Reading IOTA into pyORBIT

Runze Li

June 29 2020

1 Changes in IOTA Lattice

Before reading IOTA into pyORBIT, some modifications on IOTA sequence need to be done to match it with the MADX file parser of pyORBIT.

First, currently the dipole edge element in MADX is not supported by pyORBIT, so all dipole edge elements in IOTA sequence are changed to drift space. This modification of IOTA sequence makes q2 change from 5.3 to 5.42 and chromaticity dq2 from $-5.79 * 10^{-5}$ to 6.345. In order to get accurate result, later on the dipole edge element need to be coded in pyORBIT and added to the lattice.

Also, all apertures in the lattice are removed, since pyORBIT does not read them correctly. After printing out the lattice sequence, all markers with aperture defined has no attribute called "aperture", possible due to some bugs in pyORBIT. As the result they are moved and may also need to be added later on.

Next, all markers, monitors, kickers with 0 strength, and multipoles with 0 strength are removed in order to simplify the lattice. If IOTA needs to be modified to perform some other tasks then these elements may need to be added again.

Finally, the strength of solenoid in IOTA sequence is changed. Previously, it is defined to be 0, which makes its transfer matrix $m_{32} = \frac{-(1-cos(kL))}{k} = 0$. However, setting the denominator to be 0 causes problem in python, so instead the strength of solenoid is set to be 10^{-10} which approximates the 0 strength case.

2 Changes in pyORBIT

There are also several bugs in pyORBIT, and here they are listed and corrected.

First, in the py-orbit/py/orbit/teapot/teapot.py file, the function initialize() of class ApertureTEAPOT initializes the Aperture object with Aperture(shape, dim[0], dim[1], 0.0, 0.0), however the Aperture object is created by Aperture(int shape, double a, double b, double c, double d, double pos) so totaly 6 parameters are needed. Here a dummy variable 0 is added in the end to fix this. However based on current output, the treatment of aperture in pyORBIT still has more problems, since it seems to try to find the element of type "Aperture" in sequence, however, in IOTA sequence the aperture is defined using marker with attribute "aperture". It seems that currently the pyORBIT fails to add markers with apertures in this way.

Second, in py-orbit/py/orbit/matrix-lattice/Matrix-Lattice.py, the method trackTwissData(), which calculates phase advance and twiss functions, is fixed. This method calculates the phase advance in each turn as delta-phi and sum them up as 2π*tune, however the tune on y calculated in this way for IOTA is q2 = 4.42, which is smaller than what is calculated by MADX. The phase advance plot is shown in the left subplot of Figure 1, comparing to the right side plot calculated by MADX which is mono-increasing.

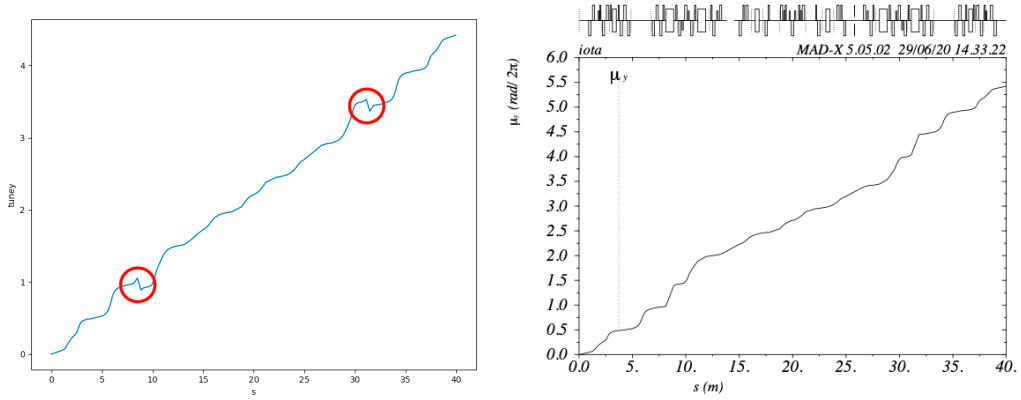

Figure 1: phase advance in y

After printing out the delta-phi value, some of them are found to be negative, which is impossible since they represent the phase advance in each iteration. As a result a condition check is added such that if delta-phi is less

than 0, it is increased by π. After that the both phase advance and phase advance plot agree with the result from MADX.

Also, the default number of parts every lattice node is divided into is modified. Since IOTA is a small ring with short radius, the length of dipoles and quadrupoles is small and the strength is large, every element need to be divided into more slices to make sure the accuracy in twiss function calculation. Here the default set up is using npart = 2 for dipole, quadrupole, multipole, and kickers. This value is now changed to 8. Actually the negative delta-phi described in last paragraph is also caused by the inaccuracy in small nparts. Either changing nparts = 8 or adding π to negative delta-phi fixes this issue, as shown in the corrected phase advance plot in y. All 3 methods give the same result line up with each other, and it is also the same as MADX result in Figure 1.

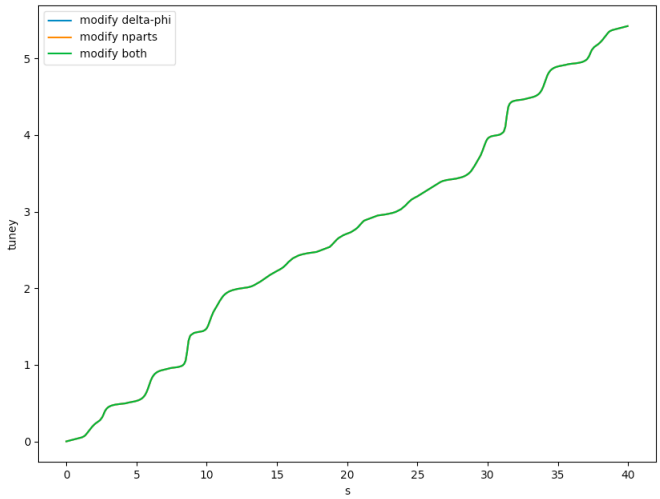

Figure 2: modified phase advance in y

3 Checking Twiss Function

In this part, the twiss parameters: β_x, β_y, q1, and q2 are plotted and compared with twiss plots generated by MADX to make sure IOTA is read into pyORBIT correctly (q2 has been shown in Figure 2 and 1).

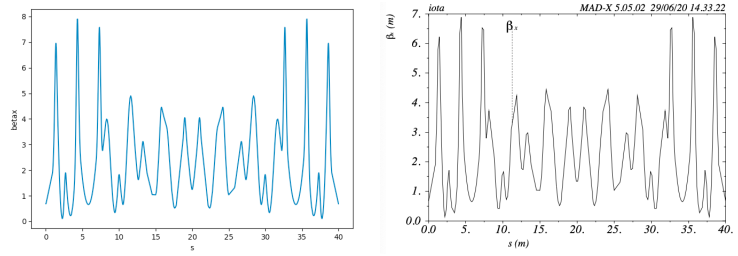

Figure 3: β_x

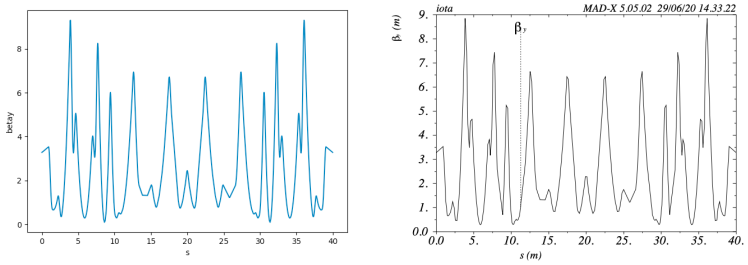

Figure 4: β_y

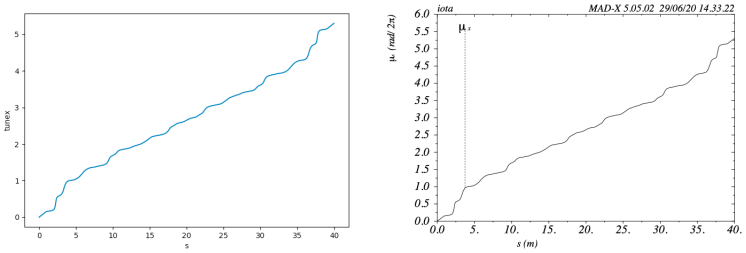

Figure 5: q1

4 Particle Tracking

In this part, 10 particles are tracked for 2000 turns first using MADX and then pyORBIT, and their phase space plots are compared. For IOTA ring the particles are 150MeV electrons. The emittance on x plane is defined to be $0.14 * 10^{-6}$, and the emittance on y plane is defined to be $0.14 * 10^{-8}$. The initial condition for these 10 electrons are (σ_x * i, σ_{px} * i, σ_y * i, σ_{py} * i, 0, 0) where i ranges from 1 to 10. The comparison is shown in Figure 6 and 7, where the left side is the pyORBIT plot and right side is the MADX plot.

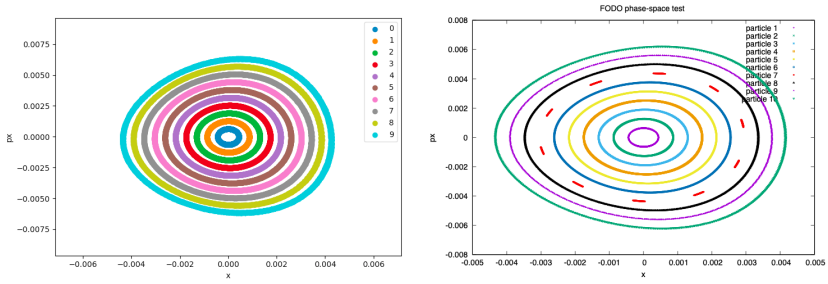

Figure 6: x-px phase space map

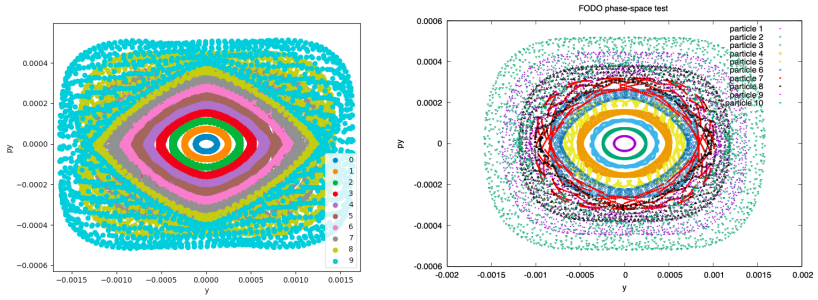

Figure 7: y-py phase space map

According to the scale on each axis, we see that these phase space maps are roughly the same although some extent of difference exists, possibly due to the numerical error. Figure 8 shows the result of pyORBIT scaled by σ

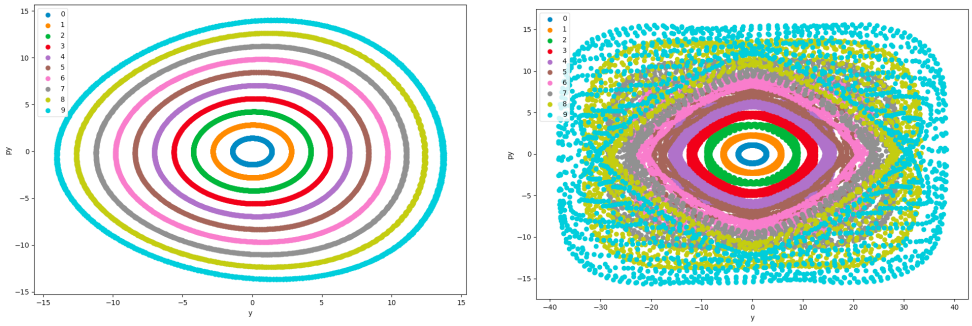

Figure 8: phase space map scaled by $\sigma_{x,y}$

5 Future Work

Currently the remaining bug in pyORBIT is the function TEAPOT-MATRIX-Lattice.getChromaticitiesXY(). However it is an isolated function so we can ignore it for now and focus on other topics.

Currently adding the space charge effect does not bring much difference to the phase space diagram, due to the fact that the field of 10 macro particles are too weak and currently 150MeV electron beam is too energetic. I will change the beam into 2.5MeV proton with more intense particle distribution and check the effect of space charge.

I am still working on learning how to add aperture and dipole edge back to lattice. Also I will learn how to use monitor, teapotTwissAnalysis, and teapotTuneAnalysis. I believe that combining these will enable me to find two quadrupolar modes of IOTA ring under the effect of space charge.