Interfacing between Python based 4D phase space reconstruction simulation and GPT

A phase ellipsoid is always defined by 6-D phase space. $\sum_{\text{Beam}}$, a 6×6 matrix describes the contour of it. For now, we're only interested to measure the transverse phase space. The most general form of beam matrix including all possible correlation between the x, x′ and y, y′ subspaces reads:

$$\sum_{4D} = \begin{pmatrix} \sigma_{xx'} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy'} \end{pmatrix}$$

[where $\sigma_{xx'} = \begin{pmatrix} \langle xx \rangle & \langle xx' \rangle \\ \langle x'x \rangle & \langle x'x' \rangle \end{pmatrix}$; $\sigma_{yy'} = \begin{pmatrix} \langle yy \rangle & \langle yy' \rangle \\ \langle y'y \rangle & \langle y'y' \rangle \end{pmatrix}$; $\sigma_{xy} = \begin{pmatrix} \langle xy \rangle & \langle xy' \rangle \\ \langle x'y \rangle & \langle x'y' \rangle \end{pmatrix}$; $\sigma_{yx} = \sigma_{xy}^{T}$]

Normalized R.M.S. Emittance along x or y: $\beta\gamma\sqrt{\langle \mu\mu \rangle \langle \mu'\mu' \rangle - \langle \mu\mu' \rangle^2}$ where μ refers to either x or y.

We need a method to reconstruct the full 4D transverse phase space from multiple 2D beam profiles.

The corresponding algorithm is as follows:

i) Measure 7 # of 2D histogram profile (x,y,Intensity) for different value of B_{fac} (in present case, 0.2 to 0.78) from the acsii sheet 'resultN_expt.txt'. In our case, the bin-width of these histograms has a limitation to 0.05 mm according to the resolution of whole screen-optical system point of view. You may start with 0.25 mm.

These are used as *input data* to reconstruct the 4D phase space.

ii) Use 4DUniformDist20K.DAT as the beam of *initial guess*.
[Here, each particle is assigned by an ID]

iii) Track the test beam from the point where the reconstruction will have to be made to the measurement point for all corresponding B_{fac} values via GPT. At the output, discretize the x-y space with a grid having same width as input data.

For the 1st iteration,

iv) Compare the measured profiles (Step (i)) with the profiles obtained from the simulations (Step (iii)) grid by grid. Note the ID of test beam that has hit the assumed grid. The weight factor of each of these particle (say i^{th}) for a particular B_{fac} setting (say B_{fac}^{1}):

$$w_{B_{fac}^{1}}^{i} = \frac{\text{Intensity of the Grid in Input Data}}{\text{Total \# of particles hitting the grid via GPT}}$$

In present case, each particle has 7 weights after this comparison. For a particle whose all weights are non-zero, sum up all of them to be the final weight (w_{tot}) of this particle. So, $w_{tot} = \sum_{N=1}^{7} w_{B_{fac}^{N}}^{i}$. Delete those particles having 0 weight from the initial beam distribution.

v) Finally set $\sum_N w^i$= Total # of particles in the input beam and scale the remaining initial beam distribution accordingly.

vi) Pick out the particle having non-zero weight (here i^{th} one) and generate new particles uniformly around it having width of ±0.25 mm and ±0.25 m rad in the initial test beam in such a way that total # of particle corresponds to scaled w^i.

vii) The discrepancy between the reconstructed profile and reference one is defined by RMSE = $\sqrt{\sum_{n=1}^{N}(\text{recon}_n - \text{ref}_n)^2 / N}$ for a particular B_{fac} .

[where, recon_n corresponds to intensity of reconstructed profile for n^{th} grid; ref_n corresponds to measured intensity for n^{th} grid; N is total # of grids assumed in 2D x-y space.]

[A plot of ‘RMSE - # of iteration’ for all B_{fac} (in our case 7) should be represented].

viii) Repeat the steps (iv)→(vii) with the newly generated test beam for a few iterations until the reconstructed results will converge.