ON THE COMPLEXITY OF THE ANISOTROPIC KEPLER PROBLEM AND THE PLANAR ISOSCELES THREE BODY PROBLEM

Jeremy Schiff
Department of Mathematics and Computer Science
Bar Ilan University
Ramat Gan 52900, Israel
e-mail: schiff@bimacs.cs.biu.ac.il

and

Hava T. Siegelmann
Department of Information Systems Engineering
Faculty of Industrial Engineering
Technion (Israel Institute of Technology)
Haifa 32000, Israel
e-mail: iehava@ie.technion.ac.il

July 1995

ABSTRACT. We show how to associate a computation to certain 2 degree of freedom hamiltonian systems, and use this to discuss the level of complexity of certain problems in the dynamics of these systems. A type of Turing machine over the reals can be identified, embedded in these systems.

1.Introduction. In two articles [1], C.Moore has explored a class of simple dynamical systems, and, in particular, has drawn conclusions about the *complexity* of these dynamical systems by connecting them with Turing machines. Moore makes the conjecture that the level of complexity of his systems is typical of the level of complexity of finite dimensional dynamical systems like the three body problem. Such a conjecture is motivated by the "physical Church-Turing thesis" [2], which states that physical systems cannot perform computations that cannot be performed by Turing machines, i.e. the dynamics of realistic physical systems *cannot* be more complex than the dynamics of Turing machines.

The purpose of this paper is to provide icing on the cake of Moore's work. It turns out that certain 2 degree of freedom hamiltonian systems (we will study the anisotropic Kepler problem (AKP) and the planar isosceles 3 body problem (PI3BP), making heavy use of the results given by Devaney in [3]) have dynamical systems very reminiscent of those of Moore embedded in them, and this provides the key for discussing the complexity of these systems. Our work continues the

work of Fredkin and Tofolli [4], who were the first to show how to build a physical system which could in some sense perform any given computation that a Turing machine, or, equivalently, a digital computer, could perform. To study the physical Church-Turing thesis, we need to do the opposite of what Fredkin and Tofolli did - that is, we need to associate computations with given physical systems. This is exactly what we plan to do here, at least for the AKP and PI3BP.

The natural computation which one might say a dynamical system performs, is that it computes the evolution of any given initial point in phase space. For systems like the AKP and PI3BP, we can use digital computers to numerically simulate this. Denoting the orbit of the point $p(0)$ as $p(t)$, and assuming existence of a convergent numerical algorithm, a digital computer can be used to verify that $p(t)$ lies in some particular open set in phase space, but cannot, in general, verify that it lies in some particular closed set (the numerical algorithm computes only with finite accuracy in finite time). Since the dynamical system computes $p(t)$ *exactly*, this is convincing evidence *against* the physical Church-Turing thesis - apparently dynamical systems *can* do something more than digital computers! This is indeed so, but what we are witnessing is the simple fact that dynamical systems of the type we will study here use real numbers[1], which digital computers, with constraints of finite memory, cannot. The "Turing machines" referred to in the statement of the physical Church-Turing thesis should be taken to be some set of fundamental computers with real number handling capabilites; for clarity, from here on, we will refer to these computers as "Turing machines over the reals", as distinct from "classic Turing machines".

What are Turing machines over the reals? One proposal has been made in [5], and we will make another proposal in this paper - that of (S, g) pairs, introduced in section 2 - motivated by the desire to keep the physical Church-Turing thesis true. The only property we should require of Turing machines over the reals, is that they should be in some sense approximable, to any degree of accuracy, by classic Turing machines. The analog neural nets of [6] are another possible candidate.

To present our main results we need to first clarify an important property of Turing machines over the reals, which we will do by introducing a machine which we call a "reactive Turing machine" - we will relate these to (S, g) pairs later. Recall that a classic Turing machine takes as input a finite string of 0's and 1's. A reactive Turing machine functions in the same way as a classic Turing machine, but takes as input an inifnite string, which consists of a finite string of 0's and 1's, embedded inside an inifnite string of 0's[2] (so there exists an obvious way to get an input for a reactive Turing machine from one for a classic Turing machine - just surround it with 0's - but *not* vice-versa). Since the "interesting" part of the input to a reactive Turing machine is surrounded with 0's, as opposed to blanks in the case of the classic Turing machine, there is no finite-time algorithm to determine the length of the "interesting" part. Consider now the following 3 questions:

(1) Is the machine in the state S at time T?
(2) Does the machine ever reach the state S?

[1]Some purists may insist that the use of real numbers in a physical model is itself an approximation to reality. Even if this is so, it is worthwhile to understand the complexity of these - often remarkably accurate - approximations.

[2]The name "reactive" is chosen because of the similarity with reactive programs, which have an open input channel throughout their computations.

(3) Is there a time T such that the machine is in the state S for all times greater than T?

(Here for simplicity we discuss reactive Turing machines that do not have a specified "halting state".) There is a simple qualitative distinction between these questions. The answer to question (1), whether it is "yes" or "no", can be ascertained by running the machine for time T. If the answer to question (2) is "yes", this will be found by running the machine for some finite time, but if it is "no", this will not be determined by running the machine for any finite time (this question is the well-known "halting problem"). Finally, for question (3), neither a "yes" or "no" answer can be determined by running the machine for any finite time. These arguments suggest that for reactive Turing machines, questions (1),(2), and (3) are (generically) of different complexities, and this is indeed the case. For *classic* Turing machines, however, questions (2) and (3) are of the *same* complexity (because we know the length of the input, we can keep track of the length of the tape during the calculation, and, thus, once we know that the machine arrives in state S we only need to check it stays there for some finite time to know that it will stay there forever). Indeed for classic Turing machines, it is hard to phrase a question of greater complexity than the halting problem; but for Turing machines over the reals, there are interesting dynamical questions of a greater complexity than the halting problem.

In the title and the opening paragraph of this paper, we referred to the notion of "the complexity of a physical system". Since we now see that for Turing machines over the reals (and therefore for physical systems) there are questions of a range of complexities, we see that that "the complexity of a physical system" is not a well-defined notion; we have to discuss the complexities of different questions about a physical system individually [2].

We now state our results. In our study of the AKP we will focus on the question of whether a given orbit is a collision orbit, and in our study of the PI3BP we will ask if a given orbit is an escape orbit. We argue that these questions are computationally equivalent to questions (2) and (3) above, respectively, for reactive Turing machines. These results are easilly motivated - for the AKP, collision takes a finite time, so asking whether a particular orbit reaches collision is equivalent to asking whether some reactive Turing machine ever reaches a certain state; for the PI3BP, escape takes an infinite time, and only happens if for all times after a certain time the escaping particle is moving away from the other pair, which is equivalent to question (3) above. These complexity results may well have practical meaning: we expect that computer simulation will be in some *qualitative* sense less useful in identifying the set of escape orbits for PI3BP than it is in identifying the set of escape orbits for AKP (the precise sense of this remains to be seen). But the main observation we want to make here is, as seen from the result for PI3BP, that *not* all interesting physical problems are reducible to the halting problem for a Turing machine. The physical Church-Turing thesis is often misunderstood as meaning this.

Two further notes are necessary before we proceed with the body of this paper. First, we should mention that even if we establish that a particular physical question is a "type (2)" or "type (3)" question, to establish its complexity we really need to prove it is a *generic* question of its type. In the terms introduced in [7], we have to prove the "computational irreducibility" of the relevant physical system (in

computer science terminology, we have to prove that the given question is *complete* in its class). We will not do this for the systems we consider, but we just note, following [7], that computational irreducibility is the norm, not the exception, for physical systems. Second, we will see in section 4 that just as there are problems of different complexity for the same physical system, there are also different ways to associate a computation with the same physical system - this arises because for general physical systems *there is no canonical way to associate an input and an output*.

This paper proceeds as follows. In section 2 we will define (S, g) pairs, our notion of real number Turing machines. In section 3 we will see how there are (S, g) pairs associated with the reactive Turing machines defined above, and also we will display (S, g) pairs associated with the AKP and PI3BP. Conjecturing that these various (S, g) pairs are of the same complexity (more will be said about this conjecture in section 4), we will deduce the results stated about the complexity of the problems stated for AKP and PI3BP. Finally in section 4 we tackle the question of whether all (S, g) pairs are of the same complexity. This paper is intended to be self-contained, and does not require a previous reading of Moore's work [1], which is nevertheless recommended, and was the starting point for these investigations. The results of [3] have been used very freely, and the meticulous reader will certainly wish to consult this work.

2.(S, g) **pairs.** Let S be a smooth 2-manifold[3] provided with a *partition*, i.e. a countable set of smooth 2-dimensional submanifolds R_i, $i = 0, 1, \ldots, d$, $(d \leq \infty)$, such that $R_i \cap R_j = \emptyset$ for all distinct i, j, and $S = \cup R_i$. Let g be a map with possible singularities on S, i.e. a function defined on $S - (\cup C_i)$, where C_i, $i = 0, \ldots, N$ $(N \leq \infty)$, are smooth 1-dimensional submanifolds of S. There is a natural way to associate a computation with such a pair (S, g): given a point $x \in S$ we write down the sequence $\{i_n(x)\}$, $n = 0, 1, \ldots$, where $i_n(x) = a$ if $g^n(x) \in R_a$. This sequence may terminate (if $g^n(x) \in \cup C_i$ for some n), or be infinite (otherwise). We say the sequence $\{i_n(x)\}$ is the *output* of the pair (S, g) for *input* x. For the most part we are interested in the case where the input x is *recursively specifiable*, i.e. its coordinates (a pair of reals, and possibly also an integer to specify which chart we are using in some countable atlas for S) are recursive.

As a simple example of such a computation, let S be the square $[0, 1] \times [0, 1]$, with the partition $S = R_0 \cup R_1$, where $R_0 = \{x < \frac{1}{2}\} \subset S$, and $R_1 = \{x \geq \frac{1}{2}\} \subset S$. Let g be the baker tansformation

$$g(x, y) = \begin{cases} (2x, \frac{1}{2}y) & x < \frac{1}{2} \\ (2x - 1, \frac{1}{2}(y + 1)) & x > \frac{1}{2}, \end{cases} \tag{1}$$

(see fig.1) which is defined and smooth on $S - C$, where C is the line $x = \frac{1}{2}$ in S. The computation associated to this pair (S, g) is exactly the binary expansion of the x-coordinate of a point in S.

The connection of (S, g) pairs with physical systems is that (S, g) pairs can arise from flows on 3-manifolds, which arise, for instance, as the flow on a constant energy surface of a 2 degree of freedom hamiltonian system. Suppose X is a vector field on

[3]All manifolds may be with or without boundary, unless otherwise stated, and are assumed oriented.

a smooth 3-manifold M which admits a *forward singular cross section*. A forward sinuglar cross section is a smooth 2-dimensional submanifold S of M such that

(1) X is transverse to S;
(2) There exist smooth 1-dimensional submanifolds C_i, $i = 0, \ldots, N$ ($N \leq \infty$), in S, such that the forward orbit of the flow determined by X through any point of $\cup C_i$ does not return to S;
(3) The forward orbit through any point of $S - (\cup C_i)$ does return to S;
(4) Each orbit of X meets S at least once

(c.f. [3]). Taking g to be the Poincaré return map on S for the flow determined by X, and supplying S with a partition, we clearly obtain an (S, g) pair. Note that if M has some physical interpretation, then we should have a notion of how to measure the coordinates of a point $x \in M$, and one (rather optimistic) way to do this is to supply M with a partition and to measure by specifying which cell of M the point x falls in. So M may come supplied with a partition, which would automatically give rise to one on S.

In [3], Devaney points out that the flows on constant energy surfaces of 2 degree of freedom hamiltonian systems frequently admit (at least after the removal of certain invariant submanifolds) forward singular cross sections.

We propose that we should view the set of (S, g) pairs (with a condition on the well-behavedness of g that we will give in section 4) as a candidate for the set of Turing machines over the reals. Comparison with [5] is left to a later date.

3.Examples of (S, g) pairs.

1. Let S be the half-open square $[0, 1) \times [0, 1)$, and let D be a positive integer. For $i \in \{0, \ldots, (D-1)\}$ and $\alpha \in \{0, 1\}$ let

$$R_{i\alpha} = \left\{ (x, y) \in S \;\middle|\; x \in \left[\frac{\alpha}{2}, \frac{\alpha+1}{2} \right), y \in \left[\frac{i}{D}, \frac{i+1}{D} \right) \right\} \tag{2}$$

(see fig.2).The $R_{i\alpha}$ clearly provide a partition of S. Define, for $j \in \{0, \ldots, (D-1)\}$, the following four maps from $R_{i\alpha}$ to S:

$$\begin{aligned}
g_{j00}(x, y) &= \left(2x - \alpha, \frac{1}{2} y + \frac{j - \frac{1}{2} i}{D} \right) \\
g_{j10}(x, y) &= \left(2x - \alpha, \frac{1}{2} y + \frac{j - \frac{1}{2}(i-1)}{D} \right) \\
g_{j01}(x, y) &= \begin{cases} \left(\frac{1}{2} \left(x - \frac{1}{2}\alpha \right), 2y + \frac{j-2i}{D} \right) & y < \frac{i+\frac{1}{2}}{D} \\ \left(\frac{1}{2} \left(1 + x - \frac{1}{2}\alpha \right), 2y + \frac{j-2i-1}{D} \right) & y \geq \frac{i+\frac{1}{2}}{D} \end{cases} \\
g_{j11}(x, y) &= \begin{cases} \left(\frac{1}{2} \left(\frac{1}{2} + x - \frac{1}{2}\alpha \right), 2y + \frac{j-2i}{D} \right) & y < \frac{i+\frac{1}{2}}{D} \\ \left(\frac{1}{2} \left(\frac{3}{2} + x - \frac{1}{2}\alpha \right), 2y + \frac{j-2i-1}{D} \right) & y \geq \frac{i+\frac{1}{2}}{D} \end{cases}
\end{aligned} \tag{3}$$

(see fig.3). Each of these maps is area and orientation preserving, with $g_{j\alpha 0}$ stretching in the x-direction and contracting in the y-direction, and $g_{j\alpha 1}$ stretching in the y-direction and contracting in the x-direction. Note $g_{j\beta\gamma}(R_{i\alpha}) \subset R_{j0} \cup R_{j1}$. If we choose a function g by specifying for each of the $R_{i\alpha}$ one map $g_{j\beta\gamma}$, we get an (S, g) pair; the map g in this case is defined on *all* of S, i.e. it has no singularities (but it has lines of discontinuity).

For any reactive Turing machine with D internal states there is an associated (S, g) pair of this sort. A Turing machine makes three "decisions" on the basis of its current internal state and the binary digit it is reading; it decides how to update its internal state, how to update the binary digit it is reading and whether to move left or right on the input/output tape. Equivalently [1], it is specified by three functions $F_1(i,\alpha)$, $F_2(i,\alpha)$, $F_3(i,\alpha)$ ($i \in \{0,\ldots,(D-1)\}$, $\alpha \in \{0,1\}$), with F_1 valued in $\{0,\ldots,(D-1)\}$, and F_2, F_3 valued in $\{0,1\}$. We just choose the function g in our (S, g) pair to be given by the action of $g_{F_1(i,\alpha)F_2(i,\alpha)F_3(i,\alpha)}$ on $R_{i\alpha}$. It is straightforward to show that the computation performed by the (S, g) pair defined this way gives as output, for some given input point $(x, y) \in S$ (x, y having finite binary expansions), a *record* of the reactive Turing machine's computation starting with a tape determined by the binary expansions of x and the fractional part[4] of yD. By a record of the computation, we mean a listing of the internal state of the machine and the binary digit the machine is reading at each step of the computation. The details are left for the reader.

We note, as in [1], that (S, g) pairs of this form can be associated with a forward singular cross section of some noncontinuous vector field on some subset of three dimensional space, which describes the motion of a free particle in a region bounded by mirrors.

The halting question for the reactive Turing machine is translated into the question whether, for a specific input (x, y), the iterates $g^n(x, y)$ ever enter the set $R_{i0} \cup R_{i1}$ for some particular i (without loss of generality take $i \neq 0$). Once we have identified a "halting region", it is natural to modify the definition of the (S, g) pair by contracting the halting region to the line $y = i/D$, on which g is now undefined, and by redefining g on regions that map to the halting region, to map them to the line $y = i/D$. We then have an (S, g) pair with singularities, and thus the possibility of finite output, corresponding to halting of the associated reactive Turing machine.

The reader will have observed that while we have associated the (S, g) pairs of this section with reactive Turing machines, whose inputs are finite strings of 0's and 1's embedded in an infinite string of 0's, they are more naturally related to Turing machines which take infinite strings of 0's and 1's as input (the coordinates of a point in S need not have finite binary expansions). Our use of reactive Turing machines is motivated by the fact that we really want to discuss the complexity of (S, g) pairs for recursively specifiable input; a reactive Turing machine is associated with an (S, g) pair of this section with a very restricted - and most definitely recursively specifiable - input.

2. Consider the flow of the hamiltonian system (the anisotropic Kepler problem) given by

$$H = \frac{\mu p_1^2}{2} + \frac{p_2^2}{2} - \frac{1}{(q_1^2 + q_2^2)^{\frac{1}{2}}} \tag{4}$$

($\mu > 1$) on the smooth manifold M defined as the surface of constant negative energy $H = -e$, $e > 0$, with the invariant submanifolds $\{q_1 = p_1 = 0\}$ and $\{q_2 = p_2 = 0\}$ removed. It was shown in [3] that if $S = S^+ \cup S^-$, where $S^+ = \{p_2 = 0, q_2 > 0\}$ and $S^- = \{p_2 = 0, q_2 < 0\}$, then S is a forward singular cross

[4]The initial state of the machine should be taken as the integer part of yD.

section for this flow. S^+ and S^- are both planes, which can be parametrized by p_1, with $-\infty < p_1 < \infty$, and an angle θ with $0 < \theta < \pi$. The Poincaré map is defined everywhere on S except for on two spirals C_0^+, C_π^+ in S^+ and on two spirals C_0^-, C_π^- in S^-. S can be given a partition as follows: the orbit through any point of S must have crossed the surface $\{q_2 = 0\}$ some time prior to reaching S, and it will also return to $\{q_2 = 0\}$. We associate with each point x of S a positive integer $\#(x)$ which is the number of times the orbit through S crosses the surface $\{q_1 = 0\}$ between its last crossing of $\{q_2 = 0\}$ before reaching x and its next crossing of $\{q_2 = 0\}$ after reaching x. We give S a partition by saying $x \in R_a$ if $\#(x) = a$. The proof that $0 < \#(x) < \infty$ and of the other facts cited here can be found in [3].

The (S, g) pair associated to this forward singular cross section computes, for any initial point on S, a sequence - possibly finite, possibly infinite - of positive integers. The set of inputs to the (S, g) pair for which the output is finite are the points of S on collision orbits; thus by comparison with example 1, assumed valid, we have the first main result stated in the introduction, that the set of points of S on collision orbits is comparable to the set of halting inputs for a Turing machine. Note here that there are a few obvious differences between the (S, g) pair of this example and that in example 1: first, in the current example we have a countably infinite, not a finite, partition of S, and second, in this example $g(R_a) \cap R_b$ is nonempty for all a, b. The first of these differences is unimportant, at least for the purpose of comparing the question of when the (S, g) pairs give finite output, for the reason that even if an (S, g) pair has access to an infinite number of regions, in any finite time it only enters finitely many of them. The second difference is also superficial; given the (S, g) pair of a Turing machine, we can consider new pairs of the form (S, g^N), for any N; generically, for large N, we expect $g^N(R_{i\alpha}) \cap R_{j\beta}$ to be nonempty for all i, j, α, β. Note that when we argue that these differences are unimportant, we mean *for the purpose of determining the complexity* of the problem.

3. Consider next the flow of the hamiltonian system (the planar isosceles three body problem) given by

$$H = p_1^2 + \frac{2+\epsilon}{4\epsilon} p_2^2 - \frac{1}{x_1} - \frac{4\epsilon}{(x_1^2 + 4x_2^2)^{\frac{1}{2}}} \tag{5}$$

($\epsilon > 0$, $x_1 \geq 0$) on the smooth manifold M defined as the surface of constant negative energy $H = -e$, $e > 0$, with the invariant submanifolds corresponding to homothetic orbits (orbits along $x_2 = 0$ and $x_2^2/x_1^2 = 3/4$) removed. In [3] it was shown that $\{x_1 = 0, x_2 \neq 0\}$ is a forward singular cross section for this flow. Any orbit through a point $y \in S$ will return to S, or reach collision at $x_1 = x_2 = 0$, in finite time, and we define $\#(y)$ to be the number of intersections of the orbit with the set $\{x_2 = 0\}$ until this return. This gives a partition of S, as in example 2, but this time there is no reason why $\#(y)$ need be greater than 0. Thus the computation associated to the (S, g) pair obtained from this forward singular cross section computes a sequence of nonnegative integers. As in example 2, the points of S on collision orbits are the inputs that give rise to finite outputs. The points of S that lie on escape orbits (i.e. for which $x_2 \to \infty$ as $t \to \infty$) are points for which the output of the (S, g) pair is zero for all but a finite number of entries; comparing with example 1, this is similar to the set of inputs to a Turing machine for which the Turing machine does not halt, but for which after a certain finite time the internal state of the machine becomes fixed in a certain predetermined state or set of states. This is the second main result stated in the introduction.

4.Are all (S, g) pairs of the same complexity? It is straightforward to construct an (S, g) pair whose output, in one time step, determines whether the input point lies in a particular subset of S. But there exist subsets of any manifold S for which the membership function should certainly not be computed in finite time by a Turing machine. So it follows that we need to limit the class of (S, g) pairs we consider, and we do this by requiring that g be continuous except on a countable union of 1-dimensional submanifolds of $S - \cup C_i$.

Our conjecture is that, with this limitation, all (S, g) pairs are of the same complexity. This is a highly nontrivial conjecture. To display some of its content, let us consider another computation associated with the PI3BP. In the notation of section 3, example 3, let $S = \{x_2^2/x_1^2 = 3/4, x_1 \neq 0\} \cup M$. This is the union of two planes, and the flow is transverse to S. S has a natural partition: it is straightforward to show that any orbit through a point of S must meet the coordinate axes before returning to S, and we determine the region that a point on S lies in by counting the number of times it meets the coordinate axes before returning (this number can be infinite, for a point on S that escapes). It is straightforward to check this does indeed define a partition, and thus although we do not have an (S, g) pair, since there are open sets of S (the escape orbits) upon which the Poincaré return map is not defined, we have a naturally associated computation, that computes, for each point in S, a sequence of elements of $\boldsymbol{N} \cup \{\infty\}$, which can be either infinite consisting only of positive integers, or finite terminating in ∞ (corresponding to escape) or finite terminating in an integer, (corresponding to collision). ∞ can only appear as the last element in a finite sequence. Suppose now that the Poincaré return map was only not defined on some finite set of open sets of S; we could then perform a simple surgery on S to remove these sets, and use the return map to construct an (S, g) pair, which would compute the above described sequence, and for which the halting problem would be equivalent to the problem of identifying an escape orbit, contradicting our conjecture. Thus our conjecture makes a prediction that the number of open sets on S of immediate escape orbits is inifnite; we are not aware of whether this question has been investigated.

Acknowledgements

JS thanks Radel Ben-Av, David Kessler, Moshe Koppel and Steve Shnider for helpful discussions, and acknowledges being supported as a Guastella fellow by the Rashi Foundation, and also by a grant from the Bar Ilan University Research Authority. Echoing Moore's sentiments, we request that noone derive, directly or indirectly, military benefit from this work.

Bibliography

[1] C.Moore, *Phys.Rev.Lett* **64** (1990) 2354; *Nonlinearity* **4** (1991) 199.
[2] R.Penrose, *The Emperor's New Mind*, Oxford University Press (1989).
[3] R.L.Devaney in *Ergodic Theory and Dynamical Systems I*, ed.A.Katok, pub.Birkhäuser, Boston (1981).
[4] E.Fredkin and T.Toffoli, *Int.J.Theo.Phys.* **21** (1982) 219.
[5] L.Blum,M.Shub and S.Smale, *Bull.Amer.Math.Soc.* **21** (1989) 1.
[6] H.T.Siegelmann and E. D. Sontag, *Theo.Comp.Sci.* **131** (1994) 331.
[7] S.Wolfram, *Phys.Rev.Lett* **54** (1985) 735.

Figure Captions

1. The baker transformation.
2. The partition for the (S, g) pair associated with a Turing machine with D internal states.
3. Possible mappings of $R_{i\alpha}$ by g of the (S, g) pair associated with a Turing machine.