Journal of Computational and Applied Mathematics 21 (1988) 203–212
North-Holland

On the numerical approximation of the rotation number

M. VAN VELDHUIZEN
Department of Mathematics and Computer Science, Vrije Universiteit, P.O. Box 7161, Amsterdam

Received 21 January 1987
Revised 16 June 1987

Abstract: The starting point of this paper is a polygonal approximation of an invariant curve of a map. Using this polygonal approximation an approximation for the circle map (the restriction of the map to the invariant curve) is obtained. The rotation number of the circle map is then approximated by the rotation number of the approximated circle map. The error in the obtained approximate rotation number is discussed, and related to the error in the polygonal approximation of the invariant curve. Simple algorithms for the approximation of the rotation number are described. A numerical example illustrates the theory.

Keywords: Rotation numbers, circle map.

1. Introduction

In this paper we consider a homeomorphism from the circle S^1 onto itself. For such maps the rotation number ρ exists, cf. [4]. We want to approximate the rotation number of the map by means of an approximation of the circle map

The circle map is usually approximated by approximating an invariant curve γ of a map Φ from $\mathbb{R}^d$ to $\mathbb{R}^d$. See [2], [5] and [10] for algorithms for the approximation of an invariant curve. We shall assume that the approximation of the invariant curve γ of the map Φ is a polygon. This polygon with vertices x_i is denoted by $\boldsymbol{p}(\{x_i\}_{i=1}^N)$. The algorithm in [10](explained briefly in Section 2) results in a polygonal approximation. So does a slight modification of the algorithm in [5]. See [11] for a full description of this modification and a convergence result.

The Fourier methods, B-spline methods described by Chan [2] may fail if the rotation number is not highly irrational. The first method described by Chan [2] looks for an invariant curve $\gamma: t \to u(t)$ determined by the equation $\Phi(u(t)) = u(t+\rho)$. Here u is a truncated Fourier series or a periodic spline, and the equations are solved in a collocation sense. An advantage of this idea is that the invariant curve γ and the rotation number ρ are approximated simultaneously. However, for purely rational rotation numbers the action of Φ along γ can usually not be described as a rigid rotation. It is not surprising that Chan [2] reports difficulties close to simple rational rotation number. The second method described by Chan [2] tries to overcome this difficulty by starting from the equation $\Phi(u(t)) = u(\sigma(t)) \in \gamma$. Again u is a truncated Fourier series. This equation is solved in a collocation sense in collocation points t_i, after adding to it

additional equations which ensure a regular distribution of the points $u(t_i)$ along the curve to be determined. This method does not determine the rotation number directly. Upon convergence—there is no convergence proof—the method defines an approximate circle map. However, it is not clear from the definition of the method whether this approximate circle map is a homeomorphism. That is not a good basis for the approximation of a rotation number. In view of these arguments we confine the discussion to approximate circle maps obtained by a polygonal approximation. As we shall see, such approximate circle maps are homeomortphisms.

In Section 2 below we outline briefly the algorithm described in [10]. Then, on the basis of a polygonal approximation for the invariant curve γ, we define an approximation for the circle map $\Phi|_\gamma$. We define an approximate rotation number as the rotation number of the approximate circle map. Finally, we discuss the error in the approximate rotation number.

In Section 3 we give some algorithms for the approximation of a rotation number for an explicitly given circle map.

In Section 4 some numerical examples are presented. In these examples we focus on the delayed logistic map studied in great detail in [1]. The polygonal approximation is obtained by the algorithm described in [10].

2. The approximation of the rotation number

In this Section we discuss the approximation of the rotation number of a circle map. We give an algorithm based on a polygonal approximation. An estimate for the error in the approximated rotation number is given.

As an introduction to the main topic of this Section, let Φ be a map from $\mathbb{R}^d$ to $\mathbb{R}^d$. We assume that there exists a Jordan curve γ such that $\Phi\gamma = \gamma$. I.e., γ is an invariant curve of the map Φ. In addition we assume that the curve γ is attracting. This means that in the neighborhood of γ the image Φx is closer to γ than x itself. By this assumption we exclude a saddle-connection. Invariant curves play a part in the description and analysis of dynamical systems, ordinary differential equations with oscillatory solutions, see [3].

A simple polygonal algorithm for the approximation of an attracting invariant curve is given in [10]. This algorithm approximates γ by a polygon $\boldsymbol{p}(\{x_i\}_{i=1}^N)$, with vertices determined by the equation $\boldsymbol{p}(\{x_i\}_{i=1}^n) = \boldsymbol{p}(\{Hx_i\}_{i=1}^N)$. This equation is solved by iteration (seccessive substitution). The nonlinear map H is defined as follows. Compute the images Φx_i for all the vertices x_i. Assume that all the vertices and their images are located along the curve in successive order. Locally, approximate the curve γ by the circumscribed circle of three successive images. Any vertex x_i is a point on a radius vector of at least one of these circumscribed circles. One chooses the circle determined by the successive images Φx_{j-1}, Φx_j and Φx_{j+1} if $\|x_i - \Phi x_l\|$ is minimized by Φx_j. Define Hx_i as the intersection of the chosen radius vector and a line segment of the polygon of the images $\boldsymbol{p}(\{\Phi x_i\}_{i=1}^N)$. This defines the map H. In fact, H is an adaptation of Φ to the polygon. In essence, this describes the algorithm for the approximation of an invariant curve given in [10]. However, see [10] for many relevant details and a more complete description.

A related algorithm very close to the one in [5] is discussed in [11]. We shall not describe it here. We only observe that this algorithm might serve as a basis of the discussion in this section as well.

Suppose $\boldsymbol{p}(\{x_i\}_{i=1}^N)$ is a polygonal approximation to γ obtained by the algorithm indicated above. I.e., $p(\{x_i\}_{i=1}^N) = \boldsymbol{p}(\{Hx_i\}_{i=1}^N)$, and thus

$$x_i = \alpha \Phi x_j + (\alpha - 1)\Phi x_{j+1}, \quad \alpha \in [0, 1) \tag{2.1}$$

for some index j depending on i. We need to count the indices correctly. To that end we assume that the circle map $\Phi|_\gamma$ is increasing (orientation preserving). The i-indices are counted $1, \ldots, N$. Let x_1 be mapped onto $[\Phi x_{j_0}, \Phi x_{j_0+1})$. Then the j-indices are counted as

$$j_0,\ j_0 + 1, \ldots, N,\ N+1,\ N+2, \ldots, N + j_0 - 1.$$

Making use of this ordering we may now define a discrete approximation ϕ_N of the inverse of the circle map $\Phi|_\gamma$ based on the polygonal approximation $\boldsymbol{p}(\{x_i\}_{i=1}^N)$ by, cf. (2.1)

$$\phi_N\left(\frac{i-1}{N}\right) = \frac{j + \alpha - 1}{N}, \quad i = 1, 2, \ldots, N \tag{2.2}$$

and ϕ_N is extended to a map on [0,1) by piecewise linear interpolation. Observe $\phi_N(1) = \phi_N(0) + 1$ by definition.

The approximation ϕ_N, or rather its lifting from the circle to the real line should be strictly monotone. If this is not so then the rotation number of ϕ_N is generally not defined. Since the rotation number is a rather fundamental characteristic number for a circle map, cf. [4], [3], it is hard to imagine how $\boldsymbol{p}(\{x_i\}_{i=1}^N)$ can be a good approximation of γ without ϕ_N being strictly monotone. The algorithm described in [10] assures monotonicity, because it breaks down without it. Upon convergence, the (modification of the) algorithm of Kevrekidis et al. [5] does not guarantee monotonicity, but monotonicity is easily checked. Henceforth we restrict ourselves to a polygonal approximation with monotone map ϕ_N. In this context one should note the importance of piecewise *linear* interpolation as a monotonicity preserving interpolation scheme. For quadratic spline interpolation the preservation of monotonicity may require additional abscissae, cf. [7]. For most interpolation schemes the preservation of monotonicity is an unsolved problem.

The circle map ϕ_N defines a rotation number ρ_N. This rotation number may be approximated by iterating the map ϕ_N. See Section 3 for a few algorithms. No evaluations of Φ are involved in these computations. One only needs the correspondence (2.2) defined by the polygonal approximation $\boldsymbol{p}(\{x_i\}_{i=1}^N)$. This correspondence is computed once and stored in an array. The evaluation of ϕ_N then uses this array rather than Φ and H.

In this Section we now discuss the the difference $\rho - \rho_N$, where ρ, ρ_N are the rotation numbers of $\Phi|_\gamma$ and ϕ_N respectively. To that end we need an estimate for the difference of $\Phi|_\gamma$ and its approximation ϕ_N. The difference between ϕ_N and $\Phi|_\gamma$ consists of two components. In order to describe these components we need a suitable coordinate on the curve γ. As in [11] we assume that in an annular neighborhood of γ the nonlinear coordinate transformation (tubular coordinates)

$$x \in \mathbb{R}^2 \to u(\theta) + Z(\theta)\xi$$

is well defined. See [4, Chapter VI] for such coordinates. We assume $\theta \in [0, 1)$ rather than $[0, 2\pi)$. The curve γ is described by $\theta \to u(\theta)$ and the vector $Z(\theta)$ is a normal vector to γ in the point $u(\theta)$. Let the polygon $\boldsymbol{p}(\{x_i\}_{i=1}^N)$ belong to this annular neighborhood. Then, by a smooth homeomorphism $\theta \to \tau$ we may assume that

$$x_i = u\left(\frac{i-1}{N}\right) + Z\left(\frac{i-1}{N}\right)\xi_i.$$

This coordinate system (τ, ξ) is suitable in view of (2.2). The components in the difference $\Phi|_\gamma - \phi_N$ are now easily identified. First ϕ_N is defined by means of a polygon approximating the curve γ. Then one error component is determined by the difference between Φx_i and $u((i-1)/N)$. In fact, only the difference between the τ-coordinates enters in the error between $\Phi|_\gamma$ and ϕ_N. The second error component is determined by the linear interpolation, cf. (2.1), (2.2). This error is quadratic in $1/N$. So the difference between the circle map $\Phi|_\gamma$ and its approximation ϕ_N is determined by the approximation error in $\boldsymbol{p}(\{x_i\}_{i=1}^N)$ and an interpolation error $O(N^{-2})$.

Now we may turn to the error in ρ_N. As a first estimate a lemma in [2, p.297] should be mentioned. It says that the rotation number depends continuously on the circle map. Hence, $\rho_N \to \rho$ if $\max|\phi_N - \Phi|_\gamma| \to 0$. Here we make a simplifying assumption. In many practical instances the circle map $\Phi|_\gamma$ is homeomorphic to a rigid rotation. Often the homeomorphism is smooth, i.e. many times differentiable. In this situation a simple estimate can be obtained.

Theorem 2.1. *Let f, g be two orientation preserving homeomorphisms of the circle S^1. Let f be homeomorphic to a rigid rotation by means of the smooth homeomorphism h. Then*

$$\rho(f) - \rho(g) = O\Big(\max_{s \in S^1} |f(s) - g(s)|\Big) \tag{2.3}$$

and the constant in the O-symbol depends on the Lipschitz constant of h.

Proof. Apply the homeomorphism h to both f and g. Then f reduces to the rigid rotation $s \to s + \sigma$, while g reduces to $\tilde{g}$ (say). Let ϵ denote the maximal difference between the rigid rotation $s \to s + \sigma$ and $\tilde{g}(s)$ on S^1. Lift both maps from S^1 to the real line. The iterates of 0 under the map $\tilde{g}$ increase at least as fast as the iterates of the rigid rotation $s \to s + \sigma - \epsilon$, and at most as fast the iterates of the rigid rotation $s \to s + \sigma + \epsilon$. Hence $\sigma - \epsilon \leqslant \rho(\tilde{g}) \leqslant \sigma + \epsilon$. This proves the result, since $\rho(g) = \rho(\tilde{g})$ and $\epsilon = O(\max|f - g|)$. □

This result implies that the error in the approximated rotation number is at most of the order of the error in the approximation of γ by $\boldsymbol{p}(\{x_i\}_{i=1}^N)$ plus an interpolation error of order $O(N^{-2})$. This is easily seen if one chooses for f the circle map $\Phi|_\gamma$ and for g the approximated circle map ϕ_N. However, the estimates are derived from estimates measuring the tangential and radial error in the obtained polygon. Clearly, in the definition of ϕ_N, and thus in ρ_N, the radial (error) component is not so important. So one might hope that the error in ρ_N is much smaller than the error in the polygonal approximation.

There is some evidence for additional accuracy in the approximated rotation number. In Theorem 2.1, let g be a piecewise linear approximation of f obtained by *interpolation* in N abscissae on the circle S^1. Then, if f is a homeomorphism, so is g. The difference between f and g is of the order of the square of the maximal distance between two successive abscissae. Nevertheless $\rho(f) = \rho(g)$ if for two interpolation abscissae we have the relation $f(t_i) = t_j$ (mod 1). This follows immediately from the equivalent definition for the rotation number given in [3, p.296] by considering $\rho_y(f)$, $y = t_i$. This suggests that some error components in the approximation $\boldsymbol{p}(\{x_i\}_{i=1}^N)$, and thus ϕ_N, do not always carry over to ρ_N. Of course, interpolation on top of another approximation does play a part in general. Cf. the discussion about $\Phi|_\gamma - \phi_N$ preceding the above theorem.

3. On the computation of a rotation number

Let ϕ be a lifting of an orientation preserving circle map to the real line. An example of such a map is ϕ_N introduced in the previous section. For such a map the rotation number ρ exists. In this section we give a few algorithms for the computation of ρ. In these algorithms ϵ is a small positive tolerance.

Algorithm 3.1. Choose an integer n (say $n = 3000$) and compute the sequence

$$\rho^{(jn)} = \frac{1}{jn}\phi^{jn}(0), \quad j = 1, 2, \ldots .$$

Accept $\rho^{(jn)}$ as an approximation to ρ if

$$|\rho^{(jn)} - \rho^{(jn-n)}| < \epsilon.$$

This algorithm is based on the classical definition of the rotation number. If the rotation number of the map ϕ is a rational number p/q, this algorithm detects it by converging to a stable fixed point of ϕ^q, provided q is much smaller than n. So n has to be large in relation to the frequencies one hopes to find.

Algorithm 3.2. This algorithm is in essence the one described in [9]. Consider the set of points

$$(j, \rho^{(j)}), \quad j = 1, 2, 3, \ldots, M. \tag{3.1}$$

The slope of a linear least squares approximation to these data tends to the rotation number for $M \to \infty$. Let r_M denote this slope. Pick n as in the above algorithm and accept r_{jn} as an approximation if

$$|r_{jn} - r_{jn-n}| < \epsilon.$$

Observe that the cost of the computation of r_{nn} is hardly different from the computational cost of $\rho^{(jn)}$. The approximation r_{jn} turns out to be a much better approximation than $\rho^{(jn)}$ if the rotation number is not a rational number with small denominator. For rational rotation numbers the previous algorithm tends to be somewhat more reliable because it is more reflects an ultimate convergence of a sequence of iterates than this one.

Algorithm 3.3. Instead of averaging in some way over one sequence of iterates of the map ϕ as in the above algorithm, one might average over many sequences of iterates of the map ϕ. The advantage of this approach is that the sequences of iterates may be computed in parallel, or on a vector processor with pipelining. Such an algorithm is advantagous if the total number of iterates for an approximation of some quality is not much larger than for the previous algorithms. Some preliminary tests have been made using up to 50 sequences of iterates of ϕ, using equidistant starting values on the circle [0, 1). For smooth conjugacies and irrational rotation numbers the experiments are promising, and this is what one would expect. But the idea needs some further elaboration, in particular if the map ϕ is not differentiable on the circle, or if ϕ is homeomorphic to a rigid rotation through a non-differentiable homeomorphism.

Algorithm 3.4. Observe that an approximation with guaranteed error bound (apart from rounding errors) may be obtained if one uses the alternative definition for the rotation number as given in

[3]. For this algorithm we need the circle map ϕ mod 1 rather than its lifting ϕ to the real line. Let $M(n)$ denote the number of instances in which $\phi^j(0)$ mod 1, $j = 1, \ldots, n$ is an element of the interval [0, $\phi(0)$ mod 1). Then, cf. [3, relation (6.2.5)], we have

$$|\rho - (1/n)M(n)| \leqslant 1/n. \tag{3.2}$$

We turn this into an algorithm by computing the upper and lower bound for ρ in each iteration step. Then we keep track of the maximum of the lower bounds and the minimum of the upperbounds. In this obvious way we are able to find an approximation to ρ with an error of at most 1.1 E − 09 using 23 million iterates. This is quite an improvement over the direct relation (3.2), which would require about 10^9 iterates.

Of these algorithms only the last one has a reliable stopping criterion. However, this algorithm is quite slow; it may be used if one really needs the error bounds. On a sequential machine Algorithm 3.2 seems most promising, whereas Algorithm 3.3 (in some form or another) might be useful on a pipeline machine or in a parallel architecture environment.

Rational rotation numbers correspond with fixed points of some power of ϕ. In order to see whether the approximation is close to a rational number with small denominator, we use a continued fraction expansion of the approximate rotation number, cf. [6, §13]. This continued fraction expansion results in a sequence of rational approximations p_i/q_i, with increasing q_i. In addition, we have the estimate

$$|\rho_{\text{appr}} - p_i/q_i| < 1/q_i q_{i+1}.$$

Hence, using these approximations we can easily find how close the approximate rotation number is to a rational number with denominator less than 100 (say). An algorithm (in PASCAL) is shown in Fig. 1. The input is a positive real number x, the output consists of two real vectors containing the successive numerators and denominators respectively. The length of this array is the output integer 'iDepth'. The algorithm uses reals instead of integers in order to avoid integer overflow. Also, the test in the while-loop is necessary for avoiding integer overflow in the function 'Trunc' (in PASCAL 'MaxInt' has the value of the largest integer and 'Trunc' returns the integer part of a real).

Periodicity is also looked for in a different way. Upon acceptance of an approximation by either Algorithm 3.1 or Algorithm 3.2 we have an iterate $\phi^m(0)$, for some large m. Now we compute $\phi^{m+j}(0)$, $j = 1, \ldots, 100$, and we look for an index j such that

$$|\phi^{m+j}(0) - \phi^m| < \delta \text{ mod } 1$$

where δ is much smaller than ϵ. If the test fails, we have no periodicity. If for some j the inequality holds, then we expect periodicity, and $\phi_m(0)$ is approximately a fixed point of the map ϕ^j. We use 100 additional iterates, but this is only a convenient number. It is not clear as yet what number may be motivated in view of the value of the number of iterates used in Algorithm 3.1 or Algorithm 3.2.

The list of algorithms is not exhaustive. In [3, p.302] the use of the conjugacy relation for finding a rotation number is discussed in relation to the *small divisor problem*. In view of that discussion the use of the conjugacy relation seems hardly promising for algorithmic use.

Some experiments have been done with the following algorithm.

Algorithm 3.5. Suppose one has a estimate ρ_0 for the rotation number ρ. Let ρ_0 be approximated by the rational number p_0/q_0. Determine l as the smallest positive integer for which

$$\phi^{q_0 l}(0) \bmod 1 - \phi^{q_0(l+1)}(0) \bmod 1$$

has an extremum. Intuitively, after l iterations of the map $\phi^{q_0} \bmod 1$ we 'cross' the value 1 for the first time. On the basis of linear interpolation we determine a fraction α such that

$$(1-\alpha)\phi^{q_0 l} = \alpha\phi^{q_0(l+1)}(0) \bmod 1.$$

Define a new approsimation ρ_1 by

$$\rho_1 = (p_0 + 1/(l+\alpha))/q_0.$$

Continue by finding a rational approximation for ρ_1 with larger denominator than q_0, etc. The rational approximations required should be obtained through successive truncated continued fractions. This process, inspired by the algorithms of the French school (cf. [8]), works well for rotation numbers not too close to a simple rational number. For such rotation numbers the approximation is surprisingly good in relation to the number of iterates (of ϕ) used. On the other hand, if $0 < \rho < 10^{-4}$, this method is costly as compared with Algorithm 3.1, with similar problems if ρ is very close to a simple rational number like $\frac{3}{7}$. If ρ equals a simple rational rotation number, the algorithm detects this by testing for stationary iterates of $\phi^{q_0}(0)$. In view of the (experimentally) checked bad behavior for rotation numbers close to p/q with small q, this

```
Const Depth = 10;
Type VecDepth = array[0..Depth] of real;
Procedure ContinuedFrac(x:real; var iDepth: integer;
var NumerVec, DenomVec:VecDepth);
   Procedure EvalCF(FracM1:real; var Frac:VecDepth)
   var i:integer;
   begin
     if (iDepth > 0) then Frac [1] := Numer Vec[1]*Frac[0]+FracM1;
     if (iDepth > 1)then
     For i := 2 to IDepth do Frac[1] := NumerVec[1] ∋ Frac[i-1]+Frac[i-2] end;
   end;
begin
   iDepth := 0;
   NumerVec[0] := Trunc(x);
   DenomVec[0] := 1.0;
   while (((x-NumerVec[iDepth])*MaxInt > 1.0) and (iDepth < Depth)) do
   begin
     x := 1.0/(x-NumerVec[iDepth]);
     iDepth := iDepth+1;
     NumerVec[iDepth] := Trunc(x);
   end;
   EvalCf(0.0,DenomVec);
   EvalCf(1.0,NumerVec);
end;
```

Fig. 1. Algorithm for the approximation of a positive real number by successive truncated continued fractions. The numerators and denominators are in the arrays 'NumerVec' and 'DenomVec' respectively.

algorithm cannot be recommended for general use. In interactive situations—on a Personal Computer—it is quite useful.

4. Numerical illustration

In this section we illustrate the results of the previous section by giving computational results for the delayed logistic map, cf. [1]. We concentrate on the approximation of the rotation number, not on the polygonal approximation. See [11] for error estimates in the polygonal approximation.

We consider the delayed logistic map Φ defined by

$$\Phi\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} y \\ \lambda y(1-x) \end{pmatrix}. \tag{4.1}$$

This map has been studied in great detail in [1]. For $\lambda > 2$ there is an invariant curve, at least until $\lambda \sim 2.20$. For $2 < \lambda \leqslant 2.16$ the invariant curve is smooth. For larger values of the parameter smoothness is gradually lost, cf. [1]. In the limit, for $\lambda \downarrow 2$ the rotation number of the invariant curve (shrinking to a point) can be computed from the eignevalues of the linearized map. This results in $\rho(\lambda = 2) = \frac{1}{6}$. A graph of $\rho(\lambda)$ is given in [1]. We compute the values of ρ by iterating the map Φ while counting the number of rotations around the fixed point (a source) (x_S, y_S), where

$$x_S = y_S, \qquad x_S = (\lambda - 1)/\lambda.$$

By using 2 250 000 iterations (4 000 000 if $\lambda = 2.16$) we obtain results believed to be accurate to about six decimal places. Observe that the algorithms of Section 3 are not applicable, because we do not know the circle map explicitly. These results are mentioned and used in the tables below.

The implementation of the method is very much as described in [10]. The initial polygon (obtained from previous conputations) contains $N = 132$ vertices and is already within a distance 0.005 of the invariant curve for $\lambda = 2.16$. Vertices are insterted if the distance between successive vertices is too large, much as in [10]. With slow convergence its hard to stop the process close to the limiting (number of iteration steps $\to \infty$) polygon. The implementation uses Φ^p where p is increased if the rate of convergence decreases. In the implementation used the process is stopped if two successive polygons have an average distance less than 1.0 E − 08.

The rotation number is approximated through Algorithm 3.2 with $n = 3000$ and tolerance $\epsilon = 1.0$ E − 07.

Some results for the method of Section 2 are shown in Table 1. The result of the computation for $\lambda = 2.16$ is used as an initial guess for the next computation, etc. One should observe that for decreasing λ the attractivity of the invariant curve decreases. This is indicated in the increase of p for decreasing λ. On the other hand, the map Φ becomes less complex and the invariant curve tends to a circle for decreasing λ. This is indicated by the results obtained for the rotation number as function of λ. For $\lambda = 2.16$ the error is in the fourth decimal place, for $\lambda = 2.02$ the error is in the sixth decimal place. The column indicated by $\rho_{p=1}$ gives the approximations for the rotation number of Φ, computed from the approximations ρ_{appr} related to Φ^p.

For the method described in [10] computations for increasing values of N have been done in order to obtain an impression about the behavior of the error if the number of vertices increases.

Table 1
Results for the Delayed Logistic Map (4.1) obtained with the method described in [10]

λ	p	N	ρ_{appr}	$\rho_{p=1}$	ρ_{exact}
2.16	2	141	0.295260	0.147630	0.1474935
2.14	2	141	0.301533	0.150767	0.1507185
2.12	2	141	0.307692	0.153946	0.1535363
2.10	2	141	0.312500	0.156250	0.1561058
2.08	6	141	0.950920	0.158487	0.1584864
2.06	6	141	0.964286	0.160714	0.1607109
2.04	6	141	0.976744	0.162791	0.1628037
2.02	6	141	0.988889	0.164815	0.1647843
2.02	12	141	0.977273	0.164773	0.1647843

Table 2
Results for the Delayed Logistic Map (4.1) obtained with the method described in [10]. Here $\lambda = 2.16$

N	p	ρ_{appr}	$\rho_{p=1}$	error
141	2	0.2952600	0.1476300	1.4 E−04
279	2	0.29508197	0.1475410	4.7 E−05
561	4	0.59000000	0.1475000	6.5 E−06
1122	4	0.58997696	0.1474942	7.5 E−07

The results for $\lambda = 2.16$ are given in Table 2. The number of vertices is essentially doubled per computation. The error in the rotation number as listed in the fourth column shows a quadratic dependence on the maximal distance between two adjacent images of vertices.

The number of iterates p of the map Φ has quite some influence on the results. We give an example. For $\lambda = 2.11066\ldots$ we have $\rho_{\text{exact}} \sim 13/84$. With $p = 21$ we obtain intervals of phase-locking on the λ-axis. In Table 3 we give some intervals for various values of the number of vertices N. Of course, in these intervals we find $\rho_{p=1} = 13/84$. Clearly, the length of the phase-locking interval decreases for increasing N. This agrees with the expectations in view of Theorem 2.1. On the other hand, if one usus $p = 19$, no phase-locking is observed in a neigborhood of $\lambda = 2.11066$. The approximations then form a smooth curve, as in [1]. Clearly, the value $p = 21$ interferes with the value of the rotation number $\rho_{\text{exact}} = 13/84$. The approximate circle map then shows phase-locking, where there is none. On the other hand, the phase-locking for $\lambda \sim 2.18$ is reproducted by the algorithms. One may then use the fixed points of ϕ_N^7 as an initial guess for the determination of sinks and saddles on the curve γ. This has been described in

Table 3
Phase-locking intervals for ρ_{appr}

N	λ_{low}	λ_{up}
38	2.110825	2.111545
97	2.110445	2.111055
293	2.110582	2.110765
398	2.110586	2.110741
613	2.110619	2.110730

some detail in [5]. For instance, the fixed points in Fig. 6 in [10] have been obtained in this way.

These examples show the possibility of approximating the rotation number accurately with polygonal approximations of the invariant curve. The approximation of the rotation number in a direct way requires about 10^6 evaluations of Φ for an accuracy of sic decimal digits. A polygonal approximation with some 1000 vertices results in a similar error in the rotation number (obtained through the process of Section 2). If one is interested in the rotation number for many successive values of the parameter λ, the approximated rotation number on the polygonal approximation of the invariant curve requires less evaluations of Φ than the direct approximation. For simple maps like (3.1) this hardly pays. The invariant curve algorithm is quite expensive in relation to an evaluation of Φ. However, in many cases of interest the map Φ is the Poincaré map corresponding to an ordinary differential equation. Then the overhead of the invariant curve algorithm is relatively small. In these instances the algorithm of Section 2 is advantageous. We intend to investigate this point for the Josephson equation, cf. [9].

Aknowledgement

I am indebted to the referee for stimulating comments on the first version of this paper.

References

[1] D.G. Aronson, M.A. Chory, G.R. Hall and R.P. McGeehee, Bifurcations from an invariant circle for two-parameter families of maps of the plane, *Commun. Math. Phys.* **83** (1982) 303–354.

[2] Tze Ngon Chan, Numerical bifurcation analysis of simple dynamical systems, Thesis (unpublished), Dept. of Computer Science, Concordia University, Montreal.

[3] J. Guckenheimer and Ph. Holmes, *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields* (Springer, New York, 1983).

[4] J.K. Hale, *Ordinary Differential Equations* (Wiley Interscience, New York, 1969).

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

[6] O. Perron, *Die Lehre von den Kettingbrüchen* (Teubner, Leipzig, 1929).

[7] L.L. Schumaker, On shape preserving quadratic spline interpolation, *SIAM J. Numer. Anal.* **20** (1983) 854–864.

[8] E. Thoulouze-Pratt, and M. Jean, Analyse numérique du comportement d'une solution presque périodique. *Int. J. Non-Linear Mechanics* **17** (1982) 319–326.

[9] M. van Veldhuizen and H.A. Fowler, Subharmonic frequency locking in the resistive Josephson thermometer, *Phys. Rev.* **B31** (1985) 5805–5810.

[10] M. van Veldhuizen, A new algorithm for the numerical approximation of an invariant curve, Report Dept. Math. Comput. Sci., Vrije Universiteit, Amsterdam, 1985; and, *SISSC*, to appear.

[11] M. van Veldhuizen, On polygonal approximations of an invariant curve, Tech. Report WS314, Dept. Math. Comput. Sci., Vrije Universiteit, Amsterdam, 1986; and, *Math. Comput*, submitted for publication.