Hellweg (RsLinac) User Guide

Prepared by *Sergey V. Kutsaev*
Latest update: **October 9, 2022**

Hellweg (RsLinac) code is used to calculate quasi-self-consistent beam dynamics in TW linacs.

1. INPUT file and Keywords

The program reads the initial data from text file «INPUT.txt» (or as defined in Hellweg.ini configuration file), which must be in the same folder as the "Hellweg.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.

```
OPTIONS REVERSE
!SOLENOID 1000 25 2
SOLENOID Solenoid.txt
!SPCHARGE GWMETHOD

!BEAM CST_PIT 5TAG.PIT
!BEAM CST_PID 5TAG.PID 90 180

!BEAM TWISS2D 2.260 2.860 0.004 NORM2D 0.165 0.0001 90 180
!BEAM FILE2D cst_x.txt NORM2D 0.1 0.001 0.1 90 180 900
!BEAM TWISS2D 0.14 7.5 0.00025 NORM2D 0.1 0.001 0.1 90 180 900
!BEAM TWISS4D 0.14 7.5 0.00025 0.28 3.0 0.0055 NORM2D 0.2 0.1 90 180
BEAM SPH2D 0.564 -15 5 NORM2D 0.30 0.0000001 90 180
!BEAM ELL2D 0.5 2.0 30 2 NORM2D 0.2 0.1 90 180
!BEAM FILE2D cst_x_rad.txt NORM2D 0.2 0.1 90 180
!BEAM FILE2D cst_x_rad.txt cst_y_rad.txt NORM2D 0.2 0.1 90 180
!BEAM FILE4D cst_xy.txt NORM2D 0.2 0.1 90 180
!BEAM TWISS4D 0.14 7.5 0.00025 0.28 3.0 0.0005 NORM2D 0.1 0.001 0.1 90 180 900

CURRENT 0.01 1000
SAVE test1 PID
DRIFT 15.0 10.0 500

POWER 5.0 2856 90.0
CELLS 2 120 0.6 400
CELLS 2 120 0.8 500
CELLS 20 120 0.99 600
SAVE test2 PIT
END
```

Figure 1. 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] Z_0[cm] Lf [cm]**

- or SOLENOID Filename **Z_0[cm]**

This keyword defines the source of the external uniform longitudinal magnetic field. There is one mandatory parameter B[Gs] that defines the longitudinal magnetic field strength Bz and three optional parameters to define solenoid specification:; solenoid length L[cm]; longitudinal coordinate of the solenoid start Z_0[cm], and the length of the fringe field region Lf[cm].

The default value for Lf is 1 cm. When defining fringe fields, please, make sure that this region is not too short (at least several mesh points), so the numerical results of dBz/dz are accurate. Z0 defines the start of a flat field region. Fringe fields region starts from Z0-Lf. The default value for Z0 is zero. If L is not defined, the magnetic field will be Bz=B, Br=0, Bth=0 in the whole simulation region.

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. If the optional parameter Z0 is defined, the imported map will be shifted by Z0 along z-axis.

Ex. SOLENOID BFIELD.txt

The one of the following input file formats must be used:

For 1D distribution the first column is z coordinate in [cm], the second is a magnetic field in [Gs]. Z-coordinates can be outside the simulation domain. The code will automatically calculated Br as $Br=-r/2 * dBz/dz$. If the imported mesh is more than 10 times coarse than the simulation mesh, the derivatives will be calculated in the simulation domain mesh.

For 2D distribution the first column is r coordinate in [cm], the second is z coordinate in [cm], columns 3-5 define magnetic field components in [Gs]. If four columns are defined, the code will interpret the last two columns as Br [Gs], Bz [Gs]. If five columns are defined, the code will interpret the last three columns as Bx [Gs], By [Gs] and Bz[Gs].

For 3D distribution the first column is x coordinate in [cm], the second is y coordinate in [cm], the third is z coordinate in [cm], columns 4-6 define Cartesian magnetic field components in [Gs]. Bx [Gs], By [Gs] and Bz[Gs].

The code will check the first row with numerical values and will plan on which distribution to consider, depending on the number of columns in this row. For 2D and 3D imported distributions the field outside the defined domain is considered zero. It is the responsibility of the user to provide the accurate distribution to avoid simulation errors related to improper Br – Bz balance (for example, when Br is not proportional to dBz/dr)

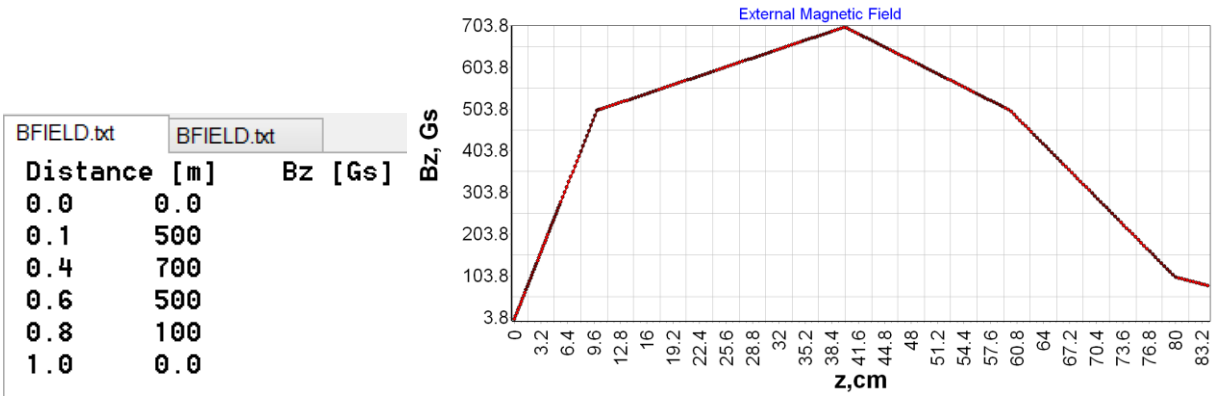


Figure 2. BFIELD.txt file structure and imported field

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] σ_φ [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 (σ_φ) 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], $(\beta\gamma)_x$ [], $(\beta\gamma)_y$ [], $(\beta\gamma)_z$ [], m_0 [kg], q_0 [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 $\varphi=-t*c/\lambda$. The format of this file is the following:

x [m], y [m], z [m], $(\beta\gamma)_x$ [], $(\beta\gamma)_y$ [], $(\beta\gamma)_z$ [], m_0 [kg], q_0 [C], Q [C], t [s]

Ex. BEAM CST_PIT cst_example.pit

Ex. BEAM CST_PIT cst_example.pit COMPRESS

WARNING: For the following types, please, define the energy in MeV/u values in case of ions.

- PARMELA_T2 FileName **COMPRESS**

This option is similar to PIT but uses Parmela T2 (text – OUTPUT 1 only). The format is the following:

x [cm], x' [mrad], y [cm], y' [mrad], phi [deg], W [MeV], particle # []

Please, make sure that only one set of the output beam dump is used, not the whole file.

- 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, p_r, and p_θ. 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] **σ_φ [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 (σ_φ) 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 α_x, β_x [cm/rad], ε_x [cm*rad]

The code will generate the x-x' distribution with α_x, β_x [cm/rad], ε_x [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, p_r, and p_θ. This type of distribution can only be used for transversal phase space.

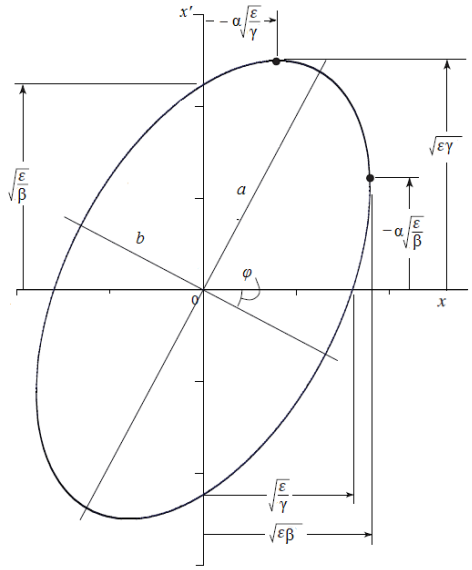


Figure 3. Phase space defined by Twiss parameters.

Ex. BEAM TWISS2D 0.14 2.0 0.0005 FILE2D energy.txt

- TWISS4D α_x, β_x [cm/rad], ε_x [cm*rad] α_y, β_y [cm/rad], ε_y [cm*rad]

The code will generate the x-x' distribution with α_x, β_x [cm/rad], ε_x [cm*rad] Twiss parameters (Figure 3), and y-y' distribution with α_y, β_y [cm/rad], ε_y [cm*rad]. x-x' and y-y' distributions are independent. Then the code converts the distribution to cylindrical coordinates r, θ, p_r. and p_θ. 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] **σ_W [MeV]** φ0 [deg] Δφ [deg] **σ_φ [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 (σ_W, σ_φ)

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 $\Delta W^2 < 12 \cdot \sigma_W^2$, 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

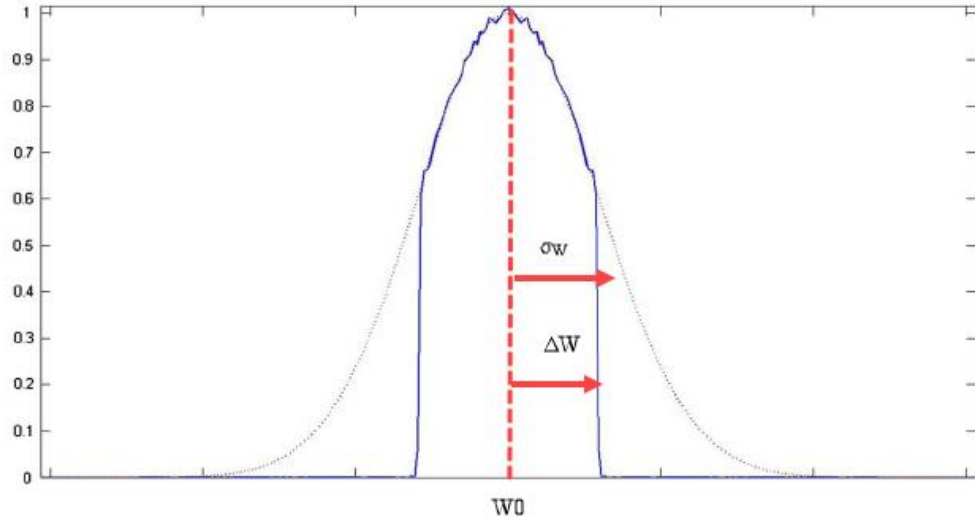


Figure 4. Example of Truncated Gaussian Distribution

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 σ_R= Rcath will 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.

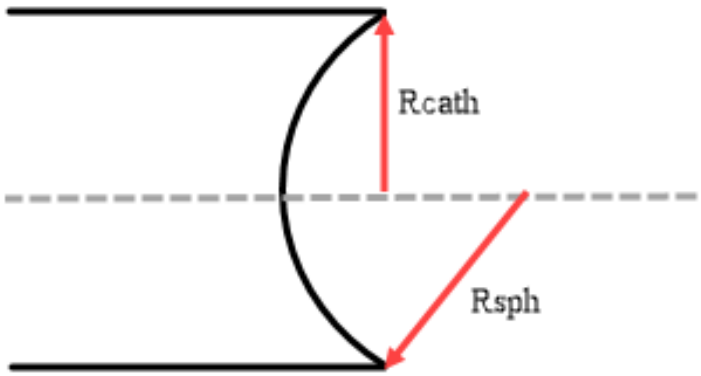


Figure 5. Example of spherical cathode distribution

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 a_x [cm], b_y [cm], φ **[deg], h**

The code will generate a radial elliptical distribution with a_x *[cm] and* b_y *[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 $\sigma_x = a_x/h$, $\sigma_y = a_y/h$. 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.

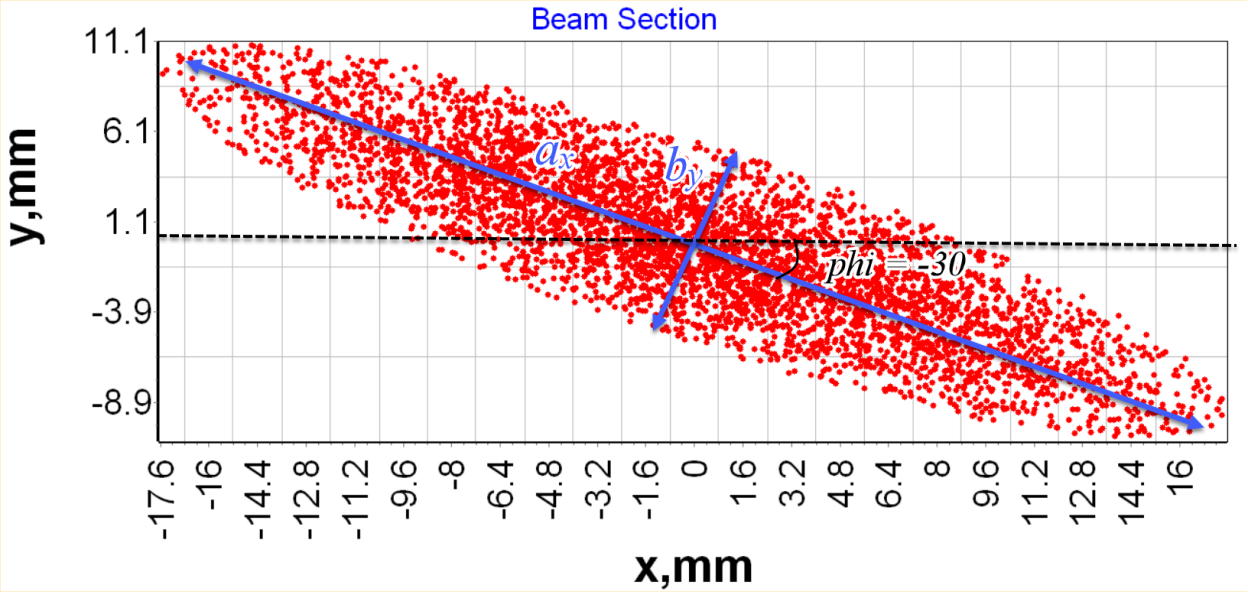


Figure 6. Example of Elliptical distribution

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

PARTICLES Keyword **A [a.m.u] Q [ee]**

This optional line defines the particles species. Default particles are electrons. Acceptable keywords are ELECTRONS (to simulate electrons, W0=511 keV, charge = 1), PROTONS (to simulate protons with W0=938 MeV, charge =1) and IONS (to simulate ions with atomic mass A a.m.u, and charge Q). In the latter case, the rest energy will be considered equal to 1 nucleon mass 931.5 MeV. Parameters A and Q are optional and have default values of 1. Parameter A can be floating value, while Q must integer. Both parameters must be positive. However, the code will treat negative charge as positive.

WARNING: It is the responsibility of the User to check that parameters A and Q are physical (i.e. A≥Q, Q≤Z etc.)!

WARNING: If IONS with A>1 is defined, the code will consider that the energy is defined in MeV/u (per nucleon)! For Q>1 the current is defined in electrical values (i.e. charge multipled by particle current)!

WARNING: Hellweg code was designed to simulate electrons in disk-loaded-waveguide-like structures. While ion simulation capability was added to the code, the User must understand the applicability limits of ion beam dynamics simulation. I.e. simulating beams with energies below 10 MeV/u in structures other than CCL-types (like RFQs) might yield incorrect results.

Ex. PARTICLES IONS 238 50

Ex. PARTICLES PROTONS

CURRENT $I_0[A]$ **Np**

This line defines ***electrical*** beam current $I_0[A]$ during the RF pulse ($I_0=Q/Tpulse$), 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

STRUCT <FileName> **Phase [deg]**

If this line is present the code will use the data from file to calculate $\alpha\lambda^{3/2}$ and a/λ for the cells with the defined phase advance if these parameters are missing in CELL line definition (see "CELL" description). If the phase advance is not defined, the parameters from the file will be used for all cells. It is possible to define multiple STRUCT lines for different phase advances. The latest STRUCT definition will have a priority. For example, if two sections with different structures are used the definition can be as follows:

STRUCT DLS120.dat 120

CELLS 0.999 120 500

STRUCT BTW120.dat 120

CELLS 0.999 120 550

WARNING: if you define the phase in the first occurrence of STRUCT line, and no phase in the later occurrence, all cells with <Phase> phase advance will still be calculated per first definition, since the line with the specified phase has a priority over default parameters.

The file with the cell parameters data should contain the tables with $E\lambda/P^{1/2}$ and $\alpha\lambda^{3/2}$ for different phase velocities and a/λ as following:

βph / a/λ	**0.08**	**0.09**	**0.10**	**0.12**
0.5	500	400	350	200
0.7	600	500	400	250
0.9	650	550	450	300
1.0	700	600	500	350

The data must be arranged in the following format:

Line 1: the values of phase velocities, separated by space. Ex. [0.5 0.7 0.9 0.1]

Line 2: the values of a/λ, separated by space Ex. [0.08 0.09 0.10 0.12]

Array of $E\lambda/P^{1/2}$ values, consisting of)#values from Line 1) rows and (#values from Line 2) columns. For example:

500 400 350 200

600 500 400 250

650 550 450 300

700 600 500 350

This must be followed by the same sized array with the values $\alpha\lambda^{3/2}$.

STRUCT line is optional. If missing and CELL parameters are under-defined, the code can use tabulated data for pi/2 and 2pi/3 DLS structures as described in the corresponding sections.

Ex. STRUCT DLS120.dat 120

Ex. STRUCT DLS

POWER P_0[MW] F_0[MHz] **Δφ[deg]**

Defines RF power parameters for the accelerating section: pulsed input power P_0[MW]; operating frequency F_0[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!

```
OPTIONS          COULOMB
SOLENOID 0.09    0.5      0
BEAM 45 45       EQ       0.025    0.0000   EQ
CURRENT 0.075    1000     1.00     1.192    0.002

POWER    5.0     2856     0.00
CELL 120         0.420    82.500   0.004802         0.190819
CELL 120         0.475    92.000   0.003981         0.188993
CELL 120         0.525    110.000  0.003364         0.182546
                                  ...
CELL 120         0.999    514.875  0.007304         0.103506


POWER    5.0     2856     0.00
CELL 120         0.999    368.000  0.003984         0.120257
CELL 120         0.999    370.500  0.004052         0.119765
CELL 120         0.999    372.688  0.004138         0.119335
CELL 120         0.999    375.188  0.004237         0.118842
CELL 120         0.999    377.375  0.004323         0.118412
CELL 120         0.999    379.875  0.004421         0.117920
CELL 120         0.999    382.375  0.004477         0.117642
```

Figure 7. Example of extra RF section

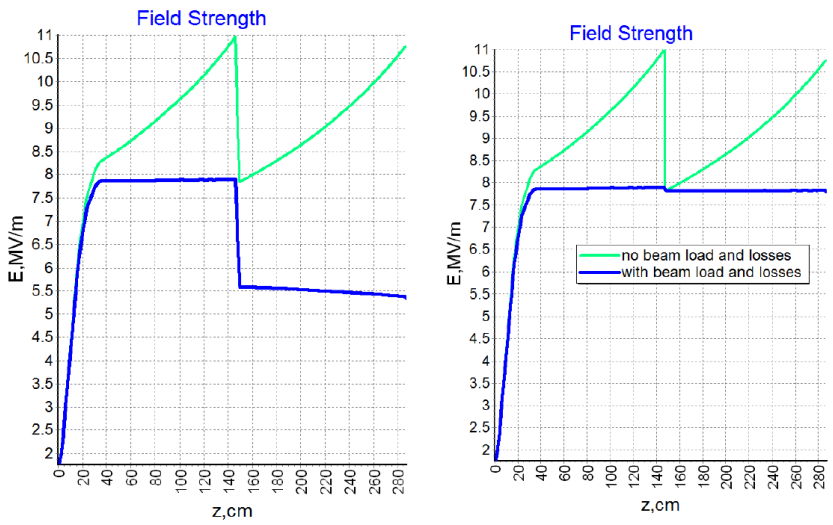


Figure 8. Longitudinal electrical field in linac (left – without extra POWER line; right- with extra POWER line)

CELL θ [deg] β_{ph} [] Eλ/$P^{1/2}$ [$\Omega^{1/2}$] **αλ$^{3/2}$ [m$^{1/2}$] a/λ []**

- CELLS N θ [deg] β_{ph} [] Eλ/$P^{1/2}$ [$\Omega^{1/2}$] **αλ$^{3/2}$ [m$^{1/2}$] a/λ []**

The keyword CELL defines an accelerating cell by the following parameters: phase advance, relative phase velocity, accelerating field invariant $\frac{E_z \lambda}{\sqrt{P}} = \sqrt{\frac{2\pi\lambda r_{sh}}{Q\beta_{gr}}}$, 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.

QUAD FileName L[cm] **kB[] Nm[]**

This keyword defines a magnetic quadrupole of the length L[cm] with the 2D field map imported from file with *FileName*. The field map should be defined in the format x[cm] y[cm] Bx[Gs] By[Gs]. The code will consider this 2D map to be uniform along the defined length. Unlike SOLENOID, where the field is overlaid over the elements, the QUAD element is inserted into accelerator lattice. In other word, the code treats the QUAD element as a drift tube with the magnetic field inside. The particles outside the imported mesh are considered lost. Optionally, it is possible to scale the field by specifying a coefficient kB (can be negative). If defined, the code will multiply all field values by kB. Also, similar to the DRIFT element it is possible to define the number of mesh points for drift element after the radius that will override the global mesh settings.

Ex. QUAD quad.txt 10.0

Ex. QUAD quad.txt 10.0 -2.0

Ex. QUAD quad.txt 10.0 -2.0 100

Please, note that the QUAD 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], p_x [eV/c], p_y [eV/c], p_z [eV/c], clock [ns], Q_0 [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 User Interface Main window is presented in Figure 9.

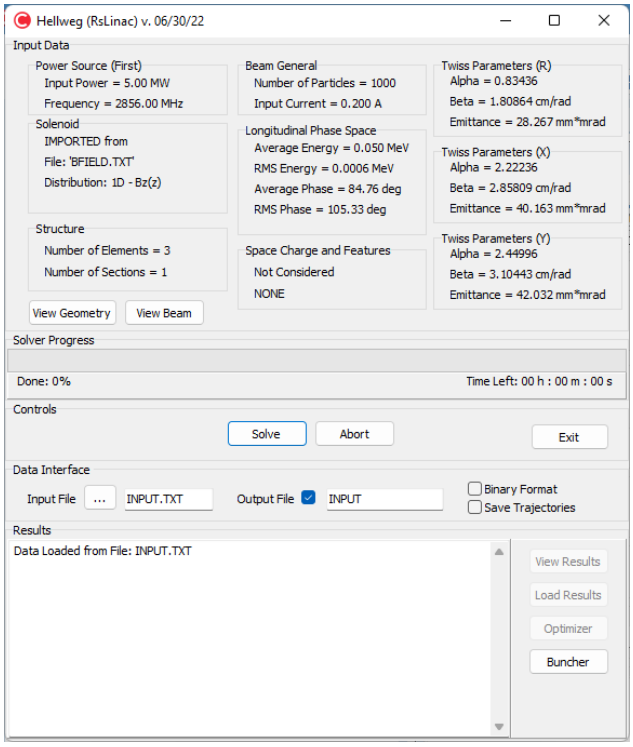


Figure 9. Hellweg GUI: Main Window

Hellweg automatically reads an input file (INPUT.TXT, or as defined in Hellweg.ini – see the Section below) from the folder with the executable file and displays user defined parameters. It is possible to manually select the other file by pressing "…" button next to the Input File lab3l. The input cells parameters can be visualized by pressing "View Geometry" button and the input beam parameters by pushing "View Beam" button. After solving the problem (button "Solve") the simulations macro results can be viewed either in the Main Window or in OUTPUT.LOG file in program folder. The name of the output file can be defined either in Hellweg.ini, or in the Main Window.

If GUI is disabled in Hellweg.ini, the program will automatically run the solver, save the data to files (if requested by user), and terminate the code. Otherwise, the user can use GUI to view the results after the solver run is finished. 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.

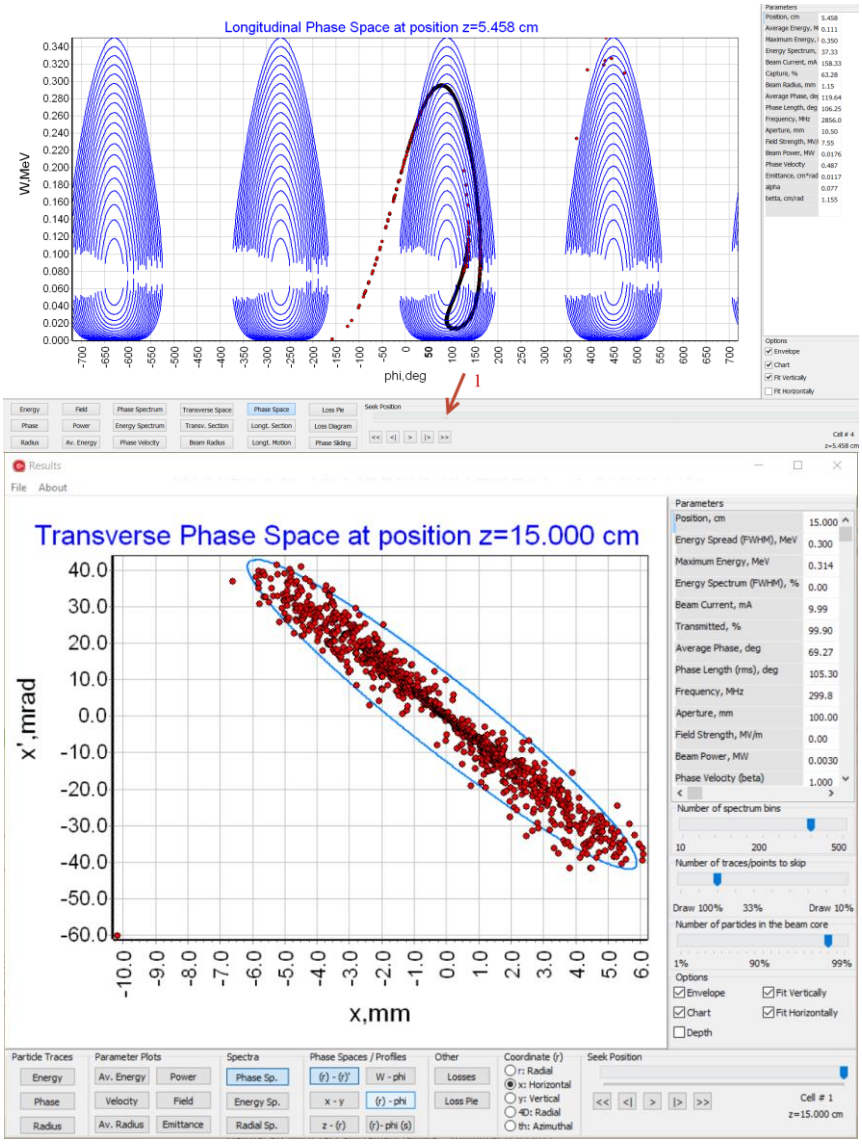


Figure 10. Results visualization interface

3. Optimizer

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

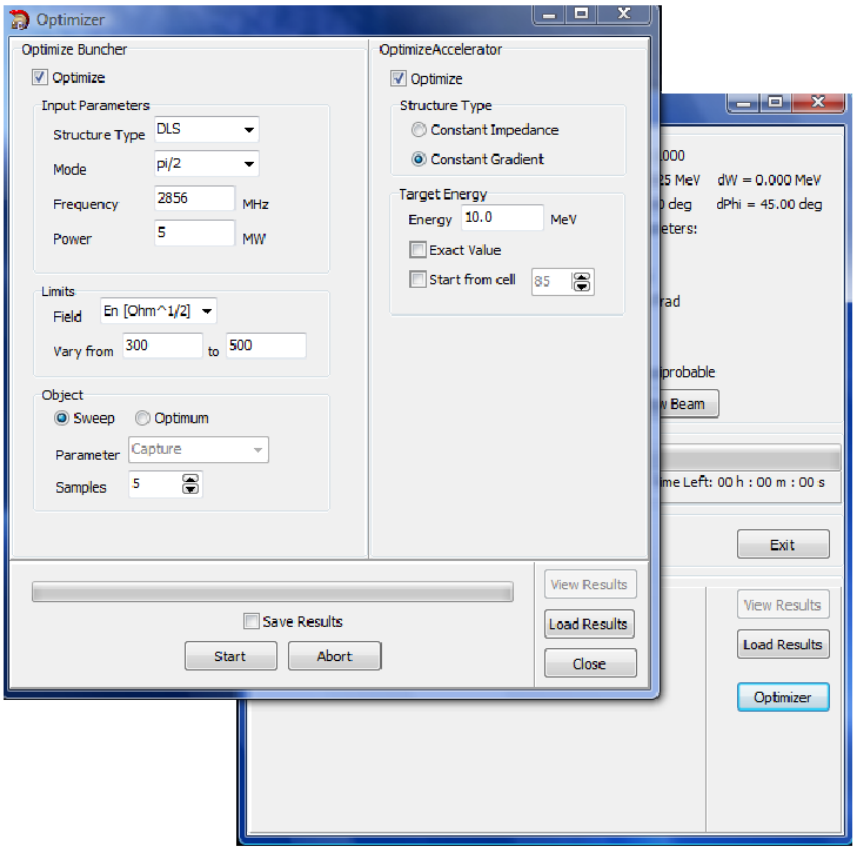


Fig. 3.1 Optimizer interface

This function can calculate DLS cells parameters (relative phase speed β_{phase}; normalized value of electrical accelerating field intensity ; normalized attenuation factor $\alpha\lambda_{3/2}[m_{1/2}]$; 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 INI file

This file allows user to control GUI and input parameters and well as mesh and interpolation parameters/ The file must be named Hellweg.ini and placed in program folder (Fig. 4.1). The file must consist of the following sections and can only contain the following parameters.

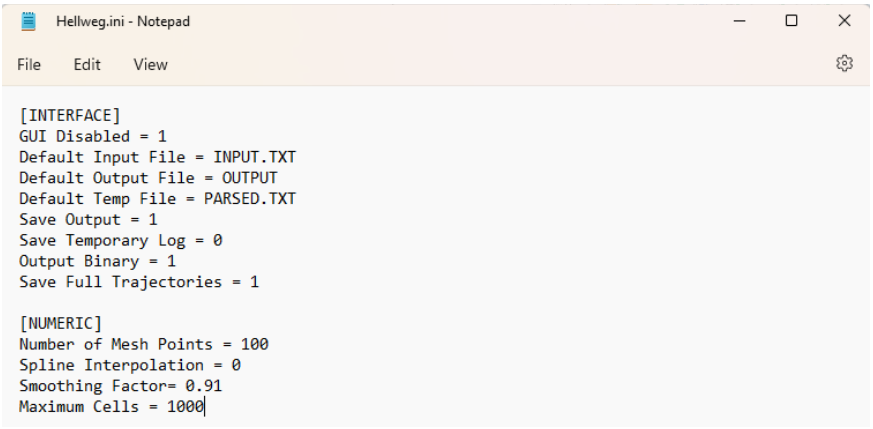


Fig. 4.1 Hellweg configuration file

[INTERFACE]

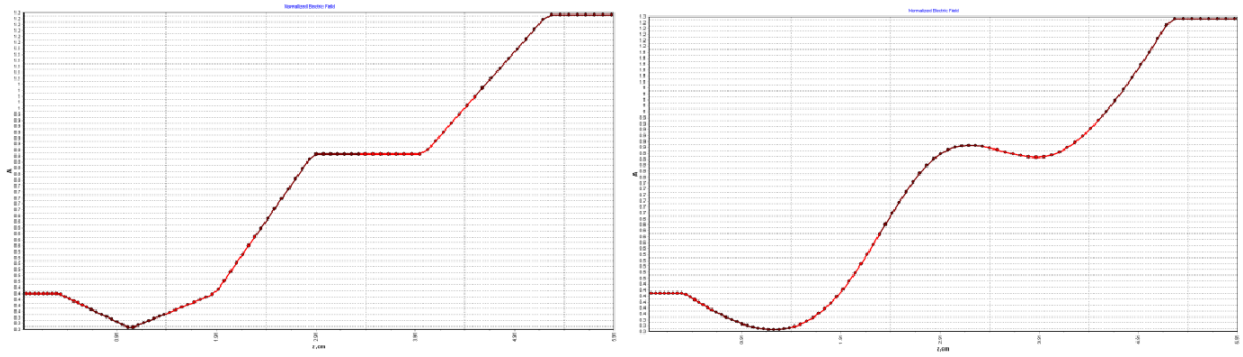
- **GUI Disabled**: If set to 0, the GUI interface will open, and the user will have control over the actions. If set to 1, the code will run automatically, save macro results in *.log file and then terminate. Default value = 0.
- **Default Input File**: The code will automatically use the defined file as input data. Default value = INPUT.TXT.
- **Save Temporary Log:** If set to 1, the code will automatically save the separate log file, even if the solver terminates incorrectly. This can be useful for debugging purposes. Default value = 0.
- **Default Temp File**: Define the temp file name (***) that will be used for temporary data storage. Default value = PARSED.TXT.
- **Save Output**: If set to 1, the code will automatically save the beam file after the solver is finished (***.out). At the moment the format is similar to Parmela T2: x [cm], x'[mrad], y [cm], y'[mrad], phi [deg], W [MeV]. The first row indicates the number of particles. Default value = 0.

- **Default Output File**: Define the output file name (***) that will be used for log, beam and trajectory files, as ***.log, ***.out and ***.traj, respectively. If NONE is defined, the name would be the same as input file. Default value = OUTPUT.
- **Output Binary**: If set to 1, the output file will be saved in binary format (except log file). In this case, the output file will have ***.bin extension. The first value has a size of <*integer*> and provides the number of particles. All other values have a size of <*double*> and correspond to sets of x [cm], x'[mrad], y [cm], y'[mrad], phi [deg], W [MeV] for each particle. Default value = 0.
- **Save Full Trajectories**: If set to 1, the code will save beam data for all mesh points as ***.traj. This option overrides Save Output option. The format of trajectories file is always binary and has the following structure. First two <*integer*> values are number of particles (Np) and number of mesh points (Nm), respectively. Then Nm values, size of <*double*>, represent z-coordinates of each mesh point. Then for each particle (total Np arrays) follows the array of the following Nm values at each mesh point (total Nm arrays): live status (not lost) x [cm], x'[mrad], y [cm], y'[mrad], phi [deg], W [MeV] – each <*double*>. Default value = 0.

[NUMERIC]

- **Number of mesh points**. Defines number of points per element (CELL, DRIFT etc.), where the differential equations will be solved. Default value = 20.
- **Spline Interpolation**. In Hellweg relative phase velocity and accelerating field amplitudes are defined ate the center of the cells. If the interpolation parameter is defined, these functions will be interpolated along z-axis: 0- linear, 1-cubic spline, 2-smoothing spline. Figure 4.2 presents thr examples of linear and cubic interpolation. It is recommended to use linear interpolation to avoid nonphysical field oscillations.

  NOTE: This feature was suppressed in versions 2017 and later and all values are now kept constant within the cells.
- **Smoothing Factor**. Obsolete parameter, used to define smoothing factor in smoothing spline interpolation. Such interpolation used both for parameter function interpolation and for the spectrum analysis. Currently is set to 0.8. Default value = 0.91.
- **Maximum Cells**. Defines the maximum number of cells that can be used in optimization module. Default value = 1000.



a b

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

Thank you for using our program!

Have a good simulation time with Hellweg!