**Hellweg2D Interface Guide**

To calculate beam dynamics in TW linacs, you need to follow some simple steps, describing below.

***1. INPUT file.***

The program reads the initial data from text file «INPUT.txt» which must be in the same folder as the “Hellweg2D.exe” file. If the data was successfully read, you would see the correct numbers in the main window. If the data is not correct, you can look through the «PARSER.txt» file to see what lines were successfully read. The example of an input file is presented in Figure 1.

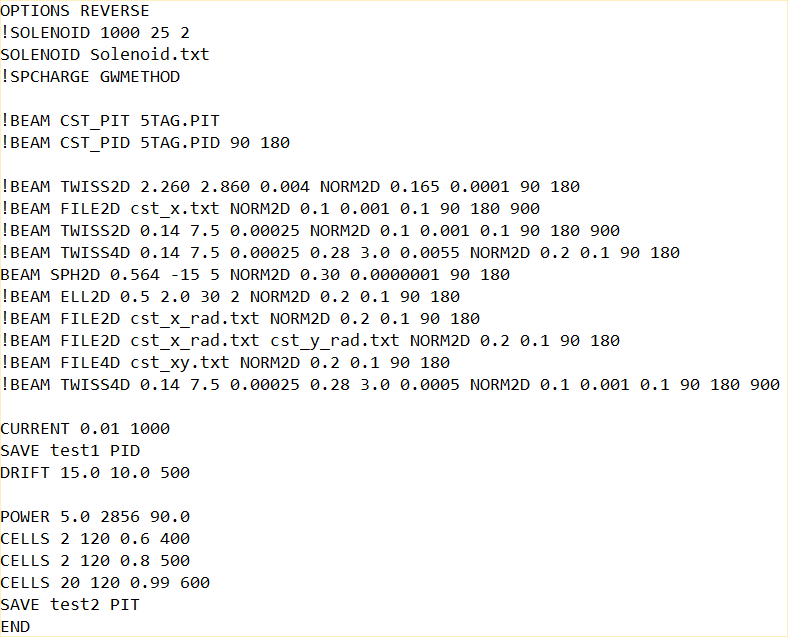


Figure . INPUT.txt structure

Keywords description (“bold” parameters are optional)::

* OPTIONS <***Keyword>***

This keyword defines extra modes that will be used in calculations. Possible options:

* *REVERSE*

Backward travelling wave regime would be considered.

*Ex. OPTIONS REVERSE*

* SOLENOID *B[Gs] L[cm]* ***Z0[cm]*** or SOLENOID *Filename*

This keyword defines the source of the external uniform longitudinal magnetic field. There are three parameters to define solenoid specification: magnetic field strength B[Gs]; solenoid length L[cm]; longitudinal coordinate of the start of the solenoid Z0[cm]

*Ex. SOLENOID 1500 20 010*

Custom magnetic field distribution can be imported from a file. In this case, you add the name of the file with magnetic field data after the keyword SOLENOID.

*Ex. SOLENOID BFIELD.txt*

The following input file format must be used: the first column is z coordinate in [cm], the second is a magnetic field in [Gs]. Z-coordinates can be outside the simulation domain.

Figure . BFIELD.txt file structure and imported field

* BEAM <***Set of Parameters>***

This line defines the parameters of the initial distribution or import of particles in transversal and longitudinal phase space. Currently, there is a possibility to input the whole particles phase space from one of the standard format files, or define/import the transversal and longitudinal distribution separately. In the latter case, the keyword for the longitudinal distribution should immediately follow the transverse distribution definition (see examples section).

Here is the list of the possible keywords

* *CST\_PID FileName* ***φ0 [deg] Δφ [deg] σφ [deg]***

The code will import the initial distribution form CST PID file with “FileName” name. The normal phase distribution (see NORM2D) can be optionally defined by two parameters: mean phase (φ0), and phase length (±Δφ). Optionally, the RMS deviation (σφ) can be defined. If no phase distribution parameters are defined, the code will generate the uniform distribution from 0 to 360 deg. If the deviation is not defined, the code will assume a uniform distribution. The format of “pid” file is the following:

x [m], y [m], z [m], (βγ)x [], (βγ)y [], (βγ)z [], m0 [kg], q0 [C], I [A]

*Ex. BEAM CST\_PID cst\_example.pid*

*Ex. BEAM CST\_PID cst\_example.pid 90 180*

*Ex. BEAM CST\_PID cst\_example.pid 90 180 50*

* *CST\_PIT FileName* ***COMPRESS***

The code will import the initial distribution from CST PIT file with “FileName” name. If the imported beam consists of multiple bunches, the keyword COMPRESS can be added to compress all particles into one bunch. The particle with time t will have the phase φ=-t\*c/λ. The format of this file is the following:

x [m], y [m], z [m], (βγ)x [], (βγ)y [], (βγ)z [], m0 [kg], q0 [C], Q [C], t [s]

*Ex. BEAM CST\_PIT cst\_example.pit*

*Ex. BEAM CST\_PIT cst\_example.pit COMPRESS*

* *FILE2D FileName1* ***FileName2***

The code will import 2D distribution from the file with *FileName1* name. In the case of longitudinal distribution, the file should have the following format: *φ [deg], W[MeV].* In case of transversal distribution, the format must be: *r [cm], r’ [rad]*. Optionally, the second file can be defined for transversal distribution only. In this case, the code will read *x [cm] x’ [rad]* distribution from the first file, and *y[cm] y’[rad]* from the second file, and converts the distribution to cylindrical coordinates r, θ, pr, and pθ. The user must ensure that the number of particles from different files matches, as well as the order of the particles in the file.

*Ex. BEAM FILE2D radius.txt FILE2D energy.txt*

*Ex. BEAM FILE2D particles\_x.txt particles\_y.txt FILE2D energy.txt*

* *FILE1D FileName φ0 [deg] Δφ [deg]* ***σφ [deg]***

The code will import the energy distribution from the file with *FileName* name. The file should have the following column: *W[MeV].* The normal phase distribution (see NORM2D) should be defined by two parameters: mean phase (φ0), and phase length (±Δφ). Optionally, the RMS deviation (σφ) can be defined. If the deviation is not defined, the code will assume a uniform distribution. In the case of multiple files are defined, the user must ensure that the number of particles from different files matches, as well as the order of the particles in the file. This type of distribution can only be used for longitudinal phase space

*Ex. BEAM FILE2D radius.txt FILE1D energy.txt 90 180*

*Ex. BEAM FILE2D radius.txt FILE1D energy.txt 90 180 20*

* *FILE4D FileName*

The code will import the 4D transversal phase space distribution from the file with *FileName* name. The file should have the following format: *x [cm] x’ [rad] y[cm] y’[rad]*. In the case of multiple files are defined, the user must ensure that the number of particles from different files matches, as well as the order of the particles in the file. This type of distribution can only be used for transversal phase space

*Ex. BEAM FILE4D phase\_space.txt FILE1D energy.txt 90 180*

* *TWISS2D αx, βx [cm/rad], εx [cm\*rad]*

The code will generate the x-x’ distribution with αx, βx [cm/rad], εx [cm\*rad] Twiss parameters (Figure 3), and y-y’ distribution for the same parameters. x-x’ and y-y’ distributions are independent. Then the code converts the distribution to cylindrical coordinates r, θ, pr, and pθ. This type of distribution can only be used for transversal phase space.

C:\Users\sitch\AppData\Local\Microsoft\Windows\INetCacheContent.Word\Fig0b.tif

Figure . Phase space defined by Twiss parameters.

*Ex. BEAM TWISS2D 0.14 2.0 0.0005 FILE2D energy.txt*

* *TWISS4D αx, βx [cm/rad], εx [cm\*rad] αy, βy [cm/rad], εy [cm\*rad]*

The code will generate the x-x’ distribution with αx, βx [cm/rad], εx [cm\*rad] Twiss parameters (Figure 3), and y-y’ distribution with αy, βy [cm/rad], εy [cm\*rad]. x-x’ and y-y’ distributions are independent. Then the code converts the distribution to cylindrical coordinates r, θ, and pr. pθ is assumed equal to zero to avoid particle magnetization. This type of distribution can only be used for transversal phase space.

*Ex. BEAM TWISS4D 0.14 2.0 0.0005 0.28 4.0 0.0015 FILE2D energy.txt*

* *NORM2D W0 [MeV] ΔW [MeV]* ***σW [MeV]*** *φ0 [deg] Δφ [deg]* ***σφ [deg]***

The code will independently generate the Gaussian energy and phase distribution with mean energy (W0), energy spread (±ΔW), mean phase (φ0), and phase length (±Δφ). Optionally, the rms deviations (σW, σφ) can be defined for both energy and phase distribution. It is impossible to define the RMS deviation for only one parameter! If deviations are not defined, the code will assume uniform distributions. If ΔW (or Δφ) is zero or negative, the code will generate the Gaussian distribution with no boundaries. If ΔW2<12·σW2, the distribution will be uniform! Otherwise, the code generates Truncated Gaussian distribution with the limits W0- ΔW and W0+ΔW. The same for phase distribution. See Figure 4. This type if distribution can only be used for longitudinal phase space

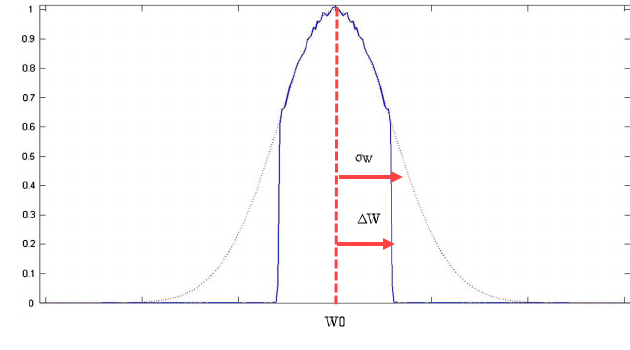


Figure . Example of Truncated Gaussian Distribution

*Ex. BEAM TWISS2D 0.14 2.0 0.0005 NORM2D 1.0 0.5 0.25 90 180 20*

*Ex. BEAM FILE2D beam.txt NORM2D 1.0 0.5 90 180*

* *SPH2D Rcath [cm],* ***Rsph [cm], kT [eV]***

The code will generate radial distribution from the spherical cathode (see Figure 5). *Rcath [cm]* defines the radial limit of the particles. Rayleigh distributions with σR=Rcathwill be modeled. *Rsph [cm]* is an optional parameter for cathode sphericity. If Rsph=0 or not defined, the cathode will be cylindrical (flat). If Rsph >0, the cathode is concave. If Rsph <0, the cathode is convex. The normal (to the cathode surface) component of r’ is defined as r’=-sin(r/Rsph). Optional parameter kT [eV] defines the thermal emittance. This type of distribution can only be used for transversal phase space.

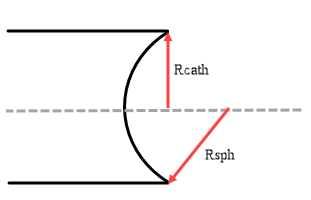


Figure . Example of spherical cathode distribution

*Ex. BEAM SPH2D 0.564 1.0 10.0 NORM2D 1.0 0.5 90 180*

*Ex. BEAM SPH2D 0.564 1.0 FILE2D energy.txt*

*Ex. BEAM SPH2D 0.564 FILE1D energy.txt 90 180*

* *ELL2D ax [cm], by [cm],* φ ***[deg], h***

The code will generate a radial elliptical distribution with *ax [cm] and by [cm]* half-axes. Optionally, the beam can be rotated in x-y space by an angle of φ[deg]. If phi is not defined, it is assumed zero. See Figure 6. Optional parameter h defines the RMS deviation of the Gaussian distribution as *σx= ax/h, σy= ay/h*. If not defined, h is assumed to equal to 1. No particles will be generated outside of the ellipse. Radial and azimuthal speeds assumed to be zero. This type of distribution can only be used for transversal phase space.

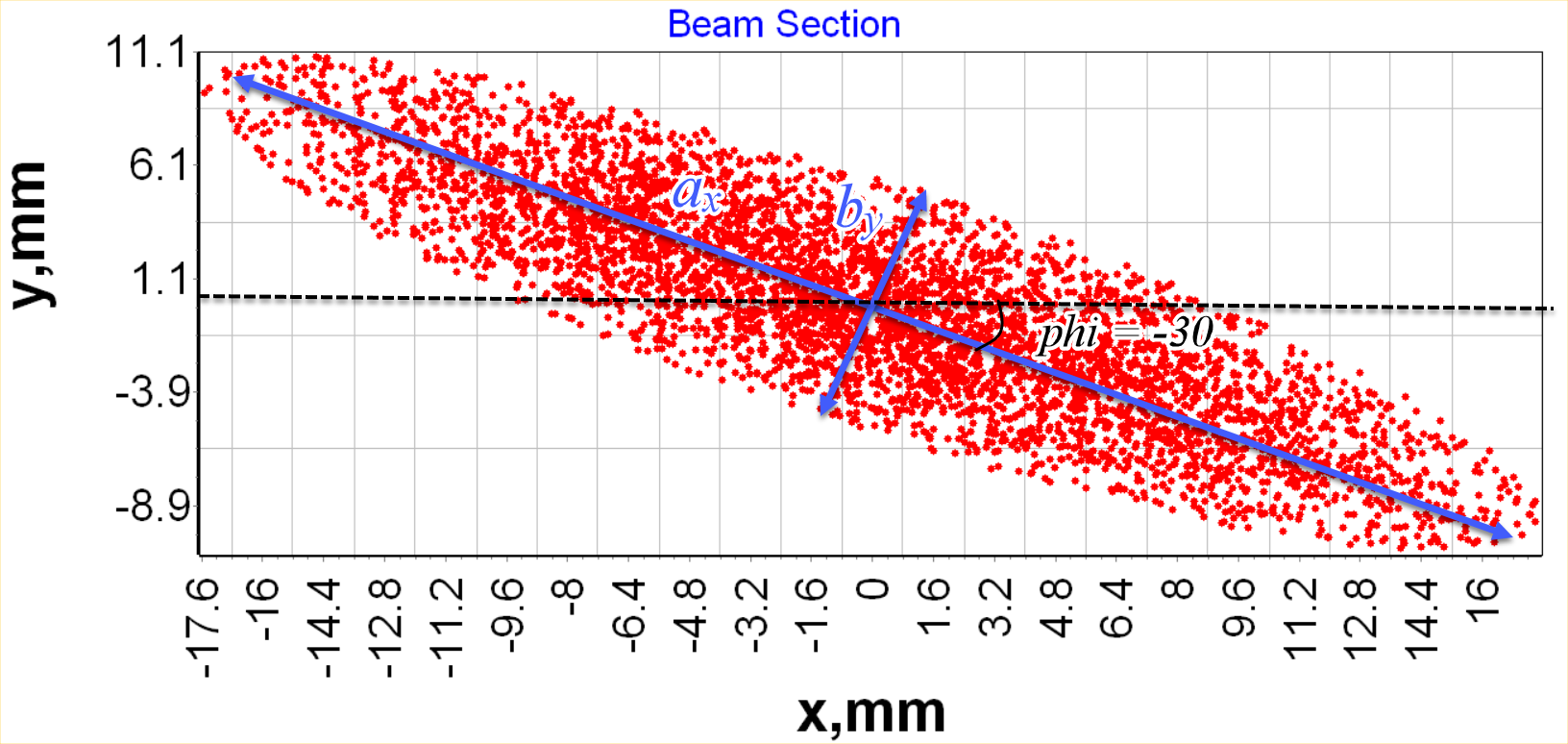


Figure . Example of Elliptical distribution

*Ex. BEAM ELL2D 2.0 0.5 -30 2 NORM2D 1.0 0.5 90 180*

*Ex. BEAM ELL2D 2.0 0.5 -30 FILE2D energy.txt*

*Ex. BEAM ELL2D 2.0 0.5 FILE1D energy.txt 90 180*

* CURRENT *I0[A]* **Np**

This line defines the beam current *I0[A]* during the RF pulse (*I0*=*Q/Tpulse*), and a number of particles in beam *Np*. If the beam is generated randomly, the Np parameters must be defined. If the particles distribution is imported from the file, the Np parameter is optional. In this case, if Np is defined and less than number of imported particles, the code will consider only the first Np of the imported particles. If Np is greater than the number of imported particles, it will be ignored.

*Ex. CURRENT 0.15 1000*

* SPCHARGE ***Keyword******Nslices TRAIN Lbunch [cm]***

This line defines if the space charge algorithm should be included in simulations. Two algorithms are available: ellipsoid bunch approximation per Lapostolle formula (COULOMB keyword), ellipsoid approximation with 3 elliptic integrals form-factors and the fields outside the bunch (ELLIPTIC) and Garnett-Wangler algorithm (GWMETHOD keyword). If the line is absent, the code will not include space charge in simulation. If no parameters are defined after the SPCHARGE keyword, the code will assume Lapostolle algorithm. For Lapostolle and Elliptic algorithms it is optionally possible to define the dimensions of the ellipsoid core in rms values. If no slices is defined, the code will define each dimension for ellipsoid as 3 rms. For Garnett-Wangler algorithm it is optionally possible to define the number of bunch slices with *Nslices* parameters. If no slices is defined, the code will assume 1 slice.

WARNING: As for now, Garnett-Wangler algorithm was disabled from simulations!

It is possible to define a keyword TRAIN, optionally followed by the parameter *Lbunch* [cm]. In this case, the code will assume that the simulated bunch is surrounded by two bunches, one is trabelling at Lbunch distance ahead of the simulated bunch, the other is travelling at Lbunch distance behind the simulated bunch. The code will then adjust the space charge forces according to this 3-bunches model. If Lbunch parameter is not defined, the code will assume the distance between bunches equal to the wavelength at the given position. Currently, only Elliptical space charge model supports this feature.

WARNING: If several bunches are simulated, the Lbunch should be multiplied by the number of bunches!

*Ex. SPCHARGE*

*Ex. SPCHARGE COULOMB*

*Ex. SPCHARGE ELLIPTIC 1.5*

*Ex. SPCHARGE GWMETHOD 5*

*Ex. SPCHARGE ELLIPTIC 1.5 TRAIN 10.5*

* POWER P0[MW] F0[MHz] **Δφ[deg]**

Defines RF power parameters for the accelerating section: pulsed input power P0[MW]; operating frequency F0[MHz]; phase shift from the reference phase Δφ[deg] (this parameter can be undefined – in this case it will be automatically assumed to be zero)

*Ex. POWER 2.0 2856*

*Ex. POWER 4.5 5712 90*

It possible to define several RF sections by using the keyword POWER before coupling CELL line (Figure 7). In this case all the power from the previous section will be assumed transferred to the load (Figure 8). If no power sources are defined, the system wavelength would be assumed to 1 meter!

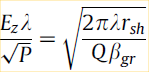


Figure 7. Example of extra RF section



Figure 8. Longitudinal electrical field in linac (left – without extra POWER line; right- with extra POWER line)

* CELL θ [deg] βph [] Eλ/P1/2 [Ω1/2] **αλ3/2 [m1/2] a/λ []**
* CELLS N θ [deg] βph [] Eλ/P1/2 [Ω1/2] **αλ3/2 [m1/2] a/λ []**

The keyword CELL defines an accelerating cell by the following parameters: phase advance, relative phase velocity, accelerating field invariant , normalized attenuation and normalized aperture. If last 2 parameters are not defined, program automatically recalculates it for DLS structure using tables. Currently, automatical calculation is only available for modes π/2 and 2π/3. To define multiple identical cells, the keyword CELLS followed by the number of cells should be used instead.

*Ex. CELL 120 0.999 380.0 0.01 0.12*

*Ex. CELL 90 0.8 200.0*

*Ex. CELLS 6 120 0.999 380.0 0.01 0.12*

* DRIFT L[cm] a[cm] **Nm**

This keyword defines a drift tube using 2 parameters: length L[cm]; radius a[cm]. Optionally, it is possible to add the number of mesh points for drift element after the radius that will override the global mesh settings.

*Ex. DRIFT 10.0 2.0*

*Ex. DRIFT 10.0 2.0 100*

Please, note that the DRIFT element will terminate any power used before.

* SAVE ***FileName*** ***Parameters***

If this line is present, the code will export the live particle parameters (phase, energy, radius, azimuth and radial velocity) at position define in the INPUT to the file with the defined name and .dat extension

*Ex. CELLS 3 120 0.999 380.0*

*SAVE test*

*DRIFT 10.0 2.0*

In this example, the particle parameters will be exported at the position between 3 cells and drift to the file *test.dat*. Multiple export commands are possible, but two SAVE lines at the same position will be overwritten.

It is possible to define the number of particles to be exported or the region of particles numbers after the name of file

*Ex. SAVE test 500*

*Ex. SAVE test 1000 2000*

In the first example, the first 500 particles will be exported. In the second one, only the particles with numbers from 1000 to 2000 will be exported. Lost particles are not exported.

Several flags are allowed to define the particular parameters to be exported. If no flags are defined, all parameters (except live status) will be exported. If at least one flag is set, only flagged parameters will be exported.

LOST – export the lost or live status of the particle

PHASE – export the phase of the particle

ENERGY – export the energy of the particle

RADIUS – export the radius of the particle

AZIMUTH – export the azimuth of the particle

DIVERGENCE – export the radial divergence r’ of the particle

The flags must be defined in any combination after the number of elements region or after the file name if the region is not defined.

*Special Formats.* If a one of the following keywords is present, the code will ignore all other keywords, and save the beam in a special format.

PID –export the beam in CST PID format (see Beam keyword section) to the file with \*.pid extension

PIT –export the beam in CST PIT format (see Beam keyword section) to the file with \*.pit extension

ASTRA – export the beam in ASTRA format to the file with \*.astra extension

x [m], y [m], z [m], px [eV/c], py [eV/c], pz [eV/c], clock [ns], Q0 [nC], index [], status []

To export the beam in the multiple format, it is necessary to define a line with SAVE keyword for each format.

*Ex. SAVE test LOST ENERGY*

*Ex. SAVE test 500 ENERGY PHASE RADIUS*

*Ex. SAVE test ENERGY PHASE*

*Ex. SAVE cst\_export PID*

*Ex. SAVE astra\_export ASTRA*

*Ex. SAVE test 500 2000 LOST RADIUS VX*

* !COMMENTS

Any line with improper format will be ignored and not copied into the output file. To make a comment use the '!' symbol at the beginning of the line

*Ex. !This line is a comment*

***2. Simulation and post-processing results.***

Program interface is presented in Figure 9

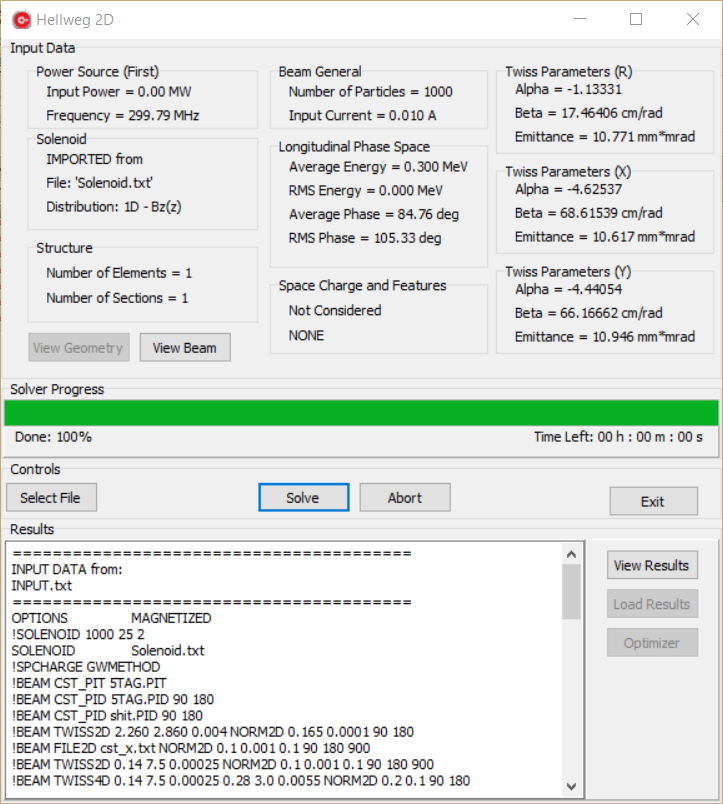


Figure 9. Hellweg2D Interface

Hellweg2D automatically reads an input file (INPUT.TXT) from the folder with the executable file and displays user defined parameters. It is possible to select the other file. The input cells parameters ocan be visualized by pressing “View Geometry” button and the input beam parameters by pushing ”View Beam” button. After solving the problem (button “Solve”) «OUTPUT.txt» file is automatically generated. The results can be viewed either in the Hellweg2D window or in OUTPUT.TXT file in program folder.

Button “View Result” lets user to view graphical results such as Field vs. coordinate; phase space; energy spectrum, etc (Figure 10). The user can choose any longitudinal position of the beam to view the particles distribution. The radio buttons in “Coordinate” section allow to choose between cylindrical and Cartesian coordinates. There are three sliders that allow changing the number of bins for the spectrum visualization, reduce the number of output points to improve the speed, and the define the number of particles in the core for the envelope.

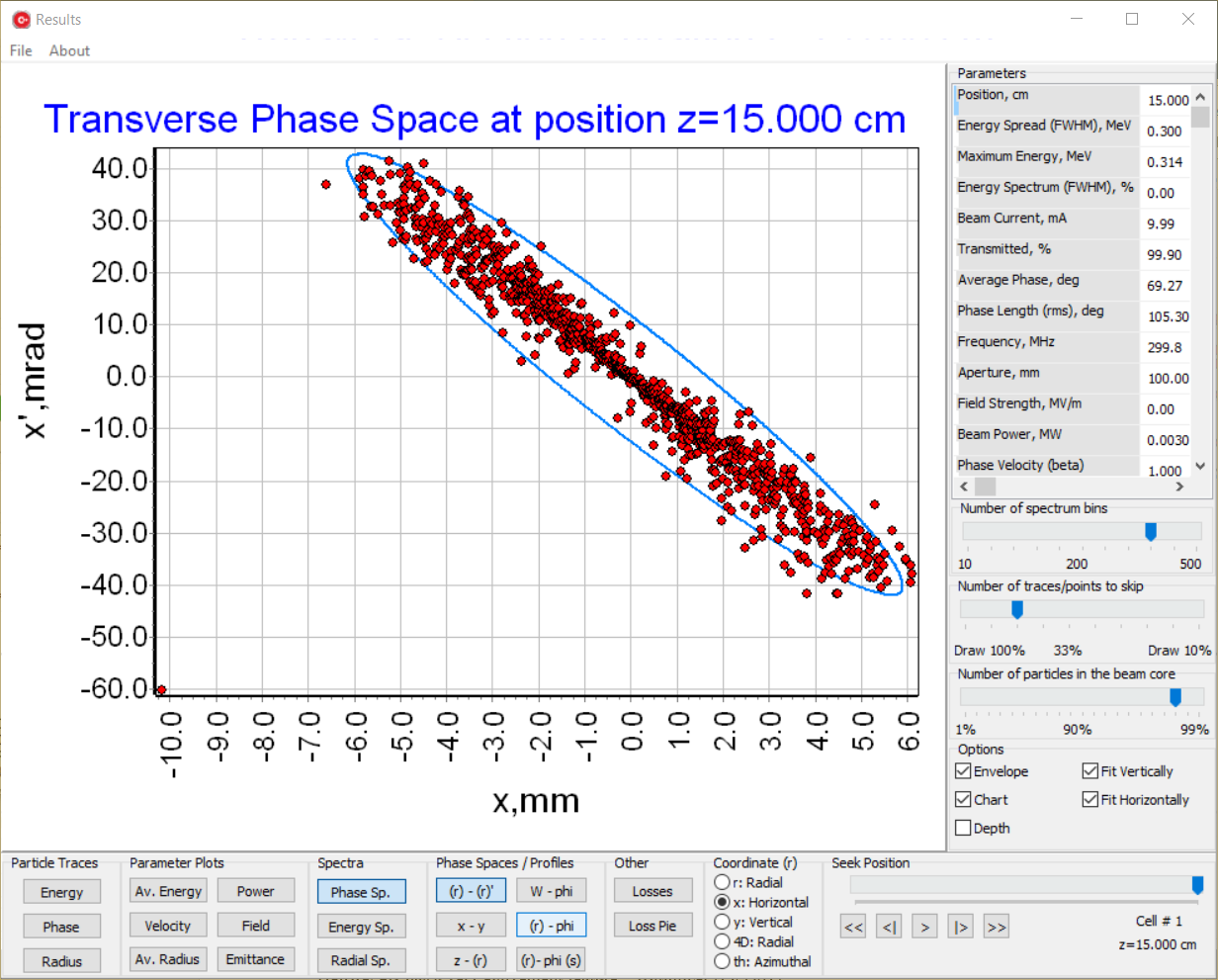


Figure 10. Results visualization interface

***3. Optimizer.***

Hellweg2D has a very convenient feature – Optimizer (Fig. 3.1)



Fig. 3.1 Optimizer interface

This function can calculate DLS cells parameters (relative phase speed βphase; normalized value of electrical accelerating field intensity ; normalized attenuation factor αλ3/2[m1/2]; normalized aperture radius a/λ) using included tables to obtain the desired energy on the end of the accelerator without access to the INPUT file.

- Optimize Buncher. In this window user defines coupler parameters and desired buncher field parameters.

*∙Input parameters:* Choice between π/2 and 2π/3 modes; operating frequency; input pulse power.

*∙Limits:* It necessary to define allowable variations of the values of the accelerating electrical field on the end of the buncher. User can choose between 3 field dimensions: dimensionless field A; normalized electrical field ; absolute value of electrical field E [MV/m].

∙*Object:* There are 2 algorithms to calculate buncher cells parameters: *Sweep* divides an interval defined in Limits for equal parts (Samples) and calculates field in them using table’s data; *Optimum* chooses the best field value to obtain the best capture coefficient/energy spectrum/phase spectrum (Parameter).

- Optimize Accelerator. In this window user defines desired output parameters.

*∙Structure type:* The choice between Constant Gradient or Constant Impedance structures. In the first case the optimizer will add the cells identical to the last one. In second case, it will adjust the aperture radius in order to maintain the same electrical field gradient.

*∙Target Energy:* It is necessary to define desired output energy. Marker “Exact Value” varies beam current to obtain exact value of average beam energy.

Marker “Start from cell” is used when it is necessary to modulate accelerating structure after the already defined cells parameters in INPUT file (Optimize Buncher marker is switched off)

- Get Results. You can find results in the OUTPUT file in the Results folder

***4. Configuration file.***

Mesh parameters and interpolation parameters are defined in file Hellweg.ini which placed in program folder (Fig. 4.1)

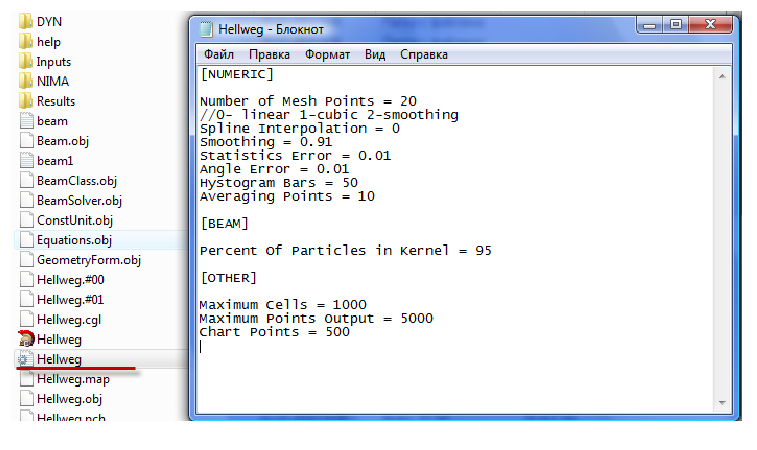


Fig. 4.1 Hellweg2D configuration file

- Number of mesh points. This value of mesh points will be used for accelerating field interpolation in cell. Please, note that the same number of points will be used for a single DRIFT element.

- Interpolation. Relative phase speed and value of accelerating electrical field are defined in the centers of cells. Then these values are interpolated for other mesh cells (Number of mesh points). In Figure 4.2 are presented 2 kinds of interpolations (linear and cubic). It is recommended to use linear interpolation to avoid nonphysical field oscillations. Smoothing spline can be used for example to obtain spectrum envelope.



a b

Fig. 4.2 Interpolation types (a-linear; b-cubic)

**Thank you for using our program!**

**Have a good simulation time with Hellweg2D!**