ON THE IMPACT OF NON-SYMPLECTICITY OF SPACE CHARGE SOLVERS*

M. Titze†, CERN, Geneva, Switzerland
and Humboldt-University of Berlin, Germany

Abstract

To guarantee long-term reliability in the predictions of a numerical integrator, it is a well-known requirement that the underlying map has to be symplectic. It is therefore important to examine in detail the impact on emittance growth and noise generation in case this condition is violated. We present a strategy of how to tackle this question and some results obtained for particular PIC and frozen space charge models.

INTRODUCTION

A typical application of a space charge solver is to simulate the behaviour of a beam of charged particles over a reasonably long period of time inside a storage ring. In particular this is the case when studying emittance growth near resonance lines in a tune diagram [1].

On the one hand, it is a well-known fact that the simulation of a system admitting a Hamiltonian has to be symplectic in order to remain on the energy shell [2]. On the other hand, this basic condition is usually violated if one integrates the underlying equations of motion in a straightforward manner. Probably the most simplest example when this happens is the Explicit-Euler method. But also in the sophisticated case of a space charge solver, now acting on the set of bunches in a large dimensional phase space, symplecticity is not necessarily be fulfilled as we shall see.

In this article we present results in which we tested an analytic (Basetti-Erskine) solver, and a so-called $(2+5)$-D Particle-In-Cell (PIC) solver, which are both implemented inside the widely-used space charge tracking program PyORBIT, against the usual symplecticity condition. Both methods involve the addition of so-called space charge nodes at particular steps around the ring, which simulate the result of interaction between the charged particles. Our reference case will be the (uncoupled) plain tracking case obtained with PyORBIT and MAD-X.

The symplecticity checks were performed by using two different, but closely related, methods of numeric differentiation. These methods are straightforward and can basically be applied to any tracking code. We are mainly considering a test ring of 1km circumference with 416 space charge nodes, but also use a FODO map with just 4 nodes.

We will see that, as the reader probably might have expected, up to the precision of our methods the previously mentioned PIC solver violates the symplecticity condition, while the analytic solver is symplectic. We expect that the outcome of this violation might have an influence on long-term studies involving PIC solvers. One such effect which clearly distinguish both methods is the generation of noise in the transversal emittances in the PIC case [3]. The natural question thus arrises whether the symplecticity violation is the main driving term behind this behaviour.

In order to give an indication to the answer, we performed several tracking studies, using a low number of macroparticles, on a FODO cell and a small test ring. There are several reasons for choosing a low number: Firstly, due to the fact that we need at least to check the Jacobi-Matrix, we can not go much higher. Secondly, it turned out that a small ring with reasonable parameters can mimik a similar situation with a large phase-space. However, the outcome is also varying more, which has to be taken care off by simulating the same situation several times.

SYMPLECTICITY CHECKS

Before we are able to apply the numeric differentiation methods, let us remark that PyORBIT is not dumping the beam in canonical coordinates, a fact which must be taken into account.

Numeric Differentiation Method

A straightforward way of how to check the symplecticity of a numerical integrator at a given point x is to approximate its Jacobi-Matrix by 1D fits for every pair of directions. Namely, if $M: P \to P$ denotes the given map from $2k$-dimensional phase space $P \subset \mathbb{K}^{2k}$ to itself, we specify a step size[1] ϵ and approximate $\partial_j M_i(x)$ for a given point x by the slope of a linear fit of the values $M_i(x + k\epsilon b_j)$, $k \in \mathbb{Z}$, where the b_j denotes a basis and M_i the ith component with respect to that basis.

Then the symplecticity condition is checked by computing $R := (M')^{tr} J M' - J$, where $M' = (\partial_j M_i(x))_{ij}$ is the now determined Jacobi-Matrix of M at x and J the matrix representation of the given symplectic structure in the above basis. In the following we will understand by the (Frobenius) norm of R the distance of M at a given point $x \in P$ towards symplecticiticity.

If we assume that in every direction b_j the amount of bunch configurations $x + k\epsilon b_j$ for variing k is the same number K, and if we denote the number of particles by N, we effectively have to track $36N^2K$ times through the ring to compute the entire Jacobi-Matrix. It is therefore not feasible to perform this computation for a large number of particles.

* Work supported by German Federal Ministry of Education and Research (BMBF)

† malte.titze@cern.ch

[1] In general this step size has to be choosen seperately for every direction and component.

ISBN 978-3-95450-178-6

Table 1: Symplecticity Error for Various Codes for a FODO Cell having 4 Space Charge Nodes

Code	$\|R\|_{2D}$
PyORBIT (pure tracking)	$1.4966 \cdot 10^{-5}$
Basetti-Erskine	$2.2518 \cdot 10^{-5}$
PIC	$3.5421 \cdot 10^{-3}$

But for a numeric confirmation of the non-symplecticity this is also not necessary.

2d Fit Method

An alternative way to check the symplecticity condition is based on the observation that around x we can write M in form of a Taylor series

$$M(x+a) = M(x) + M'(x)a + o(|a|^2).$$

Inserting for a the quantities ϵb_j and $\tilde{\epsilon} b_k$, in which ϵ and $\tilde{\epsilon}$ are sufficiently small, we obtain

$$\epsilon\tilde{\epsilon}\langle M'(x)b_j, M'(x)b_k\rangle = o(\epsilon\tilde{\epsilon}^2) + o(\epsilon^2\tilde{\epsilon}) + \langle M(x+\epsilon b_j) - M(x), M(x+\tilde{\epsilon} b_k) - M(x)\rangle.$$

M is then symplectic at x if, in the limit $\epsilon, \tilde{\epsilon} \to 0$, for every pair (j, k) of directions the coefficient in front of the $\epsilon\tilde{\epsilon}$-polynomial, given by the 2D-fit of the values

$$\langle M(x+\mu\epsilon b_j) - M(x), M(x+\nu\tilde{\epsilon} b_k) - M(x)\rangle, \qquad \mu, \nu \in \mathbb{Z},$$

equals $\langle b_j, b_k\rangle$. It is clear that this method works for any symplectic structure $\langle\cdot,\cdot\rangle$ and any basis.

BENCHMARKING RESULTS

Symplecticity Errors

Before we are going to benchmark the codes on our test ring, let us adress the question about which of the codes can we regard as 'symplectic', in the sense that its approximated derivative, given by one of the methods in the previous section, has an error which is so small, that the (uncoupled) drift case leads to a similar error.

To begin with, let us consider the case of a basic FODO cell having just 4 space charge nodes. Table 1 summarizes our findings: It shows that the symplecticity error with respect to the particle model (here 16 particles) is nearly the same for the Basetti-Erskin model and the plain tracking.

On the other hand, we see a rather significant error for the PIC case, which means that the code can hardly be symplectic.

Let us now turn to our model of a 1km ring with several space charge nodes. For the tracking around this ring (no space charge yet), it turns out that we basically require two different families of step sizes: one for the spatial directions and one for the momentum directions of the canonical coordinates (we used $\epsilon_q = 4 \cdot 10^{-4}$, $\epsilon_p = 1 \cdot 10^{-5}$).

Although the ring now contains all 416 space charge nodes and thus the sensitiveness is likely to be lower than the current step size, we can still ask how far the resulting map is away from a symplectic solution. In our case we found, for 16 macroparticles, a derivation of $|R|_{2D} = 8.9790 \cdot 10^{-5}$ and $|R|_{ND} = 3 \cdot 10^{-4}$ in case of the ND-method, so the 2D-fit method gives slightly better results. However, we found that in case of pure MAD-X tracking, we obtained for both modes $|R|_{2D} = 2.4907 \cdot 10^{-6}$ and $|R|_{ND} = 3.0440 \cdot 10^{-6}$ respectively, so the ND-method has almost the same precision here.

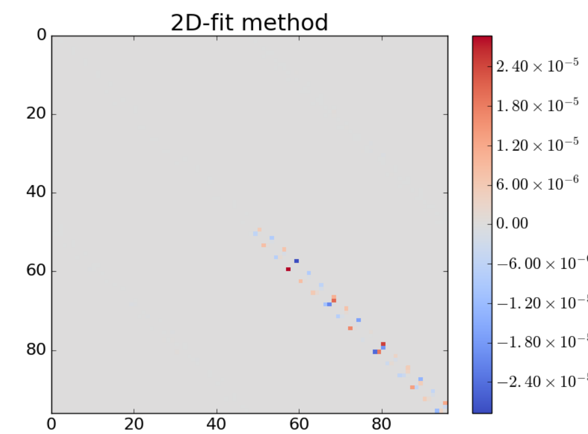

Figure 1: Typcial example of pure drift PIC error matrix R

Fig. 1 shows a typical example of the error matrix we obtained with the PIC solver, determined with the 2D-fit method. The entries in the error matrix usually do not drop below 10^{-6} and we take this scale as a rough lower limit of the precision of our methods.

Figs. 2 and 3 are showing the corresponding residuals of the fits for both methods, respectively, for a typical example of pairs of directions in which the residual were largest, by which we were able to check if the step sizes were choosen appropriately. Here we have used 16 macroparticles, the PIC space charge solver, 3rd order fits in every case and $K = 5$ different values to determine the slope at the midpoint.

The outcome of the corresponding symplecticity checks with space charge are shown respectively in Figs. 4 and 5 for the analytical- and the PIC solver.

Emittance Growth in the Sandbox Model

While starting some tracking simulations for 10k turns, we observed that our sandbox ring mimics roughly the behaviour of the 'large scale' scenario, if parameters are adjusted properly. This means that the growth in the mean of the horizontal and vertical emittances increases as in the large-scale case, and slows down when adding more particles.

However there is also a drawback: Namely the small number of macroparticles leads to a larger fluctuation of the outcome. This means that we have to perform tracking experiments repetitively to obtain results of better reliability.

ISBN 978-3-95450-178-6

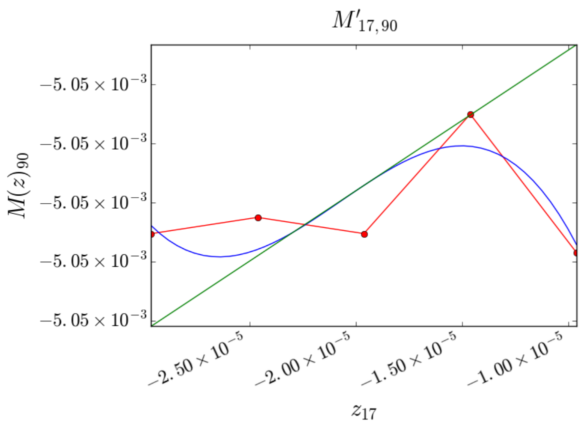

Figure 2: Slope fit of ND method.

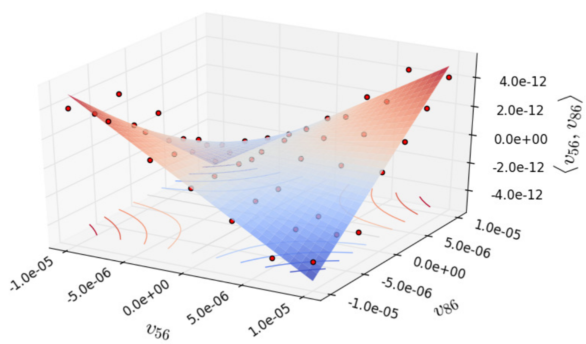

Figure 3: 2D-fit method.

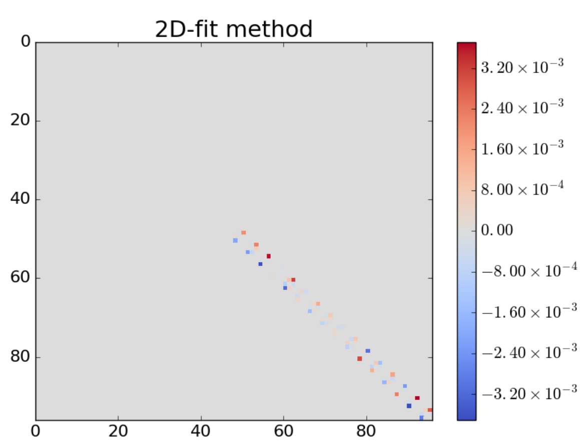

Figure 4: Basetti Erskine model with $|R|_{2D} = 0.0118$.

This was especially the case for particle numbers below approx. 25.

Our first goal was to determine, in dependency of the (random) initial coordinates, a possible correlation between the symplecitcity error and the number of particles. The result can be found in Tab. 2.

As one can see from this table, the error reduces by adding more particles, which might be contrary to the picture that by adding more dimensions, one might add more space and thus be farther away.

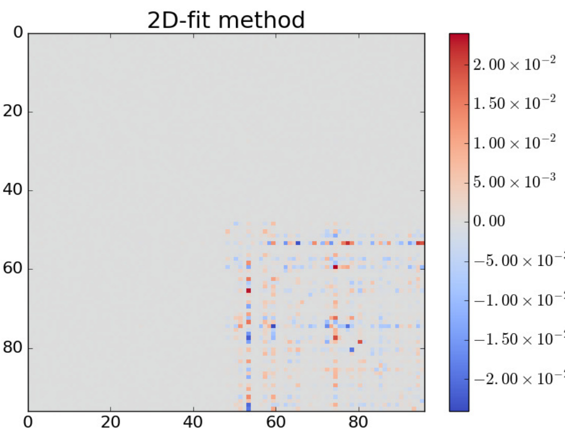

Figure 5: (2 + 5)-D PIC solver with $|R|_{2D} = 0.14461$

Table 2: Mean Values of Initial Error Matrices with Space Charge

Number of particles	$\lvert R\rvert_{ND}$	$\lvert R\rvert_{2D}$
8	3.1370	1.4629
12	0.8866	0.4711
16	0.3293	0.1950
20	0.2100	0.1063
24	0.1673	0.0877

Furthermore, we were looking at how the error in symplecticity evolves with the number of turns. Our results are summarized in the next four Figures 6 to 9 in which we were tracking a system of particles over 10k turns. The green curves shows the mean of the vertical and horizontal emittances. At every 500 turns we dumped the beam to a file and determined the error of the derivative of the one-turn map at that given point towards symplecticity (blue curves). The straight lines indicate regression fits of the green and blue data points respectively.

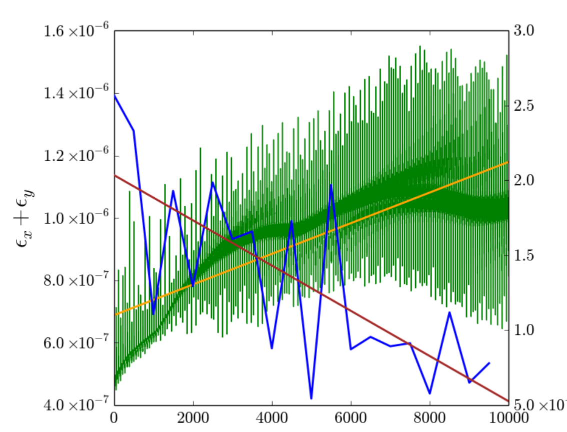

Figure 6: Mean emittance growth with 8 particles. Description: see text body.

ISBN 978-3-95450-178-6

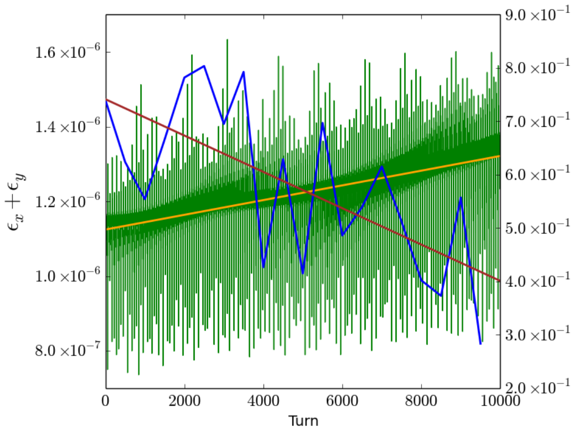

Figure 7: 12 particles.

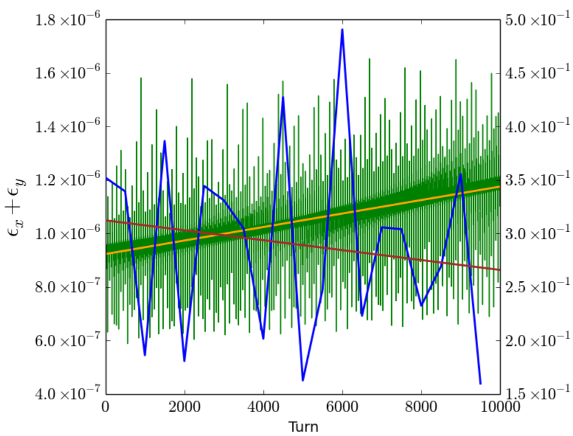

Figure 8: 16 particles.

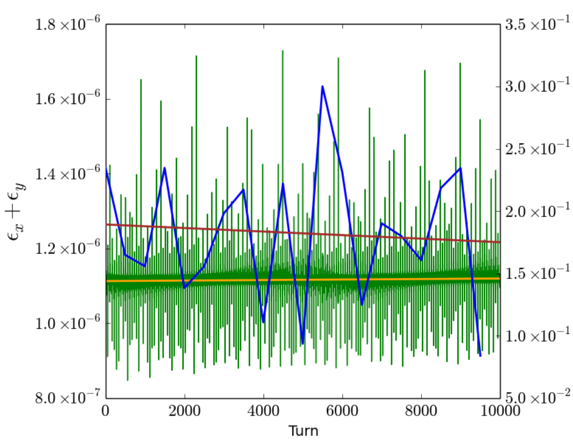

Figure 9: 20 particles.

These last benchmarkings indicate that there might be a correlation between the slopes of the emittance growth and the symplecticity errors. Our current explanation is that if the emittance increases, so does the phase space. If the particles are spread out more in phase space, there is less interaction between them, which means that we drop more and more into (symplectic) single particle tracking.

Of course, using regression lines may hide essential features and may not be appropriate here. But for a first try it should be good enough. In Fig. 10 we have plotted the slopes against each other, using 95% confidence intervals of the regression fits as error bars. It is clear that we require more data points to make further assertions, but a correlation is already been visible.

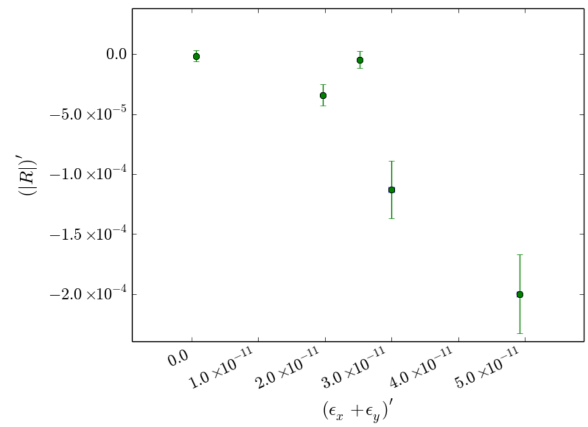

Figure 10: Slope of mean vert. and hor. emittances vs. slope of symplecticity error (2D-fit method here).

CONCLUSION

By using numeric differentiation methods we were able to determine the symplectic - respectively - non-symplectic nature of our space charge solvers. Furthermore we found certain correlations between the errors of symplecticity of the ring-map and the mean emittances growth of the beam.

We are currently gathering more data in oder to improve our picture and understanding.

ACKNOWLEDGEMENT

We would like to thank A. Oeftiger for many fruitful discussions and his essential help with certain coding issues. We also thank F. Schmidt, also for many fruitful discussions, H. Bartosik for his help with PyORBIT and last but not least we want to thank the sponsoring by the Wolfgang-Gentner-Program of the German Federal Ministry of Education and Research (BMBF).

REFERENCES

[1] H. Bartosik, Y. Papaphilippou, M. Benedikt, “Beam dynamics and optics studies for the LHC injectors upgrade”, in *CERN-THESIS-2013-257*, CERN, Geneva, Switzerland, Nov. 2013.

[2] H. Goldstein, "Klassische Mechanik", Akademische Verlagsgesellschaft, Frankfurt a.M., 1963.

[3] J. Struckmeier, “Concept of entropy in the realm of charged particle beams”, *Phy. Rev. E*, vol. 54, no. 1, p. 830 - 837, Jul. 1996.

ISBN 978-3-95450-178-6