論文の内容の要旨

Uniform computational complexity of ordinary differential equations with applications to dynamical systems and exact real arithmetic

（常微分方程式の一様計算量およびその力学系と厳密数値計算への応用）

ティース　ホルガー
Holger Thies

Dynamical systems are used to model a large number of processes in nature whose state evolves over time. They are usually described by ordinary differential equations in the time-continuous or function iterations in the time-discrete case. In both cases some of the quantities involved are real numbers.

In the theory of computation Turing machines or equivalent models of computation are used to define the notion of computability. Turing machines compute functions from finite strings to finite strings, that is, functions $F : \Sigma^* \to \Sigma^*$ for some fixed finite alphabet Σ. While computations over discrete objects like natural numbers, rational numbers or graphs can be defined by choosing an appropriate encoding for these objects as finite strings, the set of reals is uncountable and it is therefore impossible to find such an encoding. Classical computability and complexity theory can thus only be applied to problems where input and output are discrete.

Computable Analysis extends computability theory to computations with uncountable quantities such as real numbers. The theory already dates back to Turing [10] with later important contributions for example by Grzegorczyk [2]. The rigorous study of computational complexity in this model was initiated by Ko and Friedman [6]. The underlying idea is that while real numbers can not be encoded finitely, they can be approximated arbitrarily well e.g. by rationals. Thus, it is possible to encode a real number by an approximation function that gives approximations to said number up to any desired precision. The theory can be extended beyond the reals to other uncountable sets using the framework of *representations* (see e.g. [11]).

In this thesis we study several new aspects regarding the computational complexity of problems involving dynamical systems and ordinary differential equations.

The first part of the thesis deals with uniform efficient computability of operators in analysis, i.e., of functions mapping real functions to real functions. Important operators like integration or maximization have been shown to be hard from a complexity theoretical perspective (see e.g. [1, 3]). It therefore can not be expected that efficient algorithms for these operators exist on general functions. On the other hand, if we restrict our attention to the class of analytic functions, it is well known that many important operators map polynomial-time computable functions again to polynomial-time computable functions [7, 8]. The formulation of these theorems is, however, usually *non-uniform*: They do not state how to transform a description of a function to a description of the resulting function after applying the operator.

A uniform formulation requires the definition of a representation for function spaces of analytic functions. The complexity of operators can then be studied in the framework of second-order complexity [4]. Recently, some representations for one-dimensional analytic functions on different domains have been defined and uniform complexity results have been shown [5].

However, considering only one-dimensional functions does not suffice for many interesting applications including the problem of solving ordinary differential equations. The first result in this thesis is therefore a generalization of the one-dimensional representations to multidimensional analytic functions. Using these representations we show that many important operators on multidimensional analytic functions are second-order polynomial-time computable. We then apply the results to study the uniform computational complexity of solving initial value problems of ordinary differential equations with analytic right-hand side functions. We show that this problem is polynomial-time computable in terms of the output precision and some natural parameters of the right-hand side functions and thereby generalize some recent results on polynomial ordinary differential equations to the analytic case.

A main motivation for our considerations is to better understand computational properties of systems in nature. We therefore want to study the computational complexity of time-continuous dynamical systems that are used to model, for example, processes in classical physics. For this application, worst-case complexity turns out to not be an adequate model: For finite complexity bounds to exist, we have to make rather unnatural assumptions on the domain of the problem. Indeed, for many systems the computational complexity depends in some sense on the distance of trajectories to singularities of the system. Thus, while in the worst-case trajectories might come arbitrarily close to singularities, for many systems such a situation is rather unlikely. That is, the computation can be typically done efficiently even though there might be special cases where the computation time is unbounded.

Such a statement can be made formal using *average-case complexity*. We show how to apply average-case complexity to the problem of simulating a dynamical system. To this end, we characterize the computational complexity in terms of the distance of trajectories to complex singularities of the system. We then use this to show that if the probability of trajectories getting very close is small, the system can be simulated in polynomial time on average.

For an important subset of dynamical systems, the Hamiltonian systems, we can further improve that statement. Hamiltonian systems are widely used in physics to describe the motion of mechanical systems. They have the important property that volume of subsets of phase space remains constant over time. We can use this to relate the volume of subsets of phase space that are close to singularities to the volume of the initial values leading to such a state at some point in time. Hereby we can define some very general conditions under which a Hamiltonian system can be computed in polynomial time on average. As an application we show that the planar circular restricted three-body problem is average-case polynomial time computable.

Finally, we also consider the practical relevance of our theory. Our representations for analytic functions can be used as a basis for an implementation of a solver for initial value problems in exact real arithmetic. We further explore some heuristics that can be used to make the implementation more efficient in practice and provide a prototypical implementation built on the `iRRAM` `C++` framework [9].

Bibliography

[1] Friedman, H. The computational complexity of maximization and integration. *Advances in Mathematics 53*, 1 (1984), 80–98.

[2] Grzegorczyk, A. On the definitions of computable real continuous functions. *Fund. Math. 44* (1957), 61–71.

[3] Kawamura, A. Lipschitz Continuous Ordinary Differential Equations are Polynomial-Space Complete. *Computational Complexity 19*, 2 (2010), 305–332.

[4] Kawamura, A., and Cook, S. Complexity theory for operators in analysis. *ACM Transactions in Computation Theory 4*, 2 (2012), Article 5.

[5] Kawamura, A., Müller, N., Rösnick, C., and Ziegler, M. Computational benefit of smoothness: Parameterized bit-complexity of numerical operators on analytic functions and Gevrey's hierarchy. *Journal of Complexity 31*, 5 (2015), 689–714.

[6] Ko, K.-I., and Friedman, H. Computational complexity of real functions. *Theoretical Computer Science 20*, 3 (1982), 323–352.

[7] Ko, K.-I., and Friedman, H. Computing power series in polynomial time. *Advances in Applied Mathematics 9*, 1 (1988), 40–50.

[8] Müller, N. T. Uniform Computational Complexity of Taylor Series. In *Proc. 14th International Colloquium on Automata, Languages, and Programming* (1987), vol. 267 of *LNCS*, Springer, pp. 435–444.

[9] Müller, N. T. The iRRAM: Exact Arithmetic in C++. In *CCA* (2000), pp. 222–252.

[10] Turing, A. M. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society. Second Series 42* (1936), 230–265.

[11] Weihrauch, K. *Computable Analysis.* Springer, Berlin/Heidelberg, 2000.