

# EXPLAINING SIMULATED PHENOMENA

## A DEFENSE OF THE EPISTEMIC POWER OF COMPUTER SIMULATIONS

Von der Philosophisch-Historischen Fakultät  
und dem Stuttgart Research Centre for Simulation Technology  
der Universität Stuttgart zur Erlangung der Würde eines  
Doktors der Philosophie (Dr. phil.) genehmigte Abhandlung

Vorgelegt von

JUAN M. DURÁN

aus Córdoba (Argentinien)

Hauptberichterin: JProf. Dr. Ulrike Pompe-Alama (SimTech - Universität Stuttgart)  
Mitberichterin: Prof. Dr. Catrin Misselhorn (Universität Stuttgart)  
Mitberichter: JProf Dr. Gregor Betz (Karlsruher Institute of Technology)

Tag der mündlichen Prüfung: 14 August 2013

Institut für Philosophie der Universität Stuttgart  
Stuttgart Research Center for Simulation Technology

2013

All rights reserved.

© 2014 by Juan M. Durán

Institut für Philosophie

Universität Stuttgart

Seidenstr. 36

D-70174 Stuttgart

D 93 - Dissertation der Universität Stuttgart

# Contents

<b>Acknowledgements</b>	<b>6</b>
<b>List of figures and algorithms</b>	<b>7</b>
<b>Abstract</b>	<b>8</b>
<b>Introduction</b>	<b>9</b>
Background and motivation . . . . .	10
Scope and outline . . . . .	12
<b>1 A scientific image</b>	<b>31</b>
1.1 Introduction . . . . .	31
1.2 Theories and models . . . . .	33
1.2.1 Theories . . . . .	33
1.2.2 Models . . . . .	36
1.2.2.1 Representation . . . . .	39
1.2.2.2 A taxonomy for scientific models . . . . .	42
1.3 Experimentation . . . . .	45
1.3.1 The epistemology of experimentation . . . . .	46
1.4 Hot new stew . . . . .	53
1.4.1 Recasting models . . . . .	55
1.4.2 Changes in ontology . . . . .	56
1.4.3 Representational languages . . . . .	57
1.4.4 Abstraction in computer simulations . . . . .	57
1.4.5 Reliability of results . . . . .	57
1.5 Conclusion . . . . .	58
<b>2 A survey of the foundations of computer science</b>	<b>62</b>
2.1 Introduction . . . . .	62
2.2 Specifications . . . . .	63

2.3	Algorithms . . . . .	70
2.4	Computer processes . . . . .	77
2.5	Conclusion . . . . .	82
<b>3</b>	<b>Computer simulations: towards their conceptualization</b>	<b>84</b>
3.1	Introduction . . . . .	84
3.2	On the notion of simulation . . . . .	86
3.2.1	The analog and the digital . . . . .	87
3.2.2	Cellular automata, agent-based simulations, and complex systems . . . . .	93
3.2.3	Equation-based simulations . . . . .	97
3.3	A methodology for computer simulations . . . . .	100
3.3.1	An example of a two-body system . . . . .	102
3.3.1.1	A working conceptualization . . . . .	106
3.3.2	Verification and validation . . . . .	107
3.3.2.1	Verification . . . . .	109
3.3.2.2	Validation . . . . .	109
3.3.3	A brief survey on errors in computers . . . . .	111
3.4	Conclusion . . . . .	113
<b>4</b>	<b>Theoretical assumptions behind scientific explanation</b>	<b>117</b>
4.1	Introduction . . . . .	117
4.2	Knowledge and understanding . . . . .	119
4.2.1	Knowledge . . . . .	120
4.2.2	Understanding . . . . .	126
4.3	Theories of scientific explanation . . . . .	130
4.3.1	Against a new theory of scientific explanation . . . . .	130
4.3.2	Deductive nomological model . . . . .	131
4.3.3	Mathematical explanation . . . . .	136
4.3.4	Model explanation . . . . .	140
4.4	Conclusion . . . . .	144
<b>5</b>	<b>Scientific explanation by computer simulations</b>	<b>148</b>
5.1	Introduction . . . . .	148
5.2	The unificationist account . . . . .	149
5.2.1	The metaphysical question . . . . .	156
5.2.2	The epistemic question . . . . .	162
5.3	Explanatory unification for a computer simulation . . . . .	166

5.3.1	The metaphysical question (revisited)	167
5.3.2	The epistemic question (revisited)	173
5.3.3	A response to the skeptic	180
5.4	Conclusion	182
<b>6</b>	<b>Conclusions and future challenges</b>	<b>185</b>
6.1	Conclusions	185
6.2	Future challenges	189
6.2.1	For the unificationist	190
6.2.2	For a general philosophy of computer simulations	194
	<b>Summary (English)</b>	<b>198</b>
	<b>Summary (German)</b>	<b>203</b>
	<b>Lexicon</b>	<b>208</b>
	<b>Bibliography</b>	<b>212</b>

# Acknowledgements

I have numerous people and institutions to thank. First, to SimTech and the University of Stuttgart for financial support during the first three years of my studies. To Gregor Betz and Niels Gottschalk-Mazouz for believing in this project from the beginning. Thanks are also due to my advisor, Ulrike Pompe-Alama, and to the members of the Institute of Philosophy.

I am in special debt to Paul Humphreys, who pointed me in the direction that this dissertation finally took. I am also thankful to his group, for the warm welcome and the stimulating discussions during my visit at UVA. I am also grateful to Stephan Hartmann, Jan Sprenger, and the members of TiLPS, for the great discussions we found ourselves in, and for making my time as fellow a remarkable moment in my life. Many thanks to Víctor Rodríguez, Marisa Velasco, Pío García, and Javier Blanco from the Universidad Nacional de Córdoba for the warm welcome, support, and stimulating discussions during each visit.

Finally, this dissertation is dedicated to the two most important women in my life. To my mother, with love and enormous gratitude, for her encouragement of continuing my studies in philosophy and of my own pursuit of happiness. This dissertation is also dedicated to Kassandra Pomper, who went through it tirelessly, correcting my English and suggesting improvements. She has been of immense support, companionship, and a loving partner during these past few years. For each warm word, loving touch, and comprehensive look, I dedicate this dissertation to these two wonderful women.

## List of figures

2.1	General schema of a specification. . . . .	67
2.2	Specification, algorithm, and computer process . . . . .	81
3.1	The elliptical orbit of a satellite relative to the planet at one focus. . . . .	103
3.2	Tidal stress represented in the simulation model. . . . .	103
3.3	The orbital eccentricity of the satellite as a function of time. . . . .	106
3.4	The lag and lead in the tidal bulge of the satellite. . . . .	106
3.5	Phases of verification and validation for a computer simulation. . . . .	111
5.1	Process of systematization of <i>K</i> . . . . .	159

## List of algorithms

2.1	Example of Hoare Triple . . . . .	71
2.2	Example of Pseudo-code . . . . .	72
2.3	Imperative programming language C. . . . .	73
2.4	Generic Algorithm 2.4. . . . .	73
2.5	Generic Algorithm 2.5. . . . .	74
2.7	Simple operation $2 + 2$ in programming language C. . . . .	81

# Abstract

Much of the recent discussion on the epistemology of computer simulations has focused excessively on comparisons with laboratory experiments. I propose, instead, to defend their epistemic power by showing how computer simulations explain simulated phenomena. Explanation, as the philosopher of science conceives it, is an epistemic notion in the sense that it yields understanding of what it explains. Given that for computer simulations such an explanation is about a simulated phenomenon, several philosophical problems follow that require attention: in virtue of what does a computer simulation explain a simulated phenomenon? What kind of understanding does such an explanation yield? This work is my attempt to answer these and other questions in a qualitative and systematic way.

The thesis has been organized in the following way: Chapter 1 discusses central problems in the philosophy of science that are relevant for the epistemology of computer simulations. It also fosters the aims and motivations for this work. Chapter 2 discusses ontological constraints underlying computer software in general. Chapter 3 draws the ontological analysis from the previous chapter and focuses on computer simulations as a special kind of computer software. In addition, it addresses methodological issues central for a working conceptualization of computer simulations. In Chapter 4 the epistemological analysis of computer simulations becomes more stringent. In particular, this chapter discusses theoretical assumptions present in theories of scientific explanation as well as the requirements that computer simulations impose on the matter. Chapter 5 discusses the unificationist theory of scientific explanation for computer simulations. This chapter elaborates on two central questions, the first regarding the process of explaining a simulated phenomenon by using a computer simulation, and the second concerning the understanding that such an explanation yields. Finally, Chapter 6 closes with the conclusions reached by this study and addresses some open questions for future challenges.

# Introduction

*Anyone who attempts to generate random numbers  
by deterministic means is, of course,  
living in a state of sin.*

John von Neumann

Living in a state of sin is, for the religious person, a humiliation of their human condition, the mortification of the lost paradise. Von Neumann compares the ignominy of sinning of those with faith to the discredit of studying random numbers by the mathematician. Although von Neumann's remark is a bit hyperbolic, it is true that no mathematician of his time would even bother to work on random numbers. The mathematical community at the time was very skeptical and unsympathetic to the future of random numbers. But this scenario changed in a swift and drastic way. Thanks to the possibility of carrying out enormous amounts of calculations, stochastic methods and random numbers together became the most important player in the emerging field of computer science.

Back in the late 1940s, computers were enormous instruments with the capacity to solve only a handful of equations, most of which depended on the mathematical abilities of the programmer and the programming abilities of the engineer. According to Aspray, the first recorded use of computers for scientific purposes took place between 1952 and 1957. By June 1952, IAS-class computers were finished and ready to be tested.<sup>1</sup> Although it was necessary to spend a few extra months repairing the computers and carrying out general maintenance, the main problem faced by the team of scientists and engineers was understanding the new instruments.

The new digital computers, built and designed on solid theoretical foundations, posed a significant challenge: it was necessary to spend some extra time learning about their operation, identifying appropriate algorithms, and determining the range of mathematical applications these machines were capable of handling.<sup>2</sup> By the time computers became more knowledgeable and reliable instruments, scientists and engineers began to use them for specific scientific applications. The calculation

on the energy band structure of iron in 1954 responsible for testing the theory of ferromagnetism was recorded as the first scientific application to be run on a digital computer.<sup>3</sup> The art of solving complex models and equations was entering a new era.

For the next sixty years, the digital computer proved to be an essential tool for the development and advancement of science. The introduction of the integrated circuit in the late '50s facilitated the pervasive use of computers in scientific practice due to their reduction in size, increase in power of computation, and simplification in the programming. Such a wide-spread use attracted the interest of scientists in all areas: physicists and chemists sought novel elements for supporting modern and improved components; mathematicians were occupied in finding new and faster methods for solving equations; and engineers constructed microchips that would increase the capacities of computers. In a similar fashion, new areas of scientific research were created, such as computer science, and others branched out from existing ones, such as electrical engineering from engineering or numerical analysis from mathematics.

Despite their relatively short history, computers have left an indelible mark on numerous and varied scientific disciplines, such as particle physics, astronomy, behavioral science, psychology, sociology, economics, and so forth. We could even assert that there is virtually no scientific discipline that has not been involved, in one way or another, with digital computers. The numerical experiment of calculating the energy band structure of iron that I mentioned earlier qualifies, in contemporary parlance, as a *computer simulation*. The present dissertation is about the use of computer simulations in scientific practice. My intention is, therefore, to defend computer simulations as epistemically powerful methods for understanding the world.

## Background and motivation

Given the importance of computer simulations in scientific practice, it is not surprising to find attempts to theorize their aims and achievements as experimental devices. Admittedly, these are questions that have been around for quite some time. As early as 1967, Thomas Naylor, Donald Burdick, and Earl Sasser, defined a computer simulation as:

a numerical technique for conducting experiments with certain types of mathematical and logical models describing the behavior of an economic system on a digital computer over extended periods of time [...] The principal difference between a simulation experiment and a 'real world' experiment is that with

simulation the experiment is conducted with a model of the economic system rather than with the actual economic system itself (Naylor et al., 1967a, 1316)

The similarity of this quotation with contemporary literature on the topic is astonishing. Current philosophical efforts are also engaged in distinguishing between a computer simulation from an empirical experiment, or in exploring the methodological implications of implementing a scientific model on the physical computer.

Yet, despite these few similarities, much of contemporary philosophical investigation is simply not the same as in 1967. From a historical perspective, the introduction of silicon-based circuits and the subsequent standardization of the circuit board significantly helped the computer industry and the growth in the computational power of computers. Such growth in speed of calculation, size of memory, and number of programming languages forcefully challenged the established ideas and fostered the search for new questions and answers. In this vein, one of the leading questions in later debates has been whether computer simulations are a new way of doing science, or if they just represent another computational method of experimentation. The work of Fritz Rohrlich sets some grounds in that direction. According to him, computer simulations do provide a qualitatively new methodology for the physical sciences, lying somewhere halfway between theoretical physics and empirical methods of experimentation.<sup>4</sup> But perhaps the most complete discussion on the issue has been between Roman Frigg and Julian Reiss on the one hand, and Paul Humphreys on the other. While Frigg and Reiss understand computer simulations in the context of a philosophy of models and, as such, with no significant differences from other uses of models in scientific practice, Humphreys believes that the unidirectionality of Frigg and Reiss's argument is misleading for it only illuminates one side of the problem and neglects the study of computer simulations in and for themselves.<sup>5</sup> A more comprehensive portrayal of computer simulations, however, raises questions that cannot be answered by a familiar philosophy (such as a philosophy of models, theories, or experimentation) but rather indicates that they must be taken at face value.

Behind this debate lies the question about the epistemic power of computer simulations. The rationale used by Frigg and Reiss shows that computer simulations are not epistemically powerful, and therefore they are not philosophically attractive. Humphreys, on the other hand, shows how there are cases in which computer simulations yield a genuine understanding of the world, and therefore pose novel questions for the philosopher of science. I agree with Humphreys that the arguments in current literature show only half of the picture, the half that sheds light on computer simulations from the perspective of a philosophy of models or a philosophy

of scientific experimentation. And I also agree with him that there is much more philosophical work to be done. For these reasons, this work aims at addressing the epistemic power of computer simulations at face value. Specifically, I develop my defense in two stages: first, by clarifying the notion of ‘computer simulation’ which, to my mind, has been misinterpreted in many texts in the literature; and second, by adopting an epistemic account of scientific explanation and showing how computer simulations can explain simulated phenomena and why this yields understanding of the world. Explaining facilitates the formation of new beliefs about the way the world is. It is a central activity in our cognitive life, and we usually perform it remarkably well. With the introduction of computer simulation into scientific practice, there was a small conceptual hole in scientific explanation that needed to be filled. This work is my attempt to do so. Let me then begin with this introduction, which I designed as a walk-through of my work.

## Scope and outline

Computer simulations have been defined as processes of reckoning models of great complexity. In this context, ‘complexity’ is measured in terms of computational power, mathematical representation, construction methods, and the like. In this vein, the notion of computer simulation embraces a sense in which ‘to simulate’ refers to solving intractable mathematics, as well as the entire process of constructing, using, and justifying the computational system. This is a fairly common view shared among philosophers. For instance, Humphreys indicates that computer simulations are used primarily, although not exclusively, when theoretical models cannot be integrated analytically;<sup>6</sup> Stephan Hartmann takes computer simulations as systems that imitate other (dynamic) systems;<sup>7</sup> Francesco Guala emphasizes the importance of the use of computer simulations in cases when experimentation cannot yield understanding of empirical systems;<sup>8</sup> and Margaret Morrison has made a compelling argument on the similarity between computer simulations and measurement practices, with special emphasis on the computational nature of simulations.<sup>9</sup>

These philosophers are indeed right about the reckoning power of computer simulations. However, an analysis on their epistemic power should not merely focus on the computational capacity as their primary virtue. Rather, computer simulations are remarkably versatile for representing phenomena. Throughout this work I claim that computer simulations represent selected patterns of behavior of a target system. This means that they stand for a host of systems, including empirical, mathematical, logical, and even phenomenological systems. Moreover, since they represent

patterns of behavior, they do not represent one phenomenon, but rather a class of phenomena. For instance, a computer simulation of a two body interacting system can represent two planets, a planet and a satellite, two cars, two billiard balls, a billiard ball and a car, and so on. Via the possibility of changing input variables, initial conditions, boundary conditions, and the like, a computer simulation representing a host of phenomena singles out one particular phenomenon: one specific satellite-planet constellation, two billiard balls with specific masses, etc. This is, to my mind, the central virtue of computer simulations; namely, the capacity to reliably represent patterns in a host of phenomena. I capitalize on this virtue in my analysis of the role of computer simulations as explanatory devices.

I have conceptually divided this dissertation into two parts. The first part, that includes Chapter 1 through Chapter 3, is a systematic analysis on the notion of computer simulation as entrenched by scientists and philosophers, of the novelty of simulations in the philosophical arena, and several ontological and methodological discussions surrounding computer simulations, among other topics. The second part, that includes Chapter 4 through Chapter 6, analyzes the notion of scientific explanation from a metaphysical, epistemic, and semantic point of view; it also addresses my defense of the epistemic power of computer simulations by showing how they explain simulated phenomena. In this vein, I will be answering questions such as “how can scientists explain a simulated phenomenon?,” and “what kind of understanding does such an explanation yield?” I will show that these questions can only be answered by computer simulations, that they can be successfully answered, and that computer simulations genuinely yield understanding of the world.

In the following I briefly address each of the chapters of this dissertation, placing special emphasis on the motivations, results, and interdependence among them.

Chapter 1 begins by providing an introductory framework to the terminology and general notions used throughout this study. Of the large number of topics available in the general philosophy of science, I focus on the notions of *theory*, *model*, *representation*, and *experimentation*. The reason for this stems from the conceptual proximity of computer simulations to these notions. In this vein, and since my interest is only in computer simulations, the aforementioned notions are not discussed in full detail, but rather to an extent that is relevant to my study.

Despite the depreciation in the interest that the notion of *theory* has had in the general philosophy of science in the past years, it still remains a source of rich and interesting discussions. Section 1.2.1 is a historical reconstruction of the origins of the notion of theory and how philosophers have made the shift to the notion of

*model*.

In Section 1.2.2 I address the notion of *model* to a greater extent. I first construct a preliminary distinction that reduces the number of models that need to be addressed. Such distinction is based on their ontology. Briefly, a model can be divided into *material models* and *conceptual* or *scientific models*.<sup>10</sup> The former are conceived as ‘pieces of the world,’ such as a ripple-tank, whereas the latter are representations of phenomena. This study on computer simulations only focuses on *scientific models* since they are the only ones suitable for implementation on a digital computer. Along with analyzing the class of models of interest, I dedicate some time to discuss three types of scientific models, namely, *phenomenological models*, *models of data*, and *theoretical models*. In essence, the first type of models are typically used for mimicking some features of a phenomenon or a theory; the second type are reconstructions of collected data by measurement or observation; the third type of models are the most reliable among scientific practice since they provide the theoretical underpinning of the phenomenon represented.

Of equal importance to the discussion on models is the notion of *representation*, which I also discuss to some extent. Given the large number of theories of representation and the complexity of the topic, I construct the working definition of *goodness of representation* as an umbrella term that abstracts from a theory of representation. Its primary function is as a primitive for the representational relation, and its working definition is:

The *goodness of representation* of a model describes the degree of accuracy to which a model represents its intended target system. The higher the *goodness of representation*, the more accurate the model represents the intended target system. The more accurate the model represents, the more reliable it becomes as a proxy for that target system. The more reliable the model, the better the results it renders about the target system.

This is an important (and harmless) abstraction in my studies. Its importance stems from my assumption that computer simulations genuinely represent the target system without saying anything about this representational relation. It is harmless because there are plenty of theories of representation in current literature that can be accommodated to these purposes, none of which affect the core of my argument.

The second section of this chapter is dedicated to the study of scientific experimentation. The study on experiments proves to be useful for understanding fundamental discussions on the nature of computer simulations. Section 1.3 begins by introducing the notion of *intervention* and *manipulation*, a fundamental concept of the so-called ‘new experimentalism’ used for characterizing experimental practice.

Behind these notions lies the ontological commitment to a physically causal view of the empirical world; that is, experimentation is a causally-related activity whose main purpose is to manipulate our empirical world by intervening in it.

Typically, the literature on computer simulations aims at establishing their epistemic power by comparisons with laboratory experiments. At the end of this section I then show that there are central concepts in the epistemology of experiments that do not apply to the epistemology of computer simulations. Take as an example the notion of *replicability of results*, which in a laboratory experiment stands for obtaining similar results with different instruments, methods, or theories. This concept, central in the epistemology of experimentation, is elusive in computer simulations since it is too expensive (in terms of money, time, man power, etc.) to change the physical architecture of the computer, the simulation models, or even the theories implemented in order to *replicate the same results*. The aim of this section is to discourage these philosophical comparisons and to address the epistemology of computer simulations at face value.

Finally, in Section 1.4 I address the importance of philosophical studies of computer simulations. There is a recent discussion between Frigg and Reiss, on the one hand, and Humphreys on the other about the novelty of computer simulations in the philosophical arena. In a nutshell, Frigg and Reiss argue for a philosophy of models subsuming the study on computer simulations, while Humphreys replies that such a claim represents a unidirectional study that neglects what computer simulations can offer in and by themselves.<sup>11</sup> In light of this debate, I address Humphreys' position and his response to Frigg and Reiss, endorsing in this way the claim that computer simulations are of genuine interest for philosophical inquiry. Accordingly, I discuss some considerations of my own, with special emphasis on the methodological and epistemological novelty of computer simulations.

The overall aim of Chapter 1 is to discuss most of the terminology used in this dissertation creating, in this way, a suitable conceptual framework for the study of computer simulations. It also has the important aim of putting computer simulations at the center of the philosophical enterprise.

Chapter 2 focuses primarily on bringing the philosophy of computer science into the picture. Computer science is a relatively young discipline that can be traced back to the so-called 'software crisis' in 1968.<sup>12</sup> In recent years, however, there has been an increasing interest in philosophically addressing some problems that computer science poses.<sup>13</sup> Although this young branch of specialized philosophy covers numerous topics, in this chapter I focus exclusively on just a few. Of particular

interest are three units of analysis central to the concept of computer software: *specification*, *algorithm*, and *computational process*. The motivation here is that these units of analysis provide better insight into the notion of *computer simulations* as a special case of computer software. It is also important to mention that the results obtained in this chapter prove to be of central importance for the analysis of explanation of computer simulations that begins in Chapter 4.

Let me briefly say something about each unit of analysis. The *specification* is responsible for combining the researchers' knowledge about the target system together with their knowledge about the computer system. This means that all the necessary information needed for constructing the computer software converges in the specification, such as information about the model, discretization methods, implementation of protocols, initial and boundary conditions, and so forth. Now, the construction of the specification depends on several factors. Modern computer science knows that scientific models cannot be directly implemented on a physical computer, but rather they must be transformed into a computer model. Such transformations involve formal methods, such as discretization procedures (e.g., Euler or Runge-Kutta), or the implementation of programming languages (e.g., 64, 128, 512 bits), as well as non-formal knowledge, such as design decisions or information about the computer architecture. All this information is coded into the specification as part of the computer software. Thus understood, the specification is relevant to the study of computer software for *methodological* and *epistemological* reasons: it works as a blueprint for the design, construction, and use of the computer software; it is also the source of all the information that constitutes the computer software, facilitating in this way our cognitive access to it. I discuss the methodological reasons in full extent in Section 3.3, where I address them in the context of computer simulations. As for the epistemic reasons, the idea is simple: the specification is the unit of analysis that contains all the information regarding the target system as well as the computer architecture. In this sense, it is central for cognitively accessing the computer software, that is, for knowing what the system is about, for reprogramming modules, for understanding the limits of the computer software, and so on.

The *algorithm*, on the other hand, is conceived as an abstract and formal syntactic structure that facilitates the implementation of the specification on a digital computer. Due to its abstractness and formality, the algorithm works as a step-wise system, that is, as instructions to be carried out mechanically by the physical computer in order to achieve a result.<sup>14</sup> Finally, the algorithm is cognitively accessible in a similar sense as the specification, and as such I refer to both of them together

as the *simulation model*.

Finally, the *computer process* can be conceptualized as the physical concretization of the algorithm. It is structurally devised to follow the same set of rules that have been built into the computer hardware. Following the standard von Neumann architecture, the microprocessor unit is capable of reckoning because it is constructed using logic gates (e.g., ‘and,’ ‘or,’ and the like). In this sense, a computer process implements an algorithm on a physical computer by following the same step-wise process as prescribed by the algorithm (which, in turn, was prescribed by the specification). So far, I have made use of the notion of ‘implementation’ in a rather loose way. However, in this chapter I discuss it to a greater extent, considering its semantic sense; that is, in the sense that a syntactic structure, such as an algorithm, is interpreted on a semantic domain, that is, a physical computer. Following William Rapaport, then, a computer process is the materialization of an algorithm on a computer by means of semantic implementation.<sup>15</sup>

The results of this chapter can be summed up into two main points: a) The *simulation model* (consisting of a *specification* and an *algorithm*) is the most transparent unit for future cognitive access; in other words, any access that the researcher has to the information coded into a computer software is located there. b) The step-wise mechanical nature of the algorithm and the computer process provide the basis for justifying that the results of a computation are correct according to the specification (this justification is further defended in Section 4.2.1).

Despite the fact that this chapter addresses computer software in general, points a) and b) above are essential for my own conceptualization of computer simulations and for its defense as epistemically powerful. More precisely, they provide the necessary ontological grounds for elaborating on my own working conceptualization of computer simulations, which is the topic of the next chapter.

Chapter 3 has been divided into two parts: the first part includes a study on the notion of *simulation*, which covers a discussion on the differences between *analogical* and *digital* simulations and a survey of the different classes of computer simulations present in current scientific practice (Section 3.2). The second part is a discussion on the methodology of computer simulations and the analysis of one paradigmatic example (Section 3.3). This second part complements the previous chapter since it illustrates how the three units of analysis in computer science apply to computer simulations. It also elaborates more on the class of computer simulation of interest, that is, equation-based computer simulations.

Let me briefly describe each section and what it contains. I begin by discussing

the philosophical differences between *analogical* and *digital simulations* (i.e., *computer simulations*). The most important result of this first part is the notion of digital simulation as elaborated by Zenon Pylyshyn and Russell Trenholme.<sup>16</sup> These authors conceive the notion of computer simulations as a two-stage affair: on the one hand, there are mappings between the structure of the simulation model and the states of the hardware; on the other hand, there is a representation relation between the simulation model and extra computational phenomena, i.e., an empirical target system. Thus understood, computer simulations are ‘worlds of their own’ in the sense that the results obtained by computing are directly related to the simulation model via the first mapping, and only indirectly related to extra computational phenomena via the representation relation held through the simulation model.

The next topic to address is the definition of computer simulations in current philosophical literature. In this sense, I discuss Humphreys’ ‘working definition,’<sup>17</sup> Hartmann’s notion of ‘imitation of a process by another process,’<sup>18</sup> and I finish with a brief discussion on Frigg and Reiss’ general classification of the notion of computer simulation.<sup>19</sup> Here the purpose is to provide an overlook of the different definitions that can be found in current literature on computer simulations. To my mind, a common mistake made by philosophers is to try to conceptualize the universe of computer simulations into one notion. The problem is that the variety of computer simulations found in scientific practice resists any conceptual corset. Instead, I propose my own working conceptualization of computer simulation. This discussion, however, will not appear until Section 3.3.1.1.

Now, the class of computer simulations of interest are equation-based simulations, such as linear equations, differential equations, parametric equations, and the like (Section 3.2.3). The reason for this selection is partly because this class has a larger presence in the natural sciences, and partly because the epistemic project of explaining simulated phenomena is more attainable if I focalize on this class. In Section 3.2.2 I discuss cellular automata, agent-based simulations, and complex system simulations as computer simulations that are *not* within the scope of this study. The central difference is that equation-based simulations implement simulation models that provide a theoretical underpinning of the phenomenon represented, whereas cellular automata, agent-based simulations, and complex systems are rule-based simulations and as such only represent the interaction among agents or automata. This means that the two classes of simulations offer different methodological, ontological, and epistemological features. For instance, an agent-based simulation provides *emerging* properties of a system through the evolution of an initial set of rules, a desired characteristic for biological and sociological systems, but not for physical

systems. An equation-based simulation, on the other hand, consists in reckoning a set of equations that render a solution to the model and therefore there are no *emerging* properties in the sense of rule-based simulations. With these distinctions in mind, then, I provide a working taxonomy of the systems that can be represented by an equation-based simulation: *empirical*, *theoretical*, and *data systems* are possible candidates as target systems. I also identify three different methods for solving equation-based simulations: *analytic methods*, *numerical analysis* and, *stochastic techniques*.

Section 3.3 addresses the methodology behind the design, construction, and programming of a computer simulation, with special attention given to the formal aspects (e.g., mathematical transformations, formal methods), as well as non-formal aspects (e.g., design decisions, ‘know-how’). These philosophical considerations are analyzed in the form of a concrete example. For this I select a two-body interaction system of a planet and a satellite under tidal stress. This simulation exemplifies in a simple and concrete case the main characteristics of the class of computer simulations of interest. The example of a two-body interaction system also illustrates an important, although seemingly trivial, distinction between a *general computer simulation* and a *particular computer simulation*. The former concept refers to those simulations that stand for a class of target systems (e.g., the class of all two-body interactions, or the class of all flow control); whereas the latter refers to those general computer simulations whose *preconditions* have been fulfilled (i.e., what normally is known as ‘initial and boundary conditions’ and includes ‘global variables,’ ‘input values,’ and the like. See Section 2.2). This distinction, although trivial at first glance, proved to be of central importance for the study of the explanation of a simulated phenomenon, as discussed in Chapter 5. Let me present here the motives for the change in the terminology used: the *simulation model* is understood as a formal entity susceptible to formal *verification* and *validation* (see Section 3.3.2). In this sense, it is a pen-and-paper entity and as such must be differentiated from the simulation ready to be ‘run’ on the physical computer. The *general computer simulation*, then, is conceived as the simulation model implemented on the physical computer but whose preconditions have not yet been set. The *particular computer simulation*, as indicated above, is the general computer simulation with the initial and boundary conditions fulfilled. Since these terms gain importance only during my discussion on the explanatory role of computer simulation, for the rest of the dissertation then, and as long as the context is clear, I will continue using more traditional terminology such as ‘computer simulation,’ ‘simulation model,’ and the like.

At this point, and after the long exploration of philosophical notions of computer simulations, I am in the position to suggest my own working conceptualization:

A computer simulation consists of two elements: a *general computer simulation*, which is the implementation of a simulation model on a digital computer ready to be executed. The simulation model represents the general patterns of behavior of a target system, which can be ‘empirical,’ ‘theoretical,’ or ‘data systems.’ A general computer simulation produces results when its initial and boundary conditions are filled out, creating in this way a *particular computer simulation*. A particular computer simulation produces results by reckoning using stepwise ‘analytic methods,’ ‘numerical analysis,’ or ‘stochastic techniques.’

As a final discussion, and also within the methodology of computer simulations, I address three important concepts for the literature, namely, *verification* and *validation*, and *error* (Sections 3.3.2 and 3.3.3 respectively). The importance of verification and validation is that they provide the necessary justification for considering computer simulations as a reliable instrument for scientific practice. The notion of errors is important for understanding the sources of misinterpretation of results of the computation.

Regarding *verification*, the justification comes from the possibility of performing formal tests on the specification and the algorithm, specifically, to prove that the specification fulfills all the requirements of the scientific model. Similarly, it can be proven that the algorithm fulfills the requirements of the specification. The problem with verification is that, in some cases, it is not possible to fully verify a system. The *validation* method complements the verification method insofar as it confirms that the simulation is producing accurate results by matching those results with another set of well confirmed results (e.g., observed data, data from other more reliable simulations, and the like). The study on *errors*, on the other hand, is a brief survey of the literature on the topic. I discuss three sources of common errors in computer simulations: *physical*, *logical*, and *representational errors*. The first type is related to the malfunctioning of the computer microprocessor, computer bus, memory, and, in general, any physical component of the computer. The second type is related to instabilities in the behavior of computer software; it can be found as coding errors, or as part of a faulty compiler or programming language. Finally, *representational errors* can be found at the scientific model or the specification level, and could significantly compromise the interpretation of results. For instance, a grid that is too big for representing precise results leads to bad approximations, mean square errors, and the like. These concepts gain more importance during my discussion on computer simulations as *reliable processes* in Section 4.2.1.

Chapter 3, then, accomplishes two aims: on the one hand, it challenges current established ideas about the nature of computer simulation. For this I rely on the results obtained in Chapter 2 about the nature of computer software, and Section 3.3 about methodological aspects of computer simulations. The working conceptualization as elaborated in Section 3.3.1.1 is my response to this challenge. On the other hand, this chapter introduces and analyzes a concrete example that embodies much of the work done in previous chapters, and works as a case for the studies on explanation in the next chapters.

With Chapter 3 I conclude the construction of a general philosophical conceptualization of computer simulations. As I mentioned earlier, such a conceptualization is just one part of this study. The second part consists in a defense on the epistemic power of computer simulation. Such a defense is canalized via the evaluation of computer simulations as explanatory devices. Naturally, these two parts are not conceptually separate, but rather the latter depends on the former. Let me now give a brief summary on how I make the epistemic power of computer simulations visible.

Chapter 4 begins by imposing further conditions on the class of computer simulations of interest. The central issue arising here is that computer simulations produce data, and therefore an explanation must be about that data. However, since we explain (and simulate) in order to understand something about the world, it is paramount to find the conditions under which the computer simulation also explains something about it. My solution to this problem comes with the notion of *reliable process* which is understood as the conditions under which we are justified in believing that the computer simulation genuinely represents an empirical target system.<sup>20</sup> The aim of this chapter, then, is to justify our belief that the explanation of results of a computer simulation can also be applied to the empirical phenomenon. These conditions are discussed in detail in Section 4.2.1. Briefly, the first condition for a reliable process is that the simulation model is a good representation of the empirical target system (as given by the notion of *goodness of representation* in Section 1.2.2). This first condition is for excluding computer simulations with no representational content, such as heuristic or explorative computer simulations.<sup>21</sup> Second, that the process of computing does not introduce mathematical artifacts or miscalculations of any kind (as discussed in Section 3.3.3). This second condition is used for restraining results of the simulation obtained from a good model that, due to miscalculations or other sources of error, do not represent. Reliability, then, comes to solve the metaphysical problem of being justified in knowing that the results of a computer simulation genuinely represent an empirical phenomenon. For

terminological purposes, I shall refer to the results of a computer simulation that genuinely represent an empirical phenomenon the *simulated phenomenon*.

Now, knowing that a computer simulation genuinely represents an empirical target system does not entail understanding something about that target system. The notion of *understanding* remedies this epistemic deficiency. Now, since I am not interested in defining the notion of ‘understanding’ (which is typically left to the psychologist, the neurobiologist, and the philosopher of mind), I address the issue elliptically by showing how explaining a simulated phenomenon leads to increasing our general understanding of the world. In this sense, Section 4.2.2 addresses the issue of how the notion of *scientific explanation* is as an epistemic term whose goal is to yield *understanding*.

Given that the philosophical universe of scientific explanation is significantly vast, it is necessary to narrow down the number of theories that are the best candidates for computer simulations. Let us note that my interests are in making use of the conceptual framework provided by a well-established theory of scientific explanation, rather than promoting a new type of scientific explanation furnished by computer simulations (I outline my arguments for this claim in Section 4.3.1).

Following Jaegwon Kim, then, philosophical accounts of explanation can be roughly divided into *explanatory externalism* and *explanatory internalism*.<sup>22</sup> Briefly, *explanatory externalism* takes explanation as tracking an objective relation between the explanans and the explanandum. Causal explanatory realism, as Wesley Salmon or James Woodward describe, is a form of this view.<sup>23</sup> *Explanatory internalism*, on the other hand, takes explanation as something that can be accounted for in a purely epistemic way.<sup>24</sup> The Hempelian Deductive Nomological account, Model Explanation, and the Unificationist account are examples of explanatory internalism. I disregard explanatory externalism on the basis that computer simulations are abstract systems where no relevant causal interactions are taking place, and therefore the explanation of a simulated phenomenon cannot be about tracking back the causal chain (see Section 1.3.1 and Section 3.2). The challenge is now to evaluate which philosophical theory of explanatory internalism is most suitable for computer simulations. For this, Kim poses two questions that any theory of scientific explanation must answer, and that I use for organizing my evaluation:

*The Metaphysical Question:* When  $G$  is an explanans for  $E$ , in virtue of what relation between  $g$  and  $e$ , the events represented by  $G$  and  $E$  respectively, is  $G$  an explanans for  $E$ ? What is the objective relation connecting events,  $g$  and  $e$ , that grounds the explanatory relation between their descriptions,  $G$  and  $E$ ? (Kim, 1994, 56)

*The Epistemological Question:* What is it that we know (that is, what exactly is our epistemic gain) when we have an explanation of  $p$ ? (Kim, 1994, 54)

The metaphysical question is interested in making visible the relation between explanans and explanandum that renders the latter as explained. The epistemological question, instead, aims at asking about the kind of epistemic insight that one obtains by carrying out an explanation. In this vein, it is intimately related to the notion of understanding as recently discussed.<sup>25</sup>

I dedicate the rest of this chapter to the evaluation of three explanatory internalist theories of explanation and whether or not they are suitable for computer simulations. There is a fourth explanatory internalist theory, namely, the Unificationist, which is left to the next chapter.

The first account to analyze is the Hempelian *Deductive Nomological Model* (D-N Model) of explanation. From the large number of objections that can be found in the specialized literature, I have highlighted a few that demonstrate this model's unattractiveness for computer simulations. Perhaps the main objection is that the D-N Model does not provide a notion of understanding. Indeed, according to Carl Hempel, *understanding* becomes a pragmatical/psychological issue and, from the empiricist point of view, a negligible problem for the philosophy of science. Nevertheless, I am interested in a theory of explanation that yields understanding of the simulated phenomenon. I also raise three more objections to the D-N Model. First, the distinction between *science* and *mathematics*: Hempel considers these two disciplines to be separate, excluding the latter from the scope of the philosophy of science. However, since computer simulations combine mathematics with traditional science, it is desirable for a theory of explanation to be able to account for both disciplines. Second, the notions of *empirical adequacy* and *empirical content* require the statements in the explanans to be true, i.e., that they correspond to states of affairs in the world; this is too strong a demand for computer simulations, whose explanans involve abstractions, idealizations, and so forth for the representation of an empirical target system. Finally, I object to Hempel's idea that explanation is *time-independent*, that is, that an explanation of a phenomena does not depend on the available knowledge, but is somehow 'eternal.' Given that computer simulations are highly dependent on the available knowledge and technology at any given time in history, it is desirable to have a theory of explanation that accounts for such scientific change.

The second account studied is actually a cluster of theories known as *Mathematical Explanation of Physical Phenomena*. Despite its unpopularity in the philosophy of science, counting on a mathematical explanation of physical phenomena is an

attractive idea due to the abstract and formal nature of computer simulations. In Section 4.3.3 I discuss the most important theories in current literature, including *asymptotic explanations* as elaborated by Robert Batterman, the *mapping account* as defended by Christopher Pincock, as well as its ‘refined’ version, the *inferential account* as elaborated by Otávio Bueno and Mark Colyvan.<sup>26</sup> The problem I see with these theories is that they all demand a mathematical vocabulary as part of their explanatory success, restricting in this way the number of possible simulated phenomena it is feasible to explain. Indeed, if applied to the context of computer simulations, these theories restrict the explanation of simulated phenomena to those that are only effectively explainable by means of a mathematical language.

The third account discussed is also a cluster of theories known as *Model Explanation*. I briefly revisit four different approaches; namely, the *mechanistic model explanation* as elaborated by Carl Craver, the *covering-law model explanation* by Mehmet Elgin and Elliott Sober, the *hypothetico-structural model explanation* by Ernan McMullin, and the *structural model explanation* as elaborated by Alisa Bokulich.<sup>27</sup> I reject all four approaches on different grounds, principally based on the requirements imposed for a successful explanation. For instance, the *hypothetico-structural model explanation* requires a process of *de-idealization* that justifies the model (i.e., ‘adding back in’ features that have been omitted by the process of idealization of the model). Computer simulations cannot be subject to such a *de-idealization* process simply because the complexity of the simulation model prevents any attempt of ‘adding back in’ those features that have been omitted. A similar argument applies to Bokulich’s own view. In her (Bokulich, 2011) paper she takes that a genuine explanation applies counterfactual patterns of dependence, that is, ways to say *precisely* what would have happened if the initial and boundary conditions had been different. Of course no researcher would accept that it is possible to indicate, let alone to indicate *precisely*, which results would have been obtained had the initial and boundary conditions been different. Admittedly, none of the Model Explanation theories have been designed as a general theory of scientific explanation. Instead, they answer to specific purposes or needs of specific disciplines. Elgin and Sober’s work, for instance, is specifically engineered for explanation in evolutionary biology.<sup>28</sup> However, without an evaluation of these theories and how they fail to account for computer simulations, a critic of my work raise questions regarding the possibility of explaining simulated phenomena based on scientific models.

The main aim of Chapter 4, then, is to address theoretical assumptions in the literature on scientific explanation, and how these assumptions affect explanation in computer simulations. To make this last point more visible, I analyze three inter-

nalist explanatory accounts and show how and why they fail to provide a suitable conceptual framework for computer simulations.

Chapter 5 is dedicated to fleshing out the unificationist account of scientific explanation as elaborated by Philip Kitcher. To my mind, this is the most suitable theory of scientific explanation for computer simulations. For instance, the motivational force for the unificationist is that scientific theories unify a multiplicity of phenomena.<sup>29</sup> Likewise, a general computer simulation unifies the multiplicity of simulated phenomena that have been instantiated by different initial and boundary conditions (see Section 3.3.1.1). Another interesting feature is that the unificationist eliminates traditional boundaries between explanation in mathematics and in the physical sciences, a feature that I claimed was attractive for computer simulations.

But perhaps the major virtue of this account is that there are no epistemic sacrifices,<sup>30</sup> i.e., the unificationist does not ‘hide’ information from the explanatory process nor leave mechanisms unspecified. According to Michael Friedman, the literature on scientific explanation can be divided into two groups:<sup>31</sup> a first group concerned with the precision on the nature of the explanation relation (i.e., answering the metaphysical question), and a second group more interested in pinpointing explanation as scientific understanding (i.e., answering the epistemic question). The first group consists in offering a precise proposal about the nature of the explanation relation, but has little to say about how one gains understanding of the world via explanation. Hempel can be included into the first group, as his account shows precisely how to explain and what to expect by explaining, but he dismisses understanding as a psychological feature. The second group, instead, accounts for the notion of understanding as the central aim of explanation leaving vague the nature of the explanation relation. Philosophers like Stephen Toulmin, Michael Scriven, and William Dray<sup>32</sup> are good examples of this second group. The unificationist theory of explanation comes to redeem these shortfalls. Unfortunately, Friedman’s model suffers from several flaws that discourage the possibility of any real use. Philip Kitcher has amended Friedman’s deficiencies, making the unificationist theory one of the most promising of scientific explanation today.

Briefly, Kitcher’s explanatory unification takes explanation as the derivation of descriptions of phenomena by using the same patterns of behavior again and again. By doing so, the number of facts that must be taken as ultimate or brute are reduced, advancing in this way our general understanding of the world. The general idea is simple and appealing: explanation can show how apparently independent and disperse facts can be subsumed under a unified theoretical framework, so our

understanding of those facts, and therefore the world surrounding us, increases.<sup>33</sup>

In Section 5.2.1 I address the metaphysical question. The unificationist conceives a successful explanation as a derivation of a multiplicity of brute or independent phenomena by using as few and as stringent argument patterns as possible, over and over again. Naturally, part of my analysis of the unificationist consists in fleshing out how an actual explanation is carried out. For now, let us just say that to explain is to exhibit patterns<sup>34</sup> of behavior of phenomena that initially appeared to be unrelated or brute by deriving them from the *explanatory store*. Briefly, the explanatory store is the set of derivations that best systematizes our corpus of scientific beliefs. What is our epistemic gain by explaining a phenomenon? The epistemic question is fully addressed in Section 5.2.2 and now can be summarized in the following way: we understand by seeing connections and common patterns of what initially was unrelated and that now unifies with a greater conceptual framework that is our corpus of scientific beliefs. A successful explanation, then, exhibits connections and common patterns of a multiplicity of phenomena that were previously thought to be unrelated but that now can be unified with the totality of our corpus of scientific beliefs. The criterion for unification, as Kitcher takes it, is the best tradeoff between minimizing the number of patterns of derivation employed and maximizing the number of conclusions generated. The fewer and more stringent the patterns used for an explanation, the greater the unification we can achieve.<sup>35</sup> The question for us, then, is how can the unificationist theory be applied to computer simulations?

The first step is to reconstruct the *explanatory store*  $E(K)$  for computer simulations. Despite the fact that this is a fairly straightforward step, there are still specific characteristics of computer simulations that require our attention. For instance, an explanation, in order to be successful, needs to include all the information that is relevant for explaining a phenomenon. Since my aim is to explain a simulated phenomenon, we must consider the model implemented as well as information regarding discretization methods, measurement of truncation and round-off errors, design decisions, and so forth. How can this information, that is directly relevant for the success of an explanation, be included in the explanatory store? I argue that there are two possibilities: either those errors and discretization methods can be measured, anticipated, and described in the form of a sentence, and then used in the derivation of the description of the simulated phenomenon; or they can be added to the *comments* section as a source of further information for the explanation. The *comments* section is one of the four members of the explanatory store and works as a ‘repository’ of information that proved to be relevant for further understanding of the explanation of a phenomenon. I illustrate how a successful explanation in

computer simulations takes place by explaining the spikes formed by a simulation of a satellite under tidal stress (i.e., the example expounded in Section 3.3.1).

There is an interesting consequence that stems from taking computer simulations as reliable processes. As I mentioned in Section 4.2.1, a computer simulation is reliable if it genuinely represents a target system. Now, since its initial and boundary conditions can be fulfilled with different values regardless of considerations about reality, it is conceivable to simulate phenomena that are impossible to find in the physical world. For instance, the example of the satellite under tidal stress can be instantiated with a gravitational constant equal to  $G = 0 \text{ m}^3\text{kg}^{-1}\text{s}^{-2}$ , and despite the fact that such a phenomenon does not exist in physical reality, it is still within the space of solutions of the same general computer simulation that represents a gravitational constant equal to  $G = 6.67384 \times 10^{-11} \text{ m}^3\text{kg}^{-1}\text{s}^{-2}$ . Above and beyond the metaphysical riddles that this consequence poses, I am interested in showing that we are still able to explain such an impossible phenomenon. As I argue in Section 5.3.1, the derivation of this impossible simulated phenomenon can be carried out in the same fashion as with any other simulated phenomena. The problem becomes more pressing, however, when addressing the epistemic question since it must be specified in what sense we gain understanding of an impossible simulated phenomenon.

The way I answer the epistemic question also draws from the unificationist account (Section 5.3.2). By means of explaining independent simulated phenomena, we see connections and common patterns that can be unified within our scientific knowledge. By unifying a simulated phenomenon with our corpus of scientific beliefs, we are expanding our general understanding of the world. Scientists understand the multiplicity of simulated phenomena because they are able to incorporate -to unify- those phenomena into their corpus of beliefs; they can understand further simulated phenomena because they can explain them by means of using the same pattern of derivation again and again. By unifying simulated phenomena with the totality of our corpus of beliefs, we make the world a more transparent and comprehensible place.<sup>36</sup>

Let me now briefly illustrate these ideas with the example of the satellite under tidal stress as discussed in Section 3.3. By means of successfully explaining the spikes in Figure 3.3, the scientists are able to see connections and common patterns of what initially appeared unrelated and now coheres with classical Newtonian mechanics. Understanding the spikes, then, consists in unifying the explained phenomenon with the broader theoretical framework that is our corpus of scientific beliefs. Similarly, for the case of a simulation with a gravitational constant equal to  $G = 0 \text{ m}^3\text{kg}^{-1}\text{s}^{-2}$

there is no energy exchange nor measurable angular momentum, resulting in a satellite that is not moving. We can explain this situation following the same deductive process as the one that explains the spikes: a static satellite is explained by showing that there is no exchange between spin and orbital angular momentum around closest approach, and therefore there are no formations of spikes. Moreover, we are perfectly capable of understanding this case in the same way: the explanation exhibits the patterns that can be unified with classical Newtonian mechanics. Explaining what *is not* the case is as epistemically powerful as explaining what *is* the case, for in both situations we are in the position to reinforce our system of beliefs about classical Newtonian mechanics. For these cases, computer simulations prove to be an essential method since simulating ‘impossible phenomena’ is a fairly simple matter. Moreover, by successfully explaining simulated phenomena for which we would otherwise have no explanation, we are systematizing our system of beliefs in two ways: we are in the position to understand not only the deep structure of the world, but also to understand how the world could have been had the circumstances been different.

Explanation in computer simulations then, proves to be a central epistemic activity for the practice of science. In fact, we are expanding the possibilities of understanding our physical world by expanding the domain of the explicable. There is little doubt that this is a distinctive mark of an epistemologically powerful method.

Finally, Chapter 6 concludes with some of the shortcomings of this work along with two sections about future challenges for the philosophy of computer simulations. Section 6.2.1 addresses some criticisms of the unificationist account of explanation and how they can affect my study on explanation of simulated phenomena. I briefly outline an answer to each criticism. In Section 6.2.2, I urge for continuing the analysis of the epistemic power of computer simulations by examining the applicability of diverse terms in the philosophy of science to computer simulations, such as *prediction*, *evidence*, or *confirmation*. The underlying motivation is to maintain some distance from comparisons with laboratory experiments.

As I mentioned at the beginning of this introduction, the aim of this work is to defend the epistemic power of computer simulations. I try to achieve this aim by analyzing how they work as explanatory devices. This introduction is only meant to be a general map for the following chapters, where I address each topic thoroughly.

## Notes

<sup>1</sup>As William Aspray explains, von Neumann was building a much more powerful computer than the ENIAC at the Institute for Advanced Studies. Historically speaking, the ENIAC was used by the scientists at Los Alamos, but the IAS-class computers were the first stored-program computer for scientific purposes (cf. Aspray, 1990b, 48).

<sup>2</sup>Cf. (Aspray, 1990b, 155).

<sup>3</sup>Cf. (Aspray, 1990b, 159).

<sup>4</sup>See (Rohrlich, 1990).

<sup>5</sup>See (Frigg and Reiss, 2009) and (Humphreys, 2009).

<sup>6</sup>(Humphreys, 1991).

<sup>7</sup>(Hartmann, 1996).

<sup>8</sup>(Guala, 2002)

<sup>9</sup>(Morrison, 2009)

<sup>10</sup>Strictly speaking, *material models* and *conceptual models* classify as *scientific models*, where the latter is understood as the totality of models that the scientific community implements in their practice. I use the notion of *scientific model* only as a familiar category, for in this way I avoid the introduction of new terminology.

<sup>11</sup>See (Frigg and Reiss, 2009) and (Humphreys, 2009).

<sup>12</sup>For instance, (MacKenzie, 2001).

<sup>13</sup>For instance, (Turner and Eden, 2011; Rapaport, 2005b).

<sup>14</sup>Cf. (Chabert, 1994, 1).

<sup>15</sup>See (Rapaport, 2005a).

<sup>16</sup>See (Pylyshyn, 1984) and (Trenholme, 1994).

<sup>17</sup>(Humphreys, 1991).

<sup>18</sup>(Hartmann, 1996).

<sup>19</sup>(Frigg and Reiss, 2009).

<sup>20</sup>See (Goldman, 1979).

<sup>21</sup> This is the core idea in (García and Velasco, 2013).

<sup>22</sup>See (Kim, 1994).

<sup>23</sup>See (Salmon, 1984) and (Woodward, 2003).

<sup>24</sup>Cf. (Kim, 1994, 273).

<sup>25</sup>In this work I do not make differences between ‘an explanation *yields* understanding of a phenomenon’ and ‘we *gain* understanding by explaining a phenomenon.’ Both sentences refer to the same epistemic product of an agent gaining insight into the way the world is.

<sup>26</sup>See (Batterman, 2002), (Pincock, 2010), and (Bueno and Colyvan, 2006) respectively.

<sup>27</sup>See (Craver, 2006), (Elgin and Sober, 2002), (McMullin, 1978), and (Bokulich, 2011) respectively.

<sup>28</sup>(Elgin and Sober, 2002).

<sup>29</sup>As stated, the idea of theories unifying phenomena is not new and can be traced back to the early ages of the positivist. To avoid confusion, in Section 5.2 I make explicit the sense in which I am using the notion of ‘unification.’

<sup>30</sup>This term is coined by Morrison (Morrison, 2000, 3).

<sup>31</sup>Cf. (Friedman, 1974, 6).

<sup>32</sup>See (Toulmin, 1963), (Scriven, 1970), and (Dray, 1964) respectively.

<sup>33</sup>For further reading, see (Humphreys, 1993).

<sup>34</sup>Patterns here are understood as descriptions that single out natural kinds, objective causal relationships, objective natural necessities, and similar concepts that philosophers use for accounting for the metaphysics of science. As Kitcher puts it, “they are schemata that correctly capture objective dependencies [of phenomena in the empirical world]” (Kitcher, 1993, 150). See also (Kitcher, 1986, 1994).

<sup>35</sup>Cf. (Kitcher, 1989, 432).

<sup>36</sup>Cf. (Friedman, 1974, 15).

# Chapter 1

## A scientific image

### 1.1 Introduction

The scientific interest in computer simulations stems from many sources: the ubiquity of the models implemented, the high accuracy of their results, and the relatively low use of human and financial resources are among the most valuable advantages. Such advantages come with epistemic benefits that drive scientists to make use of computer simulations rather than setting up an entire experiment. This situation captivated the attention of philosophers, who conceived computer simulations as a new methodology lying somewhere intermediate between models and experimentation.<sup>1</sup> The philosopher was therefore expected to ask questions about the epistemic power of computer simulations in the light of a philosophy of scientific models and laboratory experimentation. In this way, philosophical questions related to computer simulations became naturalized as a sub-topic of more familiar philosophies. Is this the best way to approach the epistemic power of computer simulations? I believe it is not, and it is the core of this first chapter to show why computer simulations slip away from conceptual corsets imposed by more familiar philosophies.

This chapter has two chief aims. On the one hand, it is designed to dissuade us from fully analyzing computer simulations from the viewpoint of the philosophy of models and the philosophy of experimentation. As I will show in Chapter 2, the notion of computer simulation has, indeed, a bit of both, and therefore it is illusory to claim for a complete detachment from these philosophies. However, the first aim of this chapter is to show that computer simulations escape from traditional analysis suggested by these philosophies. I plan to do this by analyzing concepts that are at the core of the epistemology of scientific models and experimentation. On the other hand, and by means of such an analysis, this chapter revisits philosophical

literature on models and laboratory experimentation in search of relevant notions and conceptualizations pertinent to the study of computer simulations. The second aim, therefore, is to operate as a theoretical framework in which to place each unit of analysis used throughout this work. With these two aims in mind, this chapter has been divided into three sections, as described below.

Section 1.2 reviews the notions of *scientific theories* and of *scientific models*, as they are at the core of the philosophy of science. The notion of *scientific theory*, less important to this work, contributes to understanding the origins and advent of *models* into scientific practice. The notion of *scientific model*, on the other hand, is discussed in more detail since computer simulations not only implement models, but also have been interpreted as a special kind of model. The literature on *models*, however, is vast and heterogeneous, making it virtually impossible to address every topic. For this reason, Section 1.2.2 focuses on delineating categories and definitions that are relevant for the forthcoming chapters. In this vein, it is paramount to make a first distinction between *conceptual models*, understood as the class of models that represent a target system (such as *phenomenological models*, *models of data*, and *theoretical models*), and *concrete models*, taken as the class of models that are part of the target system (such as a piece of the system under scrutiny). Equally important is the notion of *representation*, central to the philosophy of scientific models. But again, due to the large number of theories of representation, this section offers the notion of *goodness of representation* as a working definition. This working definition has been specifically constructed to bring forward the central aim behind every theory of representation, namely, that a model can be more (or less) reliable as a surrogate for the target system depending on the degree to which it represents such a target system. This working definition is particularly important since it abstracts the study of computer simulations from the ‘model-empirical world’ relation, focusing instead on the implementation of a conceptual model on the physical computer. Throughout this section, then, I present a world of scientific models, narrowing down the kind of model of interest. This analysis is particularly important for showing why the philosophical analysis of computer simulations cannot fully be captured by a philosophy of models.

Section 1.3 continues with the same rationale as the previous section, that is, to present and discuss philosophical problems that are at the core of the philosophy of experiment. It diverges from the previous section in that it focuses on three notions that show the limits of applying the philosophy of experiments to computer simulations. These three notions are: *replicability of results*, *consistency of data*, and *confirmation via a crucial experiment*. The lesson from this section is that there are

specific problems in scientific experimentation that are inapplicable to computer simulations and yet are still at the core of epistemological questions. Admittedly, those philosophers that prefer comparisons between experimentation and simulation can still find their way around, because the number of topics in the epistemology of experiment outnumber the three aforementioned. However, the aim of the section is to discourage this kind of philosophical practice, and to substitute it with one that addresses the epistemological virtues of computer simulations at face value.

There is one final part, Section 1.4, which addresses a very simple and yet central question, namely, ‘is there anything philosophically novel and interesting about computer simulations?’ As might be expected, I believe that this question has a positive answer. However, the challenge posed by more skeptical philosophers surpasses my preferences and demands for objective reasons. This chapter, therefore, also plays the important role of motivating this entire work.

## 1.2 Theories and models

### 1.2.1 Theories

Scientific theories have been at the heart of modern philosophy of science from its beginning. And although the philosophical interest for understanding the nature of theories has not diminished, there has been a substantial shift towards the study of scientific models as they are more accurate units of analysis of scientific practice.

Examples of scientific theories abound in scientific and non-scientific literature: evolutionary theory in biology, relativity in physics, transition state theory in chemistry, just to mention a few. Now, what exactly a theory encompasses is still the subject of many discussions among scientist, historians, sociologists, and, naturally, philosophers. The problem is that the concept of scientific theory is elusive, making it difficult to pinpoint what exactly it stands for, and what it includes in its scope. Take as a simple problem the temporal validity of scientific theories: Ptolemaic interpretation of the planetary movement was valid even when the Copernican interpretation was already available. This means that the Ptolemaic theory included phenomena that, at a certain moment in history, the Copernican theory did not account for.

My interest here is not about scientific progress; it is not about delimiting a scientific theory from a pseudo-scientific theory; and it is certainly not about a thorough analysis of the nature of scientific theories. Rather, it lies in the historical grounds behind the evolution of the notion of scientific theory. When did it all begin? One is more or less justified in setting the beginnings of philosophy of science as

we know it today with the Vienna Circle. Although the philosophical interest of the members of the circle were diverse, they all shared a fierce opposition to the use of metaphysics for the sciences as well as an overstated confidence in the use of logic for the natural sciences. The rejection of any intromission of metaphysics for conceptualizing the world along with a twentieth century marked by several important achievements in mathematics, logic, and theoretical science, boosted the confidence of the Vienna Circle. The theory of relativity by the great Albert Einstein became the cornerstone for the members of the Vienna Circle. Kurt Gödel and Hans Hahn in logic and mathematics, Richard von Mises in economics and political theory, and Otto Neurath in political economy also played a central role in the construction of a logically structured and metaphysics-free philosophy of science.<sup>2</sup> In the philosophical community, the members of the Vienna Circle later became known as ‘the logical positivists.’

The logical empiricist, a relative and successor of logical positivism, took scientific theories in a language-oriented way.<sup>3</sup> Theories then became linguistic entities in the form of an axiomatic system, related by deductive calculus and consisting of an abstract formalism, axioms of the theory, and correspondence rules that provided the necessary interpretation for theoretical terms.<sup>4</sup>

The search for scientific terms with fixed and precise meanings lead the empiricist to take the language of a theory as divided into two classes: a non-logical class, known as the *observational vocabulary* ( $V_O$ ) which defined the terms in the theory by empirical means (i.e., by means of observation, experimentation, and measurement); and a *theoretical vocabulary* ( $V_T$ ), that is, the class of terms of a scientific theory have not been defined by means of empirical operations.<sup>5</sup> For instance, if a theory includes in its formulation the term ‘thermometer,’ then this term refers to a thing in the world since a thermometer can be touched, manipulated, used for measuring, and so forth. Now, if the theory includes a more abstract term such as ‘density’ and defines it as the division between ‘mass’ and ‘volume,’ then the theory will be meaningful only if these latter two terms are empirically defined. In other words, only if atomic terms used in a theory (such as ‘mass’ and ‘volume’) are empirically defined, can the notion of ‘density,’ and in turn the entire theory, be meaningful. The first of our terms, ‘thermometer,’ belongs to the *observational vocabulary* ( $V_O$ ), whereas the second term ‘density’ belongs to the *theoretical vocabulary* ( $V_T$ ) until it is empirically defined.

The distinction between unobservable and observable, a distinction which grounds the dichotomy of the two vocabularies, presupposes that a term in  $V_T$  can be ‘moved’ into  $V_O$  when given the right conditions. Indeed, what is unobservable at one mo-

ment in time becomes observable at another, as science progresses. The term ‘electron’ in the pre-Thomsonian era belonged to  $V_T$  until Joseph J. Thomson developed his famous experiment.<sup>6</sup> Now, to the empiricist’s mind, it is important to reduce the number of terms in  $V_T$  to make an impact on the scientific interpretation of theories. The question was, therefore, how can a term in  $V_T$  be moved to  $V_O$ ? The answer came with ‘operationalism,’ a theory that defined theoretical terms by means of concrete scientific procedures, such as observation and measurement. For instance, the theoretical term ‘volume’ is defined by measuring the quantity of a liquid in a marked container. Thus understood, operationalism defines a term by relating it to one specific procedure.<sup>7</sup> The problem was that a change of methods entailed a change in the definition of a term. In this sense, there were as many notions of ‘volume’ as scientific procedures available.

After enjoying a certain success, the logical empiricist view of theories came under intense attack. There were problems everywhere, including the unsatisfactory operationist approach. In fact, the root of most criticism was at the heart of the distinction between observational and theoretical vocabularies. Willard Van Orman Quine wrote a fundamental piece of work which criticizes this distinction, along with the traditional distinction between *analytic* and *syntactic*.<sup>8</sup> Similarly, Karl Popper strongly objected to the observational and inductivist based scientific method promoted by the logical empiricist. In its place, he proposed *falsificationism* as the only valid scientific method.<sup>9</sup>

These attacks had a very positive effect for the young philosophy of science, for they virtually forced philosophers to diversify the problems into subtopics, such as theories of scientific progress, the notion of scientific method, as Popper himself defended, or against scientific methods, as Paul Feyerabend did.<sup>10</sup> Research programs, scientific realism, and scientific explanation,<sup>11</sup> are also part of the subjects in the new philosophy of science. It is no exaggeration to say that the logical positivist and his successor, the logical empiricist, set the agenda for the philosophy of science as we know it today.

The semantic view of theories emerged as the only decent rival to the logical empiricist. Theories were no longer understood as linguistic entities but identified with a class of set-theoretic structures (i.e., scientific models). Indeed, the proponent of the semantic view removed the syntactic peculiarities in the formulation of a scientific theory and, instead, focused on its underlying structure. In this vein, Isaac Newton’s own formulation of classical mechanics in the *Principia*, modern Lagrangian formulations, and modern Hamiltonian formulations are all variants of the same underlying theory, that is, of classical mechanics.<sup>12</sup> Theories became,

then, families of set-theoretic models that represented more accurately the complex relationships between theories, data, and phenomena.

Over the years, various versions of the semantic view of theories have been devised, and although they eventually suffered from heavy criticism, modern philosophy of science owes much of the introduction of models to the semantic view of theories.<sup>13</sup>

Now, a pervasive objection to the semantic view was that it made models highly dependent on theories.<sup>14</sup> Indeed, the semantic view demanded that models be true of the theory, that is, that the relation between them was one of deriving the model from the theory. Proponents of an alternative conceptualization claimed that models not only do not depend on theories in the way suggested, but sometimes models do not hold any relationship to theories at all.<sup>15</sup> Newton Da Costa and Steven French conceive this issue in the following way:

according to the Semantic Approach theories are families of mathematical models; if this approach is an adequate representation of scientific practice then any scientific model should feature as a member of such a family; however, there are models which do not so feature, since they are developed independently of theory; hence the Semantic Approach is not an adequate representation of scientific practice. (Da Costa and French, 2000, 120)

Mary Morgan and Margaret Morrison, Mauricio Suarez, and others<sup>16</sup> believe that the philosophical study of models must focus on their design, construction, and function in the context of scientific practice. The basic claim is that models *mediate* between theory and phenomena, and that neither of the relationships are deductive. On the contrary, a model is ‘autonomous’ in the sense that it is not always clear how it relates to higher level theories or to phenomena.

In the remainder of this work I take the idea of ‘model as mediators’ as fundamentally correct. In the following section I analyze in more detail these considerations about scientific models.

## 1.2.2 Models

Nelson Goodman once said that “few terms are used in popular and scientific discourse more promiscuously than ‘model’ ” (Goodman, 1968, 171). According to the author, it is possible to find encapsulated within the term ‘model’ “almost anything from a naked blond to a quadratic equation” (Goodman, 1968, 171). Tibor Müller and Harmund Müller, for instance, provide nineteen enlightening examples of the different uses of the notion of model in the natural sciences.<sup>17</sup> Along with these examples, Müller and Müller also provide a host of different criteria for categorizing scientific models: their ontology, degree of semantic representation, epistemic

virtues, propaedeutic values, and the like. It is not surprising, then, that the notion of model cannot be contained in one conceptual corset.

On ontological grounds, models are designed, analyzed, modified, and applied in a wide range of scientific disciplines, from empirical systems (e.g., Petri dishes modeling the growth of bacteria) to extremely abstract ideas (e.g., the Standard Model in Quantum Mechanics). In terms of semantic representation, philosophers have accommodated diverse notions (from the formal strictness of ‘isomorphism’ to more relaxed forms of ‘similarity’) that relate the model to the represented system. And as for their epistemic and propaedeutic values, it is enough to say that today’s science and technology cannot make any tangible or conceptual progress without the aid of models.

For analytical purposes, I classify the study of scientific models into two super classes: *material models*, that is, models which are physical in a straightforward sense, such as models made of wood, steel, and any other material; and *conceptual models*, such as theoretical models, phenomenological models, and models of data, among other kinds. A taxonomy based on a dualistic ontology is useful at this early stage for filtering out the class of scientific models of interest for this work. Since material models do not qualify as the kind of models implementable on a computer simulation, conceptual models are therefore the class of models of interest here. Indeed, a material models super-class includes models intended to be a ‘piece’ of the world and, as such, a direct replica of their empirical target system. Take as example the use of a beam of light for understanding the nature of light. In such a case, the material model and the target system share obvious properties, such as physical quantities, causal relations, and the like. Hence, the distinction between a model of a beam of light and a beam of light in itself is only pragmatic: there seems to be no real differences underpinning them except the fact that it is easier for the researcher to manipulate the model rather than the real beam of light.<sup>18</sup>

Of course, a material model does not need to be an exact piece of the target system. Consider for instance the use of a ripple tank for understanding the nature of light. The ripple tank is a material model in a straightforward sense; however, since its material differs from the target system, its representational capacities are reduced solely to the wave nature of light. Such a representation stems from the fact that there is a unique way in which waves behave equally in all materials, as specified by Hooke’s law, D’Alembert’s equation, and Maxwell’s equations. Let it be noted, however, that the mere fact of being a material model does not entail complete representation of the target system. In this case, by dint of understanding the behavior of water waves one can understand the nature of light as a wave.

Now, despite the many ways in which material models can be interpreted, they cannot be implemented on a computer simulation. The reason is purely ontological: computer software is an abstract representation of the target system, not a ‘piece’ of it, and therefore it requires the conceptualization of a target system, not a way to ‘replicate’ it. The super-class of models that can be implemented on computers are, therefore, *conceptual models*.

I take *conceptual models* as abstract (and sometimes formal) representations of a target system, and as such, are on the opposite ontological side of material models. Philosophers have conceived this super-class of models as *state spaces*, that is, as “the sets of all possible states of a system or a model” (Weisberg, 2013, 27). More formally, a conceptual model is “a set-theoretical entity which is a certain kind of ordered tuple consisting of a set of objects and relations and operations on these objects” (Suppes, 1960, 290). More up-to-date ontological positions about models take them as being *fictions* or *folk-ontology*.<sup>19</sup> In its simplest form, the fictionalist takes models as being “imaginary systems that would be concrete if they were real” (Weisberg, 2013, 49). The proponent of folk-ontology, however, takes that these imaginary systems are not the models themselves but rather “*aids to thinking about the world*” (Weisberg, 2013, 68. Emphasis original.). Now, the subtleties of the discussion that set these two positions apart are irrelevant for our purposes. The specialized literature has discussed these issues to considerable extent and there is nothing really illuminating to add to the discussion. For the purposes of this work, then, I will take models as an interpreted formalism (i.e., structures that describe state spaces) plus a story (i.e., a fiction that helps scientists to understand the target system).<sup>20</sup>

This super-class of models is widely spread out in scientific practice in the form of *theoretical models*, *phenomenological models*, and *data models*. For consistency with current literature, I will refer to these models as *scientific models*. Now, one could of course argue that material models are also part of scientific practice and, as such, part of a broader conceptualization of the notion of scientific models. However, since material models are not of interest for this work for the reasons just given, there is no conceptual harm in identifying scientific models with conceptual models.

Thus understood, scientific models allow a broad spectrum of target systems because they represent empirical systems as well as non-empirical systems, such as thought experiments, purely mathematical systems, and counterfactual systems (e.g., ‘possible worlds’),<sup>21</sup> all of which are of interest for the study of computer simulations as well.

In the following section I will flesh out the notion of *representation* as used in here, and analyze the aforementioned taxonomy for the notion of scientific model.

### 1.2.2.1 Representation

Along with a theory of models comes a theory of representation. This is a time-honored problem in the philosophy of science which includes, under the same conceptual umbrella, the vast diversity of representational accounts for scientific models. A shortlist includes *representational isomorphism*, *partial isomorphism*, *structural representation*, *homomorphism*, *similarity*, and the *denotation, demonstration and interpretation (DDI)* account of representation as elaborated by R. I. G. Hughes.<sup>22</sup> The general aim of any theory of representation is to theorize the relationship between a model and its target system, including the extent to which the model reliably represents the target system. This last issue is of certain importance for this work and I will say something at the end of this section.

To illuminate the general aims of a theory of representation, let me briefly analyze *structural representation* as elaborated by Chris Swoyer. To his mind, “a structural representation depends on the existence of a common structure between a representation and that which it represents, and it is important because it allows us to reason directly about the representation in order to draw conclusions about the phenomenon that it depicts” (Swoyer, 1991, 449). Following this definition, the Lotka-Volterra predator-prey model, for instance, is a structure that represents another structure (i.e., a dynamic system in evolutionary population biology). The representational relation can be either complete, that is, when both structures share the same features; or only partial, when they only share some features.

Let me present a concrete example. The dynamics of a biological system with two members, predator-prey, evolve according to the following pair of equations:

$$\begin{aligned}\frac{dx}{dt} &= x(\alpha - \beta y) \\ \frac{dy}{dt} &= -y(\gamma - \delta x)\end{aligned}$$

where  $x$  is the number of prey and  $y$  is the number of predators. Thus understood, this model represents a biological, social, and economical system, among others. At its core lies the idea of a model *and* a target system sharing the same structure and having the right representational relation. As for the notion of structure, one could conceive it as  $\Psi = \langle U, O, R \rangle$  where  $U$  is a non-empty set of individuals called the domain or universe of the structure  $\Psi$  (i.e.,  $x =$  the prey, and  $y =$  the predator), a set of operations  $O$  on  $U$  (e.g., growth rate over time as described by

the equations  $\frac{dx}{dt}$  and  $\frac{dy}{dt}$ ), and a non-empty set of relations  $R$  on  $U$  (e.g.,  $\alpha, \beta, \gamma$  and  $\delta$  as the parameters describing the interaction of the two species).<sup>23</sup> Since a model is understood as a state space, then the Lotka-Volterra model can independently vary along the  $x$  axis (i.e., the prey population), the  $y$  axis (i.e., the predator population), and time  $t$ .<sup>24</sup> As for the representational relation, it depends partly on the theory that the philosopher is committed to, and partly on the ontological constraints of the model and what it can offer. In this example, Swoyer is committed to some version of isomorphism<sup>25</sup> since he interprets models as structures.

Undoubtedly, each theory of representation conceptualizes the representational relation differently. Nevertheless, the general motivation is the same for all of them. That is, to justify the use of a model as a *surrogate* or *proxy* for understanding something about the target system. At first sight, this idea is appealing because it facilitates the use of a model *as if* it were the target system. But questions emerge from such considerations. Of particular interest is the fact that the representational relation carries with it semantic and epistemic issues. For instance, scientific models are typically *idealizations* and *abstractions* of the target system. *Idealizations*, for instance, consist of deliberate distortions of properties, relations, entities, etc., of the target system with the purpose of making it more tractable. *Abstraction*, on the other hand, consists in ‘stripping away’ certain properties of the target system that are believed to be irrelevant for studying it. Both, *idealization* as ‘distortion’ and *abstraction* as ‘simplification’ must be taken as epistemic virtues, for they facilitate the use of a model as a proxy of the target system. Unfortunately, *idealization* and *abstraction* are elusive terms and it is not always clear how far one can abstract or idealize before completely misrepresenting the target system.<sup>26</sup> The question is, therefore, how good or reliable is the model with respect to the target system?

Theories of representation, as stated before, are a time-honored philosophical problem. So are theories of scientific models. To simplify matters, and because neither theories of representation nor of models are at the core of this work, I will follow Hans Reichenbach’s maxim: “discrepancies between an idealization and a real system constitute serious objections insofar as they defeat the purpose for which the idealization is used” (Reichenbach, 1938, 6).

With these considerations firmly in mind, we can now draw a general interpretation of model as a proxy for the target system: to make use of a model as a proxy for the target system means to assume that the target system behaves in the way specified by the model. Let me illustrate this point with the example of Lotka-Volterra’s predator-prey model. This model is used as a proxy for an empirical biological system in order to understand (and manipulate, measure, etc.) something about the

assumed predator-prey interaction. In a nutshell, it is used *as if* the target system behaves in the way that is specified by the model.

Thus understood, the predator-prey model could fail to accurately represent some aspects of the biological system in many ways, for instance, the number of abstract terms could make it too unrealistic leading to a misleading representation of the empirical biological system. Making use of a model as a proxy for the target system does not entail that it is true, adequate, nor that it isomorphically corresponds to the target system. For this reason, the researcher must be aware of the representational limits inherited with the use of any scientific model. It is also true, however, that a history of prior success of using the same model (e.g., the history of success and reliability of the Lotka-Volterra model for biological system) compensates (and sometimes even justifies) discrepancies between the model and the target system. In other words, Reichenbach's maxim. Of course, one chief aim for any theory of representation is to reduce as much as possible the 'conceptual distance' between the model and the target system.

The fact that this study is not intended to fully address these issues allows me to condense the results here obtained into a working definition. I will refer to what has been said so far about the representation of a model as the *goodness of representation* of such a model. I define it in the following way:

*Working definition:* the *goodness of representation* of a model describes the degree of accuracy to which a model represents its intended target system. The higher the *goodness of representation*, the more accurate the model represents the intended target system. The more accurate the model represents, the more reliable it becomes as a proxy for that target system. The more reliable the model, the better the results it renders about the target system.<sup>27</sup>

Thus understood, goodness of representation is an umbrella term that covers the entire variety of theories of representation in the philosophical literature without committing to any. The goodness of representation of a model, then, is a measure of the discrepancy in the representational relation between the model and the target system. One can easily state that the goodness of representation of a Newtonian model is higher than a Ptolemaic model of the same target system, that is, the planetary movement. In other words, the goodness of representation measures the accuracy of the variables, relations, properties, etc., used for a model with respect to the target system. The more accurate, the higher the goodness of representation, and therefore the results are more accurate than otherwise.

The working definition aims at capturing the core of every theory of representation, that is, that a model can be more (or less) reliable as a proxy depending on

the degree of accuracy with which it represents the target system. Admittedly, this working definition is a bit clumsy and does not answer many of the central philosophical questions about representation. However, it is useful insofar as it provides the necessary epistemic guarantees that the scientific model implemented as a computer simulation, and therefore the computer simulation itself, represents a target system. It is not within my interests to deepen the discussion of the relation between scientific models and the world, but rather take it as relatively unproblematic and focus on the implementation of a scientific model into the computer simulation. Allow me now to move to a brief discussion on sub-classes of scientific models that are of interest for this study.

### 1.2.2.2 A taxonomy for scientific models

At the beginning of this chapter, Goodman drew our attention to the promiscuous use that literature makes of the notion of model. The multiplicity of models used in scientific practice alone makes it virtually impossible to elaborate a definite taxonomy. However, there is general agreement among scientists about the use of three classes of models: *phenomenological models*, *models of data*, and *theoretical models*. This classification is mostly based on representational grounds. A phenomenological model, for instance, is based on a superficial assessment of the behavior of the target system, whereas a data model is based on the reconstructed data collected from observing or measuring something about the target system. A theoretical model, on the other hand, is based on an assessment of the deep structures and mechanisms underlying the target system.

At this point it is also important to remark that there are no *a priori* evaluations on the epistemic power of a model. In this way, a theoretical model does not need to be epistemically more powerful than a phenomenological model for the sole fact that it represents the underlying structure of the target system. Its epistemic virtues depend not only on its power to represent, but also on its propaedeutic values, capacity to justify inferences, construction assumptions, and so forth.<sup>28</sup> Now, allow me to briefly address this small taxonomy for scientific models.

#### *Phenomenological models*

The purpose of constructing phenomenological models is that sometimes mimicking the features of a theory is easier to handle than the theory itself (yet equally accurate). For instance, fundamental theories such as quantum electrodynamics for condensed matter, or quantum chromodynamics (QCD) for nuclear physics, are not used for several reasons, but a phenomenological model is used instead.<sup>29</sup>

In this vein, a phenomenological model is not completely independent of theory; rather it is inspired by a theory of incorporating some principles and laws associated with such a theory.<sup>30</sup> For instance, the liquid drop model of the atomic nucleus represents the nucleus as a liquid drop having certain properties, such as surface tension and charge. Whereas surface tension can be found in hydrodynamic systems, charge is postulated by electrodynamics. The model, then, is a combination of certain aspects of these theories plus some observed behavior that determines the static and dynamical properties of the nucleus.

The distinctive mark of phenomenological models is that they do not postulate underlying structures or hidden mechanisms, but rather represent observable properties of the target system. There is a principle of *mimicking* guiding phenomenological models: either mimicking features of theories, or mimicking observable properties of target systems. The traditional example of a phenomenological model is the London and London model of superconductivity, which was developed independently of theory in methods and aims. It is interesting to note that Fritz London himself insisted that a phenomenological model must be considered only a *temporary* model until a theoretical model could be elaborated.<sup>31</sup>

### *Models of data*

Models of data share with phenomenological models the fact that they also lack theoretical underpinning, only capturing observable and measurable features of a phenomenon. Despite this similarity, there are also differences that make a model of data worth studying on its own. To begin with, a model of data is characterized by a collection of well-organized data, and therefore its construction is significantly different from a phenomenological model. In particular, they require more statistical and mathematical techniques since the collected data need to be filtered for noise, artifacts, and other sources of error.

An interesting philosophical problem is to decide which data need to be removed and under which criteria. A related problem is deciding which curve function represents all the cleaned data. Is it going to be one curve or several curves? And what data points should be left out of consideration when no curve fits them all? Roman Frigg and Stephan Hartmann explain that the problem of filtering data is usually dealt with within the context of the philosophy of experiment.<sup>32</sup> On the other hand, the issue of fitting the data into a curve is a problem of its own, known as the *curve fitting problem*. Briefly, the curve fitting problem consists in connecting a series of data points by means of a single function. In other words, provided a clean set of data, the question is, what curve fits them all together? The problem

is that the data themselves do not indicate what form the curve must have.<sup>33</sup> The curve fitting problem, then, raises a host of mathematical problems, especially in statistical inference and regression analysis.<sup>34</sup>

Typical examples of data models come from astronomy, where it is common to find collections of large amounts of data obtained from observing and measuring astronomical events. Such data are classified by specific parameters, such as brightness, spectrum, celestial position, time, energy, and the like. The astronomer, then, wants to know what model lurks behind that pile of data. As an example, take the *virtual observatory data model*, a worldwide project where meta-data are used for classification of observatory data, construction of data models, and simulation of new data.<sup>35</sup>

### *Theoretical models*

The traditional exponent of this type of model is James Clerk Maxwell. Maxwell's equations demonstrated that electricity, magnetism, and light are all manifestations of the same phenomenon, namely, the electromagnetic field. Subsequently, all the classic laws and equations used in these disciplines became simplified cases of Maxwell's equations. Models of the dynamics of fluids, such as the Navier-Stokes' equations; statistical models of the distribution of population; logical models that represent the basic functioning of a vending machine system; or mental models of rational decision, are more examples of the forms that a theoretical model can take.

What is the underlying feature that ties all these examples together? On the one hand, they all stand for the underlying structures or mechanisms of their target systems; on the other, they embody the knowledge that comes from a well-established theory. In this vein, a theoretical model significantly differs from a phenomenological model and a model of data in that it provides different ways of understanding the target system. To illustrate this point, consider a theoretical model of the dynamics of fluids, such as Navier-Stokes, and a phenomenological model of the same system. The former model theoretically underpins the empirical system, providing better support for explaining the target system, predicting future states, and better evaluating evidence; a purely phenomenological model would only superficially represent the empirical system, and its capacity for explaining and predicting could be undermined by its lack of epistemic reliability. To simplify matters, then, I use the term *theoretical model* as an umbrella term that covers models which stand for the deep mechanisms that underlie their target systems, and which are partially based (if not entirely) on well-established theories.

The precedent overview on the philosophy of models is not meant to be exhaustive, but to place each concept discussed here into the more general conceptual framework that is this first chapter. I now turn to a discussion on experiments, as they are equally important for our study on computer simulations.

### 1.3 Experimentation

Scientific experimentation is a multidimensional activity that philosophers have been studying for some years now.<sup>36</sup> This multidimensionality stems not only from the complexity surrounding empirical phenomena, but also from the practice of experimentation in itself, which is intricate in design and complex in structure. Hence, when philosophers talk about *experiment*, they refer to a host of interwoven topics, methodologies, and ways of practicing science. Among these topics we can include causality, laboratory practice, scientific instruments, replicability of practices, observation, and measurement, among others.

Experiments play different roles in science; they are used for observing entities, detecting new entities, measuring different values, testing theories, and so forth. Historically speaking, Hanson was the first philosopher to introduce the idea that an experiment is not an ‘independent procedure,’ but rather that it is theory-loaded, that is, there is a background theory supporting the findings of an experiment.<sup>37</sup> Over time this view lost its original momentum, and philosophers adopted a more moderate view, one that endorses the idea that in an important number of cases, theory-free experiments are possible and do occur in scientific practice. Such a moderate view admits the possibility of performing ‘exploratory’ experiments, namely, experiments that require some preconceived ideas about the empirical world, laboratory equipment, and the like, but not necessarily a well-developed theory about the phenomena under scrutiny. In this vein, experimentation exhibits a helical rather than linear structure. It follows, then, that the agreement between theory and experiment rests on a reciprocal adaptation among hypotheses, instruments, methods of data processing, background knowledge, and so forth.

Computer simulations, on the other hand, have been heavily discussed under the considerations of scientific experimentation. Due to their great power of computation, some philosophers have asked whether computer simulations can work as an experiment, whereas others, maybe in a clearly bolder attitude, foresee computer simulations as a replacement *tout court* of scientific experimentation. In any case, a definite answer is not easy to get. However, comparisons with scientific experimentation only motivate the interest of understanding computer simulations as central

tools in scientific practice.

The following sections overview these ideas in the light of the ‘new experimentalism,’ i.e., the set of ideas that place experiments at the center of the scientific practice. I also briefly discuss on the epistemology of experimentation as elaborated by Alan Franklin, understood as the set of strategies that provide reasonable belief in the validity of an experimental result.<sup>38</sup> The final section addresses the notion of scientific instruments and discusses possible interpretations of computer simulations falling under that category. Despite the fact that this chapter is dedicated to discussing some of the most relevant aspects of the epistemology of experimentation, it also includes references to computer simulations as a means of establishing a basis for comparison with experiments.

### 1.3.1 The epistemology of experimentation

The first experiment ever to be recorded was Aristotle’s observation of the embryology of the chick.<sup>39</sup> The methodology used resembles modern scientific method: Aristotle observed and documented every stage of the growth of the chick. Despite Aristotle’s work, for a long time philosophers removed the study of experiment from the philosophical inquiry. It was not until the arrival of logical empiricism that experiments began to have a more relevant role in philosophy of science. However, to the logical empiricist, experimentation represented not so much a philosophical problem in itself, as a subsidiary methodology for understanding theory. The only importance of experimentation was for the confirmation and refutation of a theory; it had no value as a method of philosophical inquiry in itself. As argued in Section 1.2.1, logical empiricism experienced a series of objections and attacks that make it unappealing in several ways. One such objection virtually forced the empiricist to reconsider the role of explanation in the philosophical agenda: the problem was that the evidence available at a given time might be insufficient for confirming or refuting a theory. *In this vein, interpreting experimentation as merely a subsidiary methodology for theory is misleading and instead, it must be considered in its own right.* This objection was later known as ‘the underdetermination of theory by evidence,’ first discussed by Pierre Duhem and W. V. O. Quine.<sup>40</sup> Philosophers then began to acknowledge the importance of experimentation in scientific inquiry. Robert Ackerman and Deborah Mayo refer to this new era as ‘new experimentalism,’ which is typically understood as the set of ideas that put the experiment at the center of scientific practice.<sup>41</sup>

New experimentalism, thus, supplements the traditional, theory-based view of logical empiricism with a more experimental-based view of scientific practice. And

although its members are interested in different problems, they all share the core idea that scientific experimentation constitutes a central problem in the philosophy of science that must be discussed in its own right.

Weber proposes five general trends that characterize new experimentalism. Here I present a brief summary:

First, there is the idea that experimentation serves purposes in scientific research other than testing high-level theories [...] A second, related idea of the new experimentalism is that there is experimental knowledge that has its own internal stability which may even survive major transitions in the high-level theories [...] Third, New Experimentalists have given new life to the distinction between observation and experimentation [...] Fourth, some New Experimentalists have challenged the idea that theories are somehow matched to nature on the basis of experimental results [...] Fifth, some philosophers have stressed that much more attention must be given to the details of experimental practice in order to answer some of the old questions concerning scientific inference and theory testing [...] (Weber, 2004, 128–130)

The switch from traditional ‘top-down’ schemata (i.e., from theory to reality) into a ‘bottom-up’ conceptualization is the distinctive mark of new experimentalism. The notion of *natural phenomenon* also went through some transformations. Under the new experimentalist point of view, a phenomenon can be simply and directly observable, as the many macroscopic things and events that surround us in the world, or invisible to plain sight and more complex to observe, such as microbes or the microscopic quantum world.

Perhaps the most interesting contribution from new experimentalism to our study of experiment is the idea of *manipulation of* or *intervention into* the empirical world. The idea is simple and appealing: the scientists manipulate the experiment *as if* they were manipulating the empirical world, therefore, whatever the epistemic gain from the former is extrapolated to the latter.<sup>42</sup> Complementary to the notion of manipulation or intervention is the notion of *causality*. In this vein, when scientists want to understand something about a natural phenomenon, they ‘causally’ manipulate a suitable experiment. Consider for instance Röntgen’s discovery of the X rays: Wilhelm Röntgen needed to experiment with high voltage electricity discharged through air in a partially evacuated glass tube. Without ‘causally’ interacting with this set-up, the discovery would not have taken place.<sup>43</sup>

The problem is that *causality* is an elusive concept in the philosophy of science and logic. Philosophers engage in different and sometimes completely disjointed notions of causality. Nevertheless, since the notion of experiment of interest is experimental physics, there are a host of theories of causality that can be ruled out: for instance, *counterfactual theories*, *probabilistic causation*, and *causal calculus*.<sup>44</sup>

More specifically, *causality* here is a synonym for ‘physical causality,’ that is, the kind of interaction the empirical world is expected to have. In order to keep this section short and simple, I will only address what I believe to be the most suitable theory of causality for our notion of experiment, Phil Dowe’s conserved quantity theory.

Dowe deals with what he calls ‘an empirical analysis’ of causation, which presupposes looking at scientific practice itself and understanding the interpretation that concept adopts within that specific domain. More specifically, Dowe proposes taking physical causality as “the possession of a conserved quantity, [...] that makes a process a causal process” (Dowe, 2000, 89). Borrowing Salmon’s transmission of mark,<sup>45</sup> Dowe grounds his account in the physical quantities found in the ontology of science, such as momentum or energy. Now, a causal process is interpreted as the exchange or transmission of conserved quantities from one object to another.<sup>46</sup> Take for instance the measurement of ionizing radiation with a Geiger counter, which consists in detecting the nuclear radiation transmitted from certain types of particles (alpha or beta particles). The Geiger counter has a causal interaction with the alpha particles because those particles emit radiation to the sensor of the counter.

As I mentioned before, causality is conceptually complementary to the notions of *intervention* or *manipulation* in the sense that the latter are possible because there is a suitable notion of causality in place. The basic idea is appealing and sound: scientific practice intervenes in the empirical world by physically manipulating the causes involved in the experimental setup. In this vein, there are a number of scientific activities that are justifiably called ‘experiments.’ For instance, *measuring*, as with the case of the Geiger counter, or *detecting new entities*, as is the case of the discovery of X-Rays. *Observation* is another way of experimenting. A good example is the observation of the movement of pollen grains suspended on the surface of stagnant water, which is basically Brownian motion. In these latter case there is also an exchange of energy and momentum, since observing either with a naked eye or through an instrument requires an exchange of energy and momentum, as described above.<sup>47</sup> Aristotle’s observation of the embryology of the chick might also be considered an experiment in the sense that it was necessary for him to crack the shell of the egg, move the embryo, and perform similar activities.

It must be noted that the class of experiments that I have been discussing so far have certain characteristics which I expect to maintain; namely, they are carried out in a controlled environment, such as a laboratory (as opposed to field experiments); they are focused on physical causality (as opposed to psychological experiments); and they are fundamentally based on the physical sciences (as opposed to biological

experiments). The reason for selecting such a group of experiments is directly related to the class of computer simulations of interest, as discussed in Section 3.2.3. Thus interpreted, the notion of experiment raises philosophical questions about their epistemic virtues interpreted as providing a justification for the belief in the validity of experimental results. Typically, this justification arises from the set of accepted strategies that increase the reliability of the results of an experiment, such as the possibility to measure and observe quantities of interest with a sufficient degree of accuracy and precision.<sup>48</sup>

Allan Franklin is certainly a major name in the philosophy of experimentation. In *Experiment, Right or Wrong*, (Franklin, 1990) he introduces a set of strategies that, to his mind, provide reasons for believing in the validity of an experimental result.<sup>49</sup> In this section I revisit some of these strategies with the explicit aim of comparing them with computer simulations. The purpose of this comparison is to show to what extent experiments and computer simulations differ from (or are similar to) each other.

Experimental philosophers stress the importance of ‘controlling factors’ that could affect the results of the experimentation. Those factors can be either *internal* to the experiment, for instance the misuse of an instrument, or a poorly constructed set up; or *external*, for instance if the experiment is sensitive to specific vibrations or, as occurs in high energy experiments, there is a need to shield very sensitive energy detectors from cosmic rays, naturally raining down on the earth from outer space. Unfortunately, isolating the experiment from internal and external factors cannot be taken as complete guarantee of success.<sup>50</sup> The scientist, therefore, needs a more general method for minimizing the possibilities of something going unnoticedly wrong, and for providing consistent results. One particularly interesting method is via *replicating the results* by means of running similar experiments on different material support. For instance, when measuring the atmospheric pressure, the scientist can use a mercury barometer, a water-based barometer, or a digital barometer depending on the degree of precision and accuracy desired. The general point is that the possibility of replicating the same results by different experimental setups speaks in favor of the process of experimenting, and also speaks in favor of the precision of instruments, the team of scientists, and the general reliability of the results obtained.

Replicability is a more elusive concept in the field of computer simulations. For instance, it cannot be fully grounded on the architecture of the digital computer, for all of them (be it a 64, 128 bit architecture; a Mac or a PC; a Workstation or a

Server, etc.) provide very similar material support, defeating the original purpose of corroborating whether the same entities, hypothesis, etc., could be confirmed by different setups. The alternative is to either change the simulation model (i.e., representing the same target system by means of different simulation models) or to rewrite the simulation in a different computer language. Such possibilities, however, might not be so appealing when considering the cost-effectiveness ratio of implementing a new scientific model or rewriting an existing scientific model. Admittedly, for very specific situations (e.g., due to the sensitivity of the problem at hand, or the cost of making a mistake, etc.) it could be the case that different groups replicate results of a simulation via implementing different models in different programming languages. It is because of these situations that one is justified in only partially admitting that the results of a simulation could not be replicated.

The issue of replicability of results has some kinship with the problem of *consistency of results*. The basic problem here is that an instrument used in the experimental setup might produce *artifacts* (i.e., systemic errors) that lead to erroneous results. Consistency here is taken as results becoming more and more concentrated near the true value of the parameters being experimented or simulated. In other words, results are consistent when they fall within a given distribution that tends to converge on the true value of the parameters of the target system. Consistency, then, always speaks in favor of the correctness of such results and against the presence of artifacts.<sup>51</sup> One way to ensure consistency is by deriving, or relating somehow, such results with a theory. Ian Hacking, for instance, showed how observations with the microscope radically changed in 1873 after Ernst Karl Abbe's study on diffraction, ensuring the correctness of observable results.<sup>52</sup> Another colorful example of this is the discovery of cosmic microwave background radiation by Arnold Penzias and Robert Wilson.<sup>53</sup> The story goes that after suppressing all assumed sources of distortion, they kept measuring an unexplainable  $4^{\circ}$  K. Penzias and Wilson kept looking for sources of artifacts of their measurements, until they heard that a group at MIT released a paper on theoretical physics arguing for the possibility of finding radiation left over from the Big Bang.<sup>54</sup> This is the story of how a theoretical finding underpins an experimental finding, eliminating all sources of doubt about the results. The lesson here is that knowing about the theory that backs up the instrument and guides the experiment is an extra layer of reliability for guaranteeing the consistency of results. Are these the only sources of reliability? Absolutely not. Authority has been acknowledged as a reliable source for consistency of results, as has the acceptance by a community, or the settlement of a practice.

The case for computer simulations moves away from experiments since consis-

tency here is achieved by techniques that do not have counterparts in the experimental domain. Consistency of results of a simulation, then, is achieved either *theoretically*, that is, by knowing the distribution function and the space of possible results of the simulation model; by *formal proof*, that is, by showing that the results are formally derivable from the simulation model (so-called *verification* methods); or by *validating* the results against other reliable results, such as comparing data from empirical systems, from other simulations, etc. The first two techniques have no counterpart in experimental practice simply because they are not needed for granting consistency of results. One could, however, make the case that validation is an old, well-known technique also used in the experimental domain. Although this is only partially true, for the large amount of results of a computer simulation can only be dealt with in a specific way (e.g., by another computer model), it is part of the claimed symmetry between computer simulations and experiments. The important aspect here is that despite these symmetries, computer simulations differ from experiments in interesting ways. Discussions on verification and validation can be found in Section 3.3.2.

Another interesting problem that puts experiments and computer simulation on opposite ends is the so-called *crucial experiment*. Although nowadays this problem has been discredited, it was originally taken as a kind of experiment that had the capacity to resolve two or more competing theories. This means that, given two competing theories, the experiment is the ultimate resource for deciding on the correctness of one of them. As Imre Lakatos puts it: “the term ‘crucial experiment’ indicates that from some experiments we learn more than from others” (Lakatos, 1974, 309). Despite its unpopularity among philosophers of science, there are registered cases in which experiments resolve competing theories. Examples of this are found in works on nonconservation of parity, and in the violation of mirror symmetry in weak interactions.<sup>55</sup> A similar case is that of experiments that are highly corroborative, that is, experiments that only corroborate the central idea of a particular theory, without necessarily deciding over competing theories. Franklin has two examples that back up this scenario, those of “Davidson and Germer, (1927) and [...] G. P. Thomson (1927a; 1927b; 1928a; 1928b), which established the wave nature of the electron, corroborating de Broglie’s view” (Franklin, 1981, 368).

Although these cases are almost non-existent in science, they are an interesting methodological aspect of computer simulation. The question ‘can a computer simulation help us decide among competing theories?’ is more difficult to answer than it might seem. Consider, for instance, the computer simulation that determined the truth of the ‘four-color theorem.’<sup>56</sup> Taken this way, a computer simulation does work

as a *crucial experiment* since it helped decide over the validity of the hypothesis of the four-color theorem.

Such consideration might be taken as a distortion of the original problem, for a *crucial experiment* essentially points to a phenomenon that is located beyond the theory and, therefore, independent of it. For example, the nature of light is independent from the theory considering it to be a wave or a particle; in order to relate it with the theory, it was necessary to carry out an experiment that confirmed the wave theory, while ruling out other theories.<sup>57</sup> The four-color theorem, one could argue, needed to be programmed and solved by the computer in order to have a result. Since solving a model by a computer process is derivation-related, the results are not independent in the sense just given. This argument is sound insofar as one takes that there are certain degree of dependence between the model and the results propitiated by the process of computing. This is, to my mind, the correct view on the matter. Now, in this context, is it possible to guarantee *complete* independence between results and a model, as a crucial experiment requires? The answer is a straight no because computer simulations do not fulfill a basic assumption of crucial experiments, namely, their total independence from theory. Allow me to illustrate this point by considering a simulation that could decide over the nature of the electron. In order for such a simulation to reproduce the wave nature of the electron, it was first necessary to implement a model that represented such a nature. It follows that in order to consider what the nature of the electron is, it was necessary first to decide (and implement) a theory that speaks in favor of such a given nature. There is a certain circularity in the process of deciding over competing theories with computer simulations. The problem is that the results of a computer simulation are a byproduct of computing the simulation model, and therefore limited to that domain. In this sense, the results obtained from a computer simulation do not point beyond the model towards an extra-computational phenomenon, but rather to the endogenous simulation model that produced them.<sup>58</sup>

Science teaches us, however, that deciding between competing theories is not always a direct matter, but rather oblique. Robert A. Millikan's work on proving the wave nature of light led him to construct an experiment (his famous 'machine shop in vacuo'<sup>59</sup>) which could, allegedly, disprove Einstein's theory. A computer simulation can work precisely at this same level of analysis; that is, by addressing the problem of crucial experiments (and its relatives *confirmation*, and *refutation*) over competing theories obliquely. For instance, Olaf Cirpka et al. implemented simulation methods of uncertainty propagation and sampling strategy that allowed the authors to discriminate between cases of success and cases of failure for 'funnel-

and-gate systems.’<sup>60</sup>

A similar argument applies to the objection that computer simulations are not experiments because they will not warrant observational statements. Such an objection does not carefully distinguish between ‘direct’ observations and ‘indirect’ or ‘oblique’ observations. As I argued before, computer simulations are able to simulate phenomena that are otherwise unobservable by traditional means (e.g., most of the research on quantum mechanics depends today on computer simulations). Such observational statements are warranted by the correct implementation of scientific models and the correct computation of such models. The lesson here is also sober because, although the notion of crucial experiment does not apply directly to computer simulations as it does to scientific experimentation, an oblique approach still seems possible.

This section has addressed a few core problems in the epistemology of experiment and how they can be understood in the context of computer simulations. The general lesson here is that there is a host of specific issues in the epistemology of experimentation that might not be related to computer simulations. Despite the fact that it is possible to spot some symmetries between computer simulations and experiments, the general persuasion is that as philosophers we must address the epistemology of computer simulation at face value. The next section addresses the question regarding what is novel and philosophically relevant about computer simulations.

## 1.4 Hot new stew

In 2009, Frigg and Reiss wrote a stimulating paper questioning the philosophical importance of computer simulations. The core objection is that philosophers working on computer simulations tend to overemphasize the role that simulations play in scientific practice, deducing in this way unwarranted conclusions. More specifically, according to the authors, it has become a common practice among philosophers of simulation to claim for an entirely new epistemology, a revised ontology, or even novel semantics for the philosophy of science nourished by computer simulations. To their mind, however, computer simulations hardly call into question our basic philosophical principles of understanding scientific practice and of conceptualizing reality.<sup>61</sup> Now, I share Frigg and Reiss’ puzzlement on this matter. It is hard to sustain that a new scientific method, however powerful and novel it might be, could entirely imperil well developed and largely established philosophies. It is still

an open question, however, whether Frigg and Reiss have correctly interpreted the authors that they criticize.

Now, Frigg and Reiss' aim goes well beyond setting limits to ungrounded philosophical claims and asserts that studies on computer simulations do not raise any new philosophical questions. Indeed, to their mind philosophical studies on computer simulations can be assimilated within existing philosophies, such as a philosophy of models or a philosophy of experimentation. It is important to indicate that they are not objecting to the novelty of computer simulations in the scientific practice, nor their importance in the advancement of science, but rather that simulations raise few, if any, new philosophical question.<sup>62</sup> As the authors like to put it, "we see this literature as contributing to existing debates about, among others, scientific modelling, idealisation or external validity, rather than as exploring completely new and uncharted territory" (Frigg and Reiss, 2009, 595).

At this point it is important to ask whether there is an alternative to such a divided scenario. One way of approaching it is by rejecting the alleged claim that computer simulations represent a radical new way of doing philosophy of science, while accepting that computer simulations are philosophically interesting methods and do raise new questions for the philosophy of science. Such is the viewpoint that this work takes. But before addressing in more detail my own conceptualization of computer simulations,<sup>63</sup> allow me first to discuss Frigg and Reiss' work and its impact in the philosophical community.

The article virtually divided opinions, forcing philosophers to define their position on the matter. Paul Humphreys was the only philosopher that directly engaged into the discussion and gave an answer to the authors. According to Humphreys then, Frigg and Reiss' assertions obscure the challenges that computer simulations pose for the philosophy of science. The core of Humphreys' reply is to recognize that the question about the novelty of computer simulations has two sides: one which focuses on how traditional philosophy illuminates the study of computer simulations (e.g., through a philosophy of models, or a philosophy of experiment), and the other which focuses exclusively on aspects of computer simulations that represent, in and by themselves, genuine challenges for the philosophy of science. It is this second way of looking at the problem that grants philosophical importance to computer simulations.

One of Humphreys' main claims is that computer simulations spell out inferential relations and solve otherwise intractable models, amplifying our cognitive abilities, whether logico-inferential or mathematical. Humphreys called such an amplification the *anthropocentric predicament* as a way to illustrate current trends in the science

of computer simulations moving humans away from the center of the epistemic activity.<sup>64</sup> According to him, a brief overview on the philosophy of science shows that humans have always been at the center of knowledge. This conclusion includes the period of the logical and empirical positivism, where psychologism was removed from the philosophical arena. As Humphreys points out, “the overwhelming majority of the literature in the logical empiricist tradition took the human senses as the ultimate authority” (Humphreys, 2009, 616). A similar conclusion follows from the analysis of alternatives to the empiricist, such as Quine’s and Kuhn’s epistemologies. The central point here is, according to Humphreys, that there is an empiricist component in the philosophy of science that has prevented a complete separation between humans and their capacity to evaluate scientific activity. The anthropocentric predicament, then, comes to highlight precisely this separation: it is the claim that humans have lost their privileged cognitive position in the practice of science to computer simulations.<sup>65</sup> This claim finally gets its support from the view that scientific practice only progresses because new methods are available for handling large amounts of information. Handling information, according to Humphreys, is the key for the progress of science today, which can only be attainable if humans are moved from the center of the epistemic activity.<sup>66</sup>

The anthropocentric predicament, as philosophically relevant as it is in itself, also brings about four extra themes unanalyzed by the philosophy of science; namely, ‘epistemic opacity,’ ‘computational representation and applications,’ ‘the temporal dynamics of simulations,’ and ‘the in practice/in principle’ distinction.<sup>67</sup> All four are novel philosophical activities supplemented by computer simulations; all four have no answer in traditional accounts of models or experimentation; and all four represent a challenge for the philosophy of science.

As I have mentioned before, I do believe that computer simulation raise novel questions for the philosophy of science. One simple way of supporting this claim is by showing what distinguishes them from other units of analysis in the philosophy of science. Humphreys began this list of peculiarities by pointing out the five issues above. Allow me to briefly discuss four more issues that have also escaped Humphreys’ list.

### 1.4.1 Recasting models

The first overlooked feature of computer simulations is their capacity to *recast* a host of scientific models: phenomenological models, fictional models, data models, theoretical models, they all need to be readjusted in order to be implemented on the computer. This process of ‘readjusting’ or ‘recasting’ a host of models is part of the

methodology of computer simulations, and has no counterpart in the philosophy of models.

Indeed, a central aspect of any philosophy of modeling is to establish categories for the classification of scientific models. Such categories are at the heart of the differences among scientific models: *phenomenological models* are not *theoretical models*, which in turn differ from *data models*, and so forth (see my classification in Section 1.2.2.2). A computer simulation, however, cannot directly implement a scientific model, but rather it must be transformed into a ‘simulation model’ (see Chapter 2 for a more detailed discussion on how this is done). Such transformation procedures facilitate the implementation of a host of diverse scientific models as disparate as theoretical models and phenomenological models. In this way, a simulation model has the capacity to recast and implement a theoretical model in the same way as a fictional model. Let me illustrate this claim with a simple example: a phenomenological model, such as Tycho’s model of the motion of planets, can be used for a computer simulation in the same sense as a Newtonian model. In this sense, a simulation of Tycho’s model would produce new data, predict the next position of the planetary system, and similar epistemic activities as Newton’s model. From a purely methodological point of view, the process of adjusting Tycho’s model to a simulation model is no different from what the researcher would do with a Newtonian model. Note that I am not saying that implementing Tycho’s model provides the same knowledge as Newton’s model. Instead, I am calling attention to the fact that simulation models put a host of scientific models on equal footing, as they act as a ‘super class’ of models. The criteria typically used for classifying scientific models, then, do not apply for a simulation model.<sup>68</sup>

## 1.4.2 Changes in ontology

Recasting scientific models is simply the tip of the iceberg. There are changes of ontology (i.e., continuous models implemented as discrete models<sup>69</sup>) that take place when recasting a scientific model into the simulation model. Despite the fact that there are scientific models that are fundamentally discrete,<sup>70</sup> in the natural sciences it is more common to find continuous models that represent the diversity of empirical systems, such as ordinary differential equations or partial differential equations. Since computers are discrete systems, mathematical techniques must be in place in order to make possible the transformation from continuous equations to discrete ones.

A good discretization method, therefore, has to balance the loss of information intrinsic to these kinds of transformations along with ensuring reliability of results.<sup>71</sup>

Such a balance is not always easy to obtain, and sometimes *ad hoc*.

It is worth noting that, except for technical limitations, virtually every scientific model can be implemented on a computer.<sup>72</sup>

### 1.4.3 Representational languages

Another interesting problem that computer simulations pose for the philosopher of science (and to the philosopher of language) is the host of representational languages (i.e., programming languages<sup>73</sup>) available. Raymond Turner makes an interesting claim about programming languages and their relation to representational theories. He says. “via its data types, every programming language outlines a *theory of representation* that could be made flesh via an axiomatic account of its underlying types and operators” (Turner, 2007a, 283). Instead, in the philosophy of models there is little room for study on language, for it is mostly reduced to mathematics or physics.

### 1.4.4 Abstraction in computer simulations

The introduction of new modes of abstractions in computer simulations is also of interest. Besides the traditional idealizations, fictionalizations, and the like found in the philosophy of models (see Section 1.2.2.1), computer simulations introduce a hierarchical notion of abstraction, which includes *information hiding* and *information neglecting*. Simply put, the different levels of programming a computer software requires that information is sometimes hidden from the user. As Colburn and Shute point out:

[there are] details that are essential in a lower-level processing context but inessential in a software design and programming context. The reason they can be essential in one context and inessential in another is because of abstraction and information hiding tools that have evolved over the history of software development (Colburn and Shute, 2007, 176).

The problem of information hiding and information neglecting is typically overcome by the selection of an appropriate computer specification language (see Chapter 2).<sup>74</sup>

### 1.4.5 Reliability of results

On the justificatory side, *verification* and *validation* are methodologies for the reliability of the simulation model. William Oberkamp and Christopher Roy explain that in order to measure the correctness of a model, one must have accurate benchmarks or reference values against which to compare the model. Benchmarking

is a technique used for measuring the performance of a system, frequently by running a number of standard tests and trials against it. In verification, the primary benchmarks are highly accurate solutions to specific, although limited, mathematical models. On the other hand, in validation, the benchmarks are high-quality experimental measurements of systems responsible for the quantities of interest.<sup>75</sup> In verification, then, the relationship of the simulation to the empirical world is not necessarily at stake; whereas in validation, the relationship between computation and the empirical world, that is, the experimental data obtained by measuring and observing methods, is the issue.<sup>76</sup> These methods are novel insofar as they require specific elements of computer science (for a more detailed discussion, see Section 3.3.2).

The motivation here was only to supplement Humphreys' list on the philosophical importance of the study on computer simulations, not to engage into the topic. Indeed, this work is not about analyzing in any more depth these motivations, but rather to study what it is that makes computer simulations epistemically powerful for scientific practice

There is great interest for understanding computer simulations fostered by their central role in scientific practice. Philosophers have addressed these interests via studies on traditional philosophy of models and scientific explanation. I have argued here that, although not misleading, these approaches fail to capture the rich universe that computer simulations have to offer. Motivated by these results, I turn to my central aims, which are to address computer simulations at face value and to show their importance as epistemic devices for the scientific enterprise.

## 1.5 Conclusion

This chapter presented the general motivations and aims for this dissertation by briefly analyzing the reception that computer simulations have had in current philosophical literature. It also provided the necessary philosophical terminology for framing my study of computer simulations.

The study of classes of 'models' and 'representation' is fundamental for my own classification of computer simulations, as I develop it in Section 3.2.3. Similarly, the working definition of *goodness of representation* is paramount for understanding the correspondence between computer simulations and empirical systems.

As for the study on 'experimentation,' fleshing out the notion of *causality* as the central ontological feature of experiments provides the necessary criteria for the

first distinction from computer simulations. The discussion on the ‘epistemology of experiment’ is also useful for delimiting the scope of computer simulations as devices used for accessing some aspects of the world, as well as for discouraging certain philosophical practices (I discuss the ontology of computer simulations in Chapter 2 and its methodology in Chapter 3). Finally, the discussion in Section 1.4 shows, unequivocally to my mind, the importance and novelty of computer simulations for philosophical inquiry.

By means of these analyses, I organized the terminology that will be used in the following chapters as I encourage a change of angle in the analysis of the epistemic power of computer simulations. The following chapter will begin with these changes by analyzing the notion of ‘computer simulation.’ This time, however, I will make use of a more appropriate philosophy, one that provides the right conceptual tools: the philosophy of computer science.

## Notes

<sup>1</sup>Cf. (Rohrlich, 1990, 507)

<sup>2</sup>For further reading, see (Coffa, 1991).

<sup>3</sup>A shortlist of eminent logical empiricists includes (Ayer, 1946; Carnap, 1937, 1932; Hempel, 1935; Nagel, 1945).

<sup>4</sup>For a thorough study on these issues, see (Suppe, 1977, 1989).

<sup>5</sup>Cf. (Suppe, 1977, 45).

<sup>6</sup>Cf. (Harré, 1981, 175).

<sup>7</sup>For further reading, see (Bridgman, 1959, 1927; Hempel and Oppenheim, 1945).

<sup>8</sup>(Quine, 1951).

<sup>9</sup>(Popper, 1935).

<sup>10</sup>On scientific progress, see for instance (Kuhn, 1962). For a classic work of Feyerabend, see (Feyerabend, 1975).

<sup>11</sup>See (Lakatos et al., 1980), (Leplin, 1982), and (Hempel, 1965) respectively.

<sup>12</sup>Cf. (Humphreys, 2004, 96-97).

<sup>13</sup>Of special interest are (Suppe, 1977, 1989; van Fraassen, 1980).

<sup>14</sup>See (Cartwright, 1983; Morgan and Morrison, 1999).

<sup>15</sup>See (Cartwright et al., 1995).

<sup>16</sup>See (Morgan and Morrison, 1999), (Suárez, 2003), (Hartmann, 1999), (Giere, 2004), and (Keller, 2000) among others.

<sup>17</sup>Cf. (Müller and Müller, 2003, 1-31).

<sup>18</sup>This fact is especially true when there are scale differences between the model and the empirical system.

<sup>19</sup>See (Frigg, 2009a) and (Weisberg, 2013), respectively.

<sup>20</sup>Cf. (Hartmann, 1999, 344).

<sup>21</sup>See (Brown, 1991), (Tymoczko, 1979; Swart, 1980), and (Girle, 2003) respectively.

<sup>22</sup>For *representational isomorphism*, see (van Fraassen, 1980; Suppes, 2002). For *partial isomorphism*, see (Mundy, 1986; Díez, 1998). For *structural representation*, see (Swoyer, 1991). For *homomorphism*, see (Lloyd, 1988; Krantz et al., 1971). For *similarity*, see (Giere, 1988; Aronson et al., 1993). For the *DDI account*, see (Hughes, 1997). For a more detailed discussion on each account, see the Lexicon.

<sup>23</sup>Cf. (Frigg, 2009b, 51).

<sup>24</sup>Cf. (Weisberg, 2013, 27).

<sup>25</sup>In all fairness, the author identifies his account only provisionally with isomorphism (cf. Swoyer, 1991, 457). For the sake of the argument, however, such a provisional identification is unimportant.

<sup>26</sup>See, for instance, (Jones and Cartwright, 2005; Liu, 2004b,a; Suárez, 2009; Weisberg, 2007).

<sup>27</sup>For an analysis on *reliability*, see Section 4.2.1.

<sup>28</sup>See (Morgan and Morrison, 1999).

<sup>29</sup>Cf. (Hartmann, 1999, 327).

<sup>30</sup>For further reading, see (Redhead, 1980; McMullin, 1968; Frigg and Hartmann, 2006).

<sup>31</sup>Cf. (Da Costa and French, 2000, 121).

<sup>32</sup>For instance, see (Galison, 1997; Staley, 2004).

<sup>33</sup>Cf. (Frigg and Hartmann, 2006, 743).

<sup>34</sup>For further reading, see (Harris, 2003).

<sup>35</sup>Among international groups working on observatory data models, there is the ‘International Virtual Observatory Alliance,’ <http://www.ivoa.net/pub/info/>; and the ‘Analysis of the Interstellar Medium of Isolated GALaxies’ (AMIGA), <http://amiga.iaa.es/p/1-homepage.htm>.

<sup>36</sup>The list of philosophers of experimentation is long, but it would certainly include (Hacking, 1988, 1983; Franklin, 1986; Galison, 1997; Woodward, 2003; Brown, 1994, 1991).

<sup>37</sup>(Hanson, 1958, 1963).

<sup>38</sup>See (Franklin, 1990).

<sup>39</sup>See (Harré, 1981) and (Aristotle, 1965).

<sup>40</sup>See (Quine, 1951; Duhem, 1954).

<sup>41</sup>(Ackermann, 1989; Mayo, 1994).

<sup>42</sup>For further reading, see (Kraftmakher, 2007; Radder, 2003).

<sup>43</sup>Cf. (Nitske, 1971).

<sup>44</sup>For a study on *counterfactual theories*, see (Lewis, 1973). For *probabilistic causation*, see (Weber, 2009). For *causal calculus*, see (Hida, 1980).

<sup>45</sup>(Salmon, 1984).

<sup>46</sup>I have simplified Dowe’s definition of causal interaction. A more complete discussion must include the notions of (and distinctions between) ‘causal process’ and ‘causal interaction’ (see Dowe, 2000, 89).

<sup>47</sup>(Kosso, 2010, 1989).

<sup>48</sup>For a distinction between these two terms, see the Lexicon.

<sup>49</sup>Cf. (Franklin, 1990, 104ff).

<sup>50</sup>By ‘success’ here I simply mean that the experiment provides the correct results.

<sup>51</sup>Cf. (Franklin, 1989, 441).

<sup>52</sup>Cf. (Hacking, 1983, 194-197).

<sup>53</sup>(Penzias and Wilson, 1965a,b).

<sup>54</sup>See, for instance, (Dicke et al., 1965).

<sup>55</sup>(Franklin, 1981; Friedman and Telegdi, 1957; Wu et al., 1957).

<sup>56</sup> For instance, (Swart, 1980; Tymoczko, 1979, 1980).

<sup>57</sup> Cf. (Franklin, 1981, 368).

<sup>58</sup> See Section 3.2 for further discussion on this point.

<sup>59</sup> See (Holton, 1999).

<sup>60</sup> See (Cirpka et al., 2004).

<sup>61</sup> Cf. (Frigg and Reiss, 2009, 594-595).

<sup>62</sup> It can be read: “[t]he philosophical problems that do come up in connection with simulations are not specific to simulations and most of them are variants of problems that have been discussed in other contexts before. This is not to say that simulations do not raise new problems of their own. These specific problems are, however, mostly of a mathematical or psychological, not philosophical nature” (Frigg and Reiss, 2009, 595).

<sup>63</sup> Such a conceptualization is fully developed in Chapter 2 and Chapter 3.

<sup>64</sup> Cf. (Humphreys, 2009, 617).

<sup>65</sup> Humphreys makes the distinction between scientific practice *completely* carried out by computers (one that he calls *the automated scenario*) and one in which computers only *partially* fulfill scientific activity (*the hybrid scenario*). He restricts his analysis, however, to the hybrid scenario (cf. Humphreys, 2009, 616-617).

<sup>66</sup> Cf. (Humphreys, 2004, 8).

<sup>67</sup> See (Humphreys, 2009, 618ff). In (Durán, Unpublished) I claim that the anthropocentric predicament is silent on whether we are justified in believing that the results of a computer simulation are, as assumed, reliable. In here I also raise some objections to the ‘epistemic opacity,’ and ‘the in practice/in principle’ distinction (see Section 2.2 and Section 3.3.2).

<sup>68</sup> See, for instance (Morgan and Morrison, 1999, 10ff).

<sup>69</sup> For further reading, see (Atkinson, 1964).

<sup>70</sup> For instance (Young, 1963).

<sup>71</sup> For further reading, see (Kotsiantis and Kanellopoulos, 2006; Boulle, 2004).

<sup>72</sup> Computer models are subject to the Church-Turing thesis, that is, limited by functions whose values are ‘effectively calculable’ or algorithmically computable. See (Church, 1936; Turing, 1936).

<sup>73</sup> For instance (Butterfield, 2008; Turner, 2007b).

<sup>74</sup> For further reading, see (Ainsworth et al., 1996; Guttag et al., 1978; Spivey, 2001).

<sup>75</sup> See (Oberkampff and Roy, 2010).

<sup>76</sup> See (Oberkampff et al., 2003).

# Chapter 2

## A survey of the foundations of computer science

### 2.1 Introduction

Philosophers dealing with computer simulations have used the very notion of *computer simulation* in a rather loose way. Most prominent is the conception that a computer simulation is a dynamic system implemented on a physical computer<sup>1</sup>. But what exactly constitutes the notion of *computer simulation*? In a broad context, computer simulations are just another type of computer software, like the web application of a bank or the database at the doctor's office. I begin the study of computer simulations by analyzing them as general purpose computer software. More specifically, I propose to divide the study of general purpose computer software into three main units of analysis, namely, the *specification*, the *algorithm*, and the *computer process*. Thus understood, I foster this study from the perspective of the philosophy of computer science, rather than the philosophy of science, as it has been typically addressed.

With these interests in mind, I have divided this chapter in the following way: Section 2.2 defines the notion of computer *specification*, and deals with the *methodological* and *epistemological* uses. On the methodological side, specifications are taken as a blueprint for the design, programming, and implementation of algorithms; on the epistemic side, however, they bond the scientists' knowledge about the target system (i.e., an empirical model, a theoretical model, a phenomenological model, etc.) together with the scientists' knowledge about the computer system (architecture, OS, programming languages, etc.).

Section 2.3 discusses the notion of algorithm, as well as the formal and non-formal methods involved for its interpretation, design, and confirmation. I also

discuss some properties of algorithms that make them interesting for scientific representation, such as *syntax manipulability*, which is associated with the multiple (and equivalent) ways of representing the same target system (e.g., Lagrangian or Hamiltonian representations). The other important property of an algorithm is *syntax transference*, that is, the simple idea that by adding just a few changes, the algorithm can be used in different representational contexts.

Finally, Section 2.4 addresses the notion of *computer process* as the set of activated logical gates (by the algorithm) working at the hardware level. I then argue that both the algorithm and the computer process are step-wise processes. From here two conclusions follow: first, the results of the computation fall within the algorithm's parameter domain (or, more specifically, within the specification and algorithm's space of solutions), and second, that the algorithm and the computer process are epistemically equivalent (albeit ontologically dissimilar). These conclusions are of central importance for my conceptualization of computer simulations.

This chapter, then, has two aims in mind: first, to discuss the ontology of computer software along with the epistemic and methodological characteristics that it offers; and second, to prepare the terminology used in the remainder of this dissertation.

## 2.2 Specifications

Every scientific instrument requires the elucidation of its functionality, a description of the limits within which it can be used, how it must be managed in order to ensure maximum operability, and so forth. All this information combined constitutes the *specification* for a scientific instrument.

To illustrate this point let me use as example the mercury-in-glass thermometer. Let us assume the following specification:

1. Insert mercury in a glass bulb attached to a glass tube of narrow diameter; the volume of the mercury in the tube must be much less than the volume in the bulb; calibrate marks on the tube that vary according to the heat given; fill the space above the mercury with nitrogen.
2. This thermometer can only be used for measuring liquids, body temperature, and weather; it cannot measure above  $50^{\circ}\text{C}$ , or below  $-10^{\circ}\text{C}$ ; it has an interval of  $0.5^{\circ}\text{C}$  between values and an accuracy of  $\pm 0.01^{\circ}\text{C}$ ; the mercury in the thermometer solidifies at  $-38.83^{\circ}\text{C}$ ;
3. Instructions for the correct use of the thermometer: insert the mercury bulb

located at the end of the thermometer into the liquid (under the arm, or outside but not under direct sun) to be measured, locate the value indicated by the height of the mercury bar, and so forth.

In addition to these specifications, the researcher also knows relevant information about the empirical system. For instance, that water solidifies at  $0^{\circ}$  C; that if the liquid is not properly isolated, then the measurement may be biased by a different source of temperature; that the Zeroth law of thermodynamics explains the measurement of the physical property ‘temperature,’ and so forth.<sup>2</sup> Such information is fundamental for establishing the reliability of the results. Indeed, any misuse of the thermometer leads to inaccurate measurements and, therefore, to having good reasons for disregarding the measured data as unreliable. Similarly, to say that a thermometer carries out the required measurement, that the measurement is precise and accurate,<sup>3</sup> and that the values obtained are reliable measurements, is also to say that the measurement has been carried out within the specifications given by the manufacturer.

An interesting case was recorded by the famous physicist Richard Feynman. During his time on the research committee investigating the accident of the Challenger, he had the following dialog with the manufacturer of an infrared scanning gun:

Sir, your scanning gun has nothing to do with the accident. It was used by the people here in a way that’s contrary to the procedures in your instruction manual, and I’m trying to figure out if we can reproduce the error and determine what the temperatures really were that morning. To do this, I need to know more about your instrument. (Feynman, 2001, 165-166)

The situation was probably very frightening for the manufacturer, for he thought that the scanning gun did not work properly. That was not the case. To Feynman, as well as to any other scientist, it is paramount to follow the specifications of the instrument for they are a fundamental piece of knowledge about the instrument.

A specification, then, has a *methodological* purpose as well an *epistemic* functionality: methodologically speaking, it works as the blueprint for the design, construction, and use of an instrument; epistemically speaking, it works as the repository of the knowledge about that instrument. In this vein, the specification has a double purpose: it provides relevant information for the construction of an instrument as well as insight into its functionality. From the example above, point 1 illustrates how to construct a thermometer, including the need to calibrate it; point 2 illustrates the upper and lower boundaries in which the thermometer measures and can be used as a reliable instrument; point 3 illustrates the correct use of the thermometer for successful measurements.

Typically, scientists only need some basic instruction on how the instrument works in order to carry out their measurements or observations. However, the sociological fact that scientists know little or nothing about the internal functions of their instrument says nothing about the benefit of using the specification as a component for epistemological evaluation. Therefore, for the rest of this study I assume that the scientist has access to information about the instrument, and that access is granted via the specification. Let it be noted that scientists also know their instruments by using them. This pragmatic dimension is fundamental for the success of scientific practice and should not be neglected. Our interest here, however, lies in the conceptual information built into the instrument. These assumption are especially important in the domain of computer software, particularly if the scientists design and program their own software packages.<sup>4</sup>

Modern computer science tells us that scientific models cannot be directly implemented on the physical computer. Instead, they must first be transformed into a *computer model*, which includes a re-interpretation of the original set of equations (e.g., via discretization methods, mathematical transformations, and other mathematical and logical machinery), additional information about the target system (e.g., that the software will be used in such and such circumstances), details about the computer architecture (e.g., whether the computer is 64, 128, 512 bits), and similar features. The specification, as conceived by computer scientists, becomes the backbone for the construction, programming, execution, and subsequent comprehension of scientific models in the physical computer. Brian Cantwell Smith defines a computer specification in the following way:

*A specification:* a formal description in some standard formal language, specified in terms of the model, in which the desired behavior is described. [...] [A]ll it has to do is to specify *what proper behavior would be*, independent of how it is accomplished. For example, a specification for a milk-delivery system might simply be: “make one milk delivery at each store, driving the shortest possible distance in total”. That’s just a description of what has to happen. [...] Specifications, to use some of the jargon of the field, are essentially *declarative*; they are like indicative sentences or claims (Cantwell Smith, 1985, 20).

Cantwell Smith’s definition is too strict, since it reduces the notion of specification to a *formal description* of the behavior of the target system. Computational practice has shown us that specifications cannot be fully formalized. Rather, they must be conceived as ‘semi-formal’ descriptions of the behavior of a target system. By this I mean that formal as well as non-formal descriptions coexist in the specification. In other words, mathematical and logical formulae coexist with documentation,

instructions, and other information written in plain English. Now, although there is general agreement that full formalization of the specification is a desirable aim, it is not always attainable.<sup>5</sup>

A more precise conceptualization of the notion of specification is given by Shari Pfleeger and Joanne Atlee:

[The specification] describ[es] the external visible properties of the software unit. Just as a requirement specification describes system behavior in terms of entities at the system's boundary, an interface specification's description refers only to entities that exist at the unit's boundary: the unit's access functions, parameters, return values, and exceptions (Pfleeger and Atlee, 2009, 303).

Pfleeger and Atlee continue describing general attributes of the software that need to be specified. The following is a shortlist of what typically is included as a specification (see Figure 2.1):

1. *purpose*: it documents the functionality of each access function, modification of variables, access to I/O, etc. This should be done in enough detail that other developers can identify which access functions fit their needs;
2. *preconditions*: e.g., values of input parameters, states of global resources, other software units;
3. *protocols*: such as information about the order in which access functions should be invoked;
4. *postconditions*: all possible visible effects of accessing functions are documented, including return values, raised exceptions and changes to shared variables such as output files;
5. *quality attributes*: e.g., performance, reliability, memory use, and the like;
6. any extra information must also be documented into the specification, such as specifics of the programming languages used, libraries, protocols implemented, relations held by structures, functions implemented and so forth;
7. the specification of a software is also the place where political, ethical, and design decisions are being made, and therefore must be included in the software.

Equally important is to specify the actions to take when the software behaves in abnormal situations, such as invalid input or errors during the computation.<sup>6</sup> The functional requirements of the specification must clearly state what the software must do in these kinds of situations. A good specification must follow specific

principles of robustness, correctness, completeness, stability, and similar desirable characteristics.<sup>7</sup>

In this vein, and similarly to the case of scientific instruments, a specification for computer software plays two central roles: it plays a *methodological* role as the blueprint for the design, programming, and implementation of algorithms, and it plays an *epistemological* role of bonding the representation and knowledge about the target system together with knowledge about the computer system (i.e., the computer's architecture, OS, programming languages, etc.). Since in this section I am only interested in addressing the epistemological significance of software specifications, I leave the study of their methodological significance to Section 3.3 where I analyze a concrete example of a computer simulation.

As discussed above, by describing the behavior of the target system, the software specification restricts the representation and knowledge about a target system with our knowledge of the computational system. This means that the specification makes computer software possible because it works as a 'conceptual bridge' connecting these two kinds of knowledge. This is not a mysterious connection, of course, but rather is the basis of the professional activity of the computer scientist: a computer system must be specified in all detail before it is coded into an algorithm. This process saves time, money, resources, and more importantly, reduces the presence of errors, misrepresentations, miscalculations, etc.

It follows, then, that the specification is, cognitively speaking, the most transparent unit in a computer software. This is immediately clear when compared with an algorithm or a computer process: the former, although still accessible, is written in some programming language, making it harder to understand than the specification; the latter, on the other hand, is rather impossible to have any cognitive access to at all for it consists of machine language, positions in memory, and the like. Simply put, the specification is a repository of knowledge about the target system and the computational system as a whole which the computer scientist can access at any moment for modifications, interpretations, and similar cognitive activities.

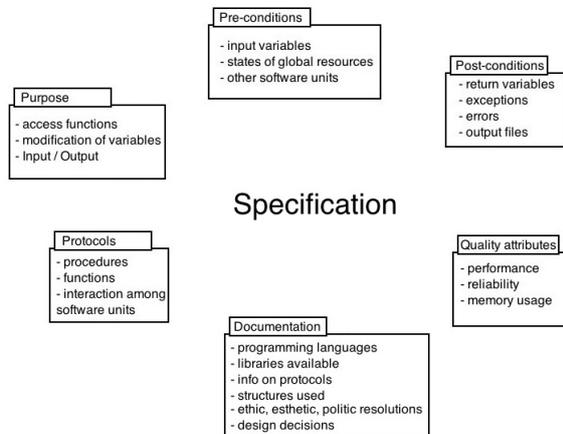


Figure 2.1: The general schema of a specification

Allow me to illustrate this point with an example: consider the specification of a simple system, such as a vending machine. Its specification includes the functionality of the vending machine when the customer inserts a coin, the procedure triggered when a beverage is selected, and so on. Such knowledge about the target system must be complemented with knowledge about the computer system. For example, the computer architecture does not allow more than six beverage choices, or no change is given back, etc. This example nicely illustrates that our understanding of the vending machine (i.e., its functionality, its limitations, etc.) has been explicitly described in this ‘repository of knowledge’ that is the *specification*.

There is another reason why the notion of specification is important for the study of computer simulations: it allows scientists to disregard considerations about computer simulations as ‘black boxes.’ Indeed, without the cognitive access that the specification grants, there are no grounds for claiming that the simulation is implementing the correct model, that the results are representative of the simulation model, that the scientists know what is being simulated, etc. The idea of computer simulations as ‘black boxes’ has been brought forward by Paul Humphreys through the notion of *epistemic opacity*. Briefly, a computer simulation is epistemically opaque when it is cognitively inaccessible for humans. He presents his argument in the following way:

In many computer simulations, the dynamic relationship between the initial and final states of the core simulation is epistemically opaque because most steps in the process are not open to direct inspection and verification. This opacity can result in a loss of understanding because in most traditional static models our understanding is based upon the ability to decompose the process between model inputs and outputs into modular steps, each of which is methodologically acceptable both individually and in combination with the others. (Humphreys, 2004, 147-148)

If Humphreys is right about the impossibility of decomposing the computer simulation, inspecting it, and verifying it, then the only way we can confirm that the results of the simulation are representative of the model is via a process of *validation*. Briefly, validation is a method for sanctioning the computer simulation by comparing the results of the simulation with data from some other sources (e.g., measured data). Epistemically speaking, however, the process of validating results does not provide grounds for the *reliability* of the computer simulation, but merely *confidence* in the correctness of the results.<sup>8</sup> A computer simulation that is not *reliable* cannot yield understanding of the results that it simulates, and therefore, lacks grounds for claiming any epistemic virtue. I leave the discussion of the notion of *reliability* and the importance for the analysis on the epistemic power of computer simulations for Chapter 4.

The situation can actually be a bit worse. Epistemic opacity not only forces scientists to adopt a process of validation for a simulation's results, but it also prevents verification of the simulation model. The process of 'checking' and 'matching' that is validation works well at the high level of results, but not at the internal level of the implemented model. The process of validation, as important as it is, must be complemented by a process of verification, which by definition requires access to the system to be verified. Verifying the simulation model, as I argue in Section 3.3.2, is paramount for the final validity of the results, and the entire reliability of the computer simulation. The process of verification of the computer simulation can only be carried out if there is an epistemically transparent unit that facilitates it; that is, the specification.

In private conversation, Humphreys admitted that his notion of epistemic opacity was too strong and needed some reconsiderations.<sup>9</sup> 'Epistemic opacity' now is related to the mathematical transformations involved in the process of converting the scientific model into the specification. I believe he is right on this point, as I have argued before (see Section 1.4). Consider for instance the process of discretization for a set of continuous equations, such as ordinary differential equations (ODE). The Runge-Kutta method, for instance, is a very successful method that would approximate the solutions of the ODE, although not without some loss of information and error.

The epistemological importance of the specification, then, is that it yields insight into the instrument, either a scientific instrument or a computer software. In the particular case of the latter, the specification becomes the most transparent and cognitively accessible unit of the computer software. This is a big responsibility since our knowledge about target system and computer system coexist in this repository of information. I will come back to the notion of specification on several other occasions, especially during my analysis of the explanatory power of computer simulation. As I explain there (especially Chapter 4 and Chapter 5), the specification enables computer simulations to be used as explanatory devices. Without it, the very notion of successful explanation would be at stake.

For the time being, there are more definitions and terms to be clarified. Allow me now to continue with the study of the algorithm, the logical structure in charge of interpreting the specification in a machine-readable language for its implementation on the digital computer.

## 2.3 Algorithms

Most of our daily activities can be described as simple sets of rules that we systematically repeat. We wake up at a certain time, brush our teeth, take a shower, and take the bus to work. We modify our routine, of course, but just enough to make it more advantageous in some way: it gives us more time in bed, it minimizes the distance between stops, it satisfies everyone in the house, etc.

Unlike our previous study on specification (and somehow different from the routines described above), the study of the *algorithm* rests on the idea that it is a systematic and formal method for implementing a specific set of instructions. Chabert defines it as “a set of step by step instructions, to be carried out quite mechanically, so as to achieve some desired result” (Chabert, 1994, Introduction). Under this definition, the following collision-less particle-in-cell system as presented by Michael Woolfson and Geoffrey Pert is an algorithm:

1. Construct a convenient grid in the one-, two- or three-dimensional space within which the system can be defined. [...]
  2. Decide on the number of superparticles, both electrons and ions, and allocate positions to them. To obtain as little random fluctuation in the fields as possible it is required to have as many particles as possible per cell. [...]
  3. Using the densities at grid-points, Poisson’s equation:  $\nabla^2\phi = -\rho/\epsilon_0$
  4. [...]
- N. If the total simulation time is not exceeded then return to 3. (Woolfson and Pert, 1999, 115)

This example of an algorithm includes statements in plain English as well as mathematical machinery. In a way, this looks more like a specification for a collision-less particle-in-cell system, rather than an algorithm. What we need, then, is a more accurate definition of an algorithm.

In the 1930’s, the concept of an algorithm was popularized in the context of computer software. In this vein, it was important to spot which features any computer algorithm must have. A shortlist includes:

1. An algorithm is defined as a finite and organized set of instructions, intended to provide the solution to a problem, and which must satisfy certain sets of conditions;
2. The algorithm must be capable of being written in a certain language;
3. The algorithm is a procedure which is carried out step by step;
4. The action at each step is strictly determined by the algorithm, the entry data and the results obtained at previous steps;

5. Whatever the entry data, the execution of the algorithm will terminate after a finite number of steps;
6. The behavior of the algorithm is physically instantiated during the implementation on the computer machine. (Chabert, 1994, 455)

Many structures in computer science fulfill these features for an algorithm: Hoare triple is a good example of this (e.g., Algorithm 2.1). Pseudo-codes, that is, descriptions of algorithms intended for human reading rather than machine reading, are another good example of algorithms (e.g., Algorithm 2.2).<sup>10</sup> Of course the notion of algorithm is not limited to abstract representations, for it could also be written in a programming language. The universe of programming languages is significantly large, but one example of each kind would suffice: C, Java, Perl, and Haskell. The first is an example of low level programming language (e.g., Algorithm 2.3); the second is an Object Oriented Programming Language; Perl is a good example of a scripting language; and Haskell is the exemplar of functional programming.<sup>11</sup>

---

**Algorithm 2.1** Hoare triple

---

A Hoare triple is a theoretical reconstruction of an algorithm. It consists of a formal system (initial, intermediate, and final states) that strictly obeys a set of logical rules. It has been developed for formal validation of the correctness of computer programs and it cannot be implemented on the computer.

The Hoare triple has the form:  $\{P\}C\{Q\}$

where  $P$  and  $Q$  are assertions and  $C$  is a command.  $P$  is named the *precondition* and  $Q$  the *postcondition*. When the precondition is met, the *command* establishes the postcondition. Assertions are formulae in predicate logic.

Rules:

Empty statement axiom schema:  $\frac{}{\{P\}Skip\{P\}}$

Assignment axiom schema:  $\frac{}{\{P[E/x]\}x:=E\{P\}}$

---

From an ontological point of view, an algorithm is a *syntactic* structure encoding the information set out in the specification. Let us note that studies in computer science make use of the notion of *syntax* and *semantics* in a rather different way than linguistics. For computer science, *syntax* is the study of symbols and their relationships within a formal system; it typically includes a grammar (i.e., a sequence of symbols as well-formed formulas), and proof-theory (i.e., a sequences of well-formed formulas that are considered theorems). On the other hand, *semantics* is the study of the relationship between a formal system, which is syntactically specified, and a

---

**Algorithm 2.2** Pseudo-code

---

Pseudocode is a non-formal, high-level description of the computer algorithm. It is intended to focus on the operational behavior of the algorithm rather than on a particular syntax. In this sense, it uses a similar language as a programming language but in a very loose sense, typically omitting details that are not essential for the understanding of the algorithm.

```
FOR all numbers in the array
  SET Temp equal the addition of each number
  IF > 9 THEN
    get the remainder of the number divided by 10 to that index
    and carry the "1"
  Decrement one
```

Do it again for numbers before the decimal

---

semantic domain, which can be specified by a domain taken to provide interpretation to the symbols in the syntactic domain. In the case of the implementation of the algorithm on the digital computer, the semantic domains are the physical states of the computer when running the instructions coded into the algorithm. I refer to these physical states as the *computer process*, and it will be discussed in the next section.

Two other ontological properties of interest are that the algorithm is *abstract* and *formal*. It is *abstract* because it consists of a string of symbols with no physical causal relations acting upon them: just like a logico-mathematical structure, an algorithm is causally inert and disconnected from space-time. It is *formal* because it follows the laws of logic that indicates how to systematically manipulate symbols.<sup>12</sup>

These ontological features offer an epistemic counterpart. Any abstract and formal system allows its variables to be syntactically transformed into logico-mathematical equivalents. Such a capacity comes in two forms: *syntax manipulability* and *syntax transference*.<sup>13</sup> Allow me to explain.

Consider a mathematical model. Such a model can be mathematically transformed by changing one formula for another formula isomorphic to it. For instance, there is mathematical isomorphism between  $2a$  and  $a + a$ , for both belong to the same algebraic groups. Thus understood, *syntax manipulability* is a feature that also applies to computer simulations. Consider the following Algorithm 2.4 and its equivalent Algorithm 2.5. Both are logically isomorphic, and the proof of this is in Table 2.1:

Paul Humphreys warns that syntax manipulability is not about mere mathemat-

---

**Algorithm 2.3** Imperative programming language C using local variables

---

Finally, an algorithm written in a programming language describes the specification in such a way that it can be implemented on the computer. In this work, when I refer to an algorithm, I will be referring to an algorithm written in a specific programming language (the particularities of the programming language are not important here).

Function *mean* finds the mean between two integers:

```
float mean(int num1, int num2)
{
    float p;
    p = (num1 + num2) / 2.0;
    return(p);
}

main()
{
    int a=7, b=10; float result;
    result = mean(a, b);
    printf("Mean=%f\n", result);
}
```

---

---

**Algorithm 2.4**

---

if (A) then {*a*} else {*b*}

---

ical power, but it also “makes the difference between a computationally intractable and a computationally tractable representation” (Humphreys, 2004, 97). For instance, Lagrangian and Hamiltonian sets of equations both represent dynamical systems. And despite the fact that both are equivalent in terms of their mathematics, the Hamiltonian representation has proven to be more suitable for some systems than Lagrangian’s (e.g., in quantum mechanics).<sup>14</sup> There is, therefore, a clear superiority of the Hamiltonian representation over the Lagrangian. Now, *representational superiority* must be accompanied by *algorithmic adequacy*, that is, the ability of an algorithm to be computationally more adequate than another. The issue is that in the same vein that syntax manipulability does not warrant representational superiority, the latter does not warrant algorithmic adequacy. The reason for this is that algorithmic adequacy depends on the capacity of a computer system to compute the mathematics, and not necessarily on its representational capacities. Indeed, an algorithm that codes Hamiltonian equations might not necessarily be algorithmically more adequate than one coding Lagrangian equations. Moreover, these two algorithms might not necessarily be isomorphic to each other. These facts nicely show that mathematical power and representational superiority are not positive features necessarily inherited by the algorithm.

---

**Algorithm 2.5**

---

if (not-A) then {b} else {a}

---

A	{a}	{b}	not-A	{a}	{b}
T	T	∅	F	T	∅
F	∅	T	T	∅	T

Table 2.1: Equivalency truth-table

The other interesting feature of algorithms is *syntax transference*. It is characterized by the simple idea that by adding just a few changes the algorithm could be reused for different representational contexts. As an example, consider Algorithm 2.6, which is a modification of Algorithm 2.3 replacing the two *local* integer variables ‘int a=7, b=10’ for two *global* integer variables. Syntax transference, then, allows scientists to reuse existing code, to generalize it, which therefore broadens the scope of the algorithm. This feature has been acknowledged by Humphreys in the following words: “[a]nother advantage of such syntactic transformations is that a successful model can sometimes be easily extended from an initial application to other, similar systems” (Humphreys, 2004, 97).

---

**Algorithm 2.6** Imperative programming language C using global variables

---

int a, b;

Function *mean* finds the mean between two integers:

```
float mean(int num1, int num2)
{
    float p;
    p = (num1 + num2) / 2.0;
    return(p);
}

main(a, b)
{
    float result;
    result = mean(a, b);
    printf("Mean=%f\n", result);
}
```

---

Admittedly, syntax manipulability and syntax transference beg the question about equivalency for a class of algorithms. The problem is the following. Syntax manipulability and syntax transference presuppose modifications of the original algorithm that lead to a new algorithm. Since algorithms are, as argued, syntactic

entities, then such modifications might entail an entirely new algorithm, one that significantly diverges from the original algorithm. Is Algorithm 2.3 equivalent to Algorithm 2.6? Equivalent in what sense? Is there a way to entrench equivalency for a class of algorithms? These are time-honored questions in the philosophy of computer science, and by no means will I provide a definite answer. However, two approaches have been agreed upon among computer scientists and philosophers alike: either two algorithms are ‘logically equivalent,’ that is, the two algorithms are formally isomorphic;<sup>15</sup> or they are ‘behaviorally equivalent,’ that is, the two algorithms behave in a similar fashion.

Let me briefly discuss these two approaches. Logical equivalency can be easily illustrated by using Algorithm 2.4 and Algorithm 2.5 since both are formally isomorphic and the proof of this is shown in Table 2.1. Although formal procedures of any kind always warrant isomorphism between two structures, it is not always attainable due to practical as well as theoretical constraints. Examples of the former include formal procedures that are humanly impossible to carry out, or are time and money consuming. Examples of theoretical constraints include programming languages that tend to have more expressive power than formal languages, making the latter less capable of being used in a formal procedure.

Behavioral equivalency, on the other hand, is easier to achieve than logical equivalency since it depends on the general behavior of the algorithm. It is exposed, however, to one concern and one objection. The concern is that behavioral equivalency is grounded in inductive principles. Indeed, behavioral equivalency could only be warranted for time  $t$ , when the observer corroborates the same behavior between algorithms, but not for  $t + 1$  which is the next unobserved state of the algorithm. In plain words, two algorithms could be behaviorally diverse in  $t + 1$  while equivalent in  $t$ . Full behavioral equivalency, therefore is only warranted, if then, by the time the two algorithms halt. The question remains if an *ex post facto* comparison between algorithms is of any interest to the computer scientist.

The objection stems from the fact that behavioral equivalency could hide logical equivalency. If this is the case, then two algorithms could diverge behaviorally while being logically equivalent. An example of this is one algorithm that implements cartesian coordinates whereas another implements polar coordinates. Both algorithms are isomorphic, but behaviorally dissimilar.

The general problem with behavioral equivalency is that it cannot ensure that two algorithms are the same, despite both behaving similarly. My concern suggests that two algorithms can be behaviorally equivalent up to certain number of steps and then diverge from each other. My objection indicates that two algorithms might

be logically equivalent, although behavioral equivalency hides it.

The lesson here is that *syntax manipulability* and *syntax transference* come at a price. Both are, however, a highly desirable feature for computer software. There are clear epistemic advantages in being able to implement (with a few changes) the Lotka-Volterra set of equations in a biological system as well as in an economic system.

Until now, I have discussed the conceptualization of an algorithm, along with its philosophical consequences as a syntactic formula. There is still one more issue to be discussed; namely, the existing link between the specification and the algorithm.

Ideally, the specification and the algorithm should be closely related, that is, the specification should be entirely interpreted as an algorithmic structure. For this there is a host of specialized languages that ‘mediate’ between the specification and the algorithm, such as Common Algebraic Specification Language (CASL), Vienna Development Method (VDM), or the Z notation,<sup>16</sup> just to mention a few specification languages. Model checking was conceived as the branch in computer science that automatically tests whether an algorithm meets the required specification, and in that sense it is also helpful in the interpretation of the specification into an algorithm.<sup>17</sup> In more general terms, the implementation of a specification into an algorithm has a long tradition in mathematics, logic, and computer science, and it does not represent a conceptual problem here.

I also mentioned that the specification includes non-formal elements, such as expert knowledge or design decisions that cannot be formally interpreted. These non-formal elements must also be interpreted in terms of the algorithm, otherwise it will not be part of the computer software. Take as an example the specification of an algorithm for computing statistics. Say that the team of statisticians decides to be very careful about the samples they collect, and pass on this concern to the group of computer scientists. Yet, the latter group fails to correctly specify some sensitive data, say the economic background of some minority included in the samples. If this specification is not corrected before the algorithm is implemented, then the results of computing the statistics will be biased.

This example shows that an algorithm must also be capable of interpreting non-formal elements included in the specification. There are no formal methods for interpreting non-formal elements of the specification. In situations like this, past experiences, expert knowledge, and ‘know-how’ become crucial in the interpretation and implementation of non-formal elements of the specification.

Where does this leave the question about the interpretation of the specification

and the algorithm? The credibility of computer software comes not only from the credentials supplied by the theoretical model used for constructing the simulation, but also (and probably more fundamentally) from the antecedently established credentials of the model building techniques employed in its construction. A history of prior success and accomplishments is the sum and substance of scientific progress. The set of techniques, assumptions, languages, and methods successfully employed for interpreting the specification into an algorithm contribute to this history. They fit well into our web of accepted knowledge and understanding of a computer system, and therefore are part of the success of science. A history of successful interpretation of the specification into the algorithm by the means described is enough credential for their reliability. Of course, better techniques will supersede current ones, but that is also part of this scientific progress, part of a history of successful methods that connect specifications and algorithms.

It is time to address the final component in computer software, namely, the *computer process*. Just one last terminological clarification. I reserve the term *computer model* or *computational model* for referring to the pair *specification-algorithm*. The reason for this is that both are essential in the representation of the target system. Let me now turn to the study of the computer process.

## 2.4 Computer processes

In previous sections, I made use of Cantwell Smith's notion of specification as the starting point for my study on computer software. Now it is time to complete his idea with the analysis of computer programs:<sup>18</sup>

The *program*: a set of instructions and representations, also formulated in the terms of the model, which the computer uses as the basis for its actions. How do these two [specifications and programs] differ? In various ways, of which one is particularly important. The program has to say *how the behavior is to be achieved*, typically in a step by step fashion (and often in excruciating detail). The specification, however, is less constrained: all it has to do is to specify *what proper behavior would be*, independent of how it is accomplished (Cantwell Smith, 1985, 22. Emphasis original.).

The first thing to notice is that 'the program' is the only physical component mentioned about the computer software. And, as Cantwell Smith points out, it is at the basis of the behavior of the physical computer. Now, unlike the specification, the program consists of a set of instructions that lead to a determinate sequence of actions to be performed by the physical computer. In this sense it is *procedural*, rather than *declarative*, as is the case of the specification.

To illustrate the difference between the computer process and the specification, take for example the description of a milk-delivery system: “make one milk delivery at each store, driving the shortest possible distance in total” (Cantwell Smith, 1985, 22). As Cantwell Smith explains, this is a description of what has to happen, but not *how* it will happen. The program is in charge of this last part, namely, it shows how the milk delivery takes place: “drive four blocks north, turn right, stop at Gregory’s Grocery Store on the corner, drop off the milk, then drive 17 blocks north-east, [...]” (Cantwell Smith, 1985, 22).

Now, what Cantwell Smith calls here ‘the program’ must be divided into two distinct components, namely, the *algorithm* and the *computer process*. This distinction is important because Cantwell Smith’s notion of program fails to differentiate a step-wise procedure understood as syntactic formulae and a step-wise procedure that puts the physical machine into causal states. In other words, there is an ontological difference that cannot be captured by his notion of program and it is crucial for a complete understanding of the nature of computer software. Let me begin by discussing this point.

The physical computer is structurally made to follow certain sets of rules built into its own hardware. Following the standard von Neumann architecture,<sup>19</sup> the arithmetic unit of the microprocessor can compute because it is constructed using logic gates. Logic gates are the physical implementation of the logic operators ‘and,’ ‘or,’ ‘not,’ and so forth, also present in the algorithm. Moreover, these logic gates are present not only in the arithmetic unit, but throughout the physical machine: memory, computer bus, I/O devices, etc. There, specific languages (so-called ‘hardware description languages’) such as VHDL or Verilog are used for building these logic gates into all the physical micro-circuits.<sup>20</sup> Such hardware description languages are, essentially, programming languages for the hardware architecture. It follows, then, that the lowest physical layer of the computer *implements* the algorithm because there is a common language that facilitates such implementation.

The implementation of the algorithm on the digital computer is what I call the *computer process*. A computer process, therefore, can be understood as the set of activated logical gates (by the algorithm) working at the hardware level. This result has two important consequences: first, an algorithm and a computer process are both step-wise processes in the sense that they both depend on a set of well-defined rules; second, the computer process is the physical realization of the algorithm, that is, the computer process physically implements the instructions set out by the algorithm.

The notion of ‘implementation’ here is taken in a semantic sense, that is, in the sense that a syntactic structure (i.e., the algorithm) is interpreted on a semantic domain (i.e., the physical computer). William Rapaport explains this in the following way: “terms get interpreted by, or mapped into, elements of the interpreting domain, and predicates (operations) are mapped into predicates of the interpreting domain” (Rapaport, 2005a, 388). Of course, the interpreting domain is, in this context, the physical states of the computer. In other words, semantic implementation is the correct terminology for the idea that the computer process is the physical concretization of the algorithm on the computer.<sup>21</sup> To complete the idea, it must be noted that such semantic implementation is carried out by the ‘compiler,’ that is, another computer program capable of mapping the elements of the domain of the algorithm into a computer language (see Figure 2.2). C. A. R. Hoare puts the same idea in the following way:

A program is a detailed specification of the behavior of a computer executing that program. Consequently, a program can be identified abstractly with a predicate describing all relevant observations that may be made of this behavior. This identification assigns a meaning to the program (Floyd, 1967), and a semantics to the programming language in which it is expressed (Hoare and Jones, 1989, 335).

It must finally be noted that if the computer architecture changes, say from an 8-bit to a 256-bit computer, then the compiler has the enormous responsibility to make this architectural change transparent for the scientist with a new mapping. For our purposes here, it is sufficient to characterize a computer process as the algorithm running on the physical machine.

Now, from a philosophical point of view, there are two outcomes of importance. The first one states that the results of the computation fall within the algorithm’s parameter domain (or, more specifically, within the specification and algorithm space of solutions). The second conclusion is that algorithm and computer process are epistemically equivalent, albeit ontologically dissimilar. Let me briefly address these two outcomes.

According to my description of the implementation, it follows that the algorithm ‘tells’ the computer how to behave. Thus interpreted, the computer process neither includes nor excludes any information that was not previously programed in the algorithm (unless, of course, there is a miscalculation or an error of some kind<sup>22</sup>). Now, this is a controversial point among philosophers. The most tendentious position is held by James Fetzer in the context of the verification debate.<sup>23</sup> Throughout his work, Fetzer holds that there are no reasons for believing that the computer process will not influence somehow the results of the calculation. That is, since the

computer process is a causal process subject to all kinds of physical conditions (e.g., changes of temperature, hardware failure, etc.), it is to be expected that there will exist differences between pen-and-paper results of calculating the algorithm and the results of the computer process. Fetzer's position has been strongly criticized on the grounds that he misrepresents the practice of computer science.<sup>24</sup> The most pervasive objection to his viewpoint is that there are no grounds for claiming that the computer process introduces unexpected modifications of results. This objection is appealing, for computer processes are, most of the time, reliable processes.<sup>25</sup> Now, Fetzer's position could be attacked from different angles. The easiest way to deal with these issues, however, is to reply that scientists replicate results and use statistical methods that provide sufficient guarantees that the results are within a given distribution of probability.

For our present purposes it is enough to presuppose that there are no miscalculations or mathematical artifacts of any kind that the computer process introduces in the results. This presupposition is harmless philosophically speaking, and achievable technically speaking. It follows, then, that the equations coded in the algorithm are more or less straightforwardly solved by the computer process. Given that no information is added or subtracted, those results must belong in the space of possible results, i.e., the algorithm's space of solutions. Allow me to illustrate this presupposition with an analogy. Suppose that any given person carries out a simple calculation in her head: unless her brain works in unexpected ways, in principle there are no reasons for thinking that the result will be wrong. If I am correct, then the consequence is that the results of the computation were already contained in the algorithm.<sup>26</sup>

The second outcome is that an algorithm and a computer process are epistemically equivalent (albeit ontologically dissimilar). This is actually a consequence of semantically implementing the algorithm, as discussed above. Since the algorithm and the computer process are linked by this semantic implementation, and since the algorithm is straightforwardly solved by the computer process, then both must be epistemically related. To be epistemically related means that the computer process contains the same information as the algorithm, albeit in a different form. It is in fact due to this epistemic equivalency that researchers are allowed to rely on the results of a computation. If they were epistemically different, then there are no guarantees that the results of a computation are reliable in any way. To illustrate this last point, take the simple mathematical operation  $2 + 2$  as an example. A possible algorithm written in language C may look like the following:

---

**Algorithm 2.7** A simple algorithm for the operation  $2 + 2$  written in language C

---

```
void main()  
{  
    return(2+2)  
}
```

---

Now, in binary code, this is represented by ‘00000010 *and* 00000010,’ for 2 in binary is 00000010 and the arithmetic operation ‘+’ is the logical operator ‘*and*.’ As I mentioned before, the compiler would ‘translate’ the algorithm into machine-readable language (in this case, it translates ‘+’ into ‘*and*’). Once the computer process is executed on the machine, it returns 4, for this is the result of running the ‘and’ operation in the machine, i.e.,  $00000100 = 4$ . The conclusion is that the knowledge included in the algorithm in the form of instructions is processed by the computer rendering reliable results. Epistemically speaking, then, they are equivalent.

Admittedly, the examples used here are rather simple, whereas running computer software is a much more complex process where a large number of instructions are activated at the same time. However, they illustrate quite accurately the basic operations of the computer, which was the purpose of this chapter.

Figure 2.2 summarizes the three units of computer software and their connections. At the top level there is the specification, where decisions for the computer software are made and integrated altogether. The algorithm is the set of instructions that interprets the specification and prescribes how the machine should behave. Finally, there is the computer process, i.e., the semantic implementation of the algorithm that will deliver the results.

The purpose of this chapter was to analyze the notion of computer software as composed of three units: the *specification*, the *algorithm*, and the *computer software*. Equally important was to understand their philosophical features, both individually and

as a working system. These results provide ontological and epistemic grounds for our study on computer simulations. In the following chapters, I restrict my view to

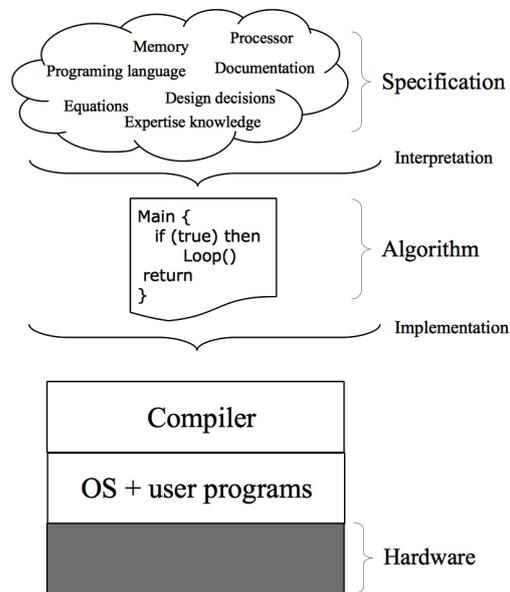


Figure 2.2: A general schema of the process of creating a computer software

one special kind of computer software, namely, computer simulations. I will study them from a methodological and epistemic point of view.

## 2.5 Conclusion

In this chapter I analyzed three central units of analysis of computer software from the perspective of the philosophy of computer science, namely, the *specification*, the *algorithm*, and the *computer process*.

The specification is understood as semi-formal descriptions of the behavior of a target system. The specification has a *methodological* purpose as well an *epistemic* functionality: methodologically speaking, it works as the blueprint for the design, construction, and use of an instrument; epistemically speaking, it works as the repository of the knowledge about that instrument.

An algorithm, on the other hand, is defined as a *syntactic* (abstract and formal) structure encoding the information set out in the specification. These ontological features offer an epistemic counterpart, namely, algorithms allow *syntax manipulability* and *syntax transference*. These concepts are fully discussed throughout the third section. Of equal importance were discussions on how the specification is coded as an algorithm.

Finally, the computer process is taken as the implementation of the algorithm on the digital computer. Thus understood, two important consequences follow: first, an algorithm and a computer process are both step-wise processes in the sense that both depend on a set of well-defined rules. Second, the computer process is the physical realization of the algorithm, that is, the computer process physically implements the instructions set out by the algorithm. The notion of ‘implementation’ is also discussed in some detail. Another essential discussion focused on the two philosophical consequences that stemmed from this notion of computer process, namely, that the results of the computation fall within the algorithm’s parameter domain, and that algorithm and computer process are epistemically equivalent (albeit ontologically dissimilar).

The results here obtained are of central importance for the conceptualization of computer simulations as a special kind of computer software. This latter point is elaborated in more detail in the next chapter.

# Notes

<sup>1</sup>See, for instance, (Hartmann, 1995; Humphreys, 1991; Parker, 2009; Winsberg, 2003).

<sup>2</sup>Admittedly I am only concerned about the specification of the instrument, in a broader context, though specifications of the experimental setup are also relevant for the understanding of results.

<sup>3</sup>For a more detailed analysis on the distinction between *precision* and *accuracy*, see the Lexicon.

<sup>4</sup>For a more thorough study on *specifications*, see (Raymond, 2006; Severi and Szasz, 2001; Turner, 2005).

<sup>5</sup>On this point, see (Arbab and Sirjani, 2009; Jalote, 2008; Pflieger and Atlee, 2009).

<sup>6</sup>I discuss errors in Section 3.3.3.

<sup>7</sup>See the Lexicon for a definition of each one of these terms.

<sup>8</sup>See Section 3.3.2.

<sup>9</sup>See also (Humphreys, 2009).

<sup>10</sup>See, for instance, (Hoare, 1969) and (Olsen, 2005) respectively.

<sup>11</sup>For more technical details on algorithms, see (Eden and Turner, 2005, 2007; Kao, 2008).

<sup>12</sup> See (Knuth, 1974, 1973) and (Dijkstra, 1974).

<sup>13</sup>These two notions were coined by (Humphreys, 2004, 95).

<sup>14</sup>Cf. (Humphreys, 2004, 97).

<sup>15</sup>Alternatives to isomorphism are also discussed in the literature (see, for instance, (Blass et al., 2009; Blass and Gurevich, 2003)). Let it be noted, however, that isomorphism is the only -morphism that could warrant total equivalency between algorithms.

<sup>16</sup>For CASL, see for instance (Bidoit and Mosses, 2004). For VDM, see for instance, (Bjørner and Jones, 1978; Bjørner and Henson, 2007). For Z notation, see for instance (Jacky, 1996; Spivey, 2001, 1992).

<sup>17</sup>See, for instance, (Queille and Sifakis, 1982; McMillan, 1992).

<sup>18</sup>I do not address the question about the computational architecture on which the computer software is run. However it should be clear from the context that I assume silicon-based computers (as opposed to quantum computers, biological computers, etc.). It must also be clear that other architectures represent a host of other philosophical problems (Berekovic et al., 2008; Rojas and Hashagen, 2000).

<sup>19</sup>See (von Neumann, 1945, 1948). Also, for an historical approach to this architecture, see (Aspray, 1990a; Eigenmann and Lilja, 1998).

<sup>20</sup>See, for instance (Cohn, 1989; Ciletti, 2010; Mermet, 1993).

<sup>21</sup>See also (Gutttag et al., 1978; Hoare, 1999; Rapaport, 1995, 2003; Hayes, 1988; Nelson, 1992).

<sup>22</sup>For a discussion on errors in computer science, see Section 3.3.3.

<sup>23</sup>For instance, in (Fetzer, 1988, 1991, 1997).

<sup>24</sup>For instance (Ashenurst, 1989).

<sup>25</sup>I discuss *reliability* in more detail in Section 4.2.1.

<sup>26</sup>This is the general claim in specialized literature on foundations of computer science. See, for instance, (Gruska, 1997; Eden, 2007; Eden and Turner, 2007; Hoare, 2004; Hoare and Jifeng, 1998; Hoare and Jones, 1989).

# Chapter 3

## Computer simulations: towards their conceptualization

### 3.1 Introduction

Current philosophical literature takes computer simulations as aids for overcoming imperfections and limitations of human cognition. Such imperfections and limitations have one common source: humans cannot process the enormous amount of information that current scientific activity manages.<sup>1</sup> Margaret Morrison, for instance, considers that although computer simulations are another form of modeling, “given the various functions of simulation [...] one could certainly characterize it as a type of ‘enhanced’ modelling” (Morrison, 2009, 47). In a similar sense, Paul Humphreys conceives a computer simulation as an ‘amplification instrument,’ one that speeds up what an unaided human could not do<sup>2</sup>

I believe that these philosophers’ claims are fundamentally correct. Computer simulations do represent novel ways of practicing science precisely because of their high speed processing and accumulation of information. However, the idea of computer simulations as ‘cognitive enhancers’ depends on the existence of the few features that computers can offer as physical instruments, such as speed, memory, or computational power. If the epistemic power of computer simulations is analyzed in this way, as many philosophers have,<sup>3</sup> I believe we are missing a more interesting characteristic of computer simulations that exhibits their true epistemic power, namely, their capacity to successfully investigate the behavior of a host of target systems (empirical or otherwise). My proposal, then, is to relate the epistemic power of computer simulations to specific activities that exploit their capacity as devices that are analytic and inquisitive about our empirical world, such as explaining phenomena, predicting future states of the target system, or offering evidence for scientific

hypothesis.

A central issue arising here is the notion of *computer simulation*. Instead of envisaging computer simulations as ‘cognitive enhancers,’ then, I take them as instruments for studying the *patterns of behavior* of a target system. The shift in emphasis leads to a different way of approaching the philosophical study of computer simulations. Specifically, instead of focusing on their mechanistic features as the means of studying their epistemic power, I focus the analysis on the kind of scientific activities that a computer simulation can perform. In this vein, to simulate the dynamics of a biological system is, to me, to simulate the patterns of behavior of a predator-prey system in order to explain or predict some behavior. Following Philip Kitcher, I take the notion of *patterns* as one which reflects the structures, the performance, and the *behavior* of the target system.<sup>4</sup> The advantage of conceptualizing computer simulations in this way is that the physical features of the computer are no longer their primary epistemic virtue, but rather, it is their capacity to represent or describe *patterns of behavior* of the target system that entrenches computer simulations as epistemically powerful.<sup>5</sup>

Of course, this shift of emphasis is not meant to downplay the role of computer simulations as cognitive enhancers. On the contrary, computational power still makes the computation of scientific models possible, and it is at the core of the claim for the novelty of computer simulations in the sciences. But here the computational power of simulations is considered a second-level epistemic feature. In this sense, instead of locating the epistemic virtues of computer simulation in their capacity for crunching large amounts of data, these virtues come from describing patterns among systems from which we obtain understanding of the world. The challenge, then, is to show how and in what way computer simulations yield understanding of the world.

Given that the universe of simulations is significantly large, I propose to first narrow it down to *computer* simulations (Section 3.2). In this first analysis, I will be distinguishing *analogical* simulations from *digital* or *computer* simulations. There are two valuable lessons from this analysis: first, to establish that computer simulations are not causally related as analogical simulations are. This result is important for the discussion on scientific explanations. In particular, ontic theories of scientific explanation are excluded on grounds that computer simulations lack the necessary causal chain traceable in an explanation. Second, the analysis of analogical simulations and digital simulations provides the notion of ‘symbolic simulation’ as a two stage mapping affair. The first mapping relates the symbolic simulation to the states of the hardware, whereas the second mapping relates the symbolic simulation

to the representation of the target system. Such a conceptualization furnishes the intuition that a computer simulation is a ‘world of its own’ in the sense that its results are directly related to the simulation model that has been computed, but only indirectly to the empirical phenomenon.

Next, I will be addressing different notions of computer simulation in current literature. The purpose of this discussion is to set the grounds for my own working conceptualization, which I elaborate later in the chapter (see Section 3.3.1.1). Section 3.2.2 also includes a discussion on cellular automata, agent-based simulations, and complex systems as computer simulations that are excluded from this work. The reason for excluding them is based on the diverse ontological, methodological, and epistemic characteristics that each kind of simulation has to offer. For instance, an agent-based simulation consists in a set of rules that describe the emerging behavior of systems. These features can be used for different epistemic or methodological purposes, such as the dynamism of sociological systems, or for self-reproducing biological systems. Since these features are not relevant in equation-based simulations, my analysis on the epistemic power of computer simulations will not take this class of computer simulation into account. In Section 3.2.3 I narrow down the class of computer simulations to equation-based simulations, such as Ordinary Differential Equations, or Partial Differential Equations (henceforth ODE and PDE, respectively).

Finally, I discuss the methodology of computer simulations and present an example of a satellite under tidal stress which I will use for the remainder of this work as a paradigmatic example of the class of computer simulations of interest (Section 3.3).

## 3.2 On the notion of simulation

This section presents and discusses the broad use of the term *simulation* in the philosophical literature. Originally, this concept was reserved for special kinds of systems in which particular pieces of the world were being manipulated, such as in analog simulations. However, the introduction of computers into scientific life made this interpretation inappropriate and it therefore must be discussed accordingly.

In the following sections I discuss both analog simulations and digital simulations, followed by some considerations on current definitions of computer simulation. Finally, I discuss different classes of computer simulations with the intention to narrow down the class of equation-based simulations of relevance to this project.

### 3.2.1 The analog and the digital

The distinction between analog and digital is the first step in clarifying the notion of *simulation*. There are different views on how to make this distinction feasible. Let me begin with Nelson Goodman's notion of *representation of numbers*.<sup>6</sup> Goodman considers analog representation of numbers as *dense*. That is, for any two marks that are not copies, no matter how nearly indistinguishable they are, there could be a mark intermediate between them which is a copy of neither.<sup>7</sup> A digital representation, on the other hand, is 'differentiated' in the sense that, given a number-representing mark (for instance, an inscription, a vocal utterance, a pointer position, an electrical pulse), it is theoretically possible to determine exactly which other marks are copies of that mark, and to determine exactly which numbers that mark and its copies represent.<sup>8</sup> The general objection is that Goodman's representational distinction does not coincide with the analog-digital distinction made in ordinary technological language. In other words, Goodman's distinction between analog and digital fails to account for the notion of analog or digital simulation.

David Lewis, the main proponent of this objection, believes that "what distinguishes digital representation [from analog representation], properly so-called, is not merely the use of differentiated unidigital magnitudes; it is the use of the many combinations of values of a few -valued unidigital magnitudes" (Lewis, 1971, 326). Thus understood, analog representation is redefined as the representation of numbers by physical primitive magnitudes, where a primitive magnitude is defined as any physical magnitude expressed by a primitive term in the language of physics.<sup>9</sup> On the other hand, a digital representation is defined as the representation of numbers by differentiated multidigital magnitudes.<sup>10</sup> The problem with Lewis' definition turns out to be the same as Goodman's, namely, the representational viewpoint fails to explain the distinction between analog and digital computations.

It is Zenon Pylyshyn who shifts the focus from types of *representations* to types of *processes*. The motivation is that the notion of a process scores better for the distinction between analog and digital than the mere notion of representation does. Pylyshyn objects that Lewis' criterion allow magnitudes to be represented in an analog manner without the process itself qualifying as an analog process. He proposes the following example to illustrate his point:

Consider a digital computer that (perhaps by using a digital-to-analogue converter to convert each newly computed number to a voltage) represents all its intermediate results in the form of voltages and displays them on a voltmeter. Although this computer represents values, or numbers, analogically, clearly it operates digitally (Pylyshyn, 1984, 202).

The example shows that Lewis takes the modern computer as an analog process, a clearly misleading conceptualization. The basic problem is, again, to treat the distinction between analog and digital as types of *representations*. Pylyshyn's shifts turn out to be an essential move, not only for a more understandable distinction between analog and digital, but also for a more comprehensive definition of computer process. Allow me now to briefly discuss Pylyshyn's distinction between analog and digital processes.

The first thing to notice is that we can reconstruct Pylyshyn's notion of analog process using the notion of laboratory experiment discussed earlier. According to the author, "for [a] process, as opposed to the representation alone, to qualify as analogue, the value of the property doing the representing must play the right causal role in the functioning of the system" (Pylyshyn, 1984, 202). This notion of analog process is at the heart of the 'new experimentalism,' as I discussed in Section 1.3: in laboratory experimentation a single independent variable is manipulated in order to investigate the possible causal relationship with the target system. Experimentation, then, is about causal processes manipulating the empirical world, intervening on it, 'twisting the lion's tail.'<sup>11</sup> Having cleared this out of the way, allow me now to discuss Pylyshyn's notion of *computational process* (or *digital process*, as referred to previously).

A *computational process* is conceptualized as involving two levels of description: a *symbolic level* that refers to the algorithm and data structures, and a description of the *physical manipulation process* (i.e., the physical states of the machine). Pylyshyn carefully distinguishes between a symbolic level, which involves the abstract and formal aspects of the computational process, from the physical manipulation process that includes the physical states into which the computer enters when running the algorithm. In plain words the first part of the computational process is entirely symbolic, whereas the second is physical and mechanical.<sup>12</sup> This distinction is extremely useful for conceptualizing computer simulations, for it accounts for the lack of causal relationships acting in the simulation (i.e., a computer process cannot be described in terms of causal relationships), and it also depicts the existing internal mappings between the symbolic level and the physical manipulation process (i.e., the computer process is an internal affair between the representation - by means of algorithms and data structures - and the physical states of the machine). Unfortunately, Pylyshyn does not delve into these notions, nor into the way these two levels interact. Russell Trenholme, however, takes a similar view as Pylyshyn.

Trenholme distinguishes between *analog simulations*, characterized by parallel causal-structures isomorphic to the phenomenon simulated,<sup>13</sup> and *symbolic simula-*

tions that is, symbolic processes that adequately describe some aspect of the world.<sup>14</sup> An analog simulation, then, is “defined as a single mapping from causal relations among elements of the simulation to causal relations among elements of the simulated phenomenon” (Trenholme, 1994, 119). According to this conception, analog simulations provide causal information about the represented aspects of the physical processes being simulated. As Trenholme puts it, “[the] internal processes possess a causal structure isomorphic to that of the phenomena simulated, and their role as simulators may be described without bringing in intentional concepts” (Trenholme, 1994, 118). This lack of intentional concepts means that an analog simulation does not require an epistemic agent conceptualizing the fundamental structures of the phenomena, as a symbolic simulation does.<sup>15</sup>

The notion of *symbolic simulation*, on the other hand, is a bit more complex for it requires a further constituent; namely, the notion of *symbolic process*. A symbolic process is defined as a mapping from syntactic relations among symbols of the simulated model onto causal relations of elements of the hardware.<sup>16</sup> A *symbolic simulation*, therefore, is defined as “a two-stage affair: first, the mapping of inference structure of the theory onto hardware states which defines symbolic [process]; second, the mapping of inference structure of the theory onto extra computational phenomena” (Trenholme, 1994, 119).<sup>17</sup> In simpler words, in a symbolic simulation there are two acting mappings, one at the level of the symbolic process, which maps the simulation model onto the physical states of the computer, and another that maps the simulation model onto the representation of an exogenous computational phenomenon. The central point here is that the results of computing a simulation model are dissociated from the target system insofar as the former only depend on the theory or the model implemented.

This conclusion proves to be very important for my definition of computer simulation. In particular, Trenholme’s studies on symbolic simulation make possible the idea of a computer simulation as a ‘world of its own’; that is, as a system whose results are directly related to the simulation model and the computational process,<sup>18</sup> and only indirectly to the target system. This is simply a metaphor comparable with the mathematician that manipulates mathematical entities regardless of their ontological reality in the empirical world. However, it should be clear that I am not suggesting that computer simulations are unrelated systems, for many simulations are of an empirical system, and therefore it is a desirable feature to be able to relate the results of the simulation to the target system. Instead, the metaphor aims to highlight the fact that, in the context of computer simulations, total independence from the represented target system can be imposed by means of assumptions. Of

course, it is also possible to implement some parts of the simulation as having genuine realistic content, but this may not necessarily be the case. What ultimately determines these proportions is our epistemic commitment to the target system and the way we represent it in the simulation model.<sup>19</sup>

With these ideas firmly in mind, let me now turn to discuss notions of computer simulation in current literature.<sup>20</sup>

In 1990, Humphreys published his first work on computer simulations. There, he maintains that scientific progress needs, and is driven by, tractable mathematics.<sup>21</sup> By *mathematical tractability*, Humphreys refers to pen-and-paper mathematics, that is, the kind of calculation that an average person is able to carry out without the aid of any computing device.<sup>22</sup> In simpler words, mathematical tractability here requires a scientific model ‘to be analytically solvable.’<sup>23</sup> In this context, computer simulations become central contributors to the (new) history of scientific progress, for they “turn analytically intractable problems into ones that are computationally tractable” (Humphreys, 1991, 501). Computer simulations, then, amend what analytic methods could not undertake, that is, to find (approximate) solutions to equations by means of reliable (and fast) calculation. It is in this context that Humphreys offers the following *working definition*:

*Working Definition:* A computer simulation is any computer-implemented method for exploring the properties of mathematical models where analytic methods are unavailable (Humphreys, 1991, 501).

There are two ideas in this working definition worth underlining. The first one has been already discussed; that is, that computer simulations provide solutions to mathematical models where analytic methods are unsuccessful. A follow up comment is that Humphreys is careful in making clear that his working definition should not be identified with numerical methods: whereas both computer simulations and numerical methods are interested in finding approximate solutions to equations, only the latter is related to numerical analysis.<sup>24</sup>

The second idea stemming from the working definition is the ‘exploratory’ capacity of computer simulations for finding the set of solutions of the mathematical model. This is certainly a major feature of computer simulations as well as a source for epistemic and methodological discussion. Computer simulations are not only rich in power for computing intractable mathematics, but also for exploring the mathematical limits of the models that they implement. Unfortunately, and to the best of my knowledge, there is little analysis of this topic.<sup>25</sup>

Despite having presented only a working definition, Humphreys received vigorous objections that virtually forced him to change his original viewpoint. One

of the main objections came from Stephan Hartmann who correctly objected that Humphreys' definition missed the dynamic nature of computer simulations. Hartmann, instead, offered his own definition:

Simulations are closely related to dynamic models. More concretely, a simulation results when the equations of the underlying dynamic model are solved. This model is designed to imitate the time-evolution of a real system. To put it another way, *a simulation imitates one process by another process*. In this definition, the term "process" refers solely to some object or system whose state changes in time. If the simulation is run on a computer, it is called *a computer simulation* (Hartmann, 1996, 83).

Philosophers have warmly welcomed Hartmann's definition. Recently, Wendy Parker has made explicit reference to it: "I characterize a simulation as a time-ordered sequence of states that serves as a representation of some other time-ordered sequence of states" (Parker, 2009, 486). Francesco Guala also follows Hartmann in distinguishing between *static* and *dynamic* models, time-evolution of a system, and the use of simulations for mathematically solving the implemented model.<sup>26</sup>

In spite of this acceptability, Hartmann's definition presents a few issues of its own. The overall assumption is that a computer simulation is the result of direct implementation of a dynamic model on the digital computer, as it follows from the first, second, and last sentences of the previous quotation. To Hartmann's mind, therefore, there is no conceptual difference between solving a dynamic model and running a computer simulation, for the latter is simply the implementation of the former on the digital computer. However, the diversity of methods and processes involved in the implementation of a dynamical model on the digital computer exceed any interpretation of 'direct implementation.' There is generalized agreement among philosophers regarding the importance of analyzing the methodology of computer simulations for their conceptualization.<sup>27</sup> A simple example will illustrate this point: consider the implementation of the Lotka-Volterra model of predator-prey as a computer simulation. Since such a mathematical model consists in a set of continuous equations that represent the dynamics of the population of a given species, then, in order to implement it as a computer simulation it must first be discretized via the many formal and non-formal techniques discussed in Chapter 2. But this is not the only modification that a scientific model typically suffers. Considerations on the computability of the model in terms of time and computational power (e.g., memory available, accuracy of results, etc.) are also a major source of changes in the original scientific model. For instance, changing from the cartesian system of coordinates to the polar system of coordinates suggests that there is 'no direct implementation' as Hartmann indicates. In addition, a simulation is defined in terms

of a dynamic model which, when implemented on the digital computer, is conceived as a computer simulation. To put the same idea in a slightly different way: a dynamic model (implemented on the computer) becomes a computer simulation while a computer simulation is the dynamic model (when implemented on the computer). The notion of computer simulation, then, is simplified by defining it in terms of a dynamic model running on a special digital device.

Overall, Hartmann believes that computer simulations play a central function in the context of discovery. To this end, simulations are used as a technique, as heuristic tools, as substitutes for an experiment, and as tools for experimentalists.<sup>28</sup> Is it possible for computer simulations to also play a central function in the context of justification? I believe this is not only a possibility, but a highly desirable aim for the studies on the epistemology of computer simulations.

After Hartmann's initial objections, Humphreys coined a new definition, this time based on the notion of *computational template*.<sup>29</sup> Briefly, a computational template is the result of a computationally tractable theoretical template. A *theoretical template*, in turn, is the kind of very general mathematical descriptions that can be found in a scientific work. This includes partial differential equations, such as elliptic (e.g., Laplace's equation), parabolic (e.g., the diffusion equation), and hyperbolic (e.g., the wave equation), ordinary differential equations, among others.<sup>30</sup>

An illuminating example of a theoretical template is Newton's Second Law: "[it] describes a very general constraint on the relationship between any force, mass, and acceleration" (Humphreys, 2004, 60). These theoretical templates need to be specified in some particular respects, for instance, in the force function: it could either be a gravitational force, an electrostatic force, a magnetic force, or any other variety of force. Finally, "if the resulting, more specific, equation form is computationally tractable, then we have arrived at a computational template" (Humphreys, 2004, 60-61). Arguably, this is less a definition than it is a characterization of the notion of computer simulation. In any case, this is, together with Hartmann's, the most accepted conceptualization among current philosophical literature.

Let me finish this section with an illuminating classification of the term *computer simulation* as elaborated by Roman Frigg and Julian Reiss. According to the authors, there are two senses in which the notion of *computer simulation* is defined in current literature:

In the *narrow* sense, 'simulation' refers to the use of a computer to solve an equation that we cannot solve analytically, or more generally to explore mathematical properties of equations where analytical methods fail (e.g., Humphreys 1991, p. 501; 2004, p. 49; Winsberg 1999, p. 275; 2001, p. 444).

In the *broad* sense, 'simulation' refers to the entire process of constructing, us-

ing, and justifying a model that involves analytically intractable mathematics (e.g., Winsberg 2001, p. 443; 2003, p. 105; Humphreys 1991, p. 501; 2004, p. 107). Following Humphreys (2004, p. 102-104), we call such a model a ‘computational model’. (Frigg and Reiss, 2009, 596)

Both categories are certainly meritorious and illuminating. Both capture the two senses in which philosophers define the notion of *computer simulation*. While the narrow sense focuses on the heuristic capacity of computer simulations, the broad sense emphasizes the methodological, epistemological, and pragmatic aspects of computer simulations.

The challenge now is to find a way to address the philosophical study of computer simulations as a whole, that is, as the combination of the narrow and broad sense as given above. Such a study would emphasize, on the one hand, the representational capacity of computer simulations and their power to enhance our cognitive abilities, while on the other provide an epistemic assessment of their centrality (and uniqueness) in the scientific enterprise. In order to achieve this, I must first set apart the class of computer simulations that are of no interest from those relevant for this study. Let me now begin with this task.

### **3.2.2 Cellular automata, agent-based simulations, and complex systems**

The universe of so-called computer simulations is significantly large. Any abstract characterization that attempts to cover all cases will necessarily fail. The challenge in this and the following section is to discuss the class of simulations of interest, which will be narrowed down to equation-based simulations. In order to achieve this aim, I analyze different classes of computer simulations based on their methodology, scientific domain of applicability, and epistemic power. The importance in carrying out this distinction is to make sure that this study is restricted to the class of simulations of interest, namely, equation-based computer simulations. As I discuss here, different classes of computer simulations provide distinctive features and characteristics, not necessarily applicable to my study on explanation. Under this premise, let me now give reasons for excluding cellular automata, agent-based simulations, and complex systems from the class of computer simulations of interest in this work.

Cellular automata are the first example of a computer simulation that is out of the scope of this work. In the 1940s, Stanislaw Ulam was studying the growth of crystals using a simple lattice network as a model; at the same time, John von

Neumann was working on the problem of self-replicating systems, experiencing great difficulty in finding good results. The story goes that Ulam suggested von Neumann use the same kind of lattice network as his, creating in this way a two-dimensional, self-replicator algorithm. These were the humble beginnings of cellular automata.

Cellular automata are extremely simple: they are abstract mathematical systems in which space and time are considered to be discrete. They consist of a regular grid of cells, each of which can be in any number of states at any given time. Typically, all the cells are governed by the same rule, which describes how the state of a cell at a given time is determined by the states of itself and its neighbors at the preceding moment. Stephen Wolfram defines them as:

Mathematical models for complex natural systems containing large numbers of simple identical components with local interactions. They consist of a lattice of sites, each with a finite set of possible values. The value of the sites evolve synchronously in discrete time steps according to identical rules. The value of a particular site is determined by the previous values of a neighborhood of sites around it. (Wolfram, 1984b, 1)

Although this is a general definition, it already suggests the possibility of complex natural systems modeled by cellular automata, instead of the traditional PDE. To Wolfram's mind, cellular automata are more adaptable and structurally similar to empirical phenomena than PDE.<sup>31</sup> In a similar vein, Gérard Vichniac believes that cellular automata not only seek numerical agreement with a physical system, but also they attempt to match the simulated system's own structure, its topology, its symmetries and its 'deep' properties.<sup>32</sup> Similarly, Tommaso Toffoli entitled a paper: *Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics* (Toffoli, 1984), highlighting cellular automata as the natural replacement of differential equations in physics.<sup>33</sup> Unfortunately, the meta-physical assumptions behind cellular automata are far from settled. It is not clear, for instance, that the natural world is actually a discretized system, as opposed to a continuous one, like today's physics describes it, and therefore whether discrete cells accurately represent it.<sup>34</sup>

On less speculative grounds, it is a fact that cellular automata have little presence in the natural sciences today.<sup>35</sup> The reason for their absence is mostly cultural since the physical sciences are still the accepted models for the natural world as we know it (the widespread use of PDE and ODE is the proof of this). Moreover, even for systems of equations that model the empirical world as discrete, cellular automata have little, if any, presence. Despite Wolfram's efforts to show that the world might be more adequately represented by a discrete point of view, the natural sciences have not made a complete shift yet.

Another reason why I exclude cellular automata from this study is because they differ from equation-based simulations in terms of solving the underlying model. While cellular automata provide exact results of the models implemented,<sup>36</sup> equation-based simulations experience all sorts of errors in their results: round-off errors, truncation errors, transformations errors, and so forth (see my study on errors in Section 3.3.3). Moreover, since approximations are almost nonexistent for cellular automata, any disagreement between the model and the empirical data can be ascribed directly to the model which realized the theory, instead of the cellular automata itself. This is clearly not the case for equation-based simulations (see my study on verification and validation in Section 3.3.2). As Evelyn Fox-Keller points out,

[cellular automata are] employed to model phenomena that lack a theoretical underpinning in any sense of the term familiar to physics (phenomena for which no equation, either exact or approximate, exists [...] or for which the equations that do exist simply fall short) [...] Here, what is to be simulated is neither a well-established set of differential equations [...] nor the fundamental physical constituents (or particles) of the system [...] but rather the phenomenon itself. (Keller, 2003, 208)

The above discussion suggests that cellular automata require a methodology and an epistemology of their own, different from the ones needed for equation-based simulations. For these reasons, and for keeping this study as focused as possible on one class of simulation, I exclude cellular automata from the scope of this work. However, it must be clear that I am not denying the possibility that this work on explanation is also applicable to cellular automata, rather I am only noticing that the existing methodological and epistemological differences from equation-based simulations call for a proper analysis, one that I will not pursue here.

In a similar vein, agent-based simulations and complex systems will be set outside the scope of this study. Here, agent-based simulations are understood as models for simulating the actions and interactions of multiple autonomous programmed agents that re-create and predict the behavior of large systems such as societies, biological systems, and the like. The notion of ‘complex system,’ on the other hand, covers different activities, such as game theory, evolution and adaptation, systems biology, etc. Both classes of simulation share the property of investigating how the total behavior of a system emerges from the collective interaction of the parts of the simulation. Also, the total behavior of these simulations is sensitively determined by their initial and boundary conditions, leading to the representation of unpredictable behaviors of natural systems. To deconstruct these simulations to their constituent

elements would remove the added value that has been provided in the first place by the computation of the set of rules and equations. It is a fundamental characteristic of these simulations, then, that the interplay of the various elements brings about a unique behavior of the entire system. Thus understood, emergence presupposes epistemic opacity for it is impossible to predict the future states of the system.<sup>37</sup> In this way, agent-based simulations and complex systems also require a methodology and an epistemology of their own, just like cellular automata do.

Another reason for rejecting agent-based simulations, complex systems, and cellular automata altogether is that, despite isolated cases, different kinds of computer simulations are used in different scientific disciplines. Traditional physics, for instance, still relies strongly on continuous equations (PDE and ODE, for instance), which are mechanistic in essence; so do physics-related disciplines such as astronomy, chemistry, and engineering. Cellular automata, agent-based and complex systems are more prominent in social and biological sciences, where the behavior of agents is better described and analyzed. Naturally, there are overlapping uses of computer simulations in the sciences: cellular automata have proven successful in traditional physics on phase transitions, replacing the use of PDE.<sup>38</sup> Conversely, the implementation of cellular automata by PDE is also under study.<sup>39</sup> More examples can be found in the use of PDE in population biology, as the Lotka-Volterra predator-prey model shows.<sup>40</sup>

Perhaps the best reason for excluding cellular automata, agent-based, and complex system from this study stems from the minimal requirements needed for a successful explanation by computer simulations. In other words, in equation-based simulations the explanans can be reconstructed directly from the simulation model since its computation does not add value to the results. In any of the other classes of computer simulations, the interplay of the various elements during the computation must be considered as part of the explanans, for they are part of the success of an explanation. To analyze exactly the leverage that such an interplay (emergent behavior, etc.) has in the explanation of data, and how this affects and influences the construction of the explanans, is a matter that exceeds the limits of this work. All that it is possible to say is that one could not expect a successful explanation when epistemic and methodological specificities of the simulations are left out. This exclusion does not entail, however, that a proper understanding of these computer simulations could bring together all classes of simulations under the same study on scientific explanation.

Let me now discuss equation-based simulations in more detail.

### 3.2.3 Equation-based simulations

The class of computer simulations of interest has been narrowed down to those simulations that implement equation-based models. As claimed at the beginning of the chapter, these models represent patterns of behavior of the target system of interest. In the following section, I discuss a preliminary taxonomy whose aim is to be as specific as possible about the class of equation-based simulations that this work includes. The reason this is a preliminary taxonomy is that it is intended to be neither exhaustive nor complete, but only a guidance of the universe of computer simulations that I cover in this work.

I organize this taxonomy into two categories; namely, *target systems* and *methods of calculating*. The first category refers to the diversity of target systems simulated in current scientific practice. The second category refers to the diverse methods for solving the simulation model. These two categories are not disjointed, for one simulation might implement different methods of resolution, whereas different simulations might solve their equations with the same method.

The taxonomy based on the *target system*, is the following:

1. *Empirical systems*: these are empirical phenomena represented in a theoretical model. Such models implement laws, principles and theories accepted by the community. A Newtonian model of the planetary movement is a good example of such an empirical system. It represents the behavior of two bodies interacting with each other by a handful of equations. So-called *numerical experiments* or *computer experiments* will be the kind of computer simulation that implements these models.

There is a host of theoretical models, some of which represent more accurately the target system than others. However, any representational difference must only be reflected in the analysis of their epistemic power, not in the ontological class to which they belong. To illustrate this point, take for instance a Newtonian model and climate model. The former is more accurate than the latter insofar as the estimated values after solving the Newtonian model are more accurate than those after solving a climate model.<sup>41</sup> However, both represent a target system by means of a set of equations, both are implemented in similar ways, and both produce a set of results by computing the simulation model. Hence, both belong to the class of equation-based computer simulations that represent an empirical system.

Practical and conceptual considerations about the representation of empirical systems are the subject of a philosophy of modeling, and not of my concern here. Rather, I am only interested in the ways to implement a scientific model

into the computer. Hence, I will assume that a computer simulation represents an empirical system to the same extent as the scientific model represents the same empirical system. Although a bit controversial, this assumption tries to avoid general skepticism regarding the representational capacity of the simulation model (for a more detailed discussion on this point, see Section 1.2.2).

2. *Theoretical systems*: these systems refer to a hypothetical world, that is, one where no empirical phenomenon is represented. Rather, an abstract, possibly conjectural world is presented. Take as examples the famous problem of the Seven Bridges of Königsberg, or the traveling salesman problem. In this way, the target system is not an exogenous empirical system, but has the properties of a mathematical system. A computer simulation implementing these models is mostly theoretical in essence, and is designed for exploring the underlying theoretical properties of the model. These systems are known as *theoretical simulations*. These simulations are not limited to mathematics, and are also common in scientific practice. They are heavily used in theoretical physics, theoretical chemistry, and astronomy, just to mention a few theoretical disciplines.<sup>42</sup>
3. *Data systems*: much of scientific practice is about collecting large amounts of data, and creating *phenomenological models* or *models of data*. The particularity of these models is their lack of all theoretical underpinning. They are not purely theoretical, for they stand for an empirical system (e.g., data can also come from another simulation), nor are they purely empirical, for they lack laws and principles represented in the model. A standard definition takes phenomenological models as “models that represent only observable properties of their targets and refrain from postulating hidden mechanisms and the like. Alternatively one can define phenomenological models as being independent of general theories” (Frigg and Hartmann, 2006, 742). A model of data, on the other hand, “is a corrected, rectified, regimented, and in many instances idealized version of the data gained from immediate observation, the so-called raw data.” (Frigg and Hartmann, 2006, 743). Take for instance data collected by measurement of the Earth’s gravity field. The model that best fits these data is usually a density distribution, indicating something about the physical parameters just measured. These kinds of computer simulations are best known as *data simulations*.

The simulation model, then, aims at representing one of the aforementioned target systems. Depending on the problem and the availability of resources, one of these

following methods for solving the simulation model might apply: *analytical methods*, *numerical analysis* and *stochastic techniques*.

1. *Analytic methods*: this is the simplest method of all. It simply consists in carrying out the mathematical operations specified in the simulation model, in the same fashion as a mathematician would do using pen and paper. The efficacy of this method depends on whether the size of the ‘word’ in a computer is large enough for carrying out the mathematical operation.<sup>43</sup> If the operation exceeds its size, then round-off and truncation mechanisms intervene for the operation to be successful.
2. *Numerical analysis*: this method consists in solving the simulation model by approximation. Mathematical studies on numerical analysis predate the use of computers: linear interpolation, Newton’s method, Lagrange interpolation polynomial, Gaussian elimination, or Euler’s method are some of the methods used for solving PDE and ODE. These methods rely on a ‘divide and conquer’ strategy, by which an integral function on a large set is broken down into smaller sets and solved. Computer science elicits a whole host of new numerical methods envisaged for coping with specific problems.<sup>44</sup>
3. *Stochastic techniques*: For higher order dimensions, both analytic and numerical methods become prohibitively expensive in terms of computational effort. Stochastic techniques rely on methods that use pseudo-random numbers; that is, numbers generated by a numerical engine.<sup>45</sup> The most famous stochastic method is the Monte Carlo method, which is particularly useful for simulating systems with many coupled degrees of freedom, such as fluids, disordered materials, strongly coupled solids, and cellular structures.<sup>46</sup> Monte Carlo methods are based on mathematical laws that guarantee convergence to the desired result. The ‘law of large numbers,’ for instance, establishes that the average of the results obtained from a large number of trials must be close to the expected value.<sup>47</sup>

Thus understood, this taxonomy comprises the kinds of equation-based simulations that are important for the rest of this work. It also outlines the methods for reaching the solution of the models that are traditionally used by scientists. I must point out, however, that this taxonomy will be narrowed down further when I discuss theories of scientific explanation (see Chapter 4 and Chapter 5). The reason is that a successful explanation requires the results of the simulation to represent the phenomenon. But equation-based simulations, as discussed here, can also encompass heuristic simulations; that is, role simulations that explore fictional worlds. A good example

of this is the Brusselator simulation, which assumes a third-order chemical reaction that is highly unlikely to occur. Let me postpone any detail about explanation to the last chapters. Instead, allow me to present an example that illustrates the general methodology of computer simulations.

### 3.3 A methodology for computer simulations

In the previous section, I narrowed down the class of computer simulations of interest. William Oberkamp calls these simulations ‘computational engineering and physics,’ which includes computational fluid dynamics, computational solid mechanics, structural dynamics, shock wave physics, and computational chemistry, among others.<sup>48</sup> The intention of labelling them like this is to emphasize the double dependence of computer simulations: on the one hand, there is a dependence on traditional science, mathematics, and engineering; on the other hand, they rely on specific characteristics of computer architecture, such as additional mathematics (e.g., discretization algorithms and grid quality), and decision implementation issues (e.g., computer hardware, operating-system software, source-code reliability, design decisions, and the like).<sup>49</sup> In this section, I address in more detail this double dependence from a methodological perspective; I will also illustrate the class of computer simulations of interest with one example that, by implementing classical Newtonian mechanics, simulates the behavior of a two-body system.

Philosophers have largely acknowledged the importance of studying the methodology of computer simulations. Eric Winsberg says that,

[...] the credibility of a simulation model must come not only from the credentials supplied to it by its theoretical ancestors, but also from the antecedently established credentials of the model-building techniques employed in its construction. There are, in other words, model-building techniques that are taken, in and of themselves, to be suitable for building credible models (Winsberg, 2010, 120).

Morrison takes a similar stand on the topic:

[...] there are issues here about the degree of departure between the discretized simulation model and the mathematical model, that is, whether the former is an accurate representation of the latter. To some extent these issues are practical problems resolved via calibration and testing and hence are not reasons for general skepticism regarding the representational capacity of the simulation model itself (Morrison, 2009, 45).

I agree with Winsberg and Morrison in their interpretations. Winsberg is right in pointing out that the reliability of the computer simulation has diverse sources, and

cannot be limited to the theoretical model from which it originates. Morrison is right in dissuading us from casting doubts upon the representational capacity of the simulation model. A history of success representing target systems eschews any skepticism about the representational capacity of computer simulations.

There are several methodological aspects involved in the design and programming techniques of computer simulations. A shortlist includes:<sup>50</sup>

1. *Random number generators*: the selection of random number generation is an important design decision. Accuracy and precision of the results depend on this. The *loci classici* on this topic are (Knuth, 1973) and (Press et al., 2007);
2. *General methods of resolution*: the idea of breaking down the complex system into a set of simpler interacting sub-systems is a powerful one. For instance, many computer simulations depend on the time-evolution of the system. It is then possible to develop the history of the simulation system through a stepwise sequential advance of each sub-system, an approach known as *time-splitting*;
3. *Discretization methods*: different discretization methods modify the representational structure of the original scientific model, turning it into a discrete model. Examples of discretization methods are the Runge-Kutta method and the Euler method. Special care must be taken in order to ensure a good description of the original scientific model;
4. *Expert knowledge and scientific expectations*: the design and programming of a computer simulation do not depend solely on mathematical machinery, but also on the expert knowledge of scientists. Such knowledge includes ‘tricks of the trade,’ ‘know-how,’ and ‘past experiences.’

Two coexisting methodologies emerge from this shortlist: on the one hand, there are *formal rules and systematic methods* that allow formal equivalency between the scientific model and the simulation model. Take for instance formal specification languages, such as the Z notation, VDM, or LePUS3; or the discretization methods mentioned in the list.<sup>51</sup> On the other hand, there are informal routines guiding general design decisions such as expert knowledge, know-how, and tricks of the trade.

These two approaches are present in the design and programming of every computer simulation. Indeed, to neglect informal routines might lead to the wrong idea that there is an axiomatic method for designing computer simulations; furthermore, failing to acknowledge the presence of systematic and formal methods might lead to a view of computer simulations as an unstructured, ungrounded discipline. The challenge, then, is to understand the role that each one plays in the complex activity that is the methodology of computer simulations.

It is also true that, despite the many ways in which computer simulations are designed and programmed, there is a standardized methodology that provides reliability to the instrument. It is hard to imagine a team of scientists and engineers changing their methodologies every time they design and program a new computer simulation. The methodology for computer simulations, then, relies on ideals of stability of behavior, reliability of functionality, and continuity in design and programming. The software industry is a good example of this since it follows (and creates) protocols that standardize the production of software. In this sense, functions, procedures, data structures, and dynamic link libraries (DLLs), are all designed and programmed for a host of systems, regardless of the different computational architectures, programming languages, or software packages available.<sup>52</sup>

### 3.3.1 An example of a two-body system

Let me now introduce a concrete example. Let us suppose that a government agency wants to put a new satellite into orbit. The safest and cheapest way of doing this is by running a computer simulation, observing the estimated behavior of the satellite, and building the real satellite following the specifications of the computer simulation. Despite its simplicity, the importance of the computer simulation is that it answers important questions about the target system. For instance, by running a simulation, scientists know what the expected trajectory of the satellite is given: a) the mass of the satellite, b) the mass of the planet, and c) the existing tidal stress between the planet and the satellite. Depending on the values of a), b), and c), the satellite might successfully orbit for a long time or crash into the planet. The cognitive insight that a computer simulation might provide about the target system is not a function of the simplicity of the problem, but rather an inherent virtue of these kinds of systems.

To illustrate this case, I make use of the example of an orbiting satellite under tidal stress as elaborated by Michael Woolfson and Geoffrey Pert.

From classical mechanics, we know the following facts:

A satellite in orbit around a planet is subjected to a tidal stress which stretches it along the direction of the radius vector. If the orbit is non-circular then the stress is variable and the satellite expands and contracts along the radius vector in a periodic fashion. Since the satellite will not be perfectly elastic there will be hysteresis effects and some of the mechanical energy will be converted into heat which is radiated away. The overall effect is that while the system as a whole is losing mechanical energy it must conserve angular momentum (Woolfson and Pert, 1999, 17-18).

The following set of equations are used for the scientific model:

For a planet of mass  $M$  and a satellite of mass  $m$  ( $\ll M$ ), in an orbit of semi-major axis  $a$  and eccentricity  $e$ , the total energy is

$$E = -\frac{GMm}{2a} \quad (3.1)$$

and the angular momentum is

$$H = \{GMa(1 - e^2)\}m \quad (3.2)$$

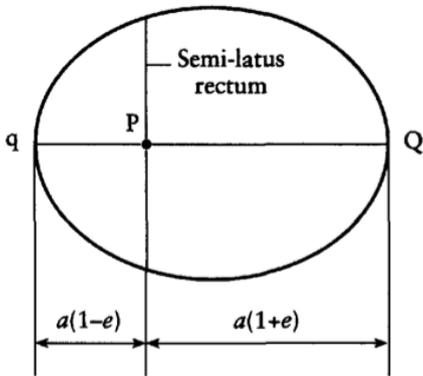


Figure 3.1: The elliptical orbit of a satellite relative to the planet at one focus. Points  $q$  and  $Q$  are the nearest and furthest points from the planet, respectively. (Woolfson and Pert, 1999, 19)

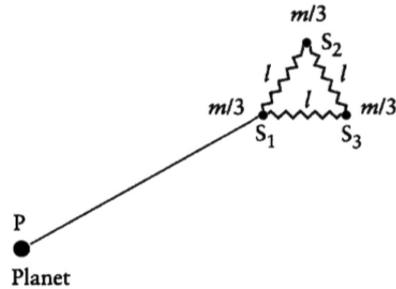


Figure 3.2: The satellite is represented by three masses, each  $m/3$ , connected by three springs each of the same unstrained length,  $l$ . (Woolfson and Pert, 1999, 19)

If  $E$  is to decrease, then  $a$  must become smaller; but if  $H$  is constant, then  $e$  must become smaller (that is to say, that the orbit must round off).

The quantity which remains constant is  $a(1 - e^2)$ , the *semi-latus rectum*, which is indicated in Figure 3.1. The model we shall use to simulate this situation is shown in Figure 3.2. The planet is represented by a point mass,  $P$ , and the satellite by a distribution of three masses, each  $m/3$ , at positions  $S_1, S_2$  and  $S_3$ , forming an equilateral triangle when free of stress. The masses are connected, as shown, by springs, each of unstressed length  $l$  and the same spring constant,  $k$ . Thus a spring constantly stretched to a length  $l'$  will exert an inward force

$$F = k(l' - l) \quad (3.3)$$

Now, we also introduce a dissipative element into our system by making the force

dependent on the rate of expansion or contraction of the spring, giving the following force law:

$$F = k(l' - l) - c \frac{dl'}{dt} \quad (3.4)$$

where the force acts inwards at the two ends. It is the second term in Equation 3.4 which gives the simulation of the hysteresis losses in the satellite.

Thus described, this model represents the pattern of behavior of *all* bodies (being satellites or otherwise) under tidal stress. The challenge now is to transform this model into a specification for a computer simulation. For this, there are two steps involved: the first step is to integrate this model into a *specification*; the second step is to code the specification into an *algorithm*.

At the level of the specification, there are formal and non-formal techniques for transforming the model. As I stated in Section 2.2, formal transformations are carried out by using ‘specification languages,’ such as the *Z notation*, where the researcher can specify instructions, objects, types, equations, and so forth, for programming the algorithm. In addition to the formal machinery, non-formal and very effective techniques are used as well: tricks-of-the-trade, rules of thumb rooted in practitioners’ skills and abilities, and expert knowledge obtained by interdisciplinary work are just a few techniques used in the design of any specification.

The *algorithm*, on the other hand, must code the specification for machine interpretation. For this, the programming of an algorithm is also partially guided by formal as well as non-formal methods. On the formal side, there is a host of languages specialized that ‘mediate’ between a specification and an algorithm, such as the Common Algebraic Specification Language (CASL), and the Vienna Development Method (VDM). Model checking is conceived as the branch in computer science that tests automatically whether an algorithm meets the required specification. On the non-formal side, the algorithm is exposed to similar techniques as the specification; namely, tricks-of-the-trade, know-how, and the like, all of which increase the confidence of the algorithm correctly coding the specification.

Thus described, the transformations of a theoretical model into a computer specification, and the codification of the latter into an algorithm is a central aspect of the methodology of computer simulations. In addition, the algorithm must be implemented on the physical computer in order to render results (i.e., the *computer process* (see Section 2.4)). Such an implementation is only possible when the values in the initial and boundary conditions are filled out. Woolfson and Pert simulate with the following values:<sup>53</sup>

number of bodies = 2  
mass of the first body (planet) =  $2 \times 10^{24}$  kg  
mass of satellite =  $3 \times 10^{22}$  kg  
initial time step = 10 s  
total simulation time = 125000 s  
body chosen as origin = 1  
tolerance = 100 m  
initial distance of satellite =  $1 \times 10^8$  m  
unstretched length of spring =  $1 \times 10^6$  m  
initial eccentricity = 0.6 (Woolfson and Pert, 1999, 20)

These values are chosen to give a reasonable computational time, initial distance from the planet, initial eccentricity, and so forth. In this vein, they single out one particular simulation and, therefore, one particular phenomenon. The interest for simulating this particular orbiting satellite lies, in part, in studying its behavior under certain conditions. For instance, one might want to see what happens if the mass of the planet increases. The simulation indicates that if the mass of the planet is similar to (or larger than) Jupiter's (i.e.,  $1.8986 \times 10^{27}$  kg), the satellite would collapse into it. Now, in the case that I described above, there is one interesting phenomenon that the simulation shows: using the set of initial and boundary conditions, the following spikes appear in the satellite's orbital eccentricity (Figure 3.3). The question, then, is why?

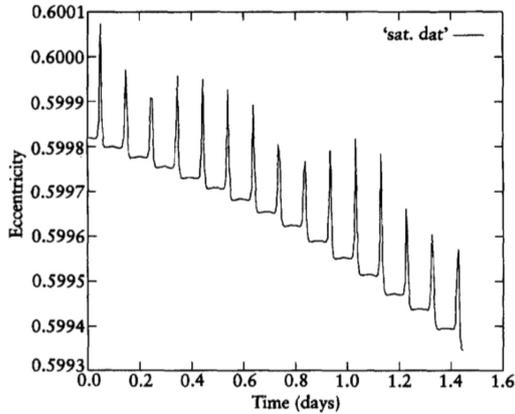


Figure 3.3: The orbital eccentricity as a function of time. (Woolfson and Pert, 1999, 20)

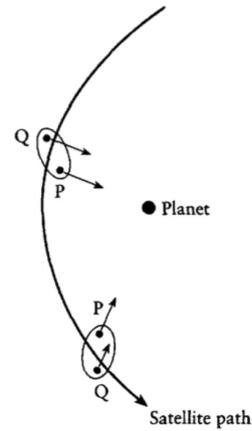


Figure 3.4: The lag and lead in the tidal bulge of the satellite give spin angular momentum on approach and subtract it on recession. (Woolfson and Pert, 1999, 21)

### 3.3.1.1 A working conceptualization

The example of the satellite under tidal stress reveals some important facts about computer simulations, namely,

1. The specification and the algorithm provide cognitive access to the computer simulation. Modifications of any kind, as well as general understanding of the simulation, are only attainable through them.
2. The specification and the algorithm represent the behavior of all phenomena that fall within their domain. In the example above, they represent the behavior of all two-body systems under tidal stress. Due to this representational capacity, I will refer to them jointly as the *simulation model*. A *simulation model* must be conceived of as a pen-and-paper unity, and in this sense must be capable of formal verification and validation. I reserve the term *general computer simulation* for the simulation model which has been implemented on the physical computer and whose initial and boundary conditions have not yet been set. This distinction becomes more relevant in Chapter 5.
3. When the values of the initial and boundary conditions of a general computer simulation are filled out, one computer simulation is instantiated and, therefore, one phenomenon is singled out. I call this instantiated simulation the

*particular computer simulation*. In the example above, the phenomenon singled out is a satellite whose orbital eccentricity describes the spikes in Figure 3.3. Only particular computer simulations can be implemented on the physical machine for rendering results.

With these facts in mind, I postulate the following working conceptualization of a computer simulation:

A computer simulation consists of two elements: a *general computer simulation*, which is the implementation of a simulation model on a digital computer ready to be executed. The simulation model represents the general patterns of behavior of a target system, which can be ‘empirical,’ ‘theoretical,’ or ‘data systems.’ A general computer simulation produces results when its initial and boundary conditions are filled out, creating in this way a *particular computer simulation*. A particular computer simulation produces results by reckoning using stepwise ‘analytic methods,’ ‘numerical analysis,’ or ‘stochastic techniques.’<sup>54</sup>

Unlike other philosophical literature, I do not need to postulate a working *definition* for computer simulations since the above conceptualization is enough for showing their epistemic power.

Let me say that this working conceptualization is essentially an analytic step towards the general notion of ‘computer simulation,’ ‘computer simulation results,’ and similar terminology found in current literature, only necessary for my discussion of explanation. Thus, I see no need to change the terminology as long as the context is clear. In this vein, and unless the context requires, I will continue making use of the same terminology as found in current literature. It is not until Chapter 5, when I address the epistemic power of computer simulations, that I need these new terms. The reason for introducing new terminology lies in the differences between a *simulation model* and a *general computer simulation* which must be pointed out, such as the fact that the former is receptive to formal verification and validation, whereas the latter is not. Similarly, the notion of *particular computer simulation* emphasizes the fact that the initial and boundary conditions of the general computer simulation have been fulfilled, singling out one concrete simulation.

### 3.3.2 Verification and validation

*Verification* and *validation*<sup>55</sup> are the names given to a host of methods used for increasing the confidence in models and simulations. Understanding their role in the design of computer simulations turns out to be essential for the assessment, credibility and power to elicit their results.

In order to measure the correctness of a model one must have either accurate benchmarks or reference values against which to compare the model, or formal methods for confirming the implementation of the model. Benchmarking, according to William Oberkampf and Christopher Roy, is the technique used for measuring the performance of a system, frequently by running a number of standard tests and trials against it. In *validation*, this is fairly straightforward since benchmarkings are high-quality experimental measurements of systems responsible for the quantities of interest.<sup>56</sup> In *verification*, however, formal methods are the focus for confirming the correct implementation of the model as a computer simulation. In validation, then, the relationship between computation and the empirical world (e.g., experimental data obtained by measuring and observing methods) is the issue,<sup>57</sup> whereas in verification, the relationship of the simulation to the empirical world is not necessarily at stake.

Now, this way of depicting verification and validation is simply too general. The scientific and computational communities offer a diversity of definitions, depending on the particularities of the systems under study.<sup>58</sup> The philosophical community, on the other hand, offers a rather sloppy and inexact account of verification and validation. Eric Winsberg, for instance, neglects the formal aspect of verification (see below: *code verification*). According to him “*Verification*, [...] is the process of determining whether or not the output of the simulation approximates the true solutions to the differential equations of the original model. *Validation*, on the other hand, is the process of determining whether or not the chosen model is a good enough representation of the real-world system for the purpose of the simulation” (Winsberg, 2010, 19-20). Morrison, alternatively, downplays the need for verification and claims that validation is the crucial method for assessing the confidence of the computer simulation results.<sup>59</sup>

Here following, I base my findings only on scientific literature (specifically, computer science). Here are two definitions that are largely accepted:

*Verification*: the process of determining that a computational model accurately represents the underlying mathematical model and its solution.

*Validation*: the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model. (Oberkampf et al., 2003)

Let us now discuss them separately.

### 3.3.2.1 Verification

*Verification* can be divided into two approaches: *code verification* and *calculation verification*. The purpose of making such a distinction is to categorize the set of methods for the assessment of the simulation model. Let us see them in turn.

*Code verification*: the process of determining that the numerical algorithms are correctly implemented in the computer code and of identifying errors in the software. (Oberkampf et al., 2003, 32)

*Code verification*, then, is a method guided by theoretical and deductive principles.<sup>60</sup> As mentioned in our previous discussion on formal methods, there is a host of methods for determining that the algorithm has been correctly implemented. A short list includes measured errors specified and implemented as part of the computer software, potential system halting, and accurate representation of the specification by the algorithm, among others.

On the other hand, *calculation verification*<sup>61</sup> is defined as the method that prevents three kinds of errors: human error in the preparation of the code, human error in the analysis of the results, and numerical errors resulting from computing the discretized solution of the simulation model. A definition for calculation verification is the following:

*Calculation verification*: the process of determining the correctness of the input data, the numerical accuracy of the solution obtained, and the correctness of the output data for a particular simulation. (Oberkampf et al., 2003, 34)

Calculation verification is the empirical side of verification. It is based on the comparison between the results of the simulation against highly accurate solutions of the scientific model. In a sense, calculation verification is similar to validation assessment insofar as both compare estimated results with correct results. It most commonly controls spatial and temporal convergence rates, iterative convergence, independence of solutions to coordinate transformations, and similar processes.<sup>62</sup>

### 3.3.2.2 Validation

The process of *validation* (also known as *testing*) presupposes an inductive way to demonstrate that the results of the simulation represent, in an accurate and precise way, the results of the implemented scientific model. William Oberkampf and Timothy Trucano highlight three key aspects of validation:

- i) quantification of the accuracy of the computational model by comparing its responses with experimentally measured responses,

- ii) interpolation or extrapolation of the computational model to conditions corresponding to the intended use of the model, and
- iii) determination if the estimated accuracy of the computational model, for the conditions of the intended use, satisfies the accuracy requirements specified. (Oberkampf and Trucano, 2008, 724)

The philosopher of science must expect that validation, as an inductive method, faces typical problems of induction. The general problem is that this method only allows validation of a model up to a certain point, and therefore complete validation is absolutely impossible due to the large number of comparisons needed (not to mention the improbability of having all the possible results at hand). Thus, validation is a method for *assisting* in the detection of errors, but not designed for *detecting* misrepresentations of the model.<sup>63</sup>

Besides, we need to keep in mind that validation depends on the capacity of comparing computer simulation results with empirical results; this dual relation between results requires the presence of both, excluding those computer simulations from the analysis for which there is no empirical data counterpart. In this sense, validation is only a suitable concept for those cases in which a computer simulation is representing an *actual* system, and not a *possible* or *conceivable* system.

Figure 3.5 shows in a flow diagram how verification (both code verification and calculation verification) and validation methods are put into effect on standard scientific practice.

The *conceptual model* here is the product of analyzing and observing the physical system of interest (i.e., what we called the *scientific model*). In key applications of computational physics (such as computational fluid dynamics, computational solid mechanics, structural dynamics, shock wave physics, and computational chemistry), the conceptual model is dominated by the PDEs used for representing physical quantities.

Two types of model can be identified: a *mathematical model*, from which the computational or simulation model is created, and a *physical model* which, for simplicity, we shall identify with an experiment. The *computational model*, finally, is an operational computer program that implements the conceptual model.

The figure also shows that *code verification* deals with the fidelity between the conceptual model and the mathematical model, whereas *calculation verification* deals with the agreement between the simulation results and the expected results of the computational model. *Validation*, on the other hand, is a quantitative adequacy between the solutions of the simulation and the experimental measurements or observations. This adequacy can be determined by a comparison that provides an acceptable level of agreement between the solutions of the two models involved.<sup>64</sup>

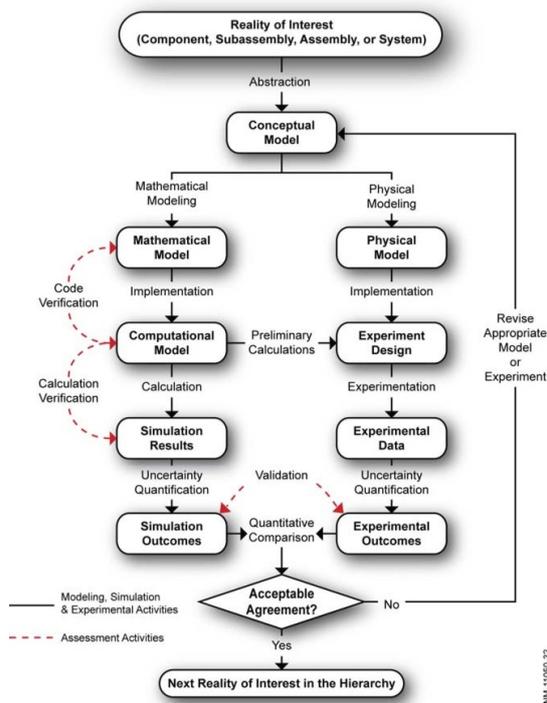


Figure 3.5: Phases of modeling and simulation, and the role of verification and validation (ASME, 2006, 5).

Verifications and validations are, together with the host of methods discussed in Section 3.3, two fundamental methods for asserting the reliability of results of computer simulations. As we shall argue in the next chapter, reliability of results supports the claim that we ‘know the results of the simulation,’ i.e., they are correct solutions of the model (Chapter 4).

A final comment on verification and validation: although neither software nor hardware can be fully verified or validated, researchers are still developing methods that reduce the possibility of errors and increases the credibility of the model. This interest from the scientific community can only mean that both methods are necessary and required for dependable computer simulations. The literature here mentioned can work as a starting point for future studies. Our sole purpose was to briefly discuss verification and validation as part of the general methodology of computer simulations, the subject of the present chapter.

### 3.3.3 A brief survey on errors in computers

*Errors* are a time-honored problem in laboratory practice as well as in computer science. Our intention here is simply to depict a general overview of errors in computers.

Errors are the origin of much discontent and skepticism among philosophers. Parker, for instance, discusses a few examples of errors related to the mathematical form of continuous model equations (e.g., due to simplifications and idealizations), error due to instability in the chosen solution method, as well as error due to programming mistakes.<sup>65</sup> Parker finds inspiration in Allan Franklin’s ‘Sherlock Holmes’ strategy,’ which is based on eliminating all plausible sources of error and alternative explanations in order to find a valid experimental result.<sup>66</sup> Although essentially cor-

rect, Parker seems to be conflating the diverse sources of errors in computer science into one. Let me elaborate more on the source of errors in computer science.

An initial classification divides errors into two groups: there are errors that are random, such as a voltage dip during computation, or a careless lab member tripping over the power cord, and therefore not relevant for philosophical analysis; and there are errors that are systematic within the computation. The study of systematic errors can, in turn, be divided into three kinds:

1. *Physical errors*: they are related to the malfunctioning of the computer microprocessor, computer bus, memory and, in general, any physical component of the computer.

To illustrate this type of error, consider the following algorithm:

---

---

```
{...
if (false) then {printout "The Democrats have won"}
    else {printout "The Republicans have won"}
...}
```

---

The logical conditional here is represented by one bit in the physical computer. This means that there is one physical value, measured in volts, between printing out ‘The Republicans have won’ instead of ‘The Democrats have won.’ Arguably, this is just an unfortunate coincidence, for the real possibilities of this error happening are extremely small. However, this simple example shows the dimension of the problem when computer hardware is faulty.

Similarly, truncation errors and round-off errors at the hardware level are always present, leading to erroneous interpretation of results. The history of computer architecture is full of these kinds of examples.<sup>67</sup> Arguably the most notorious case was the Pentium FDIV bug, for which Intel Co. lost \$500 million in revenue with the replacement of the flawed processors.<sup>68</sup>

For cases in which highly precise computations are needed, good practice dictates that it is necessary to measure and specify these errors in order to find suitable solutions. For instance, this type of error can be avoided by verifying the hardware,<sup>69</sup> or hardware redundancy,<sup>70</sup> among other solutions.

2. *Logical errors*: these kinds of errors lead to instabilities in the behavior of the computer program. They can be found in coding errors, and therefore blamed on the careless programmer; or as part of a faulty compiler or a computer language, where the skills of the programmer have no relation whatsoever. Briefly, a compiler is an

interpreter between an algorithm (written in some programming language) and the machine language. If the compiler misrepresents the data types, the instructions, and so forth, then inaccuracies must be expected in the output of the computation.<sup>71</sup> This kind of error can be avoided by software verification and validation (see my discussion in Section 3.3.2).

3. *Representational errors*: these kinds of errors are, allegedly, the most common ones. They are located at the level of the mathematical model or the specification, and deeply compromise the interpretation of results. This is the main source of error tracked back by computer scientists, researchers, and philosophers. It manifests itself in different ways: a grid too big for precise results, bad approximations, unacceptable mean square errors, and so forth.<sup>72</sup>

There is no need to dig into the details of each kind of error. Arguably, these errors are rare, therefore, there is no need for general skepticism about the reliability of computer simulations. We must keep in mind, however, that the epistemological analysis of computer simulations must admit the presence of these results. In the following chapters I delineate further conditions for a reliable computer simulation, and discuss in greater extent their epistemic importance.

## 3.4 Conclusion

The central aim of this chapter was to answer questions related to the ontology, semantics, and methodology of computer simulations. The chapter first focused on distinguishing between *analogical* simulations and *computer* simulations, drawing important ontological and semantic differences for use in future chapters. Once I established the general category of simulation, the next step was to distinguish the representational capacity of computer simulations over mere reckoning power as the most prominent epistemic virtue. Another important conceptual distinction was to narrow down the class of computer simulations of interest to equation-based simulations, excluding in this way cellular automata, agent-based simulations, and complex systems from the scope of this work. This distinction is paramount for the analysis of the explanatory power of computer simulation, mainly due to the specific representational characteristics that an equation-base simulation offers.

Finally, my analysis focused on the methodology of computer simulations. These analyses were illustrated with a concrete example of a computer simulation as used in current scientific practice. Finally, I proposed a working conceptualization of

computer simulations which corresponds to current scientific uses of equation-based simulations.

## Notes

<sup>1</sup>A recent comparison between the computer and the brain shows that even though computers are faster to calculate and can store more information, a human brain is much more efficient (Fischetti, 2011).

<sup>2</sup>Cf. (Humphreys, 2004, 110).

<sup>3</sup>For instance, (Guala, 2002; Parker, 2009; Winsberg, 2009b).

<sup>4</sup>On this last point, (cf. Kitcher, 1993, 27). More specifically, these structures are understood as natural kinds, objective causal relationships, objective natural necessities, and similar concepts that philosophers use to account for the metaphysics of science. See (Kitcher, 1986, 1994).

<sup>5</sup>In this work, I shall not make any conceptual difference among terms such as ‘describing,’ ‘representing,’ ‘standing for,’ and the like. There is a vast literature on the topic. See, for instance, (Frigg, 2009a; Hughes, 1997; Giere, 2009).

<sup>6</sup>Other philosophers and scientists that take the same stand on this distinction are (Haugeland, 1981; Von Neumann, 1958; Wiener, 1961).

<sup>7</sup>Cf. (Goodman, 1968, 159-160).

<sup>8</sup>Cf. (Goodman, 1968, 161).

<sup>9</sup>Cf. (Lewis, 1971, 324).

<sup>10</sup>Cf. (Lewis, 1971, 327).

<sup>11</sup>Virtually every philosopher of experimentation shares these ideas. A shortlist includes (Brown, 1991, 1994; Franklin, 1986; Galison, 1997; Hacking, 1983, 1988; Woodward, 2003).

<sup>12</sup>Cf. (Pylyshyn, 1984, 144).

<sup>13</sup>The controversial point here is the idea of ‘parallel causal-structures isomorphic to the phenomenon.’ For a closer look, please refer to (Trenholme, 1994, 118). For the sake of the argument, I will take it simply as a way to describe two system sharing the same causal relations. I base my interpretation on what the author says in the appendix: “the simulated system causally affects the simulating system through sensory input thereby initiating a simulation run whose causal structure parallels that of the run being undergone by the simulated system” (Trenholme, 1994, 128). All things considered, it is not clear what the author takes as causal-structures. Also, the introduction of ‘isomorphism’ as the relation of representation can be problematic. On this last point, see (Suárez, 2003).

<sup>14</sup>Cf. (Trenholme, 1994, 118).

<sup>15</sup>Unfortunately, the author leaves the notion of *intentional concepts* undefined. Given that these terms belong to the terminological canon of the cognitive sciences, and given that Trenholme is following Pylyshyn in many respects, it seems appropriate to suggest that a definition could be found in Pylyshyn’s work. However, it is not possible to find a clear definition in Pylyshyn’s work that could be of help. Instead, Pylyshyn talks of *intentional terms* (cf. Pylyshyn, 1984, 5), *intentional explanation* (cf. Pylyshyn, 1984, 212), *intentional objects* (cf. Pylyshyn, 1984, 262), *intentional descriptions* (cf. Pylyshyn, 1984, 20), and similar concepts, but no direct reference to *intentional concepts* whatsoever.

<sup>16</sup>On this point (cf. Trenholme, 1994, 119). Let it be noted that this claim is the equivalent to Pylyshyn’s description of physical manipulation processes. See (Trenholme, 1994, 144).

<sup>17</sup>Trenholme uses the notions of *symbolic process* and *symbolic computation* indistinguishably. Cf. (Trenholme, 1994, 118).

<sup>18</sup>Ian Hacking coined the term ‘life of its own’ to refer to laboratory practice that is independent and not driven by higher level theories (Hacking, 1983, 150). My concept here emphasizes the other side, that is, that the results of the simulation depend on the model, and it is only because some models represent an empirical system that those results can be related to the external world.

<sup>19</sup>Humphreys calls this view ‘selective entity realism.’ Cf. (Humphreys, 2004, 84).

<sup>20</sup>For more discussion on this issue, see (Durán, 2013a). The remainder of this section is a part of that paper.

<sup>21</sup>Cf. (Humphreys, 1991, 497-498).

<sup>22</sup>Cf. (Humphreys, 1991, 500).

<sup>23</sup>For the rest of this work, ‘to be analytically solvable,’ ‘to be solved by an analytic method,’ or similar assertions refer to finding solutions of a model by pen-and-paper mathematics.

<sup>24</sup>Cf. (Humphreys, 1991, 502).

<sup>25</sup>A notable exception is (García and Velasco, 2013).

<sup>26</sup>See (Guala, 2002).

<sup>27</sup>This is the opinion of (Humphreys, 2004; Morgan, 2005; Winsberg, 2003).

<sup>28</sup>For each one of these uses, cf. (Hartmann, 1996).

<sup>29</sup>Cf. (Humphreys, 2004, 60).

<sup>30</sup>Cf. (Humphreys, 2004, 68).

<sup>31</sup>Cf. (Wolfram, 1984a, vii).

<sup>32</sup>Cf. (Vichniac, 1984, 113).

<sup>33</sup>See, for instance, (Omohundro, 1984; Wolfram, 2002).

<sup>34</sup>Of course, not every proponent of cellular automata defends these assumptions. There is, however, a special branch of philosophy known as ‘pan-computationalism’ that defends most of the metaphysical assumptions (Wootters and Langton, 1990; Smith et al., 1984; Hillis, 1984; Gosper, 1984).

<sup>35</sup>For instance, (Lesne, Online).

<sup>36</sup>Tommaso Toffoli is of this opinion. See (Toffoli, 1984).

<sup>37</sup>See, for instance, (Bedau and Humphreys, 2008; Bedau, 1997; Humphreys, 1997).

<sup>38</sup>For further reading on this topic, see (Wootters and Langton, 1990; Toffoli, 1984).

<sup>39</sup>A good example of this kind of work is (Omohundro, 1984).

<sup>40</sup>(Lotka, 1910).

<sup>41</sup>For a more detailed analysis on the distinction *precision* and *accuracy*, see the Lexicon.

<sup>42</sup>For examples of the uses of theoretical simulations, see (Tymoczko, 1979; Babuska et al., 2000).

<sup>43</sup>A ‘word’ represents the minimum unit of data used by a particular processor design. It is a fixed sized group of bits that are handled as a unit by the hardware of the processor.

<sup>44</sup>For more on numerical methods see, for instance (Press et al., 2007).

<sup>45</sup>The prefix ‘pseudo’ is important here for at least two reasons: first, because these methods are based on an algorithm that produces numbers on a recursive basis, possibly repeating the series of numbers produced. Second, because computers make use of words that are finite in size, reducing significantly the amount of numbers generated. In any case, pure randomness in computers can never be achieved.

- <sup>46</sup>Arguably, it is the *only* stochastic method. Any other so-called stochastic method (stratified sampling, importance sampling, etc.) is an adaptation of the original Monte Carlo. See (Metropolis and Ulam, 1949; Metropolis, 1987).
- <sup>47</sup>For more on the Monte Carlo methods, see (Grimmett and Stirzaker, 1992).
- <sup>48</sup>See (Oberkampf et al., 2003).
- <sup>49</sup>See (Oberkampf and Roy, 2010; Oberkampf and Trucano, 2008).
- <sup>50</sup>For a complete list, see (Atkinson et al., 2009; Gould et al., 2007).
- <sup>51</sup>See the Lexicon and (Butcher, 2008).
- <sup>52</sup>A good example of this, but at the level of computer hardware, is von Neumann's architecture that has been the same since his 1945 report (von Neumann, 1945).
- <sup>53</sup>I have slightly modified Woolfson and Pert's example. Cf. (Woolfson and Pert, 1999, 20). In the original example, the authors indicate four bodies where one is the planet, and the satellite is divided into the other three (the three positions of the satellite under tidal stress). The modification clarifies a source of possible confusion that the simulation is about a 4-body problem, when in reality it is only a two body problem. I also reduce the mass of the first body. It originally was the size of Jupiter (i.e.,  $1.8986 \times 10^{27}$  kg), but in such a case the satellite would collapse into it.
- <sup>54</sup>Thanks to Dr. Ing. Wolfgang Nowak for discussion on this point.
- <sup>55</sup>Also known as 'internal validity' and 'external validity,' respectively.
- <sup>56</sup>Cf. (Oberkampf and Roy, 2010, Preface).
- <sup>57</sup>See (Oberkampf et al., 2003).
- <sup>58</sup>See (Oberkampf and Roy, 2010, 21-29) for an analysis on the diversity of concepts. Also, see (Salari, 2003; Sargent, 2007; Naylor et al., 1967b).
- <sup>59</sup>Cf. (Morrison, 2009, 43).
- <sup>60</sup>See also (ASME, 2006, 10) and (Mihram, 1972; Nelson, 1992; Oberkampf et al., 2003; Oreskes et al., 1994; Sargent, 2007).
- <sup>61</sup>Also referred to as *solution verification* in (Oberkampf and Roy, 2010, 26), or as *numerical error estimation* in (Oberkampf et al., 2003, 26).
- <sup>62</sup>Cf. (Oberkampf et al., 2003, 26).
- <sup>63</sup>Cf. (MacKenzie, 2006, 130).
- <sup>64</sup>For a more detailed discussion on Figure 3.5, see (Oberkampf and Roy, 2010, 23-31).
- <sup>65</sup>For a recent philosophical work on errors, see (Parker, 2008).
- <sup>66</sup>See (Franklin, 1990, 109).
- <sup>67</sup>For a cogent history of computation, see (Rojas and Hashagen, 2000; Ceruzzi, 1998).
- <sup>68</sup>See (Cipra, 1995; Moler, 1995). Also see <http://www.cs.earlham.edu/~dusko/cs63/fdiv.html>.
- <sup>69</sup>See, for instance (Cohn, 1989).
- <sup>70</sup>See, for instance (Koren and Krishna, 2007; Al-Arian, 1992; Lee and Anderson, 1990).
- <sup>71</sup>See, for instance (Tanenbaum and Woodhull, 2006; Tews et al., 2009; Klein et al., 2009).
- <sup>72</sup>For more on this point, see (Jason, 1989; Mayo and Spanos, 2010; Mayo, 2010).

# Chapter 4

## Theoretical assumptions behind scientific explanation

### 4.1 Introduction

In previous chapters, my efforts have been focused on elucidating the notion of computer simulation. These efforts are now going to be cashed out in my defense of the epistemic power of computer simulations. When philosophers discuss the epistemic power of computer simulations, they are interested in questions such as “what can we learn by running a computer simulation?” or “what kind of knowledge about the world does a computer simulation yield?” This chapter begins my attempt to answer these questions in a qualitative and systematic way. The motivation is that by showing that computer simulations have explanatory power, I will also be defending their epistemic value. In other words, since scientific explanation is an epistemic activity *par excellence*, then showing how computer simulations explain will be emphasizing precisely their epistemic value.

The many aspects of scientific explanation which must be addressed force me to divide my findings into two chapters. The current chapter discusses theoretical assumptions behind scientific explanation. This chapter also analyzes suitable theories of scientific explanation as candidates for computer simulations. One of the results of analyzing these candidates is that the unificationist theory of scientific explanation emerges as the most suitable conceptual framework for explanation in computer simulations. However, it is not until the next chapter that I discuss the unificationist approach.

A central issue arising here, then, is related to the metaphysics of scientific explanation. Indeed, to answer the question ‘*why p?*’ requires the semantics of *p* to be an empirical phenomenon.<sup>1</sup> A realist theory of scientific explanation is interested

in explaining a phenomenon that is ‘out there,’ for in this way one can say that we gain some understanding of the world by explaining it. However, in Section 3.2 I argued that computer simulations do not necessarily represent empirical systems, for they can be heuristic or merely imaginary, and therefore lacking any representational content. I even borrow the term ‘world of their own’ from Ian Hacking to depict this feature of computer simulations. Given that my purpose is to confer epistemic power on computer simulations by showing how they are explanatory devices, I must find the conditions under which there is genuine representation of an empirical target system. These conditions work as the epistemic guarantees that explaining results of a computer simulation applies equally to an empirical phenomenon and, thereby, yields understanding of the world. Let me illustrate these issues with an example. Consider for instance the spikes in Figure 3.3 which are the result of computing one particular simulation. A successful explanation of why the simulated spikes occur must also be applicable as an explanation of why the spikes occur in the empirical world. In order to justify that this is one and the same explanation, we must first justify that the computer simulation is a reliable process that produces genuine results in the sense that they are a good representation of the target system. Clarifying these issues is the first task of this chapter, and it is addressed in Section 4.2.1.

In Section 4.2.2 I address the notion of *understanding* and how it relates to scientific explanation. This is a short but important discussion that aims at establishing *scientific explanation* as an epistemic notion. Next, I address a classification for theories of scientific explanation that divides them into two kinds; namely, *explanatory internalist* accounts, and *explanatory externalists* accounts. I use this classification for narrowing down the kind of theory of explanation of interest for computer simulations to four *internalist accounts*.

Finally, Section 4.3 aims at analyzing which internalist account provides the most suitable conceptual framework for computer simulations as explanatory devices. It also raises the following question: could computer simulations stand by themselves as a new theory of scientific explanation? I answer this question negatively and outline my reasons in Section 4.3.1. The remainder of the chapter is a discussion of three internalist accounts that I reject based on theoretical assumptions that prove to be incompatible with computer simulations. I finally endorse the *unificationist account* as elaborated by Philip Kitcher as the most suitable for computer simulations. Due to the importance and breadth of this discussion, I address it more extensively in Chapter 5.

Before getting into the analysis of scientific explanation, let me briefly introduce some of the basic terminology still in use in the contemporary literature. According to Carl Hempel, a scientific explanation consists of two major constituents: an *explanans*, or the description that accounts for the phenomenon to be explained, and the *explanandum*, or the description of the phenomenon to be explained.<sup>2</sup>

## 4.2 Knowledge and understanding

Epistemologists conceive *knowledge* and *understanding* as two distinctive epistemic concepts. The epistemic difference can be illuminated with a simple example: one might know that  $2 + 2 = 4$  without understanding basic arithmetic. An analogy with computer simulations can also be established: scientists might *know* that the simulated trajectory represents the real trajectory of a satellite under tidal stress without *understanding* why the spikes would occur (see Figure 3.3). *Knowledge* consists in being informed about a fact (typically taken as ‘descriptive knowledge’ or ‘knowing that’); *understanding*, however, involves something deeper and more valuable that is comprehending why the fact obtains (typically taken as ‘explanatory knowledge’ or ‘knowing why’).

The central issue arising with the notion of *knowledge* is that computer simulations produce data, and therefore such an explanation is about that data. However, since we explain (and simulate) in order to understand something about the world, it is paramount to find the conditions under which the computer simulation also explains something about it. My solution to this problem comes with the notion of *reliable process* which is understood as the conditions under which we are justified in believing that the computer simulation genuinely represents an empirical target system. The ultimate aim here is to justify our belief that the explanation of results of a computer simulation could also be applied to the empirical phenomenon. The sense in which I take a ‘genuine representation’ has been given in Section 1.2.2 and in Section 3.2.3; that is, a computer simulation implements a model that provides theoretical underpinning of the phenomenon represented and is free of mathematical artifacts. This point raises some metaphysical riddles which I will be addressing in Section 5.3.2.

In the following sections I analyze the concept of *knowledge* and *understanding* separately. The notion of *knowledge*<sup>3</sup> is related to issues about reasons for belief, and it is important to us because it furnishes the view that a computer simulation genuinely represents an empirical target system. Section 4.2.1 deals precisely with this issue. Following Alvin Goldman, I conclude that one is justified in believing

that the computer simulation genuinely represents its target system because the simulation is a *reliable process*.<sup>4</sup> Such a justification provides the basis for the belief that the explanation of results of a computer simulation also applies as an explanation of the empirical phenomenon.

Section 4.2.2, on the other hand, addresses the notion of *understanding*, which is related to issues regarding the general structure, relations, and construction of a comprehensive corpus of belief. The notion of understanding is important because it grants epistemic value to scientific explanation and, in doing so, provides the means to conferring epistemic power on computer simulations.

### 4.2.1 Knowledge

Any theory of scientific explanation aims at answering the question ‘why  $q$ ?’ where  $q$  is the explanandum to be explained by the explanans. The nature of the explanans and explanandum varies depending on the theory of explanation: they might be an ‘argument,’ a ‘list of assemblage of statistically relevant factors,’ or a ‘sentence,’<sup>5</sup> among others. The nature of the explanatory relation also varies, for it can be ‘deductive,’ ‘causal,’ or an ‘answer to a why-question.’<sup>6</sup> Conforming to these different natures, several features follow. For instance, argument theories can be deductive as a way of giving substance to ideas of certainty (e.g., Hempel and Kitcher) and causal theories can be probabilistic, therefore conferring epistemic probability on the explanation (e.g., Salmon).

In addition, theories of scientific explanation purport to explain a particular,<sup>7</sup> which might take several forms: it might be a ‘fact,’ a ‘phenomenon,’ a ‘concrete event,’ a ‘property,’ etc. The *relatum* of  $q$  (i.e., the referent of  $q$ ) is therefore something out there, an “individual chunk of reality” as David-Hillel Ruben graphically calls it, for which the explanatory relation must account.<sup>8</sup> “The basis of explanation is in metaphysics,” Ruben says, “*objects or events in the world must really stand in some appropriate ‘structural’ relation before explanation is possible*. Explanations work, when they do, only in virtue of underlying determinative or dependency structural relations in the world.” (Ruben, 1992, 210. Emphasis original.) For metaphysical economy, then, I will consider the relatum of  $q$  as a natural phenomenon broadly construed: from an object such as pollen, to vibration over water. The notion of phenomenon, then, includes entities, events, processes, and regularities. Examples of these theoretical assumptions abound in philosophical literature on explanation. Take for instance Hempel’s pioneering explanation of why the mercury column in a closed vessel grew steadily shorter during the ascent to top of the Puy-de-Dôme (see Section 4.3.2).

Metaphysical assumptions such as these beg the question of *scientific realism*, on which every theory of explanation has a view. Since I take the unificationist account as the most suitable theory of explanation for computer simulations, I will also adopt its realist point of view on explanation. In particular, I subscribe to what Philip Kitcher calls *modest unificationism*, which adopts a naturalist view of realism (see my discussion in Section 5.2). I have little doubt that a realist theory of explanation is the most suitable for accounting for the aims and achievements of the scientific enterprise. However, I will not discuss this point any further here. The philosophical accounts of realism are not the principal problems here; rather, I am interested in showing how to accommodate computer simulations into a realist theory of explanation.

Now, the central issue raised with the metaphysics of explanation in computer simulations is this: how can we know that the explanation of results of a computer simulation also applies to an empirical phenomenon? In other words, what kind of epistemic guarantees could be provided in order to ensure that an explanation of results of a simulation are epistemically equivalent to the explanation of the empirical phenomenon? From this question it becomes clear that the problem is that the relatum of the explanandum is no longer an empirical phenomenon, as the metaphysics of scientific explanation assumes, but rather the results of the computer simulation (i.e., data obtained by computing the simulated model). The first step towards answering the above question is to find a justification that our beliefs in the results of the simulation genuinely represent the empirical phenomenon intended to be simulated.

In Section 3.2.1 I used the image of a ‘world of their own’ as a metaphor for describing simulations as systems whose results are directly related to the simulation model, but only indirectly related (via representation) to the target system. Such a metaphor depicted computer simulations as systems that might be disentangled from reality by stemming from the imagination of the scientists. For all purposes, my account of computer simulations as explanatory devices must be restricted to those simulations that represent an empirical target system.<sup>9</sup> In order to illustrate the problem at hand, and to give grounds for the former restriction over computer simulations, let me briefly discuss cases where a computer simulation could *not* be explanatory:

Case a): A simulation model that does not represent an empirical target system (e.g., a simulation of the Ptolemaic movement of planets). We cannot legitimately claim for an explanation of the planetary movement using a false model.<sup>10</sup>

Case b): A computer simulation that is purely heuristic in nature. Although it might hold some interest as an exploratory simulation, its target system is known to be nonexistent (e.g., the Oregonator is a simulation of the Belousov-Zhabotinsky chemical reaction whose system of equations is *stiff* and might lead to qualitatively erroneous results).<sup>11</sup> Although a genuine simulation, it cannot be expected to offer a successful explanation.<sup>12</sup>

Case c): A computer simulation that miscalculates, leading to erroneous data. Such data does not represent any possible state of the target system, regardless of the representational goodness of the simulation model. Therefore an explanation, although possible, would not yield understanding of the world.

The first two cases are legitimate computer simulations, for there is nothing preventing the scientist from carrying out hypothetical or heuristic research. The principle is that in computer simulations there are no limits to the imagination. However, neither would lead to a successful explanation and ulterior understanding of the world. Let us note that case b) has some kinship with cases where the simulation model represents the target system and the results of the simulation are not the byproduct of miscalculations, since the initial and boundary conditions are not realistically construed (e.g., a simulation that speculates how the tides would be if the gravitational force were weaker than it actually is). This is a special case for computer simulations that I will deal with in Chapter 5. Finally, case c) is less likely to be considered a legitimate simulation, and it certainly does not yield understanding of the empirical world. However, it is a probable scenario and as such must be considered for analysis.

What is it left after this list? My interest here lies in what constitutes a *successful explanation*, that is, what would be rational to take as the structure of the phenomenon to be explained.<sup>13</sup> Successful explanations, then, demand that a computer simulation yield insight into the way the world is. Thus understood, it is important to find the conditions under which one *knows* that the results of the computer simulation share some sort of structure with the empirical phenomenon. Which are these conditions?

Typically, epistemologists analyze the notion of knowledge in the following way: “to know something is to have justified true belief about that something.”<sup>14</sup> According to this assertion, then, knowledge is justified true belief; that is, ‘*S* knows that

$p$ ’ if and only if:

- (i)  $p$  is true,
- (ii)  $S$  believes that  $p$ ,
- (iii)  $S$  is justified in believing that  $p$

For us,  $p$  might be interpreted as the proposition ‘the results of the computer simulation genuinely represent the empirical phenomenon,’ for this is precisely what we need to justify our knowledge of. Consider now  $p$  as a proposition for our example: “the spikes produced after simulating the orbital movement of a satellite under tidal stress genuinely represent the spikes that a real satellite under real tidal stress would produce.” Following the schema above, we *know* that the results of the simulation represent an empirical phenomenon because we are justified in believing that  $p$  is true.

Epistemologists are aware that taking knowledge as justified true belief leads to problems. Most prominently are the so-called ‘Gettier problems,’ which establish situations where true belief is neither individually necessary nor jointly sufficient a condition for knowledge.<sup>15</sup> In other words, the justification condition (i.e., (iii) above), which was set for preventing lucky guesses from counting as knowledge, proved to be ineffective against Gettier’s objection about the possibility of falsely justified true belief. The solution is to settle for a bit less, that is, for a belief-forming process that, most of the time, produces justified beliefs. One such candidate is *reliabilism* as elaborated by Alvin Goldman.<sup>16</sup>

In its simplest form, *reliabilism* says that a belief is justified in the case that it is produced by a reliable process.<sup>17</sup> In our own terms, we know that we are justified in believing that the results of the simulation represent an empirical phenomenon because there is a *reliable process* (i.e., the computer simulation) that, most of the time, yields good results that represent the empirical phenomenon. The challenge is now reduced to showing the conditions under which a computer simulation is a reliable process. Let me first say something more about *reliabilism*.<sup>18</sup>

Goldman explains that the notion of *reliability* “consists in the tendency of a process to produce beliefs that are true rather than false” (Goldman, 1979, 10). His proposal highlights the place that a belief-forming process has in the steps towards knowledge. In this vein, one knows  $p$  because there is a reliable process that produces the belief that  $p$  is the case. In other words, such a belief-forming process yields beliefs that are, most of the time, true rather than false. Consider, for instance, knowledge acquired by a ‘reasoning process,’ such as doing basic arithmetic operations. Reasoning processes are, under normal circumstances and within a

limited set of operations, highly reliable. Indeed, there is nothing accidental about the truth of a belief that  $2 + 2 = 4$ , or that the tree in front of my window was there yesterday and, unless something extraordinary happens, it will be in the same place tomorrow.<sup>19</sup> Thus, according to the reliabilist, a belief produced by a reasoning process qualifies, most of the time, as an instance of knowledge.<sup>20</sup>

The question now turns to what it means for a process to be *reliable* and, specific to my interests, what this means for the analysis of computer simulations. Let us illustrate the answer to the first issue with an example from Goldman:

If a good cup of espresso is produced by a reliable espresso machine, and this machine remains at one's disposal, then the probability that one's next cup of espresso will be good is greater than the probability that the next cup of espresso will be good given that the first good cup was just luckily produced by an unreliable machine. If a reliable coffee machine produces good espresso for you today and remains at your disposal, it can normally produce a good espresso for you tomorrow. The reliable production of one good cup of espresso may or may not stand in the singular-causation relation to any subsequent good cup of espresso. *But the reliable production of a good cup of espresso does raise or enhance the probability of a subsequent good cup of espresso. This probability enhancement is a valuable property to have.* (Goldman, 2009, 28. Emphasis mine)

The probability here is interpreted objectively, that is, as the tendency of a process to produce beliefs that are true rather than false. The core idea is that if a given process is reliable in one situation then it is very *likely* that, all things being equal, the same process will be reliable in a similar situation. Let it be noted that Goldman is very cautious in demanding infallibility or absolute certainty for the reliabilist account. Rather a long-run frequency or propensity account of probability furnishes the idea of a reliable production of coffee that increases the probability of a subsequent good cup of espresso.<sup>21</sup>

Although the reliabilist account has been at the center of much criticism,<sup>22</sup> here I am more interested in its virtues. Particularly, the reliabilist analysis of knowledge facilitates the claim that we are justified in believing that computer simulations are *reliable processes* if the following two conditions are met:

- (a) The simulation model is a good representation of the empirical target system;<sup>23</sup> and
- (b) The reckoning process does not introduce relevant distortions, miscalculation, or any kind of mathematical artifact.<sup>24</sup>

Both conditions must be fulfilled in order to have a reliable computer simulation, that is, a simulation whose results *genuinely represent* empirical phenomena. Let

me illustrate what would happen if one of the conditions above were not met. As argued before, suppose that condition (a) is not met, as is the case of using the Ptolemaic model of planetary movement. In such a case, the computer simulation does not represent any empirical planetary system and therefore its results could not be considered as genuinely standing for an empirical phenomenon. It follows that there are no grounds for claiming a successful explanation of the movement of a real planet. Suppose now, instead, that condition (b) is not met. This means that during the calculation of the satellite's orbit the simulation produced an artifact of some sort (most likely, a mathematical artifact). In this case, although the simulation model represents an empirical target system, the results of the simulation fail to represent the empirical phenomenon. The reason is that miscalculations directly affect and downplay the degree of representativeness of the results. It follows that one is in no condition to claim for a successful explanation when miscalculations are present. In simpler words, without genuine representation there cannot be successful explanation either.

In Chapter 2, I described with certain detail the three levels of computer software; namely, the *specification*, the *algorithm*, and the *computer process*. I also claimed that all three levels make use of techniques of construction, language, and formal methods that make the relations among them trustworthy: there are well-established techniques of construction based on common languages and formal methods that relate the specification with the algorithm, and allow the implementation of the latter on the digital computer. The totality of relations are precisely what make the computer simulation a reliable process. In other words, these three levels of software are intimately related to the two conditions above: the design of the specification and the algorithm fulfill condition (a), whereas the running computer process fulfills condition (b). It follows that a computer simulation is a reliable process because its constituents (i.e., the specification, the algorithm, and the computer process) and the process of construing and running a simulation are based, individually and jointly, on trustworthy methods. Finally, from establishing the reliability of a computer simulation it follows that we are justified in believing (i.e., we *know*) that the results of the simulation genuinely represents the empirical phenomenon.

We can now assimilate Goldman's example of the espresso machine into our case: if a computer simulation produces good results, then the probability that the next result is good is greater than the probability that the next result is good given that the first results were just luckily produced by an unreliable process. By entrenching computer simulations as reliable processes, we are grounding representational compatibility between the results obtained and the empirical phenomenon. It is

in this way that, to my mind, computer simulations establish the necessary meta-physical foundations of scientific explanation. That is, explanation by computer simulations is possible because representational dependence between results and the empirical phenomenon has been established. It follows that explaining the former is epistemically equivalent to explaining the latter.

Before I proceed to addressing the notion of *understanding* and how it is related to scientific explanation, allow me to introduce a new term. For the remainder of this work, I shall refer to results of a computer simulation genuinely representing an empirical phenomenon as the *simulated phenomenon*. In this vein, to say that we want to explain the simulated spikes in Figure 3.3 is to say that we want to explain similar spikes found in the world. Let it be noted that such spikes might or might not have been observed or measured empirically. This means that a genuine representation of the empirical target system presupposes a genuine representation of each empirical phenomenon belonging to that target system. For instance, to genuinely represent the planetary movement by a simulation that implements a Newtonian model also means to genuinely represent each phenomenon that falls within that model.

Let me now turn to the notion of *understanding* and how we can relate it to the concept of explanation, making the latter epistemic.

## 4.2.2 Understanding

In the previous section, I established the conditions under which the scientist *knows* that the computer simulation is a reliable process that genuinely represents an empirical system. However, and as I mentioned before, to know that  $2+2$  is a reliable operation that leads to 4 does not entail an *understanding* of basic arithmetic. In logical terms, to know that something is the case is a necessary condition, although not sufficient, for understanding why that something is the case.<sup>25</sup>

Why is an analysis on scientific *understanding* important? The short answer is that scientific understanding is essentially an epistemic notion that involves scientific activities such as explaining, predicting, and experimenting. And since my aim is to defend the epistemic power of computer simulations as explanatory devices, analyzing the notion of scientific understanding is the natural starting point. Understanding and explaining, then, are conceived here as two intimately related notions which serve as the basis for my defense on the epistemic value of computer simulations. Let me begin with the notion of scientific understanding as I make my way to explanation.

There is general agreement that the notion of *understanding* is hard to define. We

say that we ‘understand’ why the Earth revolves around the Sun, or that the velocity of a car could be measured by deriving the position of the body with respect to time. Typically, epistemologists and philosophers of mind are more trained than any other philosopher for figuring out what exactly understanding amounts to. Philosophers of science, instead, seem to be content with assimilating the role that understanding plays in the construction of a scientific corpus of beliefs and the general truths about the world. A first rough conceptualization, then, takes scientific understanding as corresponding to a coherent corpus of beliefs about the world. These beliefs are mostly true (or approximately true) in the sense that our models, theories, and statements about the world represent the way it really is (or it approximately is).<sup>26</sup>

Naturally, not everything that has been scientifically represented is strictly true. We do not have a perfect understanding of how our scientific theories and models work, or a complete grasp of why the world is the way it is. For these reasons a notion of understanding must also allow some falsehoods. Catherine Elgin has coined an accurate term for these cases; she calls them ‘felicitous falsehoods’ as a way to exhibit the positive side of a theory of not being strictly true. Such felicitous falsehoods are the idealizations and abstractions that theories and models purport. For instance, scientists know very well that no actual gas behaves in the way that the kinetic theory of gases describes them. However, the ideal gas law accounts for the behavior of gases by describing their movement, properties, and relations. There is no such gas, but scientists purport to understand the behavior of actual gases by reference to the ideal gas law (i.e., to reference a coherent corpus of true beliefs).<sup>27</sup> Let it be noted, however, that although a scientific corpus of beliefs is riddled with these felicitous falsehoods, this fact does not entail that the totality of our corpus of beliefs is false. A coherent body of predominantly false and unfounded beliefs, such as alchemy or creationism, does not constitute understanding of chemistry or the origins of beings, and it certainly does not constitute a coherent corpus of true beliefs. In this vein, the first demand for having understanding of the world is that our corpus of beliefs is mostly populated with true (or approximately true) beliefs.

Taken in this way, it is paramount to account for the mechanisms by which new beliefs are incorporated into the general corpus of true beliefs, how it is populated. Gerhard Schurz and Karel Lambert, for instance, assert that “to understand a phenomenon P is to know how P fits into one’s background knowledge” (Schurz and Lambert, 1994, 66).<sup>28</sup> There are several operations that allow scientists to do this. For instance, a mathematical or logical derivation from a set of axioms incorporates new well-founded beliefs into the corpus of arithmetics or logic, making them more coherent and integrated. There is also a pragmatic dimension that considers that we

incorporate new beliefs when we are capable of using our corpus of belief for some specific epistemic activity, such as reasoning, working with hypotheses, and the like. Elgin, for instance, indicates that understanding geometry entails that one must be able to reason geometrically about new problems, to apply geometrical insight in different areas, to assess the limits of geometrical reasoning for the task at hand, and so forth.<sup>29</sup>

Here I am interested in one particular way of incorporating new beliefs, that is, by means of explaining new phenomena. For this I need to show how a theory of scientific explanation works as a coherence-making process capable of incorporating new beliefs into a corpus of beliefs. This is the core epistemic role for scientific explanation largely admitted by the philosophical literature. Jaegwon Kim, for instance, considers that “the idea of explaining something is inseparable from the idea of making it more intelligible; to seek an explanation of something is to seek to understand it, to render it intelligible” (Kim, 1994, 54). Stephen Grimm makes the same point with fewer words: “understanding is the goal of explanation” (Grimm, 2010). Explanation, then, is a driving force for scientific understanding. We can understand more about the world because we can explain why it works the way it does. In addition, a successful account of computer simulations as explanatory devices must show how, by simulating a piece of the world, they yield understanding of it. By accomplishing this aim, I would be successfully conferring epistemic power on computer simulations.

Now, there are two ways to approach the explanatory role of computer simulations: either computer simulations provide a new account of scientific explanation, independent of current theories, or they are subsumed under the conceptual framework of current theories of explanation. In Section 4.3 I outline some of the reasons that make the first option inviable. I then make my case for the second option by analyzing how different theories of explanation provide the necessary conceptual framework for computer simulations. Let me finish this section by shortly addressing a few technical aspects of scientific explanation.

Scientific explanation is concerned with questions that are relevant for scientific purposes.<sup>30</sup> As tautological as this might sound to many, it introduces the first important delimitation to the notion of explanation. For instance, scientific explanation is interested in answering why puerperal fever is contracted or why water boils at 100<sup>o</sup> C. Non-scientific explanations include why the ink was spilled all over the floor<sup>31</sup> or why the dog is outside.<sup>32</sup> Similarly, the researcher working with computer simulations wants to explain why a satellite under tidal stress forms the spikes

as shown in Figure 3.3 rather than, say, why the simulation raised an error.

The twentieth century's most influential model of scientific explanation is the covering law model as elaborated by Carl Hempel and Paul Oppenheim.<sup>33</sup> Several other models followed, leading to a number of alternative theories: the *statistical relevance model*, the *ontic approach*, the *pragmatic approach*, and the *unificationist approach*.<sup>34</sup> We must also include on this list the current efforts on 'model explanation' accounts, such as (Bokulich, 2011) and (Craver, 2006), and mathematical explanation of physical phenomena, such as (Baker, 2009) and (Pincock, 2010).

The logic of scientific explanation typically divides theories of explanation into two classes. The first belong to *explanatory externalism*,<sup>35</sup> which "assert[s] that every explanation 'tracks' an objective relation between the events described by the explanans and the explanandum" (Kim, 1994, 273). Causal theories of explanation are forms of this view (e.g., (Salmon, 1984)). The second class belongs to *explanatory internalism*, which "see explanation as something that can be accounted for in a purely epistemic way, ignoring any relation in the world that may ground the relation between explanans and explanandum" (Kim, 1994, 273). For instance, Hempel's covering law model and the unification views are theories of this latter class (e.g., (Friedman, 1974) and (Kitcher, 1981, 1989)).

This distinction is useful for our purposes because it narrows down the class of theories of explanation suitable for computer simulations. In Section 3.2.1 I argued that the results of a computer simulation do not depend on any external causal relation, but instead are the byproduct of abstract calculus (i.e., reckoning the simulation model). Given this fact about computers, theories of explanations that are classified as explanatory internalist are more suitable for explaining simulated phenomena than explanatory externalist accounts. The reason for excluding explanatory externalist accounts is very simple: we cannot track back the causal history belonging to the physical states of the computer that produced the results of the simulation.

With this firmly in mind, the search for a suitable theory of explanation is reduced to four internalist accounts provided by the philosophical literature. These are: the Hempelian Deductive-Nomological Model, mathematical explanation of physical phenomena, 'model explanation' accounts, and the unificationist account.

Equally important for organizing our analysis are the two questions that Kim poses for any theory of explanation, namely:

*The Metaphysical Question:* When  $G$  is an explanans for  $E$ , in virtue of what relation between  $g$  and  $e$ , the events represented by  $G$  and  $E$  respectively, is  $G$  an explanans for  $E$ ? What is the objective relation connecting events,  $g$  and

$e$ , that grounds the explanatory relation between their descriptions,  $G$  and  $E$ ? (Kim, 1994, 56)

*The Epistemological Question:* What is it that we know (that is, what exactly is our epistemic gain) when we have an explanation of  $p$ ? (Kim, 1994, 54)

The metaphysical question is interested in making visible the relation between explanans and explanandum that renders the latter as explained. The epistemological question, instead, aims at asking about the kind of epistemic insight that one obtains by carrying out an explanation. In this vein, it is intimately related to the notion of understanding as recently discussed.<sup>36</sup>

The next sections are organized as follows. I first argue against the possibility of computer simulations introducing a new account of scientific explanation. Then, I discuss the first three internalist accounts of explanation listed above: the Hempelian Deductive-Nomological model, mathematical explanation of physical phenomena, and ‘model explanation’ accounts. I use this discussion to show that none of these theories can account for computer simulations as explanatory devices. These discussions are organized around Kim’s metaphysical and epistemic questions just presented. The fourth internalist account, the unificationist theory of explanation, will be analyzed in full detail in the next chapter.

## 4.3 Theories of scientific explanation

### 4.3.1 Against a new theory of scientific explanation

There are several reasons that make me believe we cannot analyze the explanatory power of computer simulations independently of a framework provided by well-established theories of explanation. Let me outline these reasons in the following list:

1. Computer simulations do not introduce a new unit of analysis for the philosophy of science. Indeed, ontologically speaking, they could be interpreted as a special kind of model. Theories of explanation have already accounted for explanation using scientific models, such as the ‘model of explanation’ as defended by Bokulich, or the ‘unificationist theory’ as elaborated by Kitcher.<sup>37</sup>
2. However, if one is still tempted to argue that computer simulations could introduce a new theory of scientific explanation, one must show that:

- (a) The form of the explanatory relationship must not be: causal, nomological -inductive or deductive-, pragmatic, mechanic, or mathematic.<sup>38</sup>
- (b) The form that the explanans and the explanandum take must not be: an argument (e.g., (Kitcher, 1989)), a statement (e.g., Salmon’s Statistical Relevance model (Salmon et al., 1971)), or a single sentence (e.g., (Hempel, 1965)), and so on.

Claims asserting that computer simulations introduce a new theory of scientific explanation must accommodate points 2.a and 2.b above. That is, provided that computer simulations do introduce a new ontology (point 1 above), the new theory of explanation must either introduce a new form for the explanans and the explanandum (point 2.a), and/or the explanatory relationship must not be listed in point 2.b. In other words, for computer simulations to introduce a new theory of scientific explanation it is first necessary to break with the philosophical tradition on scientific explanation by imposing a new paradigm. I am very skeptical that such a thing can be achieved by computer simulations.

Moreover, despite the modest volume of literature on the topic, there is no author that proposes a new account of scientific explanation. Ulrich Krohs, one of the few philosophers who wrote on this topic, makes use of the mechanistic account of explanation as elaborated by Peter Machamer.<sup>39</sup> Krohs resolves the question of explanation by showing how computer simulations accommodate to the mechanistic account. I disregard this view on the fact that the mechanistic account requires causal relations to account for the explanation. It is interesting that although Krohs seems very convinced that explanation in computer simulations is possible in virtue of tracking back causal relations, he never effectively proves that computer simulations *have* causal relations to track in the first place.<sup>40</sup>

There are sufficient reasons for being skeptical about the possibility of computer simulations introducing a new theory of scientific explanation. However, the converse is, to my mind, entirely attainable. That is, of the many theories of scientific explanation at our disposal, there is at least one that could easily accommodate to computer simulations. Let me now turn to the analysis of this possibility.

### 4.3.2 Deductive nomological model

During the 1950s and 1960s, Hempel extensively and systematically wrote about the logic of scientific explanation. His work is broadly considered to be the most fundamental philosophical work on scientific explanation. According to Hempel, an explanation is “an argument to the effect that the phenomenon to be explained

[...] was to be expected in virtue of certain explanatory facts” (Hempel, 1965, 336). His work expounds three models of explanations: ‘deductive nomological,’<sup>41</sup> ‘deductive statistical,’ and ‘inductive statistical.’ There is really no need to go into detail on each model. The ‘deductive statistical’ is analogous to the ‘deductive nomological’ with the exception that it makes use of statistical laws, instead of deterministic laws. The ‘inductive statistical,’ on the other hand, uses an analogous explanatory schemata as does the ‘deductive statistical,’ except that it makes use of inductive arguments instead of deductive arguments. Of course, each model has its particularities worth exploring by themselves. However, the main reason for such a diversity of models stems from the fact that they are useful in different contexts and for different phenomena. Empirical phenomena, according to Hempel, are explained by means of the deductive nomological model of explanation. For my present purposes, then, a general analysis of this model of scientific explanation is enough for showing how it fails to provide the right conceptual framework for computer simulations.

The deductive nomological model of explanation takes the following form:

$$\left. \begin{array}{l} C, C_2, \dots, C_k \\ L_1, L_2, \dots, L_r \end{array} \right\} \textit{Explanans } S$$

$$\frac{\quad}{E} \textit{Explanandum}$$

where  $C, C_2, \dots, C_k$  are sentences describing particular facts, and  $L_1, L_2, \dots, L_r$  are universal quantified sentences asserting lawlike regularities on which the explanation will ultimately rest (this is identified as the *Explanans*  $S$ ). Finally,  $E$  is the conclusion that follows deductively from  $S$ , and will be identified as the *Explanandum*  $E$ .<sup>42</sup> Thus understood, the deductive nomological model explains in virtue of ‘nomic expectability,’ that  $E$  is expected to occur given  $S$ .

Let me illustrate Hempel’s account using his own analysis of P erier’s findings in the Puy-de-D ome experiment. This story begins in Italy, during the sixteenth century. At the time, the big engineering problem was to build suction pumps that would be able to lift water more than ten meters. In Rome, the amateur scientist Gasparo Berti set up a series of experiments to study this curious phenomenon in detail. His solution was to use a long tube, sealed at one end, and filled entirely with water; he immersed the open end in a tub also filled with water, and then stood the tube in a vertical position. As Berti noted, some of the water poured back into the tub, leaving a vertical column of water with a height of about ten meters in the tube. This was exactly the limiting height of the pumps that had puzzled the engineers. This fact triggered several other unanswered questions, such as why

the space at the top of the tube was empty, what stopped the water from pouring out, and what inhibited the water from pouring out completely into the tub? The answers were found in Florence, where Evangelista Torricelli replaced the water used by Berti with mercury, giving stability to the instrument. As it turned out later, it was the atmospheric pressure that stopped the mercury from completely pouring out of the tube (today's 'barometer'). Torricelli, of course, did not know this at the time.

Around 1647, Blaise Pascal understood that if the weight of the air is truly the driving force in all these experiments, the column of mercury in Torricelli's apparatus should be lower the higher the altitude, for the air above is thinner. The inverse is also true. Pascal then sent a letter to his brother-in-law Florin Périer describing the idea of an experiment: Périer had to measure the level of mercury in a tube at two different locations, his hometown Clérmont-Ferrand, and the peak of nearby Puy-de-Dôme; the difference in altitude between the two locations was about one thousand meters, which should have been enough for confirming (or refuting) Pascal's hypothesis. The story ends with Périer successfully showing that the level of mercury in a tube depends on the atmospheric pressure. The experiment was then repeated on different occasions and at different altitudes and, in every case, the results were very much alike. Naturally, such a wonderful and strange phenomenon needed an explanation.<sup>43</sup> In the twentieth century, the deductive nomological model was the only theory that could successfully provide an explanation of Périer's findings. Hempel explained it in the following way:

- (a) At any location, the pressure that the mercury column in the closed branch of the Torricelli apparatus exerts upon the mercury below equals the pressure exerted on the surface of the mercury in the open vessel by the column of air above it.
- (b) The pressures exerted by the columns of mercury and of air are proportional to their weights; and the shorter the columns, the smaller their weights.
- (c) As Périer carried the apparatus to the top of the mountain, the column of air above the open vessel became steadily shorter.
- (d) (Therefore,) the mercury column in the closed vessel grew steadily shorter during the ascent (Hempel, 1966, 301).

Hempel accounts for the explanation in the following way: the explanation of the phenomenon described by sentence (d) above "is just what is to be expected in view of the explanatory facts cited in (a), (b), and (c); and that, indeed, (d) follows deductively from the explanatory statement" (Hempel, 1966, 301). To explain a phenomenon, therefore, is to produce an argument whose conclusion is nomically expected. The rise in the mercury column is then explained because it is expected to follow from (a), (b), and (c), above.

As it turns out, nomic expectability is insufficient for yielding understanding of the phenomenon: explaining a storm by measuring the atmospheric pressure with a barometer is in no way an indication of the understanding of the occurrence of the storm. Hempel was well aware of this fact. In particular, he dismisses understanding as a philosophically relevant notion, restraining it (and, to the empiricist mind, degrading it) to the realm of the psychological or the pragmatic.<sup>44</sup>

Henk De Regt and Dennis Dieks argue that Hempel makes use of a notion of understanding that presupposes a subjective viewpoint. According to the authors, understanding is taken in the sense that it concerns a particular purpose of the person who uses it. Such a notion of understanding is also pragmatic in the sense that it implies the possibility of disagreement and variation of opinion based on contextual differences. The example provided by De Regt and Dieks is the group of physicists in the Copenhagen-Göttingen circle who believed that atomic phenomena could be understood with the theory of matrix mechanics, while Erwin Schrödinger and others disagreed.<sup>45</sup> As a good positivist, Hempel rejected the philosophical importance of such a subjective and pragmatic notion.

For me, and as I discussed earlier in Section 4.2.2, understanding is a condition for our general scientific comprehension of the empirical world and, as such, is not subject to the idiosyncrasy of individuals, but to the reinforcement of our general corpus of scientific beliefs. Understanding is an objective, mind-independent notion, and as such must be conceptualized throughout this work.

In more general terms, failing to incorporate the notion of understanding into the theory of explanation was not the only objection to Hempel. The *locus classicus* is the symposium on the structure of scientific theories held in Urbana in 1969 and compiled by Frederick Suppe.<sup>46</sup> Despite the many attacks assembled in that book, the Hempelian model of explanation remains a genuine source of inspiration for any theory of scientific explanation. Unfortunately the nomological deductive model cannot work as the conceptual framework that I need for computer simulations because it lacks three features that I seek in a theory of explanation, namely:

1. The relationship between *mathematics* and *the sciences*. Hempel assumes that mathematics and science are two distinct activities and, as such, they should be treated, studied, and analyzed separately.<sup>47</sup> Such a distinction presupposes a methodological dualism that isolates scientific explanation, as a general concept, from mathematical explanation of physical phenomena, as a more local discipline. As a result, many philosophers reject Hempel's dualism for it threatens to hide features that make mathematics interesting for explaining physical phenomena (see Section 4.3.3 for a discussion on this is-

sue). Kitcher, for instance, has rejected the conceptual distinction between mathematics and science, arguing that mathematical knowledge is no different from any other aspect of scientific knowledge and, therefore, there is no basis for a methodological division between mathematics and the natural sciences.<sup>48</sup> From the point of view of computer simulations, there are cases in which mathematical explanation plays a central role. Consider for instance a simulation that implements Snell's law concerning the refraction of light as it passes from one medium to another, such as the simulation of the rainbow. Now suppose that we want to explain why this simulated rainbow shows bands divided into colors with red, orange, and yellow appearing at the top, followed by green, blue, indigo, and violet at the bottom; or that we want to explain why the angle of the curvature of the simulated rainbow remains the same as a simulated observer moves towards or away from it. With an appropriate theory of mathematical explanation by computer simulations one should be able to answer these questions.

2. *Empirical adequacy and empirical content.* Hempel's nomological deductive model requires that, in order for the statements in the explanans to be true, they must: (a) correspond to states of affairs in the world (i.e., empirical adequacy condition) and, (b) they must have empirical content. As for the empirical adequacy condition, I hold some metaphysical differences with Hempel. Particularly, he claims for full reference of theoretical terms,<sup>49</sup> while I take reference naturalistically, that is, as a relation among scientists, natural kinds, and scientific method.<sup>50</sup> Therefore, a computer simulation represents a target system, and that representation holds in virtue of schemata that correctly capture objective dependencies, concepts that single out natural kinds, and so forth.<sup>51</sup> As for the condition of empirical content, 'feeding' computer simulations with empirical data, while a common practice, is not a condition for a successful simulation. Indeed, there are plenty of examples in the specialized literature of reliable computer simulations that are not dependent on empirical data as their input. In this sense, computer simulations do not require empirical content of any kind.
3. *Time dependency.* Hempel sought a time-independent and science-independent account of scientific explanation. But our theories and models are constrained by the knowledge we have at a certain moment in history, making progress as we explain and predict more things about the world. In this sense, scientific knowledge progresses as science and technology do, shaping our view of the

world. The theory of explanation used for computer simulation needs to take this time-dependency of theories and models into account.

Having abandoned the deductive nomological model of scientific explanation, I now continue my analysis of theories of explanation by discussing mathematical explanation of physical phenomena.

### 4.3.3 Mathematical explanation

Mathematics, just like computer simulations, represents a host of target systems by means of abstracting, idealizing, and fictionalizing the intended target system's properties, attributes, structures, and relations. It is natural to ask, then, whether explanation in mathematics provides the necessary theoretical framework for computer simulations. Let me now explore this possibility.

Many philosophers conjecture that mathematics can explain empirical phenomena in analogous ways just as scientific explanation does. Michael Resnik and David Kushner observe that mathematical explanation can be conceptually close to scientific explanation:

First, if to explain something is to give a systematic account of it, then mathematicians do explain: many of the greatest achievements in mathematics, such as Euclid's axiomatization of geometry or the Newton-Leibniz calculus, are systematic accounts of previously scattered results. [...] Second, it is certainly appropriate to ask why-questions concerning mathematical entities and results - these are at least symptomatic of requests for explanation. Moreover, many of the answers to such questions closely resemble the answers to requests for scientific explanations. (Resnik and Kushner, 1987, 151)

What Resnik and Kusher are pointing out is that explanations provided by mathematics are equally valid to those provided by science for explaining physical phenomena.

However, within mathematics an explanation seeking question can take different forms: it is not the same to ask why a given function is equal to zero, as it is to ask why a physical phenomenon behaves in the way it does when the function that it represents equals zero. The first question is about a mathematical property to be explained within the domain of mathematics, whereas the second question is about a mathematical property that holds some correlation with the empirical world. In plain words, we must distinguish between a mathematical explanation within mathematics, and a characteristically mathematical explanation of a physical fact. Only the latter is of interest for our work.

Consider the following case as an example of a mathematical explanation of a physical fact:

Suppose that a bunch of sticks are thrown into the air with a lot of spin so that they twirl and tumble as they fall. We freeze the scene as the sticks are in free fall and find that appreciably more of them are near the horizontal than near the vertical orientation. Why is this? The reason is that there are more ways for a stick to be near the horizontal than near the vertical. To see this, consider a single stick with a fixed midpoint position. There are many ways this stick could be horizontal (spin it around in the horizontal plane), but only two ways it could be vertical (up or down). This asymmetry remains for positions near horizontal and vertical, as you can see if you think about the full shell traced out by the stick as it takes all possible orientations. This is a beautiful explanation for the physical distribution of the sticks, but what is doing the explaining are broadly geometrical facts that cannot be causes. (Lipton, 2004, 31-32)

As Peter Lipton points out, this is an explanation of a physical phenomenon using geometrical, non-causal facts about properties of the sticks in free fall. Such an explanation makes use of mathematical procedures, objects, attributes of these objects, relations among them, and the like; it is an explanation that ignores (and is required to ignore) various physical details about the system of interest. With this in mind, we can now answer the fundamental question about mathematical explanation of physical phenomena; namely: “how is mathematics applied in scientific explanations and descriptions?” (Shapiro, 2000, 36).

In 1978, Mark Steiner published his first work on mathematical explanation of physical phenomena. In that work he states that every scientific explanation requires mathematical, as well as physical truths. For instance, the physical space is a three-dimensional Euclidean space, and the rotation of a rigid body around a point generates an orthogonal transformation. In Steiner’s mind, then, mathematical explanations of physical phenomena consist in a typical mathematical proof once all the physical concepts have been *removed*. In other words, in order to explain  $p$ , one must first remove all the terms in  $p$  referring to physical properties and then carry out a typical mathematical proof.<sup>52</sup> Steiner, then, endorses the idea that once the terms referring to physical properties have been removed, all that remains is the mathematical explanation of a mathematical truth.

There is an important metaphysical assumption here consisting in deleting the boundaries between explanation of physical phenomena and explanation in mathematics by ‘removing the physics.’ In addition, the continuity between the natural sciences and the mathematical sciences is grounded in methodological likeness: both describe an objective world of entities, both make use of methods that are remarkably similar, and one can be subsumed under the other by ‘mathematical analogies’ that allows for the removal of the physical concepts. Basically, a ‘mathematical analogy’ is a way to bring the realm of mathematics into the realm of physics, since both share equivalent properties (e.g., the linearity of a second order equation is ‘analogous’ to the applicability of the principle of superposition in

physics<sup>53</sup>).

Thus understood, mathematical explanation has few things to offer to computer simulations. First, it is very unlikely that one can perform a mathematical proof of a simulated phenomenon. And even if such a proof is possible, it would most likely take an incredible amount of time to make it humanly feasible. After all, computer simulations are powerful tools for systems otherwise cognitively inaccessible. Second, mathematical properties of physical entities cannot always be accurately represented in computer simulations; therefore, it is not true that one can remove the physical terms in order to carry out an explanation. Steiner imposes unacceptable conditions for a theory of mathematical explanation of physical phenomena.

A more suitable account seems to be Robert Batterman's *asymptotic explanations*. According to this author, explanations here "illuminate structurally stable aspects of a phenomenon and its governing equations" (Batterman, 2002, 59) by means of highly sophisticated mathematical manipulations. Similarly to Steiner, Batterman considers that we have reasons for ignoring many details of a physical system that we cannot control. In an asymptotic explanation, then, the explanatory power is given by systematically 'throwing away' the various causal and physical details. However, unlike Steiner, this is not done by 'removing the physics,' but rather by abstracting and idealizing the stable aspects of the phenomenon.

The way asymptotic reasoning explains, therefore, is by yielding understanding of the transformations of one mathematical formula into another. These transformations are carried out by means of taking the limit of some quantity, often 0 or infinity. Batterman illustrates his account with a couple of examples: one is flow, and the other is heat.<sup>54</sup> Allow me to omit the details of these examples and get directly into the objections to Batterman's account.

Christopher Pincock objects that taking the limit of a function can have positive as well as negative effects. The positive effect is that we eliminate from the representation the irrelevant details, while the negative effect is that in the transition from one representation to another, we also lose relevant representational aspects of the phenomenon.<sup>55</sup> Batterman's account also suffers from a serious drawback that makes it uninteresting for computer simulations: it leaves unexplained cases where asymptotic explanation cannot be applied. Consider applying asymptotic explanation to Lipton's example: there are no limits to take in the function describing the position of the sticks. It follows that no explanation is possible. Just like my objection to Steiner, Batterman suggests a kind of explanation that works only for specific cases, and is therefore unappealing for computer simulations.

There are many other interesting accounts of mathematical explanation, such as the *mapping account* championed by Pincock,<sup>56</sup> or its ‘refined’ version, the *inferential account* defended by Otávio Bueno and Mark Colyvan.<sup>57</sup> Unfortunately, these accounts share similar assumptions that make them inapplicable or unappealing for explaining simulated phenomena. Take for instance the usual metaphysical assumption that explanation depends solely on the mathematical machinery that governs the phenomenon. In this sense, it assumes that the phenomenon under study is governed by certain mathematical structures, and that an explanation is possible because there is a suitable language in place. However, such an assumption limits explanation to a pure mathematical language, forcing the conclusion that explanation in computer simulations is only possible in mathematical terms. It follows that a non-mathematical explanation could not be used for explaining simulated phenomena, a clearly undesirable consequence.

Restricting explanation to a purely mathematical domain seems to be too high a price for the epistemology of computer simulations. The kind of computer simulations we are interested in cannot always yield mathematical insight into the phenomenon that is being simulated (albeit that simulation yields insight into such phenomenon). Moreover, the computer simulation would not be responsible for carrying out the explanation since the explanation itself is located somewhere else: whether on the notion of *asymptotic reasoning* (in Batterman) or in the notion of *proof* (in Steiner). Hence, we cannot properly grant computer simulations explanatory capacity.

Despite these objections and drawbacks, we still need to consider mathematical explanation as important for computer simulations. This is especially important because many models simulated are mathematical models, and an account of explanation in computer simulation needs to be powerful enough to include these models as well.

Now suppose for a moment that instead of a *complete* mathematical explanation of phenomena, the explanation is only *partial*, that is, only a part of the explanation relies on mathematics whereas other equally relevant aspects rely on some other domains. Consider the example of the hexagonal structure of hive-bee honeycombs in evolutionary biology. The nature of the problem is contrastive, that is, why hive-bees chose to create honeycombs with a hexagonal structure rather than any other polygonal figure or combination. Part of the explanation depends on mathematical facts, part on evolutionary facts. Bees that use less wax and thus spend less energy have a better chance at evolving via natural selection. The explanation is completed

by pointing out that “any partition of the plane into regions of equal area has a perimeter at least that of the regular hexagonal honeycomb tiling” (Hales, 2001, 1), or “the existence of a honeycomb tiling depends on the shape and size of the lattice” (Hales, 2001, 7). Thus, the hexagonal tiling is optimal with respect to dividing the plane into equal areas and minimizing the perimeter.

This fact, known as the ‘honeycomb conjecture,’ shows how mathematical explanation of physical phenomena and classic scientific explanation depend on each other. It also motivates the claim that a successful theory of explanation for computer simulation should account for both of these kinds of explanation, given the very nature of computer simulations themselves. Here lies another reason why I believe that the unificationist account is the most suitable candidate for simulations. According to this latter account, scientific explanation is treated in the same logical way as mathematical explanation of physical phenomena, merging into one account two allegedly different kinds of explanation (recall my first objection to Hempel’s deductive nomological model). In this vein, simulated phenomena could be explained regardless of their mathematical nature or their physical characteristics. I discuss the unificationist account in more detail in Chapter 5.

Let me turn to another interesting account of explanation: model explanation. Given the model-like nature of computer simulations, it is also natural to envisage it as a possibility.

#### 4.3.4 Model explanation

Alisa Bokulich has written extensively on model explanation. She identifies three approaches present in the literature:<sup>58</sup> i) the *mechanistic model explanation*; ii) the *covering-law model explanation*; and iii) the *hypothetico-structural explanation*. Bokulich also has her own account, i.e., *structural model explanation*. Following her work on model explanation, I discuss these four approaches in the context of computer simulations.

The *mechanistic model explanations* takes the explanatory power of models to be about describing the mechanisms of the target system. Carl Craver, for instance, conceives the mechanistic model of explanation as consisting of two elements: first, a *description* of the behavior of some of the underlying mechanism of a given phenomenon; second, a *correspondence relation* between those mechanisms and the descriptions in the model.<sup>59</sup> In this way Craver elucidates the mutual dependency and correlation between the description of the mechanism (in the model) and the underlying structure of the phenomenon (in the target system). Furthermore, Craver elab-

orates on an additional distinction between ‘how-possible models’ and ‘how-actually models’ as a way to identify models that only *purport* to explain and those that *actually* explain. The difference is mostly grounded in the metaphysical assumptions built into the model. A ‘how-possible model’ would loosely describe the behavior of the underlying mechanism that produces the phenomenon, whereas ‘how-actually models’ would describe (and correspond to) real components, relations, and so forth in the mechanism that produces the phenomenon.<sup>60</sup> A genuine model explanation, therefore, is one where the model correctly and accurately reproduces the actual mechanism of the target system (i.e., a ‘how-actually model’). To Craver’s mind: “a mechanistic explanation must begin with an accurate and complete characterization of the phenomenon to be explained” (Craver, 2006, 368).

Bokulich objects that these requirements for a model to be explanatory are too strong, for one rarely has a complete and accurate description of the phenomenon to explain.<sup>61</sup> This is a fair objection that concerns not only all the theories of explanation for models, but also a theory of explanation for computer simulations, where a complete and accurate representation of the target system is rarely, if ever, achieved (or at least to the degree demanded by Craver). In addition, Craver can be interpreted in the context of Kim’s internalist/externalist distinction as taking an externalist relation grounding the explanation. Indeed, to Craver’s mind, in order to have a successful explanation, a “complete characterization of the phenomenon” (Craver, 2006, 368) must be provided; that is, there must exist an objective correspondence relation grounding the explanation. The mechanistic model explanation is, by definition, explanatory externalist and, therefore, of no use for computer simulations.

The *covering law-model explanation*, as elaborated by Mehmet Elgin and Elliott Sober, is restricted to models in evolutionary biology known as ‘optimality models’; that is, models that describe the value of a trait that maximizes fitness, given a set of constraints.<sup>62</sup> The most significant drawback of the covering law-model explanation is its *time dependency* on actual models of evolutionary biology. Computer simulations require a more adaptable theory of explanation, one that allows for time- and science-independence; that is, one whose explanatory relation does not depend on the way science conceptualizes the empirical world in one moment in history, but on general descriptions of the relations holding between explanans and explanandum. Hence, the covering law-model fails to endure as a good candidate for computer simulations.

The *hypothetico-structural explanation*, as elaborated by Ernan McMullin, explains the properties of a complex entity by postulating an underlying structural model, whose features are causally responsible for the properties to be explained.<sup>63</sup> In an explanation, the original model must successfully capture the real structure of the object of interest. For a *successful* explanation, however, one must first justify the model by *de-idealizing* it, that is, ‘adding back in’ those features that have been omitted by the process of idealization.<sup>64</sup> Now, it is precisely this process of de-idealization that makes the hypothetico-structural explanation unappealing for computer simulations. One interesting use of simulations is their capacity to represent systems about which we might only have some theoretical knowledge, but no real empirical content to ‘add back in.’ As if that were not enough, McMullin’s model of explanation would beg the question of how to ‘add back in’ features to intricate and complex systems such as computer simulations. I then conclude that the hypothetico-structural explanation presents unattainable demands for computer simulations.

Finally, Bokulich’s *structural model explanation* is a modification of James Woodward’s theory of *counterfactual dependence*, although applied to models. According to Bokulich, for a model to be successfully explanatory, three conditions must hold:

First, the explanans makes reference to an idealized or fictional model; Second, that model explains the explanandum by showing that the counterfactual structure of the model is isomorphic (in the relevant respects) to the counterfactual structure of the phenomenon. This means that the model is able to answer a wide range of “what-if-things-had-been-different” questions. And third, there is a justificatory step specifying what the domain of applicability of the model is and that the model is an adequate guide to that domain of phenomena (Bokulich, 2011, 43).

It is the second condition that seems to contribute to a new view on model explanation. However, it entails two problems that Bokulich needs to give an answer to: first, the nature of the isomorphic relation between the model and the target system is contentious. Although Bokulich is careful enough to say that this notion is being used in a loose sense,<sup>65</sup> it is not clear how isomorphism can be handled in such a way that it does not run into the typical problems of isomorphic representation. Indeed, isomorphic relations are objected to on grounds of not being sufficient (i.e., representation might fail even if isomorphism holds) nor necessary for representation (i.e., representation is possible even if isomorphism fails), and for not having the right formal properties (i.e., isomorphism is symmetric, reflexive, and transitive while representation is not), among others. Speaking of isomorphism in the context of simulation models seems to be highly contentious and inadequate.

Secondly, the ‘what-if-things-had-been-different’ kind of questions that Bokulich needs to be answered are inspired by Woodward’s *counterfactual dependence*, which can be interpreted in the following way: the explanation must allow us to see what would have been different for the explanandum if the factors cited in the explanans had been different.<sup>66</sup> It is not clear, however, that such an account of explanation is viable for computer simulations. Allow me to elaborate.

Let me begin by illuminating the notion of counterfactual dependence using Woodward’s example. Consider an explanation of why the magnitude of the electric intensity (force per unit charge) at a perpendicular distance,  $r$ , from a very long fine wire with a positive charge is given by the expression  $E = \frac{1}{2\pi\epsilon_0} \frac{\lambda}{r}$  (Coulomb’s law), where  $\lambda$  is the charge per unit length on the wire and  $E$  is at right angles to the wire.<sup>67</sup> Woodward’s proposal is that explanation is a matter of exhibiting systematic patterns of counterfactual dependence, that is, what would have happened if the initial and boundary conditions had been different.<sup>68</sup> To illustrate this last point consider that the above expression  $E$  shows how, altering the charge per unit  $\lambda$ , one obtains either a repulsive or an attractive system. For a positive charge, the direction of the electric field points along lines directed radially away from the location of the point charge, while for a negative charge the direction is the opposite. Therefore,  $E$  above shows how the field would change if the charge along the wire, the distance, or any other initial or boundary condition were increased or decreased (among other interventions that one can perform on the system). “We learn from [this example]”, says Woodward, “that despite the fact that Coulomb’s law is an inverse square law, the field produced by a long straight wire falls off as the reciprocal rather than the square reciprocal of the distance from the wire” (Woodward, 2003, 191).

The problem with this approach is simply a matter of practicality. It is humanly impossible to explain a simulated phenomenon by exhibiting what would have happened if the initial and boundary conditions of the computer simulation had been different. Scientists run simulations precisely because they are unable to anticipate the results, and are therefore unable to know what would have happened if conditions had been different. Believing otherwise is to misinterpret the uses of computer simulations in the sciences. I conclude that neither Bokulich’s model of explanation nor Woodward’s counterfactual dependence are suitable accounts of scientific explanations for computer simulations.

## 4.4 Conclusion

In this chapter, I claimed that the epistemic power of computer simulations could be defended by an analysis of their explanatory role. This chapter, then, aimed to analyze some of the theoretical assumptions behind any successful theory of scientific explanation. It also aimed at disregarding theories of explanation that, for one reason or another, would have failed to explain simulated phenomena.

For achieving all this, I discussed the minimum set of conditions for a *reliable* computer simulation, that is, the set of conditions under which a computer simulation genuinely represents an empirical target system. This, as I argued, is an indispensable condition for the success of computer simulations as explanatory devices. Finally, I showed that computer simulations did not represent a new theory of scientific explanation. Instead, they must be analyzed within the context of a well-established theory. For these purposes, I discussed three *explanatory internalists* accounts: Hempelian deductive nomological model, theories of mathematical explanation of physical phenomena, and theories of model explanation. I argued that these accounts lack the theoretical features needed by computer simulations to be explanatory devices. I then claimed, without actually showing, that the *unificationist account*, as elaborated by Philip Kitcher, was the only explanatory internalist account suitable for computer simulations. The object of the next chapter, then, is to prove this last claim.

## Notes

<sup>1</sup>Let me restate that my interests are focused on scientific explanation and, as such, I am only interested in explaining the physical world. Pragmatic theories of explanation may be interpreted as providing a different metaphysics than realist theories (van Fraassen, 1977). Unfortunately, pragmatic theories not only fail to explain basic science, but more importantly, take scientific explanation as non-epistemic (Churchland and Hooker, 1985; Kitcher and Salmon, 1987).

<sup>2</sup>Cf. (Hempel and Oppenheim, 1948). Note that explanation here is understood as a *product*, and there is no interest in *acts of explanation* (Achinstein, 1983).

<sup>3</sup>Here I am only concerned with propositional knowledge, that is, knowledge that such-and-such is the case. Propositional knowledge must be distinguished from other kinds of knowledge, such as knowing a place, a person, or how to do something.

<sup>4</sup>(Goldman, 1979).

<sup>5</sup>For each term, see (Kitcher, 1989), (Salmon, 1998, 159), and (Hempel, 1965) respectively.

<sup>6</sup>See (Hempel, 1965; Kitcher, 1989), (Salmon, 1998, 159), and (van Fraassen, 1980, 134) respectively.

<sup>7</sup>My interests in explanation are exclusively on particulars, as opposed to explanation of laws.

<sup>8</sup>Cf. (Ruben, 1992, 160). Since I am only concerned with scientific explanation in the natural sciences, I take this claim to be unproblematic.

<sup>9</sup>Note that here I focus solely on explanation of empirical systems. In Section 6.2.1 I discuss further how to approach explanation of non-empirical systems.

<sup>10</sup>Alisa Bokulich (Bokulich, 2010) presents an interesting work on the explanatory power of fictions. However, her account requires minimal degrees of representation from the model. I believe that she would agree that an entirely false model such as the Ptolemaic model has no explanatory value whatsoever.

<sup>11</sup>For Belousov-Zhabotinsky simulations, see (Field and Noyes, 1974). For the simulations to be *stiff*, cf. (Krohs, 2008, 281).

<sup>12</sup>An interesting work on the exploratory power of computer simulations can be found in (García and Velasco, 2013).

<sup>13</sup>The notion of ‘structure’ here is broadly construed. Ontic theories take it as the causal relationships that bring the phenomenon about and that can be tracked back. Unificationists, on the other hand, take it as patterns of behavior common to many phenomena.

<sup>14</sup>The justification condition only requires that the belief that qualifies as knowledge has the *property* of being justified. This rules out the possibility of arguing that an agent must engage in the activity of justifying the belief. See also (Moser, 2002; Steup and Sosa, 2005).

<sup>15</sup>See Gettier’s second counterexample to the conception of knowledge as justified true belief in (Gettier, 1963).

<sup>16</sup>See also, for instance (Goldman, 1988, 2009).

<sup>17</sup>Cf. (Goldman, 1999, 28).

<sup>18</sup>Goldman refers to his position as ‘Historical Reliabilism,’ for he emphasizes a process of belief-generation from a historical perspective.

<sup>19</sup>I have selected these two examples to show that a reliable process can be purely cognitive, as in a *reasoning process*; or external to our mind, as the example of a tree outside my window shows.

<sup>20</sup>Under this reliabilist account, neither skepticism in the form of the evil genius as presented by Descartes (Descartes, 1996 (1641), nor ‘epistemic luck’ (Radford, 1966) are sound arguments.

<sup>21</sup> See also (Gillies, 2000).

<sup>22</sup> See, for instance, the works of (Kvanvig, 2003, 2009; Zagzebski, 2000).

<sup>23</sup>The notion of ‘good representation’ stems from the working definition on *goodness of representation* as discussed in Section 1.2.2.1. The general idea is that what is represented in the simulation model does not differ significantly from the way the world is (or at least how it is understood by the sciences). This notion must be complemented with my discussion on implementing scientific models as computer simulations in Chapter 2 and Chapter 3.

<sup>24</sup>This point has been discussed in Section 3.3.2 and Section 3.3.3.

<sup>25</sup>There has been some work done where it is possible to claim for an agent to have understanding while lacking knowledge of the relevant propositions; for instance, (Elgin, 1996, 2004, 2009) and (Riggs, 2002).

<sup>26</sup>I analyze the character of this naturalist viewpoint in more depth in Section 5.2. For now, we can take a ‘true belief’ as representations that ‘match nature.’ As for the notion of ‘matching nature,’ it is enough to say that scientific practice can provide justifying arguments for concluding that the actual world is not likely to be significantly different from an ideal world that is represented in models and theories. Cf. (Kitcher, 1989, 453).

<sup>27</sup>Cf. (Elgin, 2007, 39).

- <sup>28</sup>Of a similar opinion are Henk De Regt and Dennis Dieks (De Regt and Dieks, 2005, 151).
- <sup>29</sup>Cf. (Elgin, 2009, 324).
- <sup>30</sup>Let me note here that I am not interested in *acts* of explaining, but rather in the explanation as a *product*. Hence, there will be no differences between talking of ‘theories, models, or computer simulations explaining something’ and ‘an agent using a theory, model, or computer simulation for explaining.’ In both cases the emphasis is meant to be on the logic of explanation, not on the agent doing the explanation.
- <sup>31</sup>This example is brought up by Michael Scriven (Scriven, 1970) when objecting to Hempel’s account. In (Hempel, 1965, 348), Hempel answered that such examples are not of concern for scientific explanation.
- <sup>32</sup>Note that here I made no difference between explanation of particulars (e.g., puerperal fever) and explanation of laws (i.e., the second law of thermodynamics). These explanations represent two different problems for the philosopher of science, and must be treated accordingly. In this work I only focus on explanation of particulars.
- <sup>33</sup>See (Hempel, 1965; Hempel and Oppenheim, 1948).
- <sup>34</sup>For the *statistical relevance model*, see (Salmon et al., 1971). For the *ontic approach*, see (Salmon, 1984). For the *pragmatic approach*, see (van Fraassen, 1980). Finally, for the *unificationist approach*, see (Friedman, 1974; Kitcher, 1981).
- <sup>35</sup>Also known as *explanatory realism* (Kim, 1994, 273).
- <sup>36</sup>In this work I do not make differences between ‘an explanation *yields* understanding of a phenomenon’ and ‘we *gain* understanding by explaining a phenomenon.’ Both sentences refer to the same epistemic product of an agent gaining insight into the way the world is.
- <sup>37</sup>It is worth restating here that I do consider computer simulations as methodologically and epistemically novel.
- <sup>38</sup>For each explanatory relationship see, for instance (Salmon, 1984; Woodward, 2003), (Hempel, 1965; Kitcher, 1989), (van Fraassen, 1980), Craver (2006); Machamer (2004) and (Pincock, 2010) respectively.
- <sup>39</sup>(Machamer, 2004).
- <sup>40</sup>Cf. (Krohs, 2008).
- <sup>41</sup>Also known as *Covering Law Model*.
- <sup>42</sup>Cf. (Hempel, 1965, 336).
- <sup>43</sup>At the time of Pascal and P erier, the only available model of explanation was Aristotle’s. Unfortunately, this phenomenon cannot be explained by his account because, as Aristotle believed, the existence of a vacuum in nature was absurd (*horror vacui*) (Aristotle, 1936).
- <sup>44</sup>Cf. (Hempel, 1965, 413).
- <sup>45</sup>Cf. (De Regt and Dieks, 2005, 140-141).
- <sup>46</sup>See (Suppe, 1977).
- <sup>47</sup>Cf. (Hempel, 1945, 7).
- <sup>48</sup>Cf. (Kitcher, 1989, 423).
- <sup>49</sup>Cf. (Hempel, 1965, 101)
- <sup>50</sup>See my discussion on this issue on Section 5.2.
- <sup>51</sup>Cf. (Kitcher, 1993, 150).
- <sup>52</sup>Cf. (Steiner, 1978, 19).
- <sup>53</sup>Cf. (Steiner, 1989, 453).
- <sup>54</sup>The examples are found in (Batterman, 2002, 37) and (Batterman, 2002, 46) respectively.

<sup>55</sup>See (Pincock, 2011, 4).

<sup>56</sup>(Pincock, 2011).

<sup>57</sup>(Bueno and Colyvan, 2006).

<sup>58</sup>Cf. (Bokulich, 2008, Chapter 6)

<sup>59</sup>Cf. (Craver, 2006, 361). See also (Machamer, 2004).

<sup>60</sup>Cf. (Craver, 2006, 361).

<sup>61</sup>Cf. (Bokulich, 2011, 34).

<sup>62</sup>Cf. (Elgin and Sober, 2002, 446).

<sup>63</sup>Bokulich renames it the *causal model explanation*, but here I keep the original *hypothetico-structural explanation* as coined by McMullin (McMullin, 1978).

<sup>64</sup>Cf. (McMullin, 1978, 139).

<sup>65</sup>Cf. (Bokulich, 2011, 43, footnote 16).

<sup>66</sup>Cf. (Woodward, 2003, 11).

<sup>67</sup>Cf. (Woodward, 2003, 187ff) for the complete details of the example.

<sup>68</sup>Cf. (Woodward, 2003, 191).

# Chapter 5

## Scientific explanation by computer simulations

### 5.1 Introduction

The initial motivation of this dissertation was to defend the epistemic power of computer simulations. I claimed that such a defense could be achieved by showing how computer simulations explain simulated phenomena. In previous chapters I diligently put the pieces together. Let me now briefly revisit some of the results obtained and indicate what role they play in this chapter.

Let us begin with the shift proposed in Chapter 2 consisting in analyzing computer simulations from the viewpoint of the philosophy of computer science. This shift allowed the introduction of three units of analysis, namely, the *specification*, the *algorithm*, and the *computer process*. All three concepts proved to be central for the construction of a working conceptualization of computer simulation, as entrenched in Section 3.3.1.1. Such a working conceptualization, together with the discussion on *verification*, *validation* (see Section 3.3.2), and *errors* in computers (Section 3.3.3) are at the core of the process of explaining in computer simulations.

Equally important was to discuss current trends in the philosophical literature on scientific explanation, as carried out in Chapter 4. As a result, several theoretic conclusions were reached. Among the most prominent was that the metaphysics of explanation compel us to set two conditions for explaining in computer simulations, namely, the *goodness of representation* of the simulation model, and the correctness of computability of it (see Section 4.2.1). Also central was the distinction between *internalist* and *externalist explanatory accounts*, which reduced the number of theories of explanation to analyze to the former accounts; and the *metaphysical* and *ontological* questions which are at the core of any theory of explanation (see Section

4.2.2). As I have argued in Chapter 4, the unificationist account of explanation as elaborated by Kitcher is the most suitable theory for computer simulations. This chapter, then, has the responsibility to expatiate on the reasons why I hold such a view. It must also show how a computer simulation works as an explanatory device, and in what sense it yields understanding of the simulated phenomena (Kim's metaphysical and epistemic questions respectively).

Following these indications, this chapter is organized as follows: in Section 5.2 I address the diverse commitments of the unificationist. Traditionally, there are many dissensions on the idea of a unified world to the point that the unificationist (in all its variants) has fallen into disgrace. It is important, therefore, to reconstitute their good name. In this first section, I specify the kind of unificationist underlying Kitcher's theory (i.e., the *modest unificationist*). The following two sub-sections are a detailed discussion on how the modest unificationist answers Kim's metaphysical and epistemic questions. Section 5.3 addresses at face value computer simulations as an explanatory device. As anticipated, its explanatory power flourishes within the conceptual framework provided by the unificationist. The aim of this last section, then, is to answer Kim's epistemic and metaphysical questions within the context of explanation by computer simulations.

## 5.2 The unificationist account

The last and, to my mind, most suitable theory of scientific explanation for computer simulation is the *unificationist account*. Michael Friedman and Philip Kitcher are at the core of the contemporary development of this theory, and my findings will closely follow their work. Now, before getting into the details of why the unificationist account provides the most suitable conceptual framework for computer simulations (see Section 5.3), I first need to make the notion of *unification* more precise and clarify the characteristics of the unificationist account that make it attractive for framing computer simulations.

Originally, the notion of *unification* was applied to the organization of the sciences. In 1934, Otto Neurath launched the International Encyclopedia of Unified Science, an ambitious project that intended to derive all scientific disciplines from physics. The "Unity of Science Movement," as it was later known, took the unification of science as following a reductionist principle. That is, chemistry had fundamental concepts, laws, and principles from which biology could borrow and account for its own entities, process, and the like. Biology, in turn, reduced sociology, and physics reduced chemistry. The followers of this movement, then, conceived a hier-

archical structure where the different levels represented different ways of organizing the empirical world, with physics as its basis.

Although once the strong project of Logical Positivism, the Unity of Science Movement fell into disrepute as a general program for the sciences (see (C) below<sup>1</sup>). Today, philosophers are unsympathetic to the idea that science aims to unify a diversity of phenomena. John Dupré, for instance, has written extensively on this topic denying that science constitutes a single, unified project. According to him “[t]he metaphysics of modern science, as also of much of modern Western philosophy, has generally been taken to posit a deterministic, fully law-governed, and potentially fully intelligible structure that pervades the material universe” (Dupré, 1993, 2).

A more modern construction of the *unificationist*, as defended by Friedman and Kitcher, and to which I ascribe, takes a more moderate position than the Unity of Science Movement, without ceding to a view of a local<sup>2</sup> and contrived scientific practice. Alison Wylie distinguishes a number of theses that identify this modern unificationist.<sup>3</sup> Kitcher reconstructs Wylie’s theses in the following way:

- (A) *Unity of scientific method*: all sciences share a common set of methods and standards for appraising hypotheses and theories, and for specifying what counts as explanation.
- (B) *Ontological unity of science*: the entities studied by the various sciences form a compositional hierarchy, with the fundamental entities of physics at the base and the subject-matter of chemistry, biology, and so forth at successively higher levels.
- (C) *Reductive unity of science*: the sciences form a hierarchy with theories at adjacent levels related by relations of reduction.
- (D) *Horizontal integration of sciences*: principles from different theories can often be combined to construct inter-field theories, establishing a network of links among the sciences.
- (E) *Confirmational disunity of science*: it’s often important for the confirmation of scientific claims that the sciences be disunified, in the specific senses that some of their principles describe causally independent processes, are justified independently of one another, and belong to different autonomous sciences (Kitcher, 1999, 343)

The notion of unification defended by Kitcher, then, is related to theses (A), (D), and (E) above. Indeed, Kitcher rejects theses (B) and (C) as they reconstruct science in a hierarchical fashion, failing to account for the current scientific enterprise.

The interest in theses (A), (D), and (E) lies in the fact that they are constitutive of Kitcher's idea of *modest unificationism*. Modest unificationism is not so much another “-ism” in philosophy as the attitude or belief that science is full of successful stories of conceptual and methodological unity. In this vein, modest unificationism is not a ‘movement’ trying to reduce the sciences to their fundamental basis, but rather it sees the practice of science as reflecting a (present or future) regulative ideal; that is, the ideal of finding as much unity as we can in our scientific concepts, scientific theories, and methodological principles by discovering perspectives from which we can fit a large number of apparently disparate empirical results into a small number of schemata.<sup>4</sup> Complete unification, as the proponent of the Unity of Science Movement demanded, is not an attainable (and probably not even a desirable) project for current science. Instead, the modest unificationist is content with taking unification as a conceptual and methodological *ideal* for the sciences. In this sense it is compatible with the belief that the world might be, after all, a diverse and disorderly place, and that our understanding of phenomena sometimes requires the use of concepts and theories that are not neatly integrated. Let me discuss each point in more detail in order to fully understand the principles of modest unificationism.

Thesis (A) above is compatible with modest unificationism because, despite the conspicuous senses in which different sciences have different methodologies, there are good programs in the philosophy of science that are methodologically successful in their unifying approaches. As an example take existing programs for scientific confirmation, such as Bayesianism, bootstrapping, and Neyman-Pearson, among others.<sup>5</sup> To this, one can also include the many accounts of scientific explanation present in current literature, such as ontic explanation, unification, and so forth,<sup>6</sup> all of which have methodological principles of unity.

Thesis (D), instead, aims at highlighting the ‘logical unification’ that exists when concepts, principles, schemata, and models of an established domain are extended for covering new phenomena in a (possibly) new domain.<sup>7</sup> This happens when the Lotka-Volterra model of predator-prey used in a biological system can also be applied in economics and the social sciences. Now, since the modest unificationist has no interest in ascribing to a reductionist viewpoint, there is no need to worry about fixing the number of new phenomena that must be extended to new domains.<sup>8</sup> In particular, the modest unificationist reserves the right to deny that every new phenomenon must find a pre-existing domain with which it must be unified. In plain words, the modest unificationist is not imposing conditions on the number of phenomena nor on the number of scientific domains that could be unified.

Finally, thesis (E) is the unificationist's answer to the claim that independent processes provide better evidence for the confirmation of phenomena than a single theory that unifies it. Let me explain this last point further. In his seminal book, *Representing and Intervening*, Ian Hacking claims that the use of multiple microscopes serves to convince us of the reality of the structures and entities that we see through them. And we are convinced because different microscopes using entirely different physical principles lead us to observe the same structures in the same entity.<sup>9</sup> This is indeed a strong argument in favor of independent confirmation of scientific entities. Scientists are, then, more justified in believing in the existence of some entity because there is a diversity of instruments that can provide independent observations of the same entity. Allan Franklin makes a similar point as Hacking. To Franklin's mind, good experiments are those in which scientists can measure a quantity by a multiplicity of independent measurements, such as by a meter, the wavelength of the same radiation using laser techniques, etc.<sup>10</sup> Independent confirmation of the same measurements, therefore, relates to autonomous methods for accessing the same scientific entity, which in turn supports the thesis of disunity and disintegration of the sciences.

Examples like these can be multiplied in the philosophical literature since scientists better justify the results of their experiments by relying on a sufficiently large number of independent confirmations.<sup>11</sup> The problem set for the modest unificationist is precisely that 'independent confirmation' is an argument in favor of the disunity of science. How could the unificationist reverse this claim? How could it be shown that independent confirmation is not a methodological principle that opposes unification? Admittedly, the answer might not satisfy everyone and, without the proper conviction that unification is indeed possible and desirable in the sciences, one could even take the answer as weak and unjustified. Having said that, Kitcher believes that unification is achieved by subsuming epistemically independent principles that guide the process of confirmation of the entity (e.g., for microscopes, it could be optical, electronic, fluorescent, and so forth) into a common unifying framework that maximizes explanatory and predictive success. The underlying idea is to put the modest unificationist credo into practice: the epistemic principles by which one observes different entities can be accepted as independent of one another, but there is no methodological requirement by which independent confirmation cannot be unified into a common framework for greater explanatory or predictive success.<sup>12</sup>

Unification, then, is conceived as a *regulative ideal*, and as such it goes into the future, as guidance for scientific practice: "the local knowledge of today is a spur to the unification of tomorrow" (Kitcher, 1999, 348). In metaphorical terms, the modest

unificationist finds consolation in the idea that agnosticism is more appropriate than atheism when it comes to expecting that the future development of scientific practice will unify phenomena that now escape the domain of reigning theories.

The modest unificationist is looking into the future at the same time that she keeps an eye on the present aims and achievements of science. It is at this point where Kitcher's *naturalism* and *real realism* converge. Indeed, Kitcher conceives real realism as a way to eschew any strong metaphysics about the natural world; that is, he rejects any assumptions about the empirical world as divided into objects and kinds, or as having its own language or agenda.<sup>13</sup> The truth about physical reality, instead, is to be understood in terms of the Tarskian machinery, that is, the notion of truth is reduced to the notion of reference (e.g., 'the snow is white' if and only if the snow is white) whereas reference here is understood naturalistically as a relation between scientists and the empirical world.<sup>14</sup> Naturalism, on the other hand, is interpreted as a cluster of ideas that includes epistemological, methodological, ontological, and axiological viewpoints. For my purposes, it is enough to analyze Kitcher's naturalism from two of these viewpoints: the epistemological and the methodological. Kitcher's naturalism is methodological in the sense that the natural sciences constitute all of our understanding of the world. To this view, there are no 'spooky' entities that science cannot, eventually, report. Scientists, of course, cannot represent reality to the smallest detail. This is why Kitcher's methodological view takes that the actual world is not likely to be significantly different from the ideal world represented in a model, a theory, or an experiment.<sup>15</sup> To the modest unificationist, then, the world does not appear as something mysterious, or magical, or supernatural; and neither does the way scientists represent it.

Kitcher's naturalism and real realism are philosophically sound positions because they stem from successful scientific practice. The unity behind implementing the same methods of investigation, theoretical interdependence, and the fact that the sciences share similar epistemic principles, are among the conceptual evidence that Kitcher uses for grounding his claims. In particular, Kitcher indicates that there is a strong methodological component in the ways we investigate reality, which have their roots in the general authority of the scientific method, in the interaction of scientific communities, and the personal professional development. The details of the many topics mentioned here will not be provided. I am not interested in deepening the discussion on the epistemic and methodological access to reality for it is far too complex a subject to be addressed here. Neither I am interested in questioning scientific rationality, justification, and knowledge, among other topics that stem from the analysis of realism and naturalism. What is important to keep in mind, however,

is that there is and there will always be a naturalist crouching behind the modest unificationist. And this is so insofar as the latter adopts ontological, methodological, and epistemological commitments with reality and scientific theories as nurtured by the naturalist.<sup>16</sup>

At the beginning of this section I mentioned that I needed to make precise the notion of ‘unification.’ I have discussed this concept to some extent, emphasizing the view that unification is the *regulative ideal* of finding unity in science by means of discovering perspectives from which we can fit a large number of apparently disparate phenomena into a small number of schemata.<sup>17</sup> I also mentioned that I needed to discuss what is so characteristic that makes the modest unificationist explanation attractive for my purposes. Let me now deliver this second promise.

In the context of scientific explanation, the modest unificationist is interested in maximizing explanatory power and predictive success. One particular virtue of the unificationist theory of explanation is that there are no *epistemic sacrifices*;<sup>18</sup> that is, the unificationist neither ‘hides’ information of the process of explaining, nor leaves mechanisms unspecified. These epistemic sacrifices have their grounds in traditional accounts of explanations and can be classified, according to Friedman, in roughly two groups: a first group concerned with the *precision on the nature of the explanation relation*, and a second group more interested in pinpointing explanation as *scientific understanding*. The first group offers a precise proposal about the nature of the explanation relation, but has little to say about how one gains understanding of the world via explanation. The Deductive-Nomological model as elaborated by Hempel can be included into the first group, as his account shows precisely how to explain and what to expect by explaining (i.e., nomic expectancy<sup>19</sup>). His account also dismisses understanding as a psychological feature and, from the perspective of the Logical Empiricist, as philosophically uninteresting. In this vein, Friedman objects to the Deductive-Nomological model because “to have grounds for rationally expecting some phenomenon is not the same as to understand it” (Friedman, 1974, 8). The second group, instead, accounts for the notion of understanding as the central aim of explanation leaving vague the nature of the explanation relation. Philosophers like Stephen Toulmin, Michael Scriven, and William Dray are good examples of this second group.<sup>20</sup>

The merits of the unificationist account are precisely that it neither sacrifices explanatory precision nor scientific understanding, synthesizing in this way the primary features of both groups. Thus understood, the unificationist theory is the most promising for computer simulations. In addition, we must show its unique-

ness. One way of showing this is by demonstrating that there is a unique set of features from which the unificationist stands out, that is, features that only the unificationist can offer to computer simulations. The inverse must also be true, that is, there must be a unique set of characteristics of computer simulations that single out the unificationist as the most prominent theory of scientific explanation.

Admittedly, it is very difficult to find a set of features that, taken individually, only belongs to the unificationist. For instance, derivability is a desirable feature offered by the unificationist that fits quite well into the context of computer simulations as abstract mathematical/logical structures (see Chapter 2). However, the Deductive-Nomological model also uses derivations to account for the explanatory relation. Similarly, the unificationist accounts for mathematical as well as scientific explanation of physical phenomena. As I argued in Section 4.3.3, on specific occasions it is desirable to be able to account for physical phenomena in purely mathematical terms. But again, this is not a unique feature offered by the unificationist but rather applies to all theories of mathematical explanation of physical phenomena. One could, of course, indicate that neither theories of explanation successfully address the issues put forward by computer simulations, as argued in Chapter 4. Is this a sign that the unificationist is as good a theory as any? I believe that it is a set of features *taken together*, rather than individually, that makes the unificationist the most suitable theory of explanation for computer simulations. I will discuss these features in more detail in the following sections.

Easier than making unificationism a suitable theory is to show that computer simulations adapt quite well to the unificationist framework. The reason for this is simply that computer simulations, as elaborated in this work, are systems that essentially unify the multiplicity of phenomena that they generate.<sup>21</sup> Indeed, in Section 3.3 I claimed that, by means of manipulating the values of the parameters built into the simulation model, the scientist is able to instantiate a multiplicity of different particular simulations. Since, by assumption, each particular simulation represents empirical phenomena, it follows that computer simulations unify a multiplicity of simulated phenomena. For instance, the spikes in Figure 3.3 could only be formed by the values set to the parameters as described in that example. Suppose now that a scientist triples the mass of the planet and reduces the mass of the satellite by half; in such a case the satellite would crash into the planet leaving behind an entirely different pattern of spikes.

If this is correct, then adopting the unificationist credo for computer simulations commits us to accepting an explanatory theory that takes unification as the core of the theory. Explanatory unification, as elaborated by Kitcher, is one such theory

since it takes as the explanatory relationship the derivation of a multiplicity of phenomena from a few set of schemata, and where the epistemic gain depends on reducing such a multiplicity of phenomena to a greater theoretical framework. In other words, if computer simulations are indeed unificationist systems, then the unificationist theory of explanation is our best candidate.<sup>22</sup>

The challenge for the unificationist is now to elaborate on the nature of the explanation relation in a comprehensible way, and to provide a notion of understanding that corresponds to such an explanation relation. In addition, my challenge is to modify this account of explanation in order to make it comprehensible for computer simulations.

Since Friedman's explanatory account suffers from technical problems,<sup>23</sup> especially related to Kim's metaphysical question, I will focus on Kitcher's theory instead. Explanation for Kitcher is, then, an essential reconstruction of the Hempelian approach of arguments and derivations. Naturally, Kitcher amends the deficiencies and inadequacies of the Hempelian model. In this context, the distinctive mark for the unificationist account is that, in order to explain something, one must derive descriptions of many phenomena using the same argument patterns over and over. What are those patterns? They are descriptions that single out natural kinds, objective causal relationships, objective natural necessities, and similar concepts that philosophers use for accounting for the metaphysics of science. As Kitcher puts it, "they are schemata that correctly capture objective dependencies [of phenomena in the empirical world]" (Kitcher, 1993, 150). The more phenomena explained with the same pattern, the more unified and transparent the world appears to us. The unificationist must then find the set of derivations that makes the best tradeoff between minimizing the number of patterns of derivation employed and maximizing the conclusions generated.<sup>24</sup> Let me now show how this is done by first addressing Kim's metaphysical question.

### 5.2.1 The metaphysical question

Explanation, for the unificationist, begins with the set of accepted beliefs  $K$ .  $K$  is identified with the current state of the scientific knowledge of a discipline at a given moment in time. The set of beliefs  $K$ , then, can be identified with classical Newtonian mechanics, Mendelian genetics, the kinetic theory of gases, and so on.<sup>25</sup> Given the set of beliefs  $K$ , the first problem for the unificationist consists in identifying the set of derivations that best systematizes  $K$ , called the *explanatory store* over  $K$ ,  $E(K)$ .<sup>26</sup>

The explanatory store  $E(K)$  is responsible for carrying out the derivation of a

description of a phenomenon. The systematization criteria for entrenching  $E(K)$  is, in Kitcher's own words, "a set of derivations that makes the best tradeoff between minimizing the number of patterns of derivation employed and maximizing the number of conclusions generated" (Kitcher, 1989, 432).  $E(K)$ , then, explains following the principle of unifying a multiplicity of phenomena using the smallest number of argument patterns. In plain words,  $E(K)$  is the minimal set of schemata that best captures the natural kinds and objective dependencies used for the scientific description of a phenomenon. Explanation, then, consists in exhibiting how such a phenomenon shares certain patterns with a greater theoretical framework such as a model or a theory. Although the terminology might be a bit unfamiliar at this point, these definitions will become clearer as I advance in the analysis of unificatory explanation.

Let me begin by specifying what constitutes the explanatory store  $E(K)$ . Kitcher considers four elements, namely:

1. *Schematic sentence*: "an expression obtained by replacing some, but not necessarily all, the nonlogical expressions occurring in a sentence by dummy letters" (Kitcher, 1989, 432). To use Kitcher's own example, the sentence "Organisms homozygous for the sickling allele develop sickle cell anemia" (Kitcher, 1989, 432) is associated with a number of schematic sentences, including "Organisms homozygous for  $A$  develop  $P$ " (Kitcher, 1989, 432) and "For all  $x$ , if  $x$  is  $O$  and  $A$  then  $x$  is  $P$ " (Kitcher, 1989, 432).

(a) *Schematic argument*: "is a sequence of schematic sentences" (Kitcher, 1989, 432).

2. *Filling instructions*: (for a schematic sentence) "a set of directions for replacing the dummy letters of the schematic sentence, such that, for each dummy letter, there is a direction that tells us how it should be replaced. For the schematic sentence 'Organisms homozygous for  $A$  develop  $P$ ,' the filling instructions might specify that  $A$  be replaced by the name of an allele and  $P$  by the name of a phenotypic trait" (Kitcher, 1989, 432). Let it be noted that the filling instructions are semantic in nature, rather than syntactic. The terms that replace a particular dummy letter need not have precisely the same logical form, but they must belong to a similar semantic category. For instance, a filling instruction for a pattern based on Newtonian mechanics must specify that the term to replace a dummy letter refer to a body or to a position within a Cartesian coordinate system. The filling instructions, then, have the

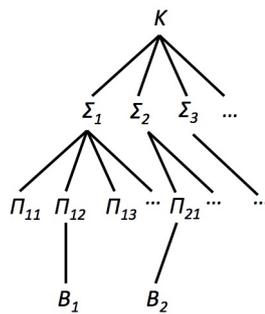
important function of facilitating the semantic connection between schematic sentences and the world;

3. *Classification*: (for a schematic argument) “a set of statements describing the inferential characteristics of the schematic argument: it tells us which terms of the sequence are to be regarded as premises, which are inferred from which, what rules of inference are used, and so forth” (Kitcher, 1989, 432)
4. The *Comments* section plays the fundamental role of being a ‘repository’ for the information that could not be rendered as a schematic sentence but is still relevant for a successful explanation. Strictly speaking, the comments section is not part of the *general argument pattern*, understood as “a triple consisting of a schematic argument, a set of sets of filling instructions, one for each term of the schematic argument, and a classification for the schematic argument” (Kitcher, 1989, 432). However, it does play a relevant role in the overall explanation of phenomena. For instance, Kitcher presents two examples on how to explain hereditary phenotypic traits through pedigree using Mendelian Genetics: ‘Mendel’ and ‘Refined Mendel.’ The first explanatory pattern is limited to one locus and two allele cases with complete dominance. The second explanatory pattern fixes this limitation by producing a more complete schema that covers a broader class of cases. ‘Refined Mendel,’ then, overcomes the limitations of the first schemata, explaining more phenomena with fewer patterns. Kitcher later asserts that ‘Refined Mendel’ “does not take account of linkage and recombination. The next step is to build these in” (Kitcher, 1989, 440) indicating, in this way, that the comments section includes the explanatory limitations of the general argument pattern, future changes to be implemented, further considerations, and the like.

The comments section brings up an important question that can be addressed now. That is, what are the criteria for entrenching the *best* systematization of  $K$ ? In other words, what are the criteria that lodges ‘Refined Mendel’ as more explanatory than ‘Mendel’? Kitcher’s solution is elegant, although a bit cumbersome, and consists in the evaluation of the unifying power of each pattern that has been employed in the systematization of  $K$ .

The solution can be outlined in the following way: find all the acceptable arguments that are relative to  $K$  (e.g., all the theories, principles, laws, etc., of a scientific theory) and call it the ‘ $\Sigma_i$  = the set of arguments over  $K$ .’ Then, from each  $\Sigma_i$ , find the argument patterns that are *complete* with respect to  $K$ ,<sup>27</sup> and call it ‘ $\Pi_{i,j}$  = the

generating set.’ The next step is to select the set with the greatest unifying power and call it the ‘ $B_i =$  the basis of  $K$ .’ The basis with the greatest unifying power is the explanatory store  $E(K)$  that *best* systematizes our corpus of beliefs  $K$  (see Figure 5.1). This is the reason why ‘Refined Mendel’ is a more suitable explanatory store  $E(K)$  than ‘Mendel’: the former has greater unifying power than the latter (i.e., it explains more phenomena with fewer patterns).<sup>28</sup>



Systematizations, sets of arguments acceptable relative to  $K$ .

Complete generating sets.  $\Pi_{i,j}$  is a generating set for  $\Sigma_i$  which is complete with respect to  $K$ .

Figure 5.1: If  $B_k$  is the basis with the greatest unifying power then  $E(K) = \Sigma_k$  (Kitcher, 1981, 520)

Bases.  $B_i$  is the basis for  $\Sigma_i$ , and is selected as the best of the  $\Pi_{i,j}$  on the basis of unifying power. (Kitcher, 1981, 520)

Admittedly, the process of finding the best  $E(K)$  is a bit cumbersome. For this reason I will assume that the explanatory store  $E(K)$  used for explaining phenomena is, indeed, the best possible systematization of  $K$ . With this assumption in place, there is no need to search for the basis with the greatest unifying power. In fact, Kitcher himself makes the same assumptions, all of which I take as unproblematic.<sup>29</sup> Choosing the best systematization of  $K$  is related to the changes on the corpus of scientific beliefs, and on whether such changes affect the unificationist theory of explanation.<sup>30</sup> The general outline for the explanatory process, however, is the same for a fixed set of beliefs as it is for a changing one. Therefore, and for the sake of simplicity, I will focus my efforts on a fixed body of knowledge  $K$ .

To explain, then, consists in deriving descriptions of different phenomena using as few and as stringent argument patterns as possible. The fewer the patterns used, the more stringent they become. The greater the range of different conclusions derived, the more unified the phenomena are. And the more we use the same argument patterns over and over again, the simpler and more transparent the world appears to us. Kitcher puts this idea in the following way: “ $E(K)$  is a set of derivations that

make the best tradeoff between minimizing the number of patterns of derivation employed and maximizing the number of conclusions generated” (Kitcher, 1989, 432)

Let me illustrate these claims with the example of explaining the expected distribution of progeny phenotypes in a cross between two individuals using Mendelian genetics. Kitcher presents the example in the following way:

Classical genetics is centrally focused on (though by no means confined to) a family of problems about the transmission of traits. I shall call them *pedigree problems*, for they are problems of identifying the expected distributions of traits in cases where there are several generations of organisms related by specified connections of descent. The questions that arise can take any of a number of forms: What is the expected distribution of phenotypes in a particular generation? Why should we expect to get that distribution? What is the probability that a particular phenotype will result from a particular mating?, and so forth. Classical genetics answers such questions by making hypotheses about the relevant genes, their phenotypic effects, and their distribution among the individuals in the pedigree. [...]

- (1) There are two alleles  $A$ ,  $a$ .  $A$  is dominant,  $a$  recessive.
- (2)  $AA$  (and  $Aa$ ) individuals have trait  $P$ ,  $aa$  individuals have trait  $P'$
- (3) The genotypes of the individuals in the pedigree are as follows:  $i_1$  is  $G_1$ ,  $i_2$  is  $G_2$ , ...,  $i_N$  is  $G_N$ . {(3) is accompanied by a demonstration that (2) and (3) are consistent with the phenotypic ascriptions in the pedigree.}
- (4) For any individual  $x$  and any alleles  $yz$  if  $x$  has  $yz$  then the probability that  $x$  will transmit  $y$  to any one of its springs is  $\frac{1}{2}$ .
- (5) The expected distribution of progeny genotypes in a cross between  $i_j$  and  $i_k$  is  $D$ ; the expected distribution of progeny genotypes in a cross ... {continued for all pairs for which crosses occur}
- (6) The expected distribution of progeny phenotypes in a cross  $i_j$  and  $i_k$  is  $E$ ; the expected distribution of progeny phenotypes in a cross ... {continued for all pairs in which crosses occur}

*Filling instructions:*  $A$ ,  $a$  are to be replaced with names of alleles,  $P$ ,  $P'$  are to be replaced with names of phenotypic traits,  $i_1, i_2, \dots, i_N$  are to be replaced with names of individuals in the pedigree,  $G_1, G_2, \dots, G_N$  are to be replaced with names of allelic combinations (e.g.,  $AA$ ,  $Aa$ , or  $aa$ ),  $D$  is replaced with an explicit characterization of a function that assigns relative frequencies to genotypes (allelic combinations), and  $E$  is to be replaced with an explicit characterization of a function that assigns relative frequencies to phenotypes.

*Classification:* (1), (2), and (3) are premises; the demonstration appended to (3) proceeds by showing that, for each individual  $i$  in the pedigree, the phenotype assigned to  $i$  by the conjunction of (2) and (3) is that assigned in the pedigree; (4) is a premise; (5) is obtained from (3) and (4) using the principles of probability; (6) is derived from (5) and (2).

4. *Comments:* Mendel is limited to one locus, two allele cases with complete dominance. We can express this limitation by pointing out that the pattern above does not have a correct instantiation for examples which do not conform to these conditions. By refining Mendel, we produce a more complete schema, one that has correct instantiations in a broader class of cases. (Kitcher, 1989, 439)

This explanatory pattern shows how an explanation-seeking question about the transmission of traits through pedigrees is addressed within the genetic theory. Specifically, we can answer questions about why we should expect to find a particular distribution in a specified generation. Explanation then consists in exhibiting (via derivation) how patterns (such as the distribution of progeny phenotypes) can be subsumed under a unified framework such as Mendelian genetics.<sup>31</sup> In more general terms, the process of explaining a phenomenon consists in exhibiting how natural kinds, objective causal relationships, natural necessities, and the like of that phenomenon can be derived and therefore unified into a larger theoretical framework.

Let us note that the derivation must also fulfill some conditions in order to be explanatory. Kitcher indicates that a *particular derivation* instantiates an explanatory store (or general argument pattern) in case that:

1. The derivation has the same number of terms as the schematic argument of the general argument pattern;
2. Each sentence or formula in the derivation can be obtained from the corresponding schematic sentence in accordance with the filling instructions for that schematic sentence;
3. The terms of the derivation have the properties assigned by the classification to corresponding members of the schematic argument. (Kitcher, 1989, 433)

These conditions are set as the epistemic guarantee that the derivation belongs to the explanatory store and, as such, is explanatory. Kitcher's basic strategy is to show that the derivations we regard as successful explanations are instances of patterns that, taken together, 'score' better than the patterns instantiated by the derivations we regard as defective explanations. Showing that a particular derivation is a good or successful explanation is, then, a matter of showing that it belongs to the explanatory store  $E(K)$ . Similarly, explanation in computer simulations will also consist in showing how a particular simulated phenomenon belongs to the explanatory store  $E(K)$ . However, and before entering into the discussion on explanation in computer simulations, let me finish the analysis of the unificationist account by addressing Kim's epistemic question.

## 5.2.2 The epistemic question

In Section 4.2.2 I briefly addressed the importance of the notion of understanding for the scientific enterprise, and in what sense it was related to scientific explanation. In this section I am answering Kim's epistemic question in the context of the unificationist theory of explanation. The chief question here is how does an explanation render the explanandum as 'understood'? Or, equivalent to our study, what sort of understanding should we expect by explaining the explanandum? Let me begin by recapitulating Kitcher's notion of understanding:<sup>32</sup>

Understanding the phenomena is not simply a matter of reducing the 'fundamental incomprehensibilities' but of seeing connections, common patterns, in what initially appeared to be different situations [...] *Science advances our understanding of nature by showing us how to derive descriptions of many phenomena, using the same patterns of derivation again and again, and, in demonstrating this, it teaches us to reduce the number of types of facts we have to accept as ultimate (or brute)* (Kitcher, 1989, 432. Emphasis original)

Let us note that Kitcher's first task is to distance himself from accounts of explanation that take *understanding* as a mere reduction of the unknown to the known, the unfamiliar to the familiar, the complex to the simple.<sup>33</sup> From a philosophical point of view, this is a naïve reduction that begs the question about the notion of understanding. Do we understand more about the planetary orbits because we describe them as circular rather than elliptical? After all, geometrically speaking, a circle is simpler than an ellipse. The problem is that the notion of 'known,' 'familiar,' or 'simpler' is essentially idiosyncratic and conjunctural in the sense that what is 'familiar' to some may not be so to others: what might be familiar to the scientist might not be so to the layman. This point nicely illustrates the fact that being familiar is not the same thing as being understood. As Friedman points out, we are all familiar with the behavior of radios and televisions, but not everyone understands why they behave the way they do.<sup>34</sup> The opposite is also true: there are many cases where the familiar has been reduced to the unfamiliar. Perhaps the most cited example in the literature is the explanation of the reflection of light by the behavior of electromagnetic waves.<sup>35</sup>

In more general terms, Friedman argues (and Kitcher agrees) that any of the 'familiarity views' (i.e., theories that show explained phenomena to be 'natural,' 'necessary,' or 'inevitable'<sup>36</sup>) fail because they assume that the explanans possess some special epistemological status or property (e.g., being known, familiar, simpler, or antecedently understood) that is somehow 'transferred' to the explanandum.<sup>37</sup> To these views, then, one understands the explanandum because it has been reduced to the explanans, which has been understood in advance.

Friedman also reports a further point of interest: the familiarity views are *local* in the sense that understanding stems from the relation between the explanans and the explanandum. In this vein, it is natural that one ends up trying to find some special epistemological status between these two. Instead, Friedman defends a *global* view of scientific understanding, one that constitutes a cognitive relation between the explanandum and the total corpus of our beliefs.<sup>38</sup> Thus understood, the unificationist theory of explanation facilitates scientific understanding *without* assuming familiarity, but rather demanding *unification* with the total corpus of beliefs. In this sense, scientific understanding becomes a global affair, where an explained phenomenon is unified with the total set of accepted scientific beliefs, and thereby somehow accommodated into the total number of accepted phenomena.

By the same token, Eric Barnes argues that Friedman's account of understanding has two important virtues; first, by "accommodating the insight that understanding is not generated by any sort of intelligibility transfer and, second, [by] providing an answer to the question of how it helps our understanding to derive one puzzling phenomenon from another" (Barnes, 1992, 5). I have argued for the first virtue above; let me now use the second virtue for showing how understanding is achieved according to the unificationist.

Let us return to Kitcher's quotation above. According to him, the aim of scientific explanation is to increase our understanding of the world by decreasing the number of phenomena that we must take as independent or brute. Such a decrease is accomplished by means of unifying an explained phenomenon with our total corpus of scientific beliefs; that is, by 'seeing connections' and 'common patterns' that initially escaped our comprehension and are now becoming intelligible to us by cohering with our set of beliefs. The criterion for unification, as Kitcher points out at the end of his quotation, is based on the idea that  $E(K)$  is a set of derivations that makes the best tradeoff between minimizing the number of patterns of derivation employed and maximizing the number of conclusions generated.<sup>39</sup> Hence, by using  $E(K)$  for explaining a brute phenomenon one exhibits connections and common patterns that cohere with the totality of our corpus of beliefs and can thereby be unified. Unification, then, yields understanding because it decreases the number of phenomena that we must accept as brute or independent by unifying them with our total corpus of scientific beliefs.<sup>40</sup> In the words of Friedman, "[scientific explanation] increases our understanding of the world by reducing the total number of independent phenomena that we have to accept as ultimate or given. A world with fewer independent phenomena is, other things equal, more comprehensible than with more" (Friedman, 1974, 15). Let us note that explanation, unification, and under-

standing are three interlaced concepts that cannot be individually studied, but must be jointly appreciated. Indeed, by means of explaining, one gains understanding of the world, which comes hand in hand with unification since explanation consists in unifying a multiplicity of phenomena with a corpus of beliefs.

Friedman makes use of a series of examples that allow him to spell out the process of understanding a brute fact via explaining it. Let me briefly reconstruct one of his examples:

Consider a typical scientific theory (e.g., the kinetic theory of gases) [...] This theory explains phenomena involving the behavior of gases [...] [adding] a significant *unification* in what we have to accept. Where we once had three independent brute facts (that gases approximately obey the Boyle-Charles law, that they obey Graham's law, and that they have the specific-heat capacities they do have) we now have only one (that molecules obey the laws of mechanics) [...] Once again, we have reduced a multiplicity of unexplained, independent phenomena to one (Friedman, 1974, 14-15. Emphasis original)

Kitcher endorses these same ideas: "Friedman argues that a theory of explanation should show how explanation yields understanding, and he suggests that we achieve understanding of the world by reducing the number of facts we have to take as brute. [...] Something like this is, I think, correct" (Kitcher, 1989, 431). In this vein, by instantiating schemata like Mendel's, we explain why certain distribution of phenotypes have specified probabilities and thereby are able to unify such a phenomenon of transmission of traits through pedigrees with Mendelian theory. It follows that the unified phenomenon is no longer 'brute,' for now it has been subsumed under a larger corpus of beliefs about genetic theory, unifying with it, and therefore increasing the general scientific understanding and coherence of biology.<sup>41</sup>

Strictly speaking, these ideas are not exclusively unificationist since their roots are in the 'unofficial' view of the covering law model.<sup>42</sup> Herbert Feigl, for instance, wrote, "The aim of scientific explanation throughout the ages has been unification, i.e., the comprehending of a maximum of facts and regularities in terms of a minimum of theoretical concepts and assumptions" (Feigl, 1970, 12); and Hempel made a similar point when he said:

What scientific explanation, especially theoretical explanation, aims at is not [an] intuitive and highly subjective kind of understanding, but an objective kind of insight that is achieved by a systematic unification, by exhibiting the phenomena as manifestations of common, underlying structures and processes that conform to specific, testable, basic principles (Hempel, 1966, 83. See also (Hempel, 1965, 444)).

The question that still remains unanswered, however, is what do Kitcher and Friedman mean by 'brute fact' and 'independent phenomena,' respectively?<sup>43</sup> The anal-

ysis of the notion of understanding is completed with the study of these concepts. The problem is that neither Kitcher nor Friedman have much to say about them. As a result, I offer here a brief analysis of current discussions in the philosophical literature.

Two particularly interesting (though conflicting) positions stand out among the rest: Ludwig Fahrbach who takes brute facts as facts for which no direct explanation exists; and Barnes, who takes them as facts for which the explanation is missing. Let me briefly elucidate these positions. To Fahrbach's mind, brute facts are 'starting points' in the order of the empirical world; that is, facts that have no predecessor, for instance the Big Bang constitutes one.<sup>44</sup> The importance of a brute fact, then, is that it is closely related to a 'single fact' (e.g., the background temperature of the universe), which is a byproduct of a brute fact (e.g., the Big Bang). A brute fact, therefore, is unexplainable in itself, but it contributes to exhibiting the place of a single fact in the order of the empirical world.<sup>45</sup>

The other interpretation belongs to Barnes who has a more complex (and more compelling) analysis. To Barnes' mind there are two kinds of brute facts: *epistemically brute facts* (i.e., those whose explanation, which we assume exists, remains unknown), and *ontologically brute facts* (i.e., facts with no explanatory basis beyond themselves, such as the ultimate laws of physics or the origins of the universe)<sup>46</sup> There are some affinities between Barnes' ontological interpretation of brute fact and Fahrbach's. However, the difference between these two scholars lies in Barnes' epistemic interpretation which takes brute facts as representing a temporal 'gap' in our understanding of the world, a 'gap' that does not need to remain an enigma but which is overcome by means of explanation. According to Barnes, then, it is through the process of explaining brute facts that we obtain understanding of the ultimate mysteries of the universe.

I believe that Barnes' positions, rather than Fahrbach's, is the correct way to interpret Kitcher and Friedman's notion of brute or independent phenomena. I have two reasons for holding this conviction. First, Barnes' naturalistic position is in better accordance with Kitcher's own naturalism. To the latter, the world does not comprise mysterious or magical entities that cannot, sooner or later, be understood within the scientific method.<sup>47</sup> Second, both Kitcher and Friedman seem to take a brute fact as some sort of gap in our understanding that can be and must be overcome through explanation. Indeed, there is no single reference that I have been able to find of brute facts as unexplainable in themselves. For these two reasons, I take a brute fact as an explanandum that we do not quite understand but that we have, or we will have, the means to explain.

Thus understood, let me reconstruct the unificationist answer to Kim’s question in the following way: the world becomes a more transparent and understandable place when we reduce the number of facts that we must take as brute or independent. Such ‘reduction’ is taken in the sense of *unification*. It is achieved by the process of incorporating the explained phenomena into our system of beliefs and of seeing how they form part of a unified picture of the empirical world.

Allow me now to discuss how the unificationist account provides a conceptual framework for explaining by a computer simulation.

### 5.3 Explanatory unification for a computer simulation

In previous sections I have claimed that the unificationist account is the most suitable theory of scientific explanation for computer simulations. Let me substantiate this claim now that I have put together all the pieces. But first, one warning and one reminder. The warning: here I will be analyzing explanation for *one* computer simulation that produces a *multiplicity* of simulated phenomena. I will say some more about explanation for *multiple* computer simulations that compete with each other in the production of similar simulated phenomena in Section 6.2.1. As for the reminder, in Section 3.3.1.1 I introduced two new concepts: the *general computer simulation* and the *particular computer simulation*. These terms were meant to add precision to the analysis of computer simulations. The former stands for the computer simulation *before* the initial and boundary conditions have been instantiated, whereas the latter stands for the computer simulation when these values have been instantiated and the simulated phenomenon produced. Let me note that I have intentionally avoided the use of these two terms until now. The reason for this is that the discussions and arguments addressed so far did not require such precision in the language used. Indeed, in previous chapters I made use of terms such as ‘computer simulation,’ ‘simulation model,’ ‘results of the simulation,’ and similar.<sup>48</sup> Although such a use is neither incorrect nor misleading (in fact, I will keep using them as long as the context is clear), the analysis of explanation in computer simulations requires the use of more precise terminology. Having said this, let me now recapitulate my working conceptualization of computer simulation from Section 3.3.1.1:

A computer simulation consists of two elements: a *general computer simulation*, which is the implementation of a simulation model on a digital computer ready to be executed. The simulation model represents the general patterns of behavior of a target system, which can be ‘empirical,’

‘theoretical,’ or ‘data systems.’ A general computer simulation produces results when its initial and boundary conditions are filled out, creating in this way a *particular computer simulation*. A particular computer simulation produces results by reckoning using stepwise ‘analytic methods,’ ‘numerical analysis,’ or ‘stochastic techniques.’

Thus defined, a *general computer simulation* describes general patterns of behavior of the target system,<sup>49</sup> whereas a *particular computer simulation* is its instantiation with the initial and boundary conditions fulfilled. Since I have set the conditions for *knowing* when the simulated phenomena genuinely represent empirical phenomena (Section 4.2.1), it follows that a particular computer simulation represents the empirical phenomenon that it simulates. Moreover, it also follows that a general computer simulation is a description of the patterns of behavior shared by a multiplicity of empirical phenomena (specifically, of those phenomena that are within the space of solutions of the general computer simulation).

Let me illustrate these points with the example of the satellite under tidal stress. The general computer simulation is the model of all two-body systems as described in the specification in Section 3.3.1. The particular computer simulation, on the other hand, is as described by the initial and boundary conditions set forth in Section 2.2. Since the general computer simulation describes all possible two-body systems, by changing the initial and boundary conditions we obtain different simulated phenomena, such as the simulation of two satellites, two planets, and so on. Let us note that by changing the initial and boundary conditions, then, it is possible to represent known phenomena, as well as new, independent phenomena.

In the following two sections I revisit Kim’s metaphysical and epistemic questions in the context of computer simulations as explanatory devices. The idea is to show how a general computer simulation explains the simulated phenomenon and in what sense such an explanation yields understanding. Briefly, the metaphysical question can be answered by showing how the principles of the unificationist account can be applied in the context of a computer simulation: we explain by producing an argument whose conclusion describes the simulated phenomenon. The epistemic gain, on the other hand, lies in seeing connections and common patterns of simulated phenomena that initially appeared to be unrelated.

### 5.3.1 The metaphysical question (revisited)

Let me begin this section by recalling again the example of the orbiting satellite under tidal stress. The challenge here is to explain why the simulation produces the spikes shown in Figure 3.3. Although the explanation of these spikes can be

easily explained by any student in physics, it may not have been obvious *ab initio* that such spikes would occur. In fact, a good simulation would also bring out new, unexpected, and important features of the system under investigation.

The aim, then, is to explain why the satellite produces the behavior visualized on the monitor. Woolfson and Pert offer the following explanation:

The reason for the spikes can be understood by reference to Fig. [3.4], where the satellite is shown approaching the near point. It is stretched by the tidal force, but due to inertia the satellite tidal bulge lags behind the radius vector. The departure of the satellite from a spherical form can be thought of as a sphere plus two bobs of matter at the points  $P$  and  $Q$  and the gravitational forces on these are shown. Since the force at  $P$  will be greater than that at  $Q$  there will be a net torque imparting angular momentum to the satellite, causing it to spin faster. [...] When receding from the near point the tidal bulge is ahead of the radius vector and the effect is reversed (Woolfson and Pert, 1999, 21)

The first thing to note is that Woolfson and Pert's explanation relies on the set of equations implemented plus background knowledge of classical mechanics. Taken as stated, this explanation relies on the mathematical machinery of the scientific model and on the scientists' theoretical knowledge of the target system. In this vein, it ignores the fact that it was the simulation that produced the simulated phenomenon and, as such, it neglects the simulation as part of the explanation. As I will show later, a computer simulation, although highly reliable, introduces distortions and alterations in the final results that must be taken into account for the explanatory process. In addition, Woolfson and Pert's explanation is incapable of accounting for highly sophisticated phenomena produced by a computer simulation. As I argued in Section 4.3.3, explanation in terms of pure mathematical vocabulary is, in some cases, an unrealistic choice.

For these reasons, I reject their explanation on the basis that the computer simulation, rather than mathematical machinery, must be the principal component for carrying out the explanation. Again, the reason for this is that the phenomenon simulated also depends on truncation errors, round off errors, discretization methods, and so forth, introduced by the process of computing and neglected by an explanation based on mathematical machinery and background knowledge. Let me now elaborate on how we can explain using a computer simulation.

As I discussed at the beginning of Section 5.2, explanation for the unificationist begins with a set of accepted beliefs,  $K$ . In the sciences,  $K$  can be classical mechanics in physics, evolutionary theory in biology, or the atomic theory in chemistry, just to mention a few examples. Finding the set  $K$  for a computer simulation is no different than in the sciences due to the way I have constructed a computer simulation: it also

relies on current scientific knowledge. Woolfson and Pert's example, for instance, relies on a set of differential equations as described by classical Newtonian mechanics. The set of accepted beliefs  $K$  for the simulation of a satellite under tidal stress is, therefore, classical Newtonian mechanics.

The real challenge is to specify what counts as the *explanatory store over  $K$* ,  $E(K)$ , as the set of acceptable arguments for explanation. Let us note that in order to specify the explanatory store  $E(K)$ , we require cognitive access to the simulation. In other words, if an explanation consists in deriving a phenomenon from the explanatory store, then the question is from where in the computer simulation do we obtain such an explanatory store? This is a delicate point that I will try to illuminate with an analogy to scientific practice.

Consider a group of scientists that are trying to explain a given phenomenon. If these scientists have a theory as a possible candidate for the explanation, then they can elaborate the explanatory store from that theory. Such a methodological step is possible because the scientists have *cognitive access* to the theory in a straightforward sense: they understand its limits, the domain of applicability, and so forth. In the literature on computer simulations, however, there is a strong case for computer simulations as *cognitively opaque*,<sup>50</sup> that is, the scientist conceives the simulation as a 'black box' that gets input data and obtains after some reckoning the simulated phenomenon. I oppose this view by arguing that the simulation model or, more precisely, the *general computer simulation* is the unit of analysis of computer software capable of yielding insight into the simulation system (see Section 2.2). The situation I used as an example for my claims was of the scientist in need of making changes to the simulation system. Without cognitive access to the heart of the simulation, it would not be possible to do any reprogramming of any kind. In plain words, simulations are not 'opaque' systems since we can access the general computer simulation, we can modify it, reprogram it, or simply inspect it to know what it is all about.

It comes naturally, then, to take  $E(K)$  as based on the general computer simulation. Let us note that this result is perfectly compatible with the assumptions of the unificationist, for  $E(K)$  is the set of argument patterns that maximally unify  $K$ . Since  $K$  is, in this case, our knowledge about classical mechanics,  $E(K)$  must then include the set of equations, round-off errors, discretization methods, and the like, specified in the general computer simulation for the production of the spikes.

As I claimed earlier, for an explanation to be successful we need to include all the pertinent information for the interpretation of the simulated phenomenon. In this vein, and since round-off errors, truncation errors, discretization methods, and

the like, play an important role in shaping the simulated phenomenon, they must be considered as constituents of the process of explanation. In a moment I will elucidate where this information must be included and how the researcher can handle it for a successful explanation. Let me first spell out how to construct the explanatory store  $E(K)$  based on the general computer simulation. For the spikes of Figure 3.3, I reconstruct  $E(K)$  in the following way:

*Schematic Sentences:*

- 1) There are two objects, one with a mass of  $M$  and another with a mass of  $m$  ( $\ll M$ )
- 2) There is an orbit of semi-major axis  $a$  and eccentricity  $e$
- 3) The object of mass  $m$  is distributed into three masses, each  $m/3$ , at positions  $S_1$ ,  $S_2$  and  $S_3$ , forming an equilateral triangle free of stress.
- 4) The masses are connected by springs, each of unstressed length  $l$  and the same spring constant,  $k$ .
- 5) A spring constantly stretched to a length  $l'$  will exert an inward force.
- 6) The expected force exerted for two objects is:  $F = k(l' - l)$
- 7) The expected total energy for the two objects is:  $E = -\frac{GMm}{2a}$
- 8) The general equation for the angular momentum is:  $H = \{GMa(1 - e^2)\}m$
- 9) The spikes formed are due to an exchange between spin and orbital angular momentum around the closest approach.

*Filling Instructions:*

The mass  $M$  and the mass  $m$  will be replaced by a planet and a satellite, respectively. A dissipative element is introduced into the structure by making the force dependent on the rate of expansion or contraction of the spring, giving a force law  $F = k(l' - l) - c\frac{dl'}{dt}$ , where the force acts inwards at the two ends. Values for  $e$ ,  $l$ ,  $S_1...S_3$  and  $a$  must be given (recall from 3.3.1 the initial and boundary conditions for this particular computer simulation).

*Classification:*

The classification of the argument indicates that 1-5 are premises, that 6-8 are obtained by substituting identicals, and that 9 follows from 6-8 using algebraic manipulation and techniques of calculus.

*Comments:*

It is the second term in the general force law which gives the simulation of the hysteresis losses in the satellite. Also note that the filling instructions may also have mathematical functions in it.

Let us note that here I have only reconstructed Woolfson and Pert's general computer simulation within the unificationist framework. However, an explanation with computer simulations must also take into consideration round off errors, truncation errors, and the like. Now, sometimes such information can be rendered into sentences; that is, the scientists have complete understanding of the general computer simulation (as good programming practice suggests) to the point that they are capable of predicting the deviation of errors, the computational consequences of implementing one rather than another discretization method, and the like. When that is the case, such detailed information must be added to the *Schematic Sentences* for the future derivation of a complete explanation.

Let me illustrate the importance of knowing these details by considering the following example: say that there is a round-off error in the computation such that, for each operation, it introduces a difference in each revolution of the satellite of approximately  $1^{-10}$  meters with respect to the true value. Although very small, this round-off error plays a crucial role in the overall eccentricity of the satellite and, therefore, in the formation of the spikes. In particular, this error is responsible for the satellite reaching an eccentricity equal to 0 (i.e., after a determined number of runs, the satellite reaches a circular orbit). If this were the case, then the tidal stress would show an entirely different shape, one that reaches a stable point without showing the spikes (i.e., it is irrelevant that the satellite is at its closest approach because the orbit is now circular). This example shows that if the explanatory process ignores this round-off error, there are no grounds for genuinely claiming an explanation of the spikes of Figure 3.3. In addition, the explanation would also fail to yield insight into the empirical system.

Unfortunately, the researcher does not always have access to such detailed information about the general computer simulation. For these situations (which occur more often than not), the scientists must include the information in the *Comments* section. For instance, say that the scientists know about the existence of a certain round-off error, but they can neither measure nor render it into a schematic sentence. One possible way to ensure that this information is used for explaining the spikes could be to add the following sentence: "for a value of  $> 10^{10}$  of total simulation time, the orbit eccentricity tends to be reduced to 0, which explains the absence of spikes in the simulation," or something similar to the *Comments* section.

This information is added to the *Comments* section for future considerations, in the same sense that Kitcher added the comments about ‘Refined Mendel’: it is crucial information for the interpretation of the phenomenon simulated that could not be rendered as a schematic sentence. Without such information there are no epistemic guarantees that the explanation will be successful.

Knowing about the presence of errors makes the epistemic difference between being aware of the existence of a disturbing factor, and thus being able to interpret and to explain the simulated phenomenon in light of those errors, and groping blindly. This is the situation to which Woolfson and Pert expose themselves by not taking into account computational errors as a possible source of disturbances of the simulated phenomenon; explanation, for the authors, remains heavily dependent upon the mathematical machinery, ignoring the physical and logical constraints of computing.

Now, why is our main interest to explain a simulated phenomenon? As I have mentioned before, the motivation for using computer simulations lies in the fact that we trust them as cheap, efficient, and reliable means for understanding something about the world. This means that an explanation-seeking question answered in the domain of the simulation also aims at answering something about the world. For instance, we explain the spikes in Figure 3.3 because we want to explain (and thereby understand) its counterpart in the empirical world (assuming, of course, that the specified satellite is put into orbit around the specified planet). The question, then, is in what sense does an explanation of a simulated phenomenon also apply to the empirical phenomenon?<sup>51</sup> In Section 4.2.1 I argued that a general computer simulation is reliable in the sense that it simulates a phenomenon that genuinely represents what is ‘out there’ in the world. In terms of the unificationist account we can say that the patterns of behavior of the simulated phenomenon are the same as the patterns of behavior of the empirical phenomenon. Naturally there are representational constraints, for a simulated phenomenon only stands for some aspects of interest of the empirical phenomenon. There are also computation constraints, for the reckoning of the simulated phenomenon might introduce imprecisions in the results. However, despite the constraints which have been largely discussed in this dissertation,<sup>52</sup> the simulated phenomenon still stands as a proxy or surrogate for the empirical phenomenon. These conditions were set for enabling the conclusion that if the phenomenon simulated has been successfully explained, then the same explanation applies to the empirical phenomenon. This conclusion stems from the fact that a computer simulation is a reliable process in the relevant sense, that is, what it simulates is also ‘out there,’ ‘in the world.’ It follows that successfully explaining

a simulated phenomenon also explains the empirical phenomenon as well.

Let me now turn to the last of our issues. So far I have made my case about general computer simulations that are genuine representations of empirical target systems. In this context, I argued, one is perfectly justified in claiming that a successful explanation of the former also applies to the latter since the simulated phenomenon genuinely represents the patterns of behavior of the empirical phenomenon. However, computer simulations are interesting in that they also simulate *impossible* phenomena, that is, phenomena that in principle we might not find in reality. I restrict these types of simulated phenomena to one specific class, namely, those produced by a general computer simulation implementing well-known sets of equations but instantiated by initial and boundary conditions that make the particular computer simulation impossible.<sup>53</sup> Let me illustrate this point with an example: suppose that the simulation of the satellite makes explicit use of the gravitational constant, or ‘big  $G$ .’ Say that the scientist wants to change this constant to something that current physics tells us is impossible: ‘big  $G$ ’ now has a value equal to  $0 \text{ m}^3\text{kg}^{-1}\text{s}^{-2}$ . The general computer simulation, then, still represents the target system of all two-body interactions (i.e., the same orbital period equations, angular momentum equations, and so forth), and unless restrictions are imposed on the initial and boundary conditions, the general computer simulation also allows us to set these conditions to any value, in particular  $G = 0 \text{ m}^3\text{kg}^{-1}\text{s}^{-2}$ . Arguably, this is one of the most attractive features of computer simulations, namely, the capacity to ‘bend’ the world by allowing scientists to imagine alternative scenarios to reality. The question is, therefore, what kind of explanation does the unificationist offer for these impossible situations?

From the point of view of the metaphysical question, the explanation of this new, impossible simulation is perfectly possible since all the actors are in place for the derivation of the description of such a simulated phenomenon. The burden of proof, however, lies on whether scientists obtain any insight into reality, for now the world is no longer telling them how it really is! The answer to this question is within the realm of Kim’s epistemic question, which brings me to my next subject, that is, a discussion on the epistemic gain of explaining a simulated phenomenon.

### 5.3.2 The epistemic question (revisited)

Earlier in this chapter I argued for the epistemic value of scientific explanation by showing that its goal is to yield understanding. I also claimed that understanding for the unificationist consists in reducing the number of phenomena that we must take as independent or brute. Finally, I showed how simulated phenomena could be ex-

plained by making use of the conceptual framework that the unificationist provides. There is still one remaining issue, namely, to give an answer to Kim’s epistemic question: what exactly is our epistemic gain when we provide an explanation of a simulated phenomenon?<sup>54</sup> The answer to this question is at the heart of my defense of the epistemic power of computer simulations.

Let me begin by restating Kitcher’s words on understanding: “understanding is [...] seeing connections, [and] common patterns, in what initially appeared to be different situations [...] that is] *facts that we have to accept as ultimate (or brute)*” (Kitcher, 1989, 432. Emphasis original). In Section 5.2.2, I showed and discussed the importance of the notion of ‘brute fact’ or ‘independent phenomena’ for the unificationist. In the context of computer simulations, however, we must be a bit more careful with the interpretation of these terms. Besides following the unificationist’s interpretation as ‘unexplained phenomena,’ there are also two senses in which a simulated phenomenon can be taken as independent:

- (a) A simulated phenomenon is *independent from other simulated phenomena* in the sense that the general computer simulation has been instantiated by different initial and boundary conditions. For instance, the simulated phenomenon ‘a satellite orbiting around a planet’ is independent from the simulated phenomenon ‘a planet orbiting around the sun’;
- (b) A simulated phenomenon is *independent from a model or a theory* in the sense that the former’s existence does not depend on the model or theory. For instance, the simulated phenomenon ‘a satellite orbiting around a planet’ is independent from the theory ‘classical Newtonian mechanics.’

A simulated phenomenon is independent only in sense (a) above. I argued for this point earlier: by fulfilling different initial and boundary conditions of the general computer simulation, the scientist is able to represent a host of different and conceptually independent phenomena. Unification, in this sense, is indeed at the core of the philosophy of computer simulations. Let me illustrate this point with an example: consider any two different instantiations of the initial and boundary conditions for the satellite under tidal stress. Let us call the first instantiation  $P_1$ , which consist of a mass of the first body of  $2 \times 10^{23}$  Kg (i.e., the planet is slightly heavier than the satellite). Let us call the second instantiation  $P_2$ , which consist of a mass of the first body of  $2 \times 10^{27}$  Kg (i.e., the planet has the mass of Jupiter). According to these two particular simulations,  $P_1$  represents a satellite orbiting around a small planet, whereas  $P_2$  stands for a satellite that will crash into Jupiter. These two particular computer simulations are successful in the sense that they both represent

a real situation, and both can be explained following the analysis expounded in the previous section. The claim here, however, is that  $P_1$  and  $P_2$  represent two *distinctive* and *independent* (from each other) phenomena in the sense that the occurrence of  $P_1$  is not determined by the occurrence of  $P_2$ .

This example also reflects an interesting feature about computer simulations: among all possible results, there are going to be some which are known and expected by the researcher, but there are also going to be results that are novel and demand an explanation (i.e., brute or independent facts in the unificationist's sense). The sense in which I take the notion of 'independent' here is precisely that the occurrence of one result has no bearing on the likelihood that the next result will be known. Such an interpretation of 'independent' is analogous to taking an empirical phenomenon as independent of other empirical phenomena despite the fact that they can be explained, predicted, or observed by the same theory.

On the other hand, simulated phenomena are *not* independent in sense (b) because they are the byproduct of reckoning the same general computer simulation. In other words, the totality of simulated phenomena are conceptually linked to each other because they all belong to the same space of solutions of the general computer simulation. This means that if the scientists are carrying out a simulation of a two-body system, there are no reasons for expecting any results other than those related to the two-body system as represented in the general computer simulation. The analogy with the natural sciences is that a simulated phenomenon is 'created,' or 'produced' by the computer whereas an empirical phenomenon is 'out there,' autonomous from our scientific theories.<sup>55</sup> This second sense of independence, then, imposes some restrictions in the epistemological virtues of computer simulations. For instance, in Section 1.3.1 I argued that a simulated phenomena is unlikely to be of any use for *replicability* of results, or for acting as a *crucial experiment*. Let me reiterate that replicability of results and crucial experiments are two techniques for confirmation and refutation of theories. Therefore, these concepts as stated in current literature do not apply to simulated phenomena.

However, when the discussion is about scientific explanation, there are no reasons for being concerned about these limitations. And this is not just blind confidence; rather it can be justified. Consider a skeptic who objects that independence in sense (b) promotes circularity in explanation. That is, if the simulated phenomenon to be explained is produced by the general computer simulation, then it seems that the *explanandum* is somehow contained in the *explanans*. It follows that there is no genuine explanation (and thereby understanding) of the explained phenomenon for we would not be obtaining any genuine insight that was not already there before

the explanation. In order to defuse any concern in that respect, let me note two facts: first, the fact that the simulated phenomenon *is* in the space of solutions of the general computer simulation does not entail that the scientist understands it *a priori*.<sup>56</sup> This is especially true when the simulated phenomenon is a novelty for the scientist. Second, and from a logical point of view, the simulated phenomenon is not a derivation (mathematical or otherwise) of the general computer simulation. Instead, the process of reckoning involves so many stages of calculations that dismantle the idea of pure, straightforward derivations: reckoning is not deriving. Therefore, we cannot genuinely claim that the derivation involved in the explanatory process is comparable with the reckoning of the simulated phenomenon. From these two facts it follows that explaining simulated phenomena is a genuine epistemic step towards the understanding of the world. I conclude that there is a clear payoff in favor of using computer simulations for explaining independent phenomena, despite the fact that they are all produced by the same general computer simulation.

Simplicity here determines that we must be careful with the terminology in order to avoid confusion. I will adopt Kitcher's notion of *brute fact* as a way to refer to a simulated phenomenon in need of explanation. Since Kitcher's notion can easily be matched with Friedman's *independent phenomena*, as argued in Section 5.2.1, I will not discuss that point here any further. I reserve the term 'independent phenomena,' then, for referring to the two senses given above (I will also specify in which sense I am using the term).

Having clarified this, let me return to Kim's epistemic question and to provide an answer as to how we gain understanding of the world by simulating specific parts of it. Following the unificationist, understanding comes from seeing connections and common patterns in what initially appeared to be unrelated simulated phenomena (i.e., brute simulated phenomena). These connections and common patterns are exhibited by the process of deriving a description of such simulated phenomena, that is, by explaining them. Therefore, by means of the epistemic act of seeing connections and common patterns, scientists are able to unify the now explained simulated phenomena with the total corpus of scientific beliefs, reducing in this way the number of simulated phenomena that they must accept as brute. Moreover, as argued in the previous section, to explain a simulated phenomenon is also to explain what it represents in the empirical world. Hence, seeing connections and common patterns of a simulated phenomenon is equivalent to seeing connections and common patterns of the empirical phenomenon. It follows that the unification exerted over the simulated phenomenon is also over the empirical phenomenon. As I claimed in

Section 4.2.1, these are the metaphysical requirements for a computer simulation to explain (and thereby yield understanding) of *the world*.

Finally, as analyzed in Section 5.2.2, reduction is interpreted as the unification of the unexplained simulated phenomenon into our general corpus of beliefs. The criterion for unification is based on the idea that  $E(K)$  is a set of derivations that makes the best tradeoff between minimizing the number of patterns of derivation employed and maximizing the number of conclusions generated.<sup>57</sup>

Let me now illustrate this point with the explanation of the satellite under tidal stress. Scientists are in the position to explain why the spikes on Figure 3.3 occur when simulating  $P_1$ , as well as why  $P_2$  collides with Jupiter. In both cases they gain understanding by seeing connections and common patterns shared by  $P_1$  and  $P_2$  that can be unified into a more general theoretical framework that is classical Newtonian mechanics. In other words,  $P_1$  and  $P_2$  are two independent (in sense (a)), unrelated simulated phenomena that, by means of sharing common patterns of behavior, can be unified and therefore understood in the greater corpus of beliefs that is our current scientific understanding of general gravitation.

We know from previous discussions that explanation for the unificationist is a matter of deriving a multiplicity of phenomena with few patterns *over and over*. Explaining with the same patterns, thus, has a positive impact in our scientific understanding for it becomes systematized and reinforced. Explanation in computer simulation involves yet another virtue because using the same patterns for explaining the multiplicity of simulated phenomena also systematizes and reinforces the computer simulation in itself. Recall from Sections 3.3, 3.3.2, and 3.3.3 the methodological and epistemological concerns about computer simulations as systems prone to errors. Thus, I claim here that by means of explaining simulated phenomena using the same explanatory store works also as a means for reinforcing the reliability of computer simulations. The result is that the epistemic access to the simulated phenomenon becomes more transparent and, with respect to the corpus of beliefs and the simulation, more unified. It follows, then, that by explaining a series of simulated phenomena with the same general computer simulation, the world becomes a more unified, understandable, and transparent place.

A possible objection to this argument originates in the scope of the unifying power of computer simulations. Indeed, one could ask whether a simulated phenomenon actually unifies with the totality of our corpus of scientific beliefs, or only with that portion of it embedded into the general computer simulation. This is a fair objection stemming from the fact that a simulated phenomenon is not independent in sense (b). However, I believe this objection to be wrong. The epistemic act of

unifying a simulated phenomenon is not limited to the knowledge embedded in the general computer simulation, but also reaches the totality of our corpus of belief. For instance, by explaining the spikes of a satellite under tidal stress, one is not only unifying classical Newtonian mechanics but also making sense of other areas of the sciences as well, such as thermodynamics and kinematics. This is especially true about models applicable to multiple scientific domains. For instance, an explanation of a simulation of a Lotka-Volterra predator-prey model yields understanding that is valid for biology, economics, and the social sciences.<sup>58</sup>

I would now like to finish this section by briefly addressing the case of simulating *impossible* phenomena as presented in the previous section. Admittedly, this issue raises several metaphysical riddles that I will not be addressing here. In any case, the issue stems from a general computer simulation that, genuinely representing empirical target systems, gets its initial and boundary conditions filled in such a way that it produces impossible simulated phenomena. Let me first note that this case is *not* equivalent to a heuristic computer simulation, where the scientist is evaluating a theoretical conjecture by means of exploring the space of results of a simulation model. In cases like this, it is not required that the general computer simulation has representational content. Instead, the case of impossible simulated phenomena is a consequence of the plasticity of having general computer simulations whose initial and boundary conditions are, to a large extent, open to be instantiated with any value. As I mentioned before, this raises metaphysical questions regarding the existence of such a phenomenon in reality, especially after demanding that the simulation model have representational content (recall my arguments from Section 4.2.1). I have nothing illuminating to say here; this is a time-honored problem in the metaphysics of scientific models and, one could conjecture, a complete answer also depends on the unwritten metaphysics of computer simulations. However, I will give an answer as to what kind of understanding an explanation of such a simulated phenomena yields.

The example was the simulation of a satellite whose value for the ‘big  $G$ ,’ the gravitational constant, is set to  $0 \text{ m}^3\text{kg}^{-1}\text{s}^{-2}$ . Given that the general computer simulation represents an empirical target system, and given that there are no restrictions imposed on the values of the initial and boundary conditions, it follows that a particular computer simulation set with such a value for the ‘big  $G$ ’ is perfectly conceivable. Now, although our imagination might soar with the number of impossible simulated phenomena that we can create, the general computer simulation imposes strict limits to our creativity. For the hypothetical case suggested here, for instance, there is no energy exchange nor measurable angular momentum,

resulting in a satellite that is not moving. Does this case approach a possible situation in the empirical world? Perhaps not. Perhaps no empirical world discovered by man would ever have a gravitational constant equal to  $0 \text{ m}^3\text{kg}^{-1}\text{s}^{-2}$ . However, scientists can still conceive of it as a thought experiment, by conceiving of systems where certain conditions are imposed and others relaxed: the energy and angular momentum equations are, after all, the same ones available from classical mechanics. This fact indicates that simulating an impossible phenomenon is not a capricious elaboration of the mind, rather it follows the principles of scientific method. Computer simulations, I then argue, become even more valuable for scientific activity since they can not only explain empirical phenomena but also (and within the limitations imposed by a reliable computer simulation) can elaborate on the universe of simulated phenomena that show what could have happened had the initial and boundary conditions been different.<sup>59</sup> Philosophers have argued that this feature of computer simulations makes them suitable as *thought experiments*.<sup>60</sup> I will not pursue this issue here any further, for that is in itself a new chapter in the philosophy of computer simulations. Let me now return to the main concern, that is, answering Kim's epistemic question in the context of impossible simulated phenomena.

Following the same deductive process, the lack of spike formation is explained by showing that there is no exchange between spin and orbital angular momentum around closest approach, all due to the new value assigned to the gravitational constant. Moreover, we are perfectly capable of understanding this case in a similar fashion: by unifying this simulated phenomenon with other simulated phenomena, and by seeing how its patterns of behavior are unified with classical Newtonian mechanics. Explaining what is *not* the case is as epistemically powerful as explaining what *is* the case, for in both situations we are in the position to reinforce our system of beliefs about a two-body system within classical mechanics. For these cases, computer simulations prove to be an essential method since simulating 'impossible systems' is a fairly simple matter. Moreover, by successfully explaining simulated phenomena for which we would otherwise have no explanation, we are reinforcing our system of belief in two ways: we are in the position to understand the deep structure of the world as well as to understand how the world cannot be (i.e., how it would have been had the circumstances been different). I conclude, then, that explanation in computer simulations is actually expanding our possibilities for understanding the world by elaborating on the domain of explaining possible phenomena as well as of explaining impossible phenomena. There are no doubts that this is the distinctive mark of an epistemologically powerful instrument.

### 5.3.3 A response to the skeptic

The philosopher that has grown skeptical about the epistemic power of computer simulations is hard to please. The motivations given in Section 1.4 about the philosophical novelty of computer simulations might not be enough. In addition, I must also defend my claims from their suspicion. I admit that this work is not immune to criticism, and there are still several issues in need of clarification and solution (see Chapter 6 for an outline of some postponed issues). However, limitations in space (and in intellectual capacity) impose a selection of topics available for discussion, especially for such a rich and complex set of ideas as the unificationist's.

To the best of my knowledge, there is no philosophical work on the explanatory power of computer simulations, or at least not to the degree that I have developed here. However, let me now try to reconstruct an objection that a forcibly imaginary skeptic could raise against this dissertation. Let us say that the skeptic begins by admitting my general notion of computer simulation (not really a commitment on their part since it is in conformity with philosophical and scientific notions), but denies that they are epistemically powerful. An outline of their argument would be something like this:

Computer simulations do not yield new understanding about the world because anything we put into them is already part of some scientific model. Since we created the model, we must know it very well. At best, the computer simulation makes the consequences of a model visible; that is, it shows the solutions within the space of solutions of the model, and allows us to see how the simulation acts given some boundary and initial conditions. Moreover, there are no guarantees that reckoning with a computer simulation will not lead to false results. This is perfectly conceivable because the computer simulation, as any other physical mechanism, is prone to errors and failure. Hence, what we learn, if anything at all, might as well be wrong and not representative of the world. Our epistemic gain is essentially zero.

Without indulging in false modesty, I believe this dissertation can answer these worries. But since it would be unnecessarily pedantic to review each chapter again, let me simply outline an answer.

The first two sentences of the skeptic's argument presuppose that a scientific model can be directly implemented on the physical computer. This assumption, as I have argued in Section 1.4, neglects the methodological novelty of computer simulations while taking the construction of the simulation model as a subtopic of the philosophy of models. This assumption is present in many criticisms of the epistemic power of computer simulations.<sup>61</sup> Furthermore, considering that we know the

results of a computer simulation because we constructed the model places, to my mind, considerable expectations on our cognitive capacities. For this kind of argument the usual response is that the time a computer takes to solve a simple equation is remarkably shorter than the time taken by a human. Moreover, the computational complexity demanded in computer simulations, even for simple equations, overpowers our cognitive capacities in every possible sense. This is probably the most shared opinion among philosopher nowadays.<sup>62</sup>

Let us say that the argument of computer simulations as *cognitive enhancers* is not entirely satisfactory. Let us say that the skeptic still argues in favor of having some sort of knowledge of the computer simulation stemming from their knowledge of the implemented model. This view is wrongly conflating *knowledge* with *understanding*; that is, to know what the results of the simulation are does not entail an understanding of them. A further epistemic step that provides such an understanding is still needed (e.g., by an explanation of the simulated phenomenon). Nevertheless, the skeptic still has one last line of defense: they can claim that a simulated phenomenon can in fact be explained using the scientific model. The problem with this view is that we stumble once again into the first assumption: because the skeptic wrongly assumes that a scientific model is directly implemented on the physical computer, they believe that such a scientific model explains (this objection has already been defused in my discussion on ‘explanatory models’ in Section 4.3.4). As I have argued in Chapter 5, the process of explaining with a computer simulation requires knowledge that exceeds the mere scientific model, such as information about round-off errors, truncation errors, discretization methods, and so forth.

Finally, I believe the skeptic is also wrong in thinking that because reckoning is prone to errors, computer simulations are therefore unreliable processes. They are wrong, but for the right reasons. Their reasons are that the scientific model, the methodology for building the simulation model, and the physical process of reckoning the simulation are prone to errors. In that sense, they are right. But so is any other physical machine, like a bubble chamber, or a thermometer. The best we can do to confer reliability to results (whether it is a thermometer or a computer simulation) is to minimize the sources of errors, reduce them by more robust mathematics and electronics, and measure them in order to make the results more comprehensible. Demanding a perfect reckoning machine is not a genuine (or fair) plea for computer simulations. Since, as I argued in Section 3.3.3, the source of errors is not limited to the computation but also includes the scientists designing the simulation model and programming the computer simulation. Scientific activity is carried out by humans that use imperfect instruments. The skeptic’s objection

therefore affects the scientific enterprise entirely.

## 5.4 Conclusion

In this chapter I synthesized the main results of previous chapters into my defense of the epistemic power of computer simulations. I argued that the explanatory power of the computer simulation could be granted by answering the *metaphysical* and *epistemic* questions posed by Kim. I also argued that the unificationist theory of explanation provides a suitable conceptual framework for computer simulations. Following the unificationist theory, then, I showed how an explanation of a simulated phenomenon could be derived from the general computer simulation. Since the general computer simulation genuinely represents an empirical target system, the explanation of a simulated phenomenon also applies as an explanation of an empirical phenomenon. I also showed in what specific sense explanation in computer simulations yields understanding of the world.

Although more work needs to be put into detailing the explanatory role of computer simulations, I believe that it has been shown beyond doubt that the epistemic power of computer simulations can be defended.

The future challenges that I foresee for my line of argumentation are compiled in Chapter 6. In addition, I also compiled a list of future challenges for the philosophy of computer simulations. While they do not necessarily relate to scientific explanation, they could be beneficial for alternative ways of defending the epistemic power of computer simulations.

## Notes

<sup>1</sup>See also (Neurath et al., 1955)

<sup>2</sup>Here the term “local” is borrowed from Nancy Cartwright (Cartwright, 1983) and refers to scientific work that is cut off in important ways from the natural world.

<sup>3</sup>See (Wylie, 1999).

<sup>4</sup>Cf. (Kitcher, 1999, 339).

<sup>5</sup>See the Lexicon.

<sup>6</sup>Cf. (Kitcher, 1999, 344).

<sup>7</sup>Cf. (Kitcher, 1999, 344).

<sup>8</sup>Cf. (Kitcher, 1999, 345).

<sup>9</sup>Cf. (Hacking, 1983, 209).

<sup>10</sup>Cf. (Franklin, 1981, 367).

<sup>11</sup>Franklin and Howson address the question about how much is ‘sufficient’ for an experiment to be reliable. See (Franklin and Howson, 1984).

- <sup>12</sup>Cf. (Kitcher, 1999, 346).
- <sup>13</sup>Cf. (Kitcher, 2013, 17). See also (Kitcher, 2001).
- <sup>14</sup>Cf. (Kitcher, 1993, 133).
- <sup>15</sup>Cf. (Kitcher, 1989, 453).
- <sup>16</sup>Kitcher has modified some of his views to accommodate criticism (Gonzalez, 2012). See also (Kim, 1988; Kitcher, 1992).
- <sup>17</sup>Cf. (Kitcher, 1999, 339).
- <sup>18</sup>See (Morrison, 2000).
- <sup>19</sup>(Hempel, 1965).
- <sup>20</sup>See (Toulmin, 1963), (Scriven, 1970), and (Dray, 1964).
- <sup>21</sup>There are many voices against unification in the sciences (for instance (Cartwright, 1994; Galison, 1996)) as well as many in favor (for instance, (Kitcher, 1999; Morrison, 1990)). Admittedly, taking computer simulations as unificatory systems requires more discussion and a deeper understanding than what I have offered here. However, above and beyond this claim, computer simulations still adapt to the unificationist fairly well, for instance, because it is an epistemic theory of explanation, one that could actually account for simulated phenomena. For the remainder of this work, then, I will take computer simulations as unificationist systems without providing any further proof.
- <sup>22</sup>Let us remember that I have discarded other unificationists theories, such as the “unofficial” version of the Deductive-Nomological account on a different basis.
- <sup>23</sup>The main objection to Friedman’s account was that he took ‘unification’ as ‘unifying laws’ rather than ‘phenomena,’ as Kitcher suggests. For the complete discussion on this problem, see (Kitcher, 1989), and also (Ruben, 1992).
- <sup>24</sup>Cf. (Kitcher, 1989, 432).
- <sup>25</sup>At the risk of being repetitive, it is worth pointing out the two senses in which I am using the term ‘knowledge.’ If we recall from Section 4.2.1, I discussed the conditions that must be met for a reliable computer simulation. Since these conditions were analyzed in the context of traditional epistemology, it was natural to say that the scientists know that the simulated phenomena represent the target system. The sense in which ‘knowledge’ is being used here is, however, slightly different for it now refers to the total corpus of information that a scientific discipline has at any given moment in time.
- <sup>26</sup>Kitcher also refers to the explanatory store  $E(K)$  as the *general argument pattern*. Cf. (Kitcher, 1989, 432).
- <sup>27</sup>Kitcher indicates that the condition of *completeness* is for excluding explanatory deviants that use patterns selectively. Cf. (Kitcher, 1981, 176).
- <sup>28</sup>Cf. (Kitcher, 1981, 177).
- <sup>29</sup>Cf. (Kitcher, 1981, 521).
- <sup>30</sup>Cf. (Kitcher, 1989, 477ff).
- <sup>31</sup>See the example in (Kitcher, 1989, 442).
- <sup>32</sup>Let it be noted that Kitcher endorses Friedman’s account of understanding. In this sense, I will be using both authors for clarifying the unificationist position on the issue.
- <sup>33</sup>Examples of this are (Bridgman, 1927; Campbell, 1920; Dray, 1964).
- <sup>34</sup>Cf. (Friedman, 1974, 10).
- <sup>35</sup>Cf. (Barnes, 1992, 3).
- <sup>36</sup>Cf. (Friedman, 1974, 18).
- <sup>37</sup>Cf. (Friedman, 1974, 9).

<sup>38</sup>Cf. (Friedman, 1974, 18).

<sup>39</sup>Cf. (Kitcher, 1989, 432).

<sup>40</sup>Cf. (Kitcher, 1989, 432).

<sup>41</sup>Cf. (Kitcher, 1981, 508).

<sup>42</sup>Cf. (Kitcher, 1981, 507).

<sup>43</sup>Cf. (Kitcher, 1989, 432) and (Friedman, 1974, 15).

<sup>44</sup>Cf. (Fahrbach, 2005, 459).

<sup>45</sup>Cf. (Fahrbach, 2005, 457).

<sup>46</sup>Cf. (Barnes, 1994, 62).

<sup>47</sup>See, for instance (Gonzalez, 2012; Kaiser and Seide, 2013).

<sup>48</sup>As I have argued in Section 3.3.1.1, the conceptual difference between a *simulation model* and the *general computer simulation* is that the former is conceived as a pen-and-paper entity and, as such, is admissible for formal verification and validation (see Section 3.3.2); the *general computer simulation*, on the other hand, is a simulation model ‘loaded’ on the physical computer and ready to be set up with input and boundary conditions. There is no harm in using these terms interchangeably when the context is clear.

<sup>49</sup>Patterns here are understood as descriptions that single out natural kinds, objective causal relationships, objective natural necessities, and similar concepts that philosophers use for accounting for the metaphysics of science. As Kitcher puts it, “they are schemata that correctly capture objective dependencies [of phenomena in the empirical world]” (Kitcher, 1993, 150). See also (Kitcher, 1986, 1994).

<sup>50</sup>See (Humphreys, 2004, 147).

<sup>51</sup>Let me note here that I am thinking of *one* explanation that applies to the two domains, as opposed to *two* explanations that are somehow epistemically equivalent.

<sup>52</sup>See Sections 1.2.2, 1.2.2.1, and 3.3.

<sup>53</sup>Recall from Section 4.2.1 that I ruled out computer simulations for which no genuine and well-known set of equations exist.

<sup>54</sup>Cf. (Kim, 1994, 54).

<sup>55</sup>Although these ideas are strongly metaphorical, it is unproblematic to show in what sense they are being used. Since current literature has discussed these two ideas to some extent, I dismiss myself from an analysis that will not contribute any new insight. See, for instance, (Arkoudas and Bringsjord, 2007; Białynicki-Birula and Białynicka-Birula, 2004; Winsberg, 2010).

<sup>56</sup>Admittedly, this point raises interesting questions that are at the center of current philosophical discussions on scientific explanation (see, for instance, (Hahn, 2011)). Unfortunately this is not the place for discussion.

<sup>57</sup>Cf. (Kitcher, 1989, 432).

<sup>58</sup>This point has special kinship with my discussion on *syntax manipulability* in Section 2.3.

<sup>59</sup>The way I fostered this issue shows the seemingly *counterfactual* nature of computer simulations. I have nothing illuminating to say about this point since more analysis needs to be done.

<sup>60</sup>Cf. (Ristić and Radulović, 2001).

<sup>61</sup>Most notably (Frigg and Reiss, 2009).

<sup>62</sup>For instance, in the work of (Paolo et al., 2000; Humphreys, 2004; Morrison, 2009; Reiss, 2011; Winsberg, 2010).

# Chapter 6

## Conclusions and future challenges

### 6.1 Conclusions

The conclusions for this work are sober. The main motivation was to defend the epistemic power of computer simulations while avoiding the temptation of comparisons with laboratory experiments. I believe that these two aims have been accomplished in a successful manner. The former by analyzing in detail the nature of computer simulations and showing how they yield understanding of the world; the latter by fathoming their role as explanatory devices.

My first initiative was to confront current literature on the topic. The overwhelming majority of studies on the epistemic power of computer simulation are interested in comparisons *vis-à-vis* with laboratory experiments. This is a natural consequence of a philosophy of science predominantly populated and governed by empiricists.

My approach is rather different. I chose to address the epistemic value of computer simulations at face value. In this vein, the conclusion of the first chapter is that there are epistemic differences between computer simulations and laboratory experiments worth exploring by themselves. What follows from this result is the need to be as specific as possible on the conceptualization of the notion of ‘computer simulation.’ Current literature has made a loose use of this notion; typically, it has been interpreted as scientific models directly implemented on physical machines for crunching large amounts of data.<sup>1</sup> There are two central ideas in that interpretation; the first is plainly wrong and the second is emphasizing the incorrect virtue of computer simulations. Indeed, the idea that a computer simulation is the direct implementation of a scientific model on the physical machine ignores methodological, semantic, epistemic, and even ontological requisites involved in the transformations of such a model into a simulation model.<sup>2</sup> On the other hand, the idea that a com-

puter simulation has an enormous power for computing large amounts of data is fundamentally correct; my observation is, however, that emphasizing this epistemic virtue over the capacity to represent a variety of systems (e.g., empirical, mathematical, heuristic, and so forth) leads to the idea that computer simulations are devices that enhance our cognitive capacities to the detriment of their representative capacities.<sup>3</sup> Computer simulations as cognitive enhancers chimes with comparing the limitations of laboratory experiments leading, in turn, to more philosophical comparisons. The shift in the viewpoint proposed here stems from considering computer simulations first and foremost as representational systems of a given target system.

My viewpoint is vindicated with a study on the notion of ‘computer simulation’ within the philosophy of computer science. Arguably a young and, to many, unknown branch of philosophy, the philosophy of computer science aims at achieving a better understanding of computer software in terms of their ontology, semantics, methodology, and epistemic values. In this context, I opted for dividing the notion of ‘computer simulation’ into three units of analysis: the *specification*, where the scientists include their scientific model and design decisions of the computer software; the *algorithm*, which in essence is the specification interpreted in a machine readable-language; and finally the *computer process*, which consists in the logical/physical process taking place on the physical computer. This division proved to be of chief importance for several reasons: first, because it clarified the otherwise vague notion of ‘computer simulation’ used in the literature; second, because it entrenched computer simulations as systems with which we represent a multiplicity of target systems; third, because it established computer simulations as closed systems, that is, ‘worlds of their own’; fourth, and perhaps most important, because it showed that the specification and the algorithm are two units of analysis of the simulation to which we do have cognitive access. They are, using common parlance, ‘epistemically transparent.’ This last point has special importance for the evaluation of the explanatory power of computer simulations.

A working conceptualization follows from this analysis. In order to add precision to this conceptualization, I had to introduce new terminology. The specification and the algorithm, which were conceptualized as the *simulation model* in the sense of a pen-and-paper logic/mathematical unit feasible of formal verification, are now conceptualized as the *general computer simulation*. In other words, a general computer simulation is a piece of computer software with the initial and boundary conditions *not* yet fulfilled. For instance, a simulation of a two-body system is a general computer simulation insofar as the initial and boundary conditions are not fulfilled; therefore, it represents all two-body interacting systems. When the initial

and boundary conditions are fulfilled, we have one *particular computer simulation* that instantiates a specific simulated phenomenon, for instance, the example of the satellite under tidal stress in Section 3.3.1. With all these elements in place, I could defend the epistemic power of computer simulations.

For this, I chose to analyze the role of computer simulations as explanatory devices. Scientific explanation occupies a special place in the studies of general philosophy of science, and due to the relevant place that computer simulations are gaining in today's scientific practice, it is essential for philosophers and scientists alike to recognize the importance of being able to explain a simulated phenomenon. Such an analysis becomes more pertinent (and difficult) when a quick overview of the current literature shows that no work on this topic has been done to date.

In this work I showed how computer simulations act as explanatory devices. In addition, and given that a good theory of explanation also prompts understanding of the phenomenon explained, I showed how we gain understanding of the explained simulated phenomenon. Arguing for epistemic insight into the simulated phenomenon was certainly the major difficulty that this work had to face. The main problem was related to the metaphysics of realistic accounts of scientific explanations, which defend that a successful explanation is one that identifies the true structure or basic mechanisms of the empirical phenomenon.<sup>4</sup> The question for computer simulations is, therefore, what kind of 'phenomenon' comes out of a particular computer simulation? What is being explained and thus understood when explaining a simulated phenomenon? In Section 3.2 I argued that a computer simulation creates 'worlds of its own' in the sense that its results might (or might not) be related to an empirical phenomenon. The problem, then, was to establish conditions that provided some epistemic guarantees that the simulated phenomenon is a genuine representation of the empirical phenomenon. If such conditions can be established, then we can say that we have explained and thus understood something about the world. With this problem in mind, I set further conditions on the class of computer simulation: a general computer simulation must be *reliable* in the sense of providing some epistemic guarantees that the simulated phenomenon genuinely represents an empirical phenomenon.

Admittedly, this point might raise some questions about the metaphysics of computer simulations. And since there is no philosophical work on the issue that could be of reference, I opted to establish further conditions on the class of computer simulations of interest. Concretely, in order to ensure that an explanation yields understanding of the world, I argued for two conditions for a reliable computer simulation: a good representation of the target system by the general computer

simulation, and the minimization of mathematical artifacts during the computation. The importance of a reliable computer simulation is that it justifies our belief in a simulated phenomenon that represents something ‘out there.’ Unfortunately, these conditions also create some metaphysical riddles. Indeed, a general computer simulation could produce a *real* simulated phenomenon, such as the example of the satellite under tidal stress, or it could produce an *impossible* simulated phenomenon, such as a satellite under tidal stress with the gravitational constant equal to  $G = 0 \text{ m}^3\text{kg}^{-1}\text{s}^{-2}$ . The metaphysical riddle, then, is the following: the previous conditions were established to ensure that the simulated phenomenon genuinely represented an empirical phenomenon, hence, explaining and understanding the former entails explaining and understanding the latter. However, since there is no known empirical system where the gravitational constant is equal to  $G = 0 \text{ m}^3\text{kg}^{-1}\text{s}^{-2}$ , the question now is what kind of understanding are we obtaining by explaining such an impossible simulation? Since the second simulation is within the space of solutions of a reliable general computer simulation, then we should be able to unify such a simulated phenomenon with our corpus of beliefs in the same manner as we did with the explanation of the spikes of the satellite under tidal stress. Indeed, we explain it in the usual way: by seeing connections and common patterns of what initially appeared unrelated. For instance, with a gravitational constant equal to  $G = 0 \text{ m}^3\text{kg}^{-1}\text{s}^{-2}$  there is no exchange between spin and orbital angular momentum around closest approach, and therefore there is no formation of spikes. Such an absence of spikes can be explained and unified with the total corpus of classical Newtonian mechanics in the usual ways. Explaining what is *not* the case is as epistemically powerful as explaining what *is* the case, for in both situations we are in the position of reinforcing our system of beliefs about any two-body system.

A criticism that might arise is that I did not provide an account of explanation in computer simulations in itself; rather, I borrow the conceptual framework from the unificationist. I have outlined the motives that led me to argue that there is no need for such an account. A theory of scientific explanation has aspirations that computer simulations, to my mind, cannot fulfill. One of these aspirations is to cover a significant number of disciplines, mostly disparate: physics, astronomy, and chemistry, but also economics, psychology, and biology. The class of computer simulations for these latter disciplines is also quite disparate from the former. Cellular automata, agent-based simulations, and complex systems are simply a different species than equation-based simulations, and their explanatory nature is due, precisely, to the characteristics that they have to offer. This, of course, does not entail that agent-based simulations, for instance, cannot explain simulated biological phe-

nomena. My point is, instead, that some considerations must be in place before they can be included into my explanatory account of computer simulations. Any ambition for a stand-alone theory of explanation for computer simulations seems to me, for the time being, to be more an aspiration than a feasible end.

A more general and equally interesting consequence of my account on computer simulations is the challenge posed to a rather empiricist-shaped view on philosophy of science. Computer simulations represent (as opposed to interact with) the world. A successful account of explanation in computer simulations (along with successful accounts of evidence, observation, measurement, prediction) substantiate a more *rationalist* position. The trade-off is simple: rather than demanding information about the world from our experience, computer simulations obtain it by representing the world. The idea is certainly appealing, and to my mind it is feasible to carry it out to a good end. A more rationalist view, then, is also the consequence of addressing the new philosophy of computer simulations as emancipated from more familiar philosophies.

As I mentioned at the beginning, my conclusions are sober. Here I reconstructed the central arguments and highlighted some of their most conflictive points. In the following sections I address some topics that have not been discussed here. For the sake of clarity, I divided them into two sections: future challenges for the unificationist account of explanation, which outlines issues that I did not address in this work; and future challenges for the philosophy of computer simulations, whose intention is to encourage lines of work for those that are not convinced by my view of explanation but are persuaded by the epistemic power of computer simulations.

## 6.2 Future challenges

Philosophers typically do not consider their work free of potential objections and shortcomings. This work is, naturally, no exception. I have claimed throughout this study that computer simulations are epistemically powerful. I have defended this claim by showing how they explain simulated phenomena and, in doing so, how scientists are in the position to understand that piece of the world. Such a claim, however, carried a great number of philosophical riddles that I have tried to answer across this work. Admittedly, this study has left some questions unanswered. This section, therefore, is my attempt to redeem shortcomings and circumvent potential objections, or at least those of which I am aware. With this in mind, I divided this section into the following: Section 6.2.1 addresses some issues related to the unificationist account for computer simulations; I will also return to the constraints

imposed for explanation on simulations and discuss possible alternatives. Section 6.2.2 brings back a lost claim, the one that argues that the defense of the epistemology of computer simulations could be approached from several angles. I had chosen scientific explanation, but this section outlines a few more angles worth consideration.

### 6.2.1 For the unificationist

In this work I have defended the unificationist account of explanation as a successful theory for explaining simulated phenomena. I have also shown that through the epistemic goal that this account of explanation carries with it, it is possible to confer epistemic power to computer simulations. However, there are a handful of unsolved issues related to the unificationist account and inherited by computer simulations that have not been addressed in this work. In this section I will briefly outline them while possible solutions are also suggested. Let me note that most of them do not represent a threat to my view of explanation of computer simulations results insofar as they are not a threat to the unificationist account. Naturally, an exhaustive analysis must not be expected; rather these issues are addressed as future challenges that the unificationist account poses for computer simulations.

Allow me to begin with a claim already mentioned in Section 5.2 about the world as a disunified place. John Dupré, for instance, has extensively written on this topic, denying the view that science constitutes a single, unified project. According to him “[t]he metaphysics of modern science, as also of much of modern Western philosophy, has generally been taken to posit a deterministic, fully law-governed, and potentially fully intelligible structure that pervades the material universe” (Dupré, 1993, 2). At the beginning of that section I argued that complete unity in science was a program that proved to be wrong. Instead, the image of the unificationist that Kitcher projects and that I support is one of ‘modest unificationist’; that is, the philosopher that, committed to unity, also leaves room for the belief that the world may be a disordered place. As argued, the modest unificationist believes that in many cases scientific disciplines share regular methodologies, common ontologies, and comparable epistemic results, all of which increases our understanding of the world, advances science, and strengthens our corpus of belief. The question is in what sense are computer simulations better off by adopting the modest unificationist rather than following Duprés’ strategy? I must here make a further distinction in the notion of ‘unification.’ Margaret Morrison distinguishes two different types of unification: “*reductive unity*, where two phenomena are identified as being of the same kind (electromagnetic and optical processes), and *synthetic unity*, which in-

volves the integration of two separate processes or phenomena under one theory (the unification of electromagnetism and the weak force)” (Morrison, 2000, 5. Emphasis mine.). With this distinction in mind, computer simulations fulfill both types of unification indicated above: by writing an algorithm that describes Maxwell’s equations one would be perfectly capable of explaining electromagnetic as well as magnetic phenomena; and by making use of different libraries and alternative software for reproducing a larger number of phenomena under the same simulation, the researcher captures the second type of unification. Computer simulations are well-known for their capabilities to bring a host of diverse phenomena under the same umbrella: simulations in economy are used for psychology and vice-versa, biology and evolutionary sociology are also a close match. Sometimes, by implementing small changes in the mathematics, new equations represent a completely different phenomena. This was my way of understanding the unificationist power of computer simulations. Dupré’s objection to the unified world does not reach the modest unificationist in the physical sciences nor does it apply to the modest unificationist that runs a computer simulation. Moreover, his objection does not even reach computer simulations applied to the biological sciences, the field that Dupré uses as a counterexample of a unified world.

Another challenge for computer simulations is the issue of *comparative explanations*. It is well known that the world of simulations changes constantly. Moore’s law, for instance, indicates that the power of computers doubles every 18 months. Such changes have several sources: either because the mathematical machinery used for computers changes over time or, more likely, because technology advances rapidly, providing new architectural support for faster and more powerful computers, along with a number of flourishing programming languages.

Such scientific and technological development provokes a flux of changes in computer simulations as well. An interesting question is how those changes affect explanation in computer simulations. In other words, if two or more simulations are competing to explain a similar phenomenon, is there any available criteria for deciding which simulation explains better? At this point it must be restated that my work here has been on *one* computer simulation producing one set of results that represents one phenomenon. The problem that I am posing here adds one dimension; namely, a question about *multiple* simulations of the same target system competing for explanation. In terms of the unificationist account, this means that two or more *explanatory stores* are competing for explanation.

To the modest unificationist, the choice of one argument pattern over another

lies in the comparison regarding their unifying power. This unifying power depends, in turn, on the paucity of patterns used, the number of descriptions derived, and the stringency of patterns.<sup>5</sup> In previous chapters, characterizing explanatory unification in terms of the fewer patterns used and number of descriptions derived was paramount for a successful account of explanation. The process of deciding between rival explanations requires, now, the notion of *stringency of patterns*. Indeed, in the unificationist account, competing explanations are possible because it presupposes that two or more derivations are *similar* in either of these ways: they are similar in terms of their logical structures (i.e., in the statements used for constructing the argument pattern), or they are similar in terms of the non-logical vocabulary. The problem, then, is how to interpret the notion of *similarity* in such a way that it is not reduced to a matter of degree.

The solution proposed by Kitcher is to recognize that one explanatory store can demand more or less from its instantiations than another explanatory store. Kitcher then introduces the notion of *stringency* as a way to decide between those competing patterns. The basic idea is that a pattern is more *stringent* when it imposes conditions that are more difficult to satisfy than those of other patterns. Kitcher continues describing the notion of stringency in the following way:

the stringency of an argument pattern is determined in part by the classification, which identifies a logical structure that instantiations must exhibit, and in part by the nature of the schematic sentence and the filling instructions, which jointly demand that instantiations should have common nonlogical vocabulary at certain places (Kitcher, 1989, 433).

In a similar vein, a computer simulation that has more unifying power would be preferable over one with less unifying power. But, what exactly does this mean? In the domain of computer simulations, it is not enough to produce more successful results with the use of fewer patterns, as Kitcher argues for science, but more accurate simulations are also preferred due to their capacity to produce more representative results. To illuminate this point, consider any model with at least one ordinary differential equation. In order to implement this model as a computer simulation, it first needs to be discretized. For the sake of the argument, let us consider only Euler and Runge-Kutta methods. While the Euler method is subject to large truncation errors, Runge-Kutta is much more accurate in its results. Given that accuracy of results is also representational of the target system, then, the more accurate a result is, the more representative of the target system they become (and vice versa). To the unificationist, accuracy is interpreted in terms of the nature of the schematic sentence and the filling instructions. A computer simulation using Runge-Kutta methods of discretization provides better schematic sentences for they

are more representative of the target system; this is also true in terms of the filling instructions because the simulation would provide more detailed results. Accuracy and representativeness are coextensive in computer simulations, and this is reflected in the stringency of the patterns of behavior. It follows that the patterns of behavior of the simulation implementing the Runge-Kutta method can better explain the results than implementing Euler and, as such, it unifies more.

Of course, a more complete analysis of the criteria for choosing one explanation over another in the context of computer simulation requires more study. Here I am only outlining the challenge that could be of greatest interest to discussions for a unificationist view on computer simulations.

The problem of competing explanations has some kinship with changes in the corpus of beliefs  $K$ . My account has been based on a fixed corpus of beliefs at one moment in time. But, as I mentioned before, mathematics changes, and so does technology. Therefore, it is to be expected that our corpus of beliefs also changes. Recall from Section 5.2, I briefly discussed how to seek the best  $E(K)$ , the explanatory store. But the analysis I propose here has less to do with finding the best explanatory store than it has to do with changes in the corpus of beliefs  $K$ . Let me recall that I reconstruct  $E(K)$  from the *specification* (and the *algorithm*), where all the information about the simulation lies. If the *specification* (and therefore, the *algorithm*) changes, it must be interpreted as a change of the explanatory store  $E(K)$ . Now, such a change in the explanatory store has nothing to do with finding the best  $E(K)$ , for this was assumed as a condition, but with the change in the corpus of beliefs  $K$ . In plain words, different computer simulations presuppose a change in the specification and in the algorithm, which presupposes in turn a change in the corpus of beliefs  $K$ . A theory of explanation for computer simulations must also account for changes in the specification and in the algorithm. This is a very difficult and complex issue within the heart of the unificationist account. Kitcher discusses it to some extent, but it requires a discussion in itself due to the amount of technical terms that need to be introduced.<sup>6</sup> This problem will also be left as a challenge for future studies on explanatory unification for computer simulations.

The last challenge lies in my own selection of computer simulations, which has two parts. The first part goes back to Chapter 3 where I narrowed down the class of computer simulations for this study. This selection explicitly excluded cellular automata, agent-based simulations, and complex systems. Narrowing down the class of computer simulations was the proper course of action for my purposes.

But in any future study, these classes of simulations must be incorporated under the unificationist account of explanation as well (or at least analyzed with the possibility of an incorporation). As indicated in Section 3.2.2, this move would require a better analysis of these classes of computer simulations, one that may diverge from what I argued in Chapter 2 and Chapter 3.

The second part of this selection has to do with the simulations that I excluded in Section 4.2.1. There I argued that the metaphysics of scientific explanation demanded us to be very careful in guaranteeing that the results of a simulation actually represented the phenomenon to be explained. This demand was based on epistemological grounds: without the proper representation we cannot claim that an explanation yields understanding of that piece of the world just simulated. Computer simulations, however, are rich heuristic systems, understood as mechanisms for exploring the properties of models.<sup>7</sup> Krohs poses another interesting example: the Oregonator, a simulation of the Belousov-Zhabotinsky chemical reaction.<sup>8</sup> The problem with this simulation was that the system of equations were *stiff* and lead to qualitatively erroneous results.<sup>9</sup> It would be interesting to develop a theory of explanation that includes these simulations as genuine producers of explanation despite the fact that they do not represent. This last point, I suspect, has direct connections with considering computer simulations as *a priori* experiments, where the reality of the world is subsumed to the rationality of representing it. The intricacy of the problem speaks for itself.

## 6.2.2 For a general philosophy of computer simulations

The strategy of analyzing computer simulations in light of different concepts in philosophy of science can be extended beyond explanation. Among these concepts, we can single out *evidence*, *prediction*, *observation*, *exploration*, and *confirmation*.<sup>10</sup> There are also concepts that are not necessarily related to the philosophy of experiment, and which have a much broader scope, such as *progress of science*, *the a priori*, and *non-empirical science*, just to mention a few.

The list can be certainly extended, and naturally some concepts will present more challenges than others. But overall, if our interest as philosophers is to understand the role that computer simulations play in scientific practice, a good starting point is to accept the challenge of current philosophy of science and work our way out from there. Let me briefly illustrate these challenges with the notion of *evidence* and how it could be reworked in the context of computer simulations.<sup>11</sup>

Contemporary accounts of *evidence* relate the world (or pieces of the world) to a statement. For instance, a mark of 45<sup>o</sup> C in a thermometer is *evidence* for the

sentence “today is a hot day.” In this vein, some philosophical accounts of evidence prefer to describe what good evidence achieves, whereas others focus on methods by which evidence is generated. The former are necessarily *post hoc* in that one must already possess and have evaluated the evidence in order to do business in this tradition;<sup>12</sup> the latter is prior to the considerations of how evidence relates to hypotheses or theories.

The first account is typically found in two versions: the *probability definition* and the *explanation definition*.<sup>13</sup> Briefly, the *probability definition*  $e$  is evidence for some hypothesis  $h$ , given background  $b$ , if and only if  $p(h|e.b) > p(h|b)$ ; in other words, the probability of the hypothesis given  $e$  and  $b$  must be greater than the probability of the hypothesis without  $e$ , if and only if  $e$  is to count as evidence for the hypothesis. The more compelling or confirming  $e$  is, the greater the inequality between  $p(h|e.b)$  and  $p(h|b)$ .<sup>14</sup> Explanation definition, on the other hand, takes  $e$  to be potential evidence that  $h$ , if and only if  $e$  is true; in turn,  $h$  would correctly explain  $e$  if  $h$  were true.<sup>15</sup> These accounts (or any varieties of them), provide *post hoc* characterizations of good evidence for a hypothesis, rather than a guidance for the *production* of it. For the latter task we need to turn to methodological features for obtaining good evidence, a different chapter in the philosophical literature on evidence.

Two common features about *evidence* come from these assumptions: i) that evidence points beyond itself; and ii) that evidence is a relationship between a sentence and (a piece of) the world.<sup>16</sup> The first assumption must be held if we want to avoid any circularity (i.e., evidence cannot be evidence for itself but must refer to something else). Therefore, the measurement of 45° C on a mercury thermometer can be evidence that the air temperature is hot, or that the mercury expanded a certain amount in space; but it cannot be evidence for itself (i.e., that it measured 45° C). The second assumption is also very strong in either account of evidence: whether the *probability definition* or the *explanation definition*, both take evidence as causally generated (this is clearer for cases of production of evidence). In any case, causal interaction with the world grounds the relationship between a sentence and that piece of the world.

The challenge for computer simulations is to analyze these assumptions and to evaluate in what sense the data produced is evidence of the simulated phenomenon. For instance, the second assumption entails that *evidence* is causally generated. Thus, the measurement of the thermometer is actually due to the causal relation existing between the air temperature and the mercury. In what sense, then, can a non-causal device such as a computer simulation be evidence of a phenomenon in the

world? A possible solution to this problem lies in restricting the notion of evidence to a counterfactual situation. For instance, one can simulate what the measurement of the thermometer would have been, had the air temperature been such and such. In these kinds of systems, there is no need for causal evidence for hypothetical systems. Another possible solution is to modify the notion of evidence entirely, leaving some room for the data that computer simulations produce as evidence of the models implemented. This last solution has some kinship with the idea that computer simulations might have some use as *thought experiments*.<sup>17</sup> I do not have the answers to these issues.

The general accomplishment of this study is that our understanding of the world is crossed by different methods and practices, all of which help in the general understanding of it. The challenge for computer simulations is to show how they fit into the larger image of scientific practice and what sort of understanding they yield.

Finally, I would like to borrow Philip Kitcher's words since, I believe, they express quite eloquently the feelings that most philosophers have at the end of their own work: "I doubt that I have done more than scratch the surface of unfamiliar terrain, but I am confident that epistemological rewards await those who are prepared to dig more deeply" (Kitcher, 1993, 389).

## Notes

<sup>1</sup>There is another misleading interpretation presented by Wendy Parker that claims for the physical machine bringing about the empirical phenomenon (Parker, 2009). See (Durán, 2013b) for details.

<sup>2</sup>This is basically Hartmann's claim (Hartmann, 1996). Parker (Parker, 2009) and Guala (Guala, 2002) follow this view.

<sup>3</sup>This is the view developed by Humphreys (Humphreys, 1991, 2004), Morrison (Morrison, 2009), and Rohrlich (Rohrlich, 1990). Lenhard (Lenhard, 2006) and Winsberg (Winsberg, 2009a, 2010) have a similar view on the matter.

<sup>4</sup>The interpretation of the notion of 'basic mechanism of a phenomenon' depends on the theory of scientific explanation: for the ontic account, for instance, it is identified with the set of causal relations that bring the phenomenon about (Salmon, 1984); for the unificationist, it consists in identifying the set of patterns of behavior that unify the phenomena. Note that the unificationist account takes unification as the systematization of the complete set of truths about the empirical world; that is, of a language that picks out genuine natural kinds, that includes all the true statements in that language (Kitcher, 1989, 495).

<sup>5</sup>Cf. (Kitcher, 1989, 478).

<sup>6</sup>See, for details (Kitcher, 1989, 447ff).

<sup>7</sup>See, for instance (Velasco, 2002; García and Velasco, 2013).

<sup>8</sup>See, for instance (Field and Noyes, 1974).

<sup>9</sup>See (Krohs, 2008, 281).

<sup>10</sup>It is worth noting that the unificationist account has also been used for explicating *confirmation* (Thagard, 1992).

<sup>11</sup>An illuminating example of the use and need of evidence in computer forensics is (Brown, 2006).

<sup>12</sup>See, for instance (Stegenga, unpublished).

<sup>13</sup>See also (Staley, 2004; Morrison, 1992; Maull, 1976; Janssen, 2002).

<sup>14</sup>Cf. (Achinstein, 1983, 328).

<sup>15</sup>Cf. (Achinstein, 1983, 335).

<sup>16</sup>See (Achinstein, 2001).

<sup>17</sup>For instance (Ristić and Radulović, 2001).

# Summary (English)

Computer simulations have proven to be a fundamental tool for the advancement and development of scientific practice. This is especially true in areas related to scientific experimentation, where computer simulations stand out with a strong presence. Such pervasiveness in the sciences draws the attention of scientists and philosophers both interested in understanding their epistemic power. Naturally, one can find advocates for as well as detractors against computer simulations, holding strong arguments against each other. This work is my attempt to build a systematic defense of the epistemic power of computer simulations, not only against the attacks perpetrated by its critics, but also for providing a substantial assessment of the claim about their epistemic power.

With these motivations firmly in mind, the general aim of this work is to show that computer simulations yield understanding of some aspects of the world by explaining simulated phenomena. Since understanding is an epistemic concept *par excellence*, and since there is general agreement that scientific explanation is a coherence-making process, it follows that the epistemic power of computer simulations can be defended by showing in virtue of what computer simulations explain and what kind of understanding they yield. In this context, I have chosen the unificationist theory of scientific explanation as a conceptual framework for explanation in computer simulations. Such is the general line of argumentation. In this thesis, I address the following questions:

1. In which respects do computer simulations differ from laboratory experimentation and why do they represent a novelty for the philosophy of science?
2. What are the advantages that computer software offer, which are absent in laboratory experimentation but which are beneficial for conceptualizing the epistemic power of computer simulations?
3. What is the nature of a computer simulation? Is it abstract, due to the mathematical/logical nature of algorithms? Or is it causal, due to the nature of the computer program?

4. What are the conditions that must be imposed onto a computer simulation in order for it to confer reliability as a process that genuinely represents an empirical target system?
5. How is the explanatory process carried out and in what sense does such an explanation yield understanding of the simulated phenomenon as well as the empirical phenomenon?

These questions parallel the structure of this work. The first question is addressed in Chapter 1, where I analyze current philosophical literature on models and scientific experimentation, with a strong emphasis on methodological and epistemological differences with computer simulations. In this chapter, I center on differences *vis à vis* laboratory experiments, urging for a change in the way philosophers approach the epistemology of computer simulations. Concretely, I urge for analyzing computer simulations at face value rather than by means of comparing them with laboratory experiments. The final section is dedicated to outlining some answers to the skepticism regarding the novelty of computer simulations in the philosophical arena.

Motivated by these results, I propose to systematically construct a working conceptualization of computer simulations; this undertaking begins in Chapter 2 and continues in Chapter 3. For this, I begin by studying the nature of computer software as expounded by philosophers of computer science. Three units of analysis are of major interest; namely, the *specification*, the *algorithm*, and the *computer process*. These units of analysis are meant to answer the second question above by showing the benefits of computer software (e.g., *syntax manipulability* and *syntax transference*) absent in laboratory experimentation that speak in favor of the epistemic power of the former. In addition, understanding the nature of computer software facilitates the conceptualization of computer simulations. In this sense, an evaluation of each unit of analysis and the relations among them is essential for the robustness of this work.

The third question is more complex since it is at the heart of the nature of computer simulations and, in a way, determines their fate as epistemic devices. For these reasons, Chapter 3 is intended to work on several levels: first, it reviews the literature on *simulations*, examining the differences between *analog* and *digital* simulations. Second, it narrows down the universe of all computer simulations that can be found in scientific practice, dividing it into equation-based simulations on the one hand, and cellular automata, agent-based, and complex systems on the other. This distinction is meant to restrict the domain of computer simulations of interest to equation-based simulations. Third, it discusses the methodology of computer simulations and analyzes one concrete paradigmatic example of the class of

equation-based simulation. Finally, it constructs a ‘working conceptualization’ that entrenches a notion of computer simulation valid for the sciences and for philosophy.

Up to this point, the philosophical effort has been focused on clarifying and constructing a conceptualization of computer simulation. With this in place, I am now in the position to address the explanatory value of computer simulations (Chapter 4 onwards). It is worth noting that the focus now shifts from a philosophy of computer science and a philosophy of experimentation towards a philosophy of science, philosophy of mathematics, and epistemology.

One of the results obtained from Chapter 3 is that computer simulations are a ‘world of their own’; that is, systems that do not necessarily represent an empirical system. This is a beneficial and desirable characteristic of computer simulations since scientists are interested in simulating hypothetical systems, or exploring the heuristics of their models regardless of empirical representation. However, the metaphysics of explanation indicates that explanation (and understanding) is of something ‘out there,’ ‘in the world.’ The question then is how can we show that the explanation of a simulated phenomenon is also applicable to the empirical phenomenon and, hence, yields understanding of what is ‘out there’? In other words, if the epistemic power of computer simulations is conferred by showing how they explain and yield understanding of a simulated phenomenon, and if the metaphysics of explanation demands for an explanation and understanding of what is ‘out there,’ then it is paramount to find the conditions under which a computer simulation is a reliable process that genuinely represents an empirical target system. The aim here is to provide epistemic guarantees that by explaining a simulated phenomenon we are also explaining an empirical phenomenon, and that by understanding the former we are also understanding the latter.

As stated, this problem involves theories of representation (which I obliquely address through the working definition of *goodness of representation* in Section 1.2.2.1), and mathematical dependability on the reckoning process (which I discuss in Chapter 3 and through the notions of *verification* and *validation*). The way to obtain genuine representation of a target system is by framing the computer simulation within the notion of a *reliable process*, which includes solutions to the problems of representation and reckoning aforementioned. A computer simulation is then reliable when the simulated model is a good representation of a target system and the reckoning process does not introduce mathematical artifacts in the production of the simulated phenomenon. Only under these conditions is one entitled to claim that the simulated phenomenon genuinely represents the empirical phenomenon.

Equally important is to show in virtue of what a computer simulation explains,

and what sort of understanding they yield. In this context, it comes naturally to ask whether computer simulations promote a new theory of scientific explanation. My answer is that this is not the case; I outline my reasons at the end of Chapter 4. Instead, I urge for a theory of scientific explanation that provides the right conceptual framework for computer simulations as explanatory devices. Since philosophical literature is replete with theories of scientific explanation, it is paramount to be able to narrow down the analysis to the best candidates. A suitable distinction is made in terms of *externalist* and *internalist explanatory* accounts. Whereas the former ‘tracks’ an objective relation between the events described by the explanans and the explanandum (e.g., causality), the latter sees that explanation can be accounted for in a purely epistemic way (e.g., nomic necessity, unification). Given my working conceptualization of computer simulations, a suitable theory of explanation must be internalist. In this vein, the list of possible theories of scientific explanation is now reduced to four: the Deductive-Nomological Model, Mathematical Explanation, Model Explanation, and the Unificationist Explanation. The second and third accounts actually include several theories of scientific explanation, all of which are discussed in turn. The aim of the fourth chapter, then, is also to show in what sense the first three accounts fail to provide a suitable conceptual framework for explanation in computer simulations. The rejections are based on different grounds, depending on what the theory has to offer and what computer simulations demand.

At this point all the players are in place for answering the fifth and last of my questions; namely, what is the relation between explanans and explanandum in virtue of which a simulated phenomenon is explained, and what is it about this relation that yields understanding of the simulated phenomenon. I argue that the most suitable conceptual framework for computer simulations stems from the unificationist theory of scientific explanation. In Chapter 5, then, I show how an explanation is carried out by deriving the description of a simulated phenomenon. Taking into consideration the conditions under which a computer simulation is a reliable process, it follows then that by explaining a simulated phenomenon we are also explaining an empirical phenomenon. Such a result is in conformity with my conceptualization of computer simulations and with the unificationist viewpoint on scientific explanation. In addition, I show that by means of explaining a simulated phenomenon one also understands it. Understanding consists in seeing connections and common patterns that govern the simulated phenomenon and that can be unified with our corpus of scientific beliefs. In other words, the process of explaining shows which patterns of behavior constitute the simulated phenomena; understanding then comes by means of subsuming the phenomena under the greater unified theoretical framework that

is our corpus of scientific beliefs. We gain understanding because the simulated phenomena that initially appeared independent and unrelated are now unified with the totality of our scientific understanding. The more independent simulated phenomena explained, and thereby unified, the more comprehensible and transparent the world will appear to us. The epistemic power of computer simulations is then defended by showing how they explain and thereby yield genuine understanding of the world.

Finally, in Chapter 6 I outline some conclusions and discuss known shortcomings of my approach. I also use this last chapter for motivating further discussions in the young philosophy of computer simulations.

# Summary (German)

Computersimulationen haben sich als grundlegend für den Fortschritt und die Entwicklung der wissenschaftlichen Praxis erwiesen. Vor allem in Bereichen des wissenschaftlichen Experimentierens zeichnen sich Computersimulationen durch eine starke Präsenz aus. Dadurch wird das Interesse von Wissenschaftlern und Philosophen auf das epistemische Potential von Computersimulationen gelenkt. Natürlich findet man Befürworter und Kritiker dieser Technik, und auf beiden Seiten starke Argumente. Diese Arbeit ist daher mein Versuch, das epistemische Potential von Computersimulationen systematisch zu begründen, nicht nur um es gegen Kritiker zu verteidigen, sondern auch um damit eine fundierte Bewertung entsprechender Positionen vorzulegen.

Ausgehend von dieser Motivation, ist es das allgemeine Ziel dieser Arbeit zu zeigen, dass Computersimulationen zum Verstehen einiger Aspekte der Welt beitragen, indem sie eine Erklärung simulierter Phänomene liefern. Da das Verstehen von etwas ein epistemisches Konzept *par excellence* ist und da eine wissenschaftliche Erklärung als Verstehens-generierender Prozess begriffen wird, folgt daraus, dass das epistemische Potential von Computersimulationen dadurch begründet werden kann, dass man zeigt, warum Computersimulationen Erklärungen liefern können und in welcher Hinsicht sie das Verstehen befördern. In diesem Zusammenhang habe ich die ‘Unificationist-Theory’ als begrifflichen Rahmen gewählt, um daraus eine Definition bzw. Bedingungen für computersimulationsbasierte Erklärungen zu gewinnen. Soviel zu den allgemeinen Ausgangspunkten der Argumentation. Daraus ergeben sich nun genauer die folgenden Fragen:

1. Soviel zu den allgemeinen Ausgangspunkten der Argumentation. Daraus ergeben sich nun genauer die folgenden Fragen:
2. Was hat Computersoftware dem Laborexperiment voraus, das sich auch zur Begründung des epistemischen Potentials von Computersimulationen heranziehen lässt?
3. Was ist das Wesen einer Computersimulation? Ist sie abstrakter Natur auf-

grund der mathematischen/logischen Art der Algorithmen? Oder ist sie kausaler Natur aufgrund der Art des Computerprogramms?

4. Unter welchen Bedingungen kann eine Computersimulation als Prozess gelten, der zuverlässig Aspekte der Realität repräsentiert?
5. Wie vollzieht sich der Vorgang der Erklärung und in welchem Sinne trägt eine Erklärung zum Verständnis des simulierten Phänomens wie auch des realen Phänomens bei?

Diese Fragen deuten die Struktur der Arbeit an. Die erste Frage ist Gegenstand von Kapitel 1, in dem ich die aktuelle philosophische Literatur zu Modellen und wissenschaftlichen Experimenten analysiere und dabei einen besonderen Schwerpunkt auf methodische und erkenntnistheoretische Unterschiedsmerkmale von Computersimulationen lege. In diesem Kapitel geht es insbesondere auch um die Unterschiede zwischen Computersimulationen und Laborexperimenten. Außerdem soll in diesem Kapitel die Notwendigkeit eines anderen epistemologischen Zugangs zu Computersimulationen aufgezeigt werden. Konkret fordere ich, Computersimulationen für sich, das heißt nicht nur im Vergleich zu Laborexperimenten zu betrachten. Der letzte Abschnitt umreißt einige Positionen in Bezug auf die Neuheit von Computersimulationen als Gegenstand philosophischer Diskussionen.

Ausgehend von diesen Ergebnissen schlage ich vor, einen Arbeitsbegriff von Computersimulation zu bestimmen. Die entsprechenden Überlegungen finden sich in Kapitel 2 und werden in Kapitel 3 fortgesetzt. Ich beginne in diesem Zusammenhang zunächst mit den Ansätzen zur Natur der Computersoftware, wie sie von Philosophen der Informatik aufgefasst wird. Dabei stehen drei Aspekte im Vordergrund, nämlich die *specification*, der *algorithm*, und der *computer process*. Die Analyse dieser Aspekte soll der Beantwortung der zweiten Frage dienen, insofern nämlich gezeigt wird, dass die Vorteile von Computersoftware (z.B. die *syntax manipulability* und die *syntax transference*) bei Laborexperimenten nicht gegeben sind, was für die epistemologische Bedeutung der computerbasierten Simulationen spricht. Außerdem trägt das Verständnis von Computersoftware dazu bei, einen Begriff von Computersimulationen zu gewinnen. Für die Beantwortung der oben genannten Fragen ist es dementsprechend wichtig, das Verhältnis der analysierten Begriffe und Aspekte zueinander nicht aus dem Auge zu verlieren.

Die dritte Frage ist komplexer, da sie der Kernpunkt der Natur der Computersimulationen ist und in gewisser Weise ihr Schicksal als epistemische Mittel bestimmt (Kapitel 3). Aus diesen Gründen wird beabsichtigt hier auf verschiedenen Ebenen zu arbeiten: erstens wird die Literatur über *Simulationen* überprüft, wobei die Unter-

schiede zwischen *analogen* und *digitalen* Simulationen untersucht werden. Zweitens wird die Welt aller Computersimulationen, die in der wissenschaftlichen Praxis zu finden sind, eingeschränkt, wobei sie einerseits in auf Gleichungen basierende Simulationen und andererseits in cellular automata, agent-based und komplexe Systeme unterteilt werden. Dieser Schritt soll ausschließen sowie den Interessensbereich der Computersimulationen dieser wissenschaftlichen Untersuchung eingrenzen. Drittens wird die Methodik von Computersimulationen erörtert und ein konkreter paradigmatischer Fall der Klasse der auf Gleichungen basierenden Simulationen analysiert. Schließlich wird eine Arbeitskonzeptualisierung konstruiert, die eine Vorstellung von Computersimulationen etabliert, die für die Wissenschaften und die Philosophie sinnvoll ist.

Bisher waren philosophische Bemühungen darauf konzentriert eine Konzeptualisierung von Computersimulationen zu erläutern und zu konstruieren. Da dies abgeschlossen ist, bin ich nun in der Position die erklärende Bedeutung von Computersimulationen anzusprechen (Kapitel 4 folgende). Es ist erwähnenswert, dass sich der Fokus nun von einer Philosophie der Computerwissenschaft und einer Philosophie der Experimentierung in Richtung einer allgemeinen Philosophie der Wissenschaft, Philosophie der Mathematik und Epistemologie verlagert.

Eines der Ergebnisse, die in Kapitel 3 gewonnen wurden, ist dass Computersimulationen eine „eigene Welt“ sind, d.h. Systeme, die nicht unbedingt ein empirisches System darstellen. Dies ist ein nützliches und erwünschtes Merkmal von Computersimulationen, da Wissenschaftler daran interessiert sind hypothetische Systeme zu simulieren oder die Heuristik ihrer Modelle unabhängig von empirischer Darstellung zu erforschen. Jedoch deutet die Metaphysik der Erklärung darauf hin, dass die Erklärung (und das Verstehen) aus etwas „da draußen“, „in der Welt“ besteht. Die Frage ist dann, wie können wir zeigen, dass die Erklärung eines simulierten Phänomens auch auf das Phänomen der realen Welt anzuwenden ist und daher ein Verständnis davon erzeugt was „da draußen“ ist. In anderen Worten, wenn die epistemische Macht der Computersimulationen dadurch übertragen wird, dass sie aufzeigen wie sie das Verständnis eines simulierten Phänomens erklären und schaffen, und wenn die Metaphysik der Erklärung nach Erklärung und Verstehen dessen was „da draußen“ ist, verlangt, dann ist es vorrangig Bedingungen zu finden unter welchen eine Computersimulation ein verlässlicher Prozess ist, der wirklich das Zielsystem darstellt. Dabei macht das Erklären eines simulierten Phänomens das Erklären des Phänomens der realen Welt erforderlich, genauso wie das Verstehen des ersteren das Verstehen des letzteren erforderlich macht.

Wie bereits erwähnt, beinhaltet dieses Problem Theorien der Darstellung (was

ich durch die Arbeitsdefinition *goodness of representation* indirekt geklärt habe. Abschnitt 1.2.2.1) und mathematische Abhängigkeit von dem Rechnungsprozess (was ich in Kapitel 3 und durch die Vorstellungen von *verification* und *validation* erörtert habe). Der einfachste Weg die benötigte Darstellung zu erhalten, ist die Computersimulation innerhalb der Vorstellung eines verlässlichen Prozesses (*reliable process*) einzugrenzen, was Lösungen des Problems der Darstellung und Berechnung, wie zuvor erwähnt, beinhaltet. Eine Computersimulation ist dann verlässlich, wenn das simulierte Modell eine gute Darstellung eines Zielsystems ist und der Rechnungsprozess keine mathematischen Artefakte in die Produktion des simulierten Phänomens einführt. Nur unter diesen Bedingungen ist man berechtigt zu behaupten, dass das simulierte Phänomen wirklich das Phänomen der realen Welt darstellt.

Gleichermaßen wichtig ist zu zeigen, auf welcher Basis eine Computersimulation erklärt und welche Art des Verstehens wir erhalten. In diesem Zusammenhang ist es nur natürlich zu fragen, ob Computersimulationen eine neue Theorie der wissenschaftlichen Erklärung fördern. Meine Antwort darauf ist, dass dies nicht der Fall ist; Ich werde meine Gründe am Ende von Kapitel 4 umreißen. Stattdessen dränge ich nach einer Theorie der wissenschaftlichen Erklärung, die den richtigen konzeptionellen Rahmen für Computersimulationen als erklärende Mittel liefert. Da die philosophische Literatur mit Theorien der wissenschaftlichen Erklärung übersättigt ist, ist es von höchster Bedeutung die Analyse auf die besten Kandidaten eingrenzen zu können. Eine angemessene Unterscheidung wird bezüglich äußeren und inneren Erklärungsberichten gemacht (*externalist* and *internalist explanatory accounts*). Wohingegen ersteres eine objektive Beziehung zwischen den Ereignissen, die durch *explanans* und *explanandum* beschrieben werden, verfolgt (z.B. Kausalität), sieht letzteres, dass die Erklärung in einer rein epistemischen Art und Weise (z.B. Notwendigkeit, Vereinigung) geschildert werden kann. Meine Arbeitskonzeptualisierung von Computersimulationen vorausgesetzt, muss eine geeignete Theorie der Erklärung verinnerlicht sein. So wird nun die Liste der möglichen Theorien der wissenschaftlichen Erklärung auf vier reduziert: das „Deductive-Nomological Model“, Mathematische Erklärung, Modell Erklärung und „Unificationist Explanation“. Bericht zwei und drei beinhalten im Grunde verschiedene Theorien der wissenschaftlichen Erklärung, von denen alle der Reihe nach erörtert werden. Das Ziel des vierten Kapitels ist es dann auch zu zeigen, in welcher Art und Weise die ersten drei Berichte scheitern einen angemessenen konzeptionellen Rahmen für die Erklärung in Computersimulationen zu schaffen. Die Ablehnungen basieren auf verschiedenen Gründen, abhängig davon, was die Theorie zu bieten hat und was

Computersimulationen erfordern.

An dieser Stelle stehen alle Spieler in Position, um die fünfte und letzte meiner Fragen zu beantworten; nämlich, was ist die Beziehung zwischen „explanans“ und „explanandum“ aufgrund dessen ein simuliertes Phänomen erklärt wird, und was genau an dieser Beziehung erzeugt das Verstehen dieses simulierten Phänomens. Ich argumentiere, dass der geeignete konzeptionelle Rahmen für Computersimulationen von der „Unificationist theory“ der wissenschaftlichen Erklärung stammt. In Kapitel 5 zeige ich dann wie eine Erklärung durch Ableiten der Beschreibung eines simulierten Phänomens ausgeführt wird. Bedenkt man die Bedingungen unter welchen eine Computersimulation ein verlässliches System darstellt, folgt dann, dass das Erklären eines simulierten Phänomens das Erklären des Phänomens der realen Welt mit sich führt. Solch ein Ergebnis stimmt überein mit meiner Konzeptualisierung von Computersimulationen und mit der „unificationist“ Sichtweise auf die wissenschaftliche Erklärung. Zudem zeige ich, dass man durch das Erklären eines simulierten Phänomens dieses Phänomen auch versteht. Verstehen besteht darin Verbindungen und gemeinsame Strukturen zu sehen, die simulierte Phänomene beherrschen, und das kann vereint werden mit unserem Großteil der wissenschaftlichen Überzeugungen. In anderen Worten, der Prozess des Erklärens zeigt welche Verhaltensmuster hinter den simulierten Phänomenen stecken; Das Verstehen kommt dann durch das Einordnen des Phänomens in den größeren vereinten theoretischen Rahmen, der unser Großteil der wissenschaftlichen Überzeugungen ist. Wir erreichen das Verstehen weil die simulierten Phänomene, die anfänglich unabhängig und ungebunden erschienen, jetzt vereint werden mit der Gesamtheit unseres wissenschaftlichen Verständnisses. Je unabhängiger simulierte Phänomene erklärt haben, und somit verbunden haben, desto verständlicher und transparenter wird die Welt uns erscheinen. Die epistemische Macht der Computersimulation wird dann gestützt, indem gezeigt wird wie sie echtes Verstehen der Welt erklären und gewinnen.

Im letzten Kapitel 6 umreißt ich einige Schlussfolgerungen und erörtere bekannte Schwächen meines Konzepts. Ich verwende dieses letzte Kapitel zudem, um weitere Diskussionen in der noch jungen Philosophie der Computersimulationen anzuregen.

# Lexicon

ACCURACY: in measurement theory, accuracy refers to the set of measurements which provide an estimated value close to the true value of the quantity being measured.<sup>1</sup> The concept of accuracy is typically related to theoretical models about a target system. For instance, the measurement of the speed of light  $c = (3.0 \pm 0.1) \times 10^{10} \text{ cm/s}$  is accurate for it gets closer to the true value of the speed of light, namely,  $299792458 \text{ m/s}$ .<sup>2</sup> However, the precision of this measurement is low compared to other measurements. See the concept of precision below.<sup>3</sup>

BAYESIANISM: One important application of Bayes' theorem is used for the confirmation of theories. In Bayesian Confirmation Theory, it is said that evidence confirms (or would confirm) hypothesis  $H$  (to at least some degree) just in case the prior probability of  $H$  conditional on  $E$  is greater than the prior unconditional probability of  $H$  :  $P_i(\frac{H}{E}) > P_i(H)$ .  $E$  disconfirms (or would disconfirm)  $H$  if the prior probability of  $H$  conditional on  $E$  is less than the prior unconditional probability of  $H$ .<sup>4</sup>

BOOTSTRAPPING: it refers to the development of successively more complex and faster programming environments. For instance, an assembler program would be a simple machine program whereas an Object Oriented Programming environment such as Objective-C is considered more complex. This term also applies to compilers, development of specifications for computer software and hardware, etc.<sup>5</sup>

COMPLETENESS: A computer software is *complete* if it implements all the requirements described in the specification.<sup>6</sup> For doing so, the specification must contemplate every aspect of the computer software, such as the programming language, architecture of the digital computer, etc.

CORRECTNESS: A specification of a system is *correct* if every requirement included in its design satisfies the requirements of that system.<sup>7</sup>

DDI ACCOUNT: it stands for *Denotation*, *Demonstration*, and *Interpretation*. To represent a target system consists in *denotating* the target system by standing for or referring to it. *Demonstration* consists in exploiting the internal dynamics of the model; if the model is a mathematical model, for instance, one can demonstrate conclusions, operate over the variables and the like. *Interpretation* is the final stage of representing; it consists in ‘transferring’ the conclusions derived in the model back to the target system in order to make predictions, evaluations and so forth.<sup>8</sup>

HOMOMORPHISM: this theory suggests most of the requirements imposed by *representational isomorphism* should be relaxed by admitting mappings that are neither surjective nor injective. In this way *homomorphism* partially amends the problem of logical properties of *representational isomorphism*, namely, it is not symmetrical nor transitive, although it is still reflexive.<sup>9</sup>

NEYMAN-PEARSON: In statistics, the Neyman–Pearson lemma measures the likelihood-ratio between two competing hypothesis.<sup>10</sup>

ODE: Acronym for ‘Ordinary Differential Equation.’ In mathematics, an ODE is a continuous equation that contains one unknown variable and its partial derivatives. ODE are widely used in science for representing systems with one variables involved, such as the terrestrial and celestial mechanics (trajectory of a projective or a planet), reaction rates, etc.

PARTIAL ISOMORPHISM: this theory of representation amends some of the objections that *representational isomorphism* received. This theory replaces the isomorphic relations between theory and model for partial relations. According to Frigg, partial isomorphism does not get around the problems of isomorphism for, at their bases, they both have the wrong logical properties. To Frigg’s mind, representation is not reflexive, symmetric, nor transitive.<sup>11</sup>

PDE: Acronym for ‘Partial Differential Equation.’ In mathematics, a PDE is a continuous equation that contains several unknown variables and their partial derivatives. PDE are widely used in science for representing systems with several variables involved, such as sound, heat, fluid flow, etc.

PRECISION: in measurement theory, precision refers to the set of measurements for

which the uncertainty of the estimated results is relatively small.<sup>12</sup> The concept of precision is typically ascribed to scientific instruments when they measure a given quantity. In order to increase precision, there are a number of statistical methods, such as *the standard error of measurement*, or *the standard deviation*. For instance, the measurement of the speed of light  $c = (2.000000000 \pm 0.000000001) \times 10^{10} \text{ cm/s}$  is precise for the uncertainty of the estimated value is of the order of 0.000000001.<sup>13</sup> Precision does not imply accuracy.<sup>14</sup>

REPRESENTATIONAL ISOMORPHISM: in representation theory, two models of a theory are isomorphic when they exhibit the same structure in terms of their basic concepts and theory.<sup>15</sup> Theories of isomorphic representation dedicate most of their effort to construct and define the isomorphic relation by means of set theory. Defined in this way, isomorphism provides equivalence relations among algebras for they are reflexive, symmetric and transitive. This has been the source of many objections, see for instance (Frigg, 2002).

ROBUSTNESS: A specification is *robust* if it describes and plans every exceptional condition, such as incorrect inputs, the incorrect value of a variable, overflow and the like.<sup>16</sup>

SIMILARITY: a model *resembles* its target system if there is a significant similarity between their visual appearance. Theories of similarity do not assert resemblance as a necessary and sufficient condition for representation. In fact it is a weak condition, which neither requires nor includes similarities in visual appearance. Suarez explains it in the following way: “A and B are similar if and only if they share a subset of their properties. In accordance with this identity-based theory similarity is reflexive (A is maximally similar to itself), and symmetric (if A is similar to B, on account of sharing properties  $p_1, p_2, \dots, p_n$ , then B is similar to A on the same grounds); but non-transitive (A may share  $p_1$  with B, and B may share  $p_2$  with C, without A and C sharing any property—other than the property of sharing a property with B!)” (Suárez, 2003, 228).

SPECIFICATION LANGUAGES: specification languages typically use mathematical notation to describe in a precise way the properties which a computer simulation must have. They describe what the simulation must do without saying how it is to be done. This abstraction makes specifications useful in the process of developing a computer system, because they allow questions about what the system does to

be answered confidently, without the need to disentangle the information from a mass of detailed program code, or to speculate about the meaning of phrases in an imprecisely-worded prose description.<sup>17</sup> Examples of specifications languages are the *Z notation*, *VDM*, or *LePUS3*.

STABILITY: The stability of a specification reflects the chances of it changing in the future. Some of the ‘core’ aspects of the simulation must be stable over time and not require changes.<sup>18</sup>

STRUCTURAL REPRESENTATION: this representational theory takes the notion of isomorphism and extends its implications. Mappings are now features from the representational system: “they yield an isomorphic copy of one *IRS* in another” (Swoyer, 1991, 456). *IRS* is the abbreviation for *intensional relational systems*: “[*IRS*] differ from their extensional kin in containing properties and relations that are not constructed out of sets of objects, possible worlds, or anything else at all” (Swoyer, 1991, 455).

## Notes

<sup>1</sup>Cf. (Morrison, 2009, 49).

<sup>2</sup>Cf. (Franklin, 1981, 367. Note 1).

<sup>3</sup>For more on this, see (Evenson et al., 1972; Krantz et al., 1971; Pfanzag, 1959; Scott and Suppes, 1958; Suppes and Zinnes, 1962; Trout, 2001).

<sup>4</sup>For more on this, see (Talbot, Summer 2011 Edition).

<sup>5</sup>See, for instance (Chernick, 2007).

<sup>6</sup>See, for instance (Jalote, 2008).

<sup>7</sup>See, for instance (Jalote, 2008).

<sup>8</sup>See (Hughes, 1997).

<sup>9</sup>See, for instance (Frigg, 2002).

<sup>10</sup>See (Neyman and Pearson, 1933).

<sup>11</sup>See (Frigg, 2003).

<sup>12</sup>Cf. (Morrison, 2009, 49).

<sup>13</sup>Cf. (Franklin, 1981, 367. Note 1).

<sup>14</sup>For more on this point, see (Evenson et al., 1972; Krantz et al., 1971; Pfanzag, 1959; Scott and Suppes, 1958; Suppes and Zinnes, 1962; Trout, 2001).

<sup>15</sup>See, for instance (Suppes, 2002, 54).

<sup>16</sup>See, for instance (Jalote, 2008).

<sup>17</sup>See, for instance (Spivey, 2001).

<sup>18</sup>See, for instance (Jalote, 2008).

# Bibliography

- Achinstein, P. (1983). *The Nature of Explanation*. Oxford University Press.
- (2001). *The Book of Evidence*. Oxford University Press.
- Ackermann, R. (1989). The New Experimentalism. *The British Journal for the Philosophy of Science*, 40(2):185–190.
- Ainsworth, M., Riddle, S. et al. (1996). Formal Validation of Viewpoint Specifications. *Software Engineering Journal*, 11(1):58–66.
- Al-Arian, S. (1992). HPTR: Hardware Partition in Time Redundancy Technique for Fault Tolerance. *IEEE Southeastcon '92, Proceedings.*, 2:630–633.
- Arbab, F. and Sirjani, M. (eds.) (2009). *Fundamentals of Software Engineering*, vol. 5961 of *Lecture Notes in Computer Science*. Springer.
- Aristotle (1936). *Physics*. Oxford University Press.
- (1965). *Historia Animalium*. Harvard University Press.
- Arkoudas, K. and Bringsjord, S. (2007). Computers, Justification, and Mathematical Knowledge. *Minds and Machines*, 17(2):185–202.
- Aronson, J., Harré, R. et al. (1993). *Realism Rescued*. Duckworth.
- Ashenurst, R. L. (1989). Editorial Process Verification. *Communication of the ACM*, 32(3):287–293.
- ASME (2006). Guide for Verification and Validation in Computational Solid Mechanics. Tech. rep., The American Society of Mechanical Engineers, ASME Standard V&V 10-2006.
- Aspray, W. (ed.) (1990a). *Computing before Computers*. Iowa State University Press.

- Aspray, W. (1990b). *John von Neumann and the Origins of Modern Computing*. MIT Press.
- Atkinson, F. V. (1964). *Discrete and Continuous Boundary Problems*. Academic Press.
- Atkinson, K. E., Han, W. et al. (2009). *Numerical Solution of Ordinary Differential Equations*. Wiley.
- Ayer, A. J. (1946). *Language, Truth and Logic*, vol. 138. Penguin Books.
- Babuska, I., Ciarlet, P. G. et al. (eds.) (2000). *Mathematical Modeling and Numerical Simulation in Continuum Mechanics*. Springer.
- Baker, A. (2009). Mathematical Explanation in Science. *The British Journal for the Philosophy of Science*, 60(3):611–633.
- Barnes, E. (1992). Explanatory Unification and Scientific Understanding. *Philosophy of Science Association*, 1:3–12.
- (1994). Explaining Brute Facts. *Proceedings of the Philosophy of Science Association*, 1:61–68.
- Batterman, R. W. (2002). *The Devil is in the Details. Asymptotic Reasoning in Explanation, Reduction, and Emergence*. Oxford University Press.
- Bedau, M. A. (1997). Weak Emergence. *Noûs*, 31(1997):375–399.
- Bedau, M. A. and Humphreys, P. W. (eds.) (2008). *Emergence Contemporary Readings in Philosophy and Science*. MIT Press.
- Berekovic, M., Simopoulos, N. et al. (eds.) (2008). *Embedded Computer Systems: Architectures, Modeling, and Simulation*. Springer.
- Białynicki-Birula, I. and Białynicka-Birula, I. (2004). *Modeling Reality How Computers Mirror Life*. Oxford University Press.
- Bidoit, M. and Mosses, P. D. (2004). *CASL User Manual: Introduction to Using the Common Algebraic Specification Language*. Springer.
- Bjørner, D. and Henson, M. C. (2007). *Logics of Specification Languages*. Springer.
- Bjørner, D. and Jones, C. B. (1978). *The Vienna Development Method: The Meta-Language*. Springer.

- Blass, A., Dershowitz, N. et al. (2009). When are Two Algorithms the Same? *The Bulletin of Symbolic Logic*, (250):145–168.
- Blass, A. and Gurevich, Y. (2003). Algorithms: A Quest for Absolute Definitions. Tech. Rep. 81, Bulletin of European Association for Theoretical Computer Science.
- Bokulich, A. (2008). *Re-examining the Quantum-Classical Relation: Beyond Reductionism and Pluralism*. Cambridge University Press.
- (2010). Distinguishing Explanatory from Non-explanatory Fictions. In *Draft of talk from PSA 2010 Montréal*, .
- (2011). How Scientific Models can Explain. *Synthese*, 180:33–45.
- Boulle, M. (2004). Khiops: A Statistical Discretization Method of Continuous Attributes. *Machine Learning*, 55:53–69.
- Bridgman, P. W. (1927). *The Logic of Modern Physics*. MacMillan.
- (1959). *The Way Things Are*. Harvard University Press.
- Brown, C. L. T. (2006). *Computer Evidence: Collection & Preservation*. Charles River Media.
- Brown, J. R. (1991). *The Laboratory of the Mind Thought Experiments in the Natural Sciences*. Routledge.
- (1994). *Smoke and Mirrors. How Science Reflects Reality*. Routledge.
- Bueno, O. and Colyvan, M. (2006). An Inferential Conception of the Application of Mathematics. *Noûs*, 1–25.
- Butcher, J. C. (2008). *Numerical Methods for Ordinary Differential Equations*. Wiley & Sons.
- Butterfield, A. (ed.) (2008). *Unifying Theories of Programming*. Springer.
- Campbell, N. R. (1920). *Physics: The Elements*. Cambridge University Press.
- Cantwell Smith, B. (1985). The Limits of Correctness. *ACM SIGCAS Computers and Society*, 14(1):18–26.
- Carnap, R. (1932). The Elimination of Metaphysics Through Logical Analysis of Language. *Erkenntnis*, 60–81.

- (1937). *Logical Syntax of Language*. Rout.
- Cartwright, N. (1983). *How the Laws of Physics Lie*. Oxford University Press.
- (1994). The Metaphysics of the Disunified World. In *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, vol. 1994, 357–364.
- Cartwright, N., Shomar, T. et al. (1995). The Tool Box of Science. In W. Herfel, W. Krajewski, I. Niiniluoto and R. Wojcicki (eds.), *Theories and Models in Scientific Processes: Proceedings of AFOS '94*, 137–149. Rodopi.
- Ceruzzi, P. E. (1998). *A History of Modern Computing*. MIT Press.
- Chabert, J.-C. (ed.) (1994). *A History of Algorithms. From the Pebble to the Microchip*. Springer.
- Chernick, M. R. (2007). *Bootstrap Methods: A Guide for Practitioners and Researchers*. Wiley-Interscience.
- Church, A. (1936). A Note on the Entscheidungsproblem. *Journal of Symbolic Logic*, 1(1):40–41.
- Churchland, P. M. and Hooker, C. A. (eds.) (1985). *Images of Science. Essays on Realism and Empiricism, with a reply from Bas van Fraassen*. University of Chicago Press.
- Ciletti, M. D. (2010). *Advanced Digital Design with Verilog HDL*. Prentice Hall.
- Cipra, B. (1995). How Number Theory Got the Best of the Pentium Chip. *Science*, 175.
- Cirpka, O. A., Bürger, C. M. et al. (2004). Uncertainty and Data worth Analysis for the Hydraulic Design of Funnel-and-Gate Systems in Heterogeneous Aquifers. *Water Resources Research*, 40(11):W11502.
- Coffa, J. A. (1991). *The Semantic Tradition from Kant to Carnap: To the Vienna Station*. Cambridge University Press.
- Cohn, A. (1989). The Notion of Proof in Hardware Verification. *Journal of Automated Reasoning*, 5(2):127–139.
- Colburn, T. and Shute, G. (2007). Abstraction in Computer Science. *Minds and Machines*, 17(2):169–184.

- Craver, C. F. (2006). When Mechanistic Models Explain. *Synthese*, 153:355–376.
- Da Costa, N. C. A. and French, S. (2000). Models, Theories, and Structures: Thirty Years On. *Philosophy of Science*, 67(3):116–127.
- Davisson, C. and Germer, H. (1927). Diffraction of Electrons by a Crystal of Nickel. *Physical Review*, 30:705–740.
- De Regt, H. W. and Dieks, D. (2005). A Contextual Approach to Scientific Understanding. *Synthese*, 144(1):137–170.
- Descartes, R. (1996 (1641)). *Descartes: Meditations on First Philosophy*. Cambridge University Press.
- Dicke, R., Peebles, P. J. E. et al. (1965). Cosmic Black-Body Radiation. *Astrophysical Journal Letters*, 142:414–419.
- Díez, J. A. (1998). Hacia una Teoría General de la Representación Científica. *Theoria*, 13(1):113–139.
- Dijkstra, E. W. (1974). Programming as a Discipline of Mathematical Nature. *American Mathematical Monthly*, 81(6):608–612.
- Dowe, P. (2000). *Physical Causation*. Cambridge University Press.
- Dray, W. (1964). *Laws and Explanation in History*. Oxford University Press.
- Duhem, P. M. (1954). *The Aim and Structure of Physical Theory*. Princeton University Press.
- Dupré, J. (1993). *The Disorder of Things. Metaphysical Foundations of the Disunity of Science*. Harvard University Press.
- Durán, J. M. (2013a). A Brief Overview of the Philosophical Study of Computer Simulations. *APA Newsletter on Philosophy and Computers*, 13(1).
- (2013b). The Use of the ‘Materiality Argument’ in the Literature on Computer Simulations. In J. M. Durán and E. Arnold (eds.), *Computer Simulations and the Changing Face of Scientific Experimentation*, Cambridge Scholars Publishing.
- (Unpublished). Computer Simulations as a Technological Singularity in the Empirical Sciences.

- Eden, A. H. (2007). Three Paradigms of Computer Science. *Minds and Machines*, 17(2):135–167.
- Eden, A. H. and Turner, R. (2005). Towards an Ontology of Software Design: The Intention/Locality Hypothesis. In *3rd European Conference Computing and Philosophy, Västerås, Sweden*, 1–4.
- (2007). Problems in the Ontology of Computer Programs. *Applied Ontology*, 2(1):1–24.
- Eigenmann, R. and Lilja, D. J. (1998). von Neumann Computers. Access June, 2011.  
**URL:** [http://www.idemployee.id.tue.nl/g.w.m.rauterberg/presentations/von-neumann-computer\[2\].pdf](http://www.idemployee.id.tue.nl/g.w.m.rauterberg/presentations/von-neumann-computer[2].pdf)
- Elgin, C. (1996). *Considered Judgement*. Princeton University Press.
- (2004). True Enough. *Philosophical Issues*, 14:113–131.
- (2007). Understanding and the Facts. *Philosophical Studies*, 132(1):33–42.
- (2009). Is Understanding Factive? In A. Haddock, A. Millar and D. H. Pritchard (eds.), *Epistemic Value*, 322–330. Oxford University Press.
- Elgin, M. and Sober, E. (2002). Cartwright on Explanation and Idealizations. *Erkenntnis*, 57(3):441–450.
- Evenson, K. M., Petersen, F. R. et al. (1972). Speed of Light from Direct Frequency and Wavelength Measurements of the Methane-Stabilizer Laser. *Physical Review Letters*, 22(19):1346–1349.
- Fahrbach, L. (2005). Understanding Brute Facts. *Synthese*, 145(3):449–466.
- Feigl, H. (1970). *Minnesota Studies in the Philosophy of Science. Volume IV*, chap. The ‘Orthodox’ View of Theories: Remarks in Defense as well as Critique. University of Minnesota Press.
- Fetzer, J. H. (1988). Program Verification: The Very Idea. *Communications of the ACM*, 37(9):1048–1063.
- (1991). Philosophical Aspects of Program Verification. *Minds and Machines*, 1(2):197–216.

- (1997). Thinking and Computing: Computers as Special Kinds of Signs. *Minds and Machines*, 7(3):345–364.
- Feyerabend, P. (1975). *Against Method*. New Left Books.
- Feynman, R. P. (2001). *What do You Care what Other People Think?* W. W. Norton & Company.
- Field, R. J. and Noyes, R. M. (1974). Oscillations in Chemical Systems IV: Limit Cycle Behavior in a Model of a Chemical Reaction. *Journal of Chemical Physics*, 60:1877–1884.
- Fischetti, M. (2011). Computers versus Brains. *Scientific American*.
- Floyd, R. W. (1967). Assigning Meanings to Programs. *Mathematical Aspects of Computer Science*, 19:19–32.
- Franklin, A. (1981). What Makes a 'Good' Experiment? *The British Journal for the Philosophy of Science*, 32(4):367.
- (1986). *The Neglect of Experiment*. Cambridge University Press.
- (1989). The Epistemology of Experiment. In D. Gooding, T. Pinch and S. Schaffer (eds.), *The Uses of Experiments. Studies in the Natural Sciences*, 437–460. Cambridge University Press.
- (1990). *Experiment, Right or Wrong*. Cambridge University Press.
- Franklin, A. and Howson, C. (1984). Why do Scientists Prefer to Vary Their Experiments? *Studies in History and Philosophy of Science*, 15(1):51 – 62.
- Friedman, J. I. and Telegdi, V. L. (1957). Nuclear Emulsion Evidence for Parity Nonconservation in the Decay Chain  $\pi^+ - \mu^+ - e^+$ . *Physical Review*, 105(5):1681–1682.
- Friedman, M. (1974). Explanation and Scientific Understanding. *The Journal of Philosophy*, 71(1):5–19.
- Frigg, R. (2002). Models and Representation: Why Structures are Not Enough. *Measurement in Physics and Economics Discussion Paper Series*, 25/02:1–42.
- (2003). *Re-presenting Scientific Representation*. Phd, London School of Economics.

- (2009a). Models and Fiction. *Synthese*, 172(2):251–268.
- (2009b). Scientific Representation and the Semantic View of Theories. *Theoria. Revista de Teoría, Historia y Fundamentos de la Ciencia*, 21(1):49–65.
- Frigg, R. and Hartmann, S. (2006). Scientific Models. In S. Sarkar and J. Pfeifer (eds.), *The Philosophy of Science. An Encyclopedia*, 740–749. Routledge.
- Frigg, R. and Reiss, J. (2009). The Philosophy of Simulation: Hot New Issues or Same Old Stew? *Synthese*, 169(3):593–613. doi:10.1007/s11229-009-9577-x.
- Galison, P. (1996). Computer Simulation and the Trading Zone. In P. Galison and D. Stump (eds.), *Disunity of Science: Boundaries, Contexts, and Power*, Stanford University Press.
- (1997). *Image and Logic: A Material Culture of Microphysics*. University of Chicago Press.
- García, P. and Velasco, M. (2013). Exploratory Strategies: Experiments and Simulations. In J. M. Durán and E. Arnold (eds.), *Computer Simulations and the Changing Face of Scientific Experimentation*, Cambridge Scholars Publishing.
- Gettier, E. (1963). Is Justified True Belief Knowledge? *Analysis*, 23:121–123.  
**URL:** <http://www.ditext.com/gettier/gettier.html>
- Giere, R. N. (1988). *Explaining Science: A Cognitive Approach*. University of Chicago Press.
- (2004). How Models are Used to Represent Reality. *Philosophy of Science*, 71(December):742–752.
- (2009). An Agent-Based Conception of Models and Scientific Representation. *Synthese*, 172(2):269–281.
- Gillies, D. (2000). *Philosophical Theories of Probability*. Routledge.
- Girle, R. (2003). *Possible Worlds*. McGill-Queen’s University Press.
- Goldman, A. I. (1979). What is Justified Belief? In G. Pappas (ed.), *Justification and Knowledge*, 1–23. Reidel Publishers Company.
- (1988). *Epistemology and Cognition*. Harvard University Press.
- (1999). *Knowledge in a Social Worlds*. Oxford University Press.

- (2009). Reliabilism and the Value of Knowledge. In A. Haddock, A. Millar and D. Pritchard (eds.), *Epistemic Value*, 19–41. Oxford University Press.
- Gonzalez, W. J. (ed.) (2012). *Scientific Realism and Democratic Society: The Philosophy of Philip Kitcher*. Rodopi.
- Goodman, N. (1968). *Languages of Art. An Approach to a Theory of Symbols*. The Bobbs-Merrill Company.
- Gosper, R. W. (1984). Exploiting Regularities in Large Cellular Spaces. *Physica 10D*, 75–80.
- Gould, H., Tobochnik, J. et al. (2007). *An Introduction to Computer Simulation Methods. Applications to Physical Systems*. Pearson Addison Wesley.
- Grimm, S. R. (2010). The Goal of Explanation. *Studies in History and Philosophy of Science*, 41:337–344.
- Grimmett, G. R. and Stirzaker, D. R. (1992). *Probability and Random Processes*. Clarendon Press.
- Gruska, J. (1997). *Foundations of Computing*. International Thomson Computer Press.
- Guala, F. (2002). Models, Simulations, and Experiments. In L. Magnani and N. J. Nersessian (eds.), *Model-Based Reasoning: Science, Technology, Values*, 59–74. Kluwer.
- Guttag, J., Horowitz, E. et al. (1978). The Design of Data Type Specifications. In R. Yeh (ed.), *Current Trends in Programming Methodology, Vol. IV: Data Structuring*, 60-79. Prentice Hall.
- Hacking, I. (1983). *Representing and Intervening. Introductory Topics in the Philosophy of Natural Science*. Cambridge University Press.
- (1988). Philosophers of Experiment. In *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, 147–156.
- Hahn, U. (2011). The Problem of Circularity in Evidence, Argument, and Explanation. *Perspectives on Psychological Science*, 6(2):172–182.
- Hales, T. C. (2001). The Honeycomb Conjecture. *Discrete & Computational Geometry*, 25(1):1–22. doi:10.1007/s004540010071.  
**URL:** <http://dx.doi.org/10.1007/s004540010071>

- Hanson, N. R. (1958). *Patterns of Discovery: An Inquiry Into the Conceptual Foundations of Science*. Cambridge University Press.
- (1963). *The Concept of the Positron: A Philosophical Analysis*. Cambridge University Press.
- Harré, R. (1981). *Great Scientific Experiments*. Phaidon Press.
- Harris, T. (2003). Data Models and the Acquisition and Manipulation of Data. *Philosophy of Science*, 70:1508–1517.
- Hartmann, S. (1995). Models as a Tool for Theory Construction: Some Strategies of Preliminary Physics. In *Theories and Models in Scientific Processes: Proceedings of AFOS'94 Workshop, August 15-26, Mqdralin, and IUHPS'94 Conference, August 27-29, Warszawa*, vol. 24, 49. Rodopi.
- (1996). The World as a Process: Simulations in the Natural and Social Sciences. In R. Hegselmann, U. Mueller and K. G. Troitzsch (eds.), *Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View*, 77–100. Kluwer.
- (1999). Models and Stories in Hadron Physics. In M. S. Morgan and M. Morrison (eds.), *Models as Mediators: Perspectives on Natural and Social Science*, Cambridge University Press.
- Haugeland, J. (1981). Analog and Analog. *Philosophical Topics*, 12:213–222.
- Hayes, J. (1988). *Computer Architecture and Organization*. McGraw Hill.
- Hempel, C. and Oppenheim, P. (1945). A Definition of ‘Degree of Confirmation’. *Philosophy of Science*, 12(2):98–115.
- (1948). Studies in the Logic of Explanation. *Philosophy of Science*, 15(2):135–175.
- Hempel, C. G. (1935). La Teoría de la Verdad de los Positivistas Lógicos. *Analysis*, II(4):49–59.
- (1945). Studies in the Logic of Confirmation (I). *Mind*, 54(213):1–26.
- (1965). *Aspects of Scientific Explanation and other Essays in the Philosophy of Science*. The Free Press.

- (1966). Laws and Their Role in Scientific Explanation. In *Philosophy of Natural Science*, 47–69. Wiley-Blackwell.
- Hida, T. (1980). Causal Calculus of Brownian Functionals and its Applications. *Int. Syrup. on Stochastic and Related Topics, Carleton University, Ottawa*.
- Hillis, W. D. (1984). The Connection Machine: A Computer Architecture Based on Cellular Automata. *Physica D: Nonlinear Phenomena*, 213–228.
- Hoare, C. and Jones, C. (1989). *Essays in Computing Science*. Prentice Hall.
- Hoare, C. A. R. (1969). An Axiomatic Basis for Computer Programming. *Communications of the ACM*, 12(10):576–580.
- (1999). A Theory of Programming: Denotational, Algebraic and Operational Semantics. Access June 2011.  
**URL:** [http://research.microsoft.com/pubs/68627/a\\_theory\\_of\\_programming.pdf](http://research.microsoft.com/pubs/68627/a_theory_of_programming.pdf)
- (2004). Communicating Sequential Processes. Access June 2011.  
**URL:** <http://www.usingcsp.com/cspbook.pdf>
- Hoare, C. A. R. and Jifeng, H. (1998). *Unified Theories of Programming*. Prentice Hall.
- Holton, G. (1999). Focus: Centennial Focus: Millikan’s Measurement of Planck’s Constant. *Physical Review Focus*, 3(23).
- Hughes, R. I. G. (1997). Models and Representation. *Philosophy of Science*, 64:325–336.
- Humphreys, P. W. (1991). Computer Simulations. *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, 2:497–506.
- (1993). Greater Unification equals Greater Understanding? *Analysis*, 53(3):183–188.
- (1997). Emergence, Not Supervenience. *Philosophy of Science*, 64:337–345.
- (2004). *Extending Ourselves: Computational Science, Empiricism, and Scientific Method*. Oxford University Press.
- (2009). The Philosophical Novelty of Computer Simulation Methods. *Synthese*, 169(3):615–626.

- Jacky, J. (1996). *The Way of Z: Practical Programming with Formal Methods*. Cambridge University Press.
- Jalote, P. (2008). *A Concise Introduction to Software Engineering*. Springer.
- Janssen, M. (2002). COI Stories: Explanation and Evidence in the History of Science. *Perspectives on Science*, 10(4):457–522.
- Jason, G. (1989). The Role of Error in Computer Science. *Philosophia*, 19(4):403–416.
- Jones, M. R. and Cartwright, N. (eds.) (2005). *Idealization XII: Correcting the Model Idealization and Abstraction in the Sciences*. Rodopi.
- Kaiser, M. I. and Seide, A. (eds.) (2013). *Philip Kitcher. Pragmatic Naturalism*. Ontos Verlag.
- Kao, M.-Y. (ed.) (2008). *Encyclopedia of Algorithms*, vol. 2. Springer.  
**URL:** <http://www.mdpi.com/1999-4893/2/4/1301/>
- Keller, E. F. (2000). Models of and Models for: Theory and Practice in Contemporary Biology. *Philosophy of Science*, 67(3):72–86.
- (2003). Models, Simulation, and ‘Computer Experiments’. In H. Radder (ed.), *The Philosophy of Scientific Experimentation*, 198–215. University of Pittsburgh Press.
- Kim, J. (1988). What Is ‘Naturalized Epistemology?’. *Philosophical Perspectives*, 2:381–405.
- (1994). Explanatory Knowledge and Metaphysical Dependence. *Philosophical Issues*, 5:51–69.
- Kitcher, P. (1981). Explanatory Unification. *Philosophy of Science*, 48(4):507–531.
- (1986). Projecting the Order of Nature. In R. E. Butts (ed.), *Kant’s Philosophy of Physical Science*, vol. 33 of *The Western Ontario Series in Philosophy of Science*, 201–235. Springer.
- (1989). Explanatory Unification and the Causal Structure of the World. In P. Kitcher and W. C. Salmon (eds.), *Scientific Explanation*, 410–505. University of Minnesota Press.
- (1992). The Naturalists Return. *The Philosophical Review*, 101(1):53–114.

- (1993). *The Advancement of Science*. Oxford University Press.
- (1994). The Unity of Science and the Unity of Nature. In P. Parrini (ed.), *Kant and Contemporary Epistemology*, vol. 54 of *The University of Western Ontario Series in Philosophy of Science*, 253–272. Springer.
- (1999). Unification as a Regulative Ideal. *Perspectives on Science*, 7(3):337–348.
- (2001). Real Realism: The Galilean Strategy. *The Philosophical Review*, 110(2):151–197.
- (2013). Pragmatic Naturalism. In M. I. Kaiser and A. Seide (eds.), *Philip Kitcher: Pragmatic Naturalism*, 15–43. Ontos Verlag.
- Kitcher, P. and Salmon, W. C. (1987). van Fraassen on Explanation. *The Journal of Philosophy*, 84(6):315–330.
- Klein, G., Huuck, R. et al. (2009). Operating System Verification. *Journal of Automated Reasoning*, 42(2-4):123–124.
- Knuth, D. E. (1973). *The Art of Computer Programming*. Addison-Wesley.
- (1974). Computer Science and Its Relation to Mathematics. *The American Mathematical Monthly*, 81:323–343.
- Koren, I. and Krishna, C. M. (2007). *Fault-Tolerant Systems*. Elsevier.
- Kosso, P. (1989). *Observability and Observation in Philosophy of Science*. Kluwer Academic Publishers.
- (2010). And Yet It Moves: The Observability of the Rotation of the Earth. *Foundations of Science*, 15(3):213–225.
- Kotsiantis, S. and Kanellopoulos, D. (2006). Discretization Techniques: A Recent Survey. *GESTS International Transactions on Computer Science and Engineering*, 32(1):47–58.
- Kraftmakher, Y. (2007). *Experiments and Demonstrations in Physics. Bar-Ilan Physics Laboratory*. World Scientific Publishing.
- Krantz, D. H., Suppes, P. et al. (1971). *Foundations of Measurement - Volume I - Additive and polynomial representations*. Dover Publications.

- Krohs, U. (2008). How Digital Computer Simulations Explain Real-World Processes. *International Studies in the Philosophy of Science*, 22(3):277–292.
- Kuhn, T. S. (1962). *The Structure of Scientific Revolutions*. University of Chicago Press.
- Kvanvig, J. L. (2003). *The Value of Knowledge and the Pursuit of Understanding*. Cambridge University Press.
- (2009). The Value of Understanding. In A. Haddock, A. Millar and D. H. Pritchard (eds.), *Epistemic Value*, 95–111. Oxford University Press.
- Lakatos, I. (1974). The Role of Crucial Experiments in Science. *Studies in the History and Philosophy of Science*, 4(4):309–325.
- Lakatos, I., Worrall, J. et al. (1980). *The Methodology of Scientific Research Programmes. Philosophical Papers - Volume I*. Cambridge University Press.
- Lee, P. A. and Anderson, T. (1990). *Fault Tolerance, Principles and Practice*. Springer.
- Lenhard, J. (2006). Surprised by a Nanowire: Simulation, Control, and Understanding. *Philosophy of Science*, 73:605–616.
- Leplin, J. (1982). The Historical Objection to Scientific Realism. In *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, 88–97.
- Lesne, A. (Online). Discrete vs Continuous Controversy in Physics. Access October 2012.  
**URL:** [www.lptl.jussieu.fr/user/lesne/MSCS-lesne-v3.pdf](http://www.lptl.jussieu.fr/user/lesne/MSCS-lesne-v3.pdf)
- Lewis, D. (1971). Analog and Digital. *Nous*, 5:321–327.
- (1973). *Counterfactuals*. Oxford University Press.
- Lipton, P. (2004). *Inference to the Best Explanation*. Routledge.
- Liu, C. (2004a). Approximations, Idealizations, and Models in Statistical Mechanics. *Erkenntnis*, 60:235–263.
- (2004b). Laws and Models in a Theory of Idealization. *Synthese*, 138(3):363–385.

- Lloyd, E. (1988). The Semantic Approach and its Application to Evolutionary Theory. In *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, 278–285.
- Lotka, A. J. (1910). Contribution to the Theory of Periodic Reaction. *The Journal of Physical Chemistry*, 14(3):271–274.
- Machamer, P. (2004). Activities and Causation: The Metaphysics and Epistemology of Mechanisms. *International Studies in the Philosophy of Science*, 18(1):27–39.
- MacKenzie, D. (2006). *An Engine, not a Camera*. MIT Press.
- MacKenzie, D. A. (2001). *Mechanizing Proof: Computing, Risk, and Trust Inside Technology*. MIT Press.
- Mauil, N. (1976). Reconstructed Science as Philosophical Evidence. In *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, 119–129.
- Mayo, D. G. (1994). The New Experimentalism, Topical Hypotheses, and Learning from Error. In *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, 270–279.
- (2010). Learning from Error, Severe Testing, and the Growth of Theoretical Knowledge. In D. G. Mayo and A. Spanos (eds.), *Error and Inference. Recent Exchanges on Experimental Reasoning, Reliability, and the Objectivity and Rationality of Science*, University of Chicago Press.
- Mayo, D. G. and Spanos, A. (eds.) (2010). *Error and Inference. Recent Exchanges on Experimental Reasoning, Reliability, and the Objectivity and Rationality of Science*. Cambridge University Press.
- McMillan, K. (1992). *Symbolic Model Checking. An Approach to the State Explosion Problem*. Ph.D. thesis, Carnegie Mellon University - Computer science department.
- McMullin, E. (1968). What Do Physical Models Tell Us? In B. van Rootselaar and J. F. Staal (eds.), *Logic, Methodology and Science III*, 385–396. North Holland.
- (1978). Structural Explanation. *American Philosophical Quarterly*, 15(2):139–147.

- Mermet, J. (ed.) (1993). *Fundamentals and Standards in Hardware Description Languages*. Springer.
- Metropolis, N. (1987). The Beginning of the Monte Carlo Method. *Los Alamos Science*, 125–130.
- Metropolis, N. and Ulam, S. (1949). The Monte Carlo Method. *Journal of the American Statistical Association*, 44(247):335–341.
- Mihram, G. (1972). Some Practical Aspects of the Verification and Validation of Simulation Models. *Operational Research Quarterly*, 23(1):17–29.
- Moler, C. (1995). A Tale of Two Numbers. *SIAM News*, 28(1):16.
- Morgan, M. S. (2005). Experiments versus Models: New Phenomena, Inference and Surprise. *Journal of Economic Methodology*, 12(2):317–329.
- Morgan, M. S. and Morrison, M. (eds.) (1999). *Models as Mediators: Perspectives on Natural and Social Sciences*. Cambridge University Press.
- Morrison, M. (1990). Unification, Realism and Inference. *The British Journal for the Philosophy of Science*, 41(3):149–305–332.
- (1992). Some Complexities of Experimental Evidence. In *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, 49–62.
- (2000). *Unifying Scientific Theories*. Cambridge University Press, Cambridge.
- (2009). Models, Measurement and Computer Simulation: the Changing Face of Experimentation. *Philosophical Studies*, 143(1):33–57.
- Moser, P. K. (ed.) (2002). *The Oxford Handbook of Epistemology*. Oxford University Press.
- Müller, T. and Müller, H. (2003). *Modelling in Natural Sciences*. Springer.
- Mundy, B. (1986). On the General Theory of Meaningful Representation. *Synthese*, 64:391–437.
- Nagel, E. (1945). Some Reflections on the Use of Language in the Natural Sciences. *The Journal of Philosophy*, 42(23):617–630.

- Naylor, T. H., Burdick, D. S. et al. (1967a). Computer Simulation Experiments with Economic Systems: The Problem of Experimental Design. *Journal of the American Statistical Association*, 62(320):1315–1337.
- Naylor, T. H., Finger, J. M. et al. (1967b). Verification of Computer Simulation Models. *Management Science*, 14(2):92–106.
- Nelson, D. (1992). Deductive Program Verification (a Practitioner’s Commentary). *Minds and Machines*, 2:283–307.
- Neurath, O., Carnap, R. et al. (eds.) (1955). *International Encyclopedia of Unified Science. Foundations of the Unity of Science: Toward an International Encyclopedia of Unified Science*. University of Chicago Press.
- Neyman, J. and Pearson, E. S. (1933). On the Problem of the Most Efficient Tests of Statistical Hypotheses. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 231(694-706):289–337.
- Nitske, W. R. (1971). *The life of Wilhelm Conrad Röntgen, discoverer of the X ray*. University of Arizona Press.
- Oberkampff, W. L. and Roy, C. J. (2010). *Verification and Validation in Scientific Computing*. Cambridge University Press.
- Oberkampff, W. L. and Trucano, T. G. (2008). Verification and Validation Benchmarks. *Nuclear Engineering and Design*, 238(3):716–743.
- Oberkampff, W. L., Trucano, T. G. et al. (2003). Verification, Validation, and Predictive Capability in Computational Engineering and Physics. Tech. rep., Sandia National Laboratories.
- Olsen, A. L. (2005). Using Pseudocode to Teach Problem Solving. *Journal of Computing Sciences in Colleges*, 21(2):231–236.
- Omohundro, S. (1984). Modelling Cellular Automata with Partial Differential Equations. *Physica D: Nonlinear Phenomena*, 10(1-2):128–134.
- Oreskes, N., Shrader-Frechette, K. et al. (1994). Verification, Validation, and Confirmation of Numerical Models in the Earth Sciences. *Science*, 263(5147):641.
- Paolo, D., E. A., J. N. et al. (2000). Simulation Models as Opaque Thought Experiments. In M. A. Bedau, J. S. McCaskill, N. H. Packard and S. Rasmussen (eds.), *Artificial Life VII: Proceedings of the Seventh International Conference on Artificial Life*, 497–506. MIT Press.

- Parker, W. S. (2008). Franklin, Holmes, and the Epistemology of Computer Simulation. *International Studies in the Philosophy of Science*, 22(2):165–183.
- (2009). Does Matter Really Matter? Computer Simulations, Experiments, and Materiality. *Synthese*, 169(3):483–496.
- Penzias, A. and Wilson, R. W. (1965a). A Measurement Of Excess Antenna Temperature At 4080 Mc/s. *Astrophysical Journal Letters*, 142:419–421.
- (1965b). A Measurement of the Flux Density of CAS A At 4080 Mc/s. *Astrophysical Journal Letters*, 142:1149–1154.
- Pfanzag, J. (1959). A General Theory of Measurement Applications to Utility. *Naval Research Logistics Quarterly*.
- Pfleeger, S. L. and Atlee, J. M. (2009). *Software Engineering: Theory and Practice*. Prentice Hall.
- Pincock, C. (2010). Mathematical Contributions to Scientific Explanation.
- (2011). Mathematical Explanations of the Rainbow. *Studies in History and Philosophy of Science Part B*, 42(1):13–22.
- Popper, K. (1935). *The Logic of Scientific Discovery*. Routledge.
- Press, W. H., Teukolsky, S. A. et al. (2007). *Numerical Recipes. The Art of Scientific Computing*. Cambridge University Press.
- Pylyshyn, Z. (1984). *Computation and Cognition: Toward a Foundation for Cognitive Science*. MIT Press.
- Queille, J. P. and Sifakis, J. (1982). Specification and Verification of Concurrent Systems in CESAR. *International Symposium on Programming*, 137:337–351.
- Quine, W. V. (1951). Two Dogmas of Empiricism. *The Philosophical Review*, 60(1):20–43.
- Radder, H. (ed.) (2003). *The Philosophy of Scientific Experimentation*. University of Pittsburgh Press.
- Radford, C. (1966). Knowledge-by examples. *Analysis*, 27:1–11.
- Rapaport, W. J. (1995). Understanding Understanding: Syntactic Semantics and Computational Cognition. *Philosophical Perspectives*, 9:49–88.

- (2003). What Did You Mean by That? Misunderstanding, Negotiation, and Syntactic Semantics. *Minds and Machines*, 13:397–427.
- (2005a). Implementation is Semantic Interpretation: Further Thoughts. *Journal of Experimental & Theoretical Artificial Intelligence*, 17(4):385–417.
- (2005b). Philosophy of Computer Science: An Introductory Course. *Teaching philosophy*, 28(4):319–341.
- Raymond, T. (2006). Computability in Specification. *Journal of Logic and Computation*, 16(6):841–865.
- Redhead, M. (1980). Models in Physics. *The British Journal for the Philosophy of Science*, 31(2):145.
- Reichenbach, H. (1938). *Experience and Prediction: an Analysis of the Foundations and the Structure of Knowledge*. University of Chicago Press.
- Reiss, J. (2011). A Plea for (Good) Simulations: Nudging Economics Toward an Experimental Science. *Simulation & Gaming*, 42(2):243–264.
- Resnik, M. D. and Kushner, D. (1987). Explanation, Independence and Realism in Mathematics. *The British Journal for the Philosophy of Science*, 38(2):141–158.
- Riggs, W. D. (2002). Reliability and the Value of Knowledge. *Philosophy and Phenomenological Research*, 64:79–96.
- Ristić, V. and Radulović, M. (2001). Thought Experiment and Computer Experiment - Similarities and Differences. *Kragujevac Journal of Mathematics*, 23:47–50.
- Rohrlich, F. (1990). Computer Simulation in the Physical Sciences. *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, 507–518.
- Rojas, R. and Hashagen, U. (eds.) (2000). *The First Computers. History and Architectures*. MIT Press.
- Ruben, D.-H. (1992). *Explaining Explanation*. Routledge.
- Salari, P. K. K. (2003). *Verification of Computer Codes in Computational Science and Engineering*. Chapman & Hall.
- Salmon, W. C. (1984). *Scientific Explanation and the Causal Structure of the World*. Princeton University Press.

- (1998). *Causality and Explanation*. Oxford University Press.
- Salmon, W. C., Jeffrey, R. C. et al. (eds.) (1971). *Statistical Explanation and Statistical Relevance*. University of Pittsburgh Press.
- Sargent, R. (2007). Verification and Validation of Simulation Models. In *Proceedings of the 37th conference on Winter simulation*, 124–137.
- Schurz, G. and Lambert, K. (1994). Outline of a Theory of Scientific Understanding. *Synthese*, 101:65–120.
- Scott, D. and Suppes, P. (1958). Foundational Aspects of Theories of Measurement. *The Journal of Symbolic Logic*, 23(2):113–128.
- Scriven, M. (1970). Explanations, Predictions, and Laws. In H. Feigl and G. Maxwell (eds.), *Minnesota Studies in the Philosophy of Science, Vol. III*, University of Minnesota Press.
- Severi, P. and Szasz, N. (2001). Studies of a Theory of Specifications with Built-in Program Extraction. *Journal of Automated Reasoning*, 61–87.
- Shapiro, S. (2000). *Thinking about Mathematics*. Oxford University Press.
- Smith, S., Watt, R. et al. (1984). Cellular Automata in Cytoskeletal Lattices. *Physica D: Nonlinear Phenomena*, 10(1-2):168–174.
- Spivey, J. M. (1992). *Understanding Z: a Specification Language and its Formal Semantics*. Press Syndicate of the University of Cambridge.
- (2001). *The Z Notation: a Reference Manual*. Prentice Hall. Access February 2010.  
**URL:** <http://spivey.oriel.ox.ac.uk/mike/zrm/>
- Staley, K. W. (2004). *The Evidence for the Top Quark: Objectivity and Bias in Collaborative Experimentation*. Cambridge: Cambridge University Press.
- Stegenga, J. (unpublished). Varieties of Evidential Experience.
- Steiner, M. (1978). Mathematics, Explanation, and Scientific Knowledge. *Noûs*, 12(1):17–28.
- (1989). The Application of Mathematics to Natural Science. *The Journal of Philosophy*, 86(9):449.

- Steup, M. and Sosa, E. (eds.) (2005). *Contemporary Debates in Epistemology*. Blackwell.
- Suárez, M. (2003). Scientific Representation: Against Similarity and Isomorphism. *International Studies in the Philosophy of Science*, 17(3):225–244.
- Suárez, M. (ed.) (2009). *Fictions in Science. Philosophical Essays on Modeling and Idealization*. Taylor & Francis.
- Suppe, F. (ed.) (1977). *The Structure of Scientific Theories*. University of Illinois Press.
- Suppe, F. (1989). *The Semantic Conception of Theories and Scientific Realism*. University of Illinois Press.
- Suppes, P. (1960). A Comparison of the Meaning and Uses of Models in Mathematics and the Empirical Sciences. *Synthese*, 287–301.
- (2002). *Representation and Invariance of Scientific Structures*. CSLI Publications.
- Suppes, P. and Zinnes, J. L. (1962). Basic Measurement Theory. Tech. rep., Institute for Mathematical Studies in the Social Sciences - Stanford University.
- Swart, E. (1980). The Philosophical Implications of the Four-Color Theorem. *American Mathematical Monthly*.
- Swoyer, C. (1991). Structural Representation and Surrogate Reasoning. *Synthese*, 87(3):449–508.
- Talbott, W. (Summer 2011 Edition). Bayesian Epistemology. In E. N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy*, .  
**URL:** <http://plato.stanford.edu/archives/sum2011/entries/epistemology-bayesian/>
- Tanenbaum, A. S. and Woodhull, A. S. (2006). *Operating Systems Design and Implementation*. Prentice Hall.
- Tews, H., Völpl, M. et al. (2009). Formal Memory Models for the Verification of Low-Level Operating-System Code. *Journal of Automated Reasoning*, 42(2-4):189–227.
- Thagard, P. (1992). *Conceptual Revolutions*. Princeton University Press.

- Thomson, G. P. (1927b). The Diffraction of Cathode Rays by Thin Film of Platinum. *Nature*, 120:802.
- (1928a). Experiments on the Diffraction of Cathode Rays. *Proceedings of the Royal Society of London*, 117(778):600–609.
- (1928b). Experiments on the Diffraction of Cathode Rays II. *Proceedings of the Royal Society of London*, 119(783):651–663.
- Thomson, G. P. and Reid, A. (1927a). Diffraction of Cathode Rays by a Thin Film. *Nature*, 119:890.
- Toffoli, T. (1984). Cellular Automata as an Alternative to (rather than an approximation of) Differential Equations in Modeling Physics. *Physica D: Nonlinear Phenomena*, 10(1-2):117–127.
- Toulmin, S. (1963). *Foresight and Understanding*. Harper & Row.
- Trenholme, R. (1994). Analog Simulation. *Philosophy of Science*, 61(1):115–131.
- Trout, J. D. (2001). Measurement. In W. H. Newton-Smith (ed.), *A Companion to the Philosophy of Science*, 265–276. Blackwell.
- Turing, A. M. (1936). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42):230–265.
- Turner, R. (2005). The Foundations of Specification. *Journal of Logic and Computation*, 15(5):623–662.
- (2007a). Computable Models. *Journal of Logic and Computation*, 18(2):283–318.
- (2007b). Understanding Programming Languages. *Minds & Machines*, 17(2):203–216.
- Turner, R. and Eden, A. (2011). The Philosophy of Computer Science. In E. N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy*, .  
**URL:** <http://plato.stanford.edu/archives/fall2012/entries/models-science/>
- Tymoczko, T. (1979). The Four-Color Problem and its Philosophical Significance. *The Journal of Philosophy*, 76(2):57–83.

- (1980). Computers, Proofs and Mathematicians: A Philosophical Investigation of the Four-Color Proof. *Mathematics Magazine*, 53(3):131–138.
- van Fraassen, B. C. (1977). The Pragmatics Of Explanation. *American Philosophical Quarterly*, 14(2):143–150.
- (1980). *The Scientific Image*. Clarendon Press.
- Velasco, M. (2002). Experimentacion y Técnicas Computacionales. *Theoria*, 17(44):317–331.
- Vichniac, G. (1984). Simulating Physics with Cellular Automata. *Physica D: Non-linear Phenomena*, 10(1-2):96–116.
- von Neumann, J. (1945). First Draft of a Report on the EDVAC. Tech. rep., United States Army Ordnance Department University of Pennsylvania.
- (1948). Electronic Methods of Computation. *Bulletin of the American Academy of Arts and Sciences*, 1(3):2–4.
- Von Neumann, J. (1958). *The Computer and the Brain*. Yale University Press.
- Weber, E. (2009). How Probabilistic Causation Can Account for the Use of Mechanistic Evidence. *International Studies in the Philosophy of Science*, 23. doi: 10.1080/02698590903197757.  
**URL:** <http://dx.doi.org/10.1080/02698590903197757>
- Weber, M. (2004). *Philosophy of Experimental Biology*. Cambridge University Press.
- Weisberg, M. (2007). Three Kinds of Idealization. *Journal of Philosophy*, 104(12):639.
- (2013). *Simulation and Similarity*. Oxford University Press.
- Wiener, N. (1961). *Cybernetics: Control and Communication in the Animal and the Machines*. MIT Press.
- Winsberg, E. (1999). Sanctioning Models: The Epistemology of Simulation. *Science in Context*, 12:275–292.
- (2001). Simulations, Models, and Theories: Complex Physical Systems and Their Representations. *Philosophy of Science*, 68(3):442–454.
- (2003). Simulated Experiments: Methodology for a Virtual World. *Philosophy of Science*, 70:105–125.

- (2009a). Computer Simulation and the Philosophy of Science. *Philosophy Compass*, 5:835–845.
- (2009b). A Tale of Two Methods. *Synthese*, 169(3):575–592.
- (2010). *Science in the Age of Computer Simulation*. University of Chicago Press.
- Wolfram, S. (1984a). Preface. *Physica 10D*, vii–xii.
- (1984b). Universality and Complexity in Cellular Automata. *Physica D: Nonlinear Phenomena*, 1–35.
- (2002). *A New Kind Of Science*, vol. 56.
- Woodward, J. (2003). *Making Things Happen*. Oxford University Press.
- Woolfson, M. and Pert, G. (1999). *An Introduction to Computer Simulations*. Oxford University Press.
- Wootters, W. and Langton, C. (1990). Is There a Sharp Phase Transition for Deterministic Cellular Automata? *Physica D: Nonlinear Phenomena*, 45:95–104.
- Wu, C. S., Ambler, E. et al. (1957). Experimental Test of Parity Conservation in Beta Decay. *Physical Review*, 105(4):1413–1415.
- Wylie, A. (1999). Rethinking Unity as a ‘Working Hypothesis’ for Philosophy of Science: How Archaeologists Exploit the Disunities of Science. *Perspectives on Science*, 7(3):293–317.
- Young, L. R. (1963). A Discrete Model for Eye Tracking Movements. *IEEE Transactions on Military Electronics*, 7(2):113–115.
- Zagzebski, L. (2000). From Reliabilism to Virtue Epistemology. In G. Axtell (ed.), *Knowledge, Belief, and Character: Readings in Virtue Epistemology*, Rowman & Littlefield Publishers.