Computation of fixed points in a circular machine

A. Verdier and J. Manuch*, CERN, Geneva, Switzerland

Abstract

This paper describes an algorithm to compute the fixed points (closed orbit or higher order fixed points) in a circular machine together with the linear transformation around it by tracking trajectories on a small number of turns. It can be applied to a phase space with an arbitrary number of dimensions. It is particularly useful to check the consistency between trajectory tracking and transfer matrix calculation in any optics code.

1 INTRODUCTION

The computation of closed orbit coordinates and its associated transfer matrices was proposed a long time ago for the 2-D case [1]. It made it possible to find a bug in the old AGS [2] optics program and to make it possible to find the off-momentum closed orbit coordinates in any case.

This algorithm has been generalised to the 6-D case for any fixed point periodicity. It can be applied in its actual form to any dimension number, 6 being the value needed for betatron oscillations in accelerators. Some basic checks have been done and the numerical accuracy has been estimated. The generalisation to non-linear computation is suggested.

2 THE ALGORITHM.

In the most general circular machine, a given trajectory is defined by its 6 conjugate coordinates expressed with respect to any origin. If the transverse motion is not chaotic, it can be considered as an oscillation around a periodic orbit of order p. At a given location of the machine, the periodic orbit is represented as a fixed point which has the same coordinates after a number of turns p. Let's denote X_{pi} the 6-vector giving the coordinates of the trajectory at the pi^{th} passage. X_c is the 6-vector giving the coordinates of the fixed point. The amplitude of the oscillation around the fixed point at the pi^{th} passage close to the fixed point is defined by the vector $\mathcal{X}_{pi}$ and we have : $X_{pi} = X_c + \mathcal{X}_{pi}$.

If we make the difference $\mathcal{D}_i$ between the actual coordinates at the passages pi and $p(i-1)$, we write :

$$\begin{aligned} \mathcal{D}_i &= X_{pi} - X_{p(i-1)} \\ &= [X_c + \mathcal{X}_{pi}] - [X_c + \mathcal{X}_{p(i-1)}] \\ &= \mathcal{X}_{pi} - \mathcal{X}_{p(i-1)} \end{aligned}$$

The vector $\mathcal{D}_i$ can hen be computed from the actual coordinates obtained from tracking. Let's call M the linear transfer matrix over one fixed point period which describes oscillations around the fixed point with infinitely small amplitudes. By definition :

$$\mathcal{X}_{pi} = M\mathcal{X}_{p(i-1)} \tag{1}$$

Adding $\mathcal{X}_{p(i-1)}$ on both sides of equation (1) and reexpressing it on the right by means of equation (1) rewritten by changing i to $i-1$, we obtain :

$$\mathcal{D}_i = M\mathcal{D}_{(i-1)} \tag{2}$$

It is now possible to construct a matrix equation by putting the k vectors $\mathcal{D}_1, \mathcal{D}_2, .., \mathcal{D}_k$ side by side. Using equation (2), we obtain readily :

$$(\mathcal{D}_2, \mathcal{D}_3, .., \mathcal{D}_{k+1}) = M(\mathcal{D}_1, \mathcal{D}_2, .., \mathcal{D}_k) \tag{3}$$

from which the transfer matrix M can be obtained by :

$$M = D_2 {D_1}^{-1} \tag{4}$$

The columns of the $k \times k$ matrix D_1 are made with the column vectors $\mathcal{D}_1, \mathcal{D}_2, .., \mathcal{D}_k$ while those of the matrix D_2 are made with the vectors $\mathcal{D}_2, \mathcal{D}_3, .., \mathcal{D}_{k+1}$.

A first consequence of this calculation is that tracking a trajectory over $p(k+1)$ periods is sufficient to obtain the elements of the transfer $k \times k$ matrix around a fixed point of period p. It is not necessary since the symplecticity condition imposes to the k^2 elements of M, $(k-1)k/2$ independent relations and reduces the number of equations needed to determine the matrix. However it is better to keep equation 4 to determine the transfer matrix since our algorithm is aimed at checking optics calculations. In this respect, it is useful to check the symplecticity condition on this matrix.

Once the transfer matrix is obtained, it is straightforward to compute the fixed point coordinates by simple algebra. We obtain :

$$X_c = (M - I)^{-1}(M\mathcal{X}_0 - \mathcal{X}_1)$$

It is extremely important to realize that the algorithm is exact for infinitesimal oscillation amplitudes. As the coordinates obtained by tracking are always finite, it is obvious that the algorithm **must be iterated** in order to obtain relevant results. This is not a problem as the fixed point coordinates obtained in the first iteration provides already a good approximation of a trajectory with a small oscillation amplitude.

It is important to note that, in this context, it does not make sense to compute a detuning with amplitude by means of this algorithm by computing the tunes around a closed orbit (p=1) from the matrix M which varies depending on the starting point of the trajectory.

* present adress, Comenius University of Bratislava, Slovakia

3 PRACTICAL IMPLEMENTATION

The programs fm6D and fm4D have been written in C^{++} to compute the fixed points coordinates and the matrix M defined above from the files of coordinates created by the MAD [3] program. In fact the dimension of the coordinate vector can be changed easily as the code uses general algorithms for $2n$ dimensions. The two versions respectively in 4D and 6D have been made for convenience.

There is a general portability problem concerning the number of digits used in the tracking tables. It is of course extremely important to use the maximum accuracy for the tables. The best solution is obviously to use the algorithm inside the program itself, so that this problem disappears.

The two codes use the MAD algorithm [4] to compute the eigen-values an "tunes" of the oscillations around the fixed point.

The errors inherent to the algorithm have been investigated for the case of a computation around a closed orbit (fixed point of order 1). Because of the finite amplitude of the oscillations, there is a detuning due to the non-linearities present in the machine which varies depending on the sequence of turns chosen for the computation. This is not worrying since the computation has to be iterated. Usually two iteration give the exact tunes around a standard closed orbit. After the calculations, the coordinates computed with the closed orbit and the transfer matrix are compared with those obtained by tracking. Their difference is given in order to have an additional idea on the accuracy of the calculation. For instance, if the oscillation amplitude is too large, the difference between the absolute value of the oscillation amplitude computed with the transfer matrix and that obtained from tracking will be of the same order of magnitude as the oscillation itself on average.

4 TESTS

The algorithm has been tested to check the off-momentum calculation of the tunes in MAD. In a previous version of the code there was a second order in momentum deviation missing. This has been corrected successfully and checked with the present algorithm applied to the off-momentum closed orbit calculation.

The accuracy problem has been investigated by adding random errors to the tracking table. It has been proved to be useful to keep this option for checking purpose as the errors can be large if the differences between the successive coordinates are too small. This can be done by means of an optional parameter in the program command. It has been observed that a small relative error of some 10^{-5} in the 6^{th} component makes an error as large as the value of the closed orbit coordinates themselves (amplitudes of the order of some milimeter for a dynamic aperture of the order of 10mm).

5 EXTENSION OF THE ALGORITHM

Two generalisations can be foreseen, they have not yet been implemented.

- In order to have a more reliable value of the matrix in a single pass computation, we can use more turns than needed and compute the matrix which fits the best the oscillation hypothesis. We can write the relationship 4 but now the matrices D_1 and D_2 are rectangular. The solution of the minimisation problem is :

  $$M_f = D_2 {D_1}^T (D_1 {D_1}^T)^{-1}$$

  The iterative procedure gives actually much better results. This procedure could be useful only to check calculations after a long tracking has been performed.

- Instead of computing a linear transfer map, it is possible to compute a non-linear one. The number of turns is simply larger since there are more coefficients to compute. It is not obvious that the accuracy of such a procedure gives relevant results. However this is the only possibility to obtain for instance detuning with amplitude from this algorithm.

6 CONCLUSION

The computation of optics quantities like tunes from tracking is useful for checking purposes. The symplecticity of the tracking can be checked, as well as the consistency of calculations of tunes with transfer matrices of FFT for a machine with errors or for off-momentum orbits. Further developments of the algorithm could concern the study of stability around higher order fixed points.

7 REFERENCES

[1] T. Risselada and A. Verdier, Improved calculation with the AGS program for the off-momentum orbits. Divisional report CERN/ISR-BOM-OP/79-38.

[2] E. Keil et al., AGS - The ISR computer program for synchrotron design, orbit analysis and insertion matching. CERN 75-13.

[3] H. Grote and F.C. Iselin, The MAD program (Methodical Accelerator Design) version 8.16, User's reference manual, CERN/SL/90-13(AP), (rev. 4) (March 27, 1995).

[4] The physics incorporated in the MAD program can be found on the MAD web page : wwwslap.cern.ch/fci/mad/mad8/request.html