Simulation questions - SixTrack

G. Robert-Demolaize, Y. Luo

OUTLINE

I - Tracking features in SixTrack – Beam-beam module

II - Modifications – Implementation

III - Preliminary runs - Optimization for CPU farm

I - Tracking features in SixTrack

- SixTrack: full 6D and chromatic treatment of particles over element-by-element tracking routines, using thin lens approximation.
- Internal limitation originally: 64 particles => need to increase drastically this number !!
- This is done by applying a DO loop over packs of 64 particles; upper limit is now set to 20000 particles (max = 357 packs).
- No apparent limitation to the number of turns, except memory issues.
- There is an option to save the coordinates of all particles at every element of the machine.
- To save CPU time, most of the output files are optional.

Beam-beam module

- From SixTrack manual:
  - define the beam-beam element (strong beam or wire):

name | *type* | *h-sep* | *v-sep* | *strength-ratio*

20 | **[mm]** | **[mm]** | **allows to split the kick**

  - define the type of interaction:

BEAM **norm. emittance [µm.rad]** **rms bunch length [m]**

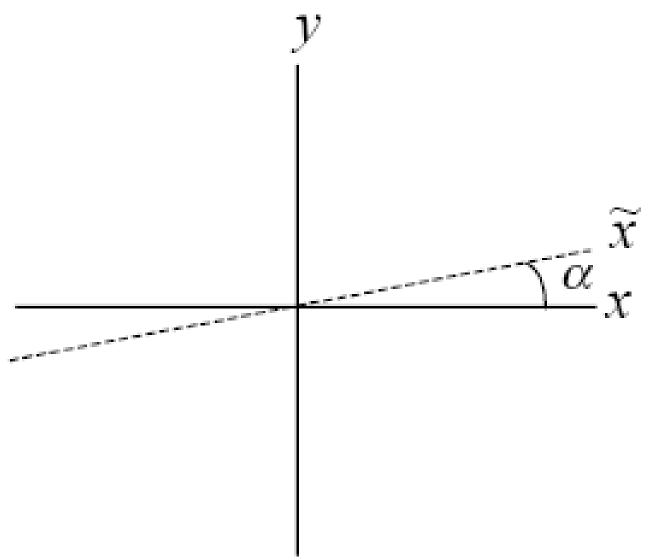
Npart | *partnum* | *emitnx emitny* | *sigz* | *sige* | *ibeco ibtyp lhc ibbc* | **optional switches**

rms energy spread

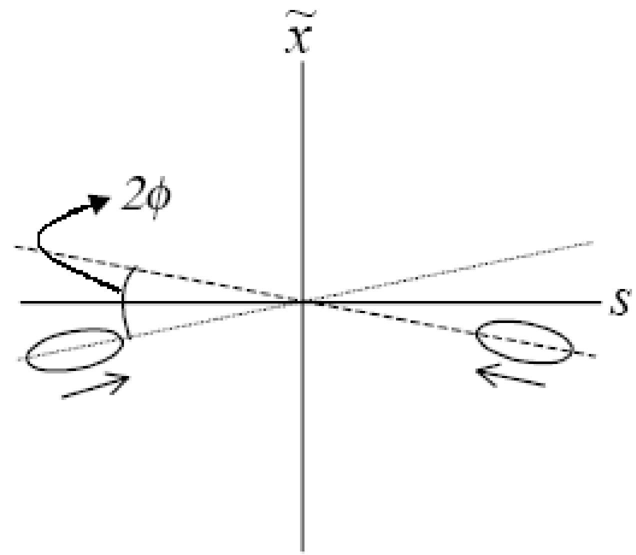
name | *ibsix* | *xang* | *xplane*

6D slices **half crossing angle [rad]** **crossing plane [rad]**

Crossing plane – crossing angle



crossing plane angle α = *xplane*
in the *(x-y)* plane



half crossing angle *φ* = *xang*
in the *(x̃-s)* plane

II - Modifications

- RHIC studies on beam-beam interaction should allow changing some of the main parameters, namely:

  - the number of particles, e.g. when modeling the wire compensator experiment to change the wire intensity,

  - the separation in each plane, if one decides to move the "other beam" towards the tracked particles,

  - the size of the "other beam", i.e. its normalized transverse emittances.

- While the population of the Strong Beam is already an existing parameter, the beam-beam distance and beam size can be modified on a turn by turn basis.

Implementation

- Lattice model is not an issue: 4D-6D, thin-thick beam-beam kicks are modified => changes are linear and "easy" to implement !!

- User has two options for the turn-by-turn modulations: random fluctuation or cosine function, e.g. for the Strong Beam population:
  - $N_{part} = N_{part} * K * \left(1 + A_N * \cos[\,2\pi\omega_N (n_{turn}-1) + \phi_N\,]\right)$
  - $N_{part} = N_{part} * K * \left(1 + A_N * 2 * [0.5 - Rand()]\right)$

  Rand() is a uniform distribution function within the interval [0;1]. The choice is made between "random" and "cosine" with the value of the frequency ω_N: if set to zero, "random" is applied.

New format of SixTrack input

```
BEAM-BEAM---------------
2.0D+11  2.5  2.5  0.25  0.0005  1 1 9 1
ip681  0 0 0  1.000  1E-3 1.000 0.005  1.000 0.000 0.000  1.000 0.000 0.000  1.000 1.000
ip682  0 0 0 -2.000  5E-3 0.000 0.000  1.000 0.000 0.000  1.000 0.000 0.000  1.000 1.000
NEXT
```

special switch value to turn ON new features (the boxed value 9)

- Red box = K, coefficient for Strong Beam population; sign gives the type of particles
- Blue box = $[A_N, \omega_N, \phi_N]$, modulation on Strong Beam population
- Green box = $[A_X, \omega_X, \phi_X]$, modulation on horizontal position
- Orange box = $[A_Y, \omega_Y, \phi_Y]$, modulation on vertical position
- Purple box = $[\eta_X, \eta_Y]$, coefficient on Strong Beam normalized emittances

III – Preliminary runs

- Test version only uses the newly implemented multi-particle feature, reading input files of up to 6400 particles. The goal of preliminary simulations is to perform emittance growth and beam lifetime benchmarking with real data.

- Tracking is done for the RHIC lattice (BB @ IP6 and IP8, eLens @ IP10), simulating 2 minutes in the machine ($\approx 10^7$ turns) so as to get meaningful statistics.

- Original plan: print out 6D coordinates of particles after every turn => this is unrealistic considering the amount of turns tracked and the ensuing CPU requirements !!

- Solution: calculate $\sum 2J_{x,y}$ and $\sum (x,y)^2$ every turn but print it out only every 10^5 turns; also check for lost particles every turn (aperture limitation at $N^*\sigma_{x,y}$, equivalent to collimation).

Optimization for CPU farm

- Some tests were with 1 CPU for 4 particles over 10^5 turns:
  - ✓ 1 step = 4 particles, 100k turns = ~50 seconds
  - ✓ 1 job = 4 particles, 1E7 turns = 5k seconds
  - ✓ 1 run = 64 particles, 1E7 turns = 80k seconds
  - ✓ 1 case = 6400 particles, 1E7 turns = 8M seconds = 92.5 days
- Current plans foresee roughly 100 cases (studying various parameters like phase advance, intensity of compensation, etc…) => need to parallelize jobs !!
- Since SixTrack is being used, worked with CERN to use BOINC (LHC@home) resources, but effort was unsuccessful; still can use regular CERN LSF queues, but these are shared with experiments and other CERN tracking studies…
- Recently started to adapt the code for IBM's BlueGene and ComPASS's NERSC farm systems; now awaiting user accounts.