AMS Proceedings Style
Volume **00**, 0000

Hamiltonian Systems as Machines Over the Reals

Jeremy Schiff

Abstract. I explain how to associate computations with physical systems such as finite degree of freedom hamiltonian systems, and thus show why we should consider such systems as real number machines. I make a few comparisons between these real number machines of those and Blum, Shub and Smale (the question of equivalence is still open), and also make some general comments on complexity in physics.

1. Introduction

Two main motivations have been given to date for studying real number machines *à la* Blum, Shub and Smale [**BSS**]. First, this is the relevant framework for studying the complexity of most numerical algorithms. Second, as the first example given in [**BSS**] shows, this gives a framework for discussing the complexity of fractal sets and their relatives, a challenge originally posed by Penrose [**P**].

I wish here to motivate the idea that finite dimensional Hamiltonian systems (such as the classical three body problem), and for that matter other simple, idealized physical systems, can be regarded as real number machines. This gives an added motivation for studying the BSS model and its relatives: this is an appropriate framework for discussing the complexity of physical systems, another issue discussed by Penrose [**P**]. The work presented here was developed in collaboration with Hava Siegelmann of the Technion [**SS1**].

Let us step back to review existing work on the complexity of physical systems. There are three main directions that have been explored:

1. Fredkin and Toffoli [**FT**] have shown that it is possible to construct a simple physical system (a "billiard ball computer") that can do the job of any Turing machine. Others have followed in this track, including more recently Moore [**M**], constructing different physical systems that do the tasks of digital computers. This has led to the moral that physical systems are computationally equivalent to Turing machines; at least noone has yet built a physical system that can solve the halting problem. (See, however, [**SS2**],

1991 *Mathematics Subject Classification.* Primary 68Q05, 70F07; Secondary 03D10, 68Q15.
The author is supported by a Guastella fellowship from the Rashi foundation.

0000-0000/00 $1.00 + $.25 per page

where we show that "external advice" can be built into physical systems, through for example coupling constants, to give physical systems super-Turing capabilities.)

2. Pour-El and Richards [**PR**] have shown — as explained by Marian Pour-El in her lecture in this workshop — that unbounded operators need not preserve computability, implying that even if the inital state of a physical system is precisely known, its state after a finite time need not be even computable. This is usually cited as evidence that suitably physical systems can do things which digital computers cannot. Similar results have been obtained in the context of information base complexity theory [**W**].
3. Finally, many authors have explored "quantum computers". As is well known, the uncertainty principle of quantum mechanics implies that physical processes need not have a unique outcome. Thus a quantum computer might be somewhat "fuzzy", possibly endowing it with super-Turing capabilities. Penrose [**P**] has already pointed out that the scales on which such effects are visible are not typically attained outside the laboratory, so it is doubtful whether they are of practical importance.

Having already criticized the third of these lines of investigation, permit me to make some observations about the other two as well. The Pour-El and Richards results, while interesting in their own right, can also have no physical relevance, because the effects of uncomputability are only visible assuming we can observe the state of a continuum physical system to infinite accuracy — this is of course impossible. So we should certainly not jump from the Pour-El and Richards results to the conclusion that we should be able to build super-Turing analog machines. On the other hand, Fredkin and Toffoli's approach has the deficiency that while they find physical systems that can do the job of digital computers, the physical systems that arise this way are somewhat simpler than realistic ones (specifically, one does not need to numerically solve differential equations to compute their motion).

The conclusion we have reached is that realistic physical systems, like the classical three body problem, should be considered as a kind of real number machine, possibly equivalent to the BSS model, thought this remains to be seen. To persuade you of this I will proceed in the *opposite* direction from that of Fredkin and Toffoli: instead of finding physical systems that perform the function of digital computers, I will discuss — in broad terms — how to associate certain computations with realistic physical systems (systems of point particles). This is the content of section 2. We will study the planar isosceles three body problem as an example. In section 3, I will explain why — given the kinds of computations they can perform — physical systems should be viewed as real number machines, and discuss differences and similarities with the BSS model. And in conclusion, in section 4, I will make some comments on Penrose's problem of the complexity of physical systems.

2. The Computations Performed by Systems of Point Particles

Consider a system of a finite number of point particles, whose motion is governed by some hamiltonian. The motion of the system is described by an orbit on a constant energy surface of an appropriate phase space[1]. The Pour-El-Richards

[1]If there are other conserved quantities, then we restrict to level sets of these too.

approach to computability in physics assumes that from observation of the physical system we can determine this orbit *precisely*, and therefore focuses on the computability of points on the orbit. In practice, this goal is too optimistic. The best we can do in measurement terms, is to divide the energy surface up into a countable number of open sets, and observe, at specified time intervals, in which open set (or sets) the orbit lies (since we are dealing with open sets, the precise determination of the time will be conveniently unimportant).

The use of open sets in the previous paragraph is critical. To determine whether an orbit is in a particular closed set (for example, at a specific point) in a certain time interval, cannot necessarilly be done without precise knowledge of the orbit, which I have explained is unattainable. There is one exception to this rule that I wish to make, essentially to compensate for a flaw that exists in the assumption of "point particles". I will say that collision — of two or more particles — *is* a measurable phenomenon. For perfect point particles, determining collision is determining membership of a closed set; in practice though, internal structure of whatever physical model of particles we are using, makes collision a determinable event. We assume, of course, that the internal degrees of freedom are irrelevant for the rest of our analysis of the motion.

Let us assign symbols $A, B, C, ...$ to the different kinds of available collision (a finite number). In addition, assign symbols $1, 2, 3, ...$ to the open sets (a countable number) we have used to carve up the relevant energy surface. A typical computation we can say such a physical system performs is that starting from some initial set of coordinates in phase space, it computes of a list corresponding to the configuration of the system at regular intervals: each element of the list is either a collision symbol, or a set of symbols denoting the open sets in which the system is to be found at the relevant time. Many variations on this theme can be thought of; we will see one shortly. Of course, we have in this way associated a large number of machines to our physical system, depending on how coarsely we carve up the constant energy surface; there is a natural partial ordering on the set of machines we obtain this way. The limit, in which the division of the energy surfaces becomes infinitely fine, is a singular one; this is just the limit where we consider the physical system as computing its orbit exactly, and the noncomputability results of Pour-El and Richards are relevant. As I have explained, this is a limit we should avoid on physical grounds.

Prior to giving an explicit example of a computation associated with a realistic physical system, let me just mention another kind of computation which we *cannot* say physical systems perform, like the computation of the exact coordinates of their orbits. Frequently in multiparticle systems, it is possible to categorize orbits by their "final motion", i.e. some simple facts about what happens to the orbit as time goes to infinity. For example, in in the three body problem one can discuss *bounded* orbits, in which all the particles never get more than some finite distance from each other for large enough times, *hyperbolic* orbits, in which for large enough times the distance between any particle and the origin grows (with the particles' speeds not tending to zero) and so on [**AKN**]. Often it is possible, from finite time observation of an orbit, to draw conclusions about its final motion. But generically, this is not so. We thus should avoid saying a system computes its final motion. Indeed we shall suggest later that the question of determining the final motion of a

system is of a level of complexity greater than the halting problem for real number machines.

We now move to the promised example of the planar isoscles three body problem (PI3BP). In the PI3BP, we consider the motion under the effect of gravity of 3 bodies in a plane, two having equal mass M, and the other with mass m. If at some instant the positions and momenta of the two masses M are mirror images of each other in the line along which the mass m is moving, then it is apparent that the system is in an isosceles triangle configuration, and that it will stay in an isosceles triangle configuration forever. The configuration is described by two coordinates, x, half the base of the isosceles triangle, and y, its height (see Figure 1), and the hamiltonian is

$$H = \frac{1}{4}p^2 + \frac{m+2M}{4m}q^2 - \frac{M}{2x} - \frac{2m}{\sqrt{x^2+y^2}},$$

where p is the canonical momentum associated to x and q that associated to y. For negative energies, it is straightforward to check that x is bounded, i.e. the two masses M can never escape from each other. It turns out furthermore that the two masses M are guaranteed to collide with each other in a finite time. Provided the collisions do not involve also the third mass m, it is straightforward to continue the flow through such a collision (a "binary collision") — the two masses M just bounce off each other. The motion of the mass m is however much less predictable. There are orbits in which y remains bounded, and there are others in which the mass m escapes ($|y| \to \infty$). Triple collision is possible. There is a very nice computation we can associate with a negative energy orbit. Between each binary collision, the mass m will cross the line joining the two masses M a number of times (possibly zero, always finite). We record, for a given set of initial conditions, these numbers. Thus we get a sequence of nonnegative integers. But it might happen that triple collision is reached; in that case we record an "A" at the end of our computed sequence of integers, and terminate the computation. So, finally, we have a computation associated with the system that computes a sequence — possibly finite, possibly infinite — of nonnegative integers, with an "A" at the end if the sequence is finite.

The sequences that emerge this way, known as the "symbolic dynamics" of the orbit under study, have been exhaustively studied some 15 years ago, and we refer the reader to the article [**D**]. It has been proven that all possible finite or infinite sequences can arise this way. The ones that terminate in an infinite string of 0's are particularly interesting, because these correspond precisely to "escape orbits", where after a certain time the mass m never returns to cross the line joining the two masses M. We will return to these sequences later. The computation of the symbolic dynamics associated with some orbit clearly is a type of computation we "permit": apart from the need to recognize collisions, there is never any need to make an exact determination of the configuration to compute a finite number of terms in the symbolic dynamics.

3. Hamiltonian Systems are Real Number Machines

To recap: we have seen that in various ways we can regard physical systems as machines, that compute, from some given initial coordinates in phase space, a list of elements in a countable alphabet. The reason to regard these as *real number*

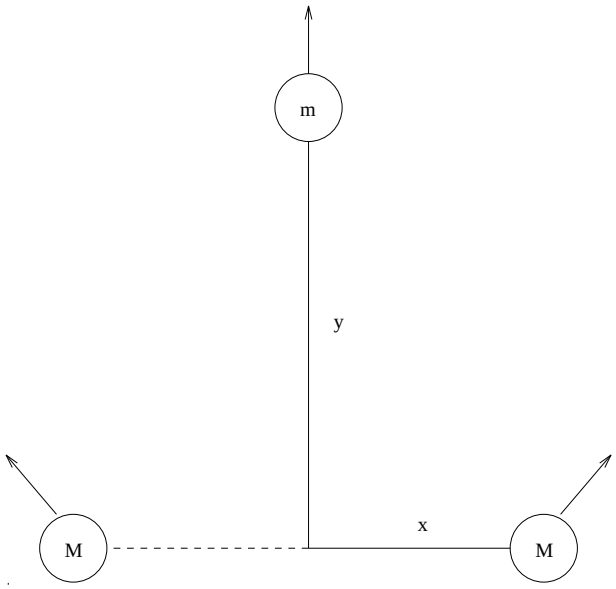

FIGURE 1. The Planar Isosceles Three Body Problem

machines is very simple — the input consists of (a finite set of) real numbers, and the system/machine, since it has a continuum of possibilities for output[2], must be processing the entirity of the information in the input — or at least not just a countable number of bits of its input — to reach its output.

I feel this point requires some further motivation, and I will go about this as follows. It should be clear that a machine of the kind we have been discussing (not hamiltonian, though, I admit) can do the job of the first example of a machine given in [**BSS**]. Suppose $g : \mathbf{C} \to \mathbf{C}$ is a polynomial map. Consider a particle moving on the complex plane, governed by the law of motion that if the particle is at position z at time $t = n$, where n is an integer, then between $t = n$ and $t = n + 1$, the particle moves, with speed $|g(z) - z|$ along a line from z to $g(z)$. At $t = n$, for all integers n, we record a "1" if the particle is in $|z| > C_g$ and a "2" if the particle is in $|z| < 2C_g$, where, as in [**BSS**], C_g is such that if $|z| > C_g$ then $|g^k(z)| \to \infty$ as $k \to \infty$. Our system will output a list of elements of the set $\{\{1\}, \{2\}, \{1, 2\}\}$, and — identifying $\{1\}$ as a "halting state" — we see the halting set of this machine is exactly the halting set of the first example in [**BSS**].

This machine we have just described is actually not obviously a real number machine in the sense of the first paragraph of this section (that it takes real number input, and uses "real number encoded information" in its processing). If $g(z) = z^2$, and the input z is given by supplying its modulus and argument, the processing of this machine is especially simple, and only requires a finite amount of information from the input. This is of course not the case for more general $g(z)$. All I am saying here is that a real number machine should not be called such if it takes real input,

[2]This is of course true for systems like the PI3BP.

but then doesn't actually use it. On the other hand, the criterion of real input is surely the minimal criterion we should impose on a real number machine.

So now we have to raise the question of whether the machines defined here are equivalent to BSS machines. BSS machines are imparted with the capability of real number output when they halt, but this would be a harmless feature to add to our machines (we would modify the dynamics of the system under consideration such that on meeting certain conditions the dynamics becomes trivial, allowing us potentially infinite observational capabilities from then on). In general, it seems that the machines discussed in this paper can certainly do the job of BSS machines. The converse, however, is far less obvious. At each step of a BSS machine, the relevant real number computation is calculation of a polynomial map, or possibly a rational map. This restriction was imposed in [**BSS**] to be able to formulate complexity theory in an algebraic framework. During each time step, hamiltonian systems typically integrate certain differential equations (without, of course, revealing to us the exact answers); the mapping associated with a time step will almost certainly not be polynomial, but on the other hand, unless there are singularities in the hamiltonian, it will be smooth, allowing polynomial or rational approximation. This is an interesting problem for further study.

4. The Complexity of Physical Systems

The aim of all this work is to provide a framework for discussing the complexity of physical systems. While the phrase "the complexity of physical systems" is used widely in the literature, correctly speaking, and as already noted by Penrose [**P**], there is no such thing, and one must discuss the complexity of different *questions* about physical systems separately.

A typical question for a physical system is the identification of a set of orbits satisfying a certain property. In the PI3BP, there are two obvious questions of interest, that of identifying all triple collision orbits, and that of identifying escape orbits. For the corresponding machine that we have defined, triple collision orbits correspond to inputs for which the output is finite, that is, ends in "A". Identifying the set of inputs which are finite, or the set of inputs which at some (unspecified) time have a particular symbol in their output string, is in an obvious way similar to identifying the set of halting inputs for a Turing machine, and thus it is natural to define R.E. sets for physical systems as sets of orbits which correspond, in the way we have just seen, to halting sets for some machine associated with the system.

We believe, but as of yet cannot prove, that the set of escape orbits is *not* an R.E. set. For the particular machine we have defined associated with the PI3BP, escape orbits correspond to inputs which lead to outputs terminating in an infinite string of zeros (the mass m, after some finite time, never returns to cross the line joining the two masses M). The problem of identifying inputs which reach and forever stay in a particular "state" is qualitatively harder than that of identifying inputs which simply, at some unspecified time, reach that state. This is an unfamiliar result, mainly because for standard Turing machines, it is not true. For standard Turing machines, because the input tape is finite, and throughout the computation we can keep track of the length of the tape, once the machine reaches a certain state, we just have to check that it stays in that state for a certain *finite* time to be sure that it will stay in that state forever. For the machine we have

associated with the PI3BP, Devaney's theorem suggests[3] that this is not the case: the mass m can continue to head away from the masses M for *any* length of time before returning, so to confirm escape, we really need to follow the mass m for an indefinite time.

Assuming the correctness of the analysis in the previous paragraph (and there is much to check), we arrive at the conclusion that there is a *qualitative difference in complexity* between the questions of identifying the set of triple collision orbits and identifying the set of escape orbits in the PI3BP. The former is an R.E. set, the latter is not. Physically, this is no surprise at all — it just restates the obvious, that triple collision is a finite time effect, but escape is not. The mathematical detail lacking is the proof that there exist orbits in the PI3BP that cannot be recognized as escape orbits in any finite time (there certainly exist some escape orbits that *can* be recognized in finite time); there is some good evidence for this, though.

Furthermore, despite the fact that the set of escape orbits is, we believe, not R.E., it *does* arise as the set of inputs to a real number machine which produce an output which gets "stuck" in one particular state. It is natural, therefore, to propose that just as standard non R.E. sets can be classified further into a hierarchy of complexity classes, so non R.E. sets for real number machines fall into a hierarchy, and the set of escape orbits in the PI3BP is just an instance of the lowest class in this hierarchy, which comprises the sets that are non R.E., but that are the sets of inputs to real number machines which give outputs that get "stuck" in some state.

Apparently there is much more work to do here, both in the area of reassesment of the PI3BP, and in the area of formalising the rather loose notions of "machines associated with physical systems" and so on that we have introduced. The overall consistency of our picture, however, persuades us that our basic identification of simple, realistic physical systems as a kind of, or possibly even a paradigm of, real number machine, is correct, and central in the discussion of complexity in physics.

References

[AKN] V.I.Arnold, V.V.Kozlov and A.I.Neihstadt, *Mathematical Aspects of Classical and Celestial Mechanics*, in *The Encyclopaedia of Mathematical Sciences, Dynamical Systems III*, ed. V.I.Arnold, Springer Verlag, 1985.

[BSS] L.Blum,M.Shub and S.Smale, *On a Theory of Computation and Complexity over the Real Numbers: NP-Completeness, Recursive Functions and Universal Machines*, Bull.Amer.Math.Soc. **21** (1989), 1–46.

[D] R.L.Devaney, *Singularities in Classical Mechanical Systems*, in *Ergodic Theory and Dynamical Systems I*, ed. A.Katok, Birkhäuser, 1981.

[FT] E.Fredkin and T.Toffoli, *Conservative Logic*, Int.J.Theo.Phys. **21** (1982), 219–253.

[M] C.Moore, *Unpredictability and Undecidability in Dynamical Systems*, Phys.Rev.Lett. **64** (1990), 2354–2357.

[P] R.Penrose, *The Emperor's New Mind*, Oxford, 1989.

[PR] M.Pour-El and I.Richards, *Computability in Analysis and Physics*, Springer-Verlag, 1989.

[SS1] J.Schiff and H.T.Siegelmann, *On the Complexity of the Anisotropic Kepler Problem and the Planar Isosceles Three Body Problem,* preprint (1995).

[SS2] H.T.Siegelmann and J.Schiff, *Welcoming Super-Turing Theories,* preprint (1995).

[3] In fact, Devaney's theorem only tells us about the behavior of the mass m between collisions of the mass M, and doesn't say anything about the frequency of collisions of the masses M and other relevant information. Numerical simulations seem consistent, however, with the picture I am presenting here.

[W] A.Werschulz, *The Computational Complexity of Differential and Integral Equations*, Oxford, 1991.

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