Chapter 2
The Linear Transverse Normal Form: One Degree of Freedom

Abstract I define the normal form on the simplest case: the one degree of freedom (1-d-f) linear symplectic case. The phase advance, the invariants and the lattice functions will be defined here using methods which extend to nonlinear systems.

Keywords One-degree-of-freedom · Phasors · Harmonic · Courant-Snyder invariant · Averages · de Moivre's formula

2.1 Conversion Table Between Linear and Nonlinear

Some rules about our story

The theory presented here is easily extended to nonlinear maps. Many statements about linear systems remain true if we extend linear notations to their obvious nonlinear equivalents:

$$\mathbf{w} = M\mathbf{z} \quad \longrightarrow \mathbf{w} = M(\mathbf{z}) \tag{2.1}$$

$$M\mathbf{z} = AB\mathbf{z} \quad \longrightarrow M(\mathbf{z}) = A(B(\mathbf{z})) = (A \circ B)(\mathbf{z}) \tag{2.2}$$

$$M = AB \quad \longrightarrow M = A \circ B \tag{2.3}$$

$$w_i = \sum M_{ij} z_j \quad \longrightarrow w_i = M_i(\mathbf{z}) \tag{2.4}$$

$$w = z^\dagger A^\dagger = (A\mathbf{z})^\dagger \quad \longrightarrow w = A(\mathbf{z})^\dagger. \tag{2.5}$$

2.2 Why Phasors and Normal Forms?

How is the normal form computed? At this stage, who cares! The important point is that you have these maps from a reliable black box library.

If you have the one-turn map from a code (say PTC) and if you have a normal form package (say FPP), you can use it the same way you use any numerical recipe

E. Forest, *From Tracking Code to Analysis*,
DOI 10.1007/978-4-431-55803-3_2

package. The first and most important application is the computation of averages and extrema. The numerical value of these objects can only depend on the invariants of the trajectory if they exist. We will look at averages because they are easy to compute in the most general situation. Extrema are only simple in the linear case.

If a particle sits on an n-torus, where n is the number of degrees of freedom, it should be clear that any time average of its iterated trajectory can only depend on the initial condition via the value of the invariants. This is why we compute a normal form, from which we define invariants and lattice functions. So let us first look at the invariant of an arbitrary function.

2.2.1 The Average of an Arbitrary Function: Need for Phasors

Imagine a 1-d-f function F of (z_1, z_2) whose time average we desire:

$$\langle F \rangle = \frac{F + F \circ m + F \circ m \circ m + \cdots + F \circ m^N}{N} \quad N \to \infty. \tag{2.6}$$

Here the map m is the one-turn map at the position where the average is needed. In the general case,[1] this average is nearly impossible to carry out analytically. However, imagine that we have a normal form for m coming from a friendly and documented numerical recipes library:

$$m = a \circ r \circ a^{-1}. \tag{2.7}$$

The map a is some horrible transformation and r is an amplitude dependent rotation. Amplitude dependent rotations, in 1-d-f, have the following form:

$$r \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} \cos \mu(J) & \sin \mu(J) \\ -\sin \mu(J) & \cos \mu(J) \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \quad \text{where } 2J = z_1^2 + z_2^2. \tag{2.8}$$

One notices that Eq. (2.8) is not a linear map because the angle μ is a function of the amplitude J. Using Eq. (2.7), we can rewrite Eq. (2.6):

$$\begin{aligned} \langle F \rangle &= \frac{F \circ a \circ a^{-1} + F \circ a \circ r \circ a^{-1} + \cdots + F \circ \left\{a \circ r \circ a^{-1}\right\}^N}{N} \quad N \to \infty \\ &= \frac{F \circ a + F \circ a \circ r + \cdots + F \circ a \circ r^N}{N} \circ a^{-1} \qquad N \to \infty \\ &= \frac{\overline{F} + \overline{F} \circ r + \cdots + \overline{F} \circ r^N}{N} \circ a^{-1} \qquad N \to \infty\,. \end{aligned} \tag{2.9}$$

[1] By "general case", I really mean general: 20 degrees of freedom, sixth-order, with magnet modulation and parameter dependence, spin, etc…!!! But concentrate on the simplest case always bearing in mind the feasibility of the general case.

Equation (2.9) expresses a simple fact: the average of a function F is gotten from the average of the transformed function $\overline{F} = F \circ a$ under a rotation.

It is easy to average a function that propagates under a rotation if it is expressed in the phasors' basis ("resonance basis"):

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = c \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = \begin{pmatrix} 1/2 & 1/2 \\ -i/2 & i/2 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$$

and

$$\begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = c^{-1} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}. \tag{2.10}$$

We use this phasors' basis to perform the average:

$$\begin{aligned} \overline{F} \circ r^k \circ c &= \underbrace{\overline{F} \circ c}_{F^r} \circ c^{-1} \circ r^k \circ c \\ &= \underbrace{\overline{F} \circ c}_{F^r} \circ \underbrace{\left\{ c^{-1} \circ r \circ c \right\}^k}_{\Lambda}. \end{aligned} \tag{2.11}$$

We then expand F^r in monomials of the phasors:

$$F^r = \sum F^r_{mn} r_1^m r_2^n \tag{2.12}$$

and now apply the diagonalized rotation Λ to Eq. (2.11):

$$F^r \circ \Lambda^k = \sum F^r_{mn} r_1^m r_2^n \exp\left(i\, k \mu\left(J\right)\left(n - m\right)\right). \tag{2.13}$$

In Eq. (2.12) and (2.13), r_1 and r_2 can be viewed as dummy variables for phase space or as the phasor functions $x + ip_x$ and $x - ip_x$ respectively. This depends on one's point of view during a calculation.
The average of F^r is given by the terms of equal powers in n and m:

$$\left\langle F^r \right\rangle = \sum F^r_{nn} r_1^n r_2^n. \tag{2.14}$$

Now I give you a simple linear example:

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \underbrace{\begin{pmatrix} \sqrt{\beta} & 0 \\ -\alpha/\sqrt{\beta} & 1/\sqrt{\beta} \end{pmatrix}}_{=a} \begin{pmatrix} x \\ p \end{pmatrix} \quad \text{and} \quad F = x^2. \tag{2.15}$$

Non-accelerator physicists reading this may simply accept Eq. (2.15) as the transformation which turns the pseudo-Hamiltonian $H = \gamma x^2 + \beta p^2 + 2\alpha xp$ into $K = x^2 + p^2$ if $\gamma = \frac{1+\alpha^2}{\beta}$. This example is obviously an enticing bait for accelerator physicists.

First, I compute $\overline{F}$,

$$\begin{aligned}\overline{F} &= F \circ a = x^2 \circ a = \left(\sqrt{\beta}x\right)^2 \\ &= \beta x^2 \end{aligned} \tag{2.16}$$

and then F^r,

$$\begin{aligned}F^r &= \overline{F} \circ c = \beta x^2 \circ c \\ &= \beta \left(\frac{r_1 + r_2}{2}\right)^2 \\ &= \beta \left\{\frac{r_1^2 + r_2^2 + 2r_1 r_2}{4}\right\}. \end{aligned} \tag{2.17}$$

and now I retain the terms of equal powers for the average:

$$\left\langle F^r \right\rangle = \frac{\beta}{2} r_1 r_2. \tag{2.18}$$

If we travel back to the space of real Floquet variables using c^{-1}:

$$\begin{aligned}\left\langle \overline{F} \right\rangle &= \frac{\beta}{2} r_1 r_2 \circ c^{-1} \\ &= \frac{\beta}{2} \left\{z_1^2 + z_2^2\right\} = \beta J. \end{aligned} \tag{2.19}$$

Surprise! The average of x^2 is the beta function times J, a well-known result in accelerator physics.

Of course, the average of F is gotten from Eq. (2.19):

$$\langle F \rangle = \beta J \circ a^{-1} = \frac{\beta}{2} \underbrace{\left(\gamma x^2 + \beta p^2 + 2\alpha xp\right)}_{\text{Courant}-\text{Snyder Invariant}}. \tag{2.20}$$

It is no huge surprise that the effect of a^{-1} is to express the average in terms of the original variables. Thus, in the linear case, $2J$ becomes the Courant-Snyder invariant.

Any function can be averaged by a normal form package using the following steps:

1. Find the closed orbit
2. Find the map around the closed orbit
3. Transform it into a normal form, i.e., $m = a \circ r \circ a^{-1}$. The angles of this rotation are called the fractional tunes when measured in revolutions.
4. Substitute a in the function f: $f \circ a$. Express $f \circ a$ in the phasors' basis and retain terms of equal powers in $r_{2i-1}r_{2i}$.
5. The average can be expressed in terms of the invariant in the original variables by composing it with $c^{-1} \circ a^{-1}$. In fact the invariant itself in the original variables is simply $r_{2i-1}r_{2i} \circ c^{-1} \circ a^{-1}$ as the example of Eq. (2.20) shows.

Nota Bene: phasors can be viewed as eigenfunctions rather than just a change of variables. This is useful in analytical calculations.

$$\varphi_{j\pm}(\mathbf{z}) = z_{2j-1} \pm i z_{2j} = \sqrt{2J_j} e^{\mp i \Phi_j} \tag{2.21}$$

It is easy to check that

$$\varphi_{j\pm} \circ r = e^{\mp i \mu_j} \varphi_{j\pm}. \tag{2.22}$$

Thus the phasors $\varphi_{j\pm}$ are eigenfunctions of r. It must be pointed out that Eq. (2.22) still holds if the rotation r is nonlinear. The tunes μ_j are functions of the action variables, the radii.

2.2.2 Linear Lattice Functions from de Moivre's Formula

In the linear case, the map is simply a matrix:

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \text{with} \quad ad - bc = 1. \tag{2.23}$$

As usual, I have assumed that the one-turn map has been extracted in some place in the machine as was described briefly in Sect. 1.7.2.

A stable map of the type of Eq. (2.23) will produce an ellipse in phase space. One can easily show that there exists a symplectic matrix A such that:

$$M = ARA^{-1} \tag{2.24}$$

$$R = \begin{pmatrix} \cos(\mu) & \sin(\mu) \\ -\sin(\mu) & \cos(\mu) \end{pmatrix}. \tag{2.25}$$

Since A has a unit determinant in 1-d-f, then we have

$$A^{-1} = \begin{pmatrix} A_{22} & -A_{12} \\ -A_{21} & A_{11} \end{pmatrix}. \tag{2.26}$$

It is possible to use the above equations to rewrite M in terms of the tune μ:

$$M = \cos(\mu)\, I + \sin(\mu) \begin{pmatrix} \underbrace{-A_{11}A_{21} - A_{12}A_{22}}_{\alpha} & \underbrace{A_{11}^2 + A_{12}^2}_{\beta} \\ \underbrace{-A_{21}^2 - A_{22}^2}_{-\gamma} & \underbrace{A_{11}A_{21} + A_{12}A_{22}}_{-\alpha} \end{pmatrix} \tag{2.27}$$

$$\Rightarrow \quad M = \begin{pmatrix} \cos(\mu) + \alpha \sin(\mu) & \beta \sin(\mu) \\ -\gamma \sin(\mu) & \cos(\mu) - \alpha \sin(\mu) \end{pmatrix}. \tag{2.28}$$

Equation (2.28) is the famous de Moivre formula for the 1-d-f symplectic case. The formula

$$1 + \alpha^2 = \beta\gamma \tag{2.29}$$

also follows from the unit determinant of A. The invariant can be easily computed since it is the circle in the normalised variables. One re-expresses it in the original variables.

$$\begin{aligned} r^2(z) &= z_1^2 + z_2^2 \\ &\downarrow \\ \varepsilon(x, p_x) &= r^2\left(z(x, p_x)\right) = \left\{a_1^{-1}(x, p_x)\right\}^2 + \left\{a_2^{-1}(x, p_x)\right\}^2 \\ \varepsilon(x, p_x) &= \underbrace{\left\{A_{21}^2 + A_{22}^2\right\}}_{\gamma} x^2 + 2\underbrace{\left\{-A_{11}A_{21} - A_{12}A_{22}\right\}}_{\alpha} x\, p_x + \underbrace{\left\{A_{11}^2 + A_{12}^2\right\}}_{\beta} p_x^2. \end{aligned} \tag{2.30}$$

Equation (2.30) is known as the Courant-Snyder invariant in accelerator physics. Equation (2.28) can be extended to coupled linear matrices (see Sect. 4.1.2 or reference [1]) but has no known nonlinear equivalents. In total generality, lattice functions can be defined as:

1. The coefficients of the invariants.
2. The coefficients of the polynomials which express averages in terms of invariants.

These two definitions of the lattice functions are totally generalizable to nonlinear problems.

2.2.3 *Lattice Functions as Coefficients of the Invariant*

We can compute the invariants by computing the radius using the new variables because the normalised motion is a rotation:

$$
\begin{aligned}
\varepsilon &= z_1^{new\,2} + z_2^{new\,2} \\
&= \left(A_{11}^{-1} z_1 + A_{12}^{-1} z_2\right)^2 + \left(A_{21}^{-1} z_1 + A_{22}^{-1} z_2\right)^2 \\
&= \left.\begin{array}{c} \underbrace{\left(\frac{1}{\sqrt{\beta}} z_1\right)^2 + \left(\frac{\alpha}{\sqrt{\beta}} z_1 + \sqrt{\beta} z_2\right)^2}_{\text{Courant}-\text{Snyder Choice}} \\ \text{or} \\ \underbrace{\left(-\frac{\alpha}{\sqrt{\beta}} z_1 - \sqrt{\beta} z_2\right)^2 + \left(\frac{1}{\sqrt{\beta}} z_1\right)^2}_{\text{Choice of Eq. (2.48) see Sect. (2.3.1)}} \end{array}\right\} = \underbrace{\frac{1+\alpha^2}{\beta}}_{\gamma \text{ per Eq. (2.29)}} z_1^2 + 2\alpha z_1 z_2 + \beta z_2^2 \\
&= \gamma z_1^2 + 2\alpha z_1 z_2 + \beta z_2^2 .
\end{aligned} \tag{2.31}
$$

2.2.4 *Lattice Functions as Coefficients of the Moments* $\langle z_i z_j \rangle$

Here let us use the matrix A directly and compute the three quadratic moments of the 1-d-f linear theory.

$$
\begin{aligned}
\langle z_1^2 \rangle &= \left\langle \left(A_{11} z_1^{new} + A_{12} z_2^{new}\right)^2 \right\rangle = A_{11}^2 \left\langle z_1^{new\,2} \right\rangle + A_{12}^2 \left\langle z_2^{new\,2} \right\rangle + 2A_{11}A_{12} \left\langle z_1^{new} z_2^{new} \right\rangle \\
&= \underbrace{\left(A_{11}^2 + A_{12}^2\right)}_{\beta} \frac{\langle r^2 \rangle}{2} = \beta \frac{r^2}{2} .
\end{aligned} \tag{2.32}
$$

In Eq. (2.32), if we are dealing with a single particle and performing a time average, then the average can be removed in the final expression. In Eq. (2.32) I use the fact that z_1^{new} and z_2^{new} move on circles and thus the value $z_1^{new\,2} + z_2^{new\,2}$ is constant on a trajectory. Other averages can be computed as well:

$$
\begin{aligned}
\langle z_2^2 \rangle &= \left\langle \left(A_{21} z_1^{new} + A_{22} z_2^{new}\right)^2 \right\rangle = A_{21}^2 \left\langle z_1^{new\,2} \right\rangle + A_{22}^2 \left\langle z_2^{new\,2} \right\rangle + 2A_{21}A_{22} \left\langle z_1^{new} z_2^{new} \right\rangle \\
&= \underbrace{\left(A_{21}^2 + A_{22}^2\right)}_{\gamma} \frac{\langle r^2 \rangle}{2} = \gamma \frac{r^2}{2} .
\end{aligned} \tag{2.33}
$$

and more interestingly,

$$
\begin{aligned}
\langle z_1 z_2 \rangle &= \left\langle \left(A_{11} z_1^{new} + A_{12} z_2^{new}\right) \left(A_{21} z_1^{new} + A_{22} z_2^{new}\right) \right\rangle \\
&= A_{11}A_{21} \left\langle z_1^{new\,2} \right\rangle + A_{12}A_{22} \left\langle z_2^{new\,2} \right\rangle + \left\{A_{11}A_{22} + A_{12}A_{21}\right\} \left\langle z_1^{new} z_2^{new} \right\rangle \\
&= \underbrace{\left(A_{11}A_{21} + A_{12}A_{22}\right)}_{-\alpha} \frac{\langle r^2 \rangle}{2} = -\alpha \frac{r^2}{2}
\end{aligned} \tag{2.34}
$$

We conclude, in the linear case, that the lattice functions can be understood in three different ways

1. the polynomial coefficients of the invariants, Eq. (2.31),
2. the coefficients multiplying the invariants in expressions for the time-averaged moments, Eqs. (2.32), (2.33) and (2.34).
3. the coefficients of $\cos \mu_i$ and $\sin \mu_i$, in the de Moivre parametrisation of the one-turn matrix in Eq. (2.28), $i = 1, n$ in n-d-f. See Sect. 4.1.2 for $n \geq 2$.

The Definitions 1 and 2 are general and extend to nonlinear coupled systems. Definition 3 extends to linear coupled lattice functions. In the coupled case, the matrix multiplying $\cos \mu_i$ is not the identity and contains useful information. In fact, in the non-symplectic case, Definition 3 is the *only* sensible definition of the lattice functions since there are no invariants and no averages but there is a de Moivre representation as can be seen in Sect. 4.1.2.

2.3 The Program one_turn_orbital_map_normal_form_2d

The example lattice ALS in Appendix A will be used without misalignment[2] errors. This insures that the lattice has perfect mid-plane symmetry and thus the maps restricted to the mid-plane (x, p_x) will be symplectic.

2.3.1 Construction of the Matrix A

The matrix A can be constructed using the eigenvectors of the matrix M. However I will use the eigenvector of the transpose of M. Why is that? The answer lies in our ultimate goal: normalising nonlinear maps. Linear maps can be represented as matrices, that is obvious. Nonlinear maps cannot…or can they? Indeed if we defined a new map acting on functions, then nonlinear maps become matrices albeit infinite in size. Truncation of the matrices to a finite size is equivalent to perturbation theory.

[2]Certain errors preserve mid-plane symmetry: rotations around the vertical axis and translations in the mid-plane.

Readers who are expert at Hamiltonian perturbation theory already know that it is all about changing a function, namely the Hamiltonian, which must be transformed into a function of the action only, a generator of rotations. In fact the Hamiltonian itself, which ultimately generates the matrix M in the linear case, also transforms an arbitrary function through Eq. (1.6), $\frac{dg}{dt} = [g, H] + \frac{\partial g}{\partial t}$. Here we have only the "code" and therefore we must find a way to reproduce Eq. (1.6) with finite maps, the *propagata* of the code: no Hamiltonians and no equations of motion.

The idea is to see how a function will change under composition by a map M which may be linear or nonlinear. Consider a linear map M defined by its matrix (notice the abuse of notation):

$$\mathscr{M} f = f \circ M \text{ where } \mathscr{M} f(z) = f(M(z)) \underbrace{\Longrightarrow}_{\text{For linear } M} \mathscr{M} f(z) = f(Mz). \quad (2.35)$$

The "calligraphic" font for a map acting on functions by substitution is due to Dragt.[3] I tend to use it in my own writing. This type of map is representable by Lie operators provided that M is infinitely differentiable. This will become important in the nonlinear case.

Now that we have defined $\mathscr{M}$, it is clear that any invariant of the motion obeys

$$\mathscr{M} \varepsilon = \varepsilon \circ f = \varepsilon \quad (2.36)$$

and thus the study of $\mathscr{M}$ is central to our "Hamiltonian-free" approach, i.e., use only the *propagata* from the code.

On a linear map, we proceed by first looking at the action of $\mathscr{M}$ on a linear function without any constant part. I will now show that the matrix for $\mathscr{M}$ in the space of linear functions is the *transpose* of the usual matrix M. Let us start with an arbitrary function f of phase space in 1-d-f.[4] Such a function can be written as:

$$f(z) = v_1 z_1 + v_2 z_2. \quad (2.37)$$

The map $\mathscr{M}$ acts on f in the usual way of a map transforming functions:

$$\begin{aligned} (\mathscr{M} f)(z) = f(Mz) &= \sum_i v_1 M_{1i} z_i + v_2 M_{2i} z_i \\ &= \left(M^\dagger v\right)_1 z_1 + \left(M^\dagger v\right)_2 z_2. \end{aligned} \quad (2.38)$$

[3] It is also the "pull back" operator of differential forms; not surprisingly the invariance of the Poisson under $\mathscr{M}$ is equivalent to the invariance of the canonical two-form $\sum_{i=1,N} dz_{2i-1} \wedge dz_{2i}$ under the pull back $\mathscr{M}$.

[4] The argument holds in any number of dimensions including odd numbers.

In Eq. (2.37), I can think of the function f as the sum of two projection functions. It then follows that the array v are the components of f in this basis. These components according to Eq. (2.38), change under the action of $\mathscr{M}$ with the transpose of M:

$$\mathscr{M}(v) = M^{\dagger} v. \tag{2.39}$$

Thus the search for the linear eigenfunctions of $\mathscr{M}$ is equivalent to the search of the eigenvectors of $M^{\dagger}$. The reader must keep in mind that this change of perspective, in a linear context, looks like an attempt at complicating an equivalent process, the diagonalisation of M. Please be patient and remember that we are aiming at the full nonlinear map.

Now, let us assume that our favourite numerical recipe package returns the eigenvalues and eigenvectors of the matrix $M^{\dagger}$:

$$M^{\dagger} w = \lambda w \quad \text{where} \quad w = w_r + i\, w_i. \tag{2.40}$$

In Eq. (2.40), w_r is the real part of the eigenvector and w_i is the imaginary part. We can construct the following "normalised" functions

$$\begin{aligned} g_1(z) &= \frac{w_r \cdot z}{\sigma |[w_r \cdot z, w_i \cdot z]|^{1/2}} \\ g_2(z) &= \frac{w_i \cdot z}{|[w_r \cdot z, w_i \cdot z]|^{1/2}} \\ \text{where } \sigma &= \text{sign}\,[w_r \cdot z, w_i \cdot z] \end{aligned} \tag{2.41}$$

where $[f, g]$ refers to the Poisson bracket as defined in Eq. (1.1). Without loss of generality, I can take $\sigma = 1$ since $\sigma = -1$ amounts to the exchange of the eigenvectors and eigenvalues $\{\lambda, w\}$ and $\{\lambda^*, w^*\}$. The eigenfunctions of $\mathscr{M}$ can be easily constructed:

$$\begin{aligned} f_{\pm}(z) &= g_1(z) \pm i g_2(z) \\ \mathscr{M} f_{\pm} &= \exp(\mp i \mu_0)\, f_{\pm}. \end{aligned} \tag{2.42}$$

Consider the following transformation:

$$\begin{aligned} z_1^{new} &= g_1(z) \\ z_2^{new} &= g_2(z). \end{aligned} \tag{2.43}$$

Theorem 2.1 *The variables z_1 and z_2 move on a circle and therefore (g_1, g_2) defines the matrix A^{-1},*

This is trivial to prove. We notice that

$$z_1^{new2} + z_2^{new2} = (f_+ f_-)(z) \tag{2.44}$$

but we can evaluate this function one turn later:

$$z_1^{new2} + z_2^{new2} = (f_+ f_-)(Mz) \tag{2.45}$$

which by the definition of the map $\mathscr{M}$ (Eq. (2.35)) is just

$$\begin{aligned}(f_+ f_-)(Mz) &= (\mathscr{M} f_+ f_-)(z) \\ &= (\{\mathscr{M} f_+\}\{\mathscr{M} f_-\})(z) \\ &= \left(\lambda f_+ \lambda^* f_-\right)(z) = (f_+ f_-)(z) \quad \text{since } \lambda\lambda^* = 1. \end{aligned} \tag{2.46}$$

Thus, the linear map defining z^{new} defines A^{-1} per Eq. (2.24).

Just for fun, I can get an eigenvector using Eq. (2.28) where the matrix is already in terms of "proper" linear lattice functions. For example, I get the following result:

$$w = \begin{pmatrix} \frac{i-\alpha}{\sqrt{\beta}} \\ -\sqrt{\beta} \end{pmatrix} \quad \text{and} \quad M^\dagger w = e^{-i\mu} w \tag{2.47}$$

where in Eq. (2.47) I have already "normalised" the eigenvector to ensure a unit Poisson bracket. Using Eq. (2.41), I construct the new variables and get A^{-1}:

$$\begin{aligned} z_1^{new} &= g_1(z) = -\frac{\alpha}{\sqrt{\beta}} z_1 - \sqrt{\beta} z_2 \\ z_2^{new} &= g_2(z) = \frac{1}{\sqrt{\beta}} z_1. \end{aligned} \tag{2.48}$$

Thus the matrix[5] for A and A^{-1} are:

$$A = \begin{pmatrix} 0 & \sqrt{\beta} \\ -\frac{1}{\sqrt{\beta}} & -\frac{\alpha}{\sqrt{\beta}} \end{pmatrix} \quad \text{and} \quad A^{-1} = \begin{pmatrix} -\frac{\alpha}{\sqrt{\beta}} & -\sqrt{\beta} \\ \frac{1}{\sqrt{\beta}} & 0 \end{pmatrix}. \tag{2.49}$$

Theorem 2.2 *In an n-d-f symplectic system, where $n \geq 2$, if all the tunes are on the unit circle and distinct (harmonic case), then the construction of* Eq. (2.41) *guaranties that the Poisson bracket of variables belonging to different planes is zero:*

$$\text{if } i \neq j \text{ then } \left[z_{2i-1}^{new}, z_{2j-1}^{new}\right] = \left[z_{2i-1}^{new}, z_{2j}^{new}\right] = \left[z_{2i}^{new}, z_{2j}^{new}\right] = 0. \tag{2.50}$$

[5]Note to accelerator physicists: this is not the usual Courant-Snyder transformation. See the next "grey box."

The accelerator physics readers may be puzzled by Eq. (2.49): this is not the so-called Courant-Snyder transformation. I wanted to emphasize that a normal form package will give you eigenvectors based on some scheme which pays no respect to any of our cultural biases. Indeed accelerator physicists will find it necessary to "canonise" the transformation of Eq. (2.49) for reasons explained later in Sect. 7.4. This amounts to a phase in the definition of the eigenvectors. In the nonlinear case, an arbitrary amplitude dependent rotation can be added in the definition of A. Thus we have

$$\underbrace{\begin{pmatrix} \sqrt{\beta} & 0 \\ -\frac{\alpha}{\sqrt{\beta}} & \frac{1}{\sqrt{\beta}} \end{pmatrix}}_{\text{Courant}-\text{Snyder Choice}} = \begin{pmatrix} 0 & \sqrt{\beta} \\ -\frac{1}{\sqrt{\beta}} & -\frac{\alpha}{\sqrt{\beta}} \end{pmatrix} \underbrace{\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}}_{\text{rotation of } -\pi/2} . \tag{2.51}$$

Other readers, astronomers for example, may wonder what this is all about: why do we care about the canonical transformation? If it makes circles, then what is the big deal? One first notices that the particular phase advance derived from Courant-Snyder, even in the 2-d-f coupled case, has a special empirical meaning which is discussed in Sect. 7.4. Moreover, since it modifies position (i.e. z_1) minimally, it is ideally suited for analytical calculations since most perturbations in accelerator physics involve an electro-magnetic potential A_z which depends only on position.

The departure from the Courant-Snyder choice was rather trivial here: a difference of $\frac{-\pi}{2}$! But it is also possible to choose A with a non-trivial difference:

$$A = \begin{pmatrix} 1/\sqrt{\gamma} & -\alpha/\sqrt{\gamma} \\ 0 & \sqrt{\gamma} \end{pmatrix}, \tag{2.52}$$

in which case many properties associated with the phase advance would have to be rewritten since the difference between the Courant-Snyder choice and Eq. (2.52) is a rotation which depends on position, on α to be precise. I like to refer, somewhat facetiously, to the choice of Eq. (2.52) as the anti-Courant-Snyder choice.

2.3.2 *The Computation of the Map*

The computation proceeds following the explanation in Sect. 1.7.2.

```
state=only_2d0

map_order=2
call init_all(state,map_order,0)

call alloc(one_turn_map,id)
call alloc(y)
call alloc(normal_form)
call alloc(e,r2,z1,z2)

call find_orbit_x(als,closed_orbit(1:6),STATE,1.e-5_dp,fibre1=1)     ! (1)

id=1   ! map is set to identity                                      ! (2)

! map is added to closed orbit and put into the 6 polymorphs
y(1:6)=closed_orbit(1:6)+id                                          ! (3)

call propagate(als,y(1:6),state,fibre1=1)                            ! (4)

one_turn_map=y(1:6) ! Six polymorphs are promoted to Taylor maps     ! (5)
closed_orbit=y                                                       ! (6)

one_turn_map=one_turn_map.sub.1                                      ! (7a)
call print(one_turn_map,6,prec)                                      ! (7b)

call  c_normal(one_turn_map,normal_form)                             ! (8)
```

The only remarkable feature of this piece of code is the `state=only_2d0`. Although PTC is a serious code, this lines allows the user to forget the dimensions of phase space besides the horizontal plane: $(z_1, z_2) = (x, p_x)$.

For the benefit of all readers, I point out again that for "ideal planar rings," the restriction to the horizontal plane is a *bona fide* nonlinear symplectic map. The literature is full of introductory treatments that concentrate heavily on the horizontal plane (see [2–4]). In describing my techniques, particularly their relationship to the code, I could start with a general discussion. But I follow here the gradual approach. The reader will notice that the software tools and the approach will **not** change as we move towards a more general situation: coupled, nonlinear and with spin.

The particular analysis program I wrote, FPP, requires us to set the maximum degree needed at the onset. Therefore our quest for quadratic invariants of linear maps forces us to set the order of the Taylor series to 2. Later, in line (7a), I truncate the map to first-order so as to retain solely the matrix part. The result is printed by the program at line (7b): it is properly truncated to first-order.

2.3.3 Numerical Computation of the Canonical Transformation

Now we come to some new features: a normal form is performed. On line (8), the map `one_turn_map` is normalised. The results are stored in an object called `normal_form`. In particular the transformation a of Eq. (2.7) is stored in `normal_form%a_t`. The routine `c_normal` is very general. In particular it can and will perform a nonlinear normal form if demanded by the user. This is discussed in 1-d-f in Chap. 3. The result of this routine, made to match Eq. (2.49), is printed by line (8d):

```
a=normal_form%a_t
write(6,*)"";write(6,*) " Canonical transformation A";
do i=1,c_%nd2
 write(6,'(a5,i1,a5,6(1x,g12.5))') "row",i,"-->",a(i,1:c_%nd2)  ! (8d)
enddo
```

The output is just

```
   Canonical transformation A
 row 1 -->    0.0000       3.3404
 row 2 -->  -0.29936      0.13421E-02
```

This transformation was selected to match the case of Eq. (2.49). To get it, I expressed the normalised map of Eq. (2.7) in phasors' basis using Eq. (2.10):

$$\begin{aligned} m &= a \circ r \circ a^{-1} \\ &= a \circ c \circ \Lambda \circ c^{-1} \circ a^{-1} \text{ where } \Lambda = \begin{pmatrix} e^{i\mu} & 0 \\ 0 & e^{-i\mu} \end{pmatrix} \\ &\Downarrow \\ m^\dagger &= \underbrace{a^{-1^\dagger} \circ c^{-1^\dagger}}_{ac} \circ \Lambda \circ c^\dagger \circ a^\dagger . \end{aligned} \tag{2.53}$$

The eigenvector of Eq. (2.47) should be the column vector of the matrix $ac = a^{-1^\dagger} \circ c^{-1}$. In the program of Appendix C, this is checked by the lines

```
write(6,*) "";write(6,*) "w_1 =",(i_-alpha)/sqrt(beta)        ! (12a)
              write(6,*) "w_2 =",-sqrt(beta)                  ! (12b)


ac=from_phasor(-1) * normal_form%a_t**(-1)                    ! (13a)
ac=transpose(ac)                                              ! (13b)

write(6,*) "";write(6,*) " Complex Canonical transformation A";write(6,*) "";
write(6,'(a8,17x,a1,18x,a1,1/)') "column","1","2"
do i=1,c_%nd2
 write(6,'(a5,i1,a5,6(1x,g12.5,1x,g12.5))') "row",i,"-->",ac(i,1:c_%nd2)
enddo
```

Lines (12a) and (12b) construct the eigenvector by direct evaluation of Eq. (2.47). Lines (13a) and (13b) is the FPP implementation[6] of Eq. (2.53). The first row of the matrix should be the same as the output of lines (12a, b). The result of this program is:

```
 w_1 = ( 1.34213518775613719E-003, 0.29936426981922848     )
 w_2 =  -3.3404120024205004

Complex Canonical transformation A

column                  1                  2

row 1 -->   0.13421E-02  0.29936      0.13421E-02 -0.29936
row 2 -->   -3.3404       0.0000      -3.3404       0.0000
```

2.4 The Phase Advance and the Invariant

The code must be able to simulate the motion from one point in the machine to another. In a real accelerator we might be content to observe a beam at various beam position monitors (BPM). Most tracking codes should give us access to all positions: the individual integration steps. Therefore it is reasonable to enquire the value of the canonical transformation a at each position s around the ring, a discrete integer in integrators such as PTC. This will lead us to the phase advance which is the code equivalent of the normalised Hamiltonian.

2.4.1 Phase Advance in a Code: Finite Map Theory

Important! The reader should look at Fig. 2.1: it contains mathematics applicable to nonlinear coupled spin-orbit maps.[7] The red dot represents a ray moving in the "real world." The blue dots (pale and dark) show the corresponding ray, transformed by a_s, moving in the world of circles. In the absence of proof to the contrary, the motion in the circle space, i.e., the angle μ_{12}, must depend on the choice of a_s. This is indeed the case in an "s"-dependent system.

The transformations a_1 and a_2 are simply the transformations one obtains by normalising the one-turn map at position $s = 1$ and $s = 2$. These canonical transformations are the output of the normalising software (FPP in my case). Generally they can be chosen arbitrarily, however in the linear case, the Courant-Snyder-Teng-Edwards choice is preferable for experimental and also analytical reasons. I will come back to this topic later (see Sect. 7.4). Now suffices to say that we have a

[6] The line `c_mess_up_vector=.true.; b_mess=-1.0_dp;` at the beginning of the program insures the choice of this section for the matrix A. The reader is invited to comment it out.

[7] As a matter of fact, it even applies to a damped map, i.e., a non-symplectic map such as the map of a classical electron in a radiation source. In that case one must define the normal form to be a "affine dilation," a damped rotation whose complex eigenvalues are slightly under the unit circle.

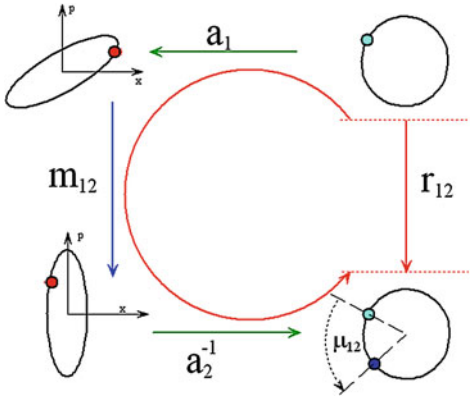

Fig. 2.1 Pictorial view of the phase advance

recipe for the (non)linear calculation of the transformation a_s that depends only on the one-turn map m_s at location s.

It can be shown in the sympletic case, quite generally, that since the normal form is made of commuting rotations, then the map r_{12} on Fig. 2.1 must a rotation.[8] The formula for r_{12} is trivially derived from Fig. 2.1.

$$r_{12} = a_2^{-1} \circ m_{12} \circ a_1 . \tag{2.54}$$

Imagine that in Eq. (2.54), the map can be diagonalised by the same transformation: $a_2 = a_1$. In that case we say that the positions 1 and 2 are matched. They have the same invariants, i.e., the same lattice functions. Moreover, it is possible to prove that the map r_{12} does not depend on the method we used to diagonalise the one-turn map provided that this method depends only on the one-turn map alone and not on extraneous factors. Incidentally this is not the case of Hamiltonian perturbation theory since the map a_s depends on the Hamiltonian, local derivatives of the map with respect to "s". See Chap. 8.

Equation (2.54) defines the phase advance μ_{12} associated to the canonical transformation a_s. This definition works in a code and it works perfectly well in analytical perturbation theory. As I pointed out previously, this is a mute point in astronomy because the normalisation of the Hamiltonian (rather than a map) immediately produces the phase advance, nothing more than the normalised Hamiltonian.

[8] This statement applies to spin as well since the normal form is a rotation around a fixed axis, say the y-axis, for all trajectories in phase space. This set of rotations forms a commutative subgroup of SO(3).

It is instructive, from a tracking code point of view, to rewrite Eq. (2.54):

$$a_2 \circ r_{12} = \underbrace{m_{12} \circ a_1}_{b_2} \Rightarrow a_2 = b_2 \circ r_{12}^{-1}. \tag{2.55}$$

The map b_2 has three remarkable properties:

1. First, it diagonalise the map at position $s = 2$ just like the map a_2. Therefore the invariant ε_2 can be computed using b_2:

$$\varepsilon_2 = r^2 \circ a_2^{-1} = r^2 \circ b_2^{-1}. \tag{2.56}$$

2. Second, b_2 can be tracked by the code using a_1 as the input. Therefore it does not require any normal form calculation.
3. Third, the difference between a_2 and b_2 is the phase advance r_{12}.

Property 1 simply comes from the invariance of a radius under the rotation r_{12}:

$$r^2 \circ b_2^{-1} = r^2 \circ r_{12}^{-1} \circ a_2^{-1} = r^2 \circ a_2^{-1} = \varepsilon_2. \tag{2.57}$$

Proposition 2 is crucial. The canonical transformation a_s can be tracked. In Eq. (2.55), the map for b_2, is made of two parts m_{12} and a_1. If this was the map for a part of the ring, one would conclude that the particle travels first through a segment described by the map a_1 and then through a segment described by m_{12}. Thus if the code that computes m_{12} is initialised by a_1 rather than by the identity, then b_2 will be the output of the propagator. In Sect. 1.7.2, for example, line (3) of the PTC code is replaced by

```
y(1:6)=closed_orbit(1:6)+a_1          ! (3)
```

Proposition 3 gives the most efficient wait to compute the phase advance: one needs only to rotate the map b_2 into the chosen form for the canonical transformation a_2. This is really different from Hamiltonian perturbation theory where, by the very definition of the normalisation process, the phase advance is the transformed Hamiltonian. In accelerator physics, we have only the code, we do **not** have this transformed Hamiltonian: we have the information at chosen surfaces of section.

Let us look for the time being at a piece of code which produces all the information of items 1, 2 and 3.

2.4.2 *Numerical Computation of the Invariant and the Phase Advance*

The program in Appendix D, `one_turn_orbital_map_normal_form_2d.f90`, computes the phase advance for the two simple choices of canonical

transformation shown of Sect. 2.2.3, namely the Eqs. (2.51) and (2.52) which I rewrite here for convenience:

$$\underbrace{\begin{pmatrix} 1/\sqrt{\gamma} & -\alpha/\sqrt{\gamma} \\ 0 & \sqrt{\gamma} \end{pmatrix}}_{Anti-Courant-Snyder} \quad \text{and} \quad \underbrace{\begin{pmatrix} \sqrt{\beta} & 0 \\ -\alpha/\sqrt{\beta} & 1/\sqrt{\beta} \end{pmatrix}}_{Courant-Snyder}. \tag{2.58}$$

The program asks the user to choose between these two definitions. For example I can select between the two transformations by typing either `t` or `f`:

```
Write't' for Courant-Snyder
Write'f' for Anti-Courant-Snyder
t
```

In the example code, the phase advance through the first drift is computed. Thus the map m_{01} is that of a drift. The matrix for a drift of length L is just:

$$m_{01} = \begin{pmatrix} 1 & L \\ 0 & 1 \end{pmatrix}. \tag{2.59}$$

Let us start with the Courant-Snyder definition and the second part of Eq. (2.55) to find the phase advance:

$$a_2 = \underbrace{\begin{pmatrix} 1 & L \\ 0 & 1 \end{pmatrix}}_{m_{12}} \underbrace{\begin{pmatrix} \sqrt{\beta} & 0 \\ -\alpha/\sqrt{\beta} & 1/\sqrt{\beta} \end{pmatrix}}_{a_1} \underbrace{\begin{pmatrix} \cos\left(\mu_{12}^{cs}\right) & -\sin\left(\mu_{12}^{cs}\right) \\ \sin\left(\mu_{12}^{cs}\right) & \cos\left(\mu_{12}^{cs}\right) \end{pmatrix}}_{r_{12}^{-1}} \tag{2.60}$$

which, expanded, is given by:

$$a_2 = \begin{bmatrix} \left(\sqrt{\beta} - \frac{\alpha L}{\sqrt{\beta}}\right)\cos\left(\mu_{12}^{cs}\right) + \frac{L\sin\left(\mu_{12}^{cs}\right)}{\sqrt{\beta}} & -\left(\sqrt{\beta} - \frac{\alpha L}{\sqrt{\beta}}\right)\sin\left(\mu_{12}^{cs}\right) + \frac{L\cos\left(\mu_{12}^{cs}\right)}{\sqrt{\beta}} \\ -\frac{\alpha\cos\left(\mu_{12}^{cs}\right)}{\sqrt{\beta}} + \frac{\sin\left(\mu_{12}^{cs}\right)}{\sqrt{\beta}} & \frac{\alpha\sin\left(\mu_{12}^{cs}\right)}{\sqrt{\beta}} + \frac{\cos\left(\mu_{12}^{cs}\right)}{\sqrt{\beta}} \end{bmatrix}. \tag{2.61}$$

The Courant-Snyder choice forces us to select the phase advance so that the A_{12} part of the matrix is zero. Therefore the equation for the phase advance is:

$$A_{12} = -\left(\sqrt{\beta} - \frac{\alpha L}{\sqrt{\beta}}\right)\sin\left(\mu_{12}^{cs}\right) + \frac{L\cos\left(\mu_{12}^{cs}\right)}{\sqrt{\beta}} = 0 \tag{2.62}$$

from which we deduce that

$$\mu_{12}^{cs} = \tan^{-1}\left(\frac{L}{\beta - L\alpha}\right). \tag{2.63}$$

The same thing can be done with the "anti-Courant-Snyder" definition:

$$a_2 = \underbrace{\begin{pmatrix} 1 & L \\ 0 & 1 \end{pmatrix}}_{m_{12}} \underbrace{\begin{pmatrix} 1/\sqrt{\gamma} & -\alpha/\sqrt{\gamma} \\ 0 & \sqrt{\gamma} \end{pmatrix}}_{a_1} \underbrace{\begin{pmatrix} \cos\left(\mu_{12}^{acs}\right) & -\sin\left(\mu_{12}^{acs}\right) \\ \sin\left(\mu_{12}^{acs}\right) & \cos\left(\mu_{12}^{acs}\right) \end{pmatrix}}_{r_{12}^{-1}} \tag{2.64}$$

which expanded out is just

$$a_2 = \begin{bmatrix} \frac{\cos\left(\mu_{12}^{acs}\right)}{\sqrt{\gamma}} + \left(-\frac{\alpha}{\sqrt{\gamma}} + L\sqrt{\gamma}\right)\sin\left(\mu_{12}^{acs}\right) & -\frac{\sin\left(\mu_{12}^{acs}\right)}{\sqrt{\gamma}} + \left(-\frac{\alpha}{\sqrt{\gamma}} + L\sqrt{\gamma}\right)\cos\left(\mu_{12}^{acs}\right) \\ \sqrt{\gamma}\sin\left(\mu_{12}^{acs}\right) & \sqrt{\gamma}\cos\left(\mu_{12}^{acs}\right) \end{bmatrix}. \tag{2.65}$$

The anti-Courant-Snyder choice forces us to select the phase advance so that the A_{21} part of the matrix is zero. Therefore the equation for this phase advance is:

$$A_{21} = 0 \;\Rightarrow\; \sqrt{\gamma}\sin\left(\mu_{12}^{acs}\right) = 0 \;\Longrightarrow\; \mu_{12}^{acs} = 0. \tag{2.66}$$

We look now at the actual code concentrating first on the one-turn map at $s = 1$

```
z1=1.e0_dp.cmono.'10'                                                    ! (1a)
z2=1.e0_dp.cmono.'01'                                                    ! (1b)
r2=z1**2+z2**2                                                           ! (1c)

call find_orbit_x(als,closed_orbit(1:6),STATE,1.e-5_dp,fibre1=1)         ! (2)

id=1   ! map is set to identity                                          ! (3)

! map is added to closed orbit and put into the 6 polymorphs
y(1:6)=closed_orbit(1:6)+id                                              ! (4)

call propagate(als,y(1:6),state,fibre1=1)                                ! (5)

one_turn_map=y(1:6) ! Six polymorphs are promoted to Taylor maps         ! (6)
closed_orbit=y                                                           ! (7)

call  c_normal(one_turn_map,normal_form)                                 ! (8a)

write(6,'(1/,a50,1/)')"Canonical Transformation coming from Normal Form"
call print(normal_form%a_t,6)                                            ! (8b)

call c_canonise(normal_form%a_t,a_1,phase=phase);phase(1)=0.0_dp;        ! (9a)

if(courant_snyder_teng_edwards) then
 write(6,'(1/,a50,1/)') "Courant-Snyder Canonical Transformation "
  else
 write(6,'(1/,a50,1/)') "Anti-Courant-Snyder Canonical Transformation "
endif
call print(a_1,6,prec)                                                   ! (9b)
```

Line (5) is the usual propagation that gives the one-turn map. Lines (8a,b) is the result of the normal form: it provides a certain form for a_s "randomly" chosen by the

programmer of the normal form routine. It is printed by (9a). In this case, the result is neither the Courant-Snyder nor this bizarre anti-Courant-Snyder transformation but something else. A normal form, like a diagonalisation routine for eigenvalues and eigenvectors, makes no special[9] guaranty: it diagonalises with a transformation of its own.

The next step consist in invoking the routine "`c_canonise`." We owe the word "canon" to the Church, it refers to a set of rules[10] and regulations imposed by the Church as in "Canon Law." Here it also refers to a certain set of rules imposed by accelerator physicists on the transformation a_s. They are somewhat justified but not truly indispensable. The routine `c_canonise` imposes either (2.62) or (2.66) and additional conditions in the nonlinear case. Let us look at the code.

```
y(1:6)=closed_orbit(1:6)+a_1                                          ! (10)

call propagate(als,y(1:6),state,fibre1=1,fibre2=pos)                  ! (11)

b_2=y(1:6)                                                            ! (12a)
call c_canonise(b_2,a_2,phase=phase)                                  ! (12b)
!!!!!!!!!!!!!!!!!!!!!!   One turn map at pos = 2 !!!!!!!!!!!!!!!!!!!!
call find_orbit_x(als,closed_orbit(1:6),STATE,1.e-5_dp,fibre1=pos)    ! (13)

y(1:6)=closed_orbit(1:6)+id                                           ! (14)

call propagate(als,y(1:6),state,fibre1=pos)                           ! (15)

one_turn_map=y(1:6)                                                   ! (16)

call  c_normal(one_turn_map,normal_form)                              ! (17)
call  c_canonise(normal_form%a_t,a_2);                                ! (18)

!!!!!!!!!!!!!!!!!!!   Computes invariants at 1 and 2 !!!!!!!!!!!!!!!!!

e1 = r2*a_1**(-1)                                                     ! (19a)
e2t = r2*b_2**(-1)                                                    ! (19b)
e2 = r2*a_2**(-1)                                                     ! (19c)

write(6,'(1/,a54,1/)')"Invariant at position = 1 computed from one turn map"
call print(e1,6)                                                      ! (20a)
write(6,'(1/,a54,1/)') "Invariant at position = 2  tracked from position = 1"
call print(e2t,6)                                                     ! (20b)
write(6,'(1/,a54,1/)') "Invariant at position = 2  computed from one turn map"
call print(e2,6)                                                      ! (20c)



write(6,'(1/,a54,1/)') "Phase advance from position = 1 to position = 2     "
!!!!!!!!!!!!!!!!!!!!!!!!!!!!   Analytic results !!!!!!!!!!!!!!!!!!!!!
```

[9] My normalisation algorithm, in the absence of parameter dependence, accidentally picks up A to be Courant-Snyder. So I have some parameters, namely `c_mess_up_vector=.true.; b_mess=-1.0_dp;` to insure a real messed up choice for pedagogical reasons. You are invited to comment that out if you want.

[10] We also owe the present usage of the word "propaganda" to the Church: the necessary dogma that must be propagated!

```
if(pos==2.and.courant_snyder_teng_edwards) then
 L=2.832695d0 ! Length of first element L1
 beta=e1.sub.'02'
 alpha=(e1.sub.'11')/2

 write(6,"(a48,G20.13)") "Based on theory, the phase advance should be =", &
                                                    atan(L/(beta-L*alpha))/twopi
else
 write(6,*) "Based on theory, the phase advance should be zero"
endif
!!!!!!!!!!!!!!!!!!!!!!!! The code's results !!!!!!!!!!!!!!!!!!!!!!!!

call print(phase(1),6,prec)                                          ! (21)
```

Line (10), line (11) and (12a, b) correspond to Eq. (2.55). The routine `c_canonise` returns a polynomial for the angle of the phase advance in units of revolutions which is customary in accelerator physics. This is printed on line (21) and compared to the analytical results of Eq. (2.63). The agreement in this case is perfect because the ring of this example is "ideal." The reader must accept that the result of line (12b) would remain correct even if nonlinearities and coupling were present. This is the virtue of a code with self-consistent tools.

Lines (19a, b, c) serve two purposes. First of all, line (19a) gives us the invariant at position 1. From this we extract the lattice functions "beta" and "alpha" needed for the analytical formula of Eq. (2.63). Lines (19b, c) are there to reassure us that the invariant at position 2 can be computed from the tracked transformation b_2. Here is the output in abbreviated form:

```
 Invariant at position = 1 computed from one turn map


 Properties, NO =    2, NV =    2, INA =  148
 ********************************************

   2  0.8962076737126187E-01   0.000000000000000       2  0
   2 -0.8966568980107243E-02   0.000000000000000       1  1
   2   11.15835234591495       0.000000000000000       0  2

 Invariant at position = 2  tracked from position = 1


 Properties, NO =    2, NV =    2, INA =  147
 ********************************************

   2  0.8962076737126187E-01   0.000000000000000       2  0
   2 -0.5167031682375807       0.000000000000000       1  1
   2   11.90288336404888       0.000000000000000       0  2

 Invariant at position = 2  computed from one turn map


 Properties, NO =    2, NV =    2, INA =  141
 ********************************************

   2  0.8962076737126205E-01   0.000000000000000       2  0
   2 -0.5167031682375732       0.000000000000000       1  1
   2   11.90288336404884       0.000000000000000       0  2
```

Finally, the phase advance is printed. If we select the Courant-Snyder phase advance then the result is:

```
Phase advance from position = 1 to position = 2

Based on theory, the phase advance should be =  0.3952456186311E-01

Properties, NO =    2, NV =    2, INA =   62
*******************************************

   0  0.3952456186310726E-01   0.000000000000000       0  0
```

More bizarrely, at least for accelerator physicists, the phase advance for the "anti-Courant-Snyder" choice of Eq. (2.66) is indeed zero!

```
Phase advance from position = 1 to position = 2

 Based on theory, the phase advance should be zero

Properties, NO =    2, NV =    2, INA =   62
*******************************************

 Complex Polynomial is zero
```

2.4.3 *Computation of the Phase: Some Theory*

The code, equipped with TPSA, is always right as I pompously proclaimed in Sect. 1.3. Therefore the call to `c_canonise` in Sect. 2.4.1 works in all cases. Nevertheless it is very instructive, using matrices only, to derive a formula for the phase advance using the most arbitrary 1-d-f Hamiltonian. First of all, I write the most arbitrary quadratic Hamiltonian in 1-d-f:

$$H = \frac{1}{2}\left\{k_{11}z_1^2 + 2k_{12}z_1z_2 + k_{22}z_2^2\right\}. \tag{2.67}$$

The matrix for an infinitesimal propagation due to the Hamiltonian of Eq. (2.67) is

$$m_{ds} = \begin{pmatrix} 1 + k_{12}ds & k_{22}ds \\ -k_{11}ds & 1 - k_{12}ds \end{pmatrix} \tag{2.68}$$

and the matrix for an infinitesimal rotation is

$$r_{ds}^{-1} = \begin{pmatrix} 1 & -d\mu_{ds}^{cs} \\ d\mu_{ds}^{cs} & 1 \end{pmatrix}. \tag{2.69}$$

Then, using the analogue of Eq. (2.60) and imposing $A_{12} = 0$, I get

$$d\mu_{ds}^{cs} = \frac{k_{22}}{\beta} ds. \tag{2.70}$$

And, it can been shown easily by symmetry that

$$d\mu_{ds}^{acs} = \frac{k_{11}}{\gamma} ds. \tag{2.71}$$

In the case of accelerator physics, we have $k_{11} = k(s)$, $k_{12} = 0$ and $k_{22} = 1$ and therefore we get the usual result:

$$d\mu_{ds}^{cs} = \frac{1}{\beta} ds. \tag{2.72}$$

Equation (2.72) is completely consistent with Eq. (2.63) if we simply let L tend towards zero.

2.4.4 *Tracking of the Invariant: Some Hamiltonian Theory*

We can first start with equation Eq. (2.55):

$$\varepsilon_2 = r^2 \circ b_2^{-1} = r^2 \circ a_1^{-1} \circ m_{12}^{-1} = \varepsilon_1 \circ m_{12}^{-1}. \tag{2.73}$$

Equation (2.73) expresses a simple fact: the invariant at position 2 is gotten from the known invariant at position 1 by simply expressing it in terms of the variables at position 2 using the map from 1 to 2. This is completely general inside the code and applies to nonlinear invariants in many degrees of freedom.

However, I can apply this to the general 1-d-f quadratic Hamiltonian of Eq. (2.67) and its related infinitesimal map (Eq. 2.68):

$$\begin{aligned}
\varepsilon_{s+ds} &= \varepsilon_s \circ m_{ds}^{-1} \\
&= \gamma \{(1 - k_{12} ds)\, z_1 - k_{22} ds z_2\}^2 \\
&+ 2\alpha \{(1 - k_{12} ds)\, z_1 - k_{22} ds z_2\} \{-k_{11} ds z_1 + \{1 - k_{12} ds\}\, z_2\} \\
&+ \beta \{-k_{11} ds z_1 + \{1 - k_{12} ds\}\, z_2\}^2.
\end{aligned} \tag{2.74}$$

I then deduce a differential equation for the lattice functions:

$$\begin{aligned}
\frac{d\varepsilon_s}{ds} &= 2(\alpha\, k_{11} - \gamma\, k_{12}) x^2 + 2(\beta k_{11} - \gamma k_{22}) xp \\
&+ 2(\beta k_{12} - \alpha k_{22}) p^2
\end{aligned}$$

$$\Downarrow$$

$$\frac{d\gamma}{ds} = 2(\alpha\, k_{11} - \gamma\, k_{12}) \tag{2.75}$$

$$\frac{d\alpha}{ds} = \beta k_{11} - \gamma k_{22} \tag{2.76}$$

$$\frac{d\beta}{ds} = 2(\beta k_{12} - \alpha k_{22}). \tag{2.77}$$

In accelerator physics, this set of equations reduces to:

$$\begin{aligned} \frac{d\gamma}{ds} &= 2\alpha k_{11} \\ \frac{d\alpha}{ds} &= \beta k_{11} - \gamma \\ \frac{d\beta}{ds} &= -2\alpha. \end{aligned} \tag{2.78}$$

It is interesting to relate the Courant-Snyder phase advance of Eq. (2.70) with the anti-Courant-Snyder one of Eq. (2.71). This can be done using the equation for α (Eq. 2.76):

$$\begin{aligned} \frac{d\mu^{cs}_{ds}}{ds} &= \frac{d\mu^{acs}_{ds}}{ds} - \frac{\frac{d\alpha}{ds}}{\beta\gamma} \\ &= \frac{d\mu^{acs}_{ds}}{ds} - \frac{\frac{d\alpha}{ds}}{1+\alpha^2} \quad \text{(according to Eq. (2.29))}. \end{aligned} \tag{2.79}$$

We can check explicitly the statement near Eq. (2.54) on the case of the two phases used in Eq. (2.79): between matched points, the phase advance does not depend on the method used to normalise the map. To do this we integrate Eq. (2.79) between two position 1 and 2:

$$\mu^{cs}_{12} - \mu^{acs}_{12} = -\int_{\alpha_1}^{\alpha_2} \frac{d\alpha}{1+\alpha^2} = \tan^{-1}(\alpha_1) - \tan^{-1}(\alpha_2)\,. \tag{2.80}$$

Equation (2.80) tells us that if the parameters α are equal, then the phase advances can only differ by an irrelevant multiple of 2π. It is certainly true looking at Eq. (2.80), that if position 1 and 2 are matched, then the phase advances are equal modulo 2π: with two specific choices for a_s the condition is found to be weaker.

References

1. P. Nishikawa, JINST **7** (2012)
2. D.A. Edwards, M.J. Syphers, *An Introduction to the Physics of High Energy Accelerators, Wiley Series in Beam Physics and Accelerator Technology* (Wiley, New York, 1993)
3. E. Wilson, Technical Report No. CERN 84–19, CERN (Unpublished)
4. S.Y. Lee, *Accelerator Physics* (World Scientific Publishing, Singapore, 2004)

http://www.springer.com/978-4-431-55802-6

From Tracking Code to Analysis
Generalised Courant-Snyder Theory for Any Accelerator Model
Forest, E.
2016, XXIII, 347 p. 11 illus. in color., Hardcover
ISBN: 978-4-431-55802-6