Particle Accelerators, 1989, Vol. 24, pp. 91–107
Reprints available directly from the publisher
Photocopying permitted by license only

Printed in the United States of America

NORMAL FORM METHODS FOR COMPLICATED PERIODIC SYSTEMS: A COMPLETE SOLUTION USING DIFFERENTIAL ALGEBRA AND LIE OPERATORS

ETIENNE FOREST

Exploratory Studies Group, Accelerator Fusion Research Division, Lawrence Berkeley Laboratory, Berkeley, CA 94720
and
SSC Central Design Group, Universities Research Association, c/o Lawrence Berkeley Laboratory, Berkeley, CA 94720

MARTIN BERZ AND JOHN IRWIN

SSC Central Design Group, Universities Research Association, c/o Lawrence Berkeley Laboratory, Berkeley, CA 94720

(Received March 14, 1988)

We present two types of formal algorithms: an order-by-order and a "superconvergent" procedure, bringing the one-period map into a so-called normal form, which displays the harmonic content of the map. The algorithm is arbitrary in order, number of parameters, and phase-space dimensions and covers the range of signatures of the unperturbed quadratic invariants found in circular-machine dynamics. The normal form and the map-extraction algorithms have all been implemented using the differential-algebra software. In fact, the work of this paper is feasible in toto because of the differential-algebra theory and software.

1. INTRODUCTION

Serious simulations of large and small circular (or periodic) machines require the use of tracking codes. For example, in the study of the proposed Superconducting Supercollider (SSC), these codes must be able to treat misaligned, mispowered, and misconstructed magnets with substantial random errors. In addition, the tracking codes must be equipped with closed-orbit correction schemes, random and systematic multipole correctors, and even sorting algorithms. At the end, the full horrendous lattice is ready for tracking. In the case of the SSC, we are dealing with a lattice file specifying approximately 10^5 multipoles of various orders and symmetries. But what is the lattice file of a tracking code? In fact, the lattice file specifies the horrible *realistic* Hamiltonian of the SSC.

One would like to find an exact recursive algorithm, implementing canonical transformations, that eliminates nonsecular terms in the motion of particles, a scheme that would work for an arbitrarily complex Hamiltonian, to an order limited only by the power of present computers, and would treat problems in a phase space and a parameter space of arbitrary dimension, where again the limits depend only upon the power of our computers.

Our approach rests upon the following premises:

i) A process that first extracts the formal power-series map for one period from the complex tracking code and then analyses it is more modular and removes the need for a local s-dependent perturbation theory.
ii) The production of the map and the analysis of the map should be independent procedures. In particular, the extraction of the map should not necessarily involve canonical variables.
iii) An adequate and desirable analysis of the final map is best achieved by expressing it in terms of Lie generators.

For *linear effects*, statements i) to iii) have been common in accelerator physics since the early work of Courant and Snyder.[1] Indeed, the Twiss parameters and the fractional tunes can be extracted from the one-turn matrix. Similar quantities are available for coupled systems. It is also true that one propagates the Twiss parameters by matrix (i.e., map) manipulations. Incidently, the Courant-Snyder invariant is just a multiple of the Lie generator of the linear map produced by the matrix! Hence it became clear to one of the authors (E.F.) that for maximum efficiency in the treatment of large and messy periodic systems one should extend the matrix manipulations to nonlinear maps and rewrite the whole of perturbation theory in a Hamiltonian-free context.[2] For the relatively simple Hamiltonians used in the design of circular machine, Forest has used the concatenator and the analysis tools of the program MARYLIE[3] to perform exact extraction and analysis of fifth-order maps in $(x, p_x, y, p_y, \tau, p_\tau)$. (The tracking codes used where TEAPOT and THINTRAC.[4])

The reader will notice that fifth-order and six variables fall short of the goals set forth in this paper. Fortunately, Berz described a tool in the preceding paper of this issue that, in the view of his co-authors, will supersede previous techniques of numerical differentiation implemented on computers. While working in the field of particle spectrometers, he realized the importance of the extraction of formal power series for arbitrarily complex optical systems (or, in other words, extraction of high-order aberrations). Numerical differentiation techniques do not allow for high-order computations because the precision decays rapidly with the order of the aberrations. This led Berz to investigate the differential algebra (DA), an application of non-standard analysis,[5,6] and to produce a DA-software package that allows the user to compute all the partial derivatives of any computed quantity with respect to any number of variables. As discussed in the preceding paper, this implies that one can always produce a formal power-series map of any system, provided a tracking code exists. (Of course, more than one exists!).

In this paper, we will present the tools needed to perform the normal-form algorithm on the map, using the mathematical operations available in the differential algebra software of Berz. All the operations are formal in the sense that we restrict ourselves to truncated power series, i.e. derivatives up to a certain order, and do not discuss questions of convergence.

This combination of extraction of mappings and the analysis software provides a definitive tool for perturbation theory on a complicated periodic system since it is only limited by the power of present computers, not by the lack of human ability and stamina.

A Short Mathematical Survey of Symplectic Maps

In this short survey we will introduce the maps to be studied in the later sections of this paper. Initially, we take some care to emphasize the difference between a function f and the value $f(x)$ the function may take once it is evaluated on a given element x. This is important because the Lie algebraic maps of classical mechanics act on functions of phase space, not on the phase-space points themselves.

Let us assume that we have a $2N$-dimensional phase space. A vector $\mathbf{x}$ in this space will be of the form:

$$\mathbf{x} = (q_1, p_1, \ldots, q_N, p_N), \tag{1a}$$

where $\mathbf{q}$ are the positions and $\mathbf{p}$ the momenta. Being canonical variables, they obey the famous Poisson-bracket condition

$$[q_i, p_j] = \delta_{ij}, \tag{1b}$$

Let us assume that we start with a transformation $\mathcal{G}$ that maps linearly the set $\mathcal{V}$ of functions of phase space and parameter space into itself, i.e. it is an endomorphism of $\mathcal{V}$:

$$f \in \mathcal{V} \xrightarrow{\mathcal{G}} \mathcal{G}f \in \mathcal{V};\ \mathcal{G} \in \text{End}\,(\mathcal{V}), \tag{2a}$$

$$\mathbf{z} \in \mathbb{R}^{2N+N_P} \overset{f}{\mapsto} f(\mathbf{z}) \in \mathbb{R}, \tag{2b}$$

$$\mathbf{z} = (\mathbf{x}, \delta);\ \delta = \text{vector of } N_P \text{ parameters}. \tag{2c}$$

Let us define $2N$ projection functions:

$$\mathbb{R}^{2N+N_P} \xrightarrow{\Pi_i} \mathbb{R};\ \Pi_i \in \mathcal{V}, \qquad \Pi_i(\mathbf{z}) = z_i. \tag{2d}$$

Let us assume that $\mathcal{G}$ leaves the Poisson bracket invariant $[f, g]$, i.e. $[f, g] = [\mathcal{G}f, \mathcal{G}g]$. Then we call $\mathcal{G}$ a symplectic map. For a symplectic map such as $\mathcal{G}$, the following can be shown[7]:

$$(\mathcal{G}f)(\mathbf{z}) = f(\tilde{\mathbf{z}}), \tag{3a}$$

$$\tilde{\mathbf{z}} = ((\mathcal{G}\Pi_1)(\mathbf{z}), \ldots, (\mathcal{G}\Pi_{2N+N_P})(\mathbf{z})) \tag{3b}$$

If $\mathcal{G}$ is assumed to represent the system under study, the function $\mathcal{G}\Pi_i$ ($\in \mathcal{V}$) gives the ith component of the final phase-space vector as a function of the initial phase-space vector $\mathbf{z}$. To simplify the notation throughout this paper, we will purposely confuse a function $f \in \mathcal{V}$ with its functional form $f(\mathbf{z})$, allowing us to replace Π_i by z_i. Consequently, $\tilde{\mathbf{z}}$ will be denoted by $\mathcal{G}\mathbf{z}$ and $\tilde{z}_i$ by $\mathcal{G}z_i$. These objects are functional forms of the functions $\mathcal{G}\Pi_i$. We also consider endomorphisms of $\mathcal{V}$ such as $\mathcal{G}$ because exponentials of Lie operators provide an infinite supply of maps of the type $\mathcal{G}$. As we will see, Lie operators are essential if symplectic maps are to be put into normal form. Clearly, as we just mentioned, if $\mathcal{G}$ is the map for one period of the system, $\mathcal{G}\mathbf{z}$ is the ray after one period. This is exactly what the DA-software gives us to any prespecified order $N_0 - 1$:

$$\mathcal{G}\mathbf{z} = (\mathcal{G}\mathbf{x}, \delta), \tag{4a}$$

$$\mathcal{G}\mathbf{z} = M\mathbf{z} + \sum_{k=2}^{N_0-1} \Gamma_k(\mathbf{z}), \tag{4b}$$

$$M = \text{Matrix or linear part,}$$

$$\Gamma_k(\mathbf{z}) = \text{homogeneous polynomial function of order } k,$$

$$N_0 - 1 = \text{maximum order of the formal power series considered} \tag{4c}$$

Notice that in general it can be arranged that $\mathcal{G}(\mathbf{0}) = \mathbf{0}$. In the next section, two other properties will be assumed.

1) There exists a DA-function $\mathcal{F}\mathbf{z}$ that brings the map $\mathcal{G}$ to its δ-dependent fixed point (closed orbit):

$$\mathcal{G}^f\mathbf{z} = (\mathcal{F}^{-1}\mathbf{z}) \circ (\mathcal{G}\mathbf{z}) \circ (\mathcal{F}\mathbf{z}), \tag{5}$$

or mathematically $\mathcal{G}^f$ has the property

$$\mathcal{G}^f(\mathbf{x}, \delta) = (\mathbf{0}, \delta) + (M_{2N}^f\mathbf{x}, \mathbf{0}) + \sum_{k=2}^{N_0-1} \Gamma_k^f(\mathbf{x}, \delta), \tag{6a}$$

$$\mathcal{G}^f(\mathbf{0}, \delta) = (\mathbf{0}, \delta)\ \forall\delta. \tag{6b}$$

Here $\circ$ is the composition of two functions. It is available in the DA-software package. For the last time we remind the reader that $\mathcal{F}$ belongs to End($\mathcal{V}$) and that $\mathcal{F}\mathbf{z}$ is an abbreviation for the functions $((\mathcal{F}\Pi_1), \ldots, (\mathcal{F}\Pi_{2N+N_p}))$, which are all elements of $\mathcal{V}$.

2) There exists an endomorphism of $\mathcal{V}$ denoted by $\mathcal{A}_L$ that leaves the subspace of linear functions globally invariant and brings M_{2N}^f into a very simple form[8]:

$$\mathcal{M}\mathbf{z} = (\mathcal{A}_L^{-1}\mathbf{z}) \circ (\mathcal{G}^f\mathbf{z}) \circ (\mathcal{A}_L\mathbf{z}), \tag{7a}$$

$$\mathcal{M}\mathbf{z} = (\mathbf{0}, \delta) + R\mathbf{x} + \sum_{k=2}^{N_0-1} \Gamma_k^0(\mathbf{x}, \delta). \tag{7b}$$

In Eq. (7), every (q_i, p_i) undergoes either stable or "unstable" rotation under the effect of R. Using the script letter to represent the symplectic map associated with $R\mathbf{x}$:

$$\mathcal{V} \xrightarrow{\mathcal{R}} \mathcal{V}, \tag{8a}$$

$$\mathcal{R}\begin{pmatrix} q_i \\ p_i \end{pmatrix} = \begin{pmatrix} \cos\mu_i & -\sin\mu_i \\ \sin\mu_i & \cos\mu_i \end{pmatrix}\begin{pmatrix} q_i \\ p_i \end{pmatrix}, \tag{8b}$$

$$= \begin{pmatrix} \cosh\mu_i & -\sinh\mu_i \\ -\sinh\mu_i & \cosh\mu_i \end{pmatrix}\begin{pmatrix} q_i \\ p_i \end{pmatrix}. \tag{8c}$$

The reader can glance at Eqs. (14) for the Lie generator equivalent of Eq. (8). It is worth pointing out that $\mathcal{F}\mathbf{z}$ will exist if all the μ_i's are different from zero. Indeed, in this paper we do not consider integer tunes.

The computation of $\mathcal{F}\mathbf{z}$ and the transformation of the matrix M_{2N}^{f} into R are all operations that have been implemented for an arbitrary phase-space dimension in the DA-software.

Therefore, in the body of this paper, we will deal exclusively with the map $\mathcal{M}$ and its DA-representation $\mathcal{M}\mathbf{z}$. It is on this map that the full combined power of the Lie algebraic methods on the symplectic group and the tools of the differential-algebra package permit a general solution of the normal-form problem.

Before going further, we should state that the reader is assumed to have a strong knowledge of Lie operators in classical mechanics. In particular, we will often go back and forth between functions belonging to $\mathcal{V}$ and their associated Lie operators belonging to $\mathrm{End}(\mathcal{V})$. The notation of Dragt is used throughout this paper:

$$:f\!: g = [f, g] = \nabla f^{\dagger} J \nabla g, \tag{9a}$$

$$J = \text{symplectic form}. \tag{9b}$$

It is worth pointing out that the commutator of two Lie operators $:f:$ and $:g:$ is just the Lie operator of their Poisson bracket $:[f, g]:$. This is a very important homomorphism of the commutor algebra of Lie operators and the Poisson-bracket algebra of functions. It is used implicitly in many manipulations performed throughout this paper.

In Section 2, we introduce and justify the existence of a map denoted by $\mathcal{T}$ critical to the normal-form algorithm on a factored Lie representation of the map. In Section 3, we introduce a new basis for the functions of phase space that uses the linear eigenfunctions of $\mathcal{T}$ (or $\mathcal{R}$). In Section 4, we discuss the evaluation of $(\mathcal{T}^{-1}f_r)(\mathbf{x})$ in terms of the differential-algebra tools.

In Section 5, we extract a Lie generator from a DA-function of the form of (7b), again using DA-tools. This Lie generator is then used in Section 6, where the full recursive algorithm is displayed.

To simplify the notation, the parameter δ is left out of the subsequent discussion until Appendix A. Hence $\mathbf{z}$ is replaced by $\mathbf{x}$. The vector $\mathbf{z}$ can be put back in by mentally letting all operators leave δ unchanged.

Finally, the appendices discuss the "coasting" plane normalization, the superconvergent procedure, and the normalization of the pseudo-Hamiltonian of the map. This last is not a new topic, except for its implementation in the differential algebra context.

2. CANONICAL TRANSFORMATIONS ON SYMPLECTIC ENDOMORPHISMS OF $\mathcal{V}$

The normal-form algorithms of Deprit[9,10,11,12,13] proceed by successive transformations on the Hamiltonian. The infinitesimal Lie generator of the map between

s and $s+ds$ is just $:-H:$. In our case, we do not work on a map near the identity but on a map far from it since it represents one period of our system.

Let us assume that our map $\mathcal{M}$ consists of two pieces, an unperturbed linear part $\mathcal{R}$ and a perturbation proportional to a smallness parameter α. Following Dragt and Finn,[7] it is best to factor the total map into two pieces:

$$\mathcal{M} = \mathcal{R} \exp(:\alpha f:), \tag{10a}$$

$$\mathcal{M} \in \mathrm{End}(\mathcal{V}); \qquad \mathcal{R} \in \mathrm{End}(\mathcal{V}). \tag{10b}$$

Consider a canonical transformation $\mathcal{A}$ whose purpose is to modify $\mathcal{M}$ into a new factorized representation $\mathcal{N}$. Using a Lie representation for $\mathcal{A}$, we get for $\mathcal{N}$

$$\begin{aligned} \mathcal{N} &= \mathcal{A}\mathcal{M}\mathcal{A}^{-1} \\ &= \exp(:\alpha F:)\mathcal{R} \exp(:\alpha f:) \exp(:-\alpha F:), \\ &= \mathcal{R} \exp(:\alpha \mathcal{R}^{-1}\mathcal{F}:) \exp(:\alpha f:) \exp(:-\alpha F:), \\ &= \mathcal{R} \exp\{:\alpha[-(\mathcal{E}-\mathcal{R}^{-1})F + f] + O(\alpha^2):\}, \\ &[\mathcal{E} = \text{identity map} \in \mathrm{End}(\mathcal{V})]. \end{aligned} \tag{11}$$

If we denote by $\mathcal{T}$ the operator $\mathcal{E}-\mathcal{R}^{-1}$, it is clear from Eq. (11) that one must study the range and the kernel of $\mathcal{T}$ to specify what possible linear terms in α can remain in Eq. (11).[14,15] Suppose f is decomposed as follows:

$$f = f_r + f_0; \quad . \ f_r \perp \mathrm{Ker}\,\mathcal{T}. \tag{12}$$

Then, we can select $\mathcal{A}$ or F such that $\mathcal{N}$ becomes $\exp[:\alpha f_0 + O(\alpha^2):]$. The function F is just given by

$$F = \mathcal{T}^{-1} f_r. \tag{13}$$

From this short discussion one sees the central importance of the map $\mathcal{R}$. The eigenvectors of $\mathcal{R}$ of unit eigenvalue will constitute the kernel Ker $\mathcal{T}$ so critical to the inversion of $\mathcal{T}$.

In the next section, we examine a suitable eigenbasis for the study of $\mathcal{R}$ in the general case.

3. THE EIGENFUNCTIONS OF $\mathcal{R}$ AND THE RESONANCE BASIS

As discussed earlier we assume that the linear map $\mathcal{R}$ has been brought to the following Lie representation:

$$\mathcal{R} = \exp(:f_2:), \tag{14a}$$

$$f_2 = \sum_{k=1}^{N} \frac{\mu_k}{2}[q_k^2 + (\epsilon_k - \bar{\epsilon}_k)p_k^2] = \sum_{k=1}^{N} f_2^k, \tag{14b}$$

$$\begin{cases} \epsilon_k = 1,\ \bar{\epsilon}_k = 0 & \text{for stable motion in } k\text{th plane,} \\ \epsilon_k = 0,\ \bar{\epsilon}_k = 1 & \text{for unstable motion in } k\text{th plane.} \end{cases} \tag{14c}$$

In Eq. (14), we purposely neglect the "coasting" plane that is obtained when $\epsilon_k = \bar{\epsilon}_k = 0$. Its inclusion would complicate the discussion, since it is no longer true that the vector space of polynomial functions is a direct sum of the range Im $\mathcal{T}$ and the kernel Ker $\mathcal{T}$ (see Appendix A).[16]

The evaluation of $\mathcal{T}^{-1}f_r$ requires a decomposition of f_r in eigenvectors of $:f_2:$. These eigenvectors are easy to obtain; the answer is given by

$$:f_2^k: h_k^{\pm} = \mp (i\epsilon_k + \bar{\epsilon}_k)\mu_k h_k^{\pm} = \mp \lambda_k h_k^{\pm}, \tag{15a}$$

$$h_k^{\pm} = q_k \pm (i\epsilon_k + \bar{\epsilon}_k)p_k, \tag{15b}$$

$$:f_2^k: = -\frac{\mu_k}{2} h_k^+ h_k^-. \tag{15c}$$

Using this new basis, we can easily find the kernel Ker $\mathcal{T}$. Let us define a new vector as follows:

$$|\mathbf{m}, \mathbf{n}\rangle = (h_1^+)^{m_1}(h_1^-)^{n_1} \cdots (h_N^+)^{m_N}(h_N^-)^{n_N}. \tag{16}$$

Using the differential property of the operator $:f_2:$, we can compute the eigenvalue of $|\mathbf{m}, \mathbf{n}\rangle$[11]:

$$:f_2: |\mathbf{m}, \mathbf{n}\rangle = (\mathbf{n} - \mathbf{m}) \cdot \boldsymbol{\lambda}\, |\mathbf{m}, \mathbf{n}\rangle. \tag{17}$$

Assuming that the λ_ks are irrational and prime amongst themselves, we conclude that

$$|\mathbf{m}, \mathbf{n}\rangle \in \text{Ker}\, \mathcal{T} \Rightarrow \mathbf{n} - \mathbf{m} = \mathbf{0}. \tag{18}$$

Providing that one can easily change basis to the $|\mathbf{m}, \mathbf{n}\rangle$, the computation of $\mathcal{T}^{-1}f_r$ is trivial,[14,15]:

$$f_r = \sum_{\mathbf{m},\mathbf{n}} A_{\mathbf{m},\mathbf{n}} |\mathbf{m}, \mathbf{n}\rangle, \tag{19a}$$

$$\mathcal{T}^{-1}f_r = \sum_{\mathbf{m},\mathbf{n}} \frac{A_{\mathbf{m},\mathbf{n}}}{1 - \exp[(\mathbf{m} - \mathbf{n}) \cdot \lambda]} |\mathbf{m}, \mathbf{n}\rangle. \tag{19b}$$

In the next section, we show how to perform the change of basis and the computation of $\mathcal{T}^{-1}$ using the differential-algebra composition and the special exponent-dependent scalar multiplication.

4. THE EVALUATION OF $\mathcal{T}^{-1}f_r$

The symplectic maps that we study have real Lie generators. However, we have seen that the evaluation of $\mathcal{T}^{-1}f_r$ requires the introduction of complex eigenvectors if one or more planes undergo stable oscillations under the action of $\mathcal{R}$. We will now describe how one uses the composition of two differential-algebra maps to avoid dealing with complex vectors.

First, we introduce the special exponent-dependent scalar multiplication available to the user of the DA-software. Consider a function f of $\mathbb{R}^{2N}$ into $\mathbb{R}$,

$$f = \sum A_{\mathbf{j}} x_1^{j_1} \cdots x_{2N}^{j_{2N}}, \tag{20}$$

and an arbitrary function $g(\mathbf{j})$ from $\mathbb{N}^{2N}$ to $\mathbb{R}$. Using $g(\mathbf{j})$, we built a new function $\phi_g f$ as follows:

$$\mathbb{R}^{2N} \xrightarrow{\phi_g f} \mathbb{R},$$

$$\phi_g f = \sum g(\mathbf{j}) A_{\mathbf{j}} x_1^{j_1} \cdots x_{2N}^{j_{2N}}. \tag{21}$$

The transformation ϕ_g, like the Lie map $\mathcal{M}$, acts on the space of phase-space function of $\mathbb{R}^{2N}$ into $\mathbb{R}$.

We now introduce a transformation ϕ_π by defining $\pi(\mathbf{j})$ to be

$$\begin{cases} M = \sum_{i=1}^{N} \epsilon_i j_{2i} = \text{number of powers of momenta} \\ \qquad\qquad\qquad \text{in the stable planes,} \\ \pi(\mathbf{j}) = (-1)^{M/2} \qquad \text{even } M, \\ \pi(\mathbf{j}) = (-1)^{\frac{M+1}{2}} \qquad \text{odd } M. \end{cases} \tag{22}$$

In the definition of $\pi(\mathbf{j})$, the function ϵ_i is the stable-plane identifier of Eq. (14).

Furthermore, we introduce two functions that will serve in the computation of $\mathcal{T}^{-1}$. Using Eq. (19b), we define the functions $a(\mathbf{j})$ and $b(\mathbf{j})$ to be, respectively, the real and imaginary part of the following function:

$$a + ib = \frac{1}{1 - \exp[(\mathbf{m} - \mathbf{n}) \cdot \lambda]}; \quad \mathbf{j} = (m_1, n_1, \ldots, m_N, n_N). \tag{23}$$

The explicit formulae for the functions a and b are

$$a(\mathbf{j}) = \frac{1 - \cos \Delta e^{\bar{\Delta}}}{1 + e^{2\bar{\Delta}} - 2e^{\bar{\Delta}} \cos \Delta}; \tag{24a}$$

$$b(\mathbf{j}) = \frac{\sin \Delta e^{\bar{\Delta}}}{1 + e^{2\bar{\Delta}} - 2e^{\bar{\Delta}} \cos \Delta}, \tag{24b}$$

$$\begin{cases} \Delta = \sum \epsilon_i (m_i - n_i) \mu_i, \\ \bar{\Delta} = \sum \bar{\epsilon}_i (m_i - n_i) \mu_i. \end{cases} \tag{24c}$$

Clearly, two more exponent-dependent scalar multiplications can be defined, namely ϕ_a and ϕ_b.

To complete the battery of tools necessary for the evaluation of $\mathcal{T}^{-1} f_r$, we will define three real changes of basis using the composition. Consider two functions $\Re$ and $\Im$ belong to $\mathcal{V}$:

$$\mathbb{R}^{2N} \xrightarrow{\Re} \mathbb{R}^{2N},$$

$$\mathbb{R}^{2N} \xrightarrow{\Im} \mathbb{R}^{2N}. \tag{25}$$

One can always define the function $\Im \circ \Re$ from $\mathbb{R}^{2N}$ to $\mathbb{R}^{2N}$ as follows:

$$(\Im \circ \Re)(\mathbf{x}) = \Im[\Re(\mathbf{x})], \qquad \mathbf{x} \in \mathbb{R}^{2N}. \tag{26}$$

The DA-composition allows for operations of the type of Eq. (26) whenever $\mathfrak{R}$ and $\mathfrak{I}$ are polynomial functions. We now define three transformations from $\mathbb{R}^{2N}$ to $\mathbb{R}^{2N}$ by their effect on the basis components q_k and p_k:

$$\begin{aligned} \mathfrak{R}_r(\mathbf{x})_{|2k} &= \mathfrak{R}_r q_k = \frac{q_k + p_k}{2} \qquad \mathbf{x} = (q_1, p_1, \ldots, q_N, p_N), \\ \mathfrak{R}_r(\mathbf{x})_{|2k-1} &= \mathfrak{R}_r p_k = \frac{q_k - p_k}{2}; \end{aligned} \tag{27a}$$

$$\begin{cases} \mathfrak{I} q_k = \bar{\epsilon}_k q_k + \epsilon_k p_k, \\ \mathfrak{I} p_k = \bar{\epsilon}_k p_k + \epsilon_k q_k, \end{cases} \tag{27b}$$

$$\begin{cases} \mathfrak{R}_r^{-1} q_k = q_k + p_k, \\ \mathfrak{R}_r^{-1} p_k = q_k - p_k. \end{cases} \tag{27c}$$

Strictly speaking, the DA-composition computes $\mathfrak{I} \circ \mathfrak{R}$ in a formal sense only since it truncates at a predetermined order. Here, however, since $\mathfrak{R}_r$, $\mathfrak{I}$, and $\mathfrak{R}_r^{-1}$ are linear transformations, the compositions will be performed exactly. We are now in a position to compute $\mathcal{T}^{-1} f_r$ in terms of the operations described in this section.

Before giving the result for $\mathcal{T}^{-1} f_r$ in terms of the real-number transformations listed in this section, let us write the result for $\mathcal{T}^{-1} f_r$ using complex-number transformations. Using Eq. (15b), we define the resonance to the Cartesian transformation $\mathfrak{R}_c$ and its inverse $\mathfrak{R}_c^{-1}$:

$$\begin{aligned} \mathfrak{R}_c q_k &= \frac{q_k + p_k}{2}, \\ \mathcal{R}_c p_k &= \frac{q_k - p_k}{2(i\epsilon_k + \bar{\epsilon}_k)}, \end{aligned} \tag{28a}$$

$$\begin{aligned} \mathfrak{R}_c^{-1} q_k &= q_k + (i\epsilon_k + \bar{\epsilon}_k) p_k, \\ \mathfrak{R}_c^{-1} p_k &= q_k - (i\epsilon_k + \bar{\epsilon}_k) p_k. \end{aligned} \tag{28b}$$

In terms of the transformation $\mathfrak{R}_c$, $\mathcal{T}^{-1} f_r$ is given by

$$\mathcal{T}^{-1} f_r = \phi_{a+ib}(f_r \circ \mathfrak{R}_c) \circ \mathfrak{R}_c^{-1}. \tag{29}$$

Finally, in terms of the real transformation directly available in the DA-software, $\mathcal{T}^{-1} f_r$ can be written as

$$\mathcal{T}^{-1} f_r = \phi_\pi \{ \phi_a[(\phi_\pi f_r) \circ \mathfrak{R}_r] \circ \mathfrak{R}_r^{-1} + \phi_b[(\phi_\pi f_r) \circ \mathfrak{R}_r \circ \mathfrak{I}] \circ \mathfrak{R}_r^{-1} \}. \tag{30}$$

The addition in Eq. (30) is defined in the usual manner:

$$(f + g)(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}). \tag{31}$$

In the next section, we examine how one produces a first-order factorized representation of the type shown in Eq. (10) from a power-series representation of the ray. This will be an essential element of the recursive loop used in the analysis of the one-period map $\mathcal{M}$.

5. THE EXTRACTION OF A FIRST-ORDER LIE REPRESENTATION

Our ultimate goal is to perform a perturbative computation of the power-serie representation of the map to arbitrary order. To achieve this we must be able t extract a Lie generator to first order in some smallness parameter α.

Consider the map $\mathcal{M}$ of Eq. (10), let us assume that we have a form power-series representation of $\mathcal{R}^{-1}\mathcal{M}$ [i.e., $\mathcal{R}^{-1}\mathcal{M}\mathbf{x} = (\mathcal{M}\mathbf{x}) \circ (\mathcal{R}^{-1}\mathbf{x})$]:

$$\mathcal{R}^{-1}\mathcal{M}\mathbf{x} = \mathbf{x} + \alpha \sum_{k=2}^{N_0-1} \mathbf{G}_k(\mathbf{x}; \alpha) = \mathbf{x} + \alpha\mathbf{G}(\mathbf{x}; \alpha). \tag{32}$$

In Eq. (32), the functions $\mathbf{G}_k(\mathbf{x})$ are homogeneous polynomials in $\mathbf{x}$, and $N_0 - 1$ i the highest power in $\mathbf{x}$. Assuming a Lie representation for $\mathcal{R}^{-1}\mathcal{M}$ as done in Eq (10), we can evaluate its effect on $\mathbf{x}$:

$$\mathcal{R}^{-1}\mathcal{M}\mathbf{x} = \exp\left(:\alpha f(\mathbf{x}; \alpha):\right)\mathbf{x} = \mathbf{x} + \alpha[f(\mathbf{x}; 0), \mathbf{x}] + O(\alpha^2)\ldots. \tag{33}$$

Comparing Eqs. (32) and (33), we get the equation for $f(\mathbf{x}; 0)$[7,17]

$$[f(\mathbf{x}; 0), \mathbf{x}] = \mathbf{G}(\mathbf{x}; 0),$$
$$-J\nabla f(\mathbf{x}; 0) = \mathbf{G}(\mathbf{x}; 0),$$
$$\Rightarrow f = \int_0^x J\mathbf{G}(\mathbf{x}'; 0)\, d\mathbf{x}' \text{ (over an arbitrary path).} \tag{34}$$

In the formula for f, the matrix J is the symplectic form used to define the Poisson bracket. The independence of f from the integration path is a consequence of the symplectic nature of $\mathcal{R}^{-1}\mathcal{M}$.[17] To perform this integration, we choose a path along the "diagonal":

$$\mathbf{x}' = \eta\mathbf{x},$$
$$d\mathbf{x}' = d\eta\mathbf{x}. \tag{35}$$

We can write the resulting f with the help of the special exponent-dependent transformation ϕ. We first define a function $\sigma(\mathbf{j})$ that simply counts the powers of $\mathbf{x}$ in each monomial of $\mathbf{G}$:

$$\sigma(\mathbf{j}) = \frac{1}{(\sum_{k=1}^{2N} j_k) + 1}. \tag{36}$$

The integral for f is then given by the DA-software operation:

$$f(\mathbf{x}; 0) = \mathbf{x}^{\dagger} J\phi_\sigma\mathbf{G}(\mathbf{x}; 0) = \sum_{k=1}^{N} \left(x_{2k}(\phi_\sigma\mathbf{G})_{2k-1} - x_{2k-1}(\phi_\sigma\mathbf{G})_{2k}\right). \tag{37}$$

Finally, we must point out that it is possible to compute the DA-function associated with this first-order approximation of the map $\mathcal{R}^{-1}\mathcal{M}$ because of our ability to perform a Poisson bracket on DA-functions:

$$\exp\left(:\alpha f(\mathbf{x}; 0):\right)\mathbf{x} = \mathbf{x} + \sum_{k=2}^{N_0-1} \frac{\alpha^{(k-1)}}{(k-1)!} :f(\mathbf{x}, 0):^{k-1}\mathbf{x}. \tag{38}$$

Needless to say, this function or its inverse ($\alpha \to -\alpha$) can be composed with another DA-function. exactly in a formal-power-series sense thanks to the DA-software. This is how we can bypass the Campbell-Baker-Hausdorff formula.

In Section 6, we will reach our final goal, that is, to write a recursive procedure that brings a map into a normal form.

6. RECURSIVE LOOP FOR THE ORDER-BY-ORDER NORMALIZATION OF THE MAP $\mathcal{M}$

Let us assume that we have a map $\mathcal{M}$ factored into the linear part $\mathcal{R}$ of Eq. (14) and a nonlinear map $\mathcal{N}_0$:

$$\mathcal{M} = \mathcal{R}\mathcal{N}_0, \qquad \mathcal{N}_0\mathbf{x} = \mathbf{x} + \sum_{k=2}^{N_0-1} \mathbf{G}_k^0. \tag{39}$$

Our final goal is to express $\mathcal{M}$ as follows:

$$\mathcal{M} = \mathcal{A}^{-1}\mathcal{R}\mathcal{N}_\Omega\mathcal{A}, \tag{40a}$$

$$\mathcal{A} = \exp(:F_\Omega:)\ldots\exp(:F_\omega:)\ldots\exp(:F_1:), \tag{40b}$$

$$\mathcal{N}_\Omega = \exp(:T_1:)\ldots\exp(:T_\Omega:). \tag{40c}$$

In the order-by-order process, the various Lie polynomials of Eq. (40) are homogeneous in $\mathbf{x}$.

In general, as we will see, the transformations ϕ_a and ϕ_b introduced in Eqs. (23) and (24) will determine the target map $\mathcal{N}_\Omega$.

Sometimes, one would like to leave one or many resonances not belonging to Ker $\mathcal{T}$ in the T_ω's. This is achieved by letting a and b be zero for the particular $\mathbf{j}$ vectors belonging to these resonances.

In one extreme case, we zero the functions a and b for all $\mathbf{j}$ (i.e., $\mathcal{A} = \mathcal{E}$); the result of the recursive loop is the factored Lie algebraic representation of the map $\mathcal{M}$, as prescribed by Dragt and Finn in Ref. 7.

Finally, in the opposite extreme case of total removal of all resonances, all the T_ω's lie in Ker $\mathcal{T}$. They are now the usual shear or detuning terms. To get the map into this particular form, we simply define a and b as follows:

$$\text{if } \sum_{k=1}^{N} |j_{2k} - j_{2k-1}| = 0, \text{ then } a = b = 0. \tag{41}$$

For the other terms, $a(\mathbf{j})$ and $b(\mathbf{j})$ are given by Eq. (24). In this case the map $\mathcal{R}\mathcal{N}_\Omega$ can be written as a single exponent:

$$\mathcal{R}\mathcal{N}_\Omega = \exp\left(:f_2 + \sum_{\omega=1}^{\Omega} T_\omega:\right). \tag{42}$$

In this particular case, we can compute the formal pseudo-Hamiltonian:

$$\mathcal{M} = \exp(:\tilde{H}:),$$
$$\tilde{H} = \mathcal{A}^{-1}\Big(f_2 + \sum_{\omega=1}^{\Omega} T_\omega\Big). \tag{43}$$

Again, we emphasize that all the operations given here are within the power of the DA-theory and *software*.

We now describe the recursive process leading from the factored representation of Eq. (39) to the final result of Eq. (40). This will be achieved in Ω steps, where Ω is just $N_0 - 2$ for an order-by-order perturbative calculation. For the superconvergent process of Appendix C, Ω is just the integer part of $1 + \log(N_0 - 2)/\log 2$. We will see that the nature of the process is partly determined by the function $\sigma(\mathbf{j})$ introduced in Eq. (36).

Let us look at the DA-operations leading from the ωth step to the $(\omega + 1)$th step. To start we assume that the following maps and DA-functions have been found:

$$\mathcal{A}_\omega \mathbf{x} = \exp(:F_\omega:)\mathbf{x} = \mathbf{x} + \mathbf{F}^{\omega}, \tag{44a}$$

$$\mathcal{N}_\omega \mathbf{x} = \mathbf{x} + \sum_{k=2}^{N_0-1} \mathbf{G}_k^{\omega} = \mathbf{x} + \mathbf{G}^{\omega}, \tag{44b}$$

$$\begin{aligned}\mathcal{N}'_\omega \mathbf{x} &= \exp(:-T_\omega:) \cdots \exp(:-T_1:)\mathbf{x},\\ &= \mathbf{x} + \mathbf{T}^{\omega}.\end{aligned} \tag{44c}$$

$$\mathcal{M} = \mathcal{A}_1^{-1} \cdots \mathcal{A}_{\omega-1}^{-1}\mathcal{A}_\omega^{-1}\mathcal{R}\mathcal{N}_\omega\mathcal{A}_\omega\mathcal{A}_{\omega-1} \cdots \mathcal{A}_1 \tag{44d}$$

We first compute $T_{\omega+1}$ and $F_{\omega+1}$:

$$F_{\omega+1} = \mathcal{T}^{-1}\mathbf{x}^\dagger J\phi_\sigma\big((\mathcal{N}_\omega\mathbf{x}) \circ (\mathcal{N}'_\omega\mathbf{x}) - \mathbf{x}\big). \tag{45}$$

For a superconvergent process $\sigma(\mathbf{j})$ is exactly given by Eq. (36). However, for the normal order-by-order process described in this section, $\sigma(\mathbf{j}; \omega)$ is ω dependent:

$$\begin{cases} \sigma(\mathbf{j}; \omega) = \dfrac{1}{\omega + 3} & \text{if } \sum_{k=1}^{2N} j_k = \omega + 2, \\ \sigma(\mathbf{j}; \omega) = 0 & \text{otherwise.} \end{cases} \tag{46}$$

Of course, $T_{\omega+1}$ is just the leftover part that is saved during the evaluation of $\mathcal{T}^{-1}$. More precisely, the terms that are zeroed by ϕ_a and ϕ_b are kept and transformed along with those leading to $F_{\omega+1}$.

To close the recursive loop, we must compute $\mathcal{A}_{\omega+1}\mathbf{x}$. This is, of course, a DA-operation since it involves a finite number of Poisson brackets, namely, at most $(N_0 - 2)/(\omega + 1)$ terms of the Lie exponential map. The same can be said about $\mathcal{A}_{\omega+1}^{-1}\mathbf{x}$. Hence we obtain the DA-map $\mathcal{N}_{\omega+1}$:

$$\mathcal{N}_{\omega+1}\mathbf{x} = (\mathcal{A}_{\omega+1}^{-1}\mathbf{x}) \circ (\mathcal{N}_\omega\mathbf{x}) \circ (\mathcal{R}\mathbf{x}) \circ (\mathcal{A}_{\omega+1}\mathbf{x}) \circ (\mathcal{R}^{-1}\mathbf{x}). \tag{47}$$

By exponentiation of the Poisson bracket or composition, the map $\mathcal{N}'_{\omega+1}$ is also computed:

$$\mathcal{N}'_{\omega+1}\mathbf{x} = (\mathcal{N}'_\omega\mathbf{x}) \circ \big(\exp(:-T_{\omega+1}:)\mathbf{x}\big]. \tag{48}$$

This last operation brings us back to the initial point of Eq. (44) with ω raised o $\omega + i$. The similar process of normalization can be now applied to the)seudo-Hamiltonian $\tilde{H}$. This will involve a few changes in the recursive process ınd a redefinition of the functions a and b. We describe this in Appendix B.

/. CONCLUSION

We have described in this paper the algorithm necessary for "normalizing" a map ısing its DA-representation. This algorithm has been written and tested against he less-powerful tools installed in MARYLIE 5.0 by Neri and Dragt.[3] The ılgorithm itself depends entirely on our ability to partially factor a map in terms)f Lie operators. Its implementation to arbitrary order, arbitrary number of parameters, and arbitrary signature of the quadratic invariants is made possible only by the power of the differential-algebra package of Berz. For example, at present, we have produced a ninth-order map (with six variables) for the realistic lattice of the SSC. For smaller machines, we can produce maps of higher order. This will be of great use in the investigation of undulators and their effects on circular light-source devices.

The maps we produce can be exactly analyzed with our algorithm, and daredevils can use them for tracking. Incidentally, the partial-map inverter of the DA-package, in addition to computing the fixed-point map $\mathcal{F}\mathbf{z}$ mentioned in the introduction, permits *several* "symplectification" schemes for any DA-function used in a tracking simulation.

As is the case with perturbation theory, it is clear that the DA-software can be useful in ways not described or even foreseen by the authors. For example, the symplectic-SU(2) maps needed in Yokoya's algorithm[15] as well as the algorithm itself can be written easily using the DA-software because the coupling from the orbital on the spin motion requires a high-order expansion in the spin directions.

We hope that the reader will derive two conclusions from this paper. First, that formal perturbation theory on the map and a Hamiltonian-free[2] approach to single-particle dynamics in complicated periodic systems are much more powerful than a striaghtforward direct application of the Birkhoff or Deprit normal-form algorithm on the messy Hamiltonian.[13] Second, that the differential-algebra method supersedes all numerical differentiation methods and map manipulators by its generality and precision.

ACKNOWLEDGMENTS

We thank Dr. Alexander Chao of the SSC Central Design Group for permitting Martin Berz and the two other authors to work together on the normal-form problem. In addition, we are grateful to Dr. Filippo Neri and Prof. Alex Dragt of the University of Maryland for providing us with a version of MARYLIE that can analyze stable fifth-order maps. It was used to benchmark our calculations and test our algorithms. The comments of the anonymous "TEX" reviewer were also very helpful. Finally, we would like to express our gratitude to Miss Valerie Kelly for her prompt and skillful typing of the manuscript.

REFERENCES

1. E. D. Courant and H. S. Snyder, *Ann. Phys.* **3,** 1 (1958).
2. E. Forest, SSC Report 111 (1987).
3. A. J. Dragt *et al., IEEE Trans. Nucl. Sci.,* **NS-32,** 2311 (1985).
4. Done to 3rd order in TEAPOT and 5th order in THINTRAC. Later done to 5th order in TEAPOT with the DA-software.
5. A. Robinson, Proc. Royal Acad. Amsterdam Ser. A, **64,** 432 (1961).
6. D. Laugwitz, *J. Ber. Deutsch. Math. Vereinigung* **75,** 66 (1973).
7. A. J. Dragt and J. M. Finn, *J. Math. Phys.* **20,** 2649 (1979).
8. E. Forest, SSC Report 29 (1985).
9. A. Deprit, *Cel. Mech.* **1,** 12 (1969).
10. J. R. Cary, *Phys. Rep.* **79,** 129–159 (1981).
11. A. J. Dragt and J. M. Finn, *J. Math. Phys.* **17,** 2215 (1976).
12. G. Hori, *Pub. Astron. Soc. Jap.* **18,** 287 (1966).
13. L. Michelotti, *Particle Accelerator* **16,** 233 (1985).
14. E. Forest, *Particle Accelerator* **22,** 15 (1987).
15. K. Yokoya, *Nucl. Instr. and Meth.* **A258,** 149 (1987).
16. R. Cushman, A. Deprit, and R. Mosak, *J. Math. Phys.* **24,** 2102 (1983).
17. E. Forest, SSC Report 138 (1987).

APPENDIX A

The Coasting Plane Normalization Procedure

Let us assume that some parts of the f_2 are "driftlike." Then f_2 can be reduced and broken into parts:

$$f = f_{2H} + f_{2D}, \tag{A-1a}$$

$$= \sum_{k=1}^{N_H} -\frac{\mu_k}{2}[q_k^2 + (\epsilon_k - \bar{\epsilon}_k)p_k^2] + \sum_{k=N_{H+1}}^{N_H+N_D} -\frac{p_k^2}{2}. \tag{A-1b}$$

If we further assume that all the p_k's $(k > N_H)$ are constants of the motion of the full nonlinear map, the procedures outlined in the body of this paper follow immediately by simply considering the $p_k(k > N_H)$ as parameters instead of canonical variables. The basis vectors are just obtained from $|\mathbf{m}, \mathbf{n}\rangle$ of Eq. (16):

$$|\mathbf{m}, \mathbf{n}; v\rangle = |\mathbf{m}, \mathbf{n}\rangle \delta_1^{v_1} \cdots \delta_{N_P+N_H}^{v_{N_P}+N_H}, \tag{A-2a}$$

$$\delta = (\delta_1, \ldots, \delta_{N_P}, p_{N_{H+1}}, \ldots, p_{N_H+N_D}). \tag{A-2b}$$

Here δ contains the noncanonical parameters as well as the driftlike momenta. Essentially, the absence of the $q_k(k > N_H)$ from the Lie generators of the map allows for a full diagonalization of $\mathscr{T}$ in terms of the $|\mathbf{m}, \mathbf{n}; \mathbf{v}\rangle$ eigenfunctions.

Physically, the longitudinal phase space is an example of a "coasting" plane that occurs whenever cavities are turned off in a circular ring.

When the p_k's $(k > N_H)$ are not constant, the transformation $\mathcal{T}$ cannot be diagonalized. At best $\mathcal{T}$ can be put in Jordan normal form. For our resonance basis, we will again factor the harmonic part from the "drifting" planes.

$$|\mathbf{m}, \mathbf{n}, \mathbf{k}, \mathbf{l}, \mathbf{v}\rangle = |\mathbf{m}, \mathbf{n}\rangle \, |\mathbf{k}, \mathbf{l}\rangle_D \, |\mathbf{v}\rangle_P, \tag{A-3a}$$

$$|\mathbf{k}, \mathbf{l}\rangle = q_{N_{H+1}}^{k_1} \cdots q_{N_H+N_D}^{k_{N_D}} p_{N_{H+1}}^{k_1} \cdots p_{N_H+N_D}^{k_{N_D}}, \tag{A-3b}$$

$$|\mathbf{v}\rangle_P = \delta_1^{\nu_1} \cdots \delta_{N_P}^{\nu_{N_P}}. \tag{A-3c}$$

We also factor the map $\mathcal{R}$ into the harmonic and the drifting part:

$$\mathcal{R} = \mathcal{R}_H \mathcal{R}_D = \mathcal{R}_D \mathcal{R}_H = \exp(:f_{2H}:) \exp(:f_{2D}:). \tag{A-4}$$

Finally, we rewrite the operator $\mathcal{T}$ as a product:

$$\mathcal{T} = (\mathcal{E} - \mathcal{R}_H^{-1} \mathcal{R}_D^{-1}) = \mathcal{T}_H(\mathcal{E} - \mathcal{R}_H^{-1} \mathcal{T}_H^{-1} \mathcal{D}), \tag{A-5a}$$

$$\mathcal{D} = \mathcal{R}_D^{-1} - \mathcal{E} = \sum_{k=1}^{\infty} \frac{:-f_{2D}:^k}{k!}, \tag{A-5b}$$

$$\mathcal{T}_H = \mathcal{E} - \mathcal{R}_H. \tag{A-5c}$$

Due to the special nature of f_{2D}, the operator $:f_{2D}:$ is nilpotent in a formal power-series sense. In fact, if our maximum order is N_0, we must have

$$\mathcal{T}^{-1} = \mathcal{T}_H^{-1} \sum_{k=0}^{N_0} (\mathcal{R}_H^{-1} \mathcal{T}_H^{-1})^k \mathcal{D}^k. \tag{A-6}$$

The form of $\mathcal{T}^{-1}$ will complicate the process outlined in Section 4. Physically it corresponds to a particle undergoing oscillations in a bucket with no linear part. Because of the rarity of this problem, we have not yet implemented the map $\mathcal{T}$ of Eq. (A-6) in our normal-form software. It should not pose any problem to the interested user.

Mathematically, the harmonic part $\mathcal{R}_H$ has a semisimple Lie generator, and, as a result, the space of polynomial functions is a direct sum of the range and the kernel of $\mathcal{R}_H$. In the coasting case, $\mathcal{R}$ can be factored into a semisimple part $\mathcal{R}_H$ and a nilpotent part $\mathcal{R}_D$ substantially complicating the process.[16]

APPENDIX B

Order-by-Order Normalization of the Pseudo-Hamiltonian $\tilde{H}$

The pseudo-Hamiltonian $\tilde{H}$ can be written as a sum of a quadratic part f_2 given by Eq. (14b) and a nonlinear part αV:

$$\tilde{H} = f_2 + \alpha V. \tag{B-1}$$

Following Dragt and Finn,[11] we transform the map $\exp(:\tilde{H}:)$ by a similarity transformation $\mathcal{A}$, as done in Section 2. One gets the result:

$$\begin{aligned} \mathcal{N} &= \mathcal{A} \exp(:\tilde{H}:) \mathcal{A}^{-1}, \\ &= \exp(:\mathcal{A}\tilde{H}:), \\ &= \exp\left(:f_2: + :-f_2: F + \alpha V + O(\alpha^2) \cdots\right). \end{aligned} \tag{B-2}$$

From Eq. (B-1), we define again a map $\mathcal{T}$:

$$\mathcal{T} = :f_2:. \tag{B-3}$$

Letting $:f_2:^{-1}$ act on $|\mathbf{m}, \mathbf{n}\rangle$ provides the new functions $a(\mathbf{j})$ and $b(\mathbf{j})$ that are to be used to produce the transformations ϕ_a and ϕ_b. This was already provided by Eq. (17):

$$\begin{aligned} :f_2:^{-1} |\mathbf{n} - \mathbf{m}\rangle &= [(\mathbf{n} - \mathbf{m}) \cdot \lambda]^{-1} |\mathbf{m}, \mathbf{n}\rangle, \\ &= \frac{1}{(i\Delta + \bar{\Delta})} |\mathbf{m}, \mathbf{n}\rangle, \\ &= \frac{\bar{\Delta} - i\Delta}{\Delta^2 + \bar{\Delta}^2} |\mathbf{m}, \mathbf{n}\rangle. \end{aligned} \tag{B-4}$$

Consequently, we get for a and b

$$\begin{aligned} a(\mathbf{j}) &= \frac{\bar{\Delta}}{\Delta^2 + \bar{\Delta}^2}. \\ b(\mathbf{j}) &= \frac{-\Delta}{\Delta^2 + \bar{\Delta}^2}. \end{aligned} \tag{B-5}$$

Let us start again the recursive process, as was done in Section 6, by postulating the existence of a few maps:

$$\begin{aligned} \tilde{H}_\omega &= f_2 + V_\omega, \\ \mathcal{A}_\omega \mathbf{x} &= \exp(:F_\omega:)\mathbf{x} = \mathbf{x} + \mathbf{F}^\omega. \end{aligned} \tag{B-6}$$

We first compute $F_{\omega+1}$, again using $\mathcal{T}^{-1}$:

$$F_{\omega+1} = \mathcal{T}^{-1}\mathcal{P}_{\omega+2}V_\omega. \tag{B-7}$$

Here $\mathcal{P}_{\omega+2}$ projects the part of V_ω that is of degree $\omega + 2$.

Then the DA-function $\mathcal{A}_{\omega+1}\mathbf{x}$ is computed with a finite expansion of the Lie exponentiation, as in Section 6. Finally, $\tilde{H}_{\omega+1}$ is simply obtained by composition:

$$\tilde{H}_{\omega+1} = \tilde{H}_\omega \circ (\mathcal{A}_{\omega+1}\mathbf{x}) = f_2 + V_{\omega+1}. \tag{B-8}$$

This final step closes the recursive loop.

Again any resonance can be left in by a suitable modification of $a(\mathbf{j})$ and $b(\mathbf{j})$. One bizarre application of this method in connection with the full DA-software package, is the ability to compute the full $(N_0 - 1)$th-order matrix of any ideal electromagnetic element exactly. This is true because the formal solution of an ideal element can be written as

$$\mathcal{M} = \exp(:f_2 + V:). \tag{B-4}$$

Then one can follow our procedure and get for $\mathcal{M}$:

$$\mathcal{M} = \mathcal{A}^{-1}\exp(:f_2 + V_\Omega:)\mathcal{A}. \tag{B-10}$$

However, since $f_2 + V_\Omega$ is a function of the f_2^k's only [see Eq. (14b)], its motion can be computed exactly in terms of the sine, the cosine, and their hyperbolic counterparts.

Having the exact solution in terms of elementary functions allows us to compute the DA-representation of the map (see preceding paper of this issue). Therefore, we can get $\mathcal{M}\mathbf{x}$:

$$\mathcal{M}\mathbf{x} = (\mathcal{A}\mathbf{x}) \circ [\exp(:f_2 + V_\Omega:)\mathbf{x}] \circ (\mathcal{A}^{-1}\mathbf{x}). \tag{B-11}$$

It is not clear whether this surprising result has practical application in the field of power-series codes (again, see the preceeding paper).

APPENDIX C

The Superconvergent Algorithm on the Map

A superconvergent algorithm can be constructed by restoring the function σ used in the integration to its general form given by Eq. (36) and defining an ω-dependent $\mathcal{T}$ map. To achieve superconvergence in the algorithm of Section 6, we define $\mathcal{T}$ as follows:

$$\mathcal{T}(\omega) = \mathcal{E} - \mathcal{R}^{-1}N'_\omega, \tag{C-1a}$$

$$\mathcal{N}'_\omega = \text{tune-shift map defined by Eq. (44c)}. \tag{C-1b}$$

Because $\mathcal{N}'_\omega - \mathcal{E}$ is nilpotent over the space of truncated polynomials, we can use the trick of Appendix A to invert the map $\mathcal{T}(\omega)$:

$$\mathcal{T}^{-1}(\omega) = \mathcal{T}_0^{-1} \sum_{k=0}^{N_0-3} (\mathcal{R}^{-1}\mathcal{T}_0^{-1})^k D^k_\omega, \tag{C-2a}$$

$$\mathcal{T}_0 = \mathcal{E} - \mathcal{R}^{-1}, \tag{C-2b}$$

$$\mathcal{D}_\omega = \mathcal{N}'_\omega - \mathcal{E}. \tag{C-2c}$$

The superconvergent algorithm is faster than the order-by-order procedure for very-high-order calculations. For a simple case, it was slower by a factor of two at the sixth but about 80 percent faster at the sixteenth order, without any special attempts to optimize the superconvergent algorithm. We leave it to the reader to construct a superconvergent process reproducing the results of Appendix B.