ELSEVIER

Physica D 149 (2001) 30–42

PHYSICA D

www.elsevier.com/locate/physd

An algorithm to approach invariant curves

Chi-Jer Yu*, Jong-Eao Lee[1]

Department of Applied Mathematics, National Chiao Tung University, Hsinchu 30050, Taiwan

Received 12 January 1999; received in revised form 13 September 2000; accepted 13 September 2000
Communicated by A.C. Newell

Abstract

We develop an algorithm to identify invariant curves efficiently. In dynamical systems, it helps do continuations of branches of quasi-periodic solutions smoothly in bifurcation diagrams. Several examples are demonstrated to provide the numerical evidence for the versatility of the algorithm. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Quasi-periodic; Invariant; Tori; Numerical; Algorithm

1. Introduction

It is well known that, in a dynamical system, the bifurcation diagram from steady-state solutions to branches of periodic solutions can be smoothly developed by the local theory and the continuation method. Due to the non-recurrence of quasi-periodic solutions along with some theoretic and computational difficulties (detailed later), a further continuation from a branch of periodic solutions to a new branch of quasi-periodic solutions is always a big challenge (in particular, for high-dimensional systems). There are two nontrivial key problems involved, one is searching for a proper starting point for the new branch; the next one is, for each parameter in this new branch, to find out the corresponding quasi-periodic solution; in other words, to identify an invariant torus (or curve). The first problem can be approached rather technically and case by case. The second one may be solved by algorithms (e.g. [3–5,7,8]). Of course, in order to complete the entire branch smoothly, one must make sure that the applied algorithm works in each and every parameter in this interval of parameters. From our experience, some algorithms cited above are truly excellent under a fixed generic parameter, but they do have their own problems in doing the continuation when the parameter varies (as commented in [3–6] and some detailed later). In fact, we have tried, in many ways and for some long time, to complete the continuation for certain high-dimensional non-autonomous system (i.e. Section 3.2). Thanks to Kaas-Petersen [3] and Kevrekidis et al. [4], we adopt and modify their ideas to create a new one, which, roughly speaking, has a smaller size (in doing computation of an invariant torus at a fixed parameter) and a better structure (to be more stable in doing continuation of the branch in varying parameters). In this paper, we provide two different formulations for our algorithm, parameterized by polar coordinate and arc length coordinate,

* Corresponding author. Tel.: +886-3-5712121, ext: 56444; fax: +886-3-5724679.
E-mail addresses: u8322805@math.nctu.edu.tw (C.-J. Yu), jlee@math.nctu.edu.tw (J.-E. Lee).
[1] Tel.: +886-3-5712121, ext: 56412; fax: +886-3-5724679.

0167-2789/01/$ – see front matter © 2001 Elsevier Science B.V. All rights reserved.
PII: S0167-2789(00)00181-0

respectively, to search for invariant tori, one of them is particularly good and convenient for high-dimensional cases. We also suggest an simple and direct idea to see how to determine the possibly least amount of nodes[2] to start with our algorithm for each fixed parameter. As demonstrated in Section 3, our algorithm succeeds in search for T^2-invariant sets in a more powerful, versatile, less cost and weakness manner as illustrated by the results and the comparisons shown; moreover, it accomplishes the entire continuation process for Section 3.2. Indeed, it has been tested to be able to work very nicely for the general problems.

2. Algorithms for identifying invariant curves

For dynamical systems restricted to a Poincaré section, flows are reduced to a map, and invariant tori become invariant curves. Here, we concentrate on how to identify invariant curves for maps.

First, we briefly review the method by Kevrekidis et al. [4]. In polar coordinates, let $F(\theta, x) = (g(\theta, x), f(\theta, x))$ be a map on $S^1 \times R$ where $\theta \in S^1 = [0, 2\pi)$ and $x \in R$. If a curve $r(\theta)$ is invariant under F, then we have

$$r(g(\theta, r(\theta))) = f(\theta, r(\theta)). \tag{1}$$

In principle, $r(\theta)$ can be derived by solving the equation. Numerically, we consider a discretization $\{(\theta_i, r(\theta_i))\}_{i=1}^N$ of the graph of $r(\theta)$. Denote the image of these points under F by $(\tilde{\theta}_i, \tilde{r}_i) = F(\theta_i, r(\theta_i))$. The points constitute a discretization of $\tilde{r}(\theta)$, the image of the graph of $r(\theta)$. We might consider the two functions to be "close" if $\tilde{r}(\theta_i)$ is near $r(\theta_i)$ for each θ_i, $i = 1, \dots, N$. Since we know only the information of $\tilde{r}(\theta)$ at $\theta = \tilde{\theta}_i$, $i = 1, \dots, N$, and, in general, the $\tilde{\theta}_i$ does not coincide with any of the θ_j, we must obtain an estimate $\rho(\theta_i)$ of $\tilde{r}(\theta_i)$ by interpolation (e.g., through splines). Kevrekids [4] solves (1) in the form

$$r(\theta_i) - \rho(\theta_i) = 0, \quad i = 1, 2, \dots, N, \quad \rho = \mathrm{Ipo}(\{(\tilde{\theta}_i, \tilde{r}_i)\}_{i=1}^N), \tag{2}$$

and Ipo$(\cdot)$ denotes the operator of interpolation. We notice that there are N independent nodes θ_i used in the system which are distributed along the entire $r(\theta)$, the system (2) is clearly big.

Kaas-Petersen [3] realizes that if the rotation number[3] of r, say ω_r, is not zero, i.e., $\omega_r > 0$, then, generically, any point on r will iterate dense image under F, and one can pick some moderate iteration points to interpolate the whole curve. With this concept, Kaas-Petersen solves (1) in the form

$$r(\theta^*) - \rho(\theta^*) = 0, \quad \rho = \mathrm{Ipo}(\{(\hat{\theta}_i, \hat{r}_i)\}_{i=1}^m), \qquad (\hat{\theta}_i, \hat{r}_i) = F^{S_i}(\theta^*, r(\theta^*)), \quad i = 1, 2, \dots, m. \tag{3}$$

As detailed in [3], the way to choose the numbers of iterations $\{S_i\}_{i=1}^m$ is, essentially, to make sure that all of $(\hat{\theta}_i, \hat{r}_i)$ are close enough to $(\theta^*, r(\theta^*))$ in order to produce a better interpolation ρ of r near $r(\theta^*)$, and the total amount of iterates needed to achieve this goal is much dependent on the rotation number of r. Here, there is only one generating angle node θ^* in the system, so the dimension of (3) is much smaller than that of (2). Yet, by observing, one may need a lot of iterates to determine $\{S_i\}_{i=1}^m$ before doing interpolation.

In our opinion, among those we cited above, the algorithms of (2) and (3) in [3,4] are most easily utilized and with simpler structures. For dynamical systems at a generic fixed parameter, the two algorithms work well in all stable, unstable and mixed cases. On the other hand, when the parameter varies in the progress of the continuation of branches of the quasi-periodic solutions, the rotation number of the corresponding invariant curve may change smoothly. If it meets the KAM condition,[4] then as indicated in [3], the continuing process of (3) will be interfered

[2] Variables of points that generate a curve by interpolation.

[3] See definition in [2].

[4] When the rotation number is close to a rational number with small numerator and denominator.

since the amount of iterates needed to recover the curve (i.e., the determination of $\{S_i\}_{i=1}^m$) may be beyond the computers' capacity, and the rounding error also becomes a big problem. Consequently, in doing the continuation, the algorithm of (2) seems more adequate. While the system (3) is much smaller in size, (2) has a much simpler structure (i.e., without the factor $\{S_i\}_{i=1}^m$). We try to reduce the size of (2), meanwhile to improve the structure of (3).

Recall the fundamental concept in [3] that, under the iteration of map F, the image of any point or segment on r will wind around the curve again and again. Let $\mathbf{y}$ be a point on r, $\mathbf{y} = (\theta_1, r(\theta_1))$, $F(\mathbf{y})$ means its image under F, i.e., $F(\mathbf{y}) = F(\theta_1, r(\theta_1))$. Denote the segment of graph of r between these two points $\mathbf{y}$ and $F(\mathbf{y})$ by $[\mathbf{y}, F(\mathbf{y})]$, and κ is the maximum integer less than or equal to $1/\omega_r$ where ω_r is the rotation number of r (for the case $\omega_r = 0$, see Section 3.3 for further argument). Note that

$$r = [\mathbf{y}, F(\mathbf{y})] \cup [F(\mathbf{y}), F^2(\mathbf{y})] \cup \cdots \cup [F^{\kappa}(\mathbf{y}), F^{\kappa+1}(\mathbf{y})], \tag{4}$$

where $[F^j(\mathbf{y}), F^{j+1}(\mathbf{y})]$ is the jth times of iteration image of $[\mathbf{y}, F(\mathbf{y})]$ under F. Since any portion of r can be reached by iterates of $[\mathbf{y}, F(\mathbf{y})]$, the segment $[\mathbf{y}, F(\mathbf{y})]$ might be regarded as a generator of the entire curve r. Note that the role of the point $r(\theta^*)$ in (3) has been replaced by the segment $[\mathbf{y}, F(\mathbf{y})]$, and the interpolation on the entire r in (2) is now substituted by interpolating only on this generator $[\mathbf{y}, F(\mathbf{y})]$ of r.

In R^2, we consider a discretization $\{(\theta_i, r(\theta_i))\}_{i=1}^{M+1}$ of the segment $[\mathbf{y}, F(\mathbf{y})]$ where $(\theta_{M+1}, r(\theta_{M+1})) = F(\theta_1, r(\theta_1))$. Denote the image of these points under F^{κ} and $F^{\kappa+1}$ by $(\bar{\theta}_i, \bar{r}_i) = F^{\kappa}(\theta_i, r(\theta_i))$, $i = 1, 2, \dots, M$, $(\bar{\theta}_{i+M}, \bar{r}_{i+M}) = F^{\kappa+1}(\theta_i, r(\theta_i))$, $i = 1, 2, \dots, M+1$. These $2M+1$ points constitute a discretization of $[F^{\kappa}(\mathbf{y}), F^{\kappa+2}(\mathbf{y})]$, and, as illustrated in Fig. 1, the angular interval between the initial and terminal points covers that of $[\mathbf{y}, F(\mathbf{y})]$, i.e., $[\bar{\theta}_1, \bar{\theta}_{2M+1}] \supset [\theta_1, \theta_{M+1}]$. Thus, we can obtain an interpolation function $\rho(\theta)$, $\theta \in [\theta_1, \theta_M]$, in $[F^{\kappa}(\mathbf{y}), F^{\kappa+2}(\mathbf{y})]$. Now, (1) is solved in a similar form

$$r(\theta_i) - \rho(\theta_i) = 0, \quad i = 1, 2, \dots, M, \quad \rho = \mathrm{Ipo}(\{(\bar{\theta}_i, \bar{r}_i)\}_{i=1}^{2M}). \tag{5}$$

Since $M \le N/(\kappa - 1)$, the dimension of the system (5) is much smaller than that of (2). A majority of those N nodes $r(\theta_i)$ in (2) has been substituted by iterates of the others according to the winding property (4), which also guarantees that the efficiency of (5) is at least as good as that of (2). On the other hand, a single node $r(\theta^*)$ in (3) has been replaced by $\{r(\theta_i)\}_{i=1}^M$ on $[\mathbf{y}, F(\mathbf{y})]$, a key point that the entire curve r now can be recovered in a finite

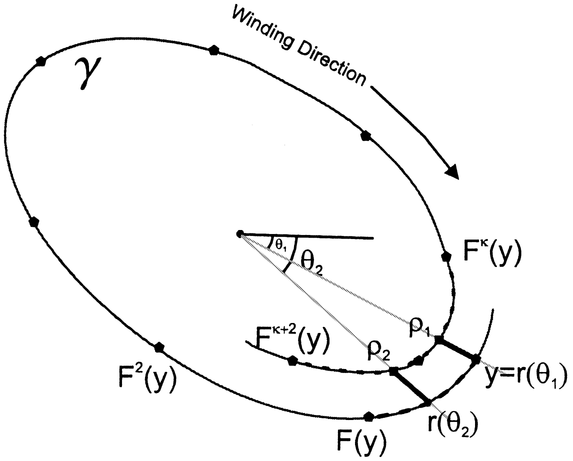

Fig. 1. The curve r is regarded invariant when the differences $r_i - \rho_i$ (the bold lines) at the discretization $\{\theta_i\}_{i=1}^M$ of $[\mathbf{y}, F(\mathbf{y})]$ (the dashed segment) are all zero. Here $M = 2$, for simplicity.

number of iterates regardless of the KAM condition, contrary to the uncertainty for the determination of $\{S_i\}_{i=1}^{m}$ in (3) when the KAM condition is present. In comparison with (3), the size of (5) is clearly larger, yet, in practical tracing for continuation of tori, the cost is well under control.

Notice that the above algorithms are all in R^2 and formulated in polar coordinates. Such polar forms impose a geometric constraint to the curve $r(\theta)$. In the algorithm of (2), $r(\theta)$ is not allowed to intersect more than once with the radial line $\theta = \theta_{\rm c}$ for each $\theta_{\rm c}$ in $[0, 2\pi)$. For the algorithm of (5), the requirement is relieved to be constrained only to the interval $\theta_{\rm c} \in [\theta_1, \theta_M]$, and the algorithm of (3) seems to have even less constraint. The geometry in R^2 under polar coordinates is clear and precise, so are those algorithms and the corresponding constraints. For the case in R^n, $n > 2$, the formulations of (2), (3) and (5) are parameterized in a single-angle coordinate θ, everything we mentioned now becomes ambiguous since the angle θ in R^n, $n > 2$, has to be well defined before anything can be carried on. To describe and apply the algorithms correctly without ambiguity, one must find a suitable origin and a two-dimensional plane H such that r and its projection $\pi(r)$ on H is 1–1 correspondent, in which case the curve r in R^n can be parameterized in terms of polar coordinates of $\pi(r)$ on H. Then everything follows just like that in R^2. Unfortunately, to carry out this entire transformation becomes a troublesome and unreliable job. Along with the problem of interpolation error (as detailed in the followings), these factors could possibly destroy the stability of the convergence process. Motivated by eliminating such problems brought by a single-angle coordinate θ, we now apply the arc length parameterization to reformulate our idea in R^n in (6), which indeed gives a solid improvement in the versatility of our algorithm in high-dimensional spaces, as seen in Section 3.2.

Suppose that $\{\mathbf{y}_i^*\}_{i=1}^{M+1}$ is a discretization of the segment $[\mathbf{y}_1^*, F(\mathbf{y}_1^*)]$ in an invariant curve r where $\mathbf{y}_{M+1}^* = F(\mathbf{y}_1^*)$. For $i = 1, \ldots, M$, H_i is a hyperplane in the Poincaré section which transverses to r at $\mathbf{y}_i^*$. In a neighborhood of $(\mathbf{y}_1^*, \mathbf{y}_2^*, \ldots, \mathbf{y}_M^*)$, say, $(N_\delta(\mathbf{y}_1^*) \cap H_1) \times (N_\delta(\mathbf{y}_2^*) \cap H_2) \times \cdots \times (N_\delta(\mathbf{y}_M^*) \cap H_M)$, we define a system of vector-valued functions $\{\mathbf{Q}_i(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M)\}_{i=1}^{M}$ such that

$$\begin{aligned} &\mathbf{Q}_1(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M) = \rho \cap H_1, \qquad \mathbf{Q}_2(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M) = \rho \cap H_2, \ldots, \\ &\mathbf{Q}_M(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M) = \rho \cap H_M, \quad \rho = \mathrm{Ipo}\left(\{F^\kappa(\mathbf{y}_i)\}_{i=1}^{M} \cup \{F^{\kappa+1}(\mathbf{y}_i)\}_{i=1}^{M}\right). \end{aligned} \tag{6}$$

Clearly, $\{\mathbf{y}_i^*\}_{i=1}^{M}$ is a fixed point solution of the system $\mathbf{Q}_i(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M) = \mathbf{y}_i$, $i = 1, \ldots, M$. For further use, we denote $\bar{\mathbf{y}} = \{\mathbf{y}_i\}_{i=1}^{M}$, $\bar{\mathbf{y}}^* = \{\mathbf{y}_i^*\}_{i=1}^{M}$, $\bar{\mathbf{Q}}(\bar{\mathbf{y}}) = \{\mathbf{Q}_1(\bar{\mathbf{y}}) - \mathbf{y}_1, \mathbf{Q}_2(\bar{\mathbf{y}}) - \mathbf{y}_2, \ldots, \mathbf{Q}_M(\bar{\mathbf{y}}) - \mathbf{y}_M\}$.

Another important factor which may interfere with the convergence process of (3), (5) and (6) is the interpolation error. It will be the best policy if a least yet sufficient amount of nodes in moderate intervals can be well chosen such that the system could correctly lock the true invariant curve (not a spurious one) for a small range of interpolation error. Here, we propose a simple direct criterion for such mesh adjustment (an idea which has often arisen in the course of error estimate for interpolation) as follows. In system (6), let $\vartheta = \mathrm{Ipo}(\{\mathbf{y}_i^*\}_{i=1}^{M+1})$ and $\mathbf{y}_i^* = \vartheta(\mu_i)$, $i = 1, \ldots, M+1$. Suppose that $F^\kappa(\mathbf{y}_i^*) = \rho(\tau_i)$, $i = 1, \ldots, M$, $F^{\kappa+1}(\mathbf{y}_i^*) = \rho(\tau_{i+M})$, $i = 1, \ldots, M+1$. Let $\bar{\tau}_j = \frac{1}{2}(\tau_j + \tau_{j+1})$, $j = 1, \ldots, 2M$ and ρ^+ be the curve interpolated from $\{F^\kappa(\mathbf{y}_i^*)\}_{i=1}^{M} \cup \{F^{\kappa+1}(\mathbf{y}_i^*)\}_{i=1}^{M+1}$ with splines one order higher than ρ. We determine whether or not to add or delete certain interpolating nodes by keeping the error of interpolation under prior given thresholds, i.e., we check the deviations between ρ and ρ^+,[5]

$$D_j = \max\{d(\rho(\bar{\tau}_j), \rho^+(\bar{\tau}_j)), d(\rho(\bar{\tau}_{j+M}), \rho^+(\bar{\tau}_{j+M}))\}, \quad j = 1, \ldots, M.$$

If D_j exceeds the maximum threshold, then a new node with initial position $\vartheta(\frac{1}{2}(\mu_j + \mu_{j+1}))$ is added to the interpolating node set $\bar{\mathbf{y}}$. If D_j and D_{j+1} are both below the minimum threshold, then the node $\mathbf{y}_j$ is deleted from $\bar{\mathbf{y}}$. The whole scheme is performed in Section 3.2 and the outcome is quite satisfactory.

[5] $d(x, y)$ means the distance between x and y.

In doing continuation of the entire branch of the quasi-periodic solutions, the stage from parameter value α_k to the next value α_{k+1} proceeds as follows. Suppose that, at the kth step, a solution $\bar{\mathbf{y}} = \bar{\mathbf{y}}_k^*$ of $\bar{\mathbf{Q}}(\bar{\mathbf{y}}, \alpha_k) = \mathbf{0}$ in system (6) is derived (which is understood to represent one invariant curve at the parameter value α_k), then we evaluate the deviations $\{D_j\}_{j=1}^M$ first; if some D_j exceeds the given threshold, then a refinement for the interpolating node set $\bar{\mathbf{y}}$ must be done before passing to the next step for the value α_{k+1}. The refinement intends to keep the error of interpolation under control to assure that the continuation process proceeds uninterruptedly, and is derived as follows: (i) according to our mesh adaptation criterion, add or delete some nodes in $\bar{\mathbf{y}}$ to create a new set, say $\tilde{\mathbf{y}}$, with its values $\tilde{\mathbf{y}}_k^*$; (ii) construct new hyperplanes $\tilde{H}_i$, $i = 1, \dots,$ which must properly transverse the interpolated curve $\mathrm{Ipo}(\tilde{\mathbf{y}}_k^*)$; (iii) correct the new discretization $\tilde{\mathbf{y}}_k^*$ by one additional Newton's method for the root of $\tilde{\mathbf{Q}}(\tilde{\mathbf{y}}, \alpha_k) = \mathbf{0}$; (iv) repeat the above process (i)–(iii) and stop at the last refined discretization, say $\tilde{\mathbf{y}}_k^{**}$, where all deviations D_j, $j = 1, \dots,$ are now between the prior given maximum and minimum thresholds. Then, for the $(k+1)$th step, a solution at α_{k+1} will be derived with the initial guess $\ddot{\mathbf{y}}_{k+1}$ in

$$\ddot{\mathbf{y}}_{k+1} = \tilde{\mathbf{y}}_k^{**} + h\left[\frac{\mathrm{d}\tilde{\mathbf{Q}}(\tilde{\mathbf{y}}_k^{**}, \alpha_k)}{\mathrm{d}\tilde{\mathbf{y}}}\right]^{-1}\left(-\frac{\mathrm{d}\tilde{\mathbf{Q}}(\tilde{\mathbf{y}}_k^{**}, \alpha_k)}{\mathrm{d}\alpha}\right), \qquad (7)$$

where $h = \alpha_{k+1} - \alpha_k$ is the step size. This formula (7) is part of the standard algorithm of continuation method for bifurcation diagrams.

3. Numerical examples

For those models which have been investigated by others, we redo and summarize all the results with comparisons. A brief demo program of the algorithm of (5) is given in Appendix A.

3.1. A map

Consider the map (7) in [4]:

$$\theta_{n+1} = \theta_n + \omega, \qquad x_{n+1} = \lambda x_n(1 - x_n) + \varepsilon\cos\theta_n,$$

where λ, ε and ω are parameters. For this model, we make a comparison between the algorithms of (2) and (5). For stability analysis, algorithm of (3) works well (since the rotation number is fixed at $\omega_r = \omega/2\pi$, not a KAM condition), and the eigenvalues of its Jacobians can be calculated.

3.1.1. Case 1: $\lambda = 3.46$, $\omega = \pi(\sqrt{5} - 1)$

As evaluated in Fig. 4 of [4], there are two attracting period 2 and one unstable period 1 invariant curves. The unstable case is nontrivial, so we test the case with ε as the varying parameter.

As in Fig. 2 and in Table 1, 160 evenly distributed nodes for discretization have been used for the computation [6] in [4]. Here, for $\kappa = 4$, it needs no more than 55 nodes (in fact, only 20 nodes) to get results of the same level in the computation of (5).

3.1.2. Case 2: $\lambda = 2.5$, $\omega = 0.5$

There are only attracting curves and stability analysis is thus trivial. Compared to Fig. 6 in [4], our algorithm of (5) still works even that curve 8 is highly jagged as depicted in Figs. 3 and 4 and in Table 2.

[6] See the caption in Fig. 4 of [4].

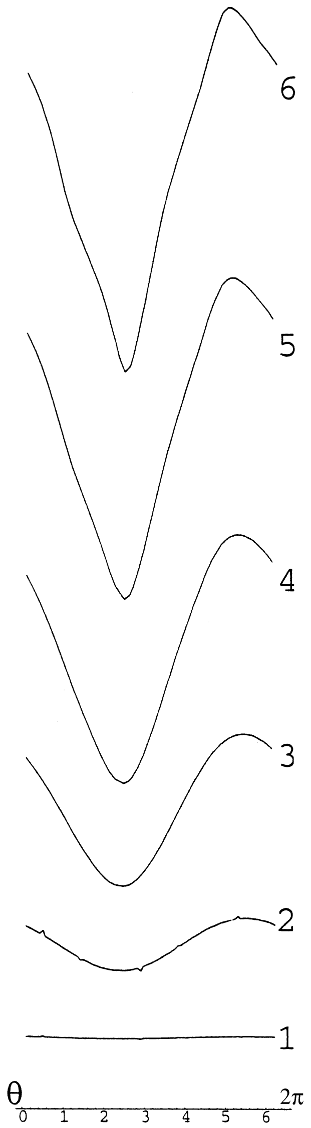

Fig. 2. Invariant curves 1–6 of Case 1 plotted for $\theta \in (0, 2\pi)$ on the x-axis.

Table 1
Data for each invariant curve in Fig. 2

No.	ε	M nodes	$r(0)$	Eigenvalue of [2]
1	0.001	20	0.7117199	−464872.32
2	0.022	20	0.7266202	−374527.30
3	0.062	20	0.7520316	−78200.64
4	0.094	20	0.7693318	−1357.00
5	0.110	20	0.7764683	−34171.94
6	0.115	20	0.7780265	−540.46

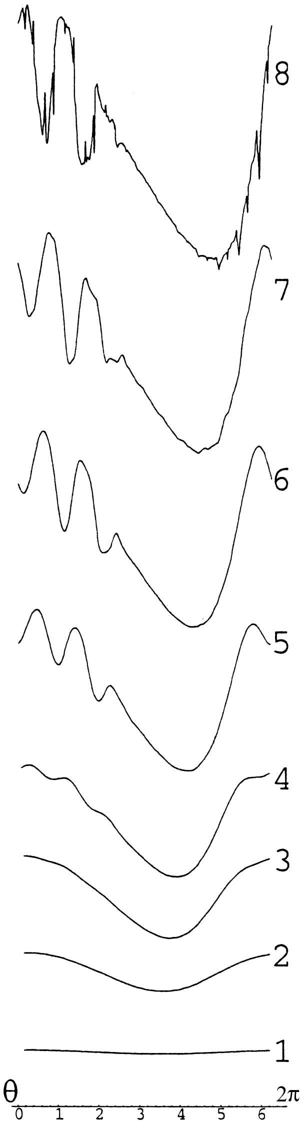

Fig. 3. Invariant curves 1–8 of Case 2 plotted for $\theta \in (0, 2\pi)$ on the x-axis.

3.2. The 2-mode truncated sine-Gordon ODE

For the N-mode truncated damped, driven sine-Gordon ODE (see a brief introduction in Appendix B), the constructions of the Poincaré maps consume heavy computing, and the computers' capacity will be eventually incapable of doing further work such as approaching invariant curves. Indeed, the initial motivation to develop our algorithm is trying to identify and continue the invariant sets for the entire branch. In this example, we apply algorithm of (6) to demonstrate its feasibility and efficiency, and, at the same time, test our criterion for mesh adjustment.

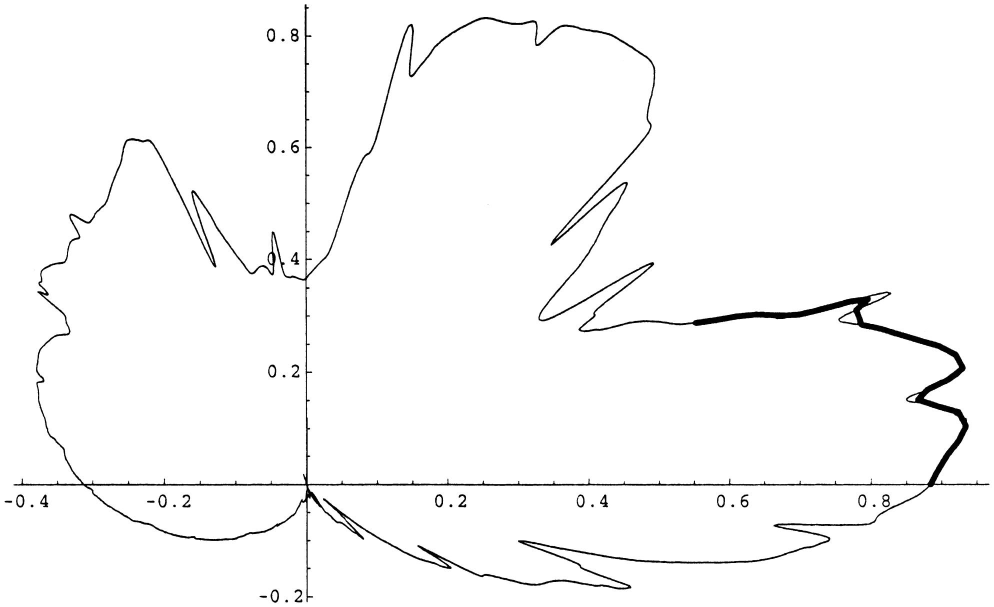

Fig. 4. The exact detailed graph of the curve 8 in Fig. 3 in 1500 iterates in (x, y)-coordinate. The bold line is the part actually interpolated in our program (the interpolation error can be seen).

For $N = 2$, (B.1) yields that

$$\begin{aligned}\ddot{a}_0 + 0.04\dot{a}_0 &= -\sqrt{12}J_0\left(\frac{a_1}{\sqrt{6}}\right)\sin\left(\frac{a_0}{\sqrt{12}}\right) - \Gamma\sqrt{12}\cos(0.87t),\\ \ddot{a}_1 + 0.04\dot{a}_1 &= -\left(\frac{2\pi}{12}\right)^2 a_1 - 2\sqrt{6}J_1\left(\frac{a_1}{\sqrt{6}}\right)\cos\left(\frac{a_0}{\sqrt{12}}\right),\end{aligned}\qquad (8)$$

where Γ is the parameter, and $J_0,\ J_1$ are the Bessel functions. Here, for conventional reason (e.g., see [1]), $\omega = 0.87$, $\alpha = 0.04$, $L = \sqrt{12}$. Let $b_0(t) = \dot{a}_0(t)$, $b_1(t) = \dot{a}_1(t)$, then (8) is reduced to a system of first-order

Table 2
Data for each invariant curve in Fig. 3

No.	ε	M nodes	$r(0)$
1	0.01	5	0.6064263
2	0.10	12	0.6601657
3	0.20	12	0.7133255
4	0.25	13	0.7338592
5	0.30	15	0.6989777
6	0.32	16	0.6991233
7	0.33	17	0.8069227
8	0.341875	27	0.8844668

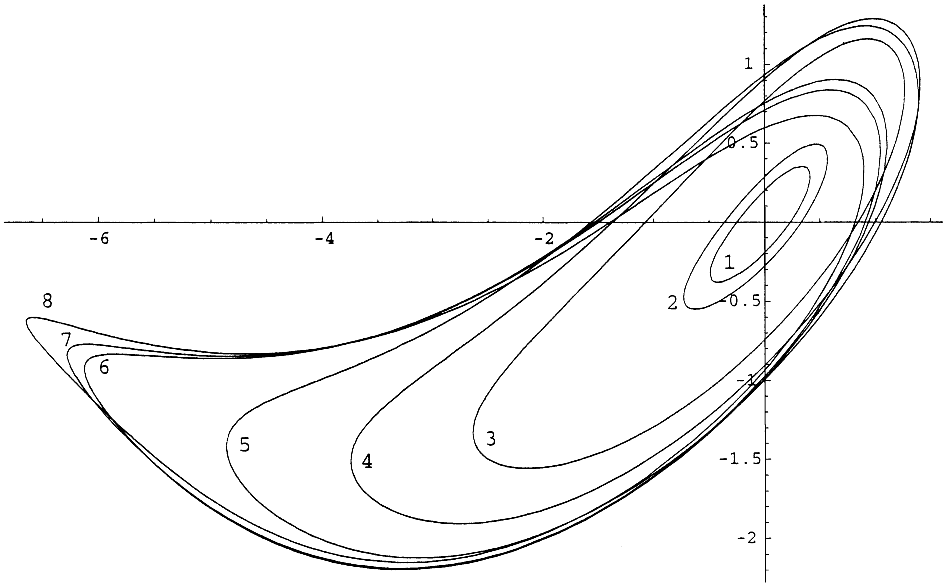

Fig. 5. Projection of invariant curves on the v_1, v_2-plane (horizontal: v_1-direction; vertical: v_2-direction), where $v_1 = (1, 0, 1, 0)$, $v_2 = (0, 1, 0, 1)$ and the origin is (3.0384, −2.4048, 1.8238, −1.0612). Curve 2 is on a degenerate torus composed of periodic flows.

non-autonomous ODEs,

$$\begin{bmatrix} \dot{a}_0 \\ \dot{b}_0 \\ \dot{a}_1 \\ \dot{b}_1 \end{bmatrix} = \begin{bmatrix} b_0 \\ -0.04b_0 - \sqrt{12}J_0\left(\dfrac{a_1}{\sqrt{6}}\right)\sin\left(\dfrac{a_0}{\sqrt{12}}\right) - \Gamma\sqrt{12}\cos(0.87t) \\ b_1 \\ -0.04b_1 - \left(\dfrac{2\pi}{12}\right)^2 a_1 - 2\sqrt{6}J_1\left(\dfrac{a_1}{\sqrt{6}}\right)\cos\left(\dfrac{a_0}{\sqrt{12}}\right) \end{bmatrix}.$$

We found a branch of invariant tori (i.e., quasi-periodic solutions) bifurcating from a periodic orbit at $\Gamma = 0.0781873$ (Hopf bifurcation). As depicted in Fig. 5 and in Table 3, a Poincaré section is set up at $t = 0$ with its origin transversed by the periodic orbit, and invariant tori are reduced to invariant curves.

For this model, the rotation number increases with decreasing Γ, so there are degenerate tori mixed through the entire branch. As pointed out early, since each one of them corresponds to a KAM condition, the continuation process of system (3) will be interrupted (of course, its stability analysis is no more valid). On the other hand, after

Table 3
A list of points on each of the invariant curves in Fig. 5

No.	Coordinates of $\mathbf{y}_1$ solved in system (6)
1	(3.00683697821019, −2.43138619315944, 2.03467601898394, −1.0915413683294)
2	(2.99576505316505, −2.43498059283767, 2.12181203847718, −1.1066790407384)
3	(3.03050303738719, −2.24295522350729, 2.66438056361979, −1.2396269543157)
4	(3.04822764315913, −2.12560947926463, 2.82216435886872, −1.3305490554311)
5	(3.03475790720348, −2.05256838707615, 2.86542464997452, −1.4388124240403)
6	(2.65216751991587, −2.42984790830223, 2.68526012034587, −1.6141144154183)
7	(2.64826996290989, −2.42040072083200, 2.66294858309643, −1.6285777516285)
8	(2.64010671044121, −2.40035773563487, 2.60384223466678, −1.6565491413303)

Table 4
Data for some invariant curves including those in Fig. 5

No.	Γ	κ	M_1	M_2	Stability of map $F^{\kappa+1}$ at $\mathbf{y}_1$
1	0.0778100	6	5	5	{1.041, 1.0024, 0.132, 0.12721}
2	0.0774334	7	8	7	{0.645 ± 1.01i, 0.044 ± 0.070i}
3	0.0720600	7	18	10	{2.33, 0.95, 0.10433, 0.042551}
4	0.0680600	8	21	11	{4.71, 1.05, 0.07071, 0.015773}
5	0.0636600	9	23	12	{5.67, 1.005, 0.0553, 0.009849}
	0.0616361	10	23	11	{10.80, 1.04, 0.0401, 0.003826}
	0.0602987	11	20	11	{25.38, 1.20, 0.0260, 0.001207}
	0.0590661	12	20	11	{37.47, 1.15, 0.0202, 0.000630}
6	0.0582711	13	20	11	{76.09, 1.28, 0.0136, 0.000199}
7	0.0576321	14	–	11	{138.8, 1.34, 0.0097, −0.00022}
	0.0571165	15	–	11	{222.0, 1.30, 0.0074, −0.00103}
	0.0567745	16	–	11	{518.6, 1.63, 0.0043, −0.00317}
	0.0564584	17	–	11	{883.6, 1.64, −0.0122, 0.00330}
8	0.0562991	18	–	11	{2490.1, 2.56, −0.020, 0.00151}

$\kappa \geq 10$, system (2) needs at least 100 nodes to obtain the same results as we have done, and the dimension of its Jacobian is much larger than ours.

In Table 4, we give a brief comparison of our programs, one is of the type (5) with M_1 automatically evenly spaced nodes, the other one is of the type (6) with M_2 nodes automatically determined by our mesh adjustment criterion. By observing, after $\Gamma < 0.0774334$, where $\kappa = 7$, M_1 blows up rapidly due to the inflation of the invariant curves; on the other hand, M_2 amount increases slowly and tends to a constant, which does depend only on the complexity of the geometry of the interpolation for curves. Note that, after $\Gamma < 0.0602987$, curves 6–8 lie close to each other, that means the dynamics of this model begins to change violently in this region of parameters (as reflected from the stability of map $F^{\kappa+1}$). System (5) with M_1 fails after $\Gamma < 0.0582711$ where $\kappa = 13$; however, system (6) with M_2 proceeds stably to at least $\Gamma = 0.0562991$ where $\kappa = 18$. In summary, (6) with M_2 under the mesh adjustment criterion is less interfered by the behaviors of the dynamics and works better in the sensitive region of parameters; moreover, the above results show that a proper distribution of nodes is much more important than their amount.

3.3. *Forced van der Pol oscillator*

$$\dot{x} = y - \alpha(\tfrac{1}{3}x^3 - x), \qquad \dot{y} = -x + \lambda \cos \omega t.$$

This is an example that cannot be accomplished by our original algorithm of (5) under the interpolation of a segment. Fig. 6 is for $\alpha = 0.4$, $\omega = \sqrt{0.84}$, and λ are given by 0.3418750, 0.3350000, 0.2798046 (in which cases, rotation numbers of tori are all 0). On a Poincaré section, each invariant curve (reduced from the corresponding invariant torus) contains a saddle and a sink, iterates of any points on the curve will go away from the saddle and forward to the sink, and obviously none of iterations of any point has dense image in the entire curve. Since the algorithms of (3) and (5) are founded on the winding property (4) (i.e., iterates of any point or segment on the curve dense in the entire curve), it is natural that both will be improper and fail in this example. However, a simple modification of (5) will do. Now (4) is replaced by

$$[\mathbf{y}_1, \mathbf{y}_M] \subseteq [F(\mathbf{y}_1), F(\mathbf{y}_M)] \subseteq [F^2(\mathbf{y}_1), F^2(\mathbf{y}_M)] \subseteq \cdots .$$

The key point is that the segment $[\mathbf{y}_1, \mathbf{y}_M]$ is chosen to contain the saddle. Let $\{\mathbf{y}_i\}_{i=1}^{M}$ be a discretization on this segment $[\mathbf{y}_1, \mathbf{y}_M]$, and ρ be the interpolation function in $[F(\mathbf{y}_1), F(\mathbf{y}_M)]$ where $\rho = \mathrm{Ipo}(\{F(\mathbf{y}_i)\}_{i=1}^{M})$. Clearly, the angular interval of $[F(\mathbf{y}_1), F(\mathbf{y}_M)]$ covers that of $[\mathbf{y}_1, \mathbf{y}_M]$, and the functions $\{r(\theta_i) - \rho(\theta_i)\}_{i=1}^{M}$ are therefore well defined as before, and hence the modified algorithm is applicable to models in $\omega_r = 0$ such as this one.

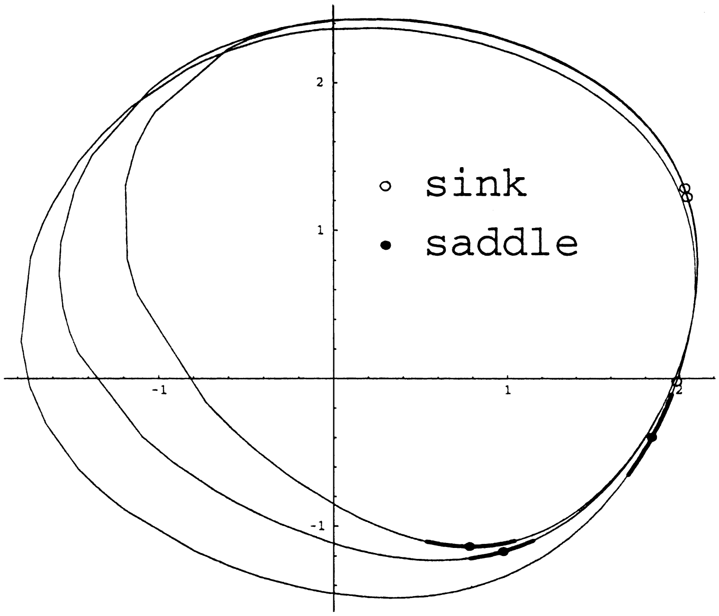

Fig. 6. Smaller curves correspond to larger λ, the bold segments are actually interpolated in our program.

This famous example has been studied in [5,7]. Here, we just emphasize that the idea for interpolation on a segment seems adequate for variety of problems.

4. Conclusion

In cooperation with Newton's method, our algorithm has shown that it is capable and efficient in identifying invariant torus for each fixed parameter and smoothly carry out the continuation of the entire branch of quasi-periodic solutions in an interval of parameters as demonstrated in the SG ODE for $N = 2$ and other examples above. In summary, our algorithm essentially includes the two ingredients: the main algorithm of (5) and (6) in two different representations and the mesh adaptation criterion (for interpolating nodes). In comparison with (2), our system has a much smaller size; in comparison with (3), our system has a much simpler and better structure. We should point out that the criterion for mesh adaptation plays more than one role in our entire scheme. First, it helps adjust the amount of the nodes to keep the system as small as possible. Secondly, in doing continuation, it helps make a smooth transition in an interval of parameters with less interruption. To explain, we claim that the criterion can verify the accuracy of the solutions derived through the techniques of interpolation, and in addition, it is capable to promote them to higher precision, which is indeed the key for smooth transitions of the continuation. This can be seen when one evaluates $\bar{\mathbf{Q}}$ in (6). According to the rule, the value of each node is iterated (under F) for $\kappa + 1$ times, and it is possible that, for models with violent dynamics, the iterated discretization $\{F^{\kappa}(\mathbf{y}_i)\}_{i=1}^{M} \cup \{F^{\kappa+1}(\mathbf{y}_i)\}_{i=1}^{M}$ may blow up, and, consequently, the induced interpolation ρ is generally erratic (e.g., it did occur in the SG case). Trying to avoid such a possibility, one must do precise evaluation for the initial guess (7) in each step of the continuation process, and this can only be done in high precision of $\tilde{\mathbf{y}}^{**}$. In fact, this was the origin that enforced us to design the criterion, and it indeed does the job (i.e., promote $\tilde{\mathbf{y}}^{**}$ to higher precision) as numerical evidences already showed in the proceeding of system (6) in Section 3.2.

In developing a bifurcation diagram, it becomes particularly clear that stability analysis for solutions of the same degree of freedom allows one, in principle, to approach solutions of one more degree of freedom. For example, stability analysis for the periodic solutions makes it possible for one to approach the dynamics of the quasi-periodic flows (or, equivalently, T^2-invariant tori) starting from a Hopf bifurcation in some branch of periodic solutions. Naturally, from this aspect, one might be able to approach and investigate much more complicated high-dimensional tori if one can do stability analysis for T^2-invariant tori, which is still open. We have tried and designed many numerical experiments for our algorithm for analyzing the stability of T^2-invariant tori, though not much has been achieved, we believe that it is important and still put a lot of effort in this subject.

It will be unrealistic for any attempt to develop a perfect algorithm. In our view, a good algorithm should be very versatile in the manner that it can do the generic problems efficiently, can be easily modified to apply to the special cases, and can contact with and help improve the investigations of other related subjects. Currently, we are continuing the improvement of our algorithms in two directions, one is associated with the stability analysis as mentioned above, the other one is doing numerical analysis of SG ODE for $N > 2$. We believe that there are unexpected difficulties ahead both theoretically and numerically, yet it is worth to go on and try to develop a better algorithm since, then, one might easily realize some degree of the complexity of the general dynamics.

Appendix A. Program for evaluating invariant curves

The following is a sample program written in Mathematica for the algorithm of (5):

```
external functions:
    FFF: a defined map
    ANGLE: angle in polar coordinates
global variables:
    DomainDimension: dimension of the domain of FFF
    windingCYCLE: the number κ in (5)
INTERPOLATE[pointSET_List]:=
Module[{i, j, pointNUMBER = Dimensions[pointSET][[1]]},
  Do[
    tempFUNC[i] = Interpolation[
            Table[{pointSET[[j]][[−1]],pointSET[[j]][[i]]},{j,1,pointNUMBER}]
        ]
  ,{i,1,DomainDimension}];
  interpolatedFUNC[angle_] := Table[tempFUNC[i][angle],{i,1,DomainDimension}];
];
quasiFUNC[pointSET_List,delta_]:=
Module[{i,j,temp,coordinates,sectionNUMBER = Dimensions[pointSET][[1]],
      iteratedSECTION1 = {}, iteratedSECTION2 = {}},
  coordinates = Table[ANGLE[pointSET[[j]]],{j,1,sectionNUMBER}];
  Do[
    temp = pointSET[[i]];
    Do[temp = FFF[temp, delta], {j, 1, windingCYCLE}];
    iteratedSECTION1 = Append[iteratedSECTION1,Append[temp,ANGLE[temp]]];
    temp = FFF[temp,delta];
    iteratedSECTION2 = Append[iteratedSECTION2,Append[temp,ANGLE[temp]]]
  ,{i,1,sectionNUMBER}];
  INTERPOLATE[Join[iteratedSECTION1,iteratedSECTION2]];
  Table[pointSET[[j]]−interpolatedFUNC[coordinates[[j]]],{j,1,sectionNUMBER}]
]
```

Appendix B. Derivation of the truncated N-mode SG ODE

The damped, driven SG PDE is a well-known nearly integrable system (e.g. [9]) given as

$$u_{xx} - u_{tt} = \sin u + \alpha u_t + \Gamma \cos(\omega t), \qquad u(x+L,t) = u(x,t).$$

The traditional Galerkin method [10] reduces the damped, driven SG PDE to the N-mode truncated SG ODE. To briefly summarize, suppose that $e_j(X) = C_j \cos(2j\pi X/L)$ on $[-\frac{1}{2}L, \frac{1}{2}L]$, $j = 1, 2, 3, \dots$, with the ansatz

$$u(x,t) = \sum_{j=0}^{N-1} a_j(t) e_j(x).$$

Substituting it into the damped, driven SG PDE yields that

$$\ddot{a}_j + \alpha \dot{a}_j = a_j \langle e_j'', e_j \rangle - \left\langle \sin\left(\sum_{i=0}^{N-1} a_i e_i\right), e_j \right\rangle - \Gamma \cos(\omega t)\langle 1, e_j \rangle, \quad j = 0, 1, \dots, N-1. \tag{B.1}$$

References

[1] C. Xiong, Low mode truncation method, Ph.D. Thesis, Ohio State University, Columbus, OH, 1991.

[2] J. Guckenheimer, P. Holmes, Nonlinear Oscillations in Dynamical Systems and Bifurcations in Vector Fields, Applied Mathematical Sciences, Vol. 42, Springer, New York, 1993.

[3] Chr. Kaas-Petersen, Computation, continuation, and bifurcation of torus solution for dissipative maps and ordinary differential equations, Physica D 25 (1987) 288–306.

[4] I.G. Kevrekidis, R. Aris, L.D. Schmidt, S. Pelikan, Numerical computation of invariant circles of maps, Physica D 16 (1985) 243–251.

[5] M. Van Veldhuizen, A new algorithm for the numerical approximation of an invariant curve, SIAM J. Sci. Statist. Comput. 8 (1987) 951–962.

[6] M. Van Veldhuizen, Convergence results for invariant curve algorithms, Math. Comput. 51 (1988) 677–697.

[7] L. Dieci, J. Lorenze, Computation of invariant tori by the method of characteristics, SIAM J. Numer. Anal. 32 (1995) 1437–1474.

[8] L. Dieci, J. Lorenze, R.D. Russell, Numerical calculation of invariant tori, SIAM J. Sci. Statist. Comput. 12 (1991) 607–647.

[9] D.W. McLaughlin, Whiskered Tori for Integrable PDEs: Chaotic Behavior in Near Integrable PDEs, Surveys in Applied Mathematics, Vol. 1, Plenum Press, New York, 1995, pp. 83–203.

[10] C.A.J. Fletcher, Computational Galerkin Methods, Springer, New York, 1984.