Hypercomputation

To my son Demetrios-Georgios
and my parents
Georgios and Vassiliki

Apostolos Syropoulos

Hypercomputation

Computing Beyond the Church–Turing Barrier

Springer

Apostolos Syropoulos
366 28th October Street
GR-671 00 Xanthi
Greece
asyropoulos@gmail.com

ISBN 978-0-387-30886-9 e-ISBN 978-0-387-49970-3
DOI 10.1007/978-0-387-49970-3

Library of Congress Control Number: 2008923106

ACM Computing Classification System (1998): F.4.1, F.1.1, F.1.0
Mathematics Subject Classification (2000): 03D10, 03D60, 03D99, 68Q05, 68Q10, 68T99

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Preface

Hypercomputation in a Nutshell

Computability theory deals with problems and their solutions. In general, problems can be classified into two broad categories: those problems that can be solved algorithmically and those that cannot be solved algorithmically. More specifically, the design of an algorithm that solves a particular problem means that the problem can be solved algorithmically. In addition, the design of an algorithm to solve a particular problems is a task that is equivalent to the construction of a Turing machine (i.e., the archetypal conceptual computing device) that can solve the same problem. Obviously, when a problem cannot be solved algorithmically, there is no Turing machine that can solve it. Consequently, one expects that a *noncomputable* problem (i.e., a problem that cannot be solved algorithmically) should become *computable* under a broader view of things. Generally speaking, this is not the case. The established view is that only problems that can be solved algorithmically are actually solvable. All other problems are simply noncomputable.

Hypercomputation deals with noncomputable problems and how they can be solved. At first, this sounds like an oxymoron, since noncomputable problems cannot really be solved. Indeed, if we assume that problems can be solved only algorithmically, then this is true. However, if we can find other ways to solve noncomputable problems nonalgorithmically, there is no oxymoron. Thus, hypercomputation is first about finding general nonalgorithmic methods that solve problems not solvable algorithmically and then about the application of these methods to solve particular noncomputable problems. But are there such methods? And if there are, can we use them to solve noncomputable problems?

In the early days of computing, for reasons that should not concern us for the moment, a Turing machine with an *oracle* was introduced. This oracle was available to compute a single arbitrary noncomputable function from the natural numbers to the natural numbers. Clearly, this new conceptual computing device can be classified as a *hypercomputer* since it can *compute* noncomputable functions. Later on, other variants of the Turing machine capable of *computing* noncomputable functions appeared in the scientific literature. However, these extensions to computability theory did not gain widespread acceptance, mainly because no one actually believed that one could *compute* the incomputable. Thus, thinkers and researchers were indirectly discouraged from studying and investigating the possibility of finding new methods to solve problems and compute things. But the 1990s was a renaissance for hypercomputation since a considerable number of thinkers and researchers took really seriously the idea of *computing* beyond computing, that is, hypercomputation. Indeed, a number of quite interesting proposals have been made ever since. And some of these proposals, although quite exotic, are *feasible*, thus showing that hypercomputation is not to the theory of computation what perpetual motion machines are to physics!

The success of the Turing machine in describing everything computable, and also its simplicity and elegance, prompted researchers and thinkers to assume that the Turing machine has a universal role to play. In particular, many philosophers, psychologists, and neurobiologists are building new theories of the mind based on the idea that the mind is actually a Turing machine. Also, many physicists assume that everything around us is a computer and consequently, the whole universe is a computer. Thus, if the universe is indeed a Turing machine, the *capabilities* of the mind and nature are limited by the capabilities of the Turing machine. In other words, according to these views, we are tiny Turing machines that live in a "Turing-verse"!

Hypercomputation poses a real threat to the cosmos described in the previous paragraph. Indeed, even today it is considered heretical or even unscientific to say that the mind in not a Turing machine! And of course, a universe where hypercomputation is possible renders certain beliefs and values meaningless. But then again, in the history of science there are many cases in which fresh ideas were faced with skepticism and in some instances with strong and prudent opposition. However, sooner or later, correct theories and ideas get widespread appreciation and acceptance. Thus, it is crucial to see whether there will be "experimental" verification of hypercomputation. But this is not an easy task, since hypercomputation is practically in its infancy. On the other hand, it should be clear that there is no "experimental" evidence for the validity of the Turing-centered ideas presented above.

Reading This Book

Who Should Read It?

This book is a presentation, in a rather condensed form, of the emerging theory of hypercomputation. Broadly, the book is a sort of compendium of hypercomputation. As such, the book assumes that readers are familiar with basic concepts and notions from mathematics, physics, philosophy, neurobiology, and of course computer science. However, since it makes no sense to expect readers to be well versed in all these fields, the book contains all the necessary definitions to make it accessible to a wide range of people. In particular, the book is well suited for graduate students and researchers in physics, mathematics, and computer science. Also, it should be of interest to philosophers, cognitive scientists, neurobiologists, sociologists, and economists with some mathematical background. In addition, the book should appeal to computer engineers and electrical engineers with a strong interest in the theory of computation.

About the Contents of the Book

The book is based on material that was readily available to the author. In many cases, the author directly requested copies of papers and/or book chapters from authors, and he is grateful to everyone who responded positively to his request. It is quite possible that some (important?) works are not discussed in this book. The reasons for any such omission are that the author did not really feel they were that important, that the author did not have at his disposal the original material describing the corresponding piece of work, or that the author simply was unaware of this particular piece of work.

For the results (theorems, propositions, etc.) that are presented in the book we have opted not to present their accompanying proofs. Since this book is an introduction to the emerging field of hypercomputation, it was felt that the proofs would only complicate the presentation. However, readers interested in proofs should consult the sources originally describing each piece of work.

The subject index of the book contains entries for various symbols, and the reader should be aware that there is only one entry for each symbol, and the unique entry corresponds to the page where the symbol is actually defined.

Mathematical Assumptions

At this point it is rather important to say that the discussion in the next nine chapters assumes that the Axiom of Choice holds. In other words, many of the ideas presented do not make sense without this axiom being valid. This axiom states that

Axiom of Choice There exists a choice function for every system of sets [88].

Assuming that S is a system of sets (i.e., a collection of sets only), a function $g : S \to S$ is called a *choice function* for S if $g(X) \in X$ for all nonempty $X \in S$. After this small but necessary parenthesis let us now describe the contents of each chapter.

The Book in Detail

The first chapter is both an introduction to hypercomputation and an overview of facts and ideas that have led to the development of classical computability theory. In addition, there is a short discussion explaining why hypercomputation is so fascinating to many thinkers and researchers.

The second chapter can be viewed as a crash course in (classical) computability theory. In particular, we discuss Turing machines, general recursive functions, recursive predicates and relations, and the Church–Turing thesis, where we present not only the "classical" version, but even quite recent versions that encompass "modern" views.

In the third chapter we begin the formal presentation of various approaches to hypercomputation. In particular, in this chapter we present early approaches to hypercomputation (i.e., proposals that were made before the 1990s). Although some proposals presented in this chapter are quite recent, we opted to present them here, since they are derivatives of certain early forms of hypercomputation. More specifically, in this chapter we present trial-and-error machines and related ideas and theories, inductive Turing machines, coupled Turing machines, Zeus machines, and pseudorecursiveness.

Conceptual machines that may perform an infinite number of operations to accomplish their computational task are presented in the fourth chapter. Since the theory of these machines makes heavy use of cardinal and ordinal numbers, the chapter begins with a brief introduction to the relevant theory. Then, there is a thorough presentation of infinite time Turing machines and a short description of infinite time automata. In addition, there is a description of a "recipe" for constructing infinite machines, and the chapter concludes with a presentation of a metaphysical foundation for computation. Notice that infinite-time Turing machines are the ideal conceptual machines for describing computations that take place during a supertask. Thus, it should be more natural to present them alongside the supertasks; however, it was felt that certain subjects should be presented without any reference to related issues. On the other hand, other subjects are presented in many places in the book so as to give a thorough view of them.

Interactive computing is known to every computer practitioner; what is not known is that interactive systems are more powerful than Turing machines. The fifth chapter begins by explaining why this is true and continues with a presentation of various conceptual devices that capture the basic characteristics of interactive computing. In particular, we discuss interaction machines, persistent Turing machines, site and Internet machines, and the π-calculus.

Is the mind a machine? And if it is a machine, what kind of machine is it? What are the computational capabilities of the mind? These and other similar questions are addressed in the sixth chapter. However, it is rather important to explain why we have opted to discuss these questions in a book that deals with hypercomputation. The main reason is that if one can show that the mind is, among other things, a computational device that has capabilities that transcend the capabilities of the Turing machine, then, clearly, this will falsify the Church–Turing thesis. In other words, hypercomputation partially falsifies computationalism. In this chapter we discuss various approaches to show that the mind is not just a Turing machine, but a *device* with many capabilities both computational and noncomputational. In particular, we discuss arguments based on Gödel's incompleteness theorems, arguments from the philosophy of mind, the relation between semiotics and the mind, and the mind from the point of view of neurobiology and psychology.

The theory of computation deals primarily with natural numbers and functions from natural numbers to natural numbers. However, in physics and analysis we are dealing with real numbers and real functions. This implies that it is important to study the computational properties of real numbers and real functions. And real-number computation leads to hypercomputation in unexpected ways, which we discuss in the seventh chapter of the book. In particular, we discuss various approaches to real-number computation and how they may lead to hypercomputation. We begin with the Type-2 Theory of Effectivity, and continue with a discussion of a special form of Type-2 machines. Next, we present BSS-machines, real-number random access machines, and we conclude with a presentation of a recursion theory on the reals.

In the eighth chapter we discuss relativistic and quantum hypercomputation. More specifically, we show how the properties of space and time can be exploited to compute noncomputable functions. Also, we show how quantum computation can be employed to compute noncomputable problems. In addition, we present our objections to a computational theory of the universe. There is also a brief discussion of supertasks in the framework of classical and quantum mechanics.

The last chapter is devoted to natural computation and its relationship to hypercomputation. It is worth noticing that natural computation includes analog computing, and that is why we present various approaches to hypercomputation via analog computation. In addition, we demonstrate how one may end up with noncomputable functions in analysis and physics and, thus, showing in an indirect way, that noncomputability is part of this world. The chapter concludes with a presentation of an optical model of (hyper)computation, membrane systems as a basis for the construction of hypermachines, and analog X-machines and their properties.

The book includes four appendices. The $P = NP$ hypothesis is discussed in the first appendix. In the second appendix we briefly discuss how hypercomputation affects complexity theory. In the third appendix, we discuss how noncomputability affects socio-economic issues. The last appendix contains some useful mathematical definitions, necessary for the understanding of certain parts of the book. Clearly, this appendix is not a substitute for a complete treatment of the subject; nevertheless, it can be viewed as a refresher for those already exposed to the concepts or as a very brief introduction to the relevant theory for those with no prior knowledge of the relevant definitions.

Acknowledgments

First of all, I would like to express my gratitude to Ioannis Kanellos for his many comments and suggestions. Our long discussions over the phone were quite stimulating and thought-provoking. Also, I would like to thank Wayne Wheeler, Springer's computer-science editor, for believing in this project and for all his help and assistance, and Ann Kostant, my editor at Springer, for her help and assistance. In addition, I am really thankful to Francisco Antonio Doria, Martin Ziegler, Joel David Hamkins, Benjamin Wells, Bruno Scarpellini, Dina Goldin, Peter Wegner, Mark Burgin, Tien D. Kieu, John Plaice, Mike Stannett, Theophanes Grammenos, and Andromahi Spanou for reading drafts of the book and providing me with many valuable comments and suggestions on how to improve the presentation. Also, I would like to thank the Springer reviewers for critically reading drafts of the book and providing me with their valuable comments and suggestions. In addition, I thank David Kramer for his excellent work in copyediting the manuscript. Naturally, for any omissions and/or remaining errors and mistakes one should blame only the author and nobody else! Furthermore, I would like to thank Barbara Beeton, Jaako Hintikka, Mike Stannett, Petros Allilomes, and Peter Kugel for providing me with copies of important papers. Last, but certainly not least, I would like to thank Yannis Haralambous for his help and Maria Douma for the drawing on page 8.

Apostolos Syropoulos
Xanthi, Greece

March, 2008

Contents