

 Open access • Journal Article • DOI:10.1007/BF01209326

Converse KAM: theory and practice — [Source link](#)

Robert S. MacKay, Ian C. Percival

Institutions: Queen Mary University of London

Published on: 01 Dec 1985 - Communications in Mathematical Physics (Springer-Verlag)

Topics: Invariant (mathematics)

Related papers:

- [A method for determining a stochastic transition](#)
- [A universal instability of many-dimensional oscillator systems](#)
- [Non-existence of invariant circles](#)
- [The discrete Frenkel-Kontorova model and its extensions: I. Exact results for the ground-states](#)
- [Existence of quasi-periodic orbits for twist homeomorphisms of the annulus](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/converse-kam-theory-and-practice-j59lvpczp>

Converse KAM: Theory and Practice

R. S. MacKay^{1,*} and I. C. Percival²

1 I.H.E.S., F-91440 Bures-sur-Yvette, France

2 Department of Applied Mathematics, Queen Mary College, London E1 4NS, England

Abstract. We unify, extend, reinterpret and apply criteria of Birkhoff [1], Herman [9], Mather [2, 3], Aubry et al. [4, 5], and Newman and Percival [6] for the nonexistence of invariant circles for area preserving twist maps. The criteria enable one to establish regions of phase space through which no rotational invariant circles pass. For families of maps the same can be done for regions of the combined space of phase points and parameters. The criteria can be implemented rigorously on a computer, and give a practical method of proving quite strong results. As an example, we present a computer program which proved that the “standard map” has no rotational invariant circles for any parameter value $|k| \geq 63/64$.

Contents

1. Introduction	469
2. Twist Maps and Birkhoff's Theorem	471
3. Cone-Crossing Criterion	473
4. Following Orbits	478
5. Application to the Standard Map	480
6. Orbits of Minimum Action	486
7. Conefields	488
8. Other Techniques	493
9. Further Discussion	495
Appendix A: Symmetries	497
Appendix B: Computer Program.	497

1. Introduction

Given an area preserving map, one often wants to know if it has any invariant circles, and if so, where they are. This is important on the one hand for confinement problems, as an invariant circle confines the orbits of points inside it, and on the

* Present address: Mathematics Institute, University of Warwick, Coventry CV4 7AL, England

other hand for transport problems, as under certain conditions if there are no invariant circles separating two points then there exist orbits from any neighbourhood of one to any neighbourhood of the other.

There are two approaches to the problem. One is to find regions guaranteed to contain invariant circles with certain characteristics. The other is to find regions through which pass no invariant circles of given class. A priori, the results of the two approaches would be equally valuable.

KAM theory solves the first approach, at least in principle. It can be used to find most invariant circles, in fact all those with Diophantine rotation number and sufficiently differentiable conjugacy to rotation. One just has to make coordinate changes bringing the system successively closer to the standard integrable form:

$$\left. \begin{aligned} \theta' &= \theta + \Omega(r) \\ r' &= r \end{aligned} \right\} \quad (1.1)$$

near the suspected circle, until any of the versions of the KAM theorem applies. Versions of the KAM theorem for which useful results can be obtained in a practical time, however, are only just being developed [7, 8].

In this paper we take the second approach which appears to be much more practical. We develop “converse KAM theorems,” i.e. criteria for nonexistence of invariant circles of given rotation class through given regions of the phase space. They can be used quite easily to get results, even rigorous ones, which are close to optimal.

There are several converse KAM theorems available, of which two are of direct interest for this paper. The first is based on a theorem of Birkhoff [1] (see also [9] for a good exposition), and was developed and applied by Herman [9] and Mather [2, 3]. The second is connected with the action principle for twist maps, was conjectured and applied by Newman and Percival [6], and independently proved and applied by Aubry and coworkers [4, 5]. One of the results of this paper is to show that the two are equivalent, so we will refer to them as “the criterion.”

In its elementary form, the criterion is for non-existence of invariant circles of given rotation class through a given point for a given C^1 area preserving map, but the criterion is such that if it succeeds in proving non-existence for a particular point for a particular map, then it will succeed for all close enough points of all close enough maps. Thus, one can prove non-existence of invariant circles of given rotation class through a whole set of phase space for a whole set of maps. We present two algorithms which implement this. The first is more efficient; the second provides, as a bonus, Lipschitz bounds on the slopes of circles that may exist.

Throughout the paper the ideas are illustrated by application to the “standard map” [10, 11]. This is a one parameter family of maps of a cylinder

$$\left. \begin{aligned} p' &= p - \frac{k}{2\pi} \sin 2\pi x \\ x' &= x + p' \end{aligned} \right\} \quad (1.2)$$

KAM theory applies for parameter k small enough. For example, Herman [8] has proved existence of a rotational golden circle for $|k| \leq 1/34$. From the other side, Mather [3] proved non-existence of any rotational invariant circles for $|k| > 4/3$.

Aubry [5] improved this to $|k|$ greater than the root (~ 1.23) of some transcendental equation. Numerical work of Newman and Percival [6], using the same criterion, suggested that there are none for $|k| \geq 1.04$, but their result is not rigorous because they tested only a finite set of parameter values and phase points, and they did not account for rounding errors.

In this paper, we apply the criterion, with error-bounding interval arithmetic on a computer, to prove that the standard map has no rotational invariant circles for all parameter values

$$|k| \geq 63/64 = 0.984375. \tag{1.3}$$

This result is probably close to optimal, as numerical work of Greene [10], using the “residue criterion” (Sect. 8), suggests that there is a value

$$k_c \sim 0.971635406 \tag{1.4}$$

such that there are no rotational invariant circles for $|k| > k_c$, and there is at least one for $|k| \leq k_c$.

The organisation of the paper is as follows. In Sect. 2 we introduce Birkhoff’s theorem for area preserving twist maps. In Sect. 3 we show how it leads to a criterion for nonexistence of circles. Section 4 develops a method for rigorously following orbits so that the criterion of Sect. 3 can be applied. Section 5 describes its rigorous application to prove (1.3). In Sect. 6 we describe the formulation of the criterion in terms of orbits of minimum action, and prove equivalence. Section 7 develops a third way of looking at the criterion which gives considerable geometrical insight. In Sect. 8, we discuss other criteria for non-existence of invariant circles, and their relation with the one discussed here. In Sect. 9, we conjecture that the criterion is exhaustive, discuss generalisations, and summarise the results. Appendix A contains some remarks on symmetries of maps, and in order that the reader may check the proof of (1.3), the computer program is provided in Appendix B.

2. Twist Maps and Birkhoff’s Theorem

An area preserving map of a surface may possess invariant circles of several rotation classes, going around ends of the surface (e.g. encircling a cylinder) or around fixed points of the map. Consider one rotation class of circles at a time. By removing the fixed point they encircle, if any, and irrelevant parts of the surface, the map can be regarded as acting on part of a cylinder.

A common feature of such maps is “twist,” also known as “shear.” A continuously differentiable (C^1) map T of a cylinder is said to be a *twist map* if there exist coordinates $(x, p) \in S^1 \times \mathbb{R}$ such that, writing

$$(x', p') = T(x, p), \tag{2.1}$$

then

$$\left. \frac{\partial x'}{\partial p} \right|_x \neq 0, \tag{2.2}$$

and hence has constant sign.

For example, near an elliptic fixed point with multipliers not a third or fourth root of unity, there are polar coordinates (θ, r) such that

$$\left. \begin{aligned} \theta' &= \theta + b_0 + b_1 r + o(r) \\ r' &= r + o(r) \end{aligned} \right\} \tag{2.3}$$

Typically $b_1 \neq 0$, so the map is a twist map near enough to the fixed point.

Other examples of area preserving twist maps are provided by

$$\left. \begin{aligned} p' &= p + f(x) \\ x' &= x + p' \end{aligned} \right\} \tag{2.4}$$

for arbitrary $C^1 f: S^1 \rightarrow \mathbb{R}$. This includes the ‘‘standard map’’ as the particular case

$$f(x) = -\frac{k}{2\pi} \sin 2\pi x. \tag{2.5}$$

Equivalent maps may be obtained by lifting from the cylinder to the plane, e.g. by considering x in (2.4) to be on the real line \mathbb{R} instead of the circle S^1 ; this increases the size of the phase space. Conversely, the phase space can sometimes be conveniently reduced by factorizing out symmetries as described in Appendix A. We will frequently carry out such transformations without mentioning it.

In all that follows, we will consider C^1 invertible, area preserving twist maps T of a cylinder, homotopic to the identity. Such maps preserve the ends of the cylinder, and preserve orientation.

We will be particularly concerned with *rotational* invariant circles, those which are homotopic to the circle $p = 0$, i.e. that encircle the cylinder. This is because they are necessary for confinement, by a corollary of the following important theorem:

Theorem (Birkhoff). *For the above class of maps, the boundary of any open invariant set homeomorphic to the cylinder and containing all points below some level and none above some other level, is the graph $\{(x, P(x)) : x \in S^1\}$ of some continuous (single-valued) function $P: S^1 \rightarrow \mathbb{R}$. In particular, it is a rotational invariant circle.*

For proofs of this theorem and the following corollaries see [1, 3, 9].

Confinement Corollary. *If the orbits of all points below some level p_- remain below some other level p_+ , then there exists a rotational invariant circle between levels p_- and p_+ .*

Proof. Let U be the set of points (x, p) with $p < p_-$, and

$$V = \bigcup_{n=-\infty}^{\infty} T^n U. \tag{2.6}$$

The idea is to apply Birkhoff’s theorem to V , but, although connected, it need not be homeomorphic to a cylinder, because it could have holes in it. So fill in the holes first as follows. Consider the complement V^c , and let W be the component of V^c

containing all points above level p_+ . Then its complement W^c is V with the holes filled in, and satisfies the hypotheses of Birkhoff's theorem, so its boundary is a rotational invariant circle between levels p_- and p_+ . \square

In general rotational circles can intersect vertical lines more than once, but Birkhoff's theorem also has the following corollary:

Circle Corollary. *Every rotational invariant circle is the graph of some continuous function $P : S^1 \rightarrow \mathbb{R}$.*

Proof. A rotational invariant circle separates the cylinder into two invariant components, the lower of which satisfies the hypotheses of Birkhoff's theorem. \square

In fact, though $P(x)$ is not necessarily differentiable, we always have:

Lipschitz Corollary. *The function $P(x)$ in Birkhoff's theorem is Lipschitz.*

A function $P : \mathbb{R} \rightarrow \mathbb{R}$ is said to be *Lipschitz* if the slopes

$$D(x_1, x_2) = \frac{p(x_2) - p(x_1)}{x_2 - x_1}, \quad x_1 \neq x_2 \tag{2.7}$$

are uniformly bounded, i.e.

$$\exists D^\pm \text{ such that } D^- \leq D(x_1, x_2) \leq D^+. \tag{2.8}$$

For $P : S^1 \rightarrow \mathbb{R}$, lift to the universal cover. Such a range of slopes we call a *Lipschitz cone*.

Proof. Since T is a C^1 twist map, the images of all vertical lines have slope bounded away from vertical in any compact domain, by

$$D^+ = \max \frac{\partial p'}{\partial p} \Big/ \frac{\partial x'}{\partial p}. \tag{2.9}$$

Since a rotational invariant circle meets each vertical line only once, it also meets their images only once. Thus its slopes are bounded above by D^+ . A lower bound can be obtained similarly by considering T^{-1} . \square

The circle corollary and its strengthening by the Lipschitz corollary are the heart of this paper.

3. Cone-Crossing Criterion

The Lipschitz corollary gives immediately a criterion for non-existence of rotational invariant circles, suggested to us by Herman. A rotational invariant circle separates the cylinder into two invariant components. If one finds a tangent orbit, i.e. an orbit of

$$\begin{aligned} (T, DT)(X, v) &= (T(X), DT_X v), \\ X &= (x, p), \quad v = (\delta x, \delta p), \end{aligned} \tag{3.1}$$

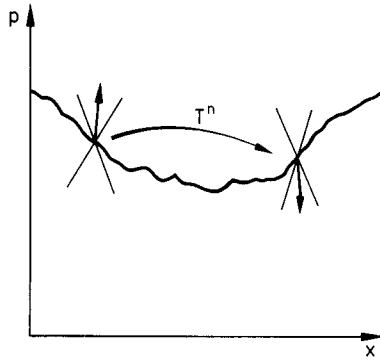


Fig. 1. Impossible situation because a tangent vector crosses an invariant circle, in (x, p) coordinates

for which v sometimes lies above the Lipschitz cone and sometimes below (Fig. 1), then the base orbit $T^n(X)$ can not lie on a rotational invariant circle. Otherwise, there would be a nearby orbit which would cross the circle. We call this the “cone-crossing” criterion.

To use this in practice, one has to know explicit Lipschitz constants. They could be obtained from the proof of the Lipschitz corollary. Instead, we will obtain an explicit cone condition, independent of the map, in a more convenient coordinate system.

Let $\pi_1 : S^1 \times \mathbb{R} \rightarrow S^1$ be the projection onto the first coordinate. Since T is invertible and preserves orientation, the induced map on a rotational invariant circle:

$$x' = h(x) = \pi_1 T(x, P(x)) \tag{3.2}$$

must also be invertible and preserve orientation. Thus for h we have

$$0 < \frac{x'_1 - x'_2}{x_1 - x_2} < \infty, \quad x_1 \neq x_2. \tag{3.3}$$

By considering T and T^{-1} this could be converted back to an explicit Lipschitz cone in (x, p) coordinates. But it is more convenient to change to (x, z) coordinates, with:

$$z(x, p) = \pi_1 T(x, p). \tag{3.4}$$

Then (3.3) gives the cone condition

$$0 < \frac{z_1 - z_2}{x_1 - x_2} < \infty, \quad x_1 \neq x_2 \tag{3.4a}$$

for points $(x_1, z_1), (x_2, z_2)$ on a rotational invariant circle. This confines the slopes of rotational invariant circles to a right angle, independent of the map. In fact, since we know $P(x)$ is Lipschitz, $h(x)$ is also Lipschitz, and so the slopes are confined to a strictly smaller angle, but which depends on the map.

As an example of the coordinate change, the map (2.4) becomes

$$\begin{cases} z' = g(z) - x \\ x' = z, \end{cases} \tag{3.5}$$

where

$$g(z) = 2z + f(z). \tag{3.6}$$

The change of coordinates (3.4) is always invertible, by the twist condition (2.2), though note that the map need not preserve the area $dx \wedge dz$. Note also that in these coordinates, the symmetry which collapses the map onto the cylinder is the unit diagonal translation

$$(x, z) \rightarrow (x + 1, z + 1) \tag{3.7}$$

rather than the unit horizontal translation. So rotational invariant circles have an average slope of 1 rather than 0.

Cone condition (3.4a) gives the following criterion for non-existence of rotational invariant circles, on which most of this paper is based:

Criterion 1. *If (δx_i) are the δx -components of a tangent orbit (3.1) to an orbit (X_i) , with*

$$\begin{aligned} \delta x_0 &\leq 0, \\ \delta x_1 &> 0, \\ \delta x_n &\leq 0 \text{ for some } n > 1, \end{aligned}$$

then the orbit (X_i) does not lie on a rotational invariant circle.

Proof. In (x, z) coordinates, the initial tangent vector

$$v_0 = (\delta x_0, \delta z_0) = (\delta x_0, \delta x_1) \tag{3.8}$$

is on one side of the cone (Fig. 2). Let $n \geq 2$ be the first time for which $\delta x_n \leq 0$. Then $\delta x_{n-1} > 0$, and so

$$v_{n-1} = (\delta x_{n-1}, \delta z_{n-1}) = (\delta x_{n-1}, \delta x_n) \tag{3.9}$$

is on the other side of the cone. \square

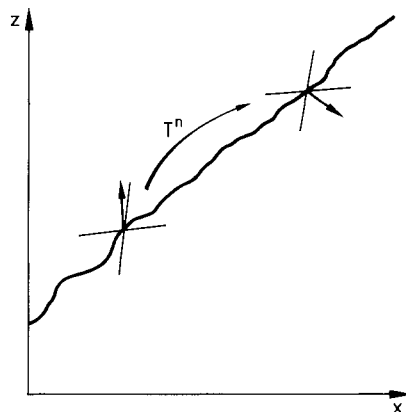


Fig. 2. Impossible situation because a tangent vector crosses an invariant circle, in (x, z) coordinates

As an application, for the map (2.4) [equivalently (3.5)], there are no rotational invariant circles if:

$$m = \min_x g'(x) \leq 0. \tag{3.10}$$

This is because (3.5) implies that:

$$\delta x_{t+1} = g'(x_t) \delta x_t - \delta x_{t-1}. \tag{3.11}$$

So choosing

$$\delta x_0 = 0, \quad \delta x_1 > 0, \tag{3.12}$$

we get

$$\delta x_2 = g'(x_1) \delta x_1. \tag{3.13}$$

Thus if

$$g'(x_1) \leq 0. \tag{3.14}$$

then $\delta x_2 \leq 0$ and there are no rotational invariant circles crossing the vertical line $x = x_1$. Every rotational invariant circle must cross each vertical, hence the result (3.10).

This is the first step in Mather's calculation in [3]. In the particular case of the standard map,

$$m = 2 - |k|, \tag{3.15}$$

so there are no rotational invariant circles if $|k| \geq 2$. Mather also used the same criterion, with $n = 2$ again, to show that there are no rotational invariant circles for the motion of a billiard ball on a table whose boundary is convex but has a point of zero curvature [2].

One can apply Criterion 1 with $n > 2$ to get stronger results. But we can save having to take n too large by doing something equivalent, namely improving the Lipschitz cone.

Knowing some cone constants

$$D_-^0 \leq \frac{z_1 - z_2}{x_1 - x_2} \leq D_+^0, \tag{3.16}$$

e.g. $D_-^0 = 0, D_+^0 = \infty$ as implied by (3.4a), we can obtain stronger ones as follows. The derivative DT on tangent vectors induces an operator on slopes, which, abusing notation, we represent by the same symbol.

If

$$\frac{z_1 - z_2}{x_1 - x_2} \leq D_+,$$

then

$$\frac{z'_1 - z'_2}{x'_1 - x'_2} \leq \max_x DT_x \cdot D_+, \tag{3.17}$$

where $X = (x, z)$. So we can replace D_+ by $\max_x DT_X \cdot D_+$, which is smaller for D_+ large. One can iterate this as many times as desired, but we might as well go straight to the limit, i.e. set D_+^0 to the largest root of

$$D_+ = \max_x DT_X \cdot D_+ . \tag{3.18}$$

Similarly, D_-^0 can be chosen as the smallest root of

$$D_- = \min_x DT_X \cdot D_- . \tag{3.19}$$

For example, for the map (3.5)

$$DT_X \cdot D = g'(z) - 1/D , \tag{3.20}$$

so we can take D_+^0 to be the largest root of

$$D_+ = M - 1/D_+ , \tag{3.21}$$

i.e.

$$D_+^0 = \frac{M}{2} + \sqrt{\frac{M^2}{4} - 1} . \tag{3.22}$$

Similarly, we can take

$$D_-^0 = \frac{M}{2} - \sqrt{\frac{M^2}{4} - 1} = 1/D_+^0 . \tag{3.23}$$

This allows us to strengthen Criterion 1 by taking the initial tangent vector $(\delta x_0, \delta z_0) = (1, D_+^0)$, and stopping if ever $\frac{\delta z_n}{\delta x_n} < D_-^0$. Note that it is only the direction of the tangent vectors, not their magnitude, that enters; and it is not necessary to calculate the directions exactly: an upper bound will do while the slope is positive, since DT_X is order preserving on directions. This is important, because it is not possible to evaluate expressions exactly on a computer. So a more practical expression of Criterion 1 is

Criterion 2. Take an initial point X and slope $D_+ \geq D_+^0$. Follow an orbit, setting

$$D'_+ = \text{u.b. } DT_X \cdot D_+ , \quad X' = T(X) . \tag{3.24}$$

If at some stage $D_+ < D_-^0$ then there is no rotational invariant circle through X .

Throughout this paper the symbol $\text{u.b.}(x)$ means any upper bound to x , and $\text{l.b.}(x)$ will mean any lower bound.

Note that Criterion 2 cannot actually produce stronger results than Criterion 1, but it will get the same results with fewer iterations. An alternative way of deriving Criterion 2 is, given a finite segment satisfying its hypotheses, push the slopes slightly so that the initial slope strictly exceeds D_+^0 , (3.24) is still satisfied, but the final slope is still strictly less than D_-^0 . Then add iterations at each end using global estimates on DT_X . Criterion 1 will be satisfied for large enough number of added iterations.

As an illustration, for map (3.5) the iteration in Criterion 2 is

$$\begin{aligned} z' &= g(z) - x, \\ x' &= z, \\ D'_+ &= \text{u.b. } [g'(z) - 1/D_+]. \end{aligned} \tag{3.25}$$

Applying Criterion 2 with one iteration is equivalent to Mather's calculation [3] that the standard map has no rotational invariant circles for $|k| > 4/3$. To get this, choose $z = z_0$ to minimise $g'(z)$. Then

$$D'_+ = m - 1/D_+^0, \tag{3.26}$$

which is less than D_-^0 if

$$m < M - \sqrt{M^2 - 4}. \tag{3.27}$$

Hence, if this condition is satisfied, there is no rotational invariant circle crossing the line $z = z_0$, and so none at all. For the standard map it reduces to

$$|k| > 4/3. \tag{3.28}$$

Equation (3.27) is also the basis for Herman's constructions of $C^{3-\epsilon}$ counter-examples to the KAM theorem [9].

By taking more iterations, one can obtain stronger results. In Sect. 5, for instance, by taking orbit segments of length up to 196, we will use Criterion 2 to prove that the standard map has no rotational invariant circles for

$$|k| \geq 63/64. \tag{3.29}$$

All that one needs is a way of following orbits sufficiently precisely. Such a way is described in the next section.

4. Following Orbits

In this section we describe one way to follow rigorously orbits of compact sets for discrete time dynamical systems. Such a method is necessary if we want to prove, with a finite amount of computation, non-existence of rotational invariant circles through *all* points, by Criterion 2: we have to follow the orbits of blobs of initial conditions. The method is completely general. It could also be used, for example, to establish rigorously regions of the outside of the Mandelbrot set, or regions of the basins of attraction of periodic orbits. It is just one way among many possibilities, but works well.

The basic idea is to choose a class of subsets of the phase space, and a way of bounding the image of such a set by a set of the same form. The simplest form to use is a parallelepiped. The simplest way to specify it is as the image of the unit box I^n , $I = [-1, 1]$ under an affine map, where n is the dimension of the space. Thus, we consider the parallelepipeds

$$P = \{c + Q \cdot s : s \in I^n\} \tag{4.1}$$

with c a point on the manifold, and Q an $n \times n$ matrix.

We want to find a parallelepiped which is guaranteed to contain the image $F(P)$ under a map F , or even the images under a whole C^1 -compact set of maps. An answer is given by the following:

Lemma. *Given a point c' and $n \times n$ invertible matrix A and C^1 map F , then*

$$F\{c + Q \cdot s : s \in I^n\} \subset \{c' + Q' \cdot s : s \in I^n\}, \tag{4.2}$$

where

$$Q'_{ij} = A_{ij} w_j, \tag{4.3}$$

with

$$w_i = \text{u.b.} \left[|(A^{-1} \cdot (F(c) - c'))_i| + \text{u.b.} \sum_{x \in P} |(A^{-1} \cdot DF_x \cdot Q)_{ij}| \right]. \tag{4.4}$$

Proof.

$$Q'^{-1} \cdot (F(c + Q \cdot s) - c')_i = Q'^{-1} \cdot (F(c) - c')_i + (Q'^{-1} \cdot DF_{\xi_i(s)} \cdot Q \cdot s)_i, \tag{4.5}$$

where

$$\xi_i(s) = c + \lambda_i(s)Q \cdot s \quad \text{for some } \lambda_i(s) \in [0, 1] \tag{4.6}$$

by the Mean Value Theorem. Thus

$$|Q'^{-1} \cdot (F(c + Q \cdot s) - c')_i| \leq \frac{1}{w_i} \left[|A^{-1} \cdot (F(c) - c')_i| + |(A^{-1} DF_{\xi_i(s)} Q \cdot s)_i| \right]. \tag{4.7}$$

Now

$$\begin{aligned} |(A^{-1} DF_{\xi_i(s)} Q \cdot s)_i| &\leq \max_{s \in I^n} |(A^{-1} DF_{\xi_i(s)} Q \cdot s)_i| \\ &\leq \max_{s \in I^n} \sum_j |(A^{-1} DF_{\xi_i(s)} Q)_{ij}| \\ &\leq \max_{x \in P} \sum_j |(A^{-1} DF_x Q)_{ij}|. \end{aligned} \tag{4.8}$$

Thus

$$|Q'^{-1} \cdot (F(c + Q \cdot s) - c')_i| \leq 1 \tag{4.9}$$

by construction of w_i in (4.4). \square

Remark. This gives close to optimal results if

$$c' \sim F(c), \quad A \sim DF_c Q, \tag{4.10}$$

but A must not be chosen too close to singular. The main problem in applying this lemma is to choose how to “fatten” $DF_c Q$, i.e. push it away from singular. Some ways of doing this are discussed in the next section.

To get the bounds on quantities involving DF_x , $x \in P$, one can use estimates like

$$|L^T(x - c)| \leq \sum_j |(L^T Q)_j|, \quad x \in P, \tag{4.11}$$

where L^T represents a row vector.

5. Application to the Standard Map

In this section, we describe an application of Criterion 2 to the standard map. The aim is to establish rectangles $Iz \times Ik$ in (z, k) , where Iz and Ik are intervals, such that for each parameter value $k \in Ik$ there are no rotational invariant circles through any point of the interval Iz on the line $x = 0$. In particular, we will establish this for the rectangle $Iz = (-\infty, \infty)$, $Ik = [63/64, \infty)$, thus proving nonexistence of rotational invariant circles for $|k| \geq 63/64$. Note that in the light of remarks in Appendix A it suffices to take $Iz = [0, 0.5]$, and since $|k| > 4/3$ is already disposed of, we can take $Ik = [63/64, 4/3]$. The algorithm we present does not succeed directly for this rectangle; but the result will be proven by subdividing it into small enough rectangles on which it does succeed.

As already discussed, to prove nonexistence of rotational invariant circles it suffices to examine one vertical line. The reason we choose the line $x = 0$ is that for $k > 0$ it appears to be the most sensitive one.

We begin by describing the details of following orbits for a single map and then generalise to an interval in parameter. Then we discuss the iteration of the slope. The remaining step is a method of automatic subdivision of rectangles in (k, z) . We wrote a computer program to implement this algorithm rigorously. It is given in Appendix B. We conclude this section by presenting and discussing results of the program.

Because we wrote the program with the order of the coordinates (x, z) reversed, we will do so also throughout this section. We did not reverse the order in the rest of the paper, to preserve the usual conventions about horizontal and vertical as far as possible.

To follow orbits we used the method of the previous section. For initial parallelograms we took vertical intervals on $x = 0$, with zero width, i.e.

$$Q = \begin{bmatrix} zr & 0 \\ 0 & 0 \end{bmatrix} \tag{5.1}$$

in coordinates (z, x) , where zr is the halflength of the interval. Since the images will also have zero width, Q will always be close to singular, so we have to find a good way of pushing A away from singular. We tried various ways and ended up using a hybrid of two methods.

The first method, which is particularly simple to implement, is to choose the second column of A always to be vertical, i.e.

$$A_{22} = 0.$$

So given

$$c = (z_c, x_c) \tag{5.2}$$

and

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & 0 \end{bmatrix}, \tag{5.3}$$

set

$$c' = (z'_c, z_c) \quad \text{with} \quad z'_c \sim g(z_c) - x_c \tag{5.4}$$

and

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & 0 \end{bmatrix}, \tag{5.5}$$

with

$$A_{11} \sim g'(z)Q_{11} - Q_{21}, \tag{5.6}$$

$$A_{21} = \text{u.b.}(|Q_{11}| + |Q_{12}|), \tag{5.7}$$

$$A_{12} = 1.$$

Then we see that $w_1 \leq 1$ and

$$w_2 = \text{u.b.} \left(\left| g'(z)Q_{11} - Q_{21} - \frac{A_{11}Q_{11}}{A_{21}} \right| + \left| g'(z)Q_{12} - \frac{A_{11}Q_{12}}{A_{21}} \right| + |g(z_c) - x_c - z'_c| \right). \tag{5.8}$$

So we can take

$$Q' = \begin{bmatrix} A_{11} & w_2 \\ A_{21} & 0 \end{bmatrix}.$$

We use this for the first few iterations, while the parallelogram is still very thin. When its width gets comparable with its length, however, this method begins to throw away accuracy.

So when the ratio of the length of the two columns of Q exceeds some number QF we switch to a more general method, which is to use $A \sim DT_c \cdot Q$, and then to push it away from singular by rotating the second column through an angle depending on the anticipated effect of the variation of DT over the parallelogram. The size of this angle is controlled in the program by the number BF . Then we apply the lemma in its generality. The values of QF and BF were chosen to minimise the total time. It turned out not to be optimal to choose $QF = 0$, which is why we used a hybrid of both methods.

In fact we want to prove non-existence of rotational circles for a whole interval in parameter (1.3). One could simply take the u.b. in (5.8) and elsewhere over k as well as the parallelogram, but in order to reduce the growth of errors, it is better to treat k as a new coordinate, and consider the map

$$F(k, z, x) = (k, g(k, z) - x, z). \tag{5.9}$$

We will refer to k, z, x as coordinates 0, 1, 2 respectively.

Clearly we can take

$$Q_{00} = kr, \quad Q_{01} = Q_{02} = 0, \tag{5.10}$$

since k never changes. In the first stage of the iteration we will fix $Q_{22} = 0$.

Initially,

$$Q = \begin{bmatrix} kr & 0 & 0 \\ 0 & zr & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{5.11}$$

The iteration is

$$Q' = \begin{bmatrix} kr & 0 & 0 \\ Q'_{10} & Q'_{11} & Q'_{12} \\ Q'_{20} & Q'_{21} & 0 \end{bmatrix} \tag{5.12}$$

with

$$Q'_{11} \sim g_z Q_{11} - Q_{21}, \tag{5.13}$$

$$Q'_{21} = \text{u.b.} (|Q_{11}| + |Q_{12}|), \tag{5.14}$$

$$Q'_{10} \sim g_k kr + g_z Q_{10} - Q_{20}, \tag{5.15}$$

$$Q'_{20} = Q_{10}, \tag{5.16}$$

$$Q'_{12} = \text{u.b.}_{k, z \in P} \left\{ |g(k_c, z_c) - x_c - z'_c| + |g_k kr + g_z Q_{10} - Q_{20} - Q'_{10}| + \left| g_z Q_{11} - Q_{21} - \frac{Q'_{11} Q_{11}}{Q'_{21}} \right| + \left| g_z Q_{12} - \frac{Q'_{11} Q_{12}}{Q'_{21}} \right| \right\}. \tag{5.17}$$

When the ratio of the size of Column 2 to Column 1 becomes larger than QF , the program switches to the more general scheme. Rather than describe its explicit implementation here, we refer the reader to the program (Appendix B).

Next we discuss the iteration of the slopes. We put a constant slope D on each parallelogram. We could have taken an affine representation for D , i.e.

$$D = D_c + dD \cdot s, \quad s \in I^3, \tag{5.18}$$

but we did not implement it, although it might considerably reduce the number of iterations required.

The slope D is updated each step by:

$$D' = \text{u.b.}_{k, z \in P} [g_z(k, z) - 1/D]. \tag{5.19}$$

If D ever gets less than D_-^0 , (3.23), then there are no rotational invariant circles through the initial parallelogram, and we say the algorithm has succeeded.

Concerning initial conditions for D , one could start with D_+^0 , (3.22), on $x = 0$, as in Criterion 2. Going one step backwards, however, if we start with D_+^0 on $z = 0$ we get slope

$$g_z(k, 0) - 1/D_+^0 \tag{5.20}$$

on $x = 0$, which is less than D_+^0 . Incidentally, it is less than D_-^0 for $k > 4/3$, so we get no circles for $k > 4/3$ without doing any iteration. This is equivalent to Mather's calculation [3] again.

But we can also use the reversor (Appendix A),

$$S(x, z) = (-z, -x), \tag{5.21}$$

to allow us to start with

$$D = \text{u.b.} [\frac{1}{2} g_z(k, 0)] \tag{5.22}$$

on $x=0$, which is even smaller for $k < 4/3$. The reason we can start with (5.22) is as follows: Suppose slope D_0 at some point $(\zeta, 0)$ on $x=0$ leads to slope $D_n < D_0^-$ under T^n . Then by reversing, slope $1/D_n > D_+^0$ at $ST^n(\zeta, 0)$ leads to slope $1/D_0$ at $(0, \zeta)$ on $z=0$ under T^n . Taking one more iteration leads to slope

$$g'(0) - D_0 \tag{5.23}$$

at $(\zeta, 0)$ on $x=0$. So if

$$g'(0) - D_0 \leq D_0, \tag{5.24}$$

i.e.

$$D_0 \geq \frac{1}{2}g'(0), \tag{5.25}$$

then putting all the above together, we have slope $1/D_n > D_+^0$ leads to slope $D_n < D_0^-$ under T^{2n+1} , and so there is no rotational invariant circle through $(\zeta, 0)$.

This completes the description of the basic algorithm for testing a rectangle $Iz \times Ik$ for nonexistence of circles. Our goal is to establish non-existence for $Iz = [0, 0.5]$, $Ik = [63/64, 4/3]$. One could simply present the algorithm with the rectangle $Iz \times Ik$, but precision would be lost pretty fast and it may never succeed in showing non-existence of circles. What one needs is to subdivide the rectangle Iz

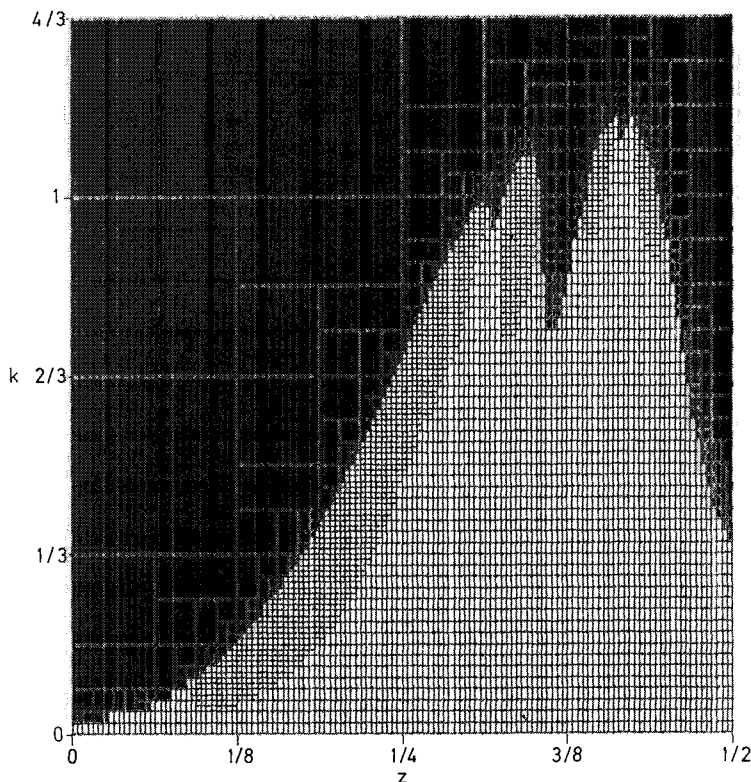


Fig. 3. Regions of the (z, k) -plane such that there are no rotational circles passing through z on the line $x=0$, invariant under the standard map for parameter value k , marked in black. Undecided regions left white

$\times Ik$ into small enough rectangles on which the algorithm can succeed before precision is lost. This can be done automatically by a recursive procedure. The procedure, which we call RTEST, applies the above algorithm to a rectangle in (z, k) , iterating until successful or it has lost too much accuracy in following the orbits to be worth continuing. If successful, its job is done. Else if the rectangle in (z, k) is not too small it subdivides it into two, in z or in k according to which it thinks will be most helpful, and calls itself on each part. Else it gives up with a failure message.

To save time a non-rigorous version TEST is used first to determine rectangles on which RTEST is likely to succeed. All the main program NOICS does is to call TEST on the initial rectangle $Iz \times Ik$ for which one wants the result. Various parameters enter to determine the number of iterations to make before giving up, and which way to subdivide. The criteria for these are somewhat ad hoc, and their values were chosen simply by minimising the total time taken.

Preliminary explorations were made on a PERQ at Queen Mary College, London, though not with rigorous arithmetic. Figure 3 shows a typical output. The black rectangles are rectangles in (z, k) for which the program claimed there are no circles. The white ones are ones on which it failed to reach a decision. In this version of the program, subdivision was allowed down to a minimum width in z of

Table 1. Results of applying NOICS to the intervals between successive values of K and accumulating the totals. The first 9 cases were run with $wfx = 1e-13$, $cf = 0.135$, $zrfl = 0.995$, $QF = 0.01$, $BF = 50$, and the last 7 with $wfx = 1e-12$, $cf = 0.09$, $zrfl = 0.999$, $QF = 0.08$, $BF = 70$

K	Time (s)	Rtotal	Maxiter	Minzrø	Minkbr
1.5	—	—	—	—	—
1.0625	3.2	1,384	12	1.2e-4	2.7e-4
1.03125	15.5	6,964	20	3.0e-5	1.5e-4
1.015625	47	21,127	25	1.5e-5	7.7e-5
1.0078125	97	44,001	33	7.6e-6	3.8e-5
1.00390625	149	68,042	35	3.8e-6	3.8e-5
1.0	244	111,598	38	3.8e-6	1.9e-5
0.99609375	428	196,375	48	1.9e-6	9.7e-6
0.994140625	598	274,961	49	1.9e-6	9.7e-6
0.9921875	846	389,809	59	9.7e-7	9.7e-6
0.990234375	1,184	548,992	65	9.5e-7	4.8e-6
0.9892578125	1,440	669,369	65	9.5e-7	4.8e-6
0.98828125	1,764	821,961	73	4.7e-7	4.8e-6
0.9873046875	2,199	1,026,659	78	4.7e-7	4.8e-6
0.986328125	2,794	1,306,652	83	4.7e-7	2.4e-6
0.9853515625	3,631	1,701,540	83	2.3e-7	2.4e-6
0.984375	4,885	2,289,261	96	2.3e-7	2.4e-6

Rtotal is the total number of iterations required to prove non-existence of rotational invariant circles for $k \geq K$. Time is the total NAS 9080 time taken, or equivalent if run on the Amdahl 470/V7 (The NAS 9080 runs about 2.16 times as fast as the Amdahl 470/V7). Maxiter is the maximum number of iterations performed from any one initial parallelogram; the equivalent orbit length in Criterion 2 is $2 * \text{Maxiter} + 4$. Minzrø and Minkbr are the smallest half widths in z and $k/2\pi$ that were required

0.002, and in k of 0.003. The largest black rectangles were decided without any iteration, and the largest number of iterations required to decide a black rectangle was 7.

The final program was run on an Amdahl 470/V7 and NAS 9080 at CIRCE, Orsay, France. It is given in Appendix B. We believe that it is rigorous, as we have checked it pretty thoroughly. Should the reader find any mistakes, however, we would greatly appreciate being informed.

Rather than present the program directly with the whole interval $[63/64, 4/3]$ in k , as we did not know beforehand how strong a result it could prove nor how long it would take, we worked down from $k=4/3$ in steps. In fact, since $4/3$ is not representable in the floating point system we used, we started from $3/2$ instead. Table 1 shows the combined results. The net result is that there are no rotational invariant circles for

$$k \geq 63/64. \tag{5.26}$$

We could have gone considerably further, but it begins to take a lot of computer time. Extrapolating from the results of Table 1, it looks as if the time $T(K)$ the program takes to prove no rotational invariant circles for $|k| \geq K$ goes like

$$T(K) \propto (K - K_0)^{-\nu} \tag{5.27}$$

with $K_0 = 0.975 \pm 0.001$, $\nu = 3 \pm 0.25$. It is believed that there are no rotational invariant circles for

$$k > k_c \sim 0.971635, \tag{5.28}$$

so one might have expected this value for K_0 . However, the program uses a fixed precision (14 hexadecimal digits), and one probably needs unbounded precision to get arbitrarily close to k_c .

We do not have a good explanation for the exponent ν . Self-similarity about the point

$$k = k_c, \quad z = z_c \sim 0.405079193, \tag{5.29}$$

where the golden circle breaks [20], plus the assumption that it is the last circle to break, might seem to imply that it would take a factor

$$\gamma = \frac{1 + \sqrt{5}}{2} \tag{5.30}$$

longer to get a factor

$$\delta \cong 1.6279500, \tag{5.31}$$

closer in k to k_c , i.e.

$$\nu = \log_\delta \gamma \sim 0.98. \tag{5.32}$$

But this assumes we are in “scaling coordinates” [20]. In fact the orbits move with k , so we have to subdivide more in k . Thus it is reasonable that $\nu > 0.98$. There is probably room for improvement, however.

6. Orbits of Minimum Action

In this section, we describe the criteria of Aubry [4, 5] and Newman and Percival [6] for non-existence of invariant circles, and we show that they are essentially the same as Criterion 1. So this gives a second way of looking at the criterion.

The dynamics of area preserving twist maps has a variational formulation, the principle of stationary action. Given an area preserving twist map $T: (x, p) \mapsto (x', p')$, there exists a generating function $L(x, x')$, unique up to addition of a constant, such that

$$\begin{aligned} p' &= L_2(x, x'), \\ p &= -L_1(x, x'), \end{aligned} \tag{6.1}$$

where subscript i refers to the derivative with respect to the i^{th} argument. For example, (2.4) is generated by

$$L(x, x') = \frac{1}{2}(x - x')^2 + \int^x f(s) ds. \tag{6.2}$$

Generating functions always satisfy

$$L_{12}(x, x') \neq 0, \tag{6.3}$$

the sign being opposite to that of the twist. Conversely, given a function $L(x, x')$ satisfying (6.3), then the relations (6.1) can be inverted to define an area preserving twist map $T: (x, p) \mapsto (x', p')$.

From (6.1), if (x_t, p_t) is an orbit, then

$$p_t = L_2(x_{t-1}, x_t) = -L_1(x_t, x_{t+1}). \tag{6.4}$$

Thus

$$\frac{\partial}{\partial x_t} [L(x_{t-1}, x_t) + L(x_t, x_{t+1})] = 0, \tag{6.5}$$

and so orbit segments make the *action*

$$W_{mn} = \sum_{t=m}^{n-1} L(x_t, x_{t+1}) \tag{6.6}$$

stationary with respect to variations keeping the endpoints x_m, x_n fixed. Conversely, sequences (x_t) with stationary action generate orbits via (6.1).

With the convention

$$L_{12}(x, x') < 0 \tag{6.7}$$

a special role is played by orbits of minimum action, i.e. whose finite segments have minimal action (locally or globally) with respect to variations fixing the ends. In fact, Criterion 1 is equivalent to the following criterion conjectured by Newman and Percival [6], and closely related to a theorem of Aubry et al. [4, 5].

Criterion 1': *If an orbit segment is not a non-degenerate local minimum of the action, then it does not lie on any rotational invariant circle.*

Proof of Equivalence. An orbit segment X_0, \dots, X_t , where $X_r = (x_r, p_r)$, is a non-degenerate local minimum of the action iff the matrix

$$D^2W_{0t} = \left[\frac{\partial^2 W_{0t}}{\partial x_r \partial x_s} \right]_{r,s=1, \dots, t-1} \tag{6.8}$$

of second variations of the action is positive definite. A matrix is positive definite iff all its leading minors are positive. Thus Criterion 1' is equivalent to saying that the minors

$$M_t = \det D^2W_{0t} \tag{6.9}$$

are all positive, for orbits on a rotational invariant circle. The matrix D^2W_{0t} is tridiagonal, so we get the recurrence relation:

$$M_{t+1} = (L_{22}[t-1, t] + L_{11}[t, t+1])M_t - (L_{12}[t-1, t])^2 M_{t-1}, \tag{6.10}$$

where $[t, t+1]$ stands for (x_t, x_{t+1}) . One can generate the M_t from the initial conditions

$$M_0 = 0, \quad M_1 = 1. \tag{6.11}$$

Now consider Criterion 1. It says that for a tangent orbit to an orbit on a rotational invariant circle, with

$$\delta x_0 = 0, \quad \delta x_1 > 0, \tag{6.12}$$

then

$$\delta x_t > 0 \quad \forall t > 0. \tag{6.13}$$

The evolution of the tangent vectors is given by differentiating (6.4):

$$L_{21}[t-1, t]\delta x_{t-1} + (L_{22}[t-1, t] + L_{11}[t, t+1])\delta x_t + L_{12}[t, t+1]\delta x_{t+1} = 0. \tag{6.14}$$

Comparing the recurrence relations and initial conditions, we see that we can make the following correspondence between tangent vectors and minors:

$$M_t = \frac{\delta x_t}{\delta x_1} \prod_{s=1}^{t-1} (-L_{12}[s, s+1]) \quad t \geq 2. \tag{6.15}$$

Thus $\delta x_t > 0$ iff $M_t > 0$, proving the equivalence of the criteria. \square

This proof of equivalence answers most of the questions of Mather [2, 3] and Aubry [5] about the connection between Mather's method and that of Aubry et al. Aubry [5], however, actually proved a slightly different result from Criterion 1', which is stronger in some respects and weaker in others. Just before we submitted the manuscript, Mather [21] informed us that he has proved that "locally" can be replaced by "global" in Criterion 1'.

This formulation of the criterion gives an interesting piece of information. It tells us that Criterion 1 (or 2) will never succeed on any minimising orbit. Now there are other minimising orbits besides those on rotational invariant circles, e.g. a minimising periodic orbit for each rational rotation number, and orbits on Cantor sets for each irrational rotation number for which there are no rotational

invariant circles [12]. Thus even if there are no rotational invariant circles, one cannot expect the criterion to succeed everywhere. This problem was not evident in Sects. 3 and 5, because we examined only the line $x = 0$, and apparently there are no points on this line with minimising orbits for $k \geq 63/64$. In fact we conjecture that for $k \geq 0$ all points on $x = 0$ with minimising orbits lie on rotational invariant circles. Newman and Percival [6] conjecture that the set of points through which there is no rotational invariant circle but for which the criterion will never succeed has measure zero. The viewpoint of the next section will suggest a subsidiary criterion “killends”, which removes some of these points. We conjecture that it can remove any of them (Sect. 9).

7. Conefields

The criterion has so far always been formulated in terms of following orbits. But one could equally well stand still and let them go by, i.e. rewrite (3.24) as iterative improvement of a whole field of directions $D_+(X)$:

$$D_+(X) := \text{u.b. } DT_{T^{-1}X} \cdot D_+(T^{-1}X) \tag{7.1}$$

starting from the uniform field $D_+(X) = D_+^0$. The distinction is the discrete time analogue of that between the Lagrangian and Eulerian viewpoints for fluid flow. One can do the same for the inverse map

$$D_-(X) := \text{l.b. } DT_{TX}^{-1} \cdot D_-(TX) \tag{7.2}$$

starting from $D_-(X) = D_-^0$.

The direction fields $D_{\pm}(X)$, together called a *cone*field, have an interpretation as local Lipschitz constants for any rotational invariant circle through $X = (x_0, z_0)$, if there exist such:

$$D_-(X) \leq \liminf_{x \rightarrow x_0} \frac{z - z_0}{x - x_0} \leq \limsup_{x \rightarrow x_0} \frac{z - z_0}{x - x_0} \leq D_+(X). \tag{7.3}$$

This is because if $D_{\pm}(X)$ are local Lipschitz constants for any rotational invariant circle then so are the new $D_{\pm}(X)$ produced by (7.1) and (7.2).

Now suppose that at some stage $D_+(X)$ becomes less than $D_-(X)$. Then the range of (7.3) becomes empty, so there can be no circle through X . This gives a third way of looking at the criterion:

Criterion 3. *Start with any cone*field $D_{\pm}(X)$ containing D_{\pm}^0 (e.g. one could take $\infty, 0$ in (x, z) coordinates). Improve it by (7.1), (7.2). If at any stage $D_-(X) > D_+(X)$, then there can be no rotational invariant circle through X .

For example, Mather’s result that there are no circles for (3.5) if (3.27) follows from one improvement of D_+ on the line $z = z_0$, where $g'(z)$ is minimum.

We now describe one implementation of Criterion 3, which can be used to get stronger results. In order to perform the iterative step (7.1), (7.2) in finite time, one has to restrict $D_{\pm}(X)$ to some class of fields with finite specification. One could represent $D_{\pm}(X)$ by something smooth like finite Fourier series, but this requires

criteria for determining extremes of functions that are difficult to implement on the computer. So we took $D_{\pm}(X)$ to be piecewise constant, being constant on the elements of some finite partition, and taking values in a finite set, namely the representable floating point numbers on a computer.

We applied Criterion 3 to maps of the form (3.5), using cone-fields constant on the elements of an $n \times n$ square partition of the (x, z) -plane. Provided $g(z)$ is non-decreasing (as for the standard map when $|k| \leq 2$), the pre-image of the square

$$S_{ij} = \left[\frac{i}{n}, \frac{i+1}{n} \right] \times \left[\frac{j}{n}, \frac{j+1}{n} \right] \tag{7.4}$$

is bounded by the rectangle

$$\left[g\left(\frac{i}{n}\right) - \frac{j+1}{n}, g\left(\frac{i+1}{n}\right) - \frac{j}{n} \right] \times \left[\frac{i}{n}, \frac{i+1}{n} \right] \tag{7.5}$$

(see Fig. 4).

Let

$$K_i^+ = \text{round up} \left(n * g\left(\frac{i+1}{n}\right) \right) - 1, \tag{7.6}$$

$$K_i^- = \text{round down} \left(n * g\left(\frac{i}{n}\right) \right) - 1, \tag{7.7}$$

where the rounding is performed to the nearest integer in the given direction. Then

(7.5) is contained in the union $\bigcup_{m \in M_{ij}} S_{mi}$ of squares with

$$M_{ij} = K_i^- - j \text{ to } K_i^+ - j. \tag{7.8}$$

Use of the reversor $S(x, z) = (-z, -x)$ and the symmetry $(x, z) \rightarrow (-x, -z)$ (Appendix A) gives the same bounds on images of squares, but with the indices interchanged.

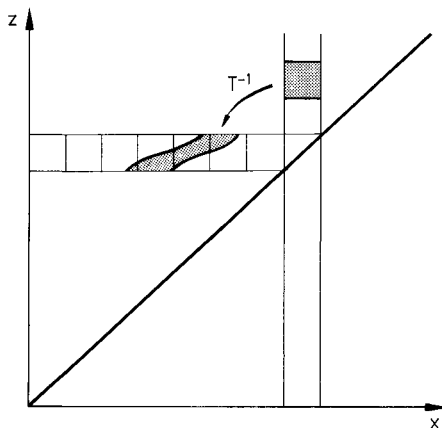


Fig. 4. Bounding the preimage of a square

Let

$$R_i = \text{u.b.}_{x \in [\frac{i}{n}, \frac{i+1}{n}]} g'(x). \tag{7.9}$$

Then (7.1) can be implemented as

$$D_{ij}^+ := \text{u.b.} \left[R_i - \frac{1}{\max_{m \in M_{ij}} D_{mi}^+} \right], \tag{7.10}$$

and (7.2) as

$$E_{ij}^- := \text{u.b.} \left[R_j - \frac{1}{\max_{m \in M_{jt}} E_{jm}^-} \right], \tag{7.11}$$

where

$$E_{ij}^- = 1/D_{ij}^-. \tag{7.12}$$

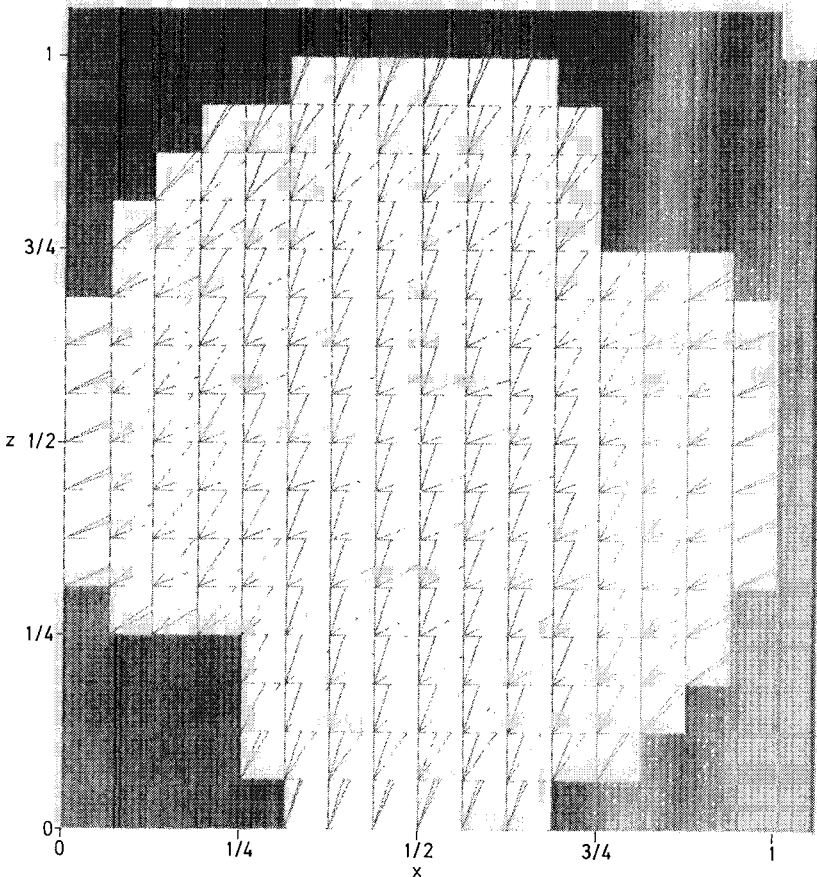


Fig. 5. Conefield for rotational invariant circles of the standard map at $k \sim 1.29$. No circles pass through the black regions

Any squares for which

$$D_{ij}^+ E_{ij}^- < 1 \tag{7.13}$$

can have no rotational invariant circles passing through them, and so can be deleted.

Clearly this can be implemented rigorously. All one needs is bounds on g and g' , and on the operations of reciprocation, subtraction and multiplication. We did not take the trouble to do so, however, because the implementation of Criterion 2 turned out to be more efficient.

Figures 5 and 6 show results obtained for the standard map by iterating (7.10), (7.11) until no further change was observed (half a dozen iterations suffice). Our implementation included special “squares” along the top and right-hand edges, to represent intervals on the edges $x=0$ and $z=0$, since these appear to be the most sensitive places. In Fig. 5, for $k \sim 1.29$, we see that all the special edge squares have been deleted. Since every rotational invariant circle must cross the edges there can be none.

Figure 6, for $k \sim 1.05$, allows rotational invariant circles to exist in the region where the cones still exist. If, following all slopes allowed by the cone field in one

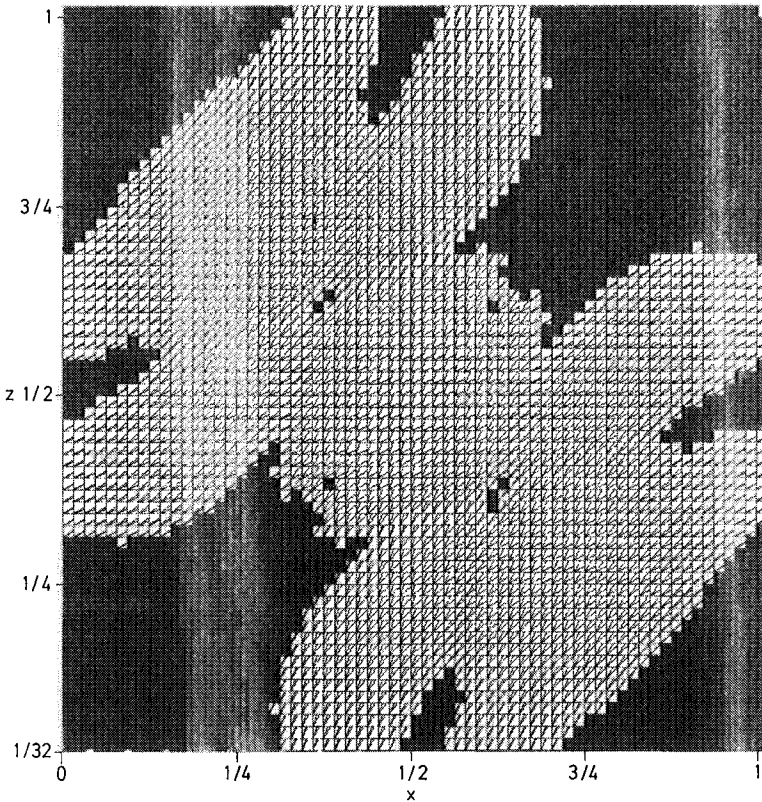


Fig. 6. Conefield for rotational invariant circles of the standard map at $k \sim 1.05$. No circles pass through the black regions

direction or the other from an allowed point, however, one necessarily enters a region through which no rotational invariant circles pass, then there can be no rotational invariant circles through that point. This allows one to delete more squares. For example, it reduces the region of Fig. 6, where rotational invariant circles can exist to that of Fig. 7. It removes everything in Fig. 5, of course. We call this supplementary criterion for non-existence of circles, "kill-ends".

At first sight, one can only use killends in conjunction with version 3 of the criterion, because it needs the Lipschitz cone field, but a weaker version of killends can be used with the earlier versions of the criterion, using just the initial explicit Lipschitz cone as in (3.18) and (3.19). Note that the version of killends used in Fig. 7 did not take into account the special edge squares. If it had, it would have deleted some more squares, as you can see. Also note that, unfortunately, the bottom two rows of Figs. 6 and 7 got lost in the hardcopy.

As discussed in Sect. 6, some supplementary criterion like killends is necessary if one is to separate orbits on rotational invariant circles from general minimising orbits. One might ask what the cone field represents on minimising orbits not lying on invariant circles. If the orbit is hyperbolic and hence has stable and unstable directions, then the cone field represents bounds on these invariant directions, because all directions outside the cones eventually get pushed inside under

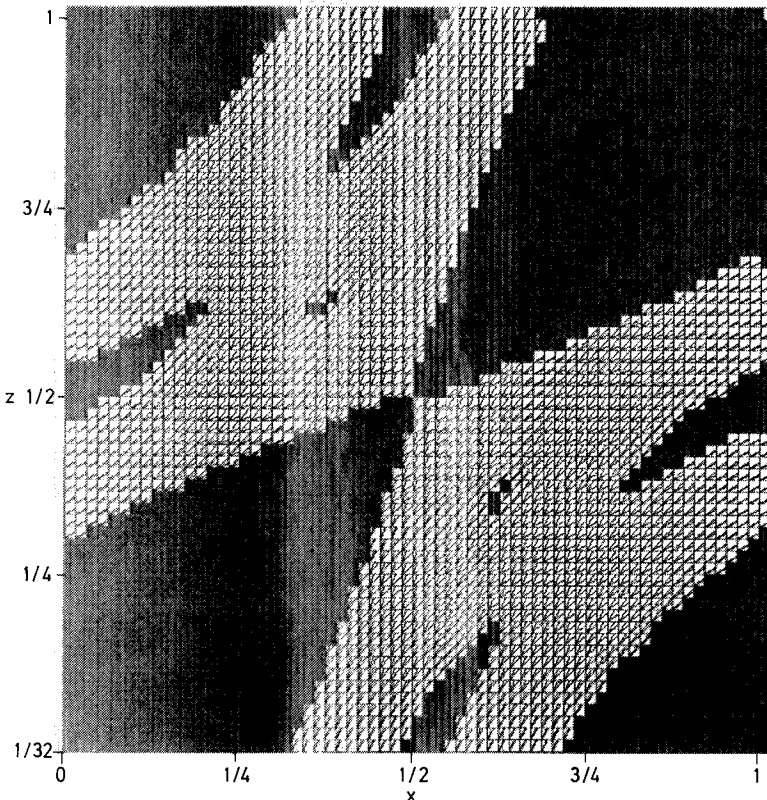


Fig. 7. Same as Fig. 6 after application of subsidiary criterion "killends"

iteration. Note that non-degenerate locally minimising periodic orbits are always hyperbolic [13].

The main disadvantage of this method is that to get strong results it is necessary to subdivide a lot, so it requires a lot of space to store the cone-fields. One can save some space if the map possesses symmetries (Appendix A). For example, when g is odd (as for the standard map) one can use the reversor $S(x, z) = (-z, -x)$ to put

$$E_-(-z, -x) = D_+(x, z). \tag{7.14}$$

Furthermore, commutation with $(x, z) \rightarrow (-x, -z)$ implies that

$$D_{\pm}(-x, -z) = D_{\pm}(x, z). \tag{7.15}$$

This enabled us to cut down the storage and computing time by a factor of $\frac{1}{4}$, though we didn't actually use it for Figs. 5–7. Killends does remove a lot of squares, but we thought of no easy way to compress the remaining information. The main reason that one has to subdivide a lot is that the method does not follow orbits very accurately. It requires an average of 4 squares to cover the image of one. To retard loss of precision one could use other classes of functions for D_{\pm} , e.g. functions which are linear on elements of some partition (triangles would be better than squares for this), but we didn't try any.

It is clear that this criterion could also handle a whole C^1 -ball of maps at once, since all one needs is bounds on the map and its derivative.

8. Other Techniques

We review here, for comparison, other techniques, whether or not rigorously based, for indicating regions through which pass no rotational invariant circles.

(a) Residue Criterion

The best practical way to see where there are circles and where there are none is Greene's residue criterion [10]. The criterion postulates a relation between existence of invariant circles and stability properties of certain periodic orbits. For each rational rotation number m/n , there is a pair of periodic orbits, known as a "resonance." Their linear stability can be measured by quantities known as their "residues" R^{\pm} . Two resonances are said to be *neighbouring* if their rotation numbers $m/n, m'/n'$ satisfy

$$mn' - m'n = 1. \tag{8.1}$$

The strongest form of Greene's criterion [19] is that there are some/no invariant circles between neighbouring resonances if the residues R^{\pm} are significantly less/greater in magnitude than

$$R_{*}^{+} \sim 0.250, \quad R_{*}^{-} \sim -0.255. \tag{8.2}$$

It can be proved that if m_k/n_k are successive continued fraction approximations to a Diophantine number ν , and if the residues R_k^{\pm} of the corresponding resonances do not tend to zero as $k \rightarrow \infty$, then there is no smooth invariant circle of rotation number ν . On its own, this cannot provide rigorous results, because limiting

behaviour is not rigorously establishable with a finite amount of computation. It may be, however, that Criterion 2 could be used to save one having to go to the limit, as will be discussed in Sect. 8(b).

(b) Island Overlap

Chirikov [11] developed an approximate heuristic criterion for nonexistence of invariant circles between two resonances, based on estimates of the widths of resonances from standard canonical perturbation theory. If the estimated widths make the islands overlap, then it is unlikely that there are any invariant circles of the appropriate class between them. If they don't overlap, there probably are circles between them. This criterion is closely linked with the residue criterion, since island width is proportional to residue in first order perturbation theory.

The methods of our paper suggest a possible way of making the island overlap criterion rigorous. In the islands, the map looks like a rotation, so the tangent vectors keep getting rotated. Thus there is a neighbourhood of the central periodic orbit through which the criterion proves that there are no rotational circles. If one could establish large enough neighbourhoods then there may be suitable quantities to estimate, e.g. residue, which, if they come out large enough would guarantee non-existence of circles in between.

(c) Difference in Action

Mather introduced a quantity $\Delta W_\nu \geq 0$, for any $\nu \in \mathbb{R}$, called a difference in action [18], which corresponds to a flux across certain curves containing orbits of rotation number ν [17]. If $\Delta W_\nu = 0$, then there is an invariant circle of rotation number ν (cf. confinement corollary, Sect. 2). Furthermore, for ν irrational there is no such circle if $\Delta W_\nu \neq 0$.

Mather proved that ΔW_ν is continuous at irrational ν . Thus if it is non-zero at some irrational ν then one might be able to prove it so by estimates on ΔW_ν at nearby rationals, which could be obtained with a finite amount of computation. To do this requires an explicit module of continuity, however, which is not yet known.

(d) Climbing Orbits

If one finds an orbit segment which climbs the cylinder sufficiently, then one can rule out existence of invariant circles separating the endpoints. This is a technique used by Chirikov [11] and Escande and Doveil [16], for example. We can formulate it rigorously as follows: If L is a global Lipschitz constant, as in Sect. 3, and one finds an orbit (x_n, p_n) such that for some n :

$$|p_n - p_0| > L|x_n - x_0| \tag{8.3}$$

then there are no rotational invariant circles crossing the straight line between (x_0, p_0) and (x_n, p_n) . For example, the standard map has an "accelerator mode" for $|k| \geq 2\pi$ [11], which is an orbit which climbs vertically by one every iteration, so there are no rotational invariant circles for $|k| \geq 2\pi$.

This method, however, requires very long orbit segments to decide delicate cases. It would be hard to follow such long orbits rigorously. The length could be

reduced by careful choice of initial condition, viz. in the “turnstiles” [17] of the strongest cantori, but would still require longer orbits than Criterion 2.

(e) *Net Flux*

For area preserving maps on the cylinder there may be a net flux [17], also known as the Calabi invariant. It is defined as the net flux across any rotational circle (not necessarily invariant), the result being independent of the choice of circle, by the area preserving property. If it is non-zero there can be *no* invariant rotational circles, yet the test can be made for *any* rotational circle. For example, for maps of the form (2.4) the net flux is

$$\int_0^1 f(x)dx . \tag{8.4}$$

Clearly, if the net flux is non-zero, then one can prove it so rigorously by precise enough integration.

It should be noted that the flux has to be measured appropriately. For example, when (x, z) coordinates are used, as in Sect. 3, the opposite sides of the cylinder are identified after a vertical shift of 1, so a rotational circle of the cylinder looks like a helix if the wrong identification is made.

The net flux criterion needs to be used only once if dealing with a single map. If the flux is non-zero there are no rotational invariant circles and no need for further tests. If it is zero nothing is achieved and other tests are necessary.

9. Further Discussion

(a) *Finite Computation Conjecture*

Given a region through which no rotational invariant circles pass, a very interesting question is whether the criteria in this paper can be used to prove this, with a finite amount of computation. We conjecture that they can. Let $A(M)$ be the space of all C^1 area preserving, end preserving, orientation preserving twist diffeomorphisms of the cylinder $M = S^1 \times \mathbb{R}$.

Conjecture. Given any compact subset B of $M \times A(M)$, such that for any $(X, T) \in B$, the map T has no rotational invariant circles through point X , then the combination of

- (i) the net flux test (Sect. 8(e)),
- (ii) cone-crossing (Criterion 1, 1', 2 or 3),
- (iii) kill-ends (Sect. 7),

will prove so with a finite amount of computation.

After this paper was submitted, Stark [22], with a slightly stronger version of kill-ends, and a careful definition of precision, has in fact proved this conjecture.

(b) *Generalisations*

There is nothing special about the vertical. If one has a foliation equivalent to the vertical, which is twisted by T , that is, the images of the leaves cross the leaves at a non-zero angle, then by coordinate change Birkhoff's theorem tells us that every rotational invariant circle cuts each leaf of the foliation precisely once.

Suppose there are two foliations, one twisted clockwise by T , the other anticlockwise by T^{-1} , e.g. the vertical and horizontal in (x, z) coordinates for (3.5). The horizontals cross the verticals once only, say from left to right. Every rotational invariant circle can cut each leaf only once. Then one could produce new such pairs of foliations which restrict invariant circles more, by taking their images under T and T^{-1} respectively, and this process can be iterated. If at some point a pair of foliations crosses the wrong way round, then any rotational circle through that point would have to cut one of the leaves through that point at least once more, so there can be no rotational invariant circles through that point.

The implementation of this criterion would look very much like that of Sect. 7 for improving cone-fields. The integral curves of the cone-fields produced in Fig. 5, for example, do not give a C^1 -foliation, but Fathi [9] has disposed of the differentiability conditions. A homeomorphism is said to twist a foliation if the image of each leaf cuts each leaf topologically transversally.

The interesting thing is that this formulation suggests an automatic extension to other classes of circles, going round fixed points or periodic points. One can always treat each class separately, by considering the appropriate iterate of the map, if there are coordinates in which it is a twist map. But with the present method this may not be necessary. Perhaps, once the foliations have crossed the wrong way, they will provide restrictions on circles of other classes. This might require keeping track of the number of half revolutions the foliations make. Birkhoff's theorem has a generalisation [3] which will probably cope with this: the twist condition can be relaxed to a *tilt condition*, allowing non-transverse intersection provided the number of half revolutions made so far is always positive or always negative.

(c) Continuous Time and Higher Dimensions

Autonomous Hamiltonian systems with two degrees of freedom, and one degree of freedom systems with Hamiltonian periodic in time, can be reduced to area preserving maps by using Poincaré's surfaces of section. Our results should lift back to these systems, most directly via the action principle of Sect. 6. We would expect Criterion 1' to apply directly.

For systems of more degrees of freedom, we would expect to have a vector of slopes, one for each $dx_i \wedge dp_i$, and it would be interesting to develop this idea.

(d) Summary of Results

We have obtained rigorous and practical criteria for proving the non-existence of rotational invariant circles in domains of phase \times parameter space for area preserving twist maps. As an example we have proved that the "standard map" has no rotational invariant circles for any parameter value $|k| \geq 63/64$, which is within 0.013 of the generally accepted optimal but non-rigorous value, obtained by the method of Greene [10].

We have also shown how to obtain Lipschitz bounds on the slopes of invariant circles that might remain in the phase space and illustrated the results graphically.

The criteria are obtained by applications and extension of theorems and methods of Birkhoff, Herman, Mather, and Aubry.

It appears to be easier to obtain rigorous bounds on the domain of phase space occupied by invariant circles of a given class by using converse KAM theory, where

regions are found through which no rotational invariant circles pass, as in this paper, than by using KAM theory, where regions are found guaranteed to contain invariant circles.

Appendix A: Symmetries

There are two notions of symmetry that we use. One is possession of a symmetry group. The other is reversibility.

Say a group G of maps is a *symmetry group* for a map T if

$$\forall g \in G \exists g' \in G \text{ such that } gT = Tg'. \tag{A.1}$$

Then as far as T is concerned, one can identify points if they are related by an element of G , i.e. identify x and y if

$$\exists g \in G \text{ such that } g(x) = y. \tag{A.2}$$

For example, the group generated by the unit horizontal and vertical translations is a symmetry group for (2.4), so (2.4) can be regarded as acting on the unit square, with edges identified to form a torus. If f in (2.4) is odd, then the symmetry group can be enlarged by including the reflection

$$\left. \begin{matrix} x \rightarrow -x \\ p \rightarrow -p \end{matrix} \right\} \tag{A.3}$$

as a generator. So (2.4) can be regarded as acting on the rectangle

$$0 \leq x \leq 1, \quad 0 \leq p \leq \frac{1}{2}, \tag{A.4}$$

with edges identified to form a sphere. Thus, in particular, to find all rotational invariant circles of (2.4), it suffices to examine the interval $0 \leq p \leq \frac{1}{2}$ on $x = 0$.

Next we define reversibility. Say a map S is a *reversor* for T if it is an involution, i.e.

$$S^2 = \text{identity} \tag{A.5}$$

reverses orientation, and conjugates T to its inverse, i.e.

$$STS^{-1} = T^{-1}. \tag{A.6}$$

For example, when f is odd, (2.4) possesses the reversor

$$\left. \begin{matrix} p' = p + f(x) \\ x' = -x, \end{matrix} \right\} \tag{A.7}$$

and (3.5) the reversor

$$S(x, z) = (-z, -x). \tag{A.8}$$

Finally, note that the standard map is conjugate to itself under changing the sign of k , by the translation $x \rightarrow x + \frac{1}{2}$. So it suffices to consider $k \geq 0$.

Appendix B: Computer Program

In this appendix we present the computer program which implemented the algorithm of Sect. 5, plus a sample output.

A couple of remarks about the program are in order here. Firstly, it is written in Pascal, which is the nicest language for rigorous programs, because the compiler is very careful not to let you do things you did not intend to, though the version at CIRCE unfortunately lacks many desirable features. It is described in CIRCE's PASCAL manual [14], and has been used sufficiently that one hopes can trust the manual.

Secondly, in order to write rigorous error bounding arithmetic, one either has to write one's own floating point routines: UPSUM, DOWNSUM etc., returning rigorous bounds on sums, products etc., or one has to find out how arithmetic is done on the machine and modify the answers to get rigorous bounds, e.g. define functions low, inc, up, and down such that:

- i) the true product (quotient) of two numbers A and B lies between $\text{low}(A * B)$ and $\text{inc}(A * B)$ [$\text{low}(A/B)$ and $\text{inc}(A/B)$],
- ii) the true sum (difference) of two numbers A and B lies between $\text{down}(A \pm B)$ and $\text{up}(A \pm B)$.

The former is the most satisfactory as one can be sure exactly what is going on, but the latter is much easier and faster running, although one is forced to trust the manuals. We adopted the latter. The machines we used are supposed to be completely IBM 370 compatible. IBM 370 floating point arithmetic is well described in the IBM 370 Principles of Operation Manual [15]. On IBM 370 machines the function "low" can be taken to be the identity, so we did not define it. We could have written the program in a machine independent way, apart from the definition of these functions, but we also took advantage of other features deduced from the Principles of Operation Manual, which are commented on where they appear.

Constant, Type and Variable Declarations

```
(*T- NO EXECUTION TESTS: THE ONLY ONE THAT MIGHT HAVE MATTERED IS DETECTION
OF OVERFLOW ON INTEGER MULTIPLICATION, WHICH IS NOT DETECTED BY THE
HARDWARE, BUT ONE CAN EASILY A PRIORI BOUND THE INTEGERS OCCURING IN
INTEGER MULTIPLICATION IN THIS PROGRAM AND SEE THAT OVERFLOW WILL NEVER
OCCUR *)

PROGRAM NOICS(INPUT,OUTPUT);

TYPE INTPAIR=RECORD I1,I2:INTEGER END;
STRING4=PACKED ARRAY(.1..4.)OF CHAR;
STRING8=PACKED ARRAY(.1..8.)OF CHAR;
STRING16=PACKED ARRAY(.1..16.)OF CHAR;
INTERVAL=RECORD L,R:REAL END; (*SO INTERVAL I IS I.L,I.R*)

VAR MAXNEG,N16,N255,ORDA.ORDA10:INTEGER;
TWOPI,TWOPII:REAL;
ITWOPI,ITWOPII:INTERVAL;

WFCOUNTER,NRECT,RNRECT,RNFAILRECT,NUSEFULRECT,
RTOTAL,USEFULTOTAL,TOTAL,RFAILURES,RMAXITER,N,NMAX,CLOCKO:INTEGER;
WF,WFX,BF,QF,ZRMAXO,CFACTOR,MINZRO,MINKBR,ZRF1:REAL;
IZO,IK,IWORSTZ,IWORSTK,IMINZZ,IMINKZ,IMINZK,IMINKK:INTERVAL;

C,OPTION,OK:INTEGER;
XC,ZC,ZR,D,ZCO,ZRO,ZRMAX,K1,DO,TEMP,S,KM,ZCN,KB,
Q22,W1,W2,DET,QP12,
B,E,COL1,COL2,M10,M12,M21,HQ10,ADET,TEMP2,QP21,QP22,
KBR,Q10,Q11,Q12,Q20,Q21,QP10,QP11,M20,DZC:REAL;
IQ10,IQ11,IQ12,IM21,IM22,IC,IGZ,IZ,IKB,IS,IZ1,IK1,
IM11,IM12,IDET,IZC,ITEMP,ITEMP2:INTERVAL;
```

Rounding Routines

```
(*CONVENTION: LARGE/SMALL, INC/DEC REFER TO ABSOLUTE VALUE,
HIGH/LOW, UP/DOWN REFER TO ORDER ON REALS *)
```

```
PROCEDURE INITROUND; (*REQUIRED ONLY BECAUSE THIS COMPILER IS SO INFLEXI
BEGIN
  N16:=16;
  N255:=255;
  ORDO:=ORD('O');
  ORDA10:=ORD('A')-10;
  MAXNEG:=PRED(-MAXINT)
END;
```

```
FUNCTION INC(X:REAL):REAL;
(* FOR X<>0 THIS RETURNS THE NEXT LARGER REPRESENTABLE NUMBER, OR HALTS
IN CASE OF OVERFLOW. NOT TO BE APPLIED TO X=0 *)
VAR X1:INTPAIR;
S:STRING4;
BEGIN
  X1:=INTPAIR(X);
  WITH X1 DO
    IF I2=MAXINT THEN I2:=MAXNEG
    ELSE BEGIN
      I2:=SUCC(I2);
      IF I2=0 THEN
        IF (I1=MAXINT)OR(I1=-1) THEN HALT(' OVERFLOW IN INC')
        ELSE BEGIN
          I1:=SUCC(I1);
          S:=STRING4(I1);
          IF ORD(S(.2.))=0 THEN S(.2.):=CHAR(N16);
          I1:=INTEGER(S) END END;
        INC:=REAL(X1)
      END;
```

```
FUNCTION DEC(X:REAL):REAL;
(* FOR X<>0 THIS RETURNS THE NEXT SMALLER REPRESENTABLE NUMBER, OR HALT!
WITH AN UNDERFLOW ERROR. NOT TO BE APPLIED TO X=0 *)
VAR S:STRING4;
ORDS:INTEGER;
X1:INTPAIR;
BEGIN
  X1:=INTPAIR(X);
  WITH X1 DO
    IF I2=MAXNEG THEN I2:=MAXINT
    ELSE IF I2=0 THEN BEGIN
      I2:=-1;
      I1:=PRED(I1);
      S:=STRING4(I1);
      IF ORD(S(.2.))=15 THEN BEGIN
        ORDS:=ORD(S(.1.));
        IF (ORDS=0)OR(ORDS=128) THEN HALT(' UNDERFLOW IN DEC')
        ELSE BEGIN
          S(.1.):=PRED(S(.1.));
          S(.2.):=CHAR(N255) END;
          I1:=INTEGER(S) END END
      ELSE I2:=PRED(I2);
      DEC:=REAL(X1)
    END;
```

```
FUNCTION UP(X:REAL):REAL;
(* FOR X<>0 THIS RETURNS THE NEXT HIGHER REP. UP(0)=0 *)
BEGIN
  IF X>0 THEN UP:=INC(X)
  ELSE IF X<0 THEN UP:=DEC(X)
  ELSE UP:=0
END;
```



```

FUNCTION DOWN(X:REAL):REAL;
(* FOR X<>0 THIS RETURNS THE NEXT LOWER REP. DOWN(0)=0 *)
BEGIN
  IF X>0 THEN DOWN:=DEC(X)
  ELSE IF X<0 THEN DOWN:=INC(X)
  ELSE DOWN:=0
END;

```

Hexadecimal Output Routines

```

FUNCTION HEX(M:INTEGER):CHAR;
BEGIN
  IF M<10 THEN HEX:=CHAR(ORDO+M)
  ELSE HEX:=CHAR(ORDA10+M)
END;

PROCEDURE WRITEHEX(X:REAL);
VAR I:INTEGER;
    S:STRING8;
    T:STRING16;
BEGIN
  S:=STRING8(X);
  FOR I:=1 TO 8 DO BEGIN
    T(.2*I-1.):=HEX(ORD(S(.I.)) DIV 16);
    T(.2*I.):=HEX(ORD(S(.I.)) MOD 16) END;
  WRITE(' ',T)
END;

```

Simple Functions

```

FUNCTION ROUNDDOWN(X:REAL):INTEGER;
(* ROUNDS X DOWN TO THE NEAREST INTEGER *)
VAR T:INTEGER;
BEGIN
  IF X>=0 THEN ROUNDDOWN:=TRUNC(X)
  ELSE BEGIN
    T:=TRUNC(X);
    IF T=X THEN ROUNDDOWN:=T
    ELSE ROUNDDOWN:=T-1 END
END;

FUNCTION ROUNDUP(X:REAL):INTEGER;
(* ROUNDS X UP TO THE NEAREST INTEGER *)
VAR T:INTEGER;
BEGIN
  IF X<=0 THEN ROUNDUP:=TRUNC(X)
  ELSE BEGIN
    T:=TRUNC(X);
    IF T=X THEN ROUNDUP:=T
    ELSE ROUNDUP:=T+1 END
END;

FUNCTION MAX(A,B:REAL):REAL;
BEGIN
  IF A>B THEN MAX:=A
  ELSE MAX:=B
END;

FUNCTION MIN(A,B:REAL):REAL;
BEGIN
  IF A<B THEN MIN:=A
  ELSE MIN:=B
END;

FUNCTION USQRT(A:REAL):REAL; (*RETURNS U.B. ON SQRT(A) *)
VAR X,E,X2:REAL;
BEGIN
  X:=SQRT(A); (*APPROXIMATE SQRT *)
  X2:=X*X;
  IF X2<A THEN BEGIN
    E:=INC(A-X2);
    USQRT:=INC(X+INC(E/(2*X))) END
  (* USING D/DX(X*X)=2*X, INCREASING FOR X>0 *)
  ELSE USQRT:=X
END;

```

Interval Arithmetic Routines

```

FUNCTION RMAXERROR(I:INTERVAL;Z:REAL):REAL; (* U.B. ON ABS(Z-I) *)
BEGIN RMAXERROR:=UP(MAX(ABS(I.R-Z),ABS(I.L-Z))) END;

FUNCTION IABS(I:INTERVAL):REAL; (* MAX(ABS(I)) *)
BEGIN IABS:=MAX(ABS(I.L),ABS(I.R)) END;

FUNCTION RIMIN(I:INTERVAL):REAL;
BEGIN
  IF I.L*I.R<=0 THEN RIMIN:=-1
  ELSE RIMIN:=MIN(ABS(I.L),ABS(I.R))
END;

(* THE FOLLOWING WOULD BE BEST DEFINED AS FUNCTIONS, BUT THIS
  COMPILER DOES NOT ALLOW FUNCTIONS TO RETURN RECORDS OR ARRAYS *)

PROCEDURE RISDIFF(I:INTERVAL;S:REAL;VAR J:INTERVAL);
(* RETURNS AN INTERVAL CONTAINING I-S *)
BEGIN J.L:=DOWN(I.L-S);J.R:=UP(I.R-S) END;

PROCEDURE RSIDIFF(S:REAL;I:INTERVAL;VAR J:INTERVAL);
(* RETURNS AN INTERVAL CONTAINING S-I *)
BEGIN J.L:=DOWN(S-I.R);J.R:=UP(S-I.L) END;

PROCEDURE RIDIFF(I,J:INTERVAL;VAR K:INTERVAL);
(* RETURNS AN INTERVAL CONTAINING I-J *)
BEGIN K.L:=DOWN(I.L-J.R);K.R:=UP(I.R-J.L) END;

PROCEDURE RISPROD(I:INTERVAL;S:REAL;VAR J:INTERVAL);
(* RETURNS AN INTERVAL CONTAINING I*S *)
(* COULD SAVE A LITTLE ACCURACY BY EXAMINING SIGNS AND USING ONLY INC *)
BEGIN
  IF S>=0 THEN BEGIN
    J.L:=DOWN(I.L*S);J.R:=UP(I.R*S) END
  ELSE BEGIN
    J.L:=DOWN(I.R*S);J.R:=UP(I.L*S) END
END;

PROCEDURE RIPRODP(I,J:INTERVAL;VAR K:INTERVAL);
(* RETURNS AN INTERVAL CONTAINING I*J, ASSUMING I>0 *)
BEGIN
  IF J.L<0 THEN K.L:=INC(I.R*J.L)
  ELSE K.L:=I.L*J.L;
  IF J.R<=0 THEN K.R:=I.L*J.R
  ELSE K.R:=INC(I.R*J.R)
END;

PROCEDURE RIFATTEN(S:REAL;VAR I:INTERVAL);
(*IF S IS A PRODUCT OR QUOTIENT OF REALS, THIS RETURNS AN INTERVAL
  CONTAINING THE TRUE ANSWER*)
BEGIN
  IF S>0 THEN BEGIN
    I.L:=S;I.R:=INC(S) END
  ELSE IF S<0 THEN BEGIN
    I.L:=INC(S);I.R:=S END
  ELSE BEGIN
    I.L:=0;I.R:=0 END
END;

```

Nonrigorous Interval Arithmetic

```

(* NONRIGOROUS ROUTINES FOR QUICK AND DIRTY ESTIMATES *)

FUNCTION ICENTRE(I:INTERVAL):REAL; (*APPROXIMATE CENTRE OF INTERVAL*)
BEGIN ICENTRE:=I.L+0.5*(I.R-I.L) END;

FUNCTION IHWIDTH(I:INTERVAL):REAL;
BEGIN IHWIDTH:=0.5*(I.R-I.L) END;

```

Nonrigorous Interval Arithmetic (continued)

```

PROCEDURE ISOIFF(I:INTERVAL;S:REAL;VAR J:INTERVAL);
BEGIN J.L:=I.L-S;J.R:=I.R-S END;

PROCEDURE SIDIFF(S:REAL;I:INTERVAL;VAR J:INTERVAL);
BEGIN J.L:=S-I.R;J.R:=S-I.L END;

PROCEDURE IDIFF(I,J:INTERVAL;VAR K:INTERVAL);
BEGIN K.L:=I.L-J.R;K.R:=I.R-J.L END;

PROCEDURE ISPROD(I:INTERVAL;S:REAL;VAR J:INTERVAL);
BEGIN
  IF S>=0 THEN BEGIN
    J.L:=I.L*S;J.R:=I.R*S END
  ELSE BEGIN
    J.L:=I.R*S;J.R:=I.L*S END
END;

PROCEDURE IPRODP(I,J:INTERVAL;VAR K:INTERVAL);
BEGIN
  IF J.L<0 THEN K.L:=I.R*J.L
  ELSE K.L:=I.L*J.L;
  IF J.R<0 THEN K.R:=I.L*J.R
  ELSE K.R:=I.R*J.R
END;

PROCEDURE ICOSTP(Z:INTERVAL;VAR C:INTERVAL);
VAR NL,NR,DN:INTEGER;
BEGIN
  NL:=ROUNDDOWN(2*Z.L);NR:=ROUNDUP(2*Z.R);
  DN:=NR-NL;
  IF DN<=1 THEN
    IF ODD(NL) THEN BEGIN
      C.L:=COS(TWOPI*Z.L);C.R:=COS(TWOPI*Z.R) END
    ELSE BEGIN
      C.L:=COS(TWOPI*Z.R);C.R:=COS(TWOPI*Z.L) END
  ELSE IF DN=2 THEN
    IF ODD(NL) THEN BEGIN
      C.R:=1;C.L:=-COS(TWOPI*MIN(Z.L-NL/2,NR/2-Z.R)) END
    ELSE BEGIN
      C.L:=-1;C.R:=COS(TWOPI*MIN(Z.L-NL/2,NR/2-Z.R)) END
  ELSE BEGIN
    C.L:=-1;C.R:=1 END
END;

```

Trigonometric Routines

(* FOR X SMALL ENOUGH, THE SERIES FOR SIN AND COS ARE ALTERNATING AND DECREASING, HENCE ALTERNATE PARTIAL SUMS PROVIDE UPPER AND LOWER BOUNDS. IN PARTICULAR, THE FOLLOWING ARE VALID FOR $0 < \text{ABS}(X) \leq 1$. THEY WOULD BE VALID FOR $X < \text{SQRT}(342)$ IF I SUMMED THE SERIES DIFFERENTLY, BUT I NEED Y ALWAYS POSITIVE FOR THE FOLLOWING TO WORK *)

```

FUNCTION DCOS(X:REAL):REAL; (* LOWER BOUND TO COS(X) *)
VAR X2,Y:REAL;
BEGIN
  X2:=INC(X*X);
  Y:=DEC(1-INC(X2/306)); (* INTEGERS ARE CONVERTED EXACTLY *)
  Y:=INC(1-Y*X2/240);
  Y:=DEC(1-INC(Y*INC(X2/182)));
  Y:=INC(1-Y*X2/132);
  Y:=DEC(1-INC(Y*INC(X2/90)));
  Y:=INC(1-Y*X2/56);
  Y:=DEC(1-INC(Y*INC(X2/30)));
  Y:=INC(1-Y*X2/12);
  DCOS:=DEC(1-INC(Y*INC(X2/2))) END;

FUNCTION UCOS(X:REAL):REAL; (* UPPER BOUND TO COS(X) *)
VAR X2,Y:REAL;

```

Trigonometric Routines (continued)

BEGIN

```

X2:=X*X;
Y:=DEC(1-INC(X2/240));
Y:=INC(1-Y*X2/182);
Y:=DEC(1-INC(Y*INC(X2/132)));
Y:=INC(1-Y*X2/90);
Y:=DEC(1-INC(Y*INC(X2/56)));
Y:=INC(1-Y*X2/30);
Y:=DEC(1-INC(Y*INC(X2/12)));
UCOS:=INC(1-Y*X2/2) END;

```

```

FUNCTION DSINO(X:REAL):REAL; (* LOWER BOUND TO SIN(X)/X *)
VAR X2,Y:REAL;

```

BEGIN

```

X2:=X*X;
Y:=DEC(1-INC(X2/210));
Y:=INC(1-Y*X2/156);
Y:=DEC(1-INC(Y*INC(X2/110)));
Y:=INC(1-Y*X2/72);
Y:=DEC(1-INC(Y*INC(X2/42)));
Y:=INC(1-Y*X2/20);
DSINO:=DEC(1-INC(Y*INC(X2/6))) END;

```

```

FUNCTION USINO(X:REAL):REAL; (* UPPER BOUND TO SIN(X)/X *)
VAR X2,Y:REAL;

```

BEGIN

```

X2:=INC(X*X);
Y:=DEC(1-INC(X2/272));
Y:=INC(1-Y*X2/210);
Y:=DEC(1-INC(Y*INC(X2/156)));
Y:=INC(1-Y*X2/110);
Y:=DEC(1-INC(Y*INC(X2/72)));
Y:=INC(1-Y*X2/42);
Y:=DEC(1-INC(Y*INC(X2/20)));
USINO:=INC(1-Y*X2/6) END;

```

```

FUNCTION DECSIN(X:REAL):REAL; (* SMALLER BOUND ON SIN(X) *)
BEGIN DECSIN:=X*DSINO(X) END;

```

```

FUNCTION INCSIN(X:REAL):REAL; (* LARGER BOUND ON SIN(X) *)
BEGIN INCSIN:=INC(X*USINO(X)) END;

```

PROCEDURE CALCITWOPI;

```

(* SETS ITWOPI TO AN INTERVAL CONTAINING 2*PI, BY FINDING AN INTERVAL
CONTAINING THE FIRST ROOT OF .SIN(X)=0.5 *)

```

VAR X,X1,X2:REAL;

BEGIN

```

X1:=0.5235;X2:=0.5236;

```

```

IF (INCSIN(X1)>0.5)OR(INCSIN(X2)<0.5) THEN (*DYADIC RATIONALS GET
CONVERTED EXACTLY, PROVIDED THEY ARE REPRESENTABLE *)

```

```

HALT(' BAD INITIAL INTERVAL IN CALCITWOPI.L');

```

REPEAT

```

X:=X1+0.5*(X2-X1); (*IT DOES NOT MATTER THAT THIS MAY NOT BE EXACT,
AS LONG AS THE LOOP TERMINATES *)

```

```

IF INCSIN(X)<=0.5 THEN X1:=X

```

```

ELSE X2:=X

```

```

UNTIL X2=INC(X1); (* SO SIN(X1)<=0.5, BUT IS VERY CLOSE *)

```

```

ITWOPI.L:=12*X1;

```

```

ITWOPI.R:=INC(1/ITWOPI.L);

```

```

X1:=0.5235;X2:=0.5237;

```

```

IF (DECSIN(X1)>0.5)OR(DECSIN(X2)<0.5) THEN

```

```

HALT(' BAD INITIAL INTERVAL IN CALCITWOPI.R');

```

REPEAT

```

X:=X1+0.5*(X2-X1);

```

```

IF DECSIN(X)<0.5 THEN X1:=X

```

```

ELSE X2:=X

```

```

UNTIL X2=INC(X1); (* SO SIN(X2)>=0.5, BUT CLOSE *)

```

```

ITWOPI.R:=INC(12*X2);

```

```

ITWOPI.L:=1/ITWOPI.R;

```

```

WRITE(' TWOPI LIES BETWEEN HEX:');

```

```

WRITEHEX(ITWOPI.L);

```

```

WRITE(' AND');WRITEHEX(ITWOPI.R);

```

```

Writeln

```

END;

Trigonometric Routines (continued)

```

FUNCTION DCOSTP(Z:REAL):REAL; (* LOWER BOUND TO COS(2*PI*Z) *)
BEGIN
  Z:=ABS(Z-ROUND(Z)); (*EXACT*)
  IF Z<0.25 THEN
    IF Z<0.125 THEN
      IF Z=0 THEN DCOSTP:=1
      ELSE (* 0<Z<.125 *) DCOSTP:=DCOS(INC(ITWOPI.R*Z))
    ELSE (* .125<=Z<.25 *) DCOSTP:=DECSIN(ITWOPI.L*(O.25-Z))
    (* (0.25-Z) IS EXACT HERE, AND THE SAME GOES FOR ALL SIMILAR CASES WHICH
    WILL OCCUR *)
  ELSE IF Z<0.375 THEN
    IF Z=0.25 THEN DCOSTP:=0
    ELSE (* .25<Z<.375 *) DCOSTP:=-INCSIN(INC(ITWOPI.R*(Z-O.25)))
    (* SIGN CHANGE IS ALWAYS EXACT *)
  ELSE IF Z=0.5 THEN DCOSTP:=-1
  ELSE (* .375<=Z<.5 *) DCOSTP:=-UCOS(ITWOPI.L*(O.5-Z))
END;

FUNCTION UCOSTP(Z:REAL):REAL; (* UPPER BOUND ON COS(2*PI*Z) *)
BEGIN
  Z:=ABS(Z-ROUND(Z));
  IF Z<0.25 THEN
    IF Z<0.125 THEN
      IF Z=0 THEN UCOSTP:=1
      ELSE (* 0<Z<.125 *) UCOSTP:=UCOS(ITWOPI.L*Z)
    ELSE (* .125<=Z<.25 *) UCOSTP:=INCSIN(INC(ITWOPI.R*(O.25-Z)))
  ELSE IF Z<0.375 THEN
    IF Z=0.25 THEN UCOSTP:=0
    ELSE (* .25<Z<.375 *) UCOSTP:=-DECSIN(ITWOPI.L*(Z-O.25))
  ELSE IF Z=0.5 THEN UCOSTP:=-1
  ELSE (* .375<=Z<.5 *) UCOSTP:=-DCOS(INC(ITWOPI.R*(O.5-Z))
END;

(* THE FOLLOWING SHOULD BE FUNCTIONS *)

PROCEDURE RICOSTP(I:INTERVAL;VAR J:INTERVAL);
(*RETURNS AN INTERVAL CONTAINING COS(2*PI*I) *)
VAR NL,NL1,NR,NR1,DN,NN:INTEGER;
    ZL,ZR:REAL;
BEGIN
  NL1:=ROUNDDOWN(I.L);
  ZL:=I.L-NL1;
  IF ZL<0.5 THEN NL:=2*NL1 (*RIGOROUS EVEN THOUGH ZL MAY NOT BE EXACT*)
  ELSE NL:=2*NL1+1;
  NR1:=ROUNDUP(I.R);
  ZR:=NR1-I.R;
  IF ZR<0.5 THEN NR:=2*NR1
  ELSE NR:=2*NR1-1;
  DN:=NR-NL;
  IF DN<=1 THEN
    IF ODD(NL) THEN BEGIN
      J.L:=DCOSTP(I.L);J.R:=UCOSTP(I.R) END
    ELSE BEGIN
      J.L:=DCOSTP(I.R);J.R:=UCOSTP(I.L) END
  ELSE IF DN=2 THEN
    IF ODD(NL) THEN BEGIN
      J.R:=1;
      NN:=NL1+1;
      J.L:=DCOSTP(MAX(NN-I.L,I.R-NN)) END (*ARGUMENTS TO DCOSTP ARE EXACT *)
    ELSE BEGIN
      J.L:=-1;
      J.R:=UCOSTP(MIN(ZL,ZR)) END (* ZL,ZR ARE EXACT IN THIS CASE *)
  ELSE BEGIN
    J.L:=-1;J.R:=1 END
END;

PROCEDURE RSINTP(Z:REAL;VAR I:INTERVAL);
(* RETURNS AN INTERVAL CONTAINING SIN(2*PI*Z) *)
VAR Z1:REAL;

```

Trigonometric Routines (continued)

```

BEGIN
  Z:=Z-ROUND(Z);          (*EXACT*)
  IF Z>=0 THEN
    IF Z>0.375 THEN
      IF Z=0.5 THEN BEGIN
        I.L:=0;I.R:=0 END
      ELSE (* .375<Z<.5 *) BEGIN
        Z1:=0.5-Z;      (*EXACT*)
        I.L:=DECSIN(ITWOPI.L*Z1);I.R:=INCSIN(INC(ITWOPI.R*Z1)) END
      ELSE IF Z>0.125 THEN
        IF Z=0.25 THEN BEGIN
          I.L:=1;I.R:=1 END
        ELSE (* .125<Z<=.375 *) BEGIN
          Z1:=0.25-Z;
          I.L:=DCOS(INC(ITWOPI.R*Z1));I.R:=UCOS(ITWOPI.L*Z1) END
        ELSE IF Z=0 THEN BEGIN
          I.L:=0;I.R:=0 END
        ELSE (* 0<Z<.125 *) BEGIN
          I.L:=DECSIN(ITWOPI.L*Z);I.R:=INCSIN(INC(ITWOPI.R*Z)) END
      ELSE IF Z<-0.375 THEN
        IF Z=-0.5 THEN BEGIN
          I.L:=0;I.R:=0 END
        ELSE (* -.5<Z<-.375 *) BEGIN
          Z1:=0.5+Z;
          I.L:=-INCSIN(INC(ITWOPI.R*Z1));I.R:=-DECSIN(ITWOPI.L*Z1) END
      ELSE IF Z<-0.125 THEN
        IF Z=-0.25 THEN BEGIN
          I.L:=-1;I.R:=-1 END
        ELSE (* -.375<Z<-.125 *) BEGIN
          Z1:=0.25+Z;
          I.L:=-UCOS(ITWOPI.L*Z1);I.R:=-DCOS(INC(ITWOPI.R*Z1)) END
      ELSE (* -.125<Z<0 *) BEGIN
        Z1:=-Z;
        I.L:=-INCSIN(INC(ITWOPI.R*Z1));I.R:=-DECSIN(ITWOPI.L*Z1) END
  END;

```

Rigorous Test Procedure

(* THIS PROCEDURE TESTS WHETHER THERE CAN BE ANY INVARIANT CIRCLES UNDER THE STANDARD MAP, PASSING THROUGH INTERVAL IZO ON THE LINE X=0, FOR THE PARAMETER INTERVAL IK, WITH RIGOROUS CONTROL OF ROUND OFF ERRORS. IF IT FINDS THAT THERE CAN BE NO CIRCLES, EXECUTION RETURNS TO THE PLACE FROM WHICH THE PROCEDURE WAS MOST RECENTLY CALLED. ELSE, IF IZO OR IK IS TOO SMALL IT HALTS EXECUTION WITH AN ERROR MESSAGE, ELSE IT SUBDIVIDES IZO OR IK AND CALLS ITSELF ON THE TWO PIECES *)

(* NUMBERS IN BRACKETS REFER TO EQUATION NUMBERS IN THE TEXT *)

```

PROCEDURE RTEST(IZO,IK:INTERVAL);
BEGIN
  C:=0;          (*ASSUMES 0<IK.L<2 *)
                (*SET ITERATION COUNTER *)

  XC:=0;        (*INITIALISE PARALLELOGRAM: *)          (*4.1*)
  IZ:=IZO;      (* (ZC,XC) + Q.UNIT CUBE *)
  ZCO:=ICENTRE(IZ); (*SPECIFYING BOUNDS ON ORBIT *)
  ZC:=ZCO;      (*OF INITIAL INTERVAL (IZO,0) *)
  ZR:=RMAXERROR(IZ,ZC);
  ZRO:=ZR;
  Q10:=0;Q11:=ZR;Q12:=0;          (*5.11*)
  Q20:=0;Q21:=0;Q22:=0;

  KM:=ICENTRE(IK);          (*INITIALISE K *)
  RIPRODP(IK,ITWOPI.I,IKB);
  KB:=ICENTRE(IKB);
  KBR:=RMAXERROR(IKB,KB);

  K1:=INC(1+INC(0.5*IK.R));      (*INITIALISE SLOPES: *)
  DO:=DEC(K1-USORT(INC(INC(K1*K1)-1))); (*GLOBAL SLOPE *)          (*3.23*)
  D:=INC(1-0.5*IK.L);          (*INITIAL SLOPE ON X=0 *)          (*5.22*)

```

(* WE COULD TEST HERE IF D<DO, IN WHICH CASE WE ARE DONE. THIS DISPOSES OF ALL INTERVALS WITH IK.L>=2, AND INDEED, ALL SHORT ENOUGH INTERVALS WITH IK.L>4/3. IN ORDER TO PROCEED, I NEED D>0, HENCE THE RESTRICTION IK.L<2 *)

Rigorous Test Procedure (continued)

```

RICOSTP(IZ,IC); (* EVALUATE BOUNDS ON G'(Z) *)
RIPRODP(IK,IC,ITEMP);
RSIDIFF(2.0,ITEMP,IGZ);

D:=UP(IGZ.R-1/D); (*UPPER BOUND ON SLOPE AT IMAGE *)(*5.19*)

OPTION:=1; (*START WITH FIRST METHOD OF FATTENING*)

WHILE (D>=DO)AND(ZR<=ZRMAXO) DO BEGIN
(*SLOPE NOT YET SMALL ENOUGH, AND PGM NOT YET TOO LARGE, SO ITERATE *)
C:=SUCC(C); (*INCREMENT COUNTER *)

RSINTP(ZC,IS); (*NEW CENTRE *)
RIFATTEN(2*ZC,ITEMP);
RISDIFF(ITEMP,XC,ITEMP2);
RISPROD(IS,KB,ITEMP);
RIDIFF(ITEMP2,ITEMP,IZC); (* T: Z' := 2Z-KB*SIN(2*PI*Z)-X *) (*5.4*)
XC:=ZC; (* X' := Z *)
ZC:=ICENTRE(IZC);
DZC:=RMAXERROR(IZC,ZC);
ZC:=ZC-ROUND(ZC); (*EXACT*)

RISPROD(IGZ,Q10,IQ10); (* IQ:=IDT.Q *)
RISPROD(IGZ,Q11,IQ11);
RISPROD(IGZ,Q12,IQ12);
RISDIFF(IQ10,Q20,IQ10);
RISDIFF(IQ11,Q21,IQ11);
RISDIFF(IZ,0.25,ITEMP);
RICOSTP(ITEMP,IS); (* I SHOULD HAVE WRITTEN RSINTP *)
RISPROD(IS,KBR,ITEMP);
RIDIFF(IQ10,ITEMP,IQ10);
(* IQ2J:=Q1J, J:=0,1,2 *)

QP10:=ICENTRE(IQ10); (*BEGIN TO SET QP:=ICENTRE(IQ)*)(*5.15*)
QP11:=ICENTRE(IQ11); (*5.13*)
(* QP20:=Q10 *)

IF OPTION=1 THEN (*FIRST METHOD OF FATTENING*)
  IF ABS(Q12)<QF*Q11 THEN BEGIN
    Q21:=INC(Q11+ABS(Q12)); (* QP21 *) (*5.14*)
    RIFATTEN(QP11/Q21,ITEMP2); (* IM:=QP**(-1).IQ *)
    M20:=RMAXERROR(IQ10,QP10);
    RISPROD(ITEMP2,Q11,ITEMP);
    RIDIFF(IQ11,ITEMP,IM21);
    RISPROD(ITEMP2,Q12,ITEMP);
    RIDIFF(IQ12,ITEMP,IM22);
    Q11:=QP11; (* NEW Q *)
    Q12:=UP(IABS(IM22)+UP(IABS(IM21)+UP(M20+DZC))) END (*5.17*)
  ELSE OPTION:=2;

IF OPTION=2 THEN BEGIN (* SECOND METHOD OF FATTENING: QP IS A OF  $\mathbb{C}^4$  *)
  RISDIFF(IQ12,Q22,IQ12); (* QP:=ICENTRE(IQ) *) (*4.10*)
  QP12:=ICENTRE(IQ12);
  QP21:=Q11;QP22:=Q12;
  TEMP:=QP11*QP22;TEMP2:=QP12*QP21; (*SEE IF QP NEEDS FATTENING*)
  DET:=TEMP-TEMP2;
  COL1:=ABS(QP11)+ABS(QP21);COL2:=ABS(QP12)+ABS(QP22);
  B:=BF*COL1;
  E:=B*COL2;IF E=0 THEN E:=1;
  ADET:=ABS(DET);
  IF ADET<=E*COL1 THEN BEGIN (* FATTEN QP *)
    IF DET*QP21<0 THEN E:=-E;
    QP12:=QP12-E;
    TEMP2:=QP12*QP21;
    ADET:=ABS(TEMP-TEMP2) END;

(*GET NONRIGOROUS EXPANSION FACTORS W1,W2.
  COULD GET RIGOROUS W1,W2, BUT CANNOT MULTIPLY COLUMNS EXACTLY BY THEM,
  SO EXPAND A LITTLE MORE THAN NECESSARY (WF), AND THEN CHECK IT IS OK*)

```

Rigorous Test Procedure (continued)

```

HWQ10:=IHWIDTH(IQ10);
M10:=ABS(HWQ10*QP22); (* IM:=QP**-1 . IQ *)
ISPROD(IQ11,QP22,ITEMP);ISDIFF(ITEMP,TEMP2,IM11);
M12:=ABS(RMAXERROR(IQ12,QP12)*QP22);
M20:=ABS(HWQ10*QP21);
M21:=ABS(IHWIDTH(IQ11)*QP21);
ISPROD(IQ12,QP21,ITEMP);ISDIFF(ITEMP,TEMP,IM22);
W1:=WF*(ABS(QP22)*DZC+M10+IABS(IM11)+M12)/ADET; (*4.4*)
W2:=WF*(ABS(QP21)*DZC+M20+M21+IABS(IM22))/ADET;
OK:=0;

REPEAT
  QP21:=QP21*W1;QP22:=QP22*W2; (* EXPAND PGM *) (*4.3*)
  QP11:=QP11*W1;QP12:=QP12*W2;
  RIFATTEN(QP12*QP21,ITEMP); (* CHECK RIGOROUSLY THAT IT IS OK *)
  RIFATTEN(QP11*QP22,ITEMP2);
  RIDIFF(ITEMP2,ITEMP,IDET);
  ADET:=RIMIN(IDET);
  HWQ10:=RMAXERROR(IQ10,QP10);
  M10:=UP(ABS(HWQ10*QP22));
  M20:=UP(ABS(HWQ10*QP21));
  RISPROD(IQ11,QP22,IM11);RIFATTEN(QP12*Q11,ITEMP);
  RIDIFF(IM11,ITEMP,IM11);
  RISPROD(IQ12,QP22,IM12);RIFATTEN(QP12*Q12,ITEMP);
  RIDIFF(IM12,ITEMP,IM12);
  RISPROD(IQ11,QP21,IM21);RIFATTEN(QP11*Q11,ITEMP);
  RIDIFF(IM21,ITEMP,IM21);
  RISPROD(IQ12,QP21,IM22);RIFATTEN(QP11*Q12,ITEMP);
  RIDIFF(IM22,ITEMP,IM22);
  W1:=UP(IABS(IM11)+UP(IABS(IM12)+UP(M10+UP(ABS(QP22*DZC))))); (*4.4*)
  W2:=UP(IABS(IM22)+UP(IABS(IM21)+UP(M20+UP(ABS(QP21*DZC)))));
  IF (W1<=ADET)AND(W2<=ADET) THEN OK:=-1 (*IT IS OK*)
  ELSE BEGIN
    W1:=WF*W1/ADET; (*TRY EXPANDING A LITTLE*)
    W2:=WF*W2/ADET;
    OK:=SUCC(OK);
    IF OK>9 THEN HALT('OK=10');
    WFCOUNTER:=SUCC(WFCOUNTER) END
  UNTIL OK=-1;
  Q21:=QP21;Q22:=QP22; (* NEW Q *)
  Q11:=QP11;Q12:=QP12 END;

Q20:=Q10; (*FINISH UPDATING Q *) (*5.16*)
Q10:=QP10;

ZR:=INC(ABS(Q11)+UP(ABS(Q10)+ABS(Q12))); (*NEW IZ *) (*4.11*)
IZ.L:=DOWN(ZC-ZR);IZ.R:=UP(ZC+ZR);

RICOSTP(IZ,IC); (*BOUNDS ON G'(Z) *)
RIPRODP(IK,IC,ITEMP);
RSIDIFF(2.0,ITEMP,IGZ);

D:=UP(IGZ.R-1/D) (*UPPER BOUND ON SLOPE AT IMAGE *)(*5.19*)

END(*WHILE LOOP*);

RTOTAL:=RTOTAL+C; (*UPDATE RECORD OF WORK DONE*)
RNRECT:=SUCC(RNRECT);
IF C>RMAXITER THEN BEGIN
  RMAXITER:=C;
  IWORSTZ:=IZO;
  IWORSTK:=IK END;
IF D<DO THEN BEGIN (*SUCCESSFUL *)
  IF ZRO<MINZRO THEN BEGIN
    MINZRO:=ZRO;
    IMINZZ:=IZO;IMINKZ:=IK END;
  IF KBR<MINKBR THEN BEGIN
    MINKBR:=KBR;
    IMINZK:=IZO;IMINKK:=IK END END
ELSE BEGIN
  RFAILURES:=RFAILURES+C;
  RNFAILRECT:=SUCC(RNFAILRECT);

```


Rigorous Test Procedure (continued)

```

IF KBR>(C+1)*CFACTOR*ZRO THEN          (*TRY TO SUBDIVIDE IN K *)
IF (KM=IK.L)OR(KM=IK.R) THEN          (*CANNOT BECAUSE IK IS TOO SHORT*)
  HALT(' CANNOT SUBDIVIDE ANY FURTHER IN K IN RTEST')
ELSE BEGIN                              (* OK *)
  IK1.L:=KM;IK1.R:=IK.R;
  IK.R:=KM;
  RTEST(IZO,IK1);
  RTEST(IZO,IK) END
ELSE                                     (*TRY TO SUBDIVIDE IN Z *)
IF (ZCO=IZO.L)OR(ZCO=IZO.R) THEN      (*CANNOT BECAUSE IZO IS TOO SHORT*)
  HALT(' CANNOT SUBDIVIDE ANY FURTHER IN Z IN RTEST')
ELSE BEGIN                              (* OK *)
  IZ1.L:=IZO.L;IZ1.R:=ZCO;
  IZO.L:=ZCO;
  RTEST(IZ1,IK);
  RTEST(IZO,IK) END END

```

END;

Nonrigorous Test Procedure

```

(* TO SAVE TIME, THIS NONRIGOROUS VERSION OF RTEST IS USED FIRST TO FIND
RECTANGLES ON WHICH RTEST IS LIKELY TO SUCCEED.
IF TEST IS SUCCESSFUL, IT PASSES THE RECTANGLE TO RTEST
ELSE IT SUBDIVIDES AND TRIES AGAIN WITH TEST,
UNLESS IK OR IZO IS TOO SMALL, IN WHICH CASE IT HALTS *)

```

```
PROCEDURE TEST(IZO,IK:INTERVAL);
```

```
BEGIN
```

```
  C:=0;
```

```
  XC:=0;
```

```
  IZ:=IZO;
```

```
  ZCO:=ICENTRE(IZ);
```

```
  ZC:=ZCO;
```

```
  ZR:=IHWIDTH(IZ);
```

```
  ZRO:=ZR;
```

```
  Q10:=0;Q11:=ZR;Q12:=0;
```

```
  Q20:=0;Q21:=0;Q22:=0;
```

```
  KM:=ICENTRE(IK);
```

```
  ISPROD(IK,TWOPI,IKB);
```

```
  KB:=ICENTRE(IKB);
```

```
  KBR:=IHWIDTH(IKB);
```

```
  K1:=1+0.5*IK.R;
```

```
  DO:=K1-SQRT(K1*K1-1);
```

```
  D:=1-0.5*IK.L;
```

```
  ICOSTP(IZ,IC);
```

```
  IPRODP(IK,IC,ITEMP);
```

```
  SIDIFF(2.0,ITEMP,IGZ);
```

```
  D:=IGZ.R-1/D;
```

```
  ZRMAX:=ZRMAXO;
```

```
  OPTION:=1;
```

```
  WHILE (D>=DO)AND(ZR<=ZRMAX) DO BEGIN
```

```
    C:=SUCC(C);
```

```
    S:=SIN(TWOPI*ZC);
```

```
    ZCN:=2.0*ZC-KB*S-XC;
```

```
    XC:=ZC;
```

```
    ZC:=ZCN-ROUND(ZCN);
```

```
    ISPROD(IGZ,Q10,IQ10);
```

```
    ISPROD(IGZ,Q11,IQ11);
```

```
    ISPROD(IGZ,Q12,IQ12);
```

```
    SIDIFF(IQ10,Q20,IQ10);
```

```
    SIDIFF(IQ11,Q21,IQ11);
```

```
    SIDIFF(IZ,0.25,ITEMP);
```

```
    ICOSTP(ITEMP,IS);
```

```
    ISPROD(IS,KBR,ITEMP);
```

```
    IDIFF(IQ10,ITEMP,IQ10);
```

Nonrigorous Test Procedure (continued)

```
QP10:=ICENTRE(IQ10);
QP11:=ICENTRE(IQ11);
```

```
IF OPTION=1 THEN
  IF ABS(Q12)<QF*Q11 THEN BEGIN
    Q21:=Q11+ABS(Q12);
    TEMP:=QP11/Q21;
    M20:=IHWIDTH(IQ10);
    ISDIFF(IQ11,TEMP*Q11,IM21);
    ISDIFF(IQ12,TEMP*Q12,IM22);
    Q11:=QP11;
    Q12:=M20+IABS(IM21)+IABS(IM22) END
  ELSE OPTION:=2;
```

```
IF OPTION=2 THEN BEGIN
  ISDIFF(IQ12,Q22,IQ12);
  QP12:=ICENTRE(IQ12);
  TEMP:=QP11*Q12;TEMP2:=QP12*Q11;
  DET:=TEMP-TEMP2;
  COL1:=ABS(QP11)+ABS(Q11);COL2:=ABS(QP12)+ABS(Q12);
  B:=BF*COL1;
  E:=B*COL2;
  IF E=0 THEN E:=1;
  ADET:=ABS(DET);
  IF ADET<=E*COL1 THEN BEGIN
    IF DET*Q11<0 THEN E:=-E;
    QP12:=QP12-E;
    TEMP2:=QP12*Q11;
    ADET:=ABS(TEMP-TEMP2) END;
  HWQ10:=IHWIDTH(IQ10);
  M10:=ABS(HWQ10*Q12);
  ISPROD(IQ11,Q12,ITEMP);ISDIFF(ITEMP,TEMP2,IM11);
  M12:=ABS(RMAXERROR(IQ12,QP12)*Q12);
  M20:=ABS(HWQ10*Q11);
  M21:=ABS(IHWIDTH(IQ11)*Q11);
  ISPROD(IQ12,Q11,ITEMP);ISDIFF(ITEMP,TEMP,IM22);
  W1:=(M10+IABS(IM11)+M12)/ADET;
  W2:=(M20+M21+IABS(IM22))/ADET;
  Q21:=Q11*W1;Q22:=Q12*W2;
  Q11:=QP11*W1;Q12:=QP12*W2 END;
```

```
Q20:=Q10;
Q10:=QP10;
```

```
ZR:=ABS(Q10)+ABS(Q11)+ABS(Q12);
IZ.L:=ZC-ZR;IZ.R:=ZC+ZR;
```

```
ICOSTP(IZ,IC);
IPRODP(IK,IC,ITEMP);
SIDIFF(2.O,ITEMP,IGZ);
```

```
D:=IGZ.R-1/D;
```

```
ZRMAX:=ZRMAX*ZRF1
```

```
END;
```

```
TOTAL:=TOTAL+C;
NRECT:=SUCC(NRECT);
```

```
IF D<DO THEN BEGIN
  USEFULTOTAL:=USEFULTOTAL+C;
  NUSEFULRECT:=SUCC(NUSEFULRECT);
  RTEST(IZO,IK) END
```

```
ELSE IF KBR>(C+1)*CFACOR*ZRO THEN
  IF (KM=IK.L)OR(KM=IK.R) THEN
    HALT(' CANNOT SUBDIVIDE ANY FURTHER IN K IN TEST')
```

Nonrigorous Test Procedure (continued)

```
ELSE BEGIN
  IK1.L:=KM;IK1.R:=IK.R;
  IK.R:=KM;
  TEST(IZO,IK1);
  TEST(IZO,IK) END
```

```
ELSE
  IF (ZCO=IZO.L)OR(ZCO=IZO.R) THEN
    HALT(' CANNOT SUBDIVIDE ANY FURTHER IN Z IN TEST')
  ELSE BEGIN
    IZ1.L:=IZO.L;IZ1.R:=ZCO;
    IZO.L:=ZCO;
    TEST(IZ1,IK);
    TEST(IZO,IK) END
```

END;

Main Program

```
BEGIN
  INITROUND;CALCITWOPI;
  TWOPI:=8.0*ARCTAN(1.0);TWOPII:=1/TWOPI; (*NONRIGOROUS, FOR USE OF TEST *)
  WRITE(' COMPARE NONRIGOROUS VALUE:');WRITEHEX(TWOPI);Writeln;
  READLN(NMAX);
  FOR N:=1 TO NMAX DO BEGIN (*LOOP TO TEST DESIRED RECTANGLES *)
    READLN(CFACTOR,ZRMAXO,WFX,ZRF1,QF,BF); (*PARAMETERS GOVERNING
      TERMINATION AND SUBDIVISION CONDITIONS IN TEST AND RTEST *)
    WF:=1+WFX;
    Writeln;Writeln;Writeln(' INPUT:');Writeln;
    Writeln(' CFACTOR=',CFACTOR,' ZRMAXO=',ZRMAXO,' WFX=',WFX);
    Writeln(' ZRF1=',ZRF1,' QF=',QF,' BF=',BF);
    READLN(IZO.L,IZO.R); (*INTERVAL IN Z ON LINE X=0 *)
    READLN(IK.L,IK.R); (*INTERVAL IN PARAMETER K *)
    Writeln(' IZO.L=',IZO.L,' IZO.R=',IZO.R);
    Writeln(' IK.L =',IK.L,' IK.R =',IK.R);
    RMAXITER:=0;WFCOUNTER:=0; (*INITIALISE COUNTERS FOR WORK DONE*)
    RTOTAL:=0;RFAILURES:=0;TOTAL:=0;USEFULTOTAL:=0;
    RNRECT:=0;RNFAILRECT:=0;NRECT:=0;NUSEFULRECT:=0;
    MINZRO:=IZO.R-IZO.L;MINKBR:=IK.R-IK.L;
    IMINZZ:=IZO;IMINKZ:=IK;IMINZK:=IZO;IMINKK:=IK;IWORSTZ:=IZO;IWORSTK:=IK;
    CLOCKO:=CLOCK;

    TEST(IZO,IK); (*TEST FOR CIRCLES THROUGH RECTANGLE IZO,IK *)

    (*IF EXECUTION DID NOT HALT, THEN TEST WAS SUCCESSFUL *)
    CLOCKO:=CLOCK-CLOCKO;
    Writeln;Writeln(' RESULTS:');Writeln;
    Writeln(' THERE ARE NO CIRCLES INVARIANT UNDER THE STANDARD MAP',
      ' FOR K IN THE INTERVAL:');
    WRITE(' HEX:');WRITEHEX(IK.L);WRITEHEX(IK.R);
    Writeln(' I.E.:',IK.L,IK.R);
    Writeln(' PASSING THROUGH THE INTERVAL:');
    WRITE(' HEX:');WRITEHEX(IZO.L);WRITEHEX(IZO.R);
    Writeln(' I.E.:',IZO.L,IZO.R);Writeln(' ON THE LINE X=0');
    Writeln;Writeln(' DETAILS OF WORK DONE:');Writeln;
    Writeln(' NO. OF RECTANGLES EXAMINED BY TEST=',NRECT);
    Writeln(' NO. PASSED TO RTEST=',NUSEFULRECT);
    Writeln(' NO. OF ITERATIONS MADE BY TEST=',TOTAL);
    Writeln(' NO. OF USEFUL ITERATIONS=',USEFULTOTAL);
    Writeln(' NO. OF RECTANGLES EXAMINED BY RTEST=',RNRECT);
    Writeln(' NO. OF FAILURES=',RNFAILRECT);
    Writeln(' NO. OF ITERATIONS MADE BY RTEST=',RTOTAL);
    Writeln(' NO. OF FAILURES=',RFAILURES);
    Writeln(' LARGEST NO. OF ITERATIONS IN RTEST FOR ONE RECTANGLE=',RMAXITER);
    Writeln(' FOR IZO=',IWORSTZ.L,IWORSTZ.R,' IK=',IWORSTK.L,IWORSTK.R);
    Writeln(' SMALLEST ZRO=',MINZRO);
    Writeln(' FOR IZO=',IMINZZ.L,IMINZZ.R,' IK=',IMINKZ.L,IMINKZ.R);
    Writeln(' SMALLEST KBR=',MINKBR);
    Writeln(' FOR IZO=',IMINZK.L,IMINZK.R,' IK=',IMINKK.L,IMINKK.R);
    Writeln(' WFCOUNTER=',WFCOUNTER);
    Writeln(' CPU TIME TAKEN=',CLOCKO,' MILLISECONDS')
  END
END.
```

Output:

TWOPI LIES BETWEEN HEX: 416487ED5110B460 AND 416487ED5110B462
COMPARE NONRIGOROUS VALUE: 416487ED5110B460

INPUT:

CFACOR= 8.999999999999999E-02 ZRMAXO= 3.999999999999999E-01 WFX= 9.999999999999999E-13
ZRF1= 9.989999999999999E-01 QF= 7.999999999999999E-02 BF= 7.000000000000000E+01
IZO.L= 0.000000000000000E+00 IZO.R= 5.000000000000000E-01
IK.L = 1.062500000000000E+00 IK.R = 1.500000000000000E+00

RESULTS:

THERE ARE NO CIRCLES INVARIANT UNDER THE STANDARD MAP FOR K IN THE INTERVAL:
HEX: 4110000000000 411800000000000 I.E.: 1.062500000000000E+00 1.500000000000000E+00
PASSING THROUGH THE INTERVAL:
HEX: 000000000000000 408000000000000 I.E.: 0.000000000000000E+00 5.000000000000000E-01
ON THE LINE X=0

DETAILS OF WORK DONE:

NO. OF RECTANGLES EXAMINED BY TEST= 631
NO. PASSED TO RTEST= 316
NO. OF ITERATIONS MADE BY TEST= 4052
NO. OF USEFUL ITERATIONS= 1392
NO. OF RECTANGLES EXAMINED BY RTEST= 316
NO. OF FAILURES= 0
NO. OF ITERATIONS MADE BY RTEST= 1392
NO. OF FAILURES= 0
LARGEST NO. OF ITERATIONS IN RTEST FOR ONE RECTANGLE= 12
FOR IZO= 4.042968750000000E-01 4.052734375000000E-01 IK= 1.069335937500000E+00 1.076171875000000E+00
SMALLEST ZRO= 2.441406250000000E-04
FOR IZO= 4.003906250000000E-01 4.008789062500000E-01 IK= 1.062500000000000E+00 1.069335937500000E+00
SMALLEST KBR= 2.7199331094808491E-04
FOR IZO= 4.057617187500000E-01 4.062500000000000E-01 IK= 1.065917968750000E+00 1.069335937500000E+00
WFCOUNTER= 0
CPU TIME TAKEN= 3004 MILLISECONDS

Acknowledgements. It is a pleasure to thank M. R. Herman, J. N. Mather and R. A. P. C. Newman for useful conversations, and R. de la Llave for checking the program. Needless to say, any errors are our own responsibility.

This work was supported in part by the U.K. Science and Engineering Research Council. We would like to thank everyone at IHES for the lively atmosphere and their warm hospitality.

References

1. Birkhoff, G.D.: Surface transformations and their dynamical applications. *Acta Math.* **43**, 1 (1920); Sur quelques courbes fermées remarquables. *Bull. Soc. Math. Fr.* **60**, 1 (1932); reprinted in *Collected Mathematical Papers*, Vol. II, pp. 111, 418, respectively (New York: AMS 1950)
2. Mather, J.N.: Glancing billiards. *Erg. Theory Dyn. Sys.* **2**, 397 (1982)
3. Mather, J.N.: Non-existence of invariant circles. *Erg. Theory Dyn. Sys.* **4**, 301 (1984)
4. Aubry, S., Le Daeron, P.Y., André, G.: Classical ground-states of a 1-D model for incommensurate structures. Preprint (1982) CEN Saclay
5. Aubry, S.: The twist map, the extended Frenkel-Kontorova model and the devil's staircase, *Physica* **7D**, 240 (1983)
6. Newman, R.A.P.C., Percival, I.C.: Definite paths and upper bounds on regular regions of velocity phase space. *Physica* **6D**, 249 (1983)
7. Rüssmann, H.: Springer Lect. Notes Math. **1007**, 677 (1983)
8. Herman, M.R.: Private communication
9. Herman, M.R.: Sur les courbes invariantes par les difféomorphismes de l'anneau *Astérisque* **103-104** (1983)
10. Greene, J.M.: A method for determining a stochastic transition. *J. Math. Phys.* **20**, 1183 (1979)
11. Chirikov, B.V.: A universal instability of many-dimensional oscillator systems. *Phys. Reports* **52**, 263 (1979)
12. Aubry, S., Le Daeron, P.Y.: The discrete Frenkel-Kontorova model and its extensions. I Exact results for the ground-states. *Physica* **8D**, 381 (1983)
13. MacKay, R.S., Meiss, J.D.: Linear stability of periodic orbits in Lagrangian systems. *Phys. Lett.* **98A**, 92 (1983)
14. Pascal 8000 Reference Manual, Version 2.0, Centre Inter-Regional de Calcul Electronique, Orsay (August 1980)
15. IBM System/370 Principles of Operation, IBM, GA 22-7000-6 (March 1980)
16. Escande, D.F., Doveil, F.: Renormalisation method for computing the threshold of the large-scale stochastic instability in two degrees of freedom Hamiltonian systems. *J. Stat. Phys.* **26**, 257 (1981)
17. MacKay, R.S., Meiss, J.D., Percival, I.C.: Stochasticity and transport in Hamiltonian systems, *Phys. Rev. Lett.* **52**, 697 (1984); Transport in Hamiltonian systems. *Physica* **13D**, 55 (1984)
18. Mather, J.N.: A criterion for non-existence of invariant circles. Preprint, Princeton (1982), submitted to *Math. Pub. IHES*
19. MacKay, R.S.: Transition to chaos, lecture notes for CERN accelerator school, Sardinia 1985 (to appear in Springer lecture Notes in Physics)
20. MacKay, R.S.: Renormalisation in area preserving maps Ph.D. Thesis (1982) Princeton (Univ. Microfilms Int., Ann Arbor, MI)
21. Mather, J.N.: Private communication
22. Stark, J.: An exhaustive criterion for the non-existence of invariant circles in area preserving twist maps. Preprint, Warwick (1984), submitted to *Erg. Theory Dyn. Sys*

Communicated by O.E. Lanford

Received April 19, 1984; in revised form July 5, 1984