

On Jacobian matrices for flows

B. Doyon^{a)} and L. J. Dubé^{b)}

Département de Physique, de Génie Physique, et d'Optique Université Laval, Cité Universitaire, Québec, Canada G1K 7P4

(Received 27 September 2004; accepted 10 December 2004; published online 10 February 2005)

We present a general method for constructing numerical Jacobian matrices for flows discretized on a Poincaré surface of section. Special attention is given to Hamiltonian flows where the additional constraint of energy conservation is explicitly taken into account. We demonstrate the approach for a conservative dynamical flow and apply the technique for the general detection of periodic orbits. © 2005 American Institute of Physics. [DOI: 10.1063/1.1854031]

It was the genius of Henri Poincaré which brought us a set of brilliant tools for exploring dynamical systems. For studying continuous flows, the surface of section, known as the Poincaré Surface of Section (PSS), is perhaps its most celebrated method. On this surface, the flow is faithfully represented by a discrete map, the Poincaré map. The study of this mapping often reveals more dynamical informations than the trajectories themselves. In most cases however, the map can only be obtained numerically as well as its Jacobian matrix necessary for the characterization of the stability of the underlying trajectories. We show how to obtain accurate Jacobian matrices from general flows with special attention to Hamiltonian systems. We take the opportunity to generalize the procedure to systems where many constraints are present other than the one imposed by the PSS. We are concerned in particular with constraints that effectively decrease the dimensionality of the system. The case of a conservative system is detailed and serves to illuminate the analysis. The technique developed finds a first application in the detection of unstable periodic orbits whose importance transcends nonlinear dynamics, from the reconstruction of invariants to the control of chaos.

I. INTRODUCTION

The choice of a *Poincaré Surface of Section (PSS)*, an arbitrary hypersurface generically not tangent to the vector field, is often the first step in the study of a continuous flow. It is always possible to find an hypersurface of codimension one (i.e., one dimension less than that of the flow) for which all points *on* the hypersurface are projected back to it by integrating the flow for a finite interval of time. The numerical methods necessary to achieve the construction of the ensuing Poincaré map, have a long history¹ and are well described in the literature.^{2,3}

The construction of a discrete map and its associated Jacobian matrix from a flow has some technical aspects that need to be considered. For instance, the explicit construction

of the Jacobian matrix on the PSS for a Hamiltonian system, although an important element among the tools for the analysis of nonlinear systems, seems to have been somewhat neglected in the literature. This paper presents a detailed derivation of the methodology necessary to construct numerical Jacobian matrices, and although the emphasis is on conservative flows, it applies equally well to dissipative systems.

One of the motivations for doing so is related to some technical problems associated with the detection of periodic orbits. Indeed, one of the faithful aspects of the Poincaré mapping are the *periodic orbits*, which play a fundamental role in the description and characterization of the dynamics of chaotic systems. For instance, for hyperbolic attractors, the natural measure, the Lyapunov exponents and the fractal dimension can be written in term of Unstable Periodic Orbits (hereafter UPOs).⁴⁻⁶ For open systems, one can estimate the escape rate from the UPOs.⁷ In Hamiltonians systems, the quantum mechanical density of states can be expressed, in the semiclassical regime, by a series expansion with respect to the stability coefficients of the periodic orbits.⁸ The UPOs therefore constitute the most fundamental building blocks of a chaotic system and this has motivated the research for efficient techniques to detect them. The reader may wish to consult⁹ for further discussion of the importance and utility of periodic orbits.

The *complete* detection of UPOs in a chaotic set is a difficult task. One reason is related to the stability of the orbits: trajectories that enter the neighborhood of the UPOs will be repelled from it. The other reason is related to their quantity: the number of UPOs grows exponentially with increasing period. Novel numerical methods have been proposed recently to detect UPOs in discrete chaotic systems.¹⁰⁻¹² When dealing with a continuous flow, these methods can be implemented using the PSS to construct the discrete map from the flow. The detection of periodic orbits benefits then from the fact that the dimension of phase space on the PSS is smaller than that of the continuous flow. Furthermore, the latest numerical techniques for finding UPOs need *both* the discrete map describing the dynamics and its Jacobian matrix.^{12,13} As we will see however, the power of the methods emerges fully when their constituting elements are accurately evaluated and it is therefore timely to develop reliable methods to calculate them.

^{a)}Author to whom correspondence should be addressed. Electronic mail: bdoyon@phy.ulaval.ca

^{b)}Also at Laboratoire de Chimie-Physique Matière et Rayonnement, Université Pierre et Marie Curie, 75231 Paris 05, France.

The paper is organized as follows: Section II reviews the method for constructing a discrete map from a flow and sets the notation for the rest of the paper. The construction of the Jacobian matrix on the PSS is considered in detail, with special attention to Hamiltonian systems. Section III presents a concrete example of the formulas of Sec. II, through the construction of a discrete map of a particular Hamiltonian flow: the *Diamagnetic Kepler Problem (DKP)*. Section IV applies the construction of the *numerical* discrete map to the detection of UPOs in the DKP. It is shown in particular that inaccurate Jacobian matrices result in an important reduction of the number of detected UPOs. Our conclusions are then summarized in the final section.

II. GENERAL FORMULATION

Our starting point is the N differential equations that describe the dynamical problem of interest

$$\frac{d\tilde{\mathbf{v}}}{dt} = \tilde{\mathbf{f}}(\tilde{\mathbf{v}}), \quad \tilde{\mathbf{v}} \in \mathcal{R}^N \tag{1}$$

or equivalently,

$$\begin{pmatrix} \frac{d\tilde{v}^1}{dt} \\ \vdots \\ \frac{d\tilde{v}^N}{dt} \end{pmatrix} = \begin{pmatrix} \tilde{f}^1(\tilde{\mathbf{v}}) \\ \vdots \\ \tilde{f}^N(\tilde{\mathbf{v}}) \end{pmatrix}.$$

In general, this system of equations is solved numerically and the flow $\tilde{\phi}(\tilde{\mathbf{v}}, t) \equiv \tilde{\mathbf{v}}(t)$ corresponds to the solution of the integration after a time t , starting from the initial condition $\tilde{\mathbf{v}}$. The symbol $\tilde{}$ is reserved for all N -component vectors. Our goal is to obtain numerically a mapping $\mathbf{v}_n \mapsto \mathbf{v}_{n+1} \equiv \mathbf{F}(\mathbf{v}_n)$ at discrete times τ_n from this continuous flow. The map will have D components with $D < N$. We will now give a general formulation of how to obtain the *numerical* discrete system $\mathbf{F}(\mathbf{v}_n)$ from the flow $\tilde{\phi}(\tilde{\mathbf{v}}, t)$.

The mapping is produced by collecting successive points of directed intersections between the flow and a predefined hypersurface \mathcal{P} of dimension $N-1$, the PSS. If this hypersurface is the only constraint available, the numerical mapping $\mathbf{F}(\mathbf{v}_n)$ will have $D=N-1$ components. In general though, we can have N_c constraints between the different variables of the system and $D=N-N_c$. For example, we could be studying a Hamiltonian system with $N=4$ and 2 constraints ($N_c=2$). One of these constraints would come from the conservation of energy and the other from the PSS.

Those N_c constraints (including the PSS) are important since they make possible the construction of the discrete map. We will denote by $\tilde{\mathbf{v}}(t)$ the state of the flow at time t and use $\tilde{\mathbf{v}}_n$ to describe the state of the system *on* the PSS at the n th intersection ($\tilde{\mathbf{v}}_n \in \mathcal{R}^N$). In distinction, the vector \mathbf{v}_n corresponds to the state in the numerically constructed discrete system $\mathbf{F}(\mathbf{v}_n)$ ($\mathbf{v}_n, \mathbf{F} \in \mathcal{R}^D$). It is important to realize that the vector \mathbf{v}_n is *not* a new coordinate system, but it is a projection of $\tilde{\mathbf{v}}_n$ in a smaller subsystem. We are simply removing N_c components from the vector $\tilde{\mathbf{v}}_n$. The N_c removed components are noted by $\check{\mathbf{v}}_n \in \mathcal{R}^{N_c}$.

The constraints are usually given by equations. By specifying the state \mathbf{v}_n , we can deduce $\check{\mathbf{v}}_n$ and recover the whole state $\tilde{\mathbf{v}}_n$ by using the equations of constraints. In general, the components $\check{\mathbf{v}}_n$ will depend on the variables that we keep (\mathbf{v}_n) and we will therefore write $\check{\mathbf{v}}_n(\mathbf{v}_n)$. The vector state on the PSS ($\tilde{\mathbf{v}}_n$) is then noted

$$\tilde{\mathbf{v}}_n = \begin{pmatrix} \mathbf{v}_n \\ \check{\mathbf{v}}_n(\mathbf{v}_n) \end{pmatrix}. \tag{2}$$

This notation will prove useful in calculating the Jacobian matrix of the discrete system $\mathbf{F}(\mathbf{v}_n)$. For simplicity and without lack of generality, we choose the first D components of $\tilde{\mathbf{v}}_n$ to form \mathbf{v}_n .

To obtain the iterate $\tilde{\mathbf{v}}_n$, we consider the intersection of the flow with the PSS. In general, this section will be a hyperplane described by the following equation:

$$h(\tilde{\mathbf{v}}(t)) = \tilde{\xi}^T(\tilde{\mathbf{v}}(t) - \tilde{\mathbf{v}}_0), \tag{3}$$

where $\tilde{\mathbf{v}}_0$ is a point on the hyperplane and $\tilde{\xi}$, a vector orthogonal to this hyperplane. If $\tilde{\mathbf{v}}(t)$ is a solution of the integration, then we will have $h(\tilde{\mathbf{v}}(t))=0$ when $\tilde{\mathbf{v}}(t)$ is precisely *on* the hyperplane. The sign of the function $h(\tilde{\mathbf{v}}(t))$ will change whenever $\tilde{\mathbf{v}}(t)$ goes from one side of the hyperplane to the other.

The application $\tilde{\mathbf{v}}_n \mapsto \tilde{\mathbf{v}}_{n+1}$ is obtained by integrating Eq. (1) from $\tilde{\mathbf{v}}_n$ for a particular time $t = \tau(\tilde{\mathbf{v}}_n) \equiv \tau_n$,

$$\tilde{\mathbf{v}}_{n+1} = \tilde{\phi}(\tilde{\mathbf{v}}_n, \tau(\tilde{\mathbf{v}}_n)). \tag{4}$$

This time will have to be chosen such that $\tilde{\mathbf{v}}_{n+1}$ is on the hyperplane and also satisfies an *orientation condition*. By convention, we will keep the intersection points when the flow is going from negative to positive values of h .

The numerically constructed map $\mathbf{F}(\mathbf{v}_n)$ is obtained by taking the first D components of $\tilde{\mathbf{v}}_{n+1}$ [Eq. (4)]. The map $\mathbf{F}(\mathbf{v}_n)$ can be noted

$$\mathbf{F}(\mathbf{v}_n) = \phi(\tilde{\mathbf{v}}_n, \tau(\tilde{\mathbf{v}}_n)) = \phi(\mathbf{v}_n, \check{\mathbf{v}}_n(\mathbf{v}_n), \tau(\mathbf{v}_n, \check{\mathbf{v}}_n(\mathbf{v}_n))). \tag{5}$$

The first equality comes from Eq. (4), where we have removed the $\tilde{}$ on the vector $\tilde{\phi}$ to remind us that we only use the first D components. The second expression comes from Eq. (2). The advantage of this notation is that it highlights the dependence of \mathbf{F} on \mathbf{v} , which, as we will see, is important to establish the Jacobian matrix of \mathbf{F} . Equation (5) has the following meaning. To evaluate the iterate of \mathbf{v}_n in the numerically constructed discrete system, we first use the equations of constraint ($\check{\mathbf{v}}(\mathbf{v}_n)$) to recover the state $\tilde{\mathbf{v}}_n$. We then proceed to the numerical integration of the flow from $\tilde{\mathbf{v}}_n$. We integrate for a specific time $t = \tau_n$ when the flow is back on the PSS and the dynamical state is $\tilde{\mathbf{v}}_{n+1}$. We finally take the first D components to form \mathbf{v}_{n+1} and repeat the procedure. The numerical map $\mathbf{F}(\mathbf{v}_n)$ is thereby constructed from the flow.

The difficult part in this process is that we do not know a priori the time τ_n required to get back on the PSS. Some numerical technique must be used to determine τ_n . For instance, one can integrate the flow and look for a change of sign of the function h [Eq. (3)]. By a bracketing procedure, we can adjust the local integration time steps to bring h to

zero within a prescribed accuracy. The total integration time to bring the system back to $h=0$ corresponds to τ_n . Another way of doing things is known as *Hénon's method*.¹ Hénon has suggested to introduce a variable $u \equiv h(\tilde{\mathbf{v}}(t))$ and to use it as the new integration variable. This approach is very elegant and powerful and has been used in most of our calculations. We refer the reader to Refs. 1 and 3 for a complete description of the method.

We now turn to the determination of the $D \times D$ Jacobian matrix $\mathbf{J}(\mathbf{v}_n)$ of the numerical map $\mathbf{F}(\mathbf{v}_n)$, i.e., the matrix relating subsequent infinitesimal variations on the PSS by

$$\delta\tilde{\mathbf{v}}_{n+1} = \mathbf{J}(\mathbf{v}_n) \delta\tilde{\mathbf{v}}_n.$$

It is obtained by differentiating the map \mathbf{F} with respect to \mathbf{v} . One notes that because of the dependence of the constrained variables and the return times on \mathbf{v} , a *total* derivative must be taken. From (5) and using the chain rule for derivatives, we get

$$\begin{aligned} \mathbf{J}(\mathbf{v}_n) &= \left. \frac{d}{d\mathbf{v}} \mathbf{F}(\mathbf{v}_n) \right|_{\mathbf{v}_n} \\ &= \left. \frac{d}{d\mathbf{v}} \phi(\mathbf{v}_n, \check{\mathbf{v}}_n(\mathbf{v}_n), \tau(\mathbf{v}_n, \check{\mathbf{v}}_n(\mathbf{v}_n))) \right|_{\check{\mathbf{v}}_n, \tau_n} \\ &= D_{\mathbf{v}} \phi|_{\check{\mathbf{v}}_n, \tau_n} + D_{\check{\mathbf{v}}} \phi|_{\check{\mathbf{v}}_n, \tau_n} D_{\mathbf{v}} \check{\mathbf{v}}|_{\check{\mathbf{v}}_n} + \left[\frac{\partial \phi}{\partial \tau} \frac{d}{d\mathbf{v}} \tau \right]_{\check{\mathbf{v}}_n, \tau_n} \\ &= D_{\mathbf{v}} \phi|_{\check{\mathbf{v}}_n, \tau_n} + D_{\check{\mathbf{v}}} \phi|_{\check{\mathbf{v}}_n, \tau_n} D_{\mathbf{v}} \check{\mathbf{v}}|_{\check{\mathbf{v}}_n} \\ &\quad + \left[\frac{\partial \phi}{\partial \tau} \{ D_{\mathbf{v}} \tau + D_{\check{\mathbf{v}}} \tau D_{\mathbf{v}} \check{\mathbf{v}} \} \right]_{\check{\mathbf{v}}_n, \tau_n}, \end{aligned} \tag{6}$$

where $D_{\mathbf{x}}$ denotes the partial derivatives with respect to the components of \mathbf{x} . By regrouping the different terms and making use of Eq. (1), it is possible to interpret the individual contributions to the Jacobian matrix. This is done by isolating all the elements depending explicitly on $\check{\mathbf{v}}$, viz.

$$\begin{aligned} \mathbf{J}(\mathbf{v}_n) &= D_{\mathbf{v}} \phi|_{\check{\mathbf{v}}_n, \tau_n} \\ &\quad + [\mathbf{f}(\check{\boldsymbol{\phi}}) D_{\mathbf{v}} \tau]_{\check{\mathbf{v}}_n, \tau_n} + \{ [D_{\check{\mathbf{v}}} \phi + \mathbf{f}(\check{\boldsymbol{\phi}}) D_{\check{\mathbf{v}}} \tau] D_{\mathbf{v}} \check{\mathbf{v}} \}_{\check{\mathbf{v}}_n, \tau_n}. \end{aligned} \tag{7}$$

This is the central equation of this section and it is worthwhile to look at each term in turn. The first term can be calculated by integrating the variational equations associated with Eq. (1). These equations will generate the $N \times N$ Jacobian matrix $\tilde{\mathbf{J}}_{\check{\boldsymbol{\phi}}}(\tilde{\mathbf{v}}, t)$. It collects the information on how a perturbation evolves with time in the neighborhood of $\check{\boldsymbol{\phi}}(\tilde{\mathbf{v}}, t)$ and takes the explicit form

$$\tilde{\mathbf{J}}_{\check{\boldsymbol{\phi}}}(\tilde{\mathbf{v}}, t) \equiv D_{\check{\mathbf{v}}} \check{\boldsymbol{\phi}}(\tilde{\mathbf{v}}, t) = \begin{pmatrix} \frac{\partial \check{\boldsymbol{\phi}}^1}{\partial \tilde{v}^1} & \dots & \frac{\partial \check{\boldsymbol{\phi}}^1}{\partial \tilde{v}^N} \\ \vdots & \ddots & \vdots \\ \frac{\partial \check{\boldsymbol{\phi}}^N}{\partial \tilde{v}^1} & \dots & \frac{\partial \check{\boldsymbol{\phi}}^N}{\partial \tilde{v}^N} \end{pmatrix}. \tag{8}$$

This matrix is a solution of the variational equations³

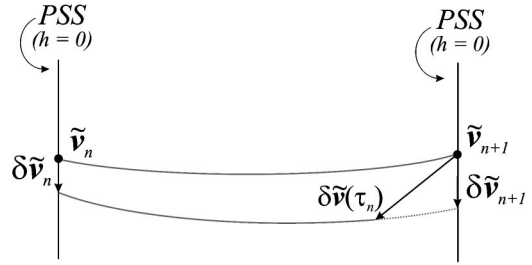


FIG. 1. Geometric interpretation of Eq. (7).

$$\frac{d\tilde{\mathbf{J}}_{\check{\boldsymbol{\phi}}}(\tilde{\mathbf{v}}, t)}{dt} = D_{\check{\mathbf{v}}} \tilde{\mathbf{f}}(\tilde{\mathbf{v}}) \tilde{\mathbf{J}}_{\check{\boldsymbol{\phi}}}(\tilde{\mathbf{v}}, t), \tag{9}$$

with initial condition $\tilde{\mathbf{J}}_{\check{\boldsymbol{\phi}}}(\tilde{\mathbf{v}}, t=0) = \mathbf{1}$, where $\tilde{\mathbf{f}}(\tilde{\mathbf{v}})$ is the vector field associated with the system (1). The first term of Eq. (7) is calculated by integrating Eq. (9) from $\tilde{\mathbf{v}}_n$ for a particular time $t = \tau_n$. The result of this integration gives the $N \times N$ matrix $\tilde{\mathbf{J}}_{\check{\boldsymbol{\phi}}}(\tilde{\mathbf{v}} = \tilde{\mathbf{v}}_n, t = \tau_n)$. The term $D_{\mathbf{v}} \phi|_{\check{\mathbf{v}}_n, \tau_n}$ corresponds then to the $D \times D$ submatrix of $\tilde{\mathbf{J}}_{\check{\boldsymbol{\phi}}}(\tilde{\mathbf{v}} = \tilde{\mathbf{v}}_n, t = \tau_n)$ made of the first D elements of the first D columns of $\tilde{\mathbf{J}}_{\check{\boldsymbol{\phi}}}(\tilde{\mathbf{v}} = \tilde{\mathbf{v}}_n, t = \tau_n)$.

Figure 1 gives a simple geometrical interpretation for the effect of the Jacobian matrix $\tilde{\mathbf{J}}_{\check{\boldsymbol{\phi}}}(\tilde{\mathbf{v}} = \tilde{\mathbf{v}}_n, t = \tau_n)$ on a small perturbation $\delta\tilde{\mathbf{v}}_n$ added to $\tilde{\mathbf{v}}_n$. The vector $\delta\tilde{\mathbf{v}}(\tau_n)$ represents the perturbation at $t = \tau_n$ since

$$\delta\tilde{\mathbf{v}}(\tau_n) = \tilde{\mathbf{J}}_{\check{\boldsymbol{\phi}}}(\tilde{\mathbf{v}} = \tilde{\mathbf{v}}_n, t = \tau_n) \delta\tilde{\mathbf{v}}_n.$$

In this figure, the two lines represent the integration, for a time $t = \tau_n$ of two different trajectories. The one starting at $\tilde{\mathbf{v}}_n$ will get back on the PSS after a time τ_n . During this *same* period of time, the other trajectory, starting at $\tilde{\mathbf{v}}_n + \delta\tilde{\mathbf{v}}_n$ will not, in general, fall on the PSS. The matrix needed to map an arbitrary deviation $\delta\tilde{\mathbf{v}}_n$ on the PSS to the subsequent one $\delta\tilde{\mathbf{v}}_{n+1}$ (also on the PSS) must acknowledge the implicit dependence of the return time τ on $\tilde{\mathbf{v}}$. In other words,

$$\begin{aligned} \delta\tilde{\mathbf{v}}_{n+1} &\equiv \tilde{\mathbf{J}}(\tilde{\mathbf{v}}_n) \delta\tilde{\mathbf{v}}_n \\ &= \frac{d}{d\tilde{\mathbf{v}}} \check{\boldsymbol{\phi}}(\tilde{\mathbf{v}}, \tau) \delta\tilde{\mathbf{v}}_n = \left[D_{\check{\mathbf{v}}} \check{\boldsymbol{\phi}} + \frac{\partial \check{\boldsymbol{\phi}}}{\partial \tau} D_{\check{\mathbf{v}}} \tau \right]_{\check{\mathbf{v}}_n, \tau_n} \delta\tilde{\mathbf{v}}_n \\ &= \left[\tilde{\mathbf{J}}_{\check{\boldsymbol{\phi}}}(\tilde{\mathbf{v}}, t) + \frac{\partial \check{\boldsymbol{\phi}}}{\partial \tau} D_{\check{\mathbf{v}}} \tau \right]_{\check{\mathbf{v}}_n, \tau_n} \delta\tilde{\mathbf{v}}_n. \end{aligned} \tag{10}$$

It is then clear that since we are looking for the Jacobian matrix of \mathbf{F} , which maps two points *on* the PSS, we need more than just the submatrix $\tilde{\mathbf{J}}_{\check{\boldsymbol{\phi}}}(\tilde{\mathbf{v}}, t)$. This brings us to the second term in Eq. (7), namely, $\mathbf{f}(\check{\boldsymbol{\phi}}) D_{\mathbf{v}} \tau|_{\check{\mathbf{v}}_n, \tau_n}$.

We know how to evaluate $\mathbf{f}(\check{\boldsymbol{\phi}})|_{\check{\mathbf{v}}_n, \tau_n}$: it is simply the first D components of Eq. (1) evaluated at $\check{\boldsymbol{\phi}}(\tilde{\mathbf{v}} = \tilde{\mathbf{v}}_n, t = \tau_n) = \tilde{\mathbf{v}}_{n+1}$. In order to calculate the derivatives $D_{\mathbf{v}} \tau$, we note that the condition that $\delta\tilde{\mathbf{v}}_{n+1}$ must be on the PSS imposes the restriction that $\xi^T \delta\tilde{\mathbf{v}}_{n+1} = 0$, which in turn leads to

$$\tilde{\xi}^T \left[D_{\tilde{\mathbf{v}}} \tilde{\Phi} + \frac{\partial \tilde{\Phi}}{\partial \tau} D_{\tilde{\mathbf{v}}} \tau \right]_{\tilde{\mathbf{v}}_n, \tau_n} = \mathbf{0}. \tag{11}$$

Indeed, since $\partial \tilde{\Phi} / \partial \tau = \tilde{\mathbf{f}}(\tilde{\Phi})$, we obtain easily the relation

$$D_{\tilde{\mathbf{v}}} \tau|_{\tilde{\mathbf{v}}_n, \tau_n} = \frac{-\tilde{\xi}^T D_{\tilde{\mathbf{v}}} \tilde{\Phi}|_{\tilde{\mathbf{v}}_n, \tau_n}}{\tilde{\xi}^T \tilde{\mathbf{f}}(\tilde{\Phi})|_{\tilde{\mathbf{v}}_n, \tau_n}}, \tag{12}$$

and hence the derivatives $D_{\tilde{\mathbf{v}}} \tau$ as its first D components.

The last term in (7) is important if the eliminated variables $\tilde{\mathbf{v}}$ depends on \mathbf{v} , as in the case of Hamiltonian systems. All the elements composing this term can be evaluated. First, $D_{\tilde{\mathbf{v}}} \tilde{\Phi}$ is the $D \times N_c$ submatrix of the variational matrix $\tilde{\mathbf{J}}_{\tilde{\Phi}}(\tilde{\mathbf{v}} = \tilde{\mathbf{v}}_n, t = \tau_n)$. Second, $D_{\tilde{\mathbf{v}}} \tau$ is the line vector composed of the last N_c elements of $D_{\tilde{\mathbf{v}}} \tau|_{\tilde{\mathbf{v}}_n, \tau_n}$ [Eq. (12)]. And finally, $D_{\tilde{\mathbf{v}}} \tilde{\mathbf{v}}$ is an $N_c \times D$ matrix consisting of derivatives of $\tilde{\mathbf{v}}$ with respect to \mathbf{v} as given by the equations of constraints.

This completes the description of the procedure to obtain from a continuous flow a numerically constructed map and its associated Jacobian matrix. In practice, (1) and (9) are integrated simultaneously. As already mentioned, the non trivial aspect of this integration is the determination of the time τ_n needed to get back on the PSS. For systems where the only constraint comes from the PSS (e.g., dissipative systems where the energy is not conserved), and with a judicious choice of the surface of section, one can make $\tilde{\mathbf{v}}$ independent of \mathbf{v} . The calculation of the Jacobian matrix then simplifies greatly since the last term of (7) drops out of the calculation.

A further point of interest is the connection of some of our matrices with the *monodromy matrix*.¹⁴⁻¹⁶ If $\tilde{\mathbf{v}}_n$ is part of closed orbit of period T (or alternatively a m -periodic orbit as registered by m intersections with the PSS, i.e., $\tilde{\mathbf{v}}_{n+m} = \tilde{\mathbf{v}}_n$ with $T = \sum_{i=1}^m \tau_m$), the monodromy matrix $\tilde{\mathbf{M}}_n$ relates an infinitesimal variation $\delta \tilde{\mathbf{v}}_n$ with the corresponding change after one period, $\delta \tilde{\mathbf{v}}_{n+m}$, as

$$\delta \tilde{\mathbf{v}}_{n+m} = \tilde{\mathbf{M}}_n \delta \tilde{\mathbf{v}}_n. \tag{13}$$

According to Eq. (10), this means that

$$\tilde{\mathbf{M}}_n = [\tilde{\mathbf{J}}(\tilde{\mathbf{v}}_{n+m-1}) \cdots \tilde{\mathbf{J}}(\tilde{\mathbf{v}}_n)] \tag{14}$$

and can easily be calculated as a by-product of our analysis. This relationship emphasizes the equivalence between the closed trajectory on the one hand and its periodic intersections with a given PSS on the other. In particular, since the stability of the orbit is directly related to the eigenvalues of $\tilde{\mathbf{M}}_n$ [or the product of $\tilde{\mathbf{J}}(\tilde{\mathbf{v}}_{n+i})$ at each intersection], the closed trajectory and its discrete PSS components will share the same property. It is also clear that one eigenvalue of $\tilde{\mathbf{M}}_n$ will always be equal to unity, corresponding to the direction along the trajectory. This observation may be used as a further test of the numerical accuracy of the construction.

III. IMPLEMENTATION

Since the general formulation may be somewhat hard to digest, a concrete example should help to visualize the implementation of the analysis.

The chosen system is a four-dimensional Hamiltonian flow representing the motion of an electron under the combined influence of a Coulomb and a magnetic field. It goes under the name, *diamagnetic Kepler problem (DKP)*, and occupies central stage in classical and quantum chaos research.¹⁷

It has proven useful^{18,19} to consider a (pseudo-) Hamiltonian function, \hat{H} , in scaled semiparabolic coordinates (here for angular momentum $L=0$)

$$\hat{H} = \frac{1}{2}(p_\mu^2 + p_\nu^2) - \epsilon(\mu^2 + \nu^2) + \frac{1}{8}\mu^2\nu^2(\mu^2 + \nu^2) \equiv 2, \tag{15}$$

for the dynamical evolution where ϵ acts as a dynamical parameter and is related to the physical energy E by $\epsilon = \gamma^{-2/3}E$. The parameter $\gamma = B/B_c$ denotes the strength of the magnetic field relative to the unit $B_c \approx 2.35 \cdot 10^5$ T. As ϵ is varied, the classical flow of \hat{H} covers a wide range of Hamiltonian dynamics reaching from bound, nearly integrable behavior to completely chaotic and unbound motion.^{18,19}

The differential equations for this Hamiltonian are given explicitly by

$$\frac{d\tilde{\mathbf{v}}}{dt} = \begin{pmatrix} \frac{d\tilde{v}^1}{dt} \\ \frac{d\tilde{v}^2}{dt} \\ \frac{d\tilde{v}^3}{dt} \\ \frac{d\tilde{v}^4}{dt} \end{pmatrix} \equiv \begin{pmatrix} \frac{dv}{dt} \\ \frac{dp_\nu}{dt} \\ \frac{d\mu}{dt} \\ \frac{dp_\mu}{dt} \end{pmatrix} = \begin{pmatrix} p_\nu \\ \nu(2\epsilon - \frac{1}{4}\mu^4 - \frac{1}{2}\nu^2\mu^2) \\ p_\mu \\ \mu(2\epsilon - \frac{1}{4}\nu^4 - \frac{1}{2}\nu^2\mu^2) \end{pmatrix}. \tag{16}$$

We have put the first two components of $\tilde{\mathbf{v}}$ as ν and p_ν since these variables are the ones we have selected to describe the dynamics of the discrete map.

The variational equations (9) are given explicitly by

$$\frac{d\tilde{\mathbf{J}}_{\tilde{\Phi}}}{dt} = \tilde{\mathbf{V}}_{\text{DKP}} \tilde{\mathbf{J}}_{\tilde{\Phi}}(\tilde{\mathbf{v}}, t),$$

where $\tilde{\mathbf{V}}_{\text{DKP}}$ stands for

$$\tilde{\mathbf{V}}_{\text{DKP}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 2\epsilon - \frac{1}{4}\mu^4 - \frac{3}{2}\mu^2\nu^2 & 0 & -\mu\nu(\mu^2 + \nu^2) & 0 \\ 0 & 0 & 0 & 1 \\ -\mu\nu(\mu^2 + \nu^2) & 0 & 2\epsilon - \frac{1}{4}\nu^4 - \frac{3}{2}\mu^2\nu^2 & 0 \end{pmatrix}.$$

These 16 differential equations are to be added to the initial system. With the notation

$$\hat{v}^i = \frac{\partial v}{\partial \tilde{v}^i}, \quad \hat{p}_\nu^i = \frac{\partial p_\nu}{\partial \tilde{v}^i},$$

$$\hat{\mu}^i = \frac{\partial \mu}{\partial \tilde{v}^i}, \quad \hat{p}_\mu^i = \frac{\partial p_\mu}{\partial \tilde{v}^i},$$

the variational equations can be written as

$$\frac{d}{dt} \begin{pmatrix} \hat{v}^j \\ \hat{p}_v^i \\ \hat{\mu}^i \\ \hat{p}_\mu^i \end{pmatrix} = \tilde{\mathbf{V}}_{\text{DKP}} \begin{pmatrix} \hat{v}^j \\ \hat{p}_v^i \\ \hat{\mu}^i \\ \hat{p}_\mu^i \end{pmatrix}, \tag{17}$$

for $i=\{1, 4\}$. We then integrate simultaneously (16) and (17) (20 equations in all) from a particular state $\tilde{\mathbf{v}}_n$ and the initial conditions for the variational equations

$$\tilde{\mathbf{J}}_{\tilde{\phi}}(\tilde{\mathbf{v}} = \tilde{\mathbf{v}}_n, t = 0) = \mathbf{1},$$

or

$$\hat{v}^j = \delta_{j1}, \quad \hat{p}_v^i = \delta_{i2},$$

$$\hat{\mu}^i = \delta_{i3}, \quad \hat{p}_\mu^i = \delta_{i4},$$

where δ_{ij} is the Kronecker symbol.

For the DKP, a ‘‘natural’’ choice for the PSS is $\mu=0$ or $\nu=0$. We take $\mu=0$ such that $\tilde{\boldsymbol{\xi}}^T=(0 \ 0 \ 1 \ 0)$ and a point $\tilde{\mathbf{v}}_0$ on the PSS is then $\tilde{\mathbf{v}}_0^T=(\nu_0 \ p_{\nu_0} \ 0 \ p_{\mu_0})$. The function $h(\tilde{\mathbf{v}}(t)) = \tilde{\boldsymbol{\xi}}^T(\tilde{\mathbf{v}}(t) - \tilde{\mathbf{v}}_0)$ is written explicitly as

$$h(\mu, \nu, p_\mu, p_\nu) = (0 \ 0 \ 1 \ 0) \begin{pmatrix} \nu(t) - \nu_0 \\ p_\nu(t) - p_{\nu_0} \\ \mu(t) - 0 \\ p_\mu(t) - p_{\mu_0} \end{pmatrix} = \mu(t). \tag{18}$$

On the PSS, h is by definition equals zero. This gives us a first constraint equation (on the PSS)

$$\text{constraint 1: } \mu_n = 0. \tag{19}$$

Furthermore, the energy is conserved and given by the (pseudo)-Hamiltonian (15) as

$$\hat{H} \equiv 2 = \frac{1}{2}(p_\mu^2 + p_\nu^2) + V(\mu, \nu),$$

where $V(\mu, \nu) = \epsilon(\mu^2 + \nu^2) + \frac{1}{8}\mu^2\nu^2(\mu^2 + \nu^2)$. With this constraint, we choose to eliminate the variable p_μ

$$p_\mu = \sqrt{2(\hat{H} - V(\mu, \nu)) - p_\nu^2},$$

(we choose the + sign in agreement with the orientation convention $d\mu/dt|_{\tau_n} > 0$) which on the PSS gives us a second equation of constraint

$$\text{constraint 2: } p_{\mu_n} = \sqrt{2(\hat{H} - V(\mu_n = 0, \nu_n)) - p_{\nu_n}^2}. \tag{20}$$

In accord with our notation, for the flow of the DKP, we have $N_c=2$ constraint equations. These allow us to write the eliminated variables ($\tilde{\mathbf{v}}_n = \{\mu_n, p_{\mu_n}\}$) as a function of the remaining variables ($\mathbf{v}_n = \{\nu_n, p_{\nu_n}\}$).

The discrete system is then composed solely of ν_n and p_{ν_n} . On the PSS chosen, these variables are area preserving,

i.e., the Jacobian matrix must have a determinant equal to 1.⁸ The four-dimensional vector $\tilde{\mathbf{v}}_n$, composed of the 2 vectors (\mathbf{v}_n and $\tilde{\mathbf{v}}_n$) is

$$\tilde{\mathbf{v}}_n = \begin{pmatrix} \mathbf{v}_n \\ \tilde{\mathbf{v}}_n(\mathbf{v}_n) \end{pmatrix} = \begin{pmatrix} \nu_n^1 \\ \nu_n^2 \\ \tilde{v}_n^1(\mathbf{v}_n) \\ \tilde{v}_n^2(\mathbf{v}_n) \end{pmatrix} \equiv \begin{pmatrix} \nu_n \\ p_{\nu_n} \\ \mu_n \\ p_{\mu_n}(\nu_n, p_{\nu_n}) \end{pmatrix}.$$

To obtain numerically the iterate \mathbf{v}_n and the Jacobian matrix at this point, we proceed as follows. We choose an initial condition $\mathbf{v}=(\nu_n, p_{\nu_n})^T$. Using the constraint equations (19) and (20), we then calculate $\tilde{\mathbf{v}}=(\mu_n, p_{\mu_n})^T$ and integrate simultaneously Eqs. (16) and (17). During the integration, the h function [$h=\mu(t)$] will change sign twice. We stop the integration just before h changes sign for the second time. Recall that we want to stop the integration when we are back on the PSS, so h must be equal 0. This can be achieved by using Hénon’s method for example. We now have in hand the initial condition ($\tilde{\mathbf{v}}_n$), its iterate on the PSS ($\tilde{\mathbf{v}}_{n+1}$) and the time of integration between these states (τ_n). We now calculate the Jacobian matrix, using Eq. (7).

We already have in hand the elements of $D_{\mathbf{v}}\tilde{\phi}$ and $D_{\tilde{\mathbf{v}}}\tilde{\phi}$ taken from the upper part of the 4×4 matrix $D_{\tilde{\mathbf{v}}}\tilde{\phi}$ that we have integrated for a time τ_n [Eq. (17)], explicitly

$$D_{\mathbf{v}}\tilde{\phi}|_{\tilde{\mathbf{v}}_n, \tau_n} \equiv \begin{pmatrix} \frac{\partial \tilde{\phi}^1}{\partial v^1} & \frac{\partial \tilde{\phi}^1}{\partial v^2} \\ \frac{\partial \tilde{\phi}^2}{\partial v^1} & \frac{\partial \tilde{\phi}^2}{\partial v^2} \end{pmatrix}_{\tilde{\mathbf{v}}_n, \tau_n} \equiv \begin{pmatrix} \hat{v}^1 & \hat{v}^2 \\ \hat{p}_v^1 & \hat{p}_v^2 \end{pmatrix}_{\tilde{\mathbf{v}}_n, \tau_n},$$

and

$$D_{\tilde{\mathbf{v}}}\tilde{\phi}|_{\tilde{\mathbf{v}}_n, \tau_n} \equiv \begin{pmatrix} \frac{\partial \tilde{\phi}^1}{\partial \tilde{v}^3} & \frac{\partial \tilde{\phi}^1}{\partial \tilde{v}^4} \\ \frac{\partial \tilde{\phi}^2}{\partial \tilde{v}^3} & \frac{\partial \tilde{\phi}^2}{\partial \tilde{v}^4} \end{pmatrix}_{\tilde{\mathbf{v}}_n, \tau_n} \equiv \begin{pmatrix} \hat{p}_v^3 & \hat{p}_v^4 \\ \hat{p}_v^3 & \hat{p}_v^4 \end{pmatrix}_{\tilde{\mathbf{v}}_n, \tau_n}.$$

Turning to the second term of (7), its first part can be calculated using (16) directly

$$\mathbf{f}(\tilde{\phi})|_{\tilde{\mathbf{v}}_n, \tau_n} = \mathbf{f}(\tilde{\mathbf{v}}_{n+1}) \equiv \begin{pmatrix} f^1(\tilde{\mathbf{v}}_{n+1}) \\ f^2(\tilde{\mathbf{v}}_{n+1}) \end{pmatrix}.$$

The vectors $D_{\mathbf{v}}\tau$ and $D_{\tilde{\mathbf{v}}}\tau$ can be obtained from the complete vector $D_{\tilde{\mathbf{v}}}\tau$ as expressed in (12) by

$$\begin{aligned}
 D_{\check{\mathbf{v}}}\tau|_{\check{\mathbf{v}}_n, \tau_n} &\equiv \left(\frac{\partial \tau}{\partial v} \frac{\partial \tau}{\partial p_v} \frac{\partial \tau}{\partial \mu} \frac{\partial \tau}{\partial p_\mu} \right)_{\check{\mathbf{v}}_n, \tau_n} \\
 &= - \frac{\begin{bmatrix} \tilde{\xi}^T D_{\check{\mathbf{v}}} \tilde{\phi} \\ \tilde{\xi}^T \tilde{\mathbf{f}}(\tilde{\phi}) \end{bmatrix}}{\begin{bmatrix} \tilde{\xi}^T \tilde{\mathbf{f}}(\tilde{\phi}) \end{bmatrix}}_{\check{\mathbf{v}}_n, \tau_n} \\
 &= - \frac{\begin{pmatrix} \hat{v}^1 & \hat{v}^2 & \hat{v}^3 & \hat{v}^4 \\ \hat{p}_v^1 & \hat{p}_v^2 & \hat{p}_v^3 & \hat{p}_v^4 \\ \hat{\mu}^1 & \hat{\mu}^2 & \hat{\mu}^3 & \hat{\mu}^4 \\ \hat{p}_\mu^1 & \hat{p}_\mu^2 & \hat{p}_\mu^3 & \hat{p}_\mu^4 \end{pmatrix}}{\begin{pmatrix} \tilde{f}^1(\tilde{\mathbf{v}}_{n+1}) \\ \tilde{f}^2(\tilde{\mathbf{v}}_{n+1}) \\ \tilde{f}^3(\tilde{\mathbf{v}}_{n+1}) \\ \tilde{f}^4(\tilde{\mathbf{v}}_{n+1}) \end{pmatrix}}_{\check{\mathbf{v}}_n, \tau_n} \\
 &= \frac{-1}{\tilde{f}^3(\tilde{\mathbf{v}}_{n+1})} (\hat{\mu}^1 \hat{\mu}^2 \hat{\mu}^3 \hat{\mu}^4)_{\check{\mathbf{v}}_n, \tau_n} \\
 &= \frac{-1}{p_{\mu_{n+1}}} (\hat{\mu}^1 \hat{\mu}^2 \hat{\mu}^3 \hat{\mu}^4)_{\check{\mathbf{v}}_n, \tau_n}. \tag{21}
 \end{aligned}$$

The vector $D_{\check{\mathbf{v}}}\tau|_{\check{\mathbf{v}}_n, \tau_n}$ ($D_{\check{\mathbf{v}}}\tau|_{\check{\mathbf{v}}_n, \tau_n}$) corresponds to the first (last) two components of this equation, namely

$$D_{\check{\mathbf{v}}}\tau|_{\check{\mathbf{v}}_n, \tau_n} \equiv \left(\frac{\partial \tau}{\partial v} \frac{\partial \tau}{\partial p_v} \right)_{\check{\mathbf{v}}_n, \tau_n} = \frac{-1}{p_{\mu_{n+1}}} (\hat{\mu}^1 \hat{\mu}^2)_{\check{\mathbf{v}}_n, \tau_n} \tag{22}$$

$$D_{\check{\mathbf{v}}}\tau|_{\check{\mathbf{v}}_n, \tau_n} \equiv \left(\frac{\partial \tau}{\partial \mu} \frac{\partial \tau}{\partial p_\mu} \right)_{\check{\mathbf{v}}_n, \tau_n} = \frac{-1}{p_{\mu_{n+1}}} (\hat{\mu}^3 \hat{\mu}^4)_{\check{\mathbf{v}}_n, \tau_n}. \tag{23}$$

The remaining contribution consists of derivatives of the constrained vector $\check{\mathbf{v}}=(\mu p_\mu)^T$ with respect to \mathbf{v} . This gives rise to the 2×2 matrix $D_{\check{\mathbf{v}}}\check{\mathbf{v}}$ given by

$$D_{\check{\mathbf{v}}}\check{\mathbf{v}} = \begin{pmatrix} 0 & 0 \\ -\frac{1}{p_\mu} \frac{\partial V}{\partial v} & -\frac{p_v}{p_\mu} \end{pmatrix} \tag{24}$$

and with the further simplification that $-\partial V / \partial v = dp_v / dt = \tilde{f}^2$, one finally gets

$$D_{\check{\mathbf{v}}}\check{\mathbf{v}}|_{\check{\mathbf{v}}_n, \tau_n} = \frac{1}{p_{\mu_n}} \begin{pmatrix} 0 & 0 \\ \tilde{f}^2(\mu_n = 0, v_n) & -p_v \end{pmatrix}. \tag{25}$$

This completes the procedure for calculating the map $\mathbf{F}(\mathbf{v}_n)$ and its Jacobian matrix on the PSS for the flow of the DKP. A typical numerical implementation gives a map whose elements are located on the PSS with an absolute accuracy of 10^{-15} (we have used Hénon’s method) with a Jacobian matrix of determinant 1 to better than 10^{-14} . (In theory, the determinant has to be 1 for this particular flow.) Figure 2 illustrates a projection of the flow in a three-dimensional phase space and on the PSS for $\epsilon=-0.1$.

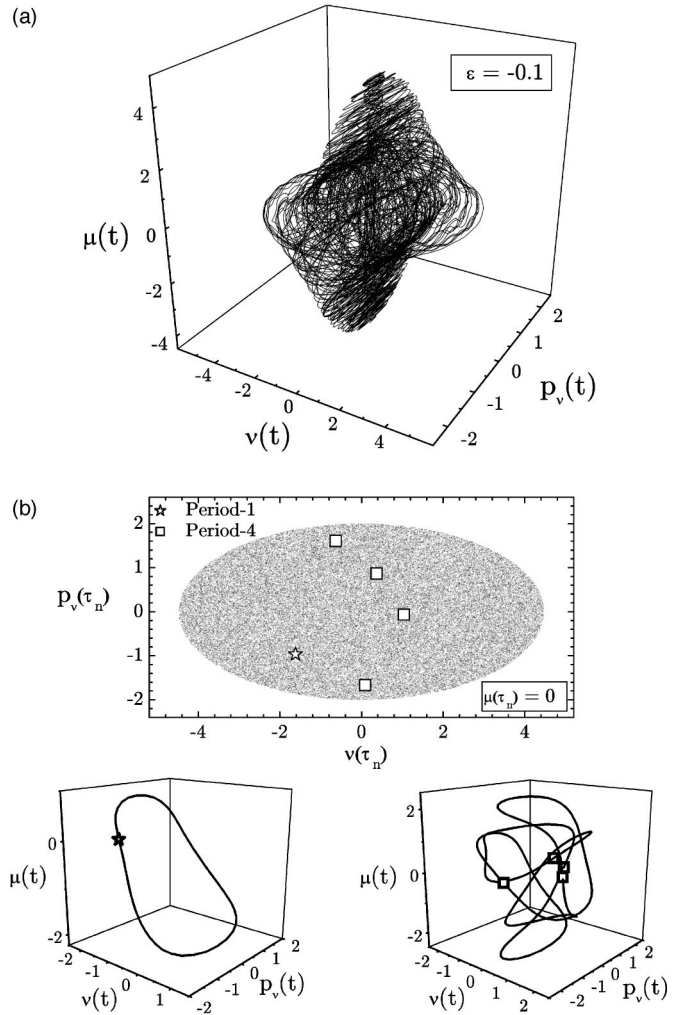


FIG. 2. (a) Projection of the flow on a three-dimensional phase space ($\epsilon=-0.1$); (b) The PSS ($\mu(\tau_n)=0$) where two typical UPOs of period-1 and period-4 are indicated. Notice the absence of (resolved) regular phase space structures at this pseudo-energy indicating almost complete chaotic behavior. In the bottom panel, the corresponding trajectories in the (μ, v, p_v) space of (a) are also sketched.

IV. APPLICATION: DETECTION OF PERIODIC ORBITS

In the last sections, we have given the details for constructing a discrete map $\mathbf{F}(\mathbf{v}_n)$ and its Jacobian matrix from a continuous flow. The procedure can be very useful if one has to work with a discrete dynamical system, for instance, while considering recent methods for the detection of periodic orbits. The effectiveness of our technique will be exemplified by a search of UPOs in the DKP.

Before embarking in the study of this specific example, it is perhaps instructive to pause to describe the classical Newton–Raphson (NR) search,²⁰ specifically for continuous flows. For an N -dimensional flow, discretized on a PSS, a period- m orbit is found as the zeros of the following function:

$$\tilde{\mathbf{G}}(\tilde{\mathbf{v}}) = \tilde{\mathbf{F}}^{(m)}(\tilde{\mathbf{v}}) - \tilde{\mathbf{v}}, \tag{26}$$

where $\tilde{\mathbf{F}}^{(m)}$ consists of m applications of the corresponding nonlinear map $\tilde{\mathbf{F}}$.

Conceptually, any numerical approach to find the zeros of a nonlinear function amounts to the choice of an initial condition $\tilde{\mathbf{v}}_0$ and its gradual modification according to an iterative scheme

$$\tilde{\mathbf{v}}_{i+1} = \tilde{\mathbf{v}}_i + \delta\tilde{\mathbf{v}}_i \quad (i \geq 0) \quad (27)$$

until the condition $\tilde{\mathbf{G}}(\tilde{\mathbf{v}}_{i+1}) \approx 0$ is met to a certain accuracy.

The correction for the NR algorithm is obtained by imposing that the linearized function $\tilde{\mathbf{G}}(\tilde{\mathbf{v}}_i + \delta\tilde{\mathbf{v}}_i)$ satisfies

$$\tilde{\mathbf{G}}(\tilde{\mathbf{v}}_i + \delta\tilde{\mathbf{v}}_i) \sim \tilde{\mathbf{G}}(\tilde{\mathbf{v}}_i) + \frac{d\tilde{\mathbf{G}}}{d\tilde{\mathbf{v}}} \delta\tilde{\mathbf{v}}_i = 0,$$

from which the equation for $\delta\tilde{\mathbf{v}}_i$ follows

$$(\mathbf{1} - \tilde{\mathbf{J}}^{(m)}) \delta\tilde{\mathbf{v}}_i = \tilde{\mathbf{G}}(\tilde{\mathbf{v}}_i), \quad (28)$$

where $\tilde{\mathbf{J}}^{(m)}$ is obtained from the integration of the variational equations after m intersections (in the same directions) with the PSS [Eqs. (8) and (9)].

For flows, the NR method runs into two difficulties. First, for every periodic orbit, one of the eigenvalues of $\tilde{\mathbf{J}}^{(m)}$ is equal to 1, and the procedure may become unstable. Secondly, $\tilde{\mathbf{G}}(\tilde{\mathbf{v}}_i + \delta\tilde{\mathbf{v}}_i)$ is in general *not* on the PSS. These two problems are solved simultaneously by the addition of one extra equation to Eq. (28).³ Furthermore, if the flow is Hamiltonian, another eigenvalue is equal to 1 and one more equation is needed to remove the instability.⁹

Within the NR scheme, our alternative solution is to work, not with the N -dimensional system, but rather with the D -dimensional map $\mathbf{F}^{(m)}(\mathbf{v})$ and its associated Jacobian matrix $\mathbf{J}^{(m)}(\mathbf{v})$ [Eqs. (5) and (7)]. By construction, since $\mathbf{J}^{(m)}$ is constrained to the PSS and takes into account the τ and $\check{\mathbf{v}}$ dependence on \mathbf{v} , the inherent instabilities are effectively removed. The function \mathbf{G} is reduced to D components

$$\mathbf{G}(\mathbf{v}) = \mathbf{F}^{(m)}(\mathbf{v}) - \mathbf{v} \quad (29)$$

as is the iterative step

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \delta\mathbf{v}_i, \quad (30)$$

where the corrections $\delta\mathbf{v}_i$ are obtained from the solution of

$$(\mathbf{1} - \mathbf{J}^{(m)}) \delta\mathbf{v}_i = \mathbf{G}(\mathbf{v}_i). \quad (31)$$

One notes that $\mathbf{G}(\mathbf{v})$ depends however implicitly on the complete state $\tilde{\mathbf{v}}$ where the $N_c = N - D$ missing components are calculated from the constraining equations [see Eq. (5)].

Whatever implementation is used, the augmented system (28) or the reduced set (31), quadratic convergence is guaranteed for sufficiently close initial starting conditions. However, the NR method loses rapidly its efficacy for increasing m since the basin of attraction of every single periodic orbit becomes very small and the choice of suitable initial conditions becomes increasingly cumbersome. This has led Schmelcher and Diakonou¹⁰ to propose an ingenious correction $\delta\mathbf{v}_i$ that has the effect of increasing the size of the basins of attraction for the periodic orbits and therefore making the convergence more global.

Further, a hybrid scheme combining the global convergence of Schmelcher–Diakonou method to the fast conver-

TABLE I. Periodic orbits in the DKP ($\epsilon = -0.1$). For each period p , we give the number of periodic orbits detected on the PSS given by $(\mu_n = 0, (d\mu/dt)|_{\tau_n} > 0)$, their stability, $\langle |\rho_u| \rangle$, gauged by the average of the largest absolute eigenvalues of the Jacobian matrix, the average time, $\langle \tau \rangle$, and the longest time, τ_{\max} , taken for p returns on the PSS.

Period	Number of POs	$\langle \rho_u \rangle$	$\langle \tau \rangle$	τ_{\max}
1	323	844.5	10.8	14.0
2	1088	37983.1	17.4	27.9
3	1232	45935.2	17.4	40.7
4	2550	131826.2	20.3	28.3
5	2932	206057.0	22.2	32.5

gence rate of the NR algorithm has been proposed by Davidchack and Lai.¹² They consider a correction $\delta\mathbf{v}_i$ given by

$$\delta\mathbf{v}_i = [\mathbf{1}\beta g(\mathbf{v}_n) - \mathbf{C}_k \mathbf{J}_{G_c}(\mathbf{v}_n)]^{-1} \mathbf{C}_k \mathbf{G}(\mathbf{v}_n), \quad (32)$$

where $g(\mathbf{v}_n) \equiv \|\mathbf{G}(\mathbf{v}_n)\| \geq 0$, \mathbf{C}_k is an orthogonal $D \times D$ matrix with element $C_{ij} \in \{0, \pm 1\}$ and whose rows and columns contain only one entry different from zero (see Ref. 10 for further details), and where β is a constant parameter adjusted to optimize the performance of the algorithm. The Jacobian matrix of \mathbf{G} , noted \mathbf{J}_G , can be evaluated from the knowledge of $\mathbf{J}^{(m)}(\mathbf{v}_n) = d\mathbf{F}^{(m)}/d\mathbf{v}|_{\mathbf{v}_n}$ as

$$\mathbf{J}_G(\mathbf{v}_n) = \mathbf{J}^{(m)}(\mathbf{v}_n) - \mathbf{1}, \quad (33)$$

according to the prescription (7). Since a complete search requires that the iterations be performed for all matrices \mathbf{C}_k (their number grows rapidly with dimension D), the DL corrections are optimally constructed for the smallest possible dimension of the Jacobian matrix; this is where our method gains its full strength.

Implementing the Davidchack–Lai algorithm on the DKP, we have made an (almost) exhaustive determination of periodic orbits up to $m=5$. The results are presented in Table I. We present in this table the number of UPOs found, the average stability of the m -cycle ($\langle |\rho_u| \rangle$), given by the average of the largest absolute eigenvalues of the Jacobian matrix, the average time ($\langle \tau \rangle$) and the longest time (τ_{\max}) taken for m returns on the the PSS. Two periodic orbits (period 1 and 4) are indicated in Fig. 2.

For the detection of a particular period, we use several initial conditions ($\approx 10^4$) and use the iterative scheme (30) with $\delta\mathbf{v}_i$ given by (32). With the method of Sec. III we construct the map $\mathbf{F}(\mathbf{v}_n)$ from the flow. Recall that the PSS is placed at $\mu=0$ and p_μ is eliminated from the map using the constraint of energy conservation. The dimension of phase space of the numerically constructed map $\mathbf{F}(\mathbf{v}_n)$ is $D=2$ and $\mathbf{v}_n \equiv (v_n p_{v_n})^T$.

For the Davidchack–Lai correction (32), we choose $\beta = 20$, but this value is not critical and other values can be used. Every initial condition is iterated for 10β steps or until convergence to a periodic orbit. The initial conditions are selected uniformly in the first quadrant of the $v-p_v$ plane and must be such that p_μ [Eq. (20)] takes on real values. Only the first quadrant has to be considered since the differential equations for the DKP are invariant under the reflection $v \rightarrow -v$ and $p_v \rightarrow -p_v$. Once a periodic orbit is found, we use

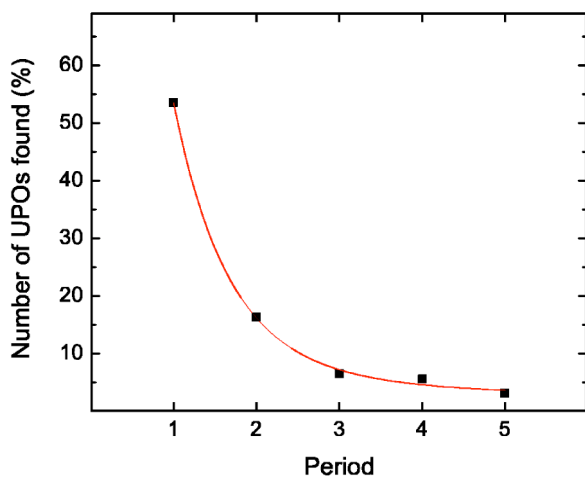


FIG. 3. This figure shows the importance of an accurate Jacobian matrix for the detection of UPOs. When adding 5% of random noise to each element of the Jacobian matrix of $F^{(m)}(\mathbf{v})$, the number of UPOs detected decreases considerably with increasing period. The ordinate corresponds to the number of UPOs detected with respect to the number found with an accurate (10^{-14}) Jacobian matrix for the same initial condition.

the same symmetry to deduce the other periodic orbits associated with the reflection $\nu \rightarrow -\nu$ and $p_\nu \rightarrow -p_\nu$.

In a recent article, Pingel *et al.*¹³ have also considered the DKP in order to extract UPOs. Our reported results show a much greater number of different UPOs detected. Although, they did not claim explicitly to have made an exhaustive search, the discrepancy (~ 2 orders of magnitude) is somewhat puzzling. Two main reasons come to mind to explain the difference.

First, the authors have used a somewhat different method (although similar in spirit) than the approach of Davidchack and Lai. They combine their correction¹⁰ for the global approach to a UPO with a NR correction once they have entered a close enough neighborhood of the UPO. In contrast, the scheme of Davidchack and Lai is a unique correction that combines global with quadratic NR convergence. However, our experience²¹ indicates that although a loss of efficiency may be experienced with their procedure, with sufficient effort it does not lead to a net loss of detected UPOs.

Second, as mentioned previously, the treatment of Hamiltonian systems must acknowledge the fact that the energy is an invariant of the dynamics. The usual procedure is to include this constraint explicitly in the NR steps [see e.g., Ref. 9, Eq. (14.18)] in order to remove the associated numerical instability. It would appear that the procedure used in Ref. 13 [their Eq. (16)] does not take into account this aspect of the calculation. This may have a small incidence on the results if one is sufficiently close to the UPO, although the individual NR increments are not quite correct.

Irrespective of the final reason for the difference in UPOs found, the situation has induced us to consider the influence of a less than accurate Jacobian matrix on the detection of the UPOs. We have therefore repeated the calculations of Table I by adding 5% of white noise to every element of our Jacobian matrix. As shown in Fig. 3, a rapid

decrease of the number of UPOs detected is observed, ranging from a loss of 1/2 of the UPOs of period 1 to 1/20 for the period 5; altogether this amounts to only 10% of the previously found UPOs.

V. CONCLUDING REMARKS

In this paper, we have presented a method to calculate the Jacobian matrix of a dynamical map $F(\mathbf{v})$ obtained numerically from an arbitrary flow. The method is general and particularly useful if there are constraints other than the one imposed by the Poincaré surface of section where the map is constructed. Hamiltonian systems fall in this category and an explicit formulation of the procedure was given. Its implementation is straightforward as exemplified by the set of equations obtained for a conservative dynamical system, namely the *Diamagnetic Kepler Problem* (DKP). The accuracy of our method was tested on the DKP as well as on the dissipative Lorenz flow (Ref. 22, not shown here), where, in both cases, analytical results exist for the Jacobian.

The usefulness of the Jacobian matrix was shown by calculating the Unstable Periodic Orbits of the DKP and the importance of its accurate construction was demonstrated.

ACKNOWLEDGMENTS

We are grateful for partial financial support by the agencies FQRNT (Québec) and NSERC (Canada).

¹M. Hénon, *Physica D* **5**, 412 (1982).

²E. A. Jackson, *Perspectives of Nonlinear Dynamics* (Cambridge University Press, Cambridge, 1989).

³T. S. Parker and L. O. Chua, *Practical Numerical Algorithms for Chaotic Systems* (Springer-Verlag, Berlin, 1989).

⁴D. Auerbach, P. Cvitanović, J. P. Eckmann, G. H. Gunaratne, and I. Proraccia, *Phys. Rev. Lett.* **58**, 2387 (1987).

⁵C. Grebogi, E. Ott, and J. A. Yorke, *Phys. Rev. A* **37**, 1711 (1988).

⁶P. Cvitanović, *Phys. Rev. Lett.* **61**, 2729 (1988).

⁷R. Artuso, E. Aurell, and P. Cvitanović, *Nonlinearity* **3**, 325 (1990).

⁸M. C. Gutzwiller, *Chaos in Classical and Quantum Mechanics* (Springer-Verlag, Berlin, 1990).

⁹P. Cvitanović, R. Artuso, R. Mainieri, G. Tanner, and G. Vattay, *Chaos: Classical and Quantum* (Niels Bohr Institute, Copenhagen, 2003), www.nbi.dk/ChaosBook

¹⁰P. Schmelcher and F. K. Diakonov, *Phys. Rev. Lett.* **78**, 4733 (1997); P. Schmelcher and F. K. Diakonov, *Phys. Rev. E* **57**, 2739 (1998); F. K. Diakonov, P. Schmelcher, and O. Biham, *Phys. Rev. Lett.* **81**, 4349 (1998).

¹¹S. M. Zoldi and H. S. Greenside, *Phys. Rev. E* **57**, R2511 (1998).

¹²R. L. Davidchack and Y. C. Lai, *Phys. Rev. E* **60**, 6172 (1999); R. L. Davidchack, Y. C. Lai, A. Klebanoff, and E. M. Bollt, *Phys. Lett. A* **287**, 99 (2001).

¹³D. Pingel, P. Schmelcher, and F. K. Diakonov, *Phys. Rev. E* **64**, 026214 (2001).

¹⁴M. Baranger, K. Davies, and J. Mahoney, *Ann. Phys. (N.Y.)* **186**, 95 (1988).

¹⁵K. T. R. Davies, T. E. Huston, and M. Baranger, *Chaos* **2**, 215 (1992).

¹⁶N. S. Simonovic, *Chaos* **9**, 854 (1999).

¹⁷R. Blümel and W. P. Reinhardt, *Chaos in Atomic Physics* (Cambridge University Press, Cambridge, 1997).

¹⁸J. B. Delos, S. K. Knudson, and D. W. Noid, *Phys. Rev. A* **30**, 1208 (1984).

¹⁹H. Friedrich and D. Wintgen, *Phys. Rep.* **183**, 37 (1989).

²⁰W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C* (Cambridge University Press, Cambridge, 1996).

²¹B. Doyon, Ph.D. thesis, Université Laval, 2002.

²²E. N. Lorenz, *J. Atmos. Sci.* **20**, 130 (1963).