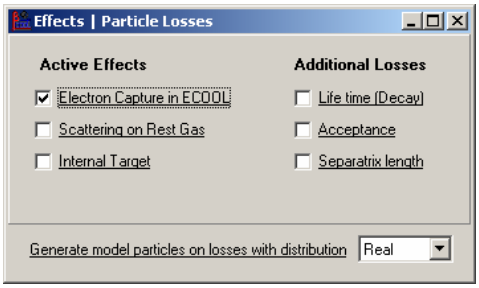


Fig. 11. *Form* **Effect | Particle losses**.

<u>Step 8</u>. Output results.

Results of the beam dynamics simulation are output in the following main *Forms*:
- **Beam | Distribution**;
- **Beam | Evolution**;
- **Beam | Real space**.

Form **Beam | Distribution** (Fig. 12) includes the following *TabSheets* for results: **Coordinate** (dependence of particle number (in percent) on momentum deviation and coordinate normalized on corresponding rms parameter and particle number), **Profile** (real particle distribution for every coordinate (analogue of previous plot) averaged on betatron or synchrotron oscillations), **Invariant** (particle number (in percent) which occupies corresponding emittance - invariants) and **Evolution** (3D plot for saving of Coordinate or Invariants evolutions in time). *TabSheet* **Control** (Fig. 14) provides settings for plot on all *TabSheets* listed above. As on every plot window user can choose which plots he wants to be redrawn on-line. More detailed description is given in **Beam Manual**.

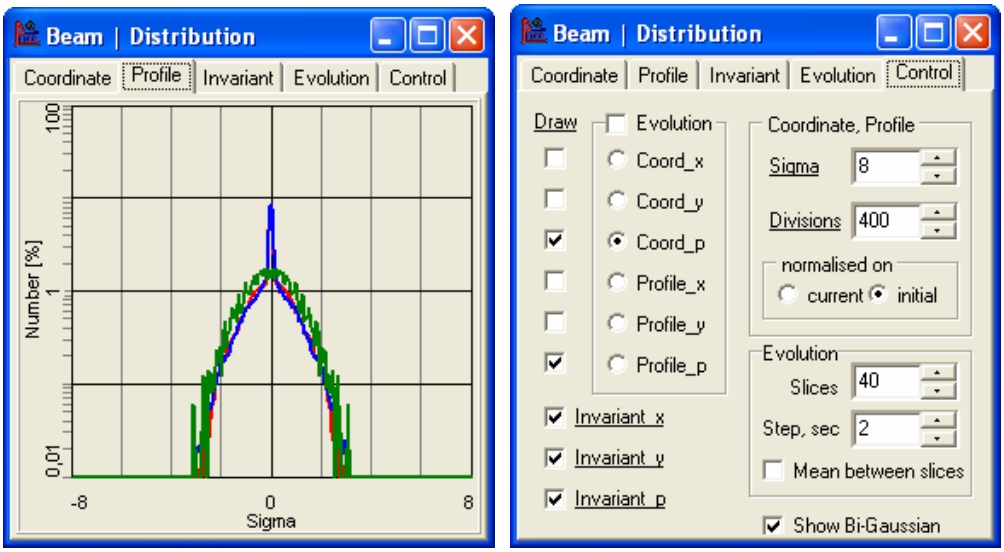


Fig. 12. *Form* **Beam | Distribution**.

Form **Beam | Real Space** (Fig. 13) includes 4 TabSheets used for visualization of the particle distribution in different planes during simulation procedure and a special *TabSheet* **Control** which is a manager of plots – user can choose which plot is active for on-line redrawing and what a coordinate plane will be represented on it (for example **X-Y** – transverse real space of particles, **X-X'** – horizontal phase space, **X-dP/P** – longitudinal real space, **S-So-dP/P** – longitudinal phase space). More detailed description is given in **Beam Manual**.

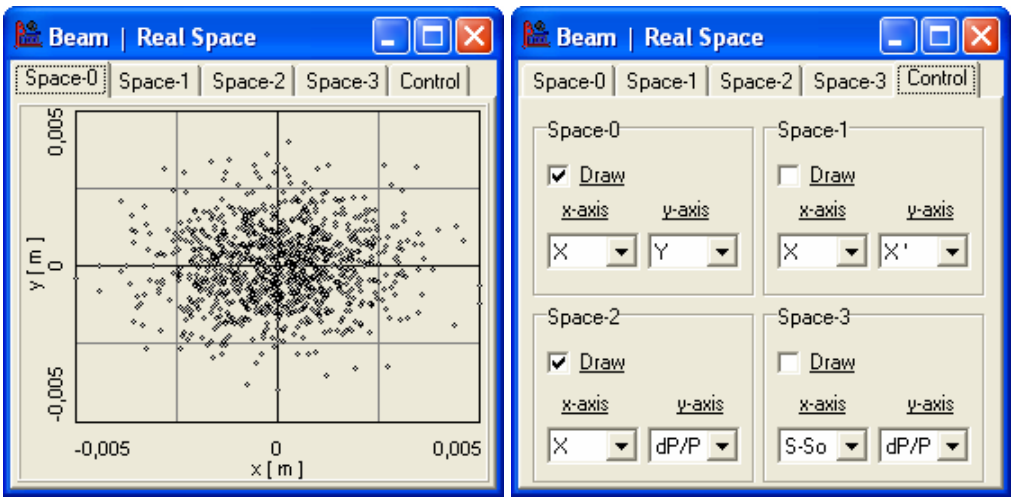


Fig. 13. *Form* **Beam | Real Space**.

Form **Beam | Evolution** (Fig. 14) is intended for visualization of beam parameters evolution during simulations and includes the following *TabSheets* for time dependencies of: horizontal and vertical emittances (**Emittance**), momentum spread (**Momentum**), particle number (**Number**), horizontal emittance on the momentum spread (**3D Diagram**), bunch length (**Bunch**). *TabSheet* **Control** allows user to manage the visualization. It contains list of all the plots of the current *Form* which correspond to their *Checkboxes*. User must check plots which are needed to be redrawn on-line and (or) check out other plots. More detailed description is given in **Beam Manual**.

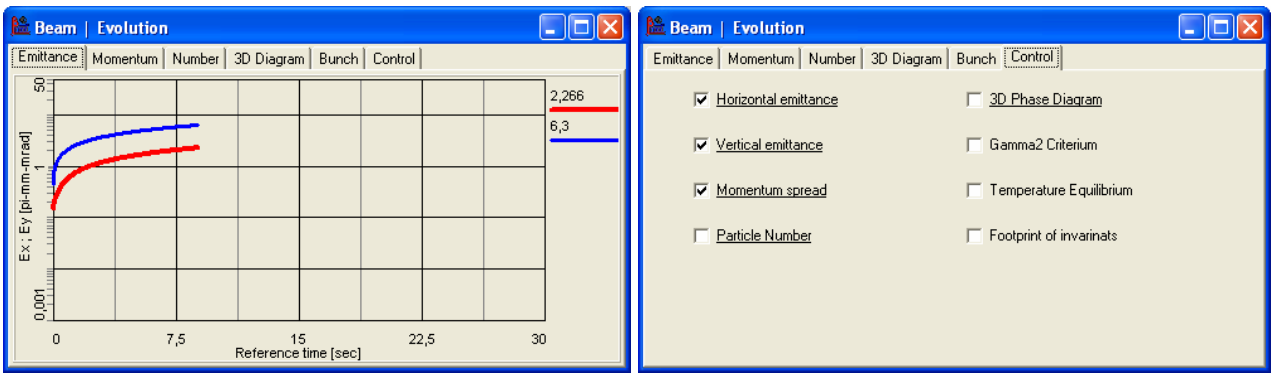


Fig. 14. Window of the **Beam | Evolution** menu item.

Note: *Instructions for working with the 2D and 3D plots are presented in the file Bolide.doc.*

V. Particle Losses

One needs to choose *MenuItem* **Effects|Particle Losses** on the *MainForm* Fig.1.

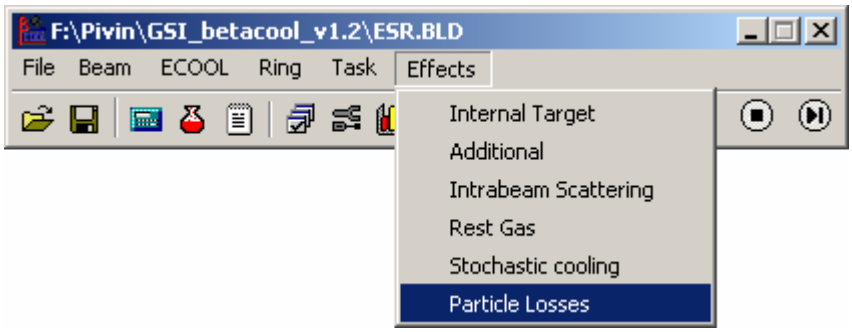


Fig.1 *Main Form*

Window of the **Effects | Particle Losses** allows you to activate effects which will influence on particle number in the beam (Fig.2).

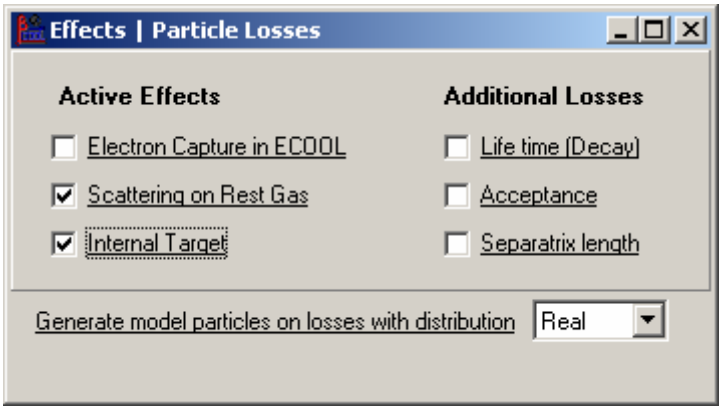


Fig. 2 **Particle Losses** options.

One has to set additional parameters (Fig. 3) of losses if **Scattering on Rest Gas** or **Internal Target** were chosen.

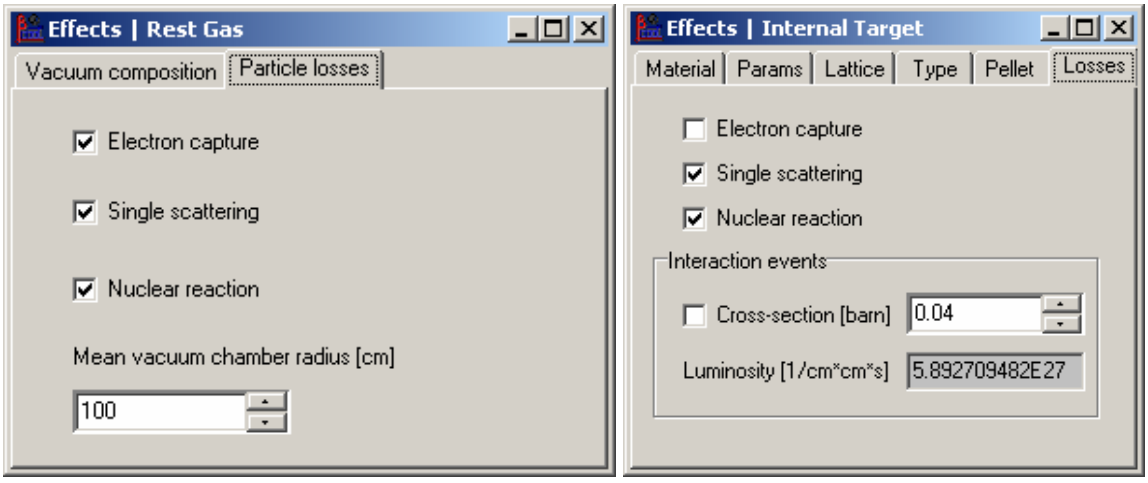


Fig. 3 Particle losses in **Rest Gas** and **Internal Target**

Also one has to choose how lost particles will be recovered in the test beam in accordance with defined distribution from the list: (Fig. 4).

There are 4 options:

- **Gaussian** – re-generation according to the Gauss distribution;
- **Real** – re-generation according to the real distribution;
- **None** – lost particles will be not recovered in model array but will reduce whole number of particle in the beam;
- **Off** – model particles don't lost (for example – beam decay) but will reduce whole number of particle in the beam.

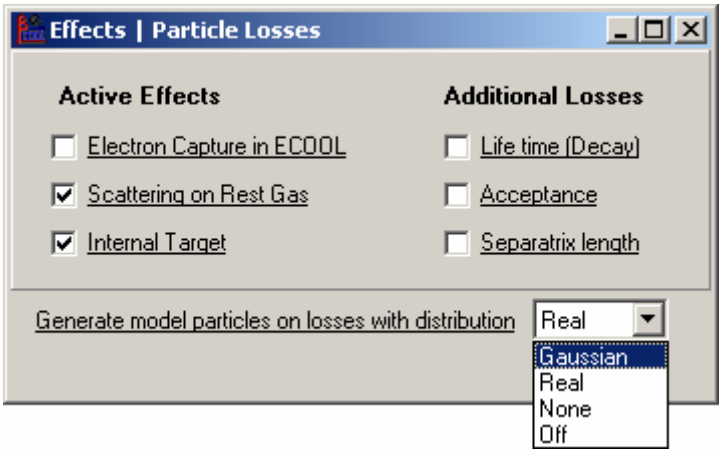


Fig.4 **Particle Losses** *form*

Finally one has to switch ON all other used effects (Fig. 5). User can push *Open* on the *Form* **Task | Growth Rates | Rates** to check in **Particle number** window the growth rates due to particle losses.

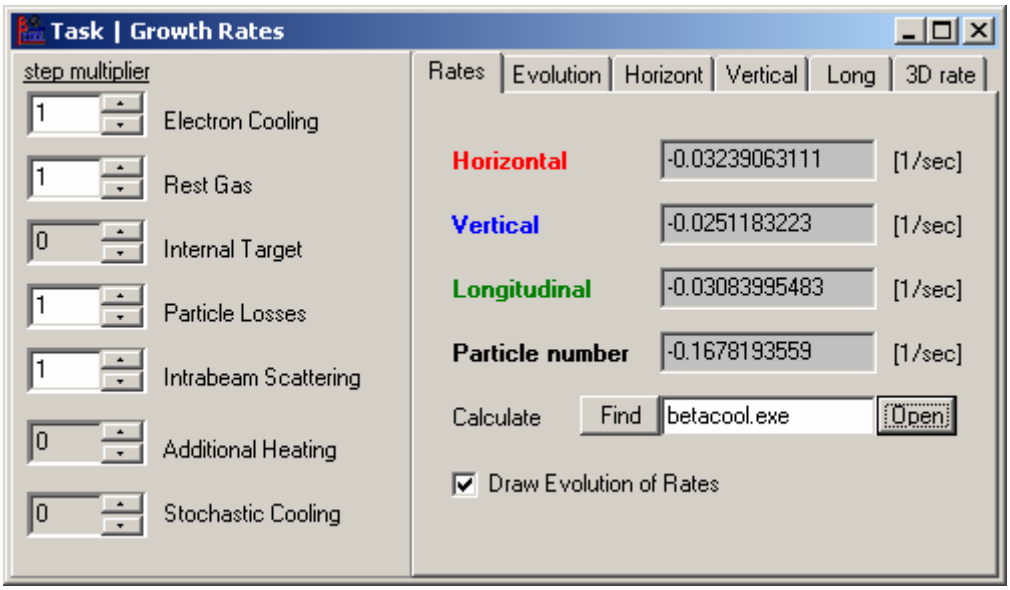


Fig. 5 **Task|Growth Rates**

VI. Ion Beam Object

1. Beam parameters

Parameters of the global variable of class `xBeam` and results of calculation of the beam parameter evolution in time are collected in the Main Menu *Item* **Beam** which includes following submenu items **Parameters, Distribution, Evolution**, **Real space**. The Window **Parameters** (Fig.1) includes 5 tab sheets: **Emittance, Model Beam, Multi-Injection, Bunch, Characteristics**.

1.1. Emittances and particle number

The *TabSheet* **Emittance** is used to define main beam parameters. When **Model Beam** Algorithm is used user must choose Emittance Definition. Here 4 types of emittance presentation are proposed: ***Root Mean Square*** (usual rms emittance – when 1 sigma of Gaussian distributed particles is presumed), ***Courant Snyder*** (when emittance is calculated as Courant Snyder invariant), ***Full Width on Half Maximum*** (emittance corresponded to particles inside full width on half maximum of distribution), ***Enclosed Percents*** (emittance occupied by the indicated percent of beam particles).

The input horizontal and vertical beam emittances correspond to **rms non-normalized** values. The relative momentum spread is also set as rms value. In the case of bunched beam simulation the **Number of particles** corresponds to the particle number in a single bunch (not in the total ring). All other parameters in the tab sheet are used in the Model Beam algorithm and for IBS rate calculation they can be arbitrary.
The *TabSheet* **Emittance** (Fig. 1) includes the following parameters which has to set:

Variable caption	**Unit**	**Comment, Formula**
Combo Box "bunched, coasting, barrier bucket"	-	different state of ion beam: 0 –coasting, 1 – bunched, 2 - bucket
Horizontal emittance	π·mm·mrad	
Vertical emittance	π·mm·mrad	
Momentum spread	-	
Number of particles		number of real particles
Model particle number		number of model particles
Combo Box "Emittance definition for Model beam"	-	Used for Model Beam Algorithm: 0 - Root Mean Square 1 - Courant-Snyder invariant 2 – FWHM 3 – Enclosed Percent
Enclosed percents for trans. and long. degrees of freedom	-	Used for definition of the emittances and momentum spread enclosed in pointed percentage of particles
Mean longitudinal invariant		parameter has to be switched on in the case when the momentum spread has some deviation from the reference energy

In the BETACOOL program the transverse emittance is define as one sigma un-normalized emittance and momentum spread is defined in relative value $\Delta P/P_0$. Usually on experiments at Recycler the transverse emittance is defined as 95% normalized emittance and momentum spread is defined in absolute value GeV/c. User can input transverse emittance and momentum spread in the

format of the BETACOOL definition and check these parameters in other units (Fig.1). Normalized emittance is calculated from one sigma un-normalized emittance as:

$$\varepsilon_{norm} = \beta\gamma n_\sigma \varepsilon_{rms},$$

where $\beta\gamma$ - relativistic factors, $n_\sigma = -2 \cdot \ln(1-\alpha/100)$ - number of sigma calculated via percents of the particle number α [%] for the Gaussian distribution. For Recycler measurements usually α=95% which is corresponds to n_σ=5.991 rms sigma. The momentum spread in the absolute value can be calculated from the relative momentum spread as:

$$\left(\frac{\Delta P}{P_0}\right)_{abs} = \left(\frac{\Delta P}{P_0}\right)_{rel} M\,[\text{GeV/c}],$$

where M – is the particle momentum in GeV/c (Fig.1.4). The energy spread is defined as:

$$\left(\frac{\Delta E}{E_0}\right)_{abs} = \frac{\gamma+1}{\gamma}\left(\frac{\Delta P}{P_0}\right)_{rel} K\,[\text{GeV}],$$

where K – particle kinetic energy. If the kinetic energy was defined per nucleon then energy spread will also defined as [GeV/u].

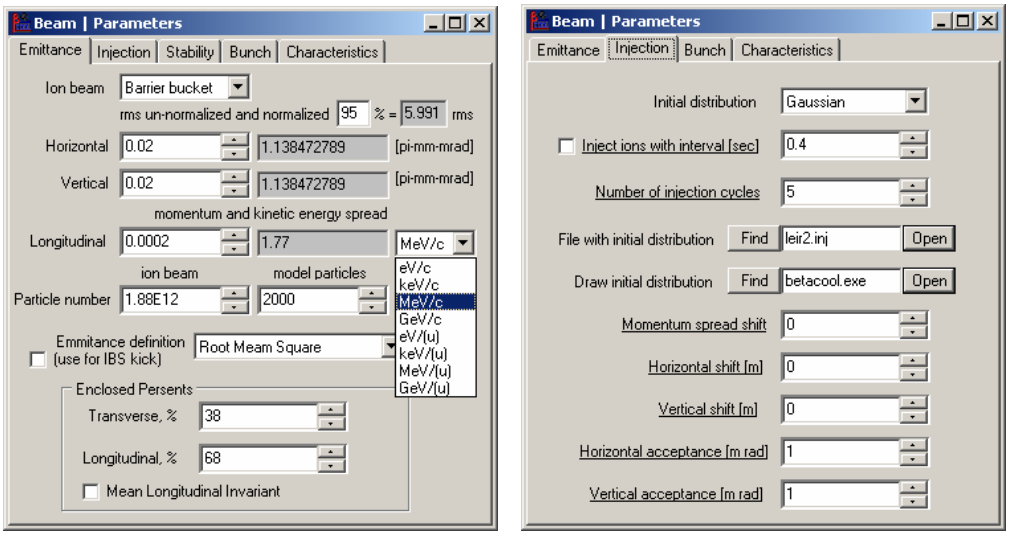


Fig.1. *TabSheet* **Emittance** & **Injection** of the menu item **Beam | Parameters**

1.2. Parameters of injection

The *TabSheet* **Injection Parameters** (Fig. 1) includes the following input and output variables:

Variable lable	**Unit**	**Comment, Formula**
Initial distribution		Choice of initial distribution type: Gaussian, Flattened, form file.
Inject ion with interval	sec	Injection repetition period
Number of injection cycles	-	Number of injections
File with initial distribution	-	Filename containing initial array of particles
Momentum spread shift		Shift of longitudinal momentum
Horizontal shift	m	Shift of horizontal position
Vertical shift	m	Shift of vertical position
Horizontal acceptance	m rad	Horizontal acceptance at injection
Vertical acceptance	m rad	Vertical acceptance at injection

Here a *TBrowse component* **Draw initial distribution** is intended to visualize an initial distribution of model particles which is generated in accordance to the user defined choice (**Initial distribution: gaussian, flatten, from file**). User can watch the distribution on different phase space at *Form* **Beam | Real Space**.

1.3. Ion bunch parameters

To adjust RF system parameters to the required bunch length (in the case of bunched beam) the TabSheet **Bunch** can be used (Fig. 2). This tab sheet contains one input parameter – **Number of bunches**. This parameter is necessary for luminosity calculation only and in IBS rate calculation it does not play a role. It can be arbitrary positive non zero number. After starting Betacool the program outputs rms bunch length corresponding to relative momentum spread of the bunch and RF system parameters. By a few subsequent runs of simulations one can adjust RF voltage (or harmonic number) to have required rms bunch length.

The *TabSheet* **Bunch** (Fig. 2) includes the following input and output variables:

Variable lable	Unit	Comment, Formula
Number of bunches	-	Used for luminosity calculation
R.m.s. bunch length	m	$\sigma_s = \beta_s \frac{\Delta p}{p}$, $\beta_s = \frac{\lvert\eta\rvert R}{Q_s}$
Maximum particle number	-	$N_{\max} = \frac{L_b^3 \gamma^2 heV}{3\pi Z G_L R^2 r_p U_p}$, $L_b = \sqrt{2\pi}\sigma_s$
Synchrotron tune	-	$Q_{s,N} = Q_s \sqrt{1 - \frac{N_i}{N_{\max}}}$
Bunching factor	-	$B_f = \frac{L_b}{L_{sep}}$, L_{sep} is the separatrix length

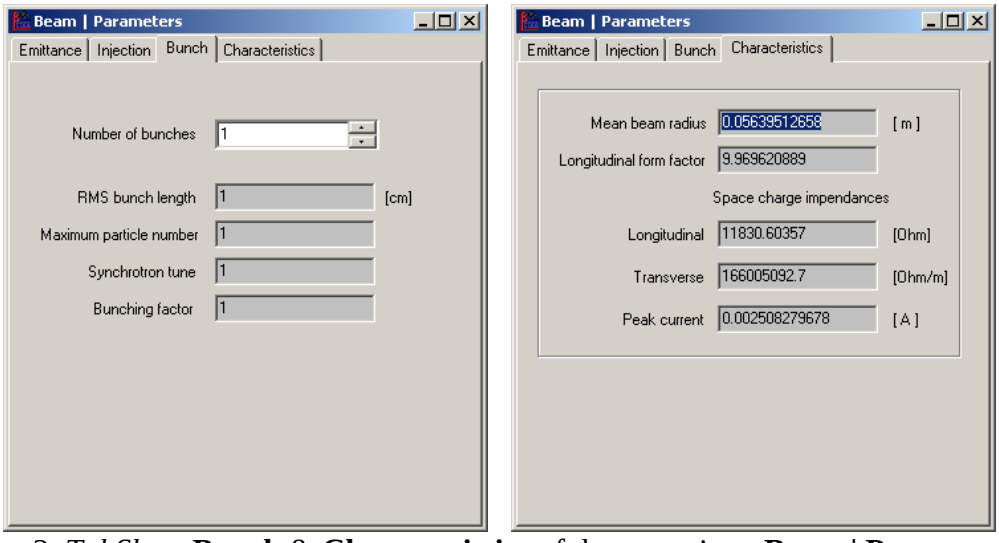


Fig. 2. *TabSheet* **Bunch** & **Characteristics** of the menu item **Beam | Parameters**

1.4. Calculated characteristics

The *TabSheet* **Characteristics** includes beam parameters calculated when main parameters of the beam and ring are determined.

This *TabSheet* (Fig.2) includes the following input and output variables:

Mean beam radius, a	m	Used for longitudinal form-factor calculation $a = 2\sqrt{\sigma_x \sigma_z}$
Longitudinal form factor , G_L		$G_L = 1 + 2\ln\frac{b}{a}$, b is the mean radius of the vacuum chamber
Longitudinal	Ohm	space charge longitudinal coupling impedance $Z_{L,sc}$ $Z_{L,sc} = \frac{Z_0 G_L}{2\beta\gamma^2}$, Z_0 = 377 Ohm
Transverse	Ohm/m	Space charge transverse coupling impedance $Z_{t,sc}$, $Z_{t,sc} = \frac{Z_0 R}{\beta^2\gamma^2}\left(\frac{1}{a^2} - \frac{1}{b^2}\right)$, R is the ring mean radius
Peak current	A	Current of the coasting beam or peak current for the bunched beam, $I = \frac{ZeN_i}{B_f T_{rev}}$, B_f is the bunching factor

2. Visualization of beam parameters

2.1. Evolution of emittances

The Window **Beam | Evolution** (Fig. 3) is used for visualization of the beam parameter time dependencies. *TabSheet* **Emittance** (Fig.3) contains 2D plot for output two curves: `Ex2t.cur`, `Ey2t.cur` - time dependencies of horizontal and vertical emittances. *TabSheet* **Momentum** (Fig.3) contains 2D plot for output curve `Dp2t.cur` - time dependence of momentum spread, and 2D plot `dpmo2t.cur` – deviation of momentum spread. *TabSheet* **Number** contains plot for output `Num2t.cur` – particle number time dependence. *TabSheet* **Bunch** contains plot for output `Bunch2t.cur` for time dependence of bunch length.

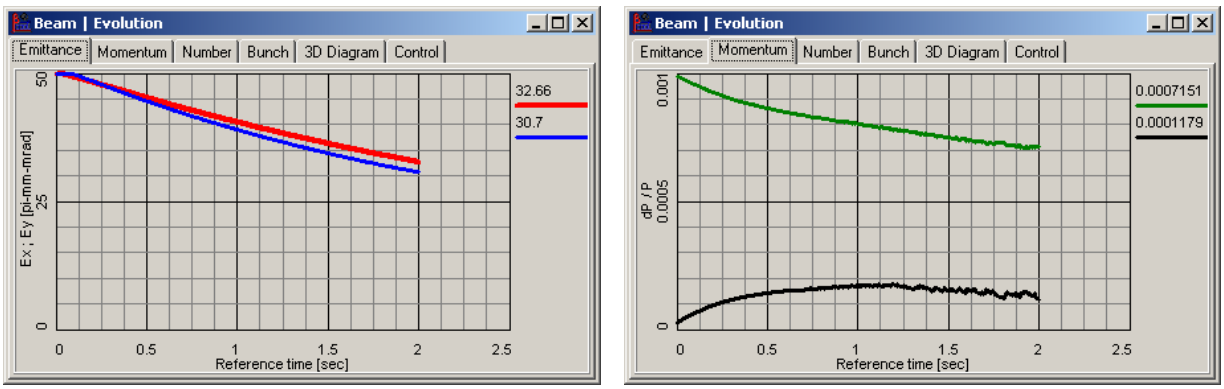


Fig. 3. Window of the **Beam | Evolution** | **Emittance** & **Momentum**

TabSheet **3D Diagram** (Fig.4) contains plot for output `gamma2.cur`– criterion $\Gamma_2=\pi$, `gamma3.cur` – equilibrium between transverse and longitudinal ion temperature, `footprint.cur` – unsorted invariants of particles, the dependence (evolution) of the horizontal emittance on the momentum spread `txy2t.cur`. The *TabSheet* **Control** (Fig.4) allows user managing of the plots visualization. It contains list of all the plots of the current window which presented by Checkboxes. User must check plots which are needed to be redrawn on-line and (or) check off unnecessary plots. Such a scheme allows to manage CPU resource in order to fasten calculation time.

An evolution of transverse emittances and momentum spread can be plotted with different definitions (Fig.4): the transverse emittance – one sigma un-normalized and normalized with certain percent value of the particle number, the momentum spread – relative and absolute values. These definitions correspond to parameters on **Beam Parameters** window (Fig.1).

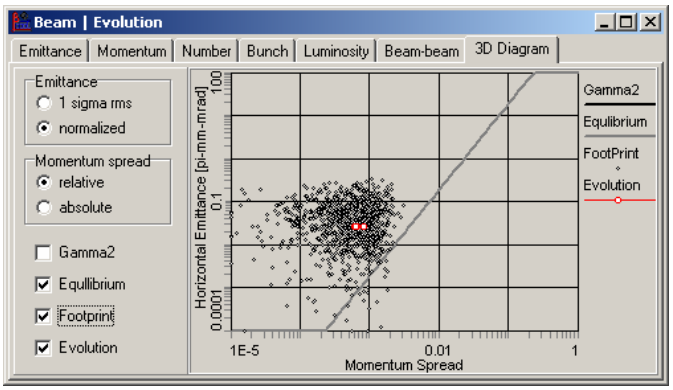


Fig. 4. Window of the **Beam | Evolution | 3D Diagram & Control** .

2.2. Phase space of model particles

The Window **Beam | Real Space** (Fig. 5) is used for visualization of the particle distribution in different planes during simulation procedure. *TabSheet* **Control** here is a manager of plots – user can choose which plot is active for on-line redrawing and what a coordinate plane will be represented on it (for example **Y-X** – transverse real space of particles, **X'-X** – horizontal phase space, **Y-S** – longitudinal real space etc).

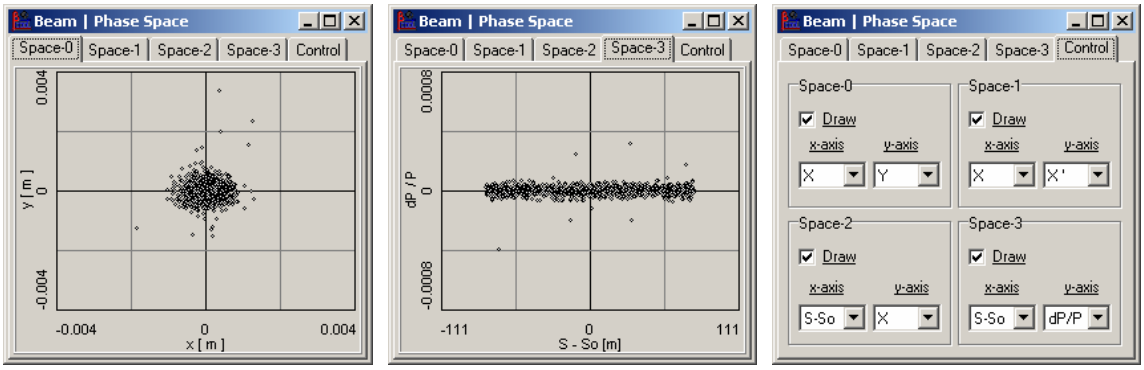


Fig. 5. Windows of the **Beam | Phase Space** menu item.

2.3. Distribution of profiles and invariants

The Window **Beam | Distribution** (Fig. 6, 7) is used for visualization of the real particle distribution for model beam. *TabSheet* **Coordinate** – is the dependence of particle number (in percent) on momentum deviation (`dist_sp.cur`) and coordinate (`dist_sx.cur` and

`dist_sy.cur`) normalized on corresponding rms parameter and particle number ($100\% \cdot (\Delta P / P)/(\sigma_p \cdot N)$). *TabSheet* **Profile** - is real particle distribution for every coordinate (analogue of previous plot) averaged on betatron or synchrotron oscillations (`dist_ix.cur`, `dist_iy.cur`, `dist_ip.cur`). *TabSheet* **Invarinat** - is plot of all three sorted invariants (`dist_ex.cur`, `dist_ey.cur`, `dist_dp.cur`) – actually it shows particle number (in percent) which occupies corresponding emittance (abscissa axis).

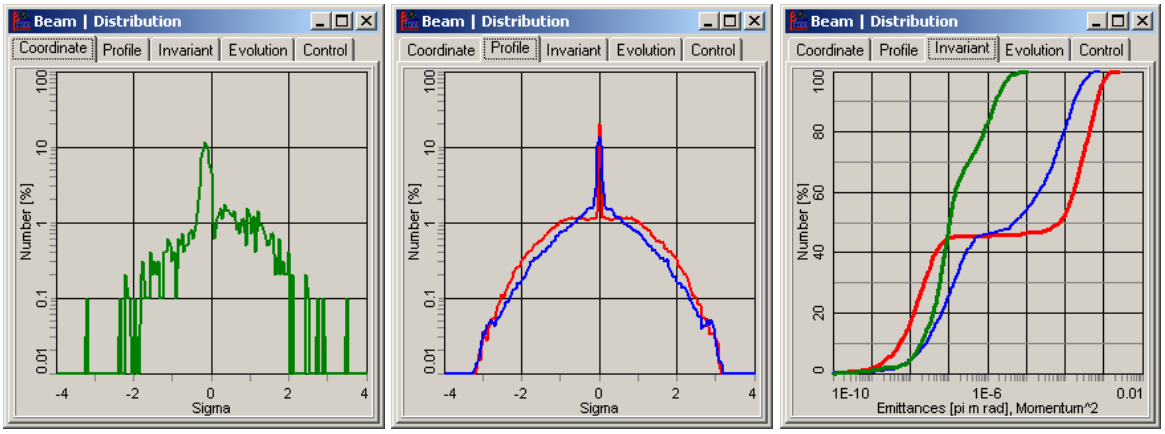


Fig. 6. Windows of the **Beam | Distribution** menu item.

TabSheet **Evolution** (Fig.7) – 3D plot for evolution of Coordinates or Invariants in time (evolution.sur). *Tabsheet* **Control** – settings for plot on all *TabSheets* listed above. As on every plot window user can choose which plots he wants to be redrawn on-line. *GroupBox* ***Coordinate, Profile*** correspond to first two *TabSheets* and defines: **Sigma** – is range in number of sigmas, **Division** – split number, **normalized on emittance** – choice of rms parameter for sigma to be normalized on . *GroupBox* **Evolution** defines settings for 3D plot: **Slices** – number of steps in time scale, **Step, sec** – value of step in time. **Bi-Gaussian** – show interpolation for profiles, **[sum]** – averaging on time.

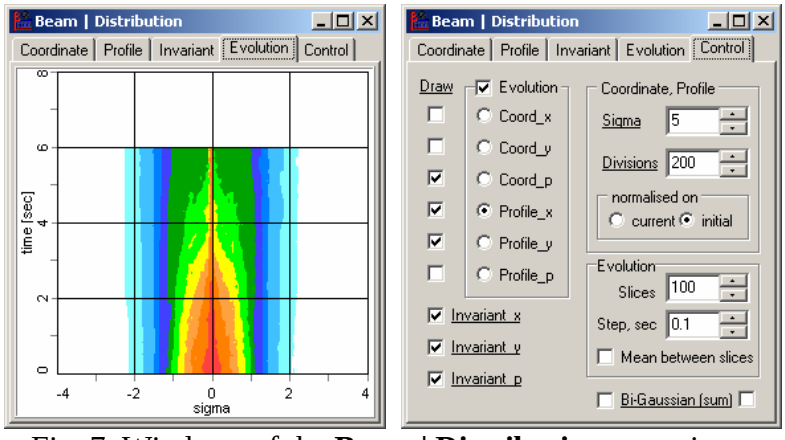


Fig. 7. Windows of the **Beam | Distribution** menu item.

VII. Storage Ring Object

Parameters of the global variable of *class* **xRing** and results of calculation of the Ring main parameters are collected in the *Main Menu item* **Ring** that includes submenu items **Lattice Structure** and **Parameters**.

1. Parameters of Ring

Window **Ring | Parameters** contains the *TabSheets* **Ion kind, Lattice, Mean params, RF system, Reference point.**

1.1. Setting of ion kind parameters.

The corresponding *visual form* (Fig. 1) has 6 *TabSheets* containing input and output parameters of the storage ring.

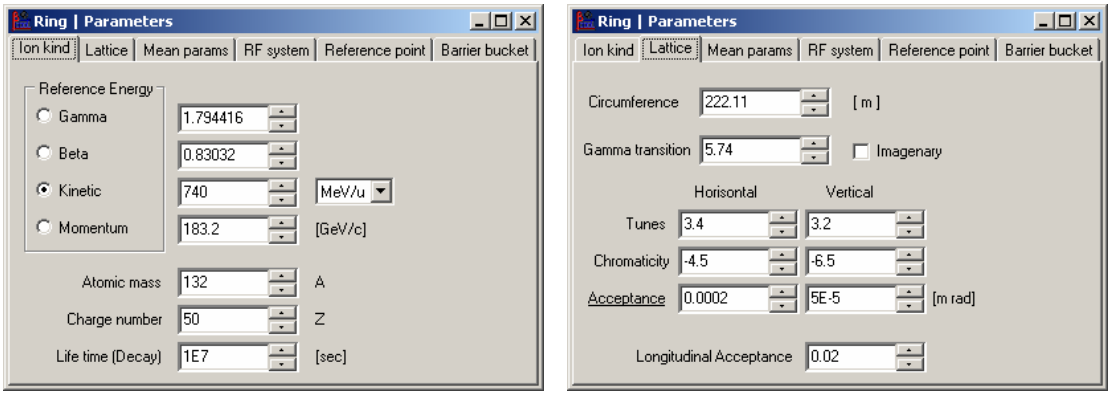


Fig. 1. **Ring | Parameters** *visual form*. *TabSheet* **Ion kind & Lttice**. This example corresponds to the totally stripped gold ions at kinetic energy 100 GeV/u.

To set the beam energy one can use any of the following possibilities chosen with *RadioButton* **Reference Energy**:

- Lorenz factor γ (**Gamma**);
- Particle velocity in the units of the speed of light (**Beta**);
- Kinetic energy (**Kinetic**);
- Particle momentum in GeV/c (**Momentum**).

The chosen option corresponds to input parameter; all other numbers will be recalculated after launching the Betacool program. The kinetic energy can be set as a total value or divided by number of nucleons. The units of the kinetic energy are to be chosen using combo box aligned with the corresponding edit window.

The **Atomic mass** is set as a number of nucleons in the nuclei and usually it is integer number. The rest energy of the particle is calculated in the program as A*938 MeV. In principle, the atomic mass can be arbitrary positive number and the IBS growth rates can be calculated for different particles – muons, electrons etc.

The parameter **Charge number** corresponds to the charge state of the ion and can vary from 1 to the atomic number. The **Charge number** is set as an absolute value – for instance, as for protons as for antiprotons Z = 1.

The input parameter **Life Time (Decay)** is used for the particle loss simulations only. For the rate calculation it can be arbitrary.

1.2. Lattice parameters of Ring

The storage ring **circumference**, **Lorenz factor** corresponding to the ring transition energy, **horizontal and vertical tunes** have to be set in the tab sheet **Lattice** (Fig. 1). If the **transition energy** is imaginary the corresponding *check box* has to be checked.

Variable lable	Unit	Comment, Formula
Circumference	m	Ring circumference
Gamma transition	-	used for off momentum factor calculation
Imagenary	-	used to determine complex value of γ_{tr} factor
Tunes	-	Horizontal and vertical tune values
Chromatisity	-	used for tune spread calculation
Acceptance	π·m·rad	used for calculations of life time due to single scattering on big angles
Longitudinal acceptance		used for calculations of life time due to particle off the separatrix length

Chromaticities are used for stability estimation and for the rate calculation the values can be arbitrary. The acceptance values are used for the particle loss simulation. For the rate calculation the acceptances can be set as arbitrary nonzero positive numbers.

1.3. Mean parameters

The *TabSheet* **Mean params** (Fig. 2) contains only output parameters which will be calculated after start of Betacool calculations.

Variable caption	Unit	Comment, Formula
Mean radius	m	$R = C/2\pi$
Horiz. beta function	m	$\beta_x = R/Q_x$
Vertical beta function	m	$\beta_z = R/Q_z$
Dispersion	m	$D = \beta_x/Q_x$
Revolution period	sec	$T_{rev} = C/\beta c$
Off momentum factor	-	$\eta = \frac{1}{\gamma^2} - \frac{1}{\gamma_{tr}^2}$

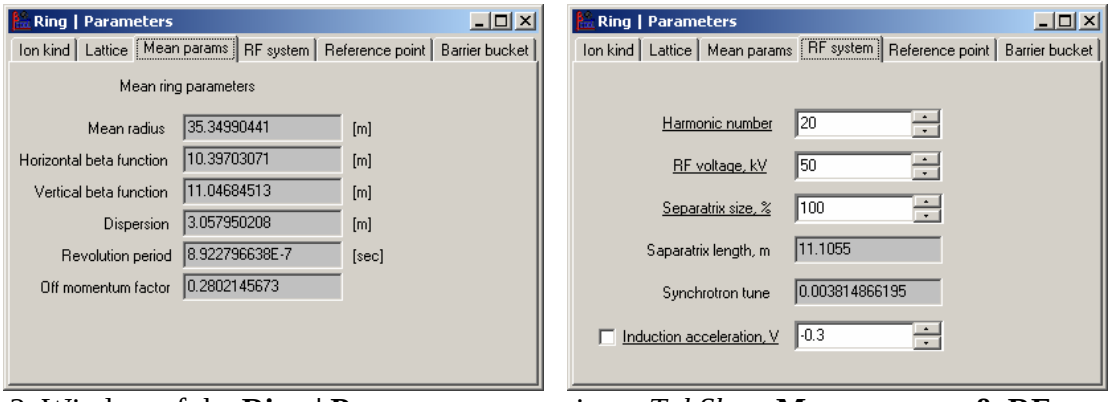


Fig. 2. Window of the **Ring | Parameters** menu item. *TabSheet* **Mean params & RF system.**

1.4. RF system for bunched beam

If the beam is bunched one needs to set RF **harmonic number** and **amplitude** of the RF voltage in the *TabSheet* **RF system** (Fig. 2).

The **Separatrix size** parameter is used for the particle loss simulation and it can be arbitrary positive number. **Separatrix length** and **Synchrotron tune** are output parameters. The check box **Induction acceleration** has to be not checked.

Variable caption	Unit	Comment, Formula
Harmonic number	-	
RF voltage	kV	
Separatrix size	%	% of particles involved into longitudinal invariant
Induction acceleration		[boolean]. Used if induction acceleration is presented
Induction acceleration voltage	V	Amplitude of the induction acceleration field
Separatrix length	m	$L_{sep} = C/h$
Synchrotron tune	-	$Q_s = \frac{1}{\beta}\sqrt{\frac{eh\lvert\eta\rvert ZV}{2\pi U_p A\gamma}}$

1.5. Reference point

The *TabSheet* **Reference point** (Fig.3) is used for matching of the particle array with the ring optic structure in the frame of Model Beam algorithm. For the rate calculation the lattice functions in injection point can be arbitrary. It determines parameters of some Ring point – it can be point of beam injection or just a reference point for dynamics calculation. Here a *ComboBox* **Lattice functions at point** is intended for the selection of lattice parameters at reference point. To avoid errors at initialization of the program variables the beta functions have to be nonzero positive.

Variable caption	Unit	Comment, Formula
Beta [m]	[m]	Horizontal and vertical beta functions
Alpha	-	Horizontal and vertical alpha functions
Dispersion [m]	[m]	Horizontal and vertical Dispersions
Dispersion derivative	-	Horizontal and vertical Dispersion Derivatives

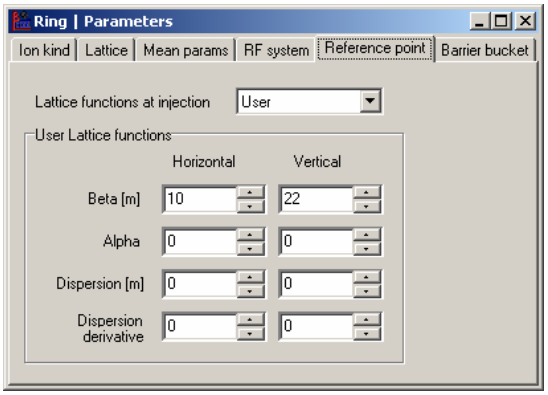


Fig. 3. Window of the **Ring | Parameters** menu item.
TabSheet **Reference point & Burrier Bucket.**

There are the following 3 *Items* are presented. **Mean Ring** – when mean Ring lattices are taken for calculations. **First Optics** – when lattices of the first optic element are taken from file with lattice

structure. **User** – when lattices at reference point are taken in accordance to user definition which can be made using *EditWindows* at the current *TabSheet* below presuming the following input and output variables:

1.6. Burrier Bucket model

Parameters of barrier bucket (BB) are defined in Form (Fig.4). Three options of BB model are realized in BETACOOL: 2 square barriers, stationary bucket and moving bucket. Position and duration of BB are defined in the unit of circumference in the range -0.5 < s <= 0.5, amplitude of buckets are given in volts.

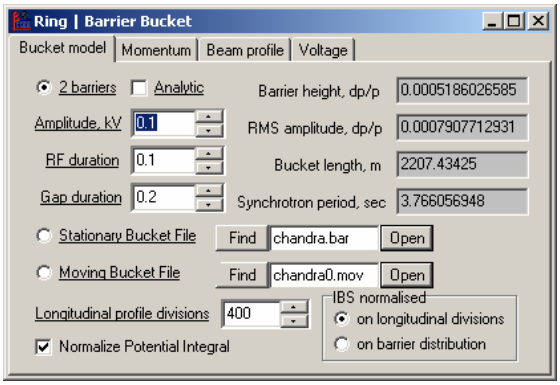


Fig.4. Parameters of BB.

2 barriers

Options **2 barriers** is used for the simple model of two barriers which has as analytical as numerical approximation. **Analytic** solution was done for the region between barriers only. If the particle momentum spread becomes larger than the barrier height than the particle does not move in the longitudinal direction. The numerical solution simulates the longitudinal particle dynamics around whole ring. If the initial distribution of particles is chosen not from file than the parameters of **2 barriers** model are used for the initial longitudinal distribution of particles which will be fitted in the accordance with barriers parameters – particle can not have the momentum spread larger than barrier height.

Amplitude, kV defines the amplitude of the barrier. **RF duration** – barrier width in the circumference unit, **Gap duration** – distance between barriers in the circumference units. **Barrier height, dp/p** calculates the barrier height in the momentum spread units. **RMS amplitude, dp/p** – momentum spread after fitting in the accordance with barriers parameters (can be substantially less than input momentum spread). **Bucket length, m** – RMS bucket length, **Synchrotron period, sec** – synchrotron period of RMS particle.

Stationary bucket

Stationary Bucket File (Fig.5a) has two columns: first is position of barrier in the unit of circumference, second is amplitude in volts. First barrier always start from position -0.5 and file can not have position -0.5. If barrier position 0.5 is not included in the file it means that last barrier has position 0.5 with zero amplitude. Red line (Fig.3b) is amplitude distribution, blue line is integral of barrier amplitudes [kV*m]. It can be normalized on the maximum amplitude of barriers if **Normalize Potential Integral** options is chosen (Fig.4).

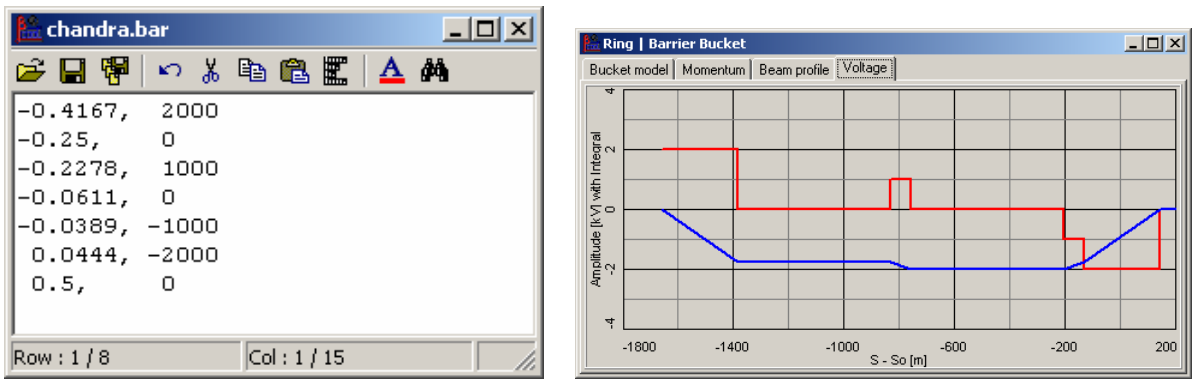


Fig.5. (a) File for definition of stationary BB, (b) amplitude and integral distributions.

The particle distribution and barrier positions in the longitudinal phase space are shown on Fig.6a. The longitudinal profile (Fig6b, green line) is calculated in accordance with parameter **Longitudinal profile divisions** (Fig.4), red line is the average particle density in each barrier, black line – integration of particle along longitudinal coordinate. Intrabeam scattering heating rates are calculated via longitudinal density of particle which can be chosen in parameter **IBS normalized** (Fig.4): **on longitudinal divisions** means that IBS is calculated in accordance with particle density with longitudinal divisions, **on barrier divisions** – in the accordance with particle density in barrier region.

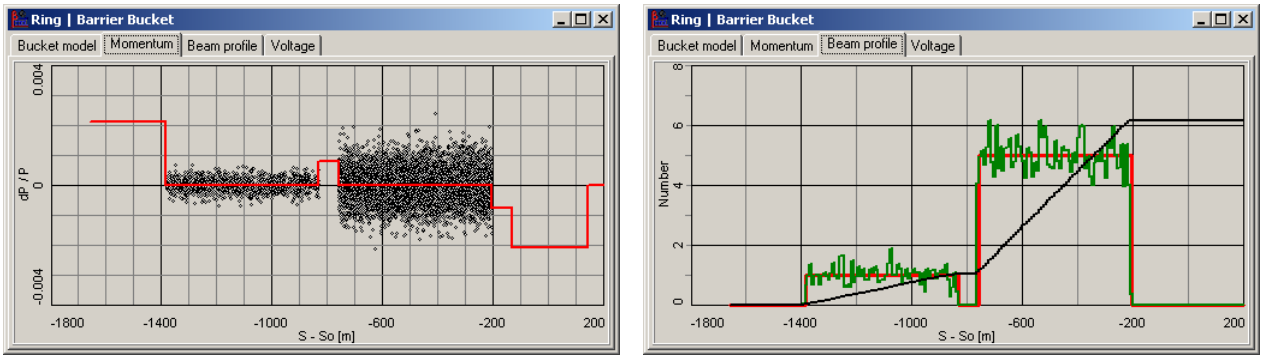


Fig.6. (a) Particle and barrier distribution in the longitudinal phase space
and (b) particle density along longitudinal coordinate.

Moving barrier bucket

The moving barrier can be used for manipulation with particles in the longitudinal phase space. Note that integration step for this task should be much smaller than synchrotron tune and option **Random synchrotron** (Fig.7) should be switched off.

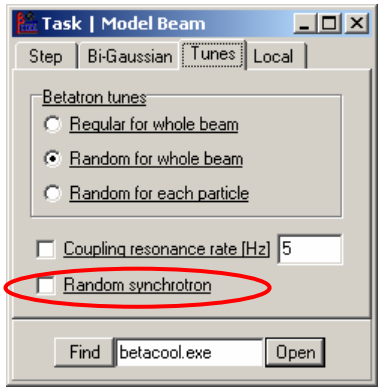


Fig.7. Options for betatron and synchrotron tunes.

Moving Bucket File (Fig.8) has any number of columns: first column is time in seconds, each other pairs define position of barrier in circumference units and amplitude in volts. During simulation process the linear fitting of barrier positions and barrier amplitudes are realized between lines. When reference time is reached the final line then the parameters of barriers don't change more.

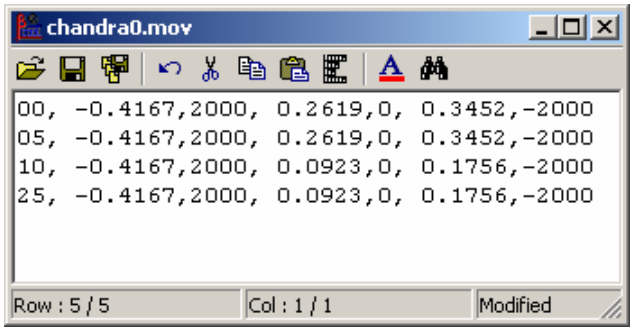


Fig.8. (a) File for definition of moving bucket.

2. Lattice Structure

To specify ring lattice structure which is necessary first of all for IBS effect simulation user must use *Window* **Lattice Structure** (Fig. 9) which contains *TabSheets* **Lattice Filename, Output MAD format,** and three *TabSheets* connected to lattices of optic channel and its transformation matrix: two of 2D plots **beta functions** and **alfa functions,** and **Matrix**.

2.1. Input files with lattice structure

Using *TabSheet* **Lattice Filename** (Fig.9) one must select and save to the input file the names of files containing information about ring optics structure. This *TabSheet* contains:

- *ComboBox* **Lattice structure File** with three possible items: O**utput MAD format**, I**nput MAD format**, **No file** and three of *TBrowse* components. This *ComboBox* is used when user has a file with some lattice structure obtained from MAD program or assembled in accordance to the input MAD file standard. **No file** option is chosen when user does not need Lattice file.

- *ComboBox* ***Lattice Structure*** has also three possible selections: r**educe (filename), extend (step, cm), no changes**. Here choice of active option depends on the definition made in described above **Lattice structure File** *ComboBox*.

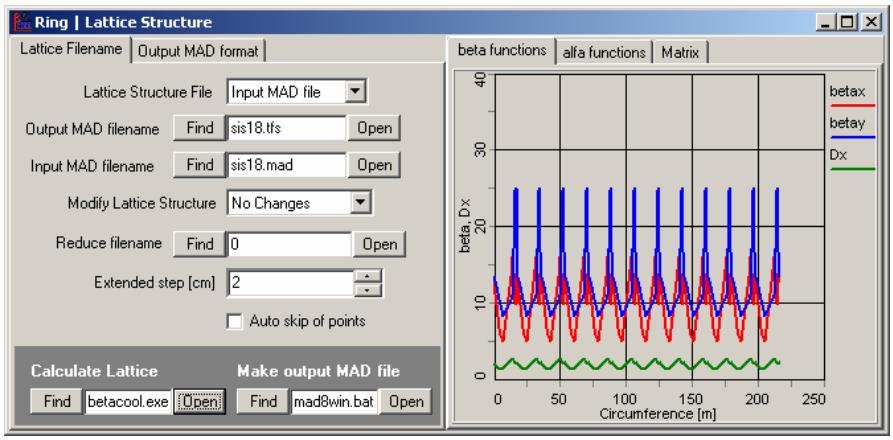


Fig. 9. Window of the **Ring | Lattice structure** menu item. *TabSheet* **File names**.

In this *TabSheet* in the combo box **Lattice Structure File** the option **Output MAD file** (or **No Files** for Piwinski model) has to be chosen. The file name has to be pointed in the *TBrowse* element **Output MAD filename** (Fig.10). The *button* **Open** of this *TBrowse* element starts internal BOLIDE text editor and pushing this button user can check existence and validity of the chosen file.

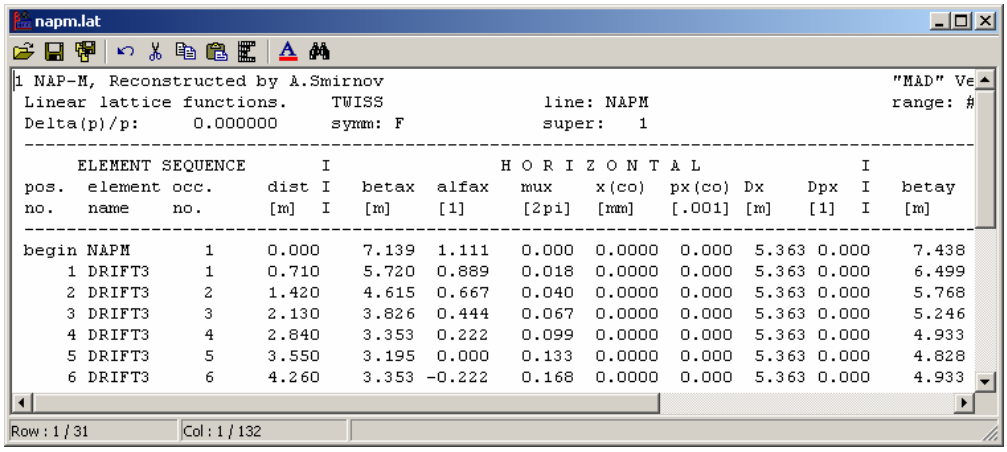


Fig.10. Example of output MAD file

Another *TBrowse* component at the Form **Ring | Lattice structure** (Fig. 9) is **Make output MAD file** which allows to launch MAD application in the framework of BETACOOL and to create file with lattices in format of MAD output file. Here user must indicate MAD input file with **find Input MAD file** component and then to launch *TBrowse* component **Make output MAD file** to start MAD program generate output file with tracked lattices. **Input MAD format –** is used for choosing the input file name for multi particle tracking. BETACOOL program can read optics and lines from the standard input MAD file and translate to the optics structure of the storage ring (Fig. 11).

Input MAD file is needed to provide tracking using Molecular Dynamics algorithm. Betacool translator does not support all the possibilities of MAD input file syntax and sometimes it is necessary to modify the input file manually. Therefore for IBS rate calculation it is easier to use output MAD file. To provide a choice of the lattice file name and its specification the *menu item* **Ring** *submenu item* **Lattice Structure** and corresponding visual form (Fig.9) are used.

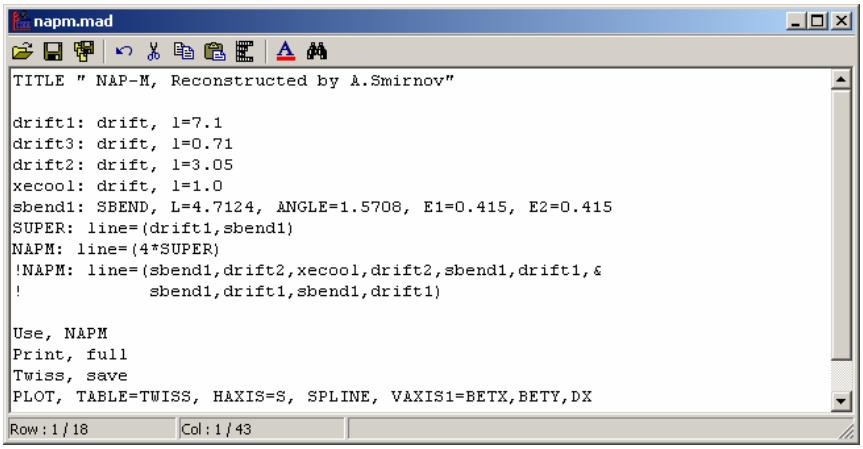


Fig 11. Example of input MAD file.

The result of translation is saved to file with same name as MAD input file and extension *.use. BETACOOL can translate the following elements from MAD file:

- *DRIFT;*
- *SBEND (ANGLE – bend angle, E1 and E2 – edge angles);*

- *QUADRUPOLE (K1 – quadrupole gradient, TILT - rotation);*
- *SEXTUPOLE (K2 – sextupole gradient);*
- *RFCAVITY (FREQ – RF frequency, V – RF Voltage)*
- *SOLENOID (Ks – solenoid gradient);*
- *LINE;*
- *USE.*

All the elements have obligatory parameter LENGTH

Depending on the task, the BETACOOL program can be a very powerful instrument for processing of ring lattices and (or) transformation matrices of optics element. If user has a MAD input file with optics element consequence, there is an algorithm that provides a calculation of transformation matrix for whole ring, then the second step – is calculation of lattice parameters in the point S_0, and after tracking of lattice parameters takes place. If one has a lattice at the chosen point and matrix after, it is possible to calculate lattices after matrix, and so on and so force. For this procedure user has to choose **Input MAD format** *ComboBox Item* and to click **Calculate Optics Structure**. After some time period, calculated lattices will be visualized onto corresponded plots.

Reverse procedure is also provided – co called "reverse-tracking" – when one chooses the **Output MAD format** *ComboBox Item* – the consequence of transformation matrices for optical elements of the structure will be restored from the consequence of lattices.

2.2. Parameters of lattice structure

If *CheckBox* **Auto skip of points** (Fig. 9) is switched on then the following procedure takes place: when the number of points in any curve (here the curves with lattice parameters) is to be exceeded then because of the limited size of the curve with every next point inserted to the curve array one looses first point from the array. So if *CheckBox* is ON, every second point in this array will be deleted and all left points will be suppressed, then at least a half of the curve array will be released for the expected calculated data. This procedure can be eternal - always when array is nearly to be overfilled, every second cell in it will be released.

In *ComboBox* **Modify Lattice Structure** the option **No Changes** has to be chosen. All other parameters do not play a role in calculations but it is better to input where required the names of arbitrary files existed in the current folder.

Reduce(filename) (Fig. 9) – is useful option foreseen for large lattice (optics) structure. Here one must built a special file *.red with reduced structure of the ring. User has to leave only optical elements and corresponded which he wants to be taken into account for calculations and visualization. Here the following algorithm takes place: for the tracking using matrices the whole structure is taken into account. However necessary matrices will be built only in user selected points (by multiplying all the intermediate matrices between selected points). Then only these matrices will track the beam. And IBS effect will be calculated only in these points. Finally, lattices in selected points will be plotted. So the calculation time may be sufficiently reduced.

Extend (step, cm) – (Fig. 9) this option is used only when Tracking algorithm is presumed and Molecular dynamics technique is used. Here the step over longitudinal coordinate is defined between points where transformation matrix is calculated. This choice is active when option ***Matrixes*** is switched on in **Task | Algorithm | Tracking | Equations of motion** window. The second necessary condition – is **Input MAD format** must be chosen in **Lattice structure File** *RadioGroup*.

A table below presents different modes of usage the **Ring | Lattice Structure | Lattice Filename** *RadioButtons* for selected tasks (for details see description of **Task | Algorithm**):

Radio button / Task	Output MAD format	INPUT MAD FORMAT	No optics	COMMENTS
RMS DYNAMICS	√	√	√*	* - in case when Piwinsky model of IBS calculation was chosen
Model Beam	√	√	√**	** - if chosen No optics – user has to fill (type in) the ring transformation matrix

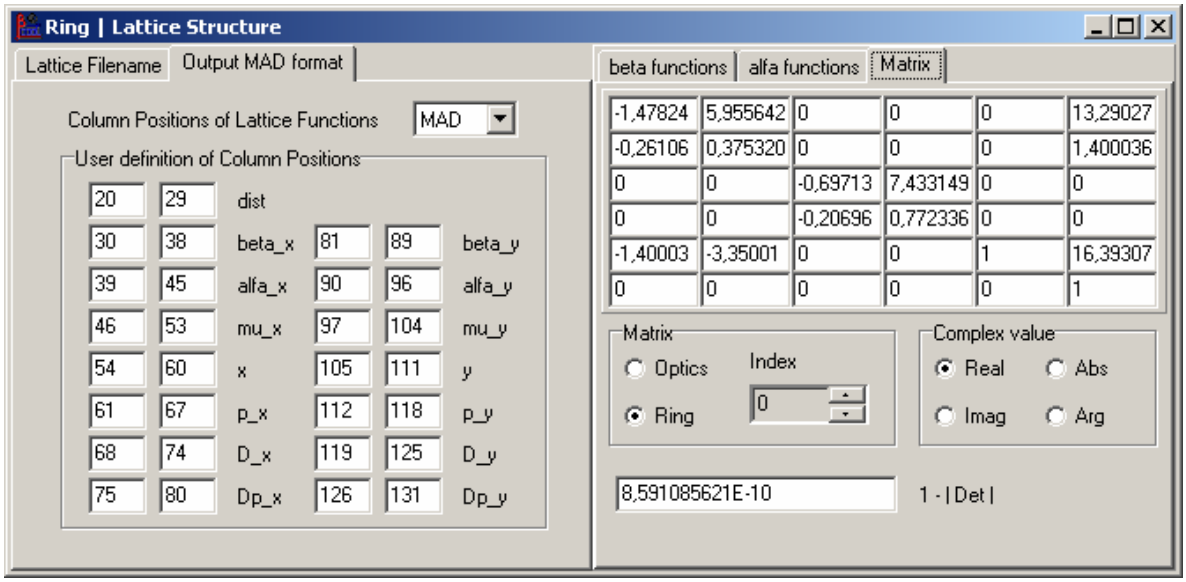


Fig. 12. Window of the **Ring | Optic structure** menu item. *TabSheet* **Matrix**.

The next *TabSheet* **Matrix** (see Fig. 12) is useful tool for looking at the transformation matrix of the whole ring or selected optic element. One can choose to look at either transformation matrix of the whole ring (**Ring**) or selected element from optic structure (**Optics**) by choosing its number with **Index** *Counter*. Here complete transformation matrix is visualized. There is an option to look at any part of numerical appearance of matrix elements, if it is introduced as complex one. One can choose the representation by switching **Real | Imag | Abs | Arg** *RadioButtons*. Here *EditWindow* **|1-Det|** introduces the precision of calculated matrix determinant.

2.3. Checking of input files

To check validity of the MAD file specification one needs to start Betacool using button **Open** of *TBrowse* element **Calculate Lattice** (in the left bottom corner in the Fig. 9). This button starts Betacool with parameter **lattice**. Betacool reads MAD file in accordance with specification at the tab sheet **Output MAD format** (Fig. 10). If the positions of the lattice functions in the file correspond to the specification, Betacool outputs the lattice functions into corresponding *.cur files (BetaX.cur, BetaY.cur, DispX.cur, AlfaX.cur, AlfaY.cur, DispX_.cur - these curves are shown in the *TabSheets* **beta functions** and **alpha functions** of the *Form* **Ring | Lattice Structure** (Fig. 9). Betacool stops the calculation after without any warning (see example in the Fig. 13).

If the lattice function positions do not correspond to the specification, Betacool generates the following warning:

```
START - 2007 / 2 / 25 - 17 : 9 : 49
D:\betacool\betacool.exe esr.BLD /lattice
Error in Lattice Structure file : esr.tfs
Press Enter to continue ...
```

and stops the calculation after pressing Enter. The warning is saved in the Betacool.war file and can be read using Bolide text editor (*menu item* **File**, *submenu item* **Editor** or corresponding button in the Main tool bar. Example of the Betacool.war file generated after three runs of calculations is shown in the Fig. 13.

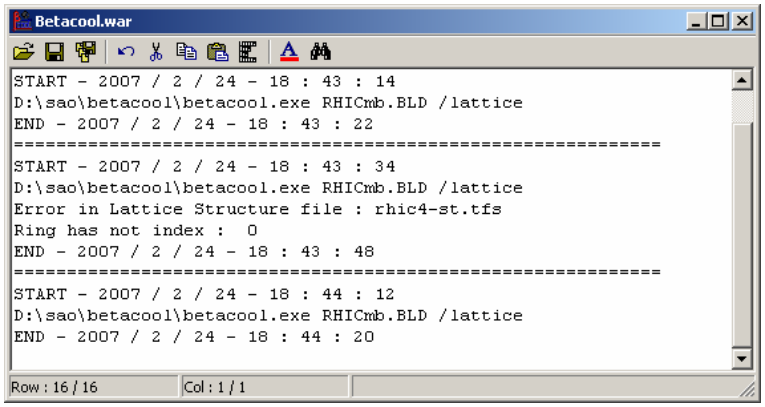


Fig. 13. Betacool warnings. The first and the last runs of the program have been successive. In the second run the positions of lattice parameters in the input file do not coincide with specification in *TabSheet* **Output MAD format**.

Betacool supports three standard of output MAD file:
- 132 columns;
- 157 columns and
- the output file can be generated by mad8win.but file included into Betacool package (**MAD** *option*). The corresponding file specification can be chosen using *ComboBox* **Column Position of Lattice Functions** (Fig. 14).

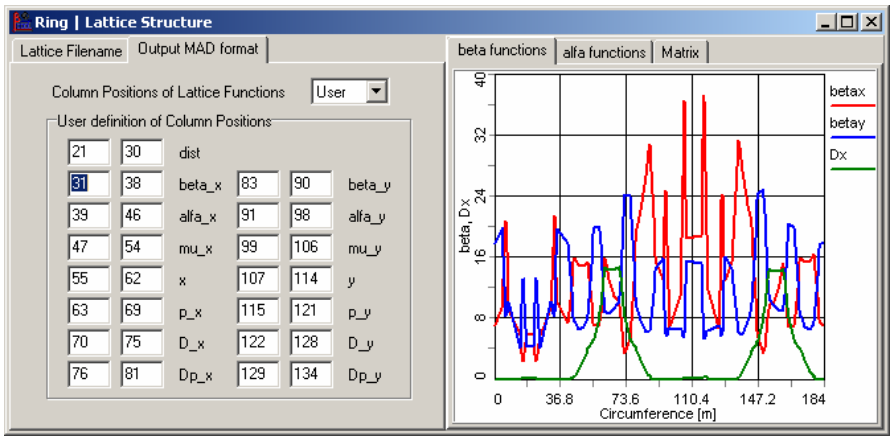


Fig. 14. **Ring | Lattice Structure** *Form*. *TabSheet* **Output MAD format**.

If the column number in the output MAD file does not coincide with one of the standards the *option* **User** has to be chosen. In this case initial and final column in the output file corresponding to each lattice function have to be input manually in the table of edit windows **User definition of Column Position**. The column corresponding to the cursor position is indicated in the BOLIDE text editor in the left bottom corner.

Names of the lattice functions in the **User definition of Column Position** table coincide with the names of corresponding lattice function in output MAD file.

To be sure the lattice parameters are initialized correctly one needs to push right button of the mouse at the plot of lattice function in the **Ring | Lattice Structure** *Form*. In the plot control window one of the curves containing lattice parameters has to be chosen in **Curves** *TabSheet* (Fig.15 left window). In the *TabSheet* **Values** (right window in the Fig. 15) one can see the values of chosen lattice function. The first column in the tab sheet contains the distance from the ring initial point in meters, the second column – corresponding lattice function.

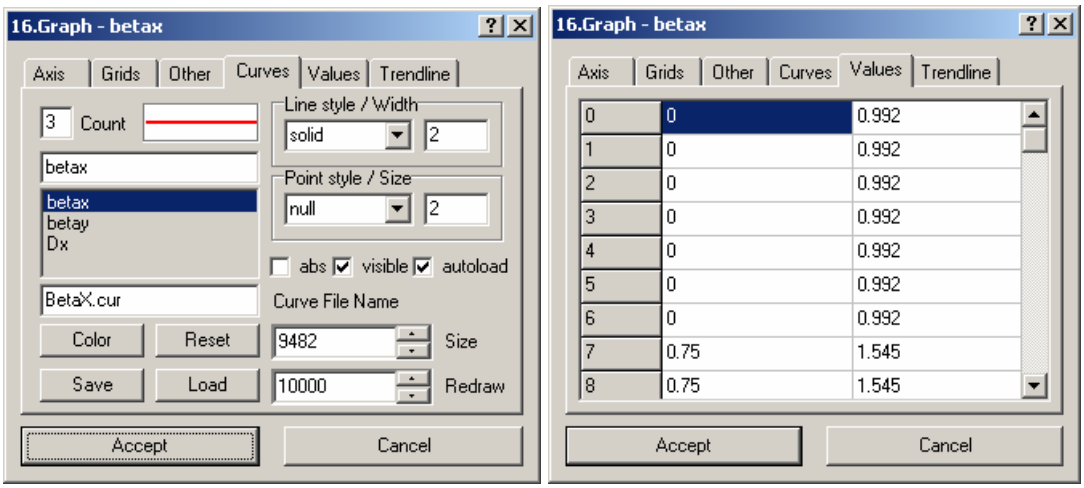


Fig. 15. Values of horizontal beta function initialized from the file.

VIII. Intrabeam Scattering

After launching the **Bolide.exe** file and checking a validity of an input file one needs to set the storage ring and beam parameters using corresponding menu items of the main interface window (Fig. 1).

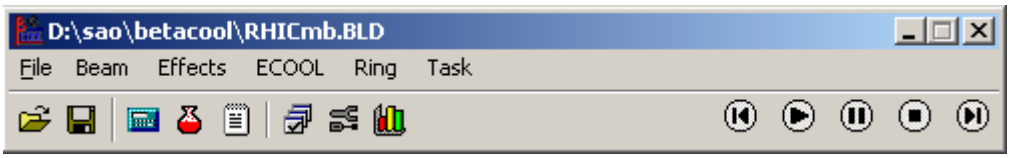


Fig 1. Main interface window

Thereafter the model for IBS rate calculation has to be chosen using *menu item* **Effects** *submenu item* **Intrabeam Scattering**. If the model requires a lattice structure of the ring it has to be imported from external file using *menu item* **Ring** *submenu item* **Lattice Structure**.
If user wants to switch IBS effect as active into simulation it is necessary to switch **step multiplier** corresponding to the **Intrabeam Scattering** on the *Form* **Task | Growth Rates** to non-zero value (for details please look for Numerical algorithm manuals).

At the first step one needs to choose *menu item* **Ring**, *submenu item* **Parameters**. How to set the Ring parameters please look [RING manual].

Parameters of six-dimensional phase volume of the beam required for IBS rate calculation are set in the *TabSheet* **Emittance** of the **Beam | Parameters** *visual form*. For more detailed description of beam parameters definition please look [BEAM Manual].

1. IBS parameters

The *menu item* **Effects** *submenu item* **Intrabeam Scattering** is used to provide a choice of the model for IBS rate calculation. The corresponding visual form (Fig. 2) contains *ComboBox* **IBS model**, *check box* **Average transverse** and four *TabSheets* for input parameters of different models.

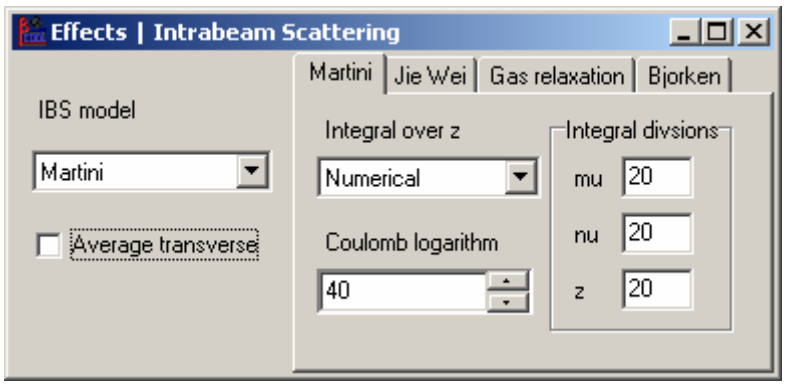


Fig. 2. The *visual form* for choice of the IBS model.

The current version of Betacool calculates the IBS growth rates for uncoupled transverse motion only. In the case of residual coupling or at work near coupling resonance the transverse rates can be averaged using the *check box* **Average transverse**. If it is checked the program calculates the transverse rates for uncoupled motion, thereafter calculates mean value and the both – horizontal and vertical rates are put to be equal to this mean value. (This option is used, for instance, for RHIC

simulation when the coupling is introduced to have a round beam in the collision point. In the general case this check box has to be not checked.)

A few analytical models proposed for IBS growth rate calculation are realized in Betacool now. The choice between them is provided using *ComboBox* **IBS models** (Fig. 2).

2. Martini model

General theory of IBS process in a storage ring based on binary collision approach is described in the article M. Martini "Intrabeam scattering in the ACOOL-AA machines", CERN PS/84-9 AA, Geneva, May 1984. (this model is called **Martini** model in our program, in another sources it can be refer as "Extended Piwinski" model.).

This model is valid independently of the mode of a ring operation and does not require additional assumption about optic structure of the ring and Coulomb logarithm value. It is more accurate model and, if the calculation speed is not important for a solving task, it is preferable to use this model.

From the side of mathematics the growth rate calculation is reduced to evaluation in each optic element a 3D integrals over μ, ν and z of the following form:

$$f_i = k_i \int_0^{\infty}\int_0^{\pi}\int_0^{2\pi} \sin\mu\, g_i(\mu,\nu)\exp\left[-D(\mu,\nu)z\right]\ln\left(1+z^2\right)d\nu\, d\mu\, dz\ ,$$

$$D(\mu,\nu) = \frac{\left[\sin^2\mu\cos^2\nu + \sin^2\mu\left(a\sin\nu - \tilde{d}\cos\nu\right)^2 + b^2\cos^2\mu\right]}{c^2},$$

$$g_1(\mu,\nu) = 1 - 3\sin^2\mu\cos^2\nu\ ,$$

$$g_2(\mu,\nu) = 1 - 3\sin^2\mu\sin^2\nu + 6\tilde{d}\sin\mu\sin\nu\cos\nu/a\ ,$$

$$g_3(\mu,\nu) = 1 - 3\cos^2\mu\ .$$

here the coefficients k_i, a, b, c, d depend on the beam phase volume and lattice parameters of the optic element.

If in the combo box **Integral over z** (in the *TabSheet* **Martini** Fig. 3) the option **Numerical** is chosen then all three integrals are calculated using rectangular method. The interval of integration over each of variables is divided by a few equal steps and number of divisions over each of the variables is introduced in the corresponding edit window.

The integral over *z* has the infinite upper limit:

$$\int_0^{\infty}\exp\left[-D(\mu,\nu)z\right]\ln\left(1+z^2\right)dz$$

and in the program the upper limit is chosen corresponding to exponent power equal to -20.

Necessary number of steps over each variable depends on beam and ring parameters and required accuracy of the growth rate calculation. The numbers of steps has to be chosen manually by a few subsequent calculations of the growth rates. The numbers have to be increased until the growth rates reach saturation with required accuracy. (Algorithm for automatic choice of the step number is under development.)

To speed up the calculation the integral over *z* variable can be calculated using simplified expressions. If the option **Analytical** is chosen in combo box **Integral over z** the integral is calculated using analytical expression in accordance with the book by Abramowitz & Stegun.

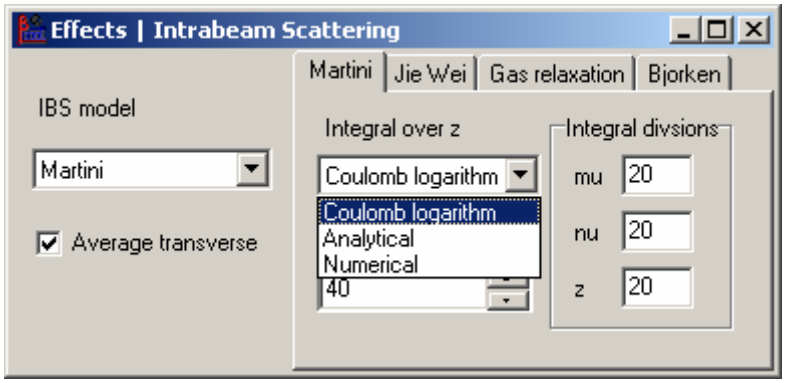


Fig. 3. The *visual form* for choice of the **IBS model**. *TabSheet* **Martini**. *ComboBox* **Integral over z**.

If the option Coulomb logarithm is chosen the integral is replaced by expression

$$\int_0^\infty \exp\left[-D(\mu,\nu)z\right]\ln\left(1+z^2\right)dz \approx \frac{2L_C}{D(\mu,\nu)},$$

where the value of $2L_C$ is introduced in the edit window Coulomb logarithm.

It was shown that in the last case the 2D integral over μ, ν variables can be reduced to 1D integral and the Martini model coincides with the model described in the article:

J.D. Bjorken, S.K. Mtingwa, "Intrabeam scattering", Particle Accelerators, Vol. 13, p.115, 1983.

3. Other models of IBS

The **Bjorken-Mtingwa** model is based on plasma physics approach. The growth rates are calculated as combinations of collision integral for three degrees of freedom. This model is realized as evaluation of 1D integral at infinite upper limit. To calculate the growth rates one needs to introduce Coulomb logarithm value (it should be estimated from the beam density before the calculation in Betacool, usual value is about 20), upper limit of the integral and number of the integration steps by rectangular method (Fig. 4). The two last values have to be chosen manually by a few subsequent calculations of the rates.

The Bjorken-Mtingwa model is accurate within the Coulomb logarithm accuracy.

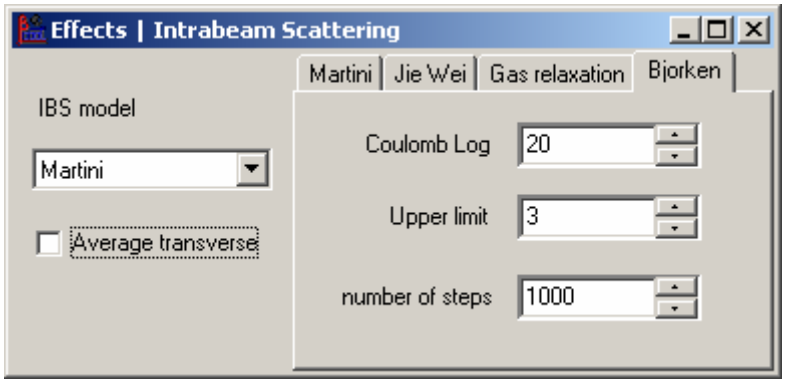


Fig. 4. The *visual form* for choice of the **IBS model. Bjorken** *TabSheet*.

The models **Piwinski, Jie Wei and Gas relaxation** (Fig.5) are valid **only** when a storage ring is operated **above transition energy**.

The model **Detailed** is used in **Model Beam** algorithm and for the rate calculation can not provide an accurate result.

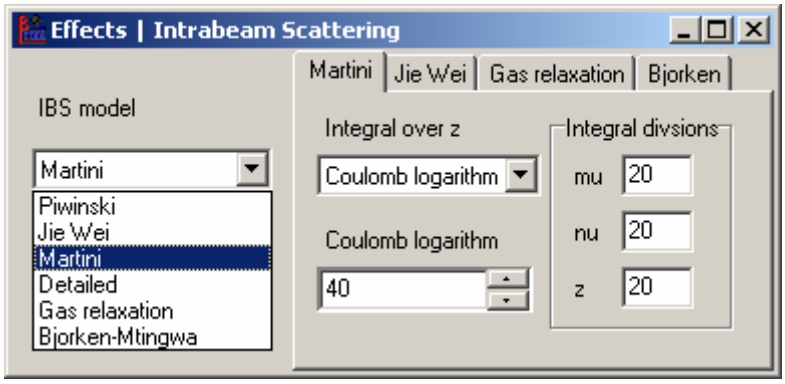


Fig. 5. The visual form for choice of the IBS model. *ComboBox* **IBS model**.

When alpha function and dispersion derivative are equal to zero the 3D Martini integrals can be reduced to 1D integral as it was shown for the first time in the article:

A.Piwinski, Proc. 9th Int. Conf. on High Energy Accelerators, p. 105, 1974.

This integral is calculated in Betacool with rectangular method (the model is called **Piwinski**). Number of the integration steps has to be input in the *TabSheet* **Martini** in the edit window **Integral divisions mu** (Fig. 5). This number has to be adjusted manually. Calculation of the IBS rates using **Piwinski** model does not require the ring optic structure. For the rate calculation the program uses mean lattice functions that output in the tab sheet **Mean params** of the **Ring | Parameters** *visual form* (Fig. 6).

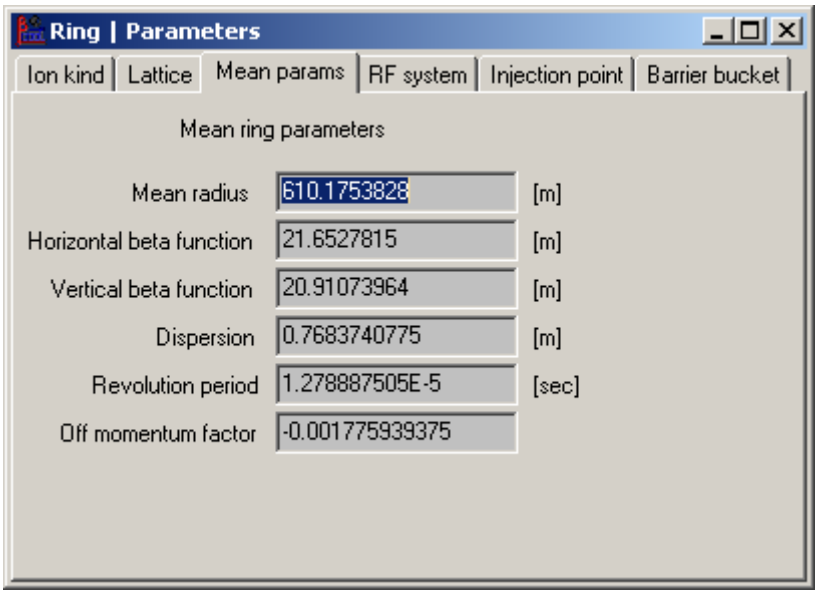


Fig. 6. Mean lattice parameters using for IBS rate calculation in the frame of **Piwinski** model.

Above the transition energy and under a few additional assumption the Martini integrals can be reduced to elementary functions as it was shown in

Jie Wei “Evolution of Hadron Beams under Intrabeam Scattering”, Proc. of PAC’1993, p.3651.

To calculate the growth rates in accordance with **Jie Wei** model one needs to input Coulomb logarithm value in the corresponding tab sheet (usually the log is about 20). Additional simplifications are related with high energy approximation and neglecting of dispersion.

The **Gas Relaxation** formula is based on additional assumption that the beam has a flattened velocity distribution so that the temperature of longitudinal degree of freedom is sufficiently less than transverse one.

4. Summary: how to provide a choice of IBS model?

Initially one needs to choose **Martini** model and **Integral over z – Numerical**. Thereafter by a few consequent calculations of the growth rates the **Integral divisions** have to be adjusted to required accuracy (typical number of integration steps can be a few hundreds for each variable if 1% accuracy is necessary).

(To calculate the rates one needs to use *menu item* **Task** *submenu item* **Growth Rates** and push the *button* **Open** at the *TabSheet* **Rates** of the corresponding visual form. See in details below.)

To estimate the Coulomb logarithm value required for other models one needs to chose **Integral over z – Coulomb logarithm** and adjust the log value to have the rates close to numerically calculated. The value in edit window **Coulomb logarithm** in **Martini** tab sheet has to be divided by 2 and thereafter it can be used as a Coulomb logarithm in **Jie Wei** or **Bjorken-Mtingwa** models.

In presence of the vertical dispersion in the ring one needs to use **Bjorken-Mtingwa** model. This model is realized in Betacool in accordance with its modification proposed in the article

M.Venturini, “Study of intrabeam scattering in low-energy electron rings”, Proceedings of the 2001 PAC, Chicago, pp. 2961 – 2963,

and the growth rates are calculated tacking into account the vertical dispersion and its derivative. To use this model one needs initially to estimate **Coulomb logarithm** value using **Martini** model.

If the ring is operated above transition energy, to speed up the calculations one can use **Jie Wei** or **Gas relaxation** model.

Piwinski model can be used when the ring is operated above the transition energy and this model does not require ring optic structure. This model can be used for preliminary estimations by the order of magnitude using mean ring parameters.

All other IBS models require to import a file containing the ring optic structure.

IX. Internal Target Effect

One needs to choose *MenuItem* **Effects|Internal Target** on the *MainForm* (Fig.1).

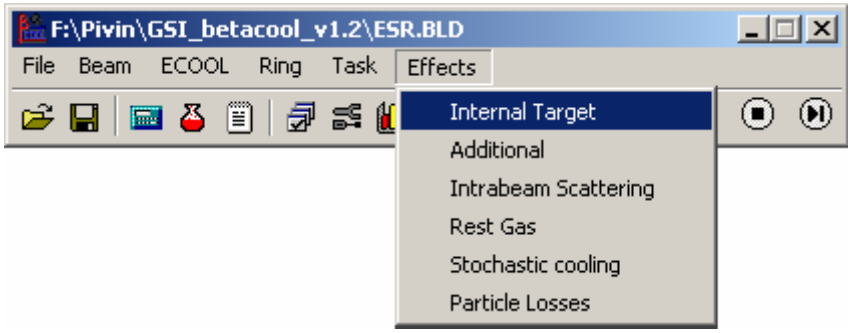


Fig.1 *MainForm.*

Window of the **Effects | Target** includes five *TabSheets*: **Material, Params, Lattice, Type**, **Pellet and Losses**.

1. Material of target

Effects | Internal Target | Materials
One needs to set **Mass number** and **Charge number** of the target atoms, **Length** and **Density** of the target on the *TabSsheet* **Material. Density** can be set as gram/cm^3 as atom/cm^3.

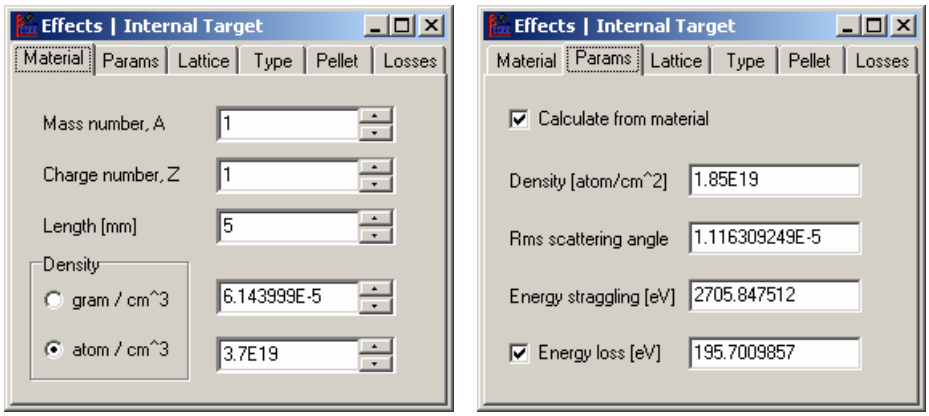


Fig.2 Material & Parameters

2. Parameters of internal target

TabSheet **Params** (Fig.2) shows parameter of the target which can be calculated from **Material** if *Checkbox* **Calculate from material** is switched ON. *Checkbox* **Energy loss** indicates usage of the energy loss value in the simulation. If one uses **Calculate from material** and wants to see value of **Density [atom/cm^2]** one should push *Button* on **Task|Growth Rates**. Sometime one needs to push *Button*  on the *Main Form* to redraw parameters.

3. Lattice functions

Effects | Internal Target | Lattice
TabSheet **Lattice** (Fig. 3) defines the lattice functions in the target position. Here the following parameters has to be set: **Horizontal** and **vertical** values for **beta**, **alpha**, **dispersion**, **dispersion derivative**.

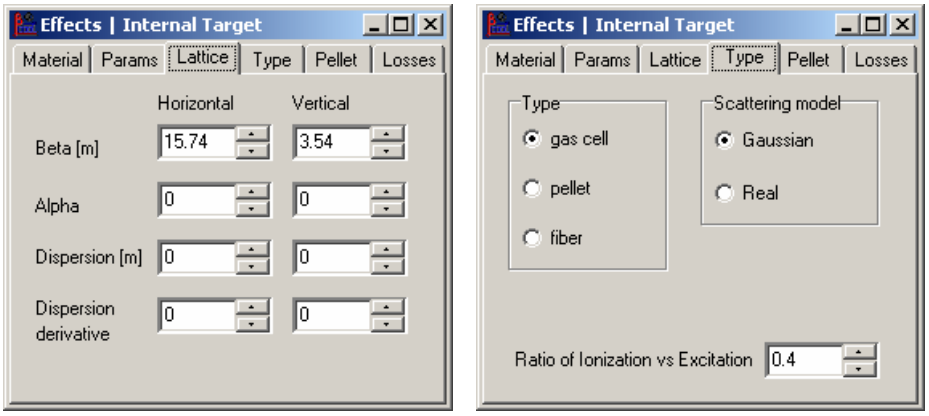


Fig. 3. **Lattices & Type**.

4. Target type

Effects | Internal Target | Type
TabSheet **Type** is used for choosing the following parameters:
Type of the target (**gas sell**, **pellet** or **fiber**): **gas cell** is the uniform gas cylinder, **pellet** is the flux of the macroparticles (Fig. 3).

Scattering Model defines distribution it the target **Gaussian** - according Gauss law, **Real** – uniform distribution; **Ratio of Ionization vs Excitation** defines parity between ionization and excitation energy losses.

5. Pellet target

Effects | Internal Target | Pellet
If type of the target will be **pellet** or **fiber** one has to set parameters on the *TabSheet* **Pellet**: **Horizontal shift**, **Flux width**, **Horizontal size**, **Vertical size**, **Vertical velosity** and **Interval between pellets**. Description of parameters is on Fig. 6.

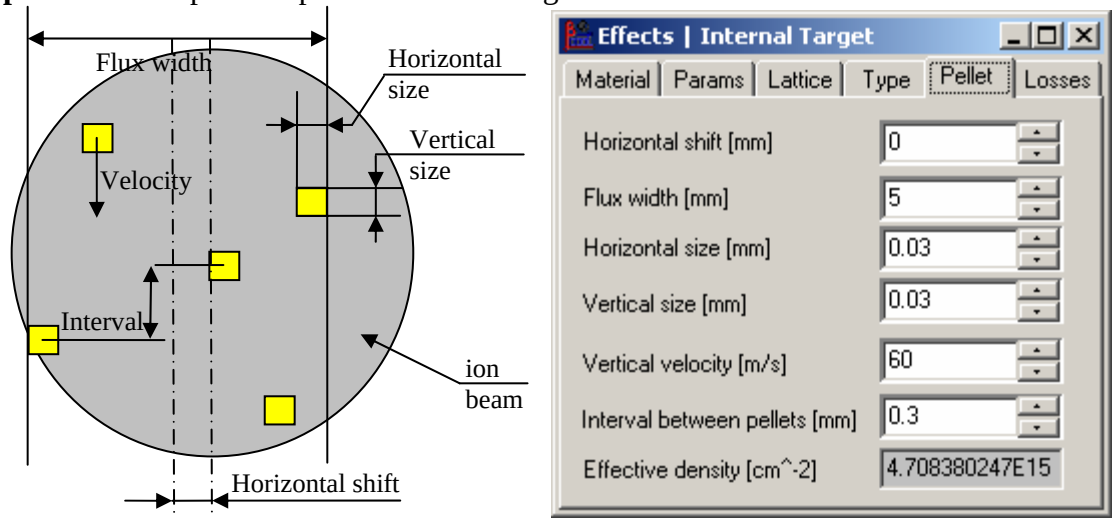


Fig. 4. **Pellet target** scheme and parameters.

6. Particle losses

Effects | Internal Target | Losses
TabSheet **Losses** (Fig. 5) gives a possibility to take into account particle losses on target interaction. There are 3 kind of losses are foreseen:

- **Electron capture** (recombination)
- **Single scattering**
- **Nuclear reactions**

Cross-section for probability of every event can be defined as a value in [barn]. If one wants to set **Cross-section** for probability it is necessary to set *CheckBox* **Interaction events** ON and to set value in *EditWindow* **Cross-section**.

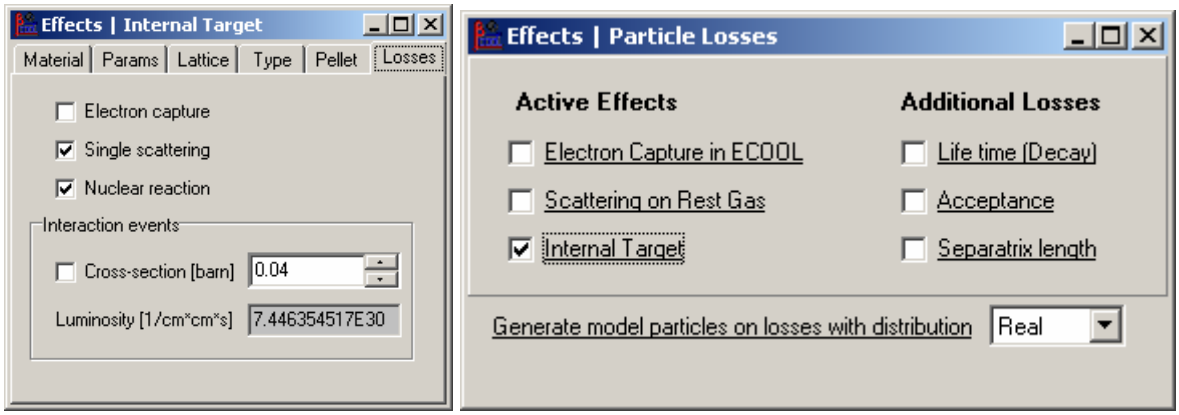


Fig.5. Particle losses in target

Note! One has to choose *MenuItem* **Effects|Particle Losses** (Fig.5) on the *MainForm* and switch ON **Internal Target,** to choose *MenuItem* **Task|Growth Rates** (Fig.6) and to switch ON **Particle Losses** if it necessary to consider particle losses effect in target. Information about **Generate model particles on losses with distribution** is given in **Particle Losses** manual.

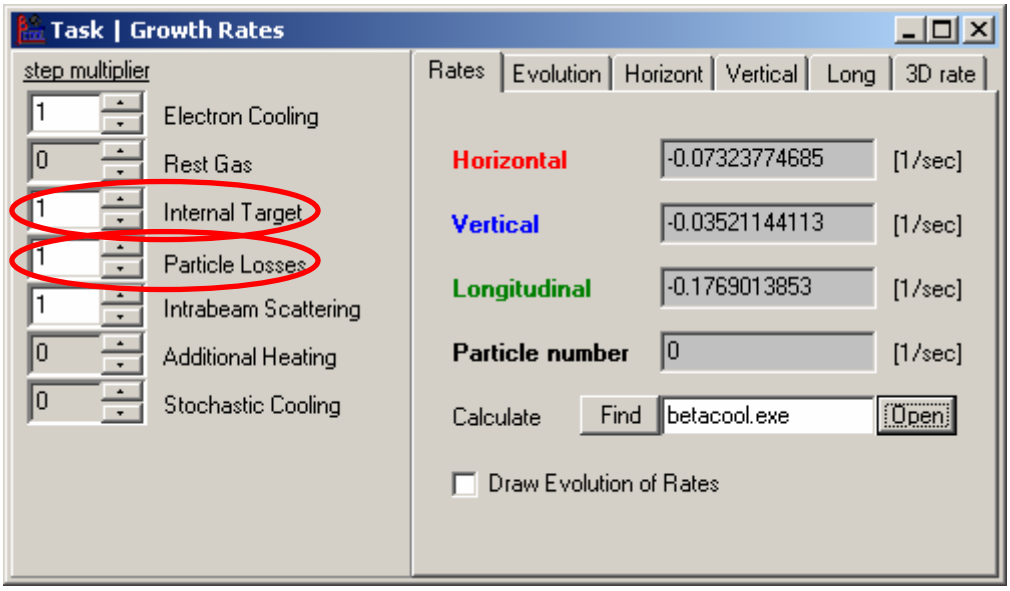


Fig. 6. Growth Rates calculation

Finally one has to choose *MenuItem* **Task|Growth Rates** (Fig. 6) and switch ON **Internal Target**.

X. Rest Gas

1. Model of Rest Gas

Growth rates due to the scattering on the residual gas are calculated with the same mechanism to the internal target effect (for example gas-jet target). In program assumed that residual gas is the gas cell target which is distributed along whole circumference of the ring. Lattice functions are used from each optics element. It means that summary rates due to the heating on the residual gas are integrated over the lattice structure. In BETACOOL program two objects need the real lattice structure: Intrabeam Scattering and Rest Gas.

Accordingly to the residual gas pressure the effective density of the gas is calculated, then energy loss on scattering with Bethe-Bloch formulae, r.m.s. angles of ions after scattering on atoms of the residual gas, and emittance and momentum deviation are calculated consequently. After in accordance with the percentage of components the characteristic growth rates are obtained. More detailed information is given in physical description of BETACOOL program.

Window of the **Effects | Rest Gas** (Fig.1) includes a kit of parameters characterized vacuum composition in the ring.

Effects | Rest Gas | Vacuum composition
Pressure defines vacuum pressure in Torr under room temperature. Four-component vacuum is presumed for now in the program. Below the above mentioned parameters the table of the vacuum components is situated. For every component provided the following kit of parameters: percentage of the component in a whole composition (**%**), Atomic number of the component (**A**) and Charge number correspondingly (**Z**).

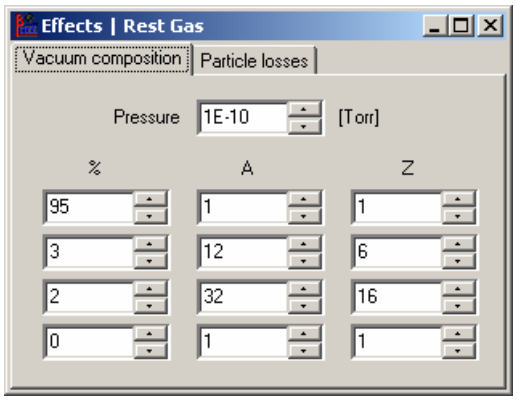


Fig.1. Form of Rest gas object.

2. Particle losses

Effects | Rest Gas | Particle losses
TabSheet **Particle Losses** (Fig.2) gives a possibility to take into account particle losses on target interaction (the same definition as for internal target: **Electron capture**, **Single scattering, Nuclear reactions.** Parameter **Mean vacuum chamber radius** is used for the calculation of the beam stability: average beam radius should be smaller than mean vacuum chamber. Do not forget check options **Scattering on Rest Gas** on the Form **Effects | Particle losses.**

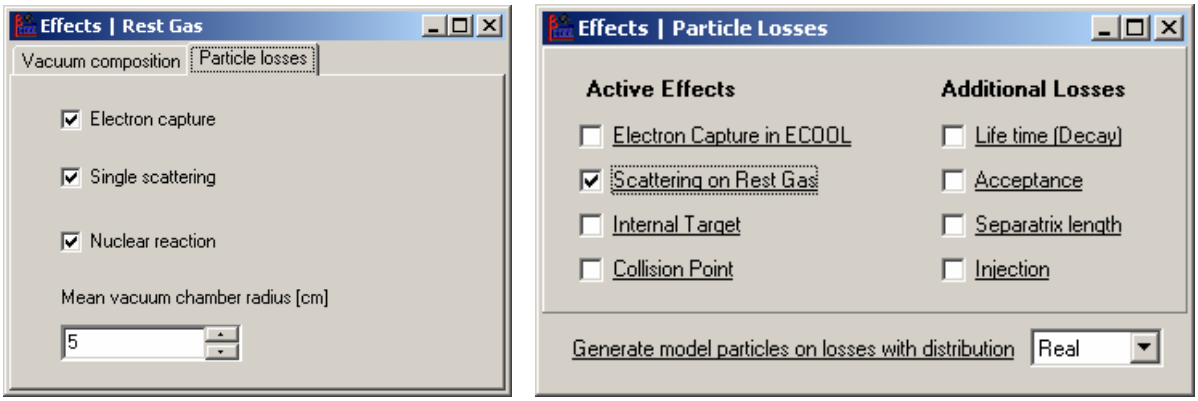


Fig.2. Particle losses parameters for Rest Gas.

XI. Electron cooler

Introduction

After launching the **Bolide.exe** file and checking a validity of an input file one needs to set the storage ring and beam parameters using corresponding menu items of the main interface window (Fig. 1).

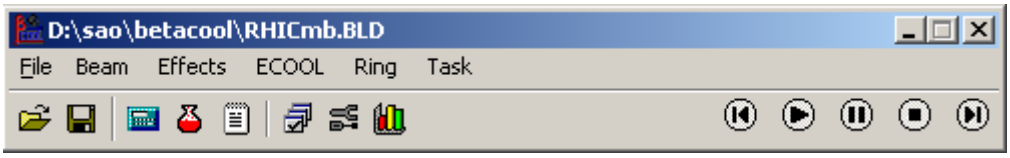


Fig 1. Main interface window

Electron cooling effect (thereinafter Ecool) implemented in BETACOOL is the most developed effect and has rather complicated structure. That's why it is excluded from the effect list (*Menu Item* **Effects**) and a special *Menu Group* **ECOOL** is dedicated to the setting of electron cooling. User must use this menu item and step by step to set parameters of the cooler, electron beam, friction force, etc. This procedure is given in details below.

As a matter of fact if user wants to switch ECOOL effect as active into simulation it is necessary to switch **step multiplier** corresponding to the **Electron cooling** on the *Form* **Task | Growth Rates** (Fig.2) to non-zero value (for details please look for Numerical algorithm manuals).

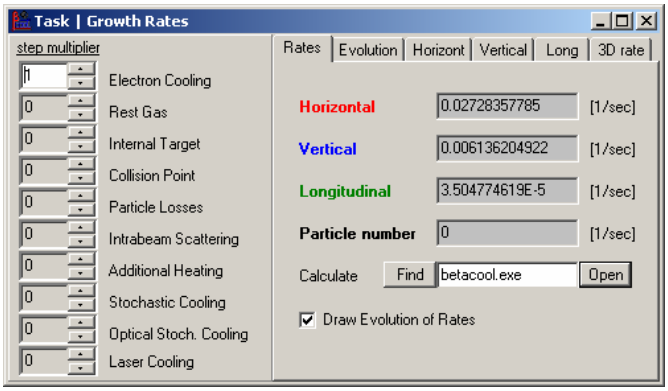


Fig.2 Effect switching panel.

One needs to choose *menu item* **Ring**, *submenu item* **Parameters**. For Ecool effect all the parameters have to be defined (depending on the algorithm chosen). How to set the Ring parameters please look [RING manual].

Parameters of six-dimensional phase volume of the beam required for Ecool rate calculation are set in the *TabSheet* **Emittance** of the **Beam | Parameters** *visual form*. For more detailed description of beam parameters definition please look [BEAM Manual].

1. Electron cooler

In this BETACOOL version electron cooling is treated as an **Effect** acting on the r.m.s. parameters of the ion distribution function and changing the ion number. Like any other **Effect** (**IBS**, **Additional Heating**, etc) the electron cooling model returns heating and loss rates what is a part of

the Effect library. Separate *menu item* **ECOOL** is intended for electron cooling due to complicated structure of this **Effect**. To calculate the cooling rates user needs to determine:

- models of the cooler, general parameters of the cooling section, ion beam;
- models of the electron beam;
- formulae for the friction force calculation from corresponding library;
- parameters of the tables of pre-calculated friction forces;
- plot tools for checking Ecool parameters validity.

To make the procedure clear these four steps are divided into four *submenu Items* and can be done independently using corresponding windows. Additional useful tool is developed for studying of cooling friction force behaviour. It is pointed in the submenu *Item* **Draw Forces.**

1.1. Cooler parameters

On the *TabSheet* **Cooler** of the **ECOOL | Cooler** Window (Fig. 3) using *ComBox* element **Cooler model** user must choose calculation model for the ion coordinates after crossing the cooling section: **Thin lens** model (only angle variation), or numerical integration of the ion motion equation. The numerical integration can be performed by two methods: Euler (**Euler method**) or 4-th order Runge Kutta (**RK method**). For the numerical integration one needs to determine the number of **integration steps** along the cooling section.

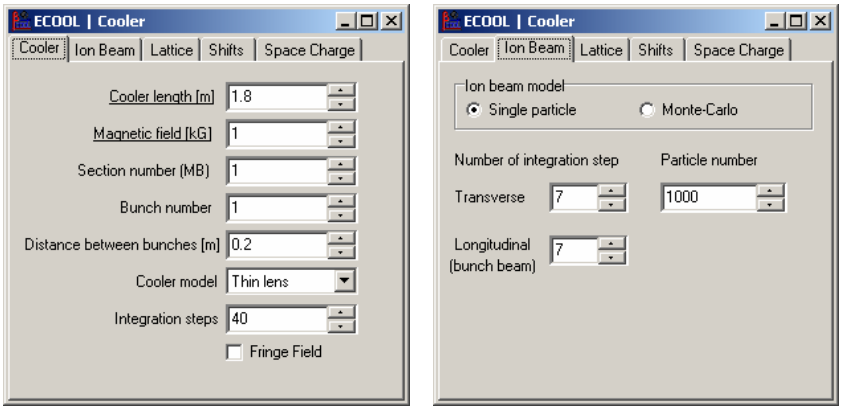


Fig. 3. Window of the **ECOOL | Cooler** menu item. *TabSheet* **Cooler & Ion beam.**

User must specify the following cooler parameters on this *TabSheet*: **cooler length**, **magnetic field magnitude**, **section number** (number of section of cooler - this parameter is used only when **Model Beam** algorithm. If to switch ON *RadioButton* **Fringe Fields** then effect of fringe fields of cooling solenoid will be taken into account for calculations.

1.2. Ion beam model

The cooling rates can be calculated for r.m.s. particle using **Single particle model** using settings on the *TabSheet* **Ion beam** of the **ECOOL | Model** Window (Fig. 3).

Rates are calculated by averaging over phases of betatron and synchrotron oscillations. Numbers of integration steps over the phases are input in corresponding *EditWindows* **Transverse** and **Longitudinal**.

1.3. Lattice parameters

The lattice parameters at the cooler position must be defined on *TabSheet* **Lattice** (Fig.4): **Beta** (Horizontal / Vertical), **Alpha** (Horizontal / Vertical), **Dispersion** (Horizontal / Vertical), **Dispersion derivative** (Horizontal / Vertical).

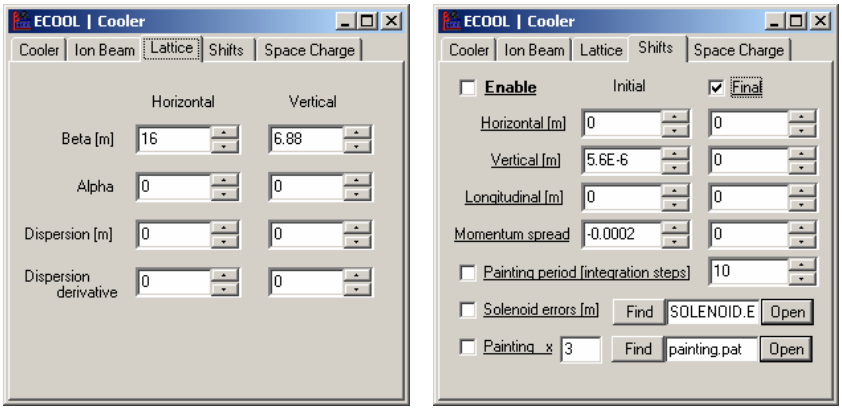


Fig. 4. **ECOOL | Cooler** *menu item*. *TabSheet* **Lattice & Shifts**.

1.4. Electron beam shifts

In BETACOOL code was realized different procedures for the changing of the electron beam position in transverse and longitudinal plans, the distance between electron and ion bunches, solenoid errors and so on. Parameters for these procedures were placed on different windows and sometime duplicate each other. Now all these possibilities are placed on the same *TabSheet* **Shifts** on **ECOOL | Cooler** *Form* (Fig.4).

The definition for the electron beam shifts in the laboratory rest frame is written as following:

$$\begin{aligned}
\Delta x' &= \left(\Delta x_{fin} - \Delta x_{ini}\right)/l \\
\Delta x &= \Delta x_{ini} + \Delta x' h \\
\Delta y' &= \left(\Delta y_{fin} - \Delta y_{ini}\right)/l \\
\Delta y &= \Delta y_{ini} + \Delta y' h \\
\Delta\left(dp/p\right) &= \Delta\left(dp/p\right)_{ini} \\
\Delta s &= \Delta s_{ini}
\end{aligned} \qquad (2.1)$$

where $\left(\Delta x, \Delta x', \Delta y, \Delta y', \Delta s, \Delta(dp/p)\right)$ - vector of current shifts, $\left(\Delta x_{ini}, \Delta x'_{ini}, \Delta y_{ini}, \Delta y'_{ini}, \Delta s_{ini}, \Delta(dp/p)_{ini}\right)$ and $\left(\Delta x_{fin}, \Delta x'_{fin}, \Delta y_{fin}, \Delta y'_{fin}, \Delta s_{fin}, \Delta(dp/p)_{fin}\right)$ - vectors of initial and final shifts, l – distance between points of shift, h – current position of ion.

If **Solenoid error** option is not enabled then $l = L_{ecool}$ and $h = L_{ecool} / 2$, L_{ecool} – electron cooler length. If **Solenoid errors** option is enabled then l is the length of magnetic field unhomogeneity which calculated from input data file *.err with field errors, h is equal to difference between ion longitudinal coordinate and position of the correspondence field unhomogeneity. This file has 3 columns which correspond to the longitudinal coordinate, horizontal and vertical shifts along the longitudinal axis. Note that the **Solenoid errors** option will work properly if **Euler** or **Runge-Kutta** cooler model is chosen on the *TabSheet* **Cooler**.

If parameter **Final** is not enabled then final vector of shifts is equal to initial vector. The painting procedure is used when the parameter **Painting period** in enabled:

$$(\Delta x, \Delta x', \Delta y, \Delta y', \Delta s, \Delta(dp/p)) = (\Delta x, \Delta x', \Delta y, \Delta y', \Delta s, \Delta(dp/p)) \times R, \qquad (2.2)$$

where R is the remainder of integer division N_{step}/P_{step}, N_{step} – step number of beam simulation, P_{step} – parameter of **Painting period**. If **Final** parameters is enabled for painting procedure then longitudinal coordinate and momentum spread shifts are calculated as

$$\begin{aligned} \Delta(dp/p) &= \Delta(dp/p)_{ini} + \left[\Delta(dp/p)_{fin} - \Delta(dp/p)_{ini}\right] \times R \\ \Delta s &= \Delta s_{ini} + \left(\Delta s_{fin} - \Delta s_{ini}\right) \times R \end{aligned} \qquad (2.3)$$

Electron beam shifts can be read from the file for the painting procedure if parameters Painting is enabled. This file includes 6 columns which correspond to vector of current shifts. Number of rows equal to the period of the painting procedure and each row corresponds to the current step of the integration process. Scaling parameters **Painting x** __ can increase (positive value) or decrease (negative value) a speed of the painting procedure. For example, if scaling parameters equal 2 it means that only each second row is used in the painting procedure. If scaling parameter equal -2 it means that each row is used twice. User can change all parameters on the *TabSheet* **Shifts** during simulation and even change data filenames with solenoid errors or painting procedure.

1.5. Space charge of electron beam

TabSheet **ECOOL | Cooler** | **Space charge** (Fig. 5) gives a possibility to draw beam potential distribution and real particle distribution (for model beam). To launch the visualization drawing user can use *TBrowse* component Redraw Space Charge. If it is necessary to monitor paricle dynamics on-line – there is a *CheckBox* Show particle dynamics which must be ON in that case.

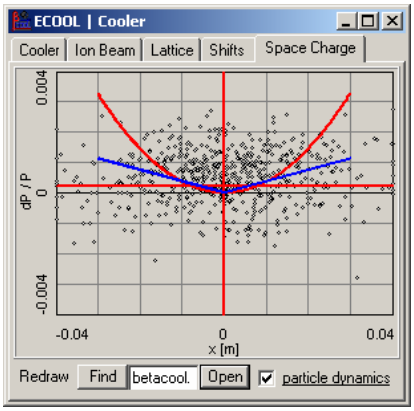


Fig. 5. **ECOOL | Parameters** *menu item. TabSheet* **Space Charge**.

2. Electron beam models

General parameters of the cooling section must be set using *Form* **ECOOL | Electron beam.** This Form contains the following *TabSheets*: **Uniform cylinder, Hollow beam, Gaussian cylinder, Parabolic, From File.**

2.1. Uniform cylinder

On the *TabSheet* **Uniform Cylinder** (Fig. 6) user must choose model of the electron beam using *ComboBox* **Electron beam model (there are the following models implemented:** DC cylindrical electron beam with uniform electron density, DC electron beam with elliptic cross-section and Gaussian distribution in the transverse plane, Hollow electron beam with determined densities in the center and edges) and define parameters for the corresponding beam model. If **Uniform cylinder** chosen the current form contains *EditWindows* for its setting (electron **Beam radius**, electron **Beam current**, **V_tr gradient** - derivative of the transverse temperature over radius). Neutralization factor (it has to be positive and < 1).

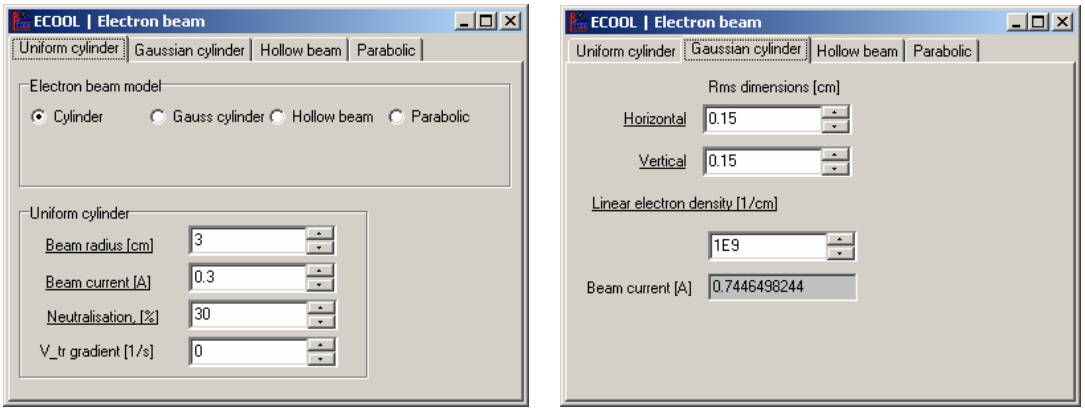


Fig. 6. **ECOOL | Electron beam**. *TabSheet* **Uniform Cylinder** & **Gaussian Cylinder**.

2.2. Gaussian cylinder

If DC electron beam with elliptic cross-section and Gaussian distribution in the transverse plane is chosen user must use *TabSheet* **Gaussian cylinder** (Fig. 6) to set parameters: **Horizontal** rms dimension, **Vertical** rms dimension and Linear electron density. **Beam current** will be automatically calculated.

2.3. Hollow beam

If hollow beam is chosen user must use *TabSheet* **Hollow beam** (Fig. 7) to set parameters: **Hole radius**, **Density in the hole**, **Beam radius**, **Density in circle**. **Beam current** will be automatically calculated. To take into account space charge effect inside electron beam user must Check **Beam current**.

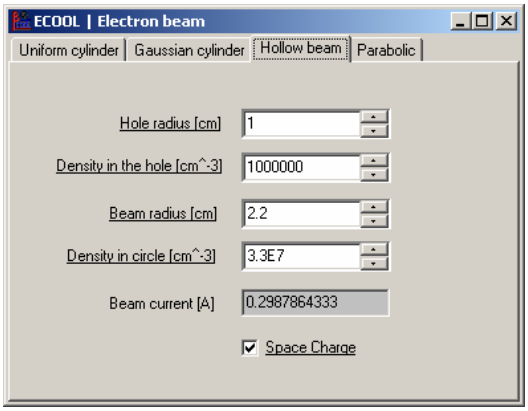


Fig. 7. **ECOOL | Electron beam**. *TabSheets* **Hollow beam** and **Parabolic**.

2.4. Parabolic beam

If parabolic beam is chosen user must use *TabSheet* **Parabolic beam** (Fig.8) to set parameters: R**adius**, **Current, V_tr gradient**. **Current density** will be automatically calculated.

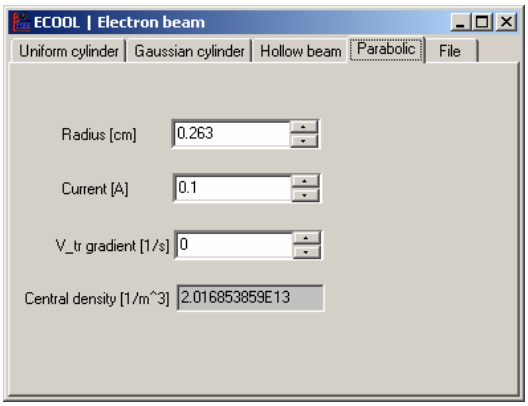
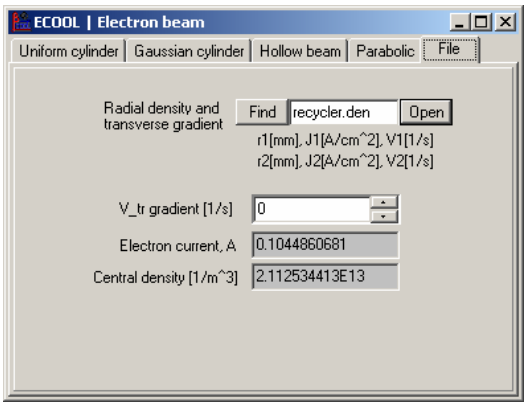


Fig. 8. **ECOOL | Electron beam**. *TabSheets* **Parabolic**.

2.5. From file

The distribution of the electron beam on the radius can be calculated in other program and result of the calculation can be read into BETACOOL program as text file. *TabSheet* **File** for reading of the external file with radial distribution of electron density and transverse gradient is shown on Fig.9a. Fist column of the file is the radial coordinate in mm, second column is the electron density in A/cm^2, and third column is the transverse gradient $G_\perp$ in 1/sec. If the third column is absent in the file then the constant transverse gradient will be used from window (Fig.8). Example of the text file with the electron beam density is shown on Fig.9b.



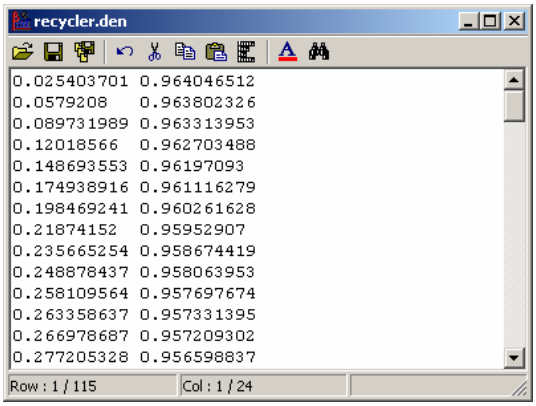


Fig. 9. **ECOOL | Electron beam**. *TabSheets* **File** and example of input file

3. Library of the friction forces

3.1. Models of friction forces

The friction force components acting on the ion inside the electron beam can be calculated using different analytic formulae and, in principle, using results of numerical calculations. Choice between different presentation of the friction force is provided by *ComboBox* **Friction Force Model** of the *TabSheet* **Model** of the **ECOOL | Friction force** menu *Item* window (Fig.10). For the

moment the following formulae realized in the program: **Budker formula, Non-magnetized, Derbenev-Skrinsky-Meshkov, Parkhomchuk, Erlangen, Tabulated, 3D (for electron Array).**
When user chooses a model for friction force calculation some necessary parameters can be determined in corresponded *EditWindows* of *TabSheets* on the current *Form*. If **Tabulated** model is chosen then user must indicate file with tabulated values of friction force component pre-calculated by another program (see **ECOOL | Tabulated**). At the *TabSheet* **Model** electron beam characteristics (quality) can be selected to be represented as one of the following corresponding parameters: transverse and longitudinal emittance, temperature or rms velocity. User must set the appropriate values in selected pair of parameters.

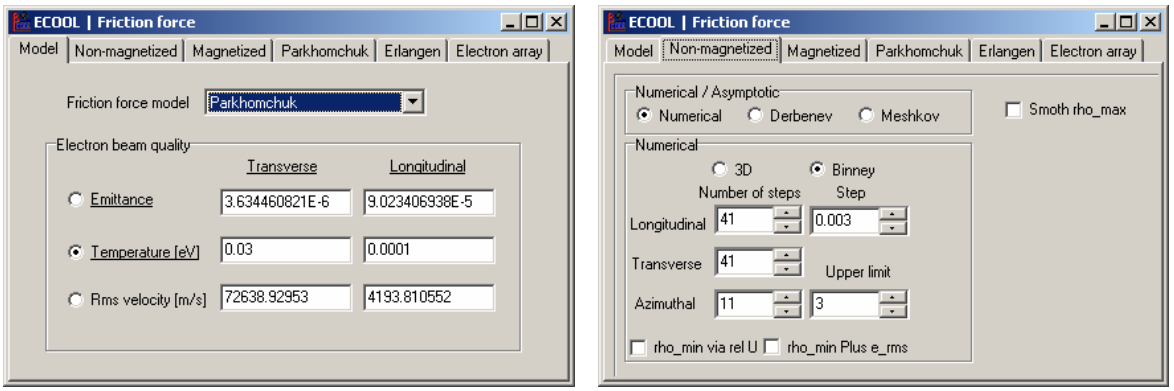


Fig. 10. **ECOOL | Friction force**. *TabSheet* **Model & Non-magnetized**.

3.2. Non-magnetized fiction force

The *TabSheet* **ECOOL | Friction force** | **Non Magnetized** (Fig.10) is used to determine and define parameters for Non-magnetized friction force model calculation. Here user can change the following parameters on the corresponding *Panel*:
RadioButton **Asimptotic** or **Numerical** – choice of the calculation method for this model – either using asymptotic assumption (asymptotic formulae obtained with Coulomb analogy are used – formalism derived either by **Derbenev** or by **Meshkov**), or integrating over longitudinal and transverse ion velocities – this is **Numerical** choice. If **Numerical** calculation is chosen – here user can select method for numerical integration with *RadioGroup* **Numerical**: either **3D** or using **Binney** formalism. If integration will be performed over all three velocities (two transverse and one azimuthal) – then user have to define **Number of steps** for each integral.

The following *CheckBoxes* if switched on are intended for definition of integration parameters:
- Smooth rho_max - is used for maximal impact parameter calculation (either via plasma period or time of flight);
- rho_min via rel U - minimal impact parameter is calculated via RMS electron velocity spread or velocity of individual electron;
- rho_min Plus e_rms - minimal impact parameter can be calculated either via sum of electron and ion velocities or via ion velocity.

3.3. Magnetized model of friction force

The *TabSheet* **ECOOL | Friction force** | **Magnetized** (Fig.11) is used to determine and define parameters for the Magnetized friction force model calculation. Here user can change the following parameters on the corresponding *Panel*. When **Derbenev-Skrinsky-Meshkov** model is chosen, user can choose calculation method for it's realization – **Numerical** or **Asymptotic** one and define so-called **Smoothing coefficient -** coefficient proposed by Meshkov to smooth the friction shape.

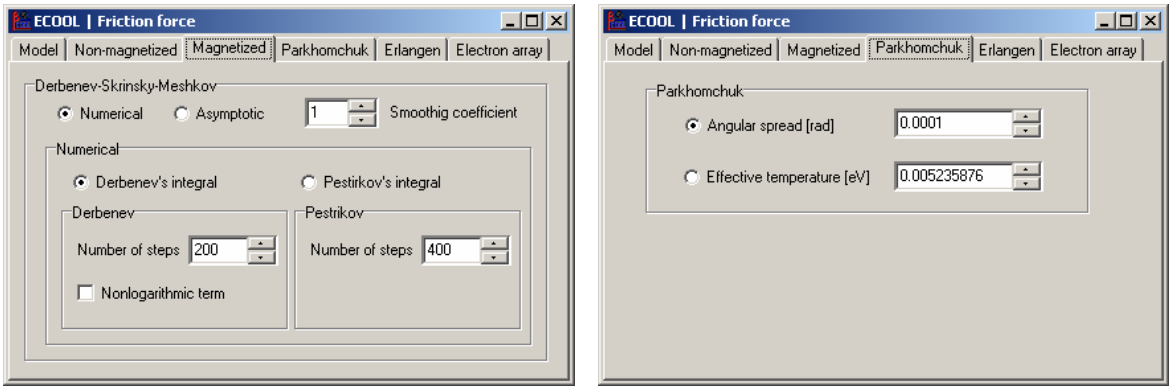


Fig. 11. **ECOOL | Friction force**. *TabSheet* **Magnetized**.

When **Numerical** integration is chosen here user also has a choice which formalism to select for integration *Item* of *RadioButton* – **Derbenev's integral** or **Pestrikov's integral**. For each integration method user has to determine **Number of steps**. If *CheckBox* **Nonlogarithmic term** is switched ON then plasma wave irradiation is taken into account.

3.4. Parkhomchuk's model

If **Parkhomchuk** model for the friction force calculation is chosen, user via **Parkhomchuk** *TabSheet* (Fig. 11) can define parameter **Effective temperature [eV]** (or it can be interpreted as **Angular spread [rad]**) which is used in Parhomchuk formula.

3.5. Erlangen model

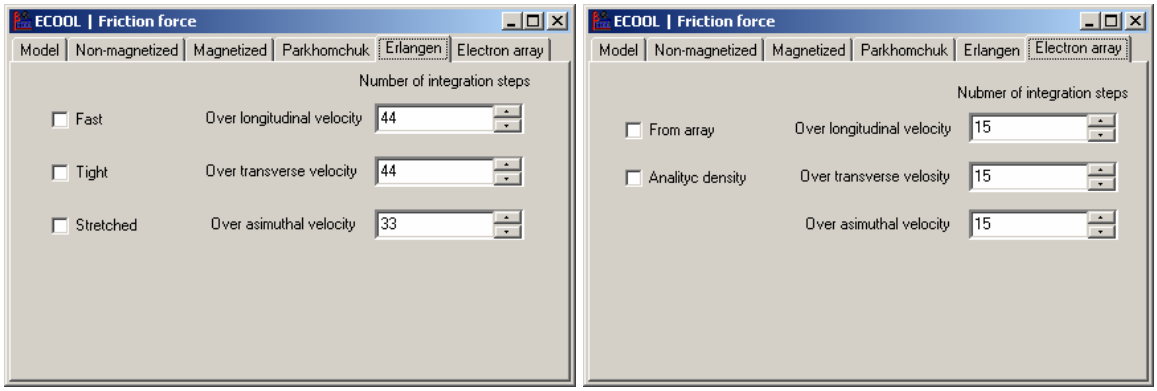


Fig. 12. **ECOOL | Friction force**. *TabSheet* **Erlangen & Electron array**.

TabSheet **ECOOL | Friction force** | **Erlangen, 3D** (Fig. 12) lets user to specify parameters for the model of friction force calculation developed by group from Erlangen University. This model has algorithm which combine calculation of the friction force for both magnetized and non-magnetized (fast) collisions. On the *TabSheet* there are 3 *CheckBoxes* presented: **Fast**, **Tight**, **Stretched**. When any combination of them is switched ON – different options of calculation are taken into account in algorithm. If to take into account during calculations fast collisions – user has to check **Fast** *CheckBox*. During magnetized interaction there are 2 models of the electron Larmour spiral to be presented – as a **Tight** helices and **Stretched** helices. Also there are 3 *EditWindows* for definition of the number of integration steps over velocities during numerical calculation of the friction force value: **over longitudinal velocity**, **over transverse velocity**, **over azimuth** are presented.

3.6. Electron array

TabSheet **ECOOL | Friction force | Electron array** (Fig.12)**.** *EditWindows* for integration step numbers are presented which are parameters of the 3D friction force calculation for the case when electron beam is presented as an **Array of particles**. Here are number of integration steps over velocities can be defined: **over longitudinal velocity**, **over horizontal velocity** and **over vertical velocity**.

4. Using of Friction forces drawing tool

Form **ECOOL | Draw Forces** (Fig. 13) is a toolkit for checking 3D shape of friction force. There are several TabSheets for controlling visualization process: **Control, Transverse, Longitudinal, 1-D force, Force along angle.**

4.1. Control of friction force drawing

TabSheet **ECOOL | Draw forces | Control** (Fig. 19) lets user to determine parameters of the mesh and boundaries for the range where friction force will be calculated and drawn. The model of the friction force to be drawn and calculated has to be chosen on the form **ECOOL | Friction force | Model** (Fig.8)**.**

Edit windows here are used for setting the boundaries of transverse and longitudinal ion velocities (minimum and maximum values) and division number in every range. User has to set the following parameters: In the program code they are following variables:

- boundaries for transverse velocity range: **transverse velocity minimum** value and **maximum** value;
- boundaries for longitudinal velocity range: **long. velocity minimum** value and **maximum** value;
- number of divisions in the velocity range: **Divisions**;
- boundaries for velocity range to draw a plot of 1-D force (Fig.21): **Velocity minimum** value and **maximum** value, and number of **Divisions** for 1-D force plot;
- angle of ion velocity inclination: **Angle**;
- boundaries for velocity range: **Velocity**;
- number of divisions for the plot Force(angle): **Divisions**.

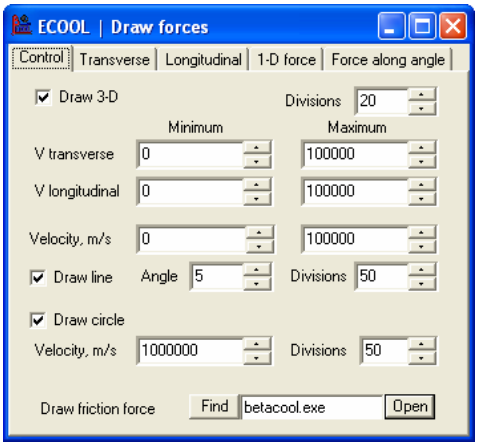


Fig.13. **ECOOL | Draw forces**. *TabSheet* **Control**.

4.2. Transverse and longitudinal components of friction forces

Component *TBrowse* **Draw friction force** is used for visualization of the friction force dependence on ion velocity. Button **Open** of the *TBrowse* starts BETACOOL program with parameter **/fr**. At this parameter BETACOOL calculates and saves two 3D plots: **FFtr.sur** and **FFlong.sur**. The 3D plots are loaded and visualized into corresponding tab sheets **Transverse** and **Longitudinal** of the **ECOOL | Friction force** *Form* (see example of the friction force 3D plot in the Fig.14).

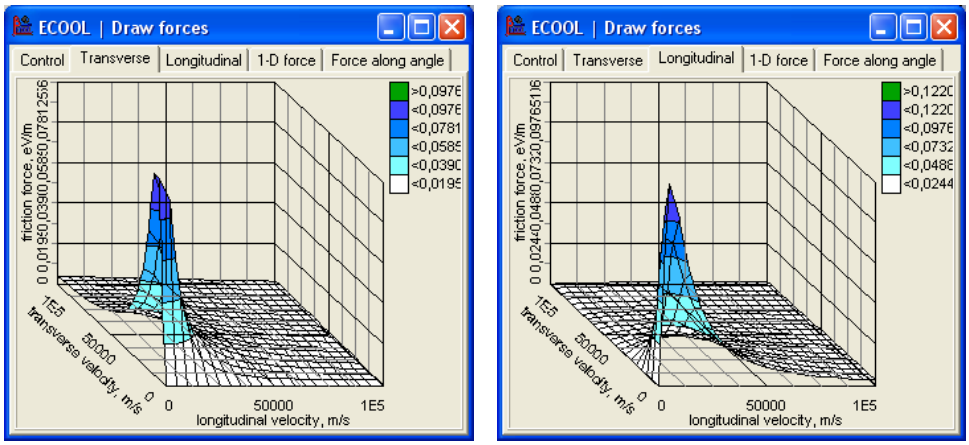


Fig.14. Example of the friction force shape calculation.
ECOOL | Draw forces | Transverse & **Longitudinal.**

4.3. 2-D plots of friction forces

The friction force is performed in [eV/m] units, the velocity components are measured in [m/sec]. Another two *TabSheets* are: **1-D Force** (Fig.15) which is intended to draw 2-D plot of the friction force component [eV/m] over velocity in pre-determined range and **Force along angle** (Fig.15) – here user can plot friction force component [eV/m] when the ion velocity directed along specified angle.

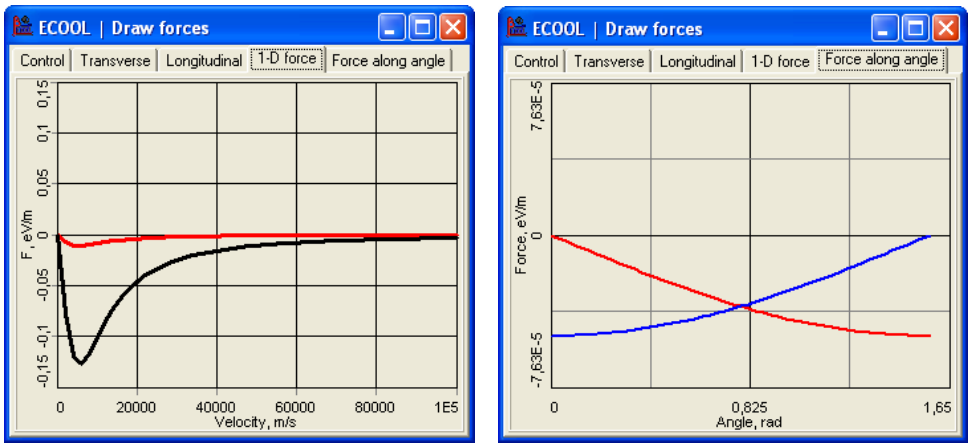


Fig. 15. Example of 2-D friction force line over velocity calculation.
ECOOL | Draw forces | 1D-Force & Force along angle.

5. Tabulated friction force

Window **ECOOL | Tabulated** is a special toolkit for testing and interpolation of tables with pre-calculated transverse and longitudinal friction force values which are created by another code (for example tables generated by program created in Erlangen Univ.).

5.1. Input files

TabSheet **Control** (Fig. 16) – here 3 *TBrowse* components – two for choosing files containing tables with transverse and longitudinal friction force values: **File with transverse velocity table, File with longitudinal velocity table**. These files have to have special extensions: for transverse component - *.tvt, for longitudinal component *.lvt. And one *TBrowse* component **Generate table with velocities.** Here user must launch betacool.exe to generate tables with friction force values in accordance with mesh parameters and interpolation method which are defined in next Tab Sheets on this window;

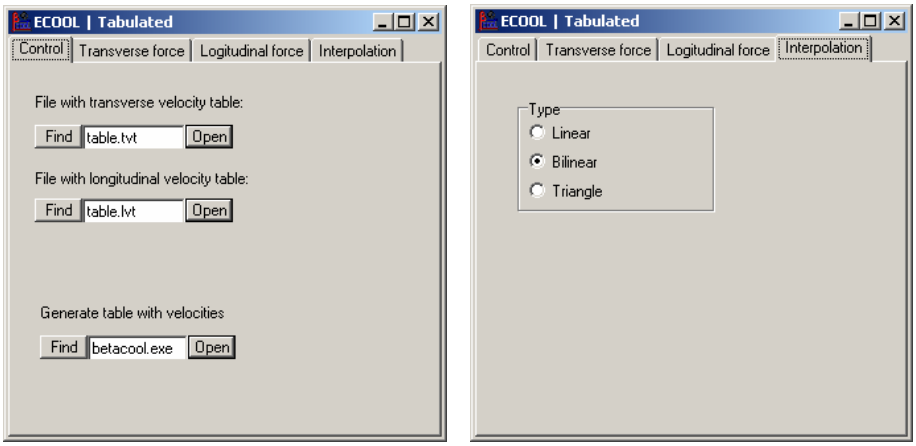


Fig. 16. Window of the **ECOOL | Tabulated** menu item. *TabSheet* **Control**.

5.2. Interpolation

TabSheet **Interpolation** (Fig. 16) is used for definition of the interpolation method for processing of the tables with pre-calculated friction force values. Here *RadioGroup* **Type** is presented and user can choose one of three possible interpolations:

- **Linear** (linear interpolation – closest nodes. For uniform mesh)
- **Bilinear** (non-linear interpolation. For uniform mesh)
- **Triangle** (non-linear interpolation. For arbitrary non-uniform mesh).

5.3. Transverse and longitudinal velocities

TabSheet **Transverse Force** (Fig 15) is used for definition of the parameters of the table for transverse component of the friction force. This table is a mesh with friction force values in nodes which is generated versus transverse and longitudinal ion velocity values. Mesh for this table has 3 ranges for every velocity with independent splitting in every range. Parameters of this mesh are presented on this Tab Sheet.

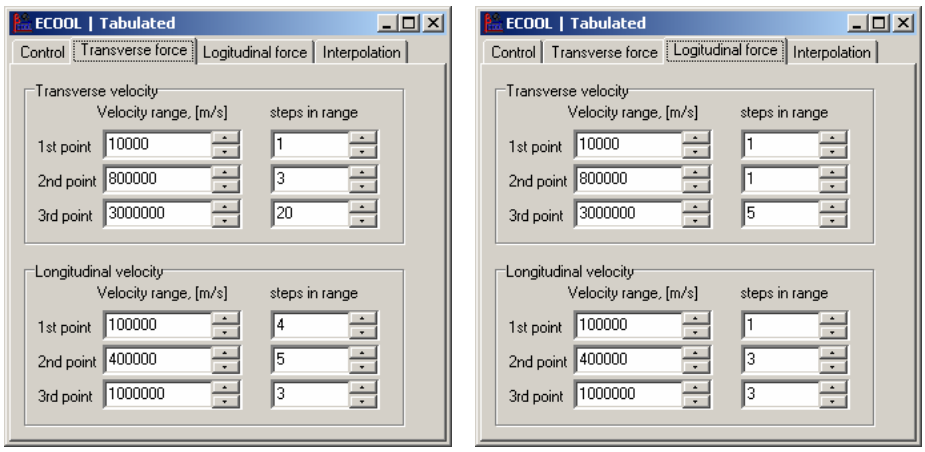


Fig. 17. *TabSheets* **Transverse force, Longitudinal force**.

First range is from 0 to **1st point** value with splitting **steps in range**, second range is from **1st point** to **2nd point** with splitting **steps in range** corresponding to **2nd point**, third range is from **2nd** point to **3rd** point with splitting **steps in range** corresponding to **3rd** point.
TabSheet **Longitudinal Force** (Fig 15) is used for definition of the parameters of the table for longitudinal component of the friction force. All the parameters are analogue to *TabSheet* **Transverse Force**.

General note:

The algorithm of using this **ECOOL | Tabulated** is the following. If user wants to create a table by himself using any of existing friction force models in BETACOOL it is necessary to: 1) choose a **model** in **Ecool | Friction force| Model** (any but **Tabulated**), 2) set up the mesh (**Ecool | Tabulated | Transverse & Longitudinal** – Fig.17), 3) Push *button* **Open** at *TBrowse* **Generate table with velocities**. As a result 2 files (table.lvt and table.tvt) will be automatically created in the current folder.

If user want to use pre-generated tables for electron cooling effect calculation or for drawing the 3D friction force shape it is necessary to: 1) choose the model **Tabulated** in **Ecool | Friction force| Model**, 2) Specify 2 files with tables using at **Ecool | Tabulated | Control** *TBrowse* components for files 3) Choose at **Ecool | Tabulated | Interpolation** a method for interpolation for calculation of friction force values using the table.

XII. Stochastic cooling

1. Standard Stochastic cooling

1.1. Transverse degrees of freedom

For each transverse plane the program uses standard set of input parameters for the cooling chain description (Fig.1), that are listed in the Table 1.

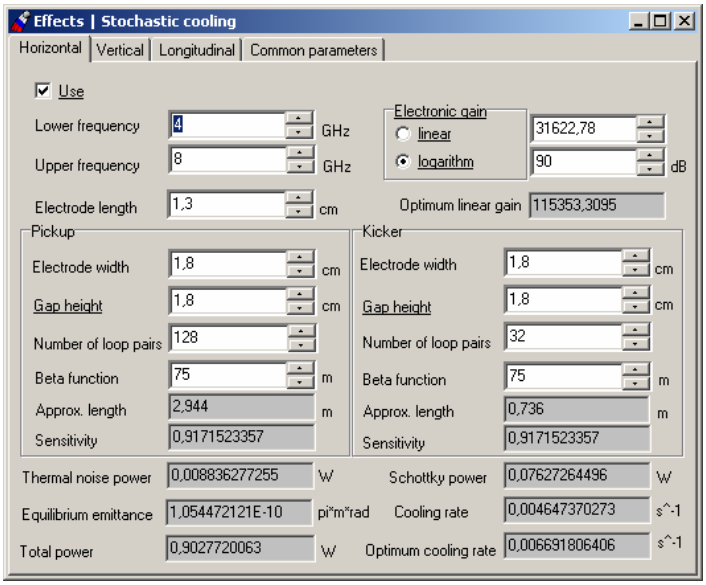


Fig.1. Visual form for input and output parameters for transverse cooling chain.

Table 1. Input parameters for the cooling chain description

Variable	Notation in the report	Dimension
Lower frequency	$f_{\min}$	GHz
Upper frequency	$f_{\max}$	GHz
Electronic Gain	G_A	Dimensionless or in dB: $G_A[dB] = 20\log G_{A,lin}$
Loop length	l_{loop}	cm
Pickup and kicker parameters		
Electrode width	w_p, w_k	cm
Gap height	h_p, h_k	cm
Number of loop pairs	n_p, n_k	-
Beta function in the pickup and kicker positions	β_p, β_k	m

For the pickup and kicker the program calculates the following output parameters:
- sensitivity - $\sigma_{p,k}/2$ in accordance with the formula (G.IV.2.8);
- approximate length of the electrodes in accordance with $l_{electrode} = n_{p,k}\left(l_{loop} + 1cm\right)$.

For the equilibrium emittance, cooling rate and consumption power calculation the program uses the following input parameters common for both transverse planes (Fig.2, Table 2)

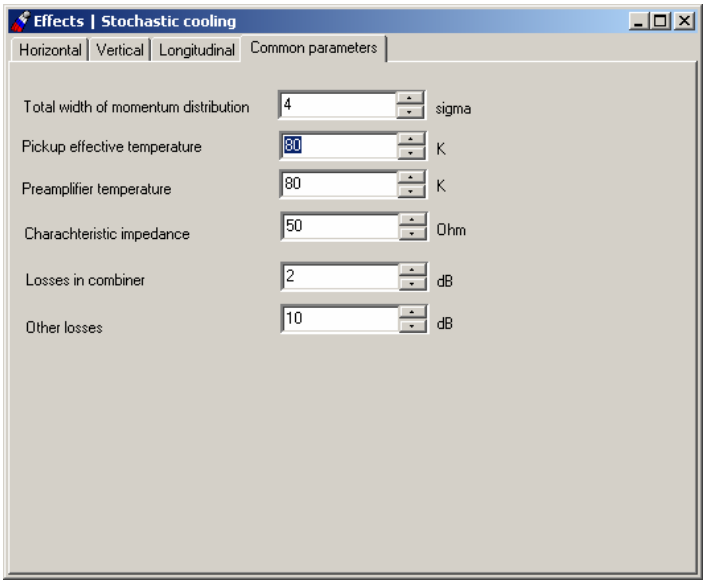


Fig.2. Visual form for input parameters for power consumption calculation.

Table 2. Input parameters for the cooling rate calculation

Variable	Notation in the report	Dimension
Total width of the beam momentum distribution	δ	Number of r.m.s.
Pickup effective temperature	T_A	K
Preamplifier temperature	T_R	K
Losses in combiner	P_{comb}	dB
Power of additional losses	P_{loss}	dB
Characteristic impedance	Z	Ohm

To control a validity of input data the program outputs the parameters listed in the Table 3. These parameters can be calculated without beam dynamics simulation: for this one needs to calculate sum of the rates of active effects.

Table 3. Output parameters

Variable	Notation in the report	Dimension
Thermal noise power	P_{th}	W
Schottky power	P_S	W
Total power	P_{tot}	W
Equilibrium emittance	ε_∞	π·m·rad
Cooling rate	$1/\tau_{cool}$	s^{-1}

1.2. Longitudinal degree of freedom

For longitudinal plane (Fig.3) the program uses standard set of input parameters for the cooling chain description, that are listed in the Table 4.

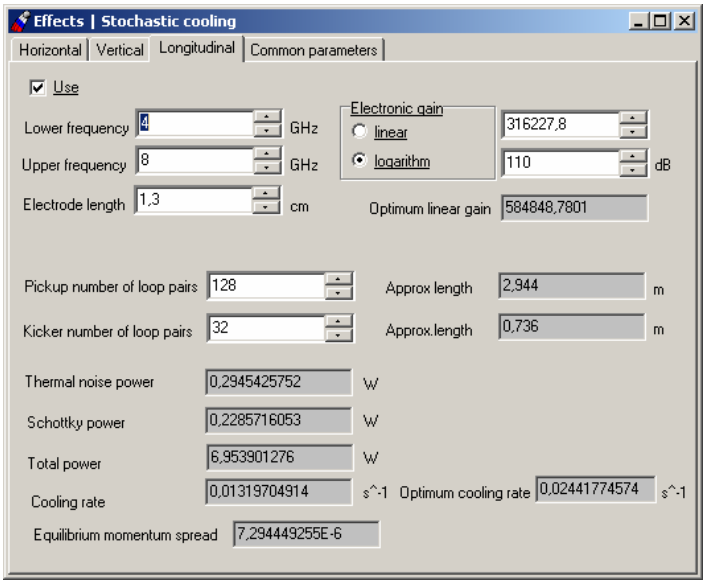


Fig. 3.2. Visual form for input and output parameters for longitudinal cooling chain.

Table 4. Input parameters for the cooling chain description

Variable	Notation in the report	Dimension
Lower frequency	$f_{\min}$	GHz
Upper frequency	$f_{\max}$	GHz
Electronic Gain	G_A	Dimensionless or in dB: $G_A[dB] = 20\log G_{A,lin}$
Loop length	l_{loop}	cm
Pickup and kicker parameters		
Number of loop pairs	n_p, n_k	-

For the pickup and kicker the program calculates approximate length of the electrodes in accordance with the same estimation as for transverse planes: $l_{electrode} = n_{p,k}\left(l_{loop} + 1cm\right)$.

For the cooling rate and consumption power calculation the program uses the input parameters which are the same as for both transverse planes, which are listed in the Table 2.

Output parameters for longitudinal degree of freedom are listed in the Table 5.

Table 5. Output parameters

Variable	Notation in the report	Dimension
Thermal noise power	P_{th}	W
Schottky power	P_S	W
Total power	P_{tot}	W
Equilibrium momentum spread	$\frac{\Delta p}{p}_{eq}$	-
Cooling rate	$1/\tau_{cool}$	s^{-1}

2. Gated Stochastic cooling

The stochastic cooling can be interpreted as the composition of the coherent single particle effect, when each particle is cooled by means of signal generated itself and the incoherent noise signal generated by all its neighbour particles. The basic formula shows the relation between these two processes and determines the cooling rate of transverse emittance $1/\tau_{tr} = (1/\varepsilon_{tr})d\varepsilon_{tr}/dt$ or longitudinal momentum spread $1/\tau_l = (1/\Delta p)d\Delta p/dt$:

$$\frac{1}{\tau_{tr,l}} = \frac{W}{N}\left[\underbrace{2g(1-1/M_{pk}^2)}_{\substack{coherent\\ effect(cooling)}} - \underbrace{g^2(M_{kp}+U)}_{\substack{incoherent\\ effect(heating)}}\right], \tag{1}$$

where N is the number of particles in the ring, $W = f_{\max} - f_{\min}$ is the cooling system bandwidth, g is the gain parameter, U is the noise to signal ratio, M_{pk} is the mixing factors from pick-up to kicker, and M_{kp} – from kicker to pick-up. Mixing facto is

$$M_{pk,kp} = \frac{1}{2W\left|\eta_{pk,kp}\right|T_{pk,kp}\dfrac{\Delta p}{p}}, \tag{2}$$

where $\eta_{pk,kp}$ – slip factor from pick-up to kicker and from kicker to pick-up, $T_{pk,kp}$ – time flight from pick-up to kicker and from kicker to pick-up, $\Delta p/p$ – momentum spread.

The gated stochastic cooling has four identical **Sections** (Fig.1) with the same input parameters: **Type** (horizontal, vertical or longitudinal), **Bandwidth [GHz]**, **Initial position** and **Final position** of the cooling section in the unit of circumference (the same as position of Barrier Buckets).Check enabled option **More parameters** for simulation with Formula (1). **Time (pick-kick)** – flight time from pick-up to kicker, flight time from kicker to pick-up is calculated via revolution time. **Slip factor** – slip factors between pick-up and kickers. Noise supposed to be equal 0. User can define **Gain** manually or use **Optimal gain** which will be calculated with formula $g_{opt} = \dfrac{1-1/M_{pk}^2}{M_{kp}+U}$.

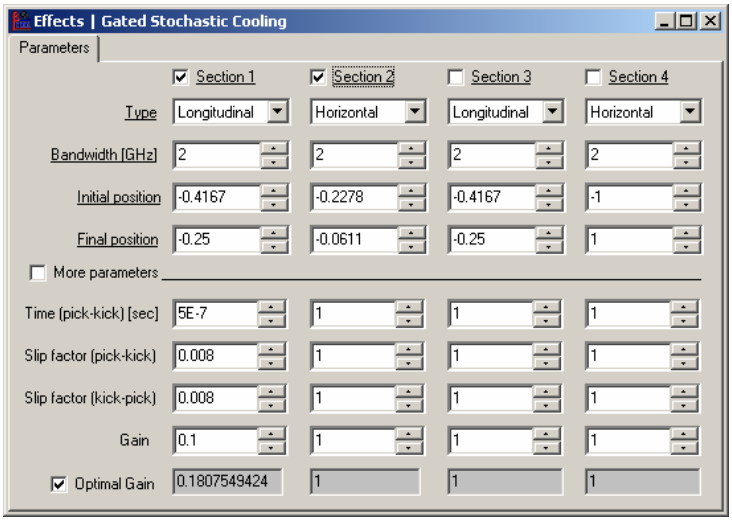


Fig.1. Parameters of gated stochastic cooling.

For optimal gain and assuming that T_{kp} equal to the revolution time T_{rev}, η_{kp} equal to the ring sleep factor η_{ring}, neglect term $1/M_{pk}^2$ one can use simple approximation of cooling rates:

$$\frac{1}{\tau_{tr,l}} = \frac{W}{N} \cdot \frac{1}{M_{kp}} = \frac{2W^2 \eta_{ring} T_{rev} \dfrac{\Delta p}{p}}{N}. \qquad (3)$$

To use this approximation switch off option **More parameters** (Fig.1) Position of each active cooling section is indicated by the black horizontal line with square points on the Barrier Bucket Form (Fg.2).

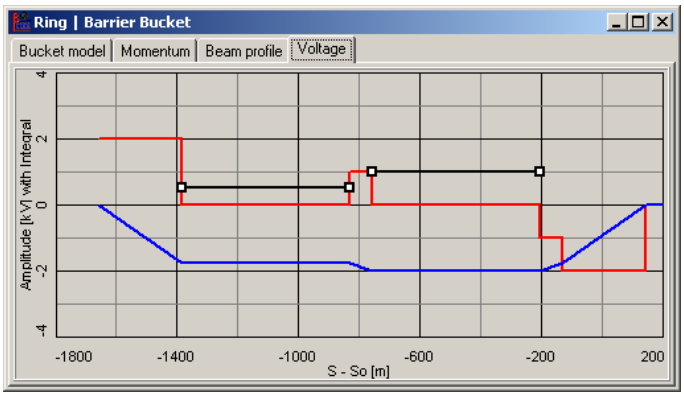
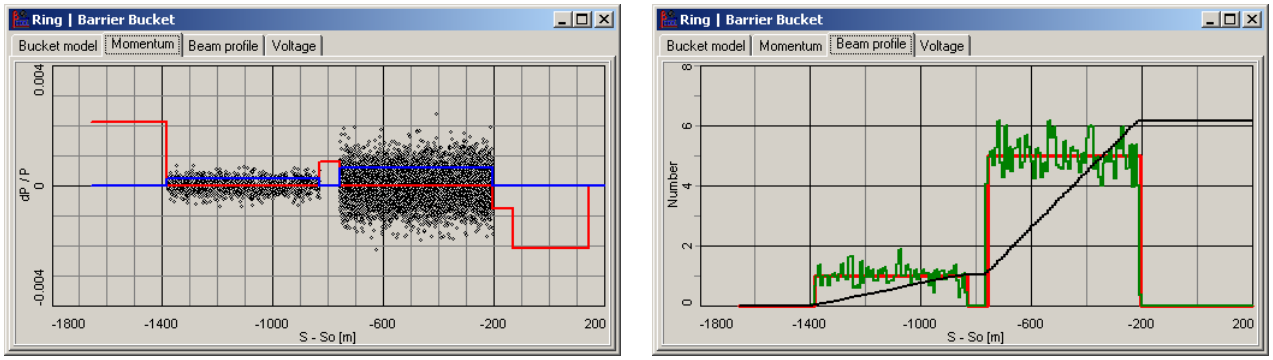


Fig.2. Barrier Bucket voltage distribution.

For simulation of gated cooling rates for each model particle the average distribution of the momentum spread (Fig.3, blue line) and average density of particles (Fig.3b, red line) is used.



Fgi.3. (a) Longitudinal distribution of model particles and barrier bucket in momentum space, (b) longitudinal beam profile along longitudinal coordinate.