Modeling of spin-orbital dynamics in a storage ring

April 10, 2018

Python code

- Integrates the equations of spin-orbital dynamics with scipy::odeint;
- Classes defining most commonly utilized accelerator elements (dipoles, quadrupoles, Wien filters, etc);
- Two versions of element positioning imperfections (tilting):
  - via computing the tilt matrix, and applying it to the computed field at run time (more general but time-consuming, doesn't preserve guiding field strength by default);
  - customized tilting for dipole, WF (less time-consuming, preserves the Lorentz force acting on the particle), and shift for quadrupole;
- Made up of three major classes:

Lattice
: for element management (e.g., tilting, section-plots);

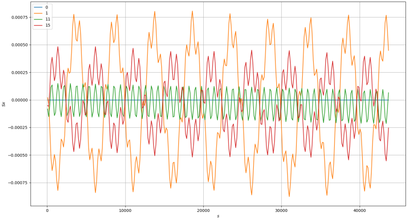
Tracker
: handles integration proper, data backup into an .hdf5 file;

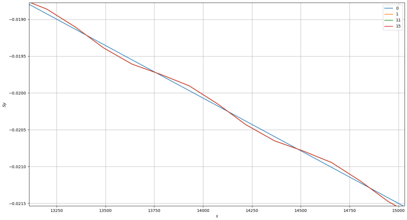
Log
: output of Tracker::track; based off numpy::recarray, handles plotting.

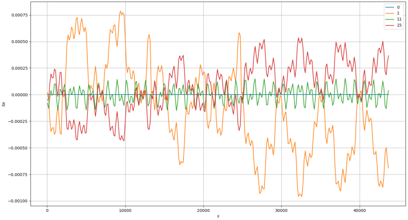
- Vectorized RHS computation.

Example plots

- $S_x \sim s$ in FS lattice w/o tilt
- $S_x \sim s$ in FS structure w/ all elements tilted about $\hat{s}$
- $S_y \sim s$ in FS structure w/ elements randomly tilted Norm(0, 10^{-4} rad)







C++ code

- Python not fast enough; a Frozen spin lattice of 397 elements takes 2 secs/turn to run. (Integration takes $3 - 7 \cdot 10^{-3}$ secs/element, depending on the field complexity.)
- Rewrote the program core in c++ with Boost::Odeint's integrator and Eigen Matrix for field and state type.
- Still speed and precision problems:
  - vectorizing code (via VexCL, most likely);
  - step-size control (this'll probably need redesigning state-type).

Tasks from supervisor

- Study effects of WF tilts (preserves Lorentz force) in FS lattice on S_x, S_y, S_z;
- Same for quadrupole shifts (doesn't preserve LF);
- Study decoherence as a function of the inital beam distribution $(x, y, \delta W)$;
- Study optimal sextupole placement for the suppression of decoherence and chromaticity;
- Modeling of field calibration by effective gamma in the horizontal plane (CW/CCW procedure);

Decoherence test histograms

- 100 turns; 70 trials; 20-particle bunches: $x \sim N(0, 10^{-3})$, $y \sim 10^{-3}$, $dK \sim N(0, 10^{-4})$

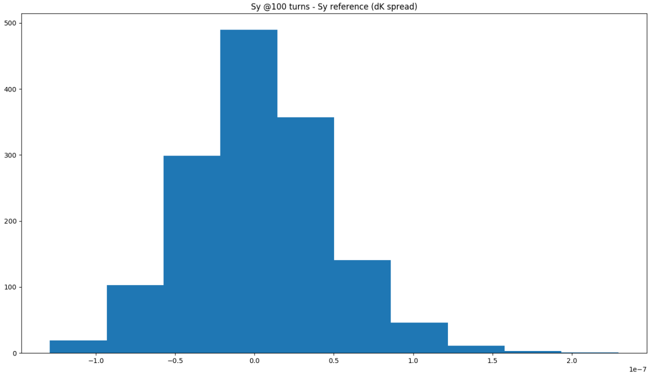


Figure: Histogram from $dK \sim N(0, 10^{-4})$ test

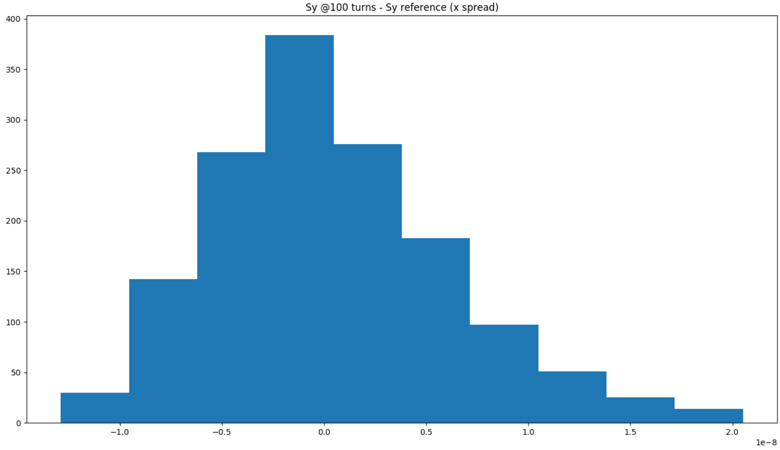


Figure: Histogram from $x \sim N(0, 10^{-3})$ test

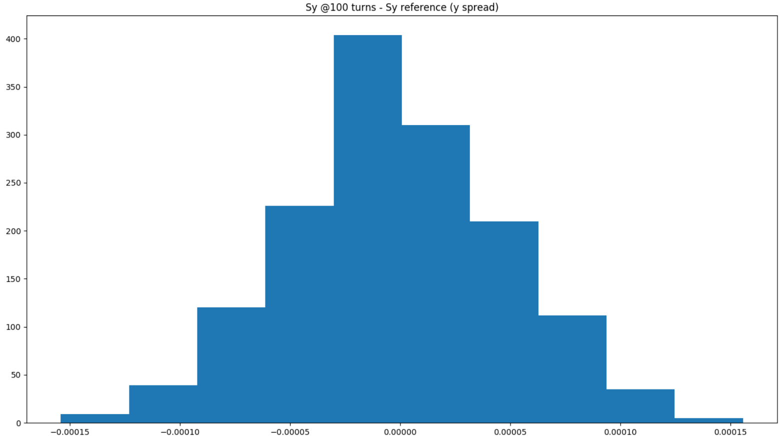


Figure: Histogram from $y \sim N(0, 10^{-3})$ test