

CONVERGENCE MAP WITH ACTION-ANGLE VARIABLES BASED ON SQUARE MATRIX FOR NONLINEAR LATTICE OPTIMIZATION

L. H. Yu, Y. Hao, Y. Hidaka, F. Plassard, V. Smalyuk
Brookhaven National Laboratory, NY11973,USA

Abstract

We apply square matrix method to obtain in high speed a "convergence map", which is similar but different from frequency map [1]. The convergence map is obtained from solving nonlinear dynamical equation by iteration of perturbation method and study the convergence. The map provides information about the stability border of dynamical aperture. We compare the map with frequency map from tracking. The result indicates the new method may be applied in nonlinear lattice optimization, taking the advantage of the high speed (about 10~50 times faster) to explore x , y and the off-momentum phase space.

INTRODUCTION: SQUARE MATRIX EQUATION FOR NONLINEAR DYNAMICS

We consider the equations of motion of a nonlinear dynamic system with periodic structure such as Hills equation, it can be expressed by a square matrix.

If we use the complex Courant-Snyder variable $z = x - ip$, its conjugate and powers z, z^*, z^2, \dots as a column Z , the one turn map can be represented by a large square matrix M using

$$Z = MZ_0.$$

All square matrices can be transformed into Jordan form [2, 3]. A detailed description is given, e.g., in [4, 5]. For any given square matrix M , for an eigenvalue μ which is the linear tune, there is a transformation matrix U and a Jordan matrix τ so that every row of the matrix U is a (generalized) left eigenvector of M satisfying

$$UM = e^{i\mu I + \tau} U \quad (1)$$

For the case of 4 variables x, p_x, y, p_y (to be abbreviated as X in the following), at 3rd order, the column becomes $Z = z_x, z_x^*, z_y, z_y^*, z_x^2, z_x z_x^*, \dots$, the matrix M is a 35×35 matrix, I is a 2×2 identity matrix, the matrixes U_x and U_y are 2×35 transformation matrix, while the Jordan matrix τ also has dimension 2 for both eigenvalue μ_x and μ_y . For tune μ_x or

μ_y , the Jordan matrix τ has the form $\tau = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$.

As after one turn $Z_1 = MZ_0$, Eq. (1) gives

$U_x Z = U_x MZ_0 = e^{i\mu_x I + \tau} U_x Z_0$. Define a transformation $W_{x0} \equiv U_x Z_0, W_{x1} \equiv U_x Z_1$. And, for y , there are correspondingly U_y, W_y , respectively, so in the next paragraph we neglect the subscript for x, y :

W represents the projection of the vector Z onto the invariant subspace spanned by the left eigenvectors u_j given by the j 'th row of the matrix U , such that j 'th row of W is $w_j = u_j Z$, a polynomial of z_x, z_x^*, z_y, \dots . Then Eq. (1)

implies the operation of one turn map $Z_1 = MZ_0$ corresponds to a rotation in the invariant subspace represented by $W_1 = e^{i\mu I + \tau} W_0$. After n turns, if linear tune is μ and tune shift is ϕ , the particle has a tune $\mu + \phi$, we must have $W = e^{in(i\mu I + \tau)} W_0 \equiv e^{in(\mu + \phi)} W_0$. Hence we must have $\tau W_0 \equiv i\phi W_0$. For the case of 4 variables x, p_x, y, p_y at 3th order, U is a 2×35 matrix and $W_0 = \{w_0, w_1\}$ are two polynomials each has 35 terms of monomials with power up to 3rd order, and $\tau W_0 \equiv i\phi W_0$ gives $\phi = -iw_1/w_0$ as the tune shift. This has been confirmed using tracking results for many lattices,

And, in particular, our tracking results confirmed that the polynomials $w_{x0}, w_{x1}, w_{y0}, w_{y1}$, are excellent action-angle variables: after each turn they are multiplied by a factor $e^{in(\mu + \phi)}$ to a good approximation, ie. the trajectory represented by these variables follow a circle with very small deviation from a rotation with uniform rotation speed. Hence we can use their linear combination to form two better action-angle variables, i.e., with smaller deviation from a circle. For the 4 variable case in the following, to distinguish x from y , we have

$$\begin{aligned} v_1(\theta_1) &\equiv r_1 e^{i\theta_1} \equiv a_{11} w_{x0}(x, p) + \\ &a_{12} w_{x1}(x, p) + a_{13} w_{y0}(x, p) + a_{14} w_{y1}(x, p) \quad (2) \\ v_2(\theta_2) &\equiv r_2 e^{i\theta_2} \equiv a_{21} w_{x0}(x, p) + \\ &a_{22} w_{x1}(x, p) + a_{23} w_{y0}(x, p) + a_{24} w_{y1}(x, p) \end{aligned}$$

We establish one turn map using θ_1, θ_2 as dynamic variables, i.e, we consider $\theta_{1,k+1}, \theta_{2,k+1}$ as function of θ_{1k}, θ_{2k} (k is the turn number): For a given θ_{1k}, θ_{2k} we find X_k by the inverse function of Eq. (2). From the one turn map for X_k we find X_{k+1} . Then we use Eq. (2) again to obtain $v_1(\theta_{k+1}), v_2(\theta_{k+1})$, then use them to obtain $\theta_{1,k+1}, \theta_{2,k+1}$. This one turn map between $\theta_{1,k+1}, \theta_{2,k+1}$ and θ_{1k}, θ_{2k} can be calculated exactly, independent from the power order of M , and we can derive an exact equation, even if we take only 3rd order for the Jordan decomposition. If there are quasi-periodic solution we use $\alpha_1 \equiv k\omega_1 + \theta_0, \alpha_2 \equiv k\omega_2 + \theta_0$ to represent phase advance as turn number k , the equation for $\theta_1(\alpha_1, \alpha_2), \theta_2(\alpha_1, \alpha_2)$ is

$$\begin{aligned} \theta_1(\alpha_1 + \omega_1) - \theta_1(\alpha_1) &= \omega_1 + \Delta\phi_1(\theta_1(\alpha_1, \alpha_2), \theta_2(\alpha_1, \alpha_2)) \\ \theta_2(\alpha_2 + \omega_2) - \theta_2(\alpha_2) &= \omega_2 + \Delta\phi_2(\theta_1(\alpha_1, \alpha_2), \theta_2(\alpha_1, \alpha_2)) \quad (3) \end{aligned}$$

where $\omega_1 \equiv \overline{\phi_1}, \omega_2 \equiv \overline{\phi_2}$ is the phase advance rate of v_1, v_2 per turn, averaged over many turns, corresponding to x, y tunes. $\Delta\phi_1, \Delta\phi_2$ are the fluctuation from ω_1, ω_2 .

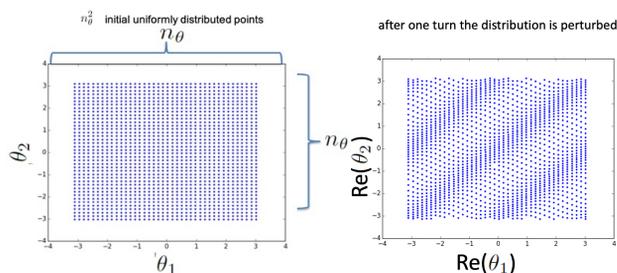


Figure 1: Points in initial $\theta_1^{(0)}(\alpha_1, \alpha_2), \theta_2^{(0)}(\alpha_1, \alpha_2)$, and their distribution after one turn.

If v_1, v_2 are sufficiently close to a pure rotation, then $|\Delta\phi_1|, |\Delta\phi_2| \ll 1$. The real part of $|\Delta\phi_1|, |\Delta\phi_2|$ are the phase fluctuation, and its imaginary part is the amplitude fluctuation. This exact equation can be solved by perturbation: initially we take zero order approximation $\theta_1^{(0)} = \alpha_1, \theta_2^{(0)} = \alpha_2$. When this is substituted to the right hand side of Eq. (3), since $|\Delta\phi_1|, |\Delta\phi_2| \ll 1$ the error is of second order. Then the equation can be solved using Fourier transform, the result is the first order approximation $\theta_1^{(1)}, \theta_2^{(1)}$. Then $\theta_1^{(1)}, \theta_2^{(1)}$ is substituted in the right hand side of Eq. (3) to obtain $\theta_1^{(2)}, \theta_2^{(2)}$ as 2nd order approximation. This process can be iterated to generate convergent solution to high precision if the amplitude of zero'th order $X^{(0)}$ (obtained from $\theta_1^{(0)}, \theta_2^{(0)}$) is sufficiently close the origin, or, within the dynamical aperture.

The main issue is the convergence of this iteration process. In the case in the neighbourhood of a pure rotation, KAM theory [6–9] proved the existence of analytical solution. In our practical application, we study the periodic solution exploring area with large amplitude or near resonance, numerically. In our application of this method, we add one step to each step in the iteration: we minimize the fluctuation by varying the coefficients a_{1j}, a_{2j} in Eq. (2). Because this step always reduce the fluctuation, it makes the intermediate solution more close to a pure rotation, and hence makes the convergence faster and within larger range. With these provision, we discuss the numerical application of this iteration steps in the following.

NUMERICAL APPLICATION: CONVERGENCE MAP

In practical numerical application, α_1, α_2 take only discrete number of points within $(0, 2\pi)$, one of the main parameters is the number of points n_θ . In Fourier transform of Eq. (3) the variables θ_1, θ_2 and α_1, α_2 also take discrete values at n_θ^2 points in the θ_1, θ_2 plane. Figure 1 is an illustration of $\theta_1^{(0)}(\alpha), \theta_2^{(0)}(\alpha)$ and their result of tracking them one turn.

In the following example of numerical application of the iteration steps, we use the matrix M derived from one of the lattices for NSLSII storage ring “20140204_bare_1supcell”. We use four polynomials $w_{x0}, w_{x1}, w_{y0}, w_{y1}$ derived from Jordan vector of power order 3, and corresponding linear combination coefficients $\{a_{1j}^{(n)}, a_{2j}^{(n)}, j = 0, 1, 2, 3$ in step n . Inini-

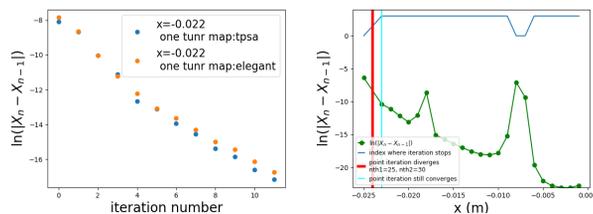


Figure 2: a) $\ln(\delta X_n)$ vz. Iteration number n b) Minimum of $\ln \delta X_n$ vz. x , the red line is where iteration diverges for both $n_\theta = 25$ and 30.

tially we take $a_1 = 1, 0, 0, 0, a_2 = 0, 0, 1, 0$. So initially we only use w_{x0}, w_{y0} for $v_1^{(0)}$ and $v_2^{(0)}$ respectively, as iteration n increases the contribution from $a_{11}, a_{12}, a_{13}, a_{20}, a_{21}, a_{22}$ vary from 0 to improve the precision of $X^{(n)}$ closer to real trajectory.

For a trajectory starting from $x = -22$ mm, $y = 4$ mm, $\delta p = -0.025$, very close to dynamic aperture, when we take $n_\theta = 12$, the iteration leads to convergence as shown in Fig. 2a. For each step, we calculate the solution X_n at step n . The difference between the solution at step n and $n - 1$ is $\delta X_n = X^{(n)} - X^{(n-1)}$. We use the sequence the RMS value of δX_n as Cauchy criterion for convergence test. The convergence is indicated by the exponential decay of δX_n as function of n in Fig. 2a. When we use tracking by “elegant” [10] to calculate the one turn map from x_k, p_k to x_{k+1}, p_{k+1} , as the right hand side of Eq.(3), the minimum of $\ln(\delta X_n)$ (orange) is -17 at the last iteration labeled as ‘ELEGANT’. Another way to use square matrix of power order 5 constructed using Truncated Power Series Algebra (tpsa) [11–15]. The result is the blue dots (“tpsa”) with the minimum at -17.5. Notice that even though the Jordan vector for the action-angle variables is of power order 3, the one turn map can be exact, as given by “elegant” tracking, or approximated by square matrix of power order 5 or 7. The results have almost same convergence rate, and at the end of the iteration, the trajectory difference is very small (order of $e^{-17} = 4 \times 10^{-8}$ mm).

For a scan from $x = -1$ mm to -18 mm for every mm, we plot the minimum of iteration for each x in Fig. 2b, with $n_\theta = 12$. The maximum iteration number is set at $n = 4$. The blue curve is the number of iterations reached vs. x . For $x < -25$ mm and the resonance at $x = -8$ mm the iteration diverges while for other x the iteration converges. The red line in Fig. 2b represents the position where the iteration for $n_\theta = 25, 30$ both diverge. This provides information about dynamical aperture and the relation between divergence and n_θ . When n_θ increases the dynamical aperture reduces. This topic of the relation between maximum n_θ of convergence, iteration stability, and the aperture will be addressed in future publication because the limited space.

The same iteration for initial value over the xy plane with initial value of $p_x, p_y = 0, \Delta p = -2\%$ is shown in Fig. 3a with $n_\theta = 10$. The minimum of $\ln(\delta X_n)$ is represented by the color scale. For the total number of scan points $81 \times 41 = 3321$, the cpu time for this map is 131 seconds using 1 cpu core of the computer. The white area represents

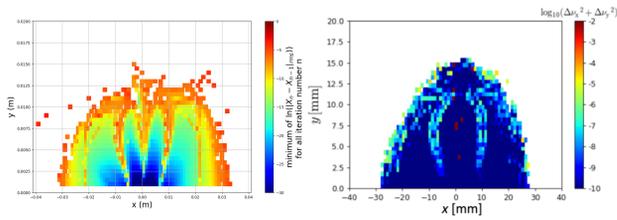


Figure 3: a) $\ln(\delta X_n)$ as color scale for iteration scan over xy plane for $\Delta p = -2\%$ b) Frequency diagram for same scan from 'elegant'.

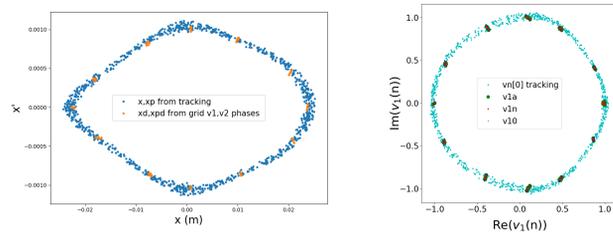


Figure 4: a) Trajectory in xp_x plane within the range $-3 \text{ mm} < y < 3 \text{ mm}$, blue (tracking by "elegant"), orange (square matrix iteration). b) Trajectory in v_1 complex plane (tracking: light blue) (square matrix iteration:red).

divergence of the iteration. We refer this map as a convergence map. As a comparison, the frequency map [1] using same number of points is shown in Fig. 3b. The cpu time with same computer used 40 core with 40 seconds, corresponding to 1600 seconds for one core. These two maps are entirely different maps, but both provide similar space structure. This suggests that the convergence map can be used in the nonlinear lattice optimization.

We use one point as an example to show the precision of the iteration result, i.e. initial value $x = -22 \text{ mm}$, $y = 4 \text{ mm}$, $\Delta p = -2.5\%$. Figure 4a compares trajectory in x, p_x plane calculated from elegant and from last iteration. The blue points are tracking, the 12 points (orange) are the iteration result. Figure 4b show the trajectory in v_1 complex plane. Dark blue dots represent initial trial $v_1^{(0)}$, calculated with initial $\theta_1^{(0)}(\alpha_1), \theta_2^{(0)}(\alpha_2)$ and $n_\theta = 12$. The red dots represent the result of $v_1^{(n)}$ at the end of the iteration. The light blue dots represent $v_1^{(n)}$ calculated from tracking trajectory x, p_x, y, p_y for 1024 turns. There is a very good agreement between tracking and iteration results.

In tracking result, the particle lost at $N = 5448$ turns. Similarly when n_θ increased to 17, the iteration diverges. There is a qualitative relation between n_θ^2 when iteration diverges and number of turns N when particle lost, obtained from numerical experiences. We have some intuitive understanding of this relation, but lack an analytical analysis so far.

The spectrum of $v_1^{(n)}$ for $x = -21 \text{ mm}$ is shown in Fig. 5a. The main peaks are normalized 1 (outside of the plot). The fluctuation peaks (red-yellow $v^{(n-1)}$ and green $v^{(n)}$ dots) agree well with tracking blue with peak (magenta dots) even with limited $n_\theta = 12$ for $x = -21 \text{ mm}$. Figure 5b shows the tune footprint in v_x, v_y plane, the square matrix tune at the last iteration agree well with tracking except for points

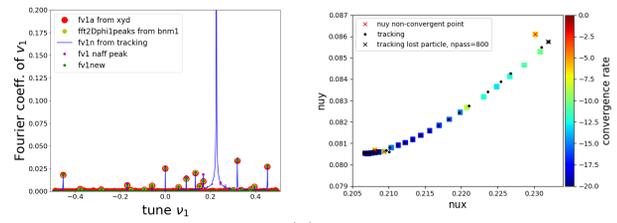


Figure 5: a) Spectrum of $v_1^{(n)}$ at the last iteration n . b) tune footprints in v_x, v_y plane.

where the iteration diverges and particle lost in the tracking (represented by crosses).

ITERATION SPEED

The time spent on a convergence map consists of two parts. The first part is the construction of square matrix using data extract from lattice file (e.g. an 'elegant' lte file). For a 5th order 126×126 matrix used for one turn map and a 3rd order 35×35 matrix with its Jordan decomposition, the time is 1.7 sec. The second part is the iteration time for all the grid points on xy plane. In the example we tested for Fig. 3 the number of points is $81 \times 41 = 3321$, the iteration time is 131 sec for one core on the cluster at NSLSII. So the time of iteration per point is $131/3321=39\text{ms/core}$. For the frequency diagram of Fig. 3b, for the same number of points, the tracking time using 40 core on the same cluster is 40 sec. So the time is $40 \times 40/3321 = 480\text{ms/core}$.

If the number of points used in the optimization is less, for example if it is 280 points for a map, then the time for one map is $280 \times 39 \text{ ms} + 1.7 \text{ sec} = 12.6 \text{ sec/core}$ for iteration by square matrix method. The time for tracking would be $280 \times 0.48 \text{ sec} = 134.4 \text{ sec}$ for a map. So for this lattice for one superperiod of the ring, the ratio of time on square matrix over tracking is $134/12.6 = 11$ times faster. However if we need to study a lattice without 15 fold symmetry (e.g. , a lattice with errors), the time ratio will be different. In this case the time on square matrix construction increases by 15 times. The iteration time would be the same as before, so the total time is $280 \times 39 \text{ ms} + 15 \times 1.7 \text{ sec} = 36.4 \text{ sec/core}$. The tracking time would be increased by 15 times to 2016 sec. The ratio would be $2016/36.4 = 55$ time faster.

CONCLUSION

The action-angle variables derived from square matrix method is close to a pure rotation, so that an iteration steps developed using perturbation method to solve the nonlinear dynamical equation are convergent up to dynamical aperture. The convergence rate is plotted over xy plane as color scale. This convergence map is similar but different from frequency map calculated by tracking. The results agree with tracking well on dynamical aperture, tune footprint and space trajectory, and frequency spectrum. The iteration method is about 10~50 times faster than tracking, hence it has a potential to be used for nonlinear optimization.

REFERENCES

- [1] L. Nadolski, J. Laskar, and J. Irwin, "Review of single particle dynamics for third generation light sources through frequency map analysis", *Phys. Rev. ST Accel. Beams*, vol. 6, no. 11, pp. 114801, Nov. 2003.
doi:10.1103/PhysRevSTAB.6.114801
- [2] R. Kastrom, "An algorithm for numerical computation of Jordan normal form of a complex matrix", *ACM Trans. Math. Softw.*, vol. 6, pp. 398-419, 1980.
doi:10.1145/355900.355912
- [3] R. Kastrom, "Algorithm 560JNF: An algorithm for numerical computation of Jordan normal form of a complex matrix", *ACM Trans. Math. Softw.*, vol. 6, pp. 437-443, 1980.
doi:10.1145/355900.355917
- [4] L. H. Yu, "Analysis of nonlinear dynamics by square matrix method", *Phys. Rev. Accel. Beams*, vol. 20, no. 3, pp. 034001, Mar. 2017.
doi:10.1103/PhysRevAccelBeams.20.034001
- [5] L.-H. Yu and B. Nash, "Linear Algebraic Method for Non-Linear Map Analysis", in *Proc. 23rd Particle Accelerator Conf. (PAC'09)*, Vancouver, Canada, May 2009, paper TH6PFP067, pp. 3862-3864.
- [6] A. J. Lichtenberg and M. A. Lieberman, *Regular and Chaotic Dynamics*, New York, Springer-Verlag New York, 1992.
- [7] A. Bazzani, E. Todesco, G. Turchetti, and G. Servizi, "A normal form approach to the theory of nonlinear betatron motion", CERN, Geneva, Switzerland, Rep. CERN-1994-002, 1994.
- [8] Henk W. Broer, "KAM theory: The legacy of Kolmogorov's 1954 paper", *Bulletin (New Series) of The American Mathematical Society*, vol. 41, no. 4, pp. 507-521, 2004.
doi:10.1090/S0273-0979-04-01009-2
- [9] V. I. Arnold, "Small denominators. I. Mapping of the circumference onto itself.", *Collected Works: Representations of Functions, Celestial Mechanics and KAM Theory*, vol. 46, pp. 213-284, 1965. doi:10.1007/3-540-26866-9_2
- [10] M. Borland, "elegant: a Flexible SDDS-Compliant Code for Accelerator Simulation", Argonne National Lab., Illinois, United States, Rep. LS-287, Aug. 2000.
- [11] Alex J. Dragt, "Lie algebraic methods for charged particle optics", *AIP Conference Proceedings*, vol. 177, pp. 261-264, 1998. doi:10.1063/1.37819
- [12] M. Berz, "Differential Algebra, A New Tool", in *13th IEEE Particle Accelerator Conference*, Chicago, IL, USA, 20 - 23 Mar 1989, pp. 1419.
- [13] E. Forest, M. Berz, and J. Irwin, "Normal form methods for complicated periodic systems : a complete solution using differential algebra and lie operators", in *Proc. Part. Accel.* 24, 1989, pp. 91-107.
- [14] A. Chao, "Lecture Notes on Special Topics in Accelerator Physics", SLAC, Menlo Park, CA, United States, Rep. SLAC-PUB-9574, Nov. 2002.
- [15] A. Chao, "Lie Algebra Techniques for Nonlinear Dynamics", SLAC, Menlo Park, CA, United States, Rep. SLAC-PUB-9574, Nov. 2002.