# Upgrade of the code to 3D

## Azimuthal coordinates equations

Hellweg2D code was originally developed to solve the beam dynamics problems with 2D cylindrical asymmetry. While this approach is enough for the standard DLS-type accelerating structures with fundamental wave mode, it limited to the further essential upgrades that are required for solving the 3D problems or problems with broken 2D symmetry: solenoids, quadrupole optical elements, higher order modes (HOM), asymmetrical structures etc.

There are two ways to upgrade the code to 3D: Cartesian (x-y-z) or cylindrical (r-θ-z) coordinates. While the first may seem simpler and more convenient, we have decided to stick with the cylindrical coordinates for the following reasons:

* The accelerating structures for high power linacs are cylindrical, so the wave analytical equations are written in cylindrical coordinates and use cylindrical functions
* The particle in solenoids are moving in curvilinear coordinates, and cylindrical equations describe this motion more accurately

Since the original code was developed for the 2D cylindrical coordinates and based on the equations for *dvz/dz* and *dvr/dz*, we derived the equations for *dvθ/dz* to make the code three-dimensional. Also, to prepare the code for HOMs and asymmetrical fields we added the field components *Eθ*, *Hz*, *Hr* and external *Bθ* (see Yuri’s report for details):



These equations were implemented to the Hellweg code and made it full 3D. Several tests that will be described elsewhere were done to verify the beam dynamics such as electron motion in the solenoid, and emittance conservation of the beam with non-symmetrical particles distribution. These tests demonstrated that the conversion was done successfully.

## Initial particles distribution

The 3D conversion of the code required the major revision of the particles initialization. In the original code the beam could only be initialized in transverse with three Twiss parameters: α, β and ε. Moreover, the distribution was done for x-coordinate (Gaussian) rather than for radius (Rayleigh), which can be technically not correct for the realistic beam.

In this code upgrade, we introduced a couple of new options of the beam initialization that cover many cases of the standard and realistic 3D distribution:

* *Initialization with 2D Twiss parameters αx, βx [cm/rad], εx [cm\*rad]*

In this case, the code generates the x-x’ distribution with αx, βx [cm/rad], εx [cm\*rad] Twiss parameters (Figure 1), 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.

* *Initialization with 4D Twiss parameters 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 1), 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.

* *Gaussian distribution with parameters: 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 2. This type if distribution can only be used for longitudinal phase space

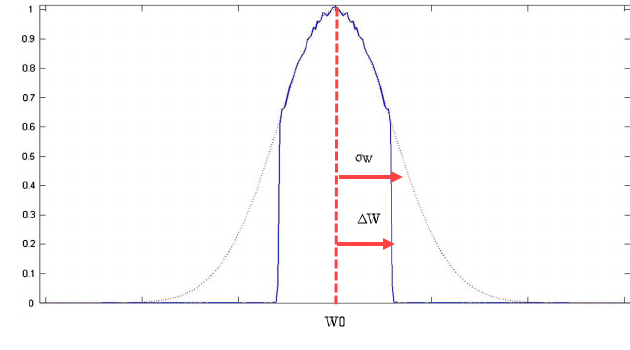


Figure . Example of Truncated Gaussian Distribution

* *Emission from spherical cathode: Rcath [cm],* ***Rsph [cm], kT [eV]***

The code will generate radial distribution from the spherical cathode (see Figure 3). *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.

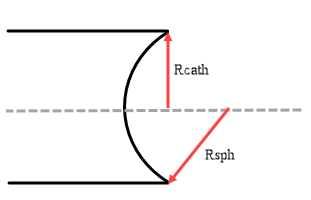


Figure . Example of spherical cathode distribution

* *Transverse elliptical distribution: 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 4. 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. The code developed for this type of distribution can be applicable in Phase II to model KV and waterbag distribution.

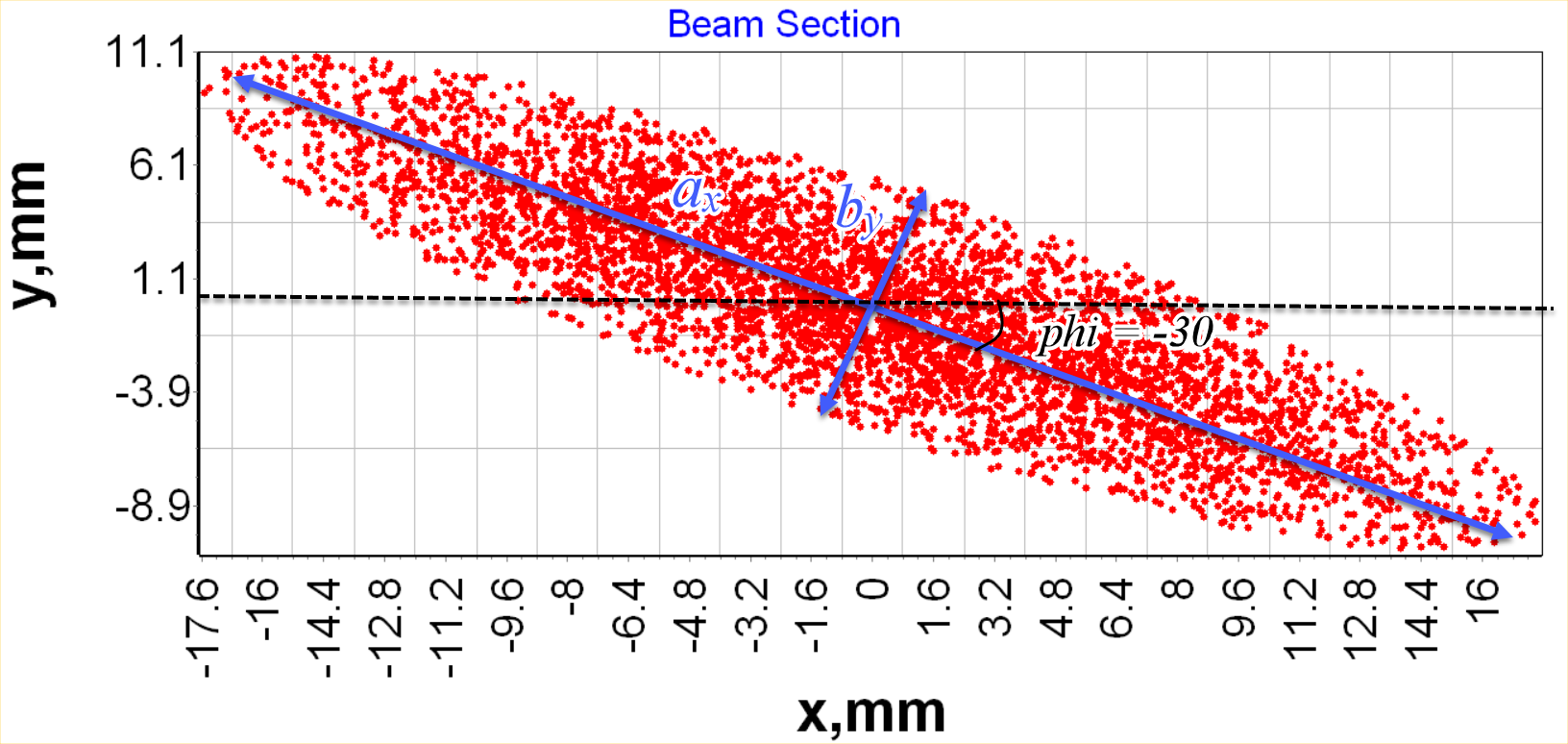


Figure . Example of Elliptical distribution

## Particles export and import

Generic beam initialization does not always provide the required particle distribution, especially when the electrons come from the external such as a gun or another accelerator. Also, the users often need to track a single particle with the defined parameters or a beam with some exotic distribution. For these cases, we have added the capabilities to import the particle parameters from the text files.

* *Import 1D distribution from file*

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

* *Import 2D distribution from file*

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.

* *Import 4D distribution from file*

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

* *Import distribution from CST Particle Interface in PID format*

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]

* *Import distribution from CST Particle Interface in PIT format*

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]

Similar to particle import, we have added the export capabilities, so that the user is able to process the data or continue simulations in the external code. If the SAVE command 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. It is possible to define the number of particles to be exported or the region of particles numbers after the name of file. 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
* VX – export the radial velocity of the particle
* PID – additionally export the file in CST PID format (see Beam keyword section) to the file with \*.pid extension
* PIT – additionally export the file in CST PIT format (see Beam keyword section) to the file with \*.pit extension

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.

## GUI update

We have added several visualizations features to reflect the above-described changes and facilitate the results interpretation for the user:

* The user can now choose between r-r’, r-θ’, x-x’, and y-y’ coordinates in transverse phase space view. The Twiss parameters will be calculated for the selected coordinate.
* It is possible to overlap and compare the particle distribution from two different files (Figure 5)

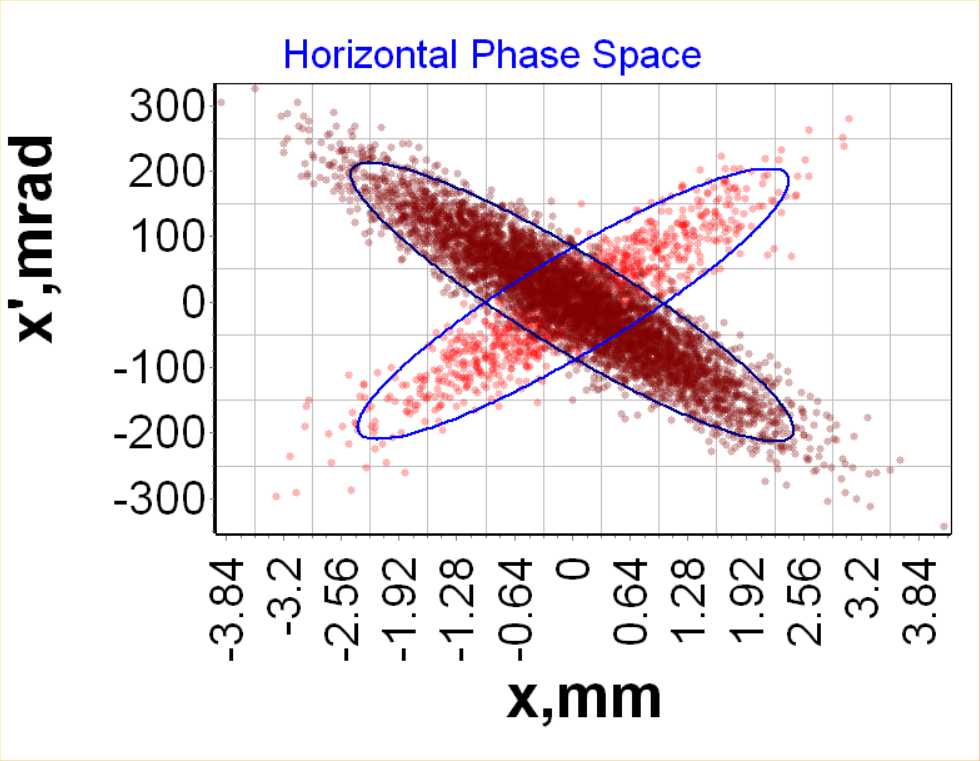


Figure . Particle distribution comparison

* Emittance variation along the linac is now possible to view
* The user can now change the number of bins for spectrum visualization. The radial spectrum was added
* The user can define the number of particles in the core for the beam envelope
* The depth was added to the plot to view the particle concentration (Figure 6). It is now possible to skip the particles from visualization to enhance the speed.

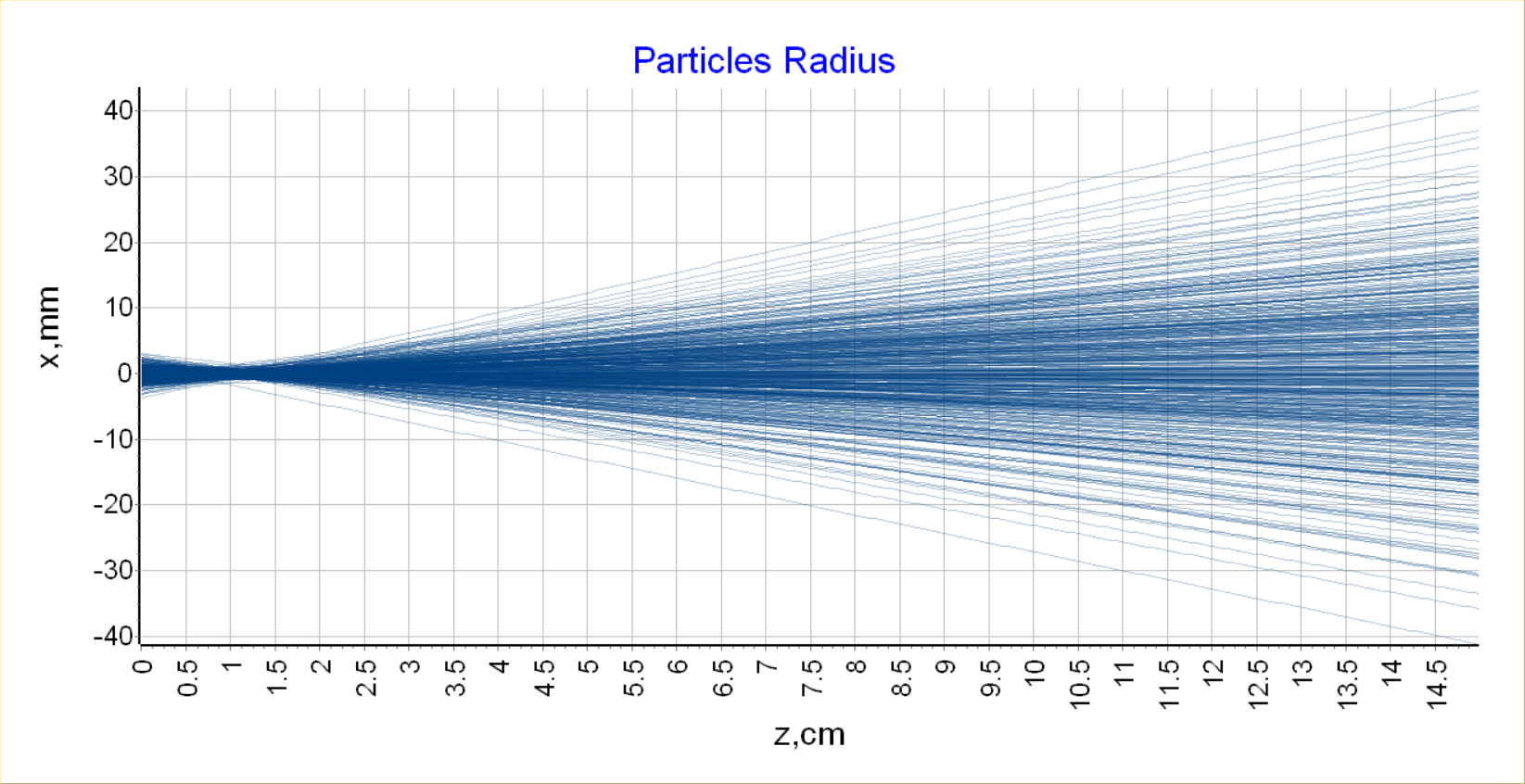


Figure . Particle visualization with depth