Computer simulations

Durán, Juan M.

DOI
10.4324/9781003205647-14

Publication date
2024

Document Version
Final published version

Published in
The Routledge Handbook of Philosophy of Scientific Modeling

11
COMPUTER SIMULATIONS

Juan M. Durán

1. Introduction

Computer simulations are found in a myriad of scientific fields and practices. In some cases, they constitute whole lines of research (e.g., climate modeling and molecular simulations in chemistry (Goldman 2014). The debate over their philosophical merits involves a wide range of topics, including, but not restricted to, their function as experiments (e.g., Beisbart 2017; Boge 2019; El Skaf and Imbert 2013); their value as sources of scientific evidence (e.g., Morgan 2004; Parker 2020); their role as measuring devices (e.g., Morrison 2009; Tal 2011); their place in the scientific methodological map (e.g., Rohrlich 1990); and their scientific and philosophical novelty (e.g., Humphreys 2009; Frigg and Reiss 2009).

A key issue common to many of these debates is how philosophers have conceived—and even defined—computer simulations and the models they implement. This chapter presents and discusses three chief views found in the literature. The first one takes computer simulations to implement mathematical models *simpliciter*. A second one takes computer simulations to be a richer and more complex unit of analysis than mathematical models, yet still related to mathematics. A third viewpoint is sketched, where computer simulations depart even further from implementing mathematical models, gaining the status of modeling in its own right. To simplify the analysis, the focus will primarily be on equation-based simulations and their application to medicine and the natural sciences. Since significant philosophical issues also emerge in relation to diverse fields such as biology, sociology, and psychology, and in relation to a variety of other kinds of computer simulations such as cellular automata, agent-based simulations, and Monte Carlo simulations, let us first look briefly at these. The chapter ends with a discussion on *epistemic opacity*, arguably a chief philosophical issue pertaining to all computer simulations.

2. Kinds of computer simulations

Cellular automata are the first of our examples of computer simulations. They were devised in the 1940s by Stanislaw Ulam and John von Neumann while Ulam was studying the growth of crystals using a simple lattice network as a model and von Neumann was

DOI: 10.4324/9781003205647-14

working on the problem of self-replicating systems. It is said that Ulam suggested to von Neumann that the latter use the same kind of lattice network to create a two-dimensional, self-replicator algorithm.

Cellular automata are simple forms of computer simulations. Their simplicity inheres in both their programming and underlying conceptualization. A standard cellular automaton is an abstract mathematical system in which space and time are considered to be discrete; it consists of a regular grid of cells, each of which can be in any state at a 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. Wolfram defines cellular automata 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 evolves 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 1984, 1)*

Although a rather general characterization of this class of simulation, the definition already provides the first ideas as to their domain of applicability. Cellular automata have been successfully used for modeling many areas in social dynamics (e.g., Thomas Schelling's social segregation model), biology (e.g., patterns of some seashells), and chemical types (e.g., the Belousov–Zhabotinsky reaction). But perhaps the most canonical example is Conway's *Game of Life*. This simulation is remarkable because it constitutes a key example of self-organization dynamics and the emergence of patterns seen in some real-world systems. In this simulation, a cell can survive only if there are either two or three other living cells in its immediate neighborhood. Without these companions, the rule indicates that the cell dies either from overcrowding if it has too many living neighbors or from loneliness if it has too few.

Cellular automata embody a unique set of methodological and epistemological virtues. To name a few, they deal better with errors because they render exact results of the model they implement. Since there is rarely any attempt to approximate the detailed setup of the target system, any disagreement between the model and the empirical data can be ascribed directly to the model that realized the set of rules. Another epistemologically interesting characteristic of cellular automata pointed out by Fox-Keller is that they lack theoretical underpinning in the familiar sense of the term: "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" (Fox-Keller 2003, 208). Consequently, approximations, idealizations, abstractions, and the like are concepts that worry the practitioner of cellular automata very little.

Having said that, cellular automata have been criticized on several grounds. One of these criticisms touches on the metaphysical assumptions behind this class of simulation. It is not clear, for instance, that the natural world is characterized by discrete rather than continuous phenomena, as assumed by the cellular automata. Much contemporary work in science and engineering work assumes that phenomena are, in fact, continuous. On less speculative grounds, it is a fact that cellular automata lack presence in many scientific and engineering fields. The reasons for this might be partially cultural. The physical sciences

are still the accepted viewpoint for describing the natural world, which largely takes form in the language of partial differential equations (PDEs) and ordinary differential equations (ODEs).

Advocates of cellular automata have made efforts to demonstrate their relevance. It has been argued that cellular automata are more adaptable and structurally similar to empirical phenomena than are PDEs or ODEs. Lesne (2007) points out that discrete and continuous behaviors coexist in many natural phenomena (with their proportions depending on the scale of observation) and suggests that this is an indicator not only of the metaphysical basis of natural phenomena, but also of the need to deploy cellular automata to understand them. In a similar vein, Gérard Vichniac believes that cellular automata not only seek numerical agreement with a physical system, but also attempt to match the simulated system's own structure, its topology, its symmetries, and its "deep" properties (Vichniac 1984, 113). Despite these and many other authors' efforts to show that the world might be more adequately described by cellular automata, the majority of scientific and engineering disciplines have not made a significant shift in that direction as of yet. Most of the work done in these disciplines is predominantly based on agent-based and equation-based simulations. As mentioned before, in the natural sciences and engineering, most physical and chemical theories used in astrophysics, geology, climate change, and the like implement PDEs and ODEs, the primary forms of equation-based simulations. Social and economic systems, on the other hand, are better described and understood by means of agent-based simulations.

While there is no general agreement on what precisely an "agent" is, the term typically refers to self-contained programs that control their own actions based on perceptions of their overall operating environment: agent-based simulations "intelligently" interact with their peers as well as their environment.

A key characteristic of these simulations is that they can show how the total behavior of a system emerges from the collective interaction of their parts. Deconstructing these simulations into their constituent elements would remove the added value provided in the first place by the computation of the agents. It is a fundamental characteristic of these simulations, then, that the interplay of the various agents and their environment generates unique behavior in the entire system.

Good examples of agent-based simulations come from the social and behavioral sciences, where they are heavily represented. Perhaps the most well-known example of an agent-based simulation is Schelling's Model of Social Segregation.[1] A very simple description of Schelling's model consists of two groups of agents living in a 2-D,[2] n by m matrix "checkerboard" where agents are placed randomly. Each individual agent has a 3 by 3 neighborhood, which is evaluated by a utility function that indicates the migration criteria. That is, the set of rules that indicates how to relocate—if possible—in case of discontent by an agent.

Schelling's model is a canonical example, but other, more complex agent-based simulations can also be found in the literature. It is now standard for researchers to model a range of different attributes, preferences, and overall behavior in agents. Gilbert and Troitzsch list the attributes that are typically modeled by agent-based simulations, including knowledge and beliefs of the agents, inferences from beliefs, goals, overall planning, and language (Gilbert and Troitzsch 2005).[3]

Monte Carlo methods are the second of our examples of computer simulations. Their basic operation is to use stochastic techniques to compute the properties of a model. A key feature of these methods is that they use random sampling for target systems that could

in principle be deterministic. Monte Carlo is a very powerful technique that is typically applied to systems with many coupled degrees of freedom, such as fluids, gases, crystallizable polymers, and strongly coupled solids, among others. Within the philosophical literature, there has been some debate over its status as a method for discovery and experimentation. Grüne-Yanoff and Weirich, for instance, indicate that "the Monte Carlo approach does not have a mimetic purpose: It imitates the deterministic system not in order to serve as a surrogate that is investigated in its stead but only in order to offer an alternative computation of the deterministic system's properties. In other words, the probabilistic analogy does not serve as a representation of the deterministic system" (Grüne-Yanoff and Weirich 2010, 30). To these authors, then, Monte Carlo experiments are merely methods of calculation and not simulations in a proper sense, for the latter are "used to learn something about the world, and they are used as stand-ins or surrogates for whatever is of interest for the simulationist" (Grüne-Yanoff and Weirich 2010, 30). Beisbart and Norton seem to agree with this idea when they claim that "Monte Carlo simulations are like experiments that discover novel results. We will argue, however, that these sorts of similarities are superficial. They do not and cannot make them function like real experiments epistemically" (Beisbart and Norton 2012, 404).

In what follows, the focus is on the use of computers to find solutions to a set of equations. Equation-based simulations are most commonly used in scientific domains in which the governing theories and models are based on differential equations.

3. Equation-based computer simulations

Suppose we are interested in a simulation of a satellite orbiting around a planet under tidal stress such that it stretches along the direction of the radius vector. Suppose further that this model represents the orbit as non-circular with variable stress, making the satellite expand and contract periodically along the radius vector. Since the satellite is not perfectly elastic, the mechanical energy is converted into heat and radiated away. Despite this, the system as a whole is capable of conserving angular momentum (see, for details, Woolfson and Pert 1999, 18–19). In this context, we have equations of total energy (e.g., Eq. (1) below), angular momentum, and others. We also have other relevant components of the system and their interactions represented in the model. The planet has mass M; the satellite mass m ($<< M$); the orbit is of semi-major axis a; and the gravitational constant is represented by G; and so forth. The masses are represented by connected 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 (e.g., Eq. (2)—see also Woolfson and Pert 1999, 19, fig. 1.8).

$$E = -\frac{GMm}{2a} \tag{1}$$

$$F = k\left(i' + l\right) \tag{2}$$

For simplicity, the above set of equations will be referred to as a *mathematical model*[4] that describes the behavior of and interaction between any planet and any satellite under the specified conditions. Now, to have a simulation, this mathematical model needs to be implemented in the form of an *algorithmic structure*. That is, the sets of variables, procedures, data, functions, and other structures that are tractable in a digital computer (e.g.,

algorithms (3) and (4) partially implementing the mathematical equations). Let us call this algorithmic structure a *simulation model*.

```
TOTM = CM(1) + CM(2) + CH(3) + CM(4);

EN = –G * TOTM + 0.5 * V2                                    (3)

R = SQRT(POS(1)**2 + POS(2)**2 + POS(3)**2)                  (4)
```

The above algorithms suggest that mathematical equations can be implemented as a simulation model rather straightforwardly. These algorithms effectively do so. Algorithm (3) partially implements equation (1) *simpliciter*, and algorithm (4) does something similar with equation (2). Naturally, the simulation model will require some *discretizations* for tractability reasons (i.e., continuous equations cannot be implemented on physical computers), aggregation of procedures for the treatment of *errors*, and a handful of *ad hoc modifications* for smooth numerical integration (e.g., computers cannot represent infinite orbiting).

A critical issue that divides philosophers is how to interpret the simulation model that is at the basis of computer simulations, as well as the computer simulations themselves. To some, computer simulations are numerical methods for finding sets of solutions to mathematical models. To some others, computer simulations are more than numerical methods destined to have merely instrumental value. Instead, they are part of—or stand for—a novel and more comprehensive form of scientific methodology. Thus understood, simulation models are conceived as a new type of model, related to but not entirely obtained from mathematical models and modeling. Key observations favoring this latter view are that any given simulation model will, in fact, involve several layers of models, each potentially requiring differing modeling practices; it will represent structures that are not necessarily present in mathematical models nor secured by mathematical modeling; and it will not necessarily derive from a chain of inferences and varying adjustments and aggregations that started with one or more mathematical models. This second view revolves around the idea that a proper methodology of simulations requires a distinctive ontology leading to specific epistemic and methodological issues.

The remainder of this chapter discusses some of these interpretations and their resulting characterization of simulation models and computer simulations.

3.1 Simulations for analytically intractable mathematics

Let us start with an often-quoted working definition of computer simulation:

> A computer simulation is any computer-implemented method for exploring the properties of mathematical models where analytic methods are unavailable.
>
> *(Humphreys 1990, 501)*

According to this working definition, computer simulations are instrumental in finding the set of solutions to an analytically intractable mathematical model. Understood as numerical methods, they explore the mathematical properties of the simulation models. Hartmann presents a similar definition. According to him, (a) a simulation is the result of solving the equations of a dynamic model, and (b) a computer simulation is the result of having a simulation

run on a physical computer. Taken together, (a) and (b) entail that a computer simulation results when a dynamic mathematical model is solved by a physical computer (Hartmann 1996). Let us note that Hartmann is also claiming that the physical dimension of the computer plays a relevant role in imitating the dynamics of a real-world system. Interestingly, some philosophers have developed this idea (e.g., Parker 2009, and Boge 2020), arguing for meaningful morphisms between the (physical) computer processes and the target system.[5] Others, opposing this claim (e.g., Beisbart 2014; Durán 2018), argue that the multirealizability of physical processes means that the resulting analogy is thin and contrived.

These definitions come with varying methodological and epistemological assumptions. For starters, the adjustments required for implementing the mathematical model onto the computer must be minimal. That is, the discretizations and ad hoc modeling must go only as far as is required for the tractability of the mathematical model. By themselves, simulations do not possess—nor should they possess—any representational value other than that inherited from the mathematical models they deploy. No aggregates to the simulation model could suggest a deviation from the implemented mathematical models.

Humphreys' and Hartmann's definitions loom large in the philosophical and technical literature. Parker, for instance, adopts Hartmann's definition in her analysis of the experimental value of simulations. In her 2009 paper, she makes explicit reference to it by characterizing a computer simulation as a time-ordered sequence of states that represents another time-ordered sequence of states. In her latest publication, however, she seems to have distanced herself from this commitment. She states that "a *computer simulation model* is a computer program that is designed to iteratively solve a set of dynamical modeling equations, either exactly or approximately, following a particular algorithm" (Parker 2020, sec. 2). Moreover, Parker also calls attention to the plurality of models in simulation practice and their role in computer simulations in climate models (see the next section). It would require some argumentative acrobatics to make a convincing case that climate simulations hold nontrivial morphisms at the physical level.

Guala has also made explicit reference to Hartmann's definition in discussing the time evolution of systems, the use of simulations to provide numerical solutions to sets of mathematical equations, and in distinguishing between static and dynamic models (Guala 2002). Krohs (2008) adopts Humphreys' and Hartmann's definitions to account for the role and merits of computer simulations in scientific explanation (Durán 2017). Frigg and Reiss largely base their disapproval of the philosophical novelty of computer simulations on a *narrow* sense of simulations, assuming that they are, ultimately, about mathematical models (Frigg and Reiss 2009, 596).

Recently, Boge has claimed that a simulation model "will usually (if not always) be based on some previously existing numerical, i.e., discrete mathematical model of a system of interest (the 'target system'), which in many cases is an approximation to another model based on continuous mathematics, and hence not suited for a translation into algorithms" (Boge 2019, 3). Boge goes on to discuss simulations in terms of mathematical language and derivations, as well as the physical characteristics of the target system mimicked by, and emerging from, the execution of such simulations.

3.2 Simulations as a "new type" of mathematical model

The alternative viewpoint takes that simulation models are related to, but not entirely obtained from, mathematical models and modeling. Weisberg, in his analysis of the anatomy

of models, considers simulation models as "a subset of mathematical models" (Weisberg 2013, 30) but holds that they constitute an especially important subset. Morrison has also urged that more philosophical attention must be given to computer simulations, in light of their being a special kind of experimental practice related to modeling (Morrison 2015).[6] In his recent book, Lenhard explicitly refers to simulations as a "*new* type" of mathematical model. There are two sides to this interpretation. Whereas simulation models must be "counted into the established classical and modern class of mathematical modeling," one must also take stock on how they "contribute to a novel explorative and iterative mode of modeling characterized by the ways in which simulation models are constructed and fitted" (Lenhard 2019, 7). Lenhard cements this view by saying: "[o]ne direction seems self-evident: the (further) development of computers is based primarily on mathematical models. However, the other direction is at least just as important: the computer as an instrument channels mathematical modeling" (Lenhard 2019, 8). Simulations are a "new type" of model primarily because of the *plasticity* of their modeling, which "draws on the effects that arise from the ways in which the (artificial) parameters are set. The more flexible a model is, the more significant is the phase of modeling during which the parameters are adjusted."[7] (Lenhard 2019, 11).

What does the methodology of simulations as a "new type" of mathematical model look like? Winsberg provides an answer to this question. This author advances a hierarchy of models that begins, at the top, with a given theory (i.e., general physical and modeling assumptions) and terminates, after a series of specifications, alterations, and inferences at each level of modeling with a model of the phenomena, which represents the outcome of the simulation research in question (Winsberg 1999, 277). In Winsberg's view, this inferential hierarchy suggests a distinct epistemology—and, it could be added, a distinct methodology—for simulations whose chief features are being *downwards*, *autonomous*,[8] and *motley* (Winsberg 2001, S447). It follows that "simulations often do not bear a simple, straightforward relation to the theories from which they stem" (Winsberg 1999, 276).

Humphreys also offers an elaborated, multi-level methodology and epistemology for simulation models. He presents it in the following way: "System S provides a core simulation of an object or process B just in case S is a concrete computational device that produces, via a temporal process, solutions to a *computational model* [...] that correctly represents B, either dynamically or statically. If in addition the computational model used by S correctly represents the structure of the real system R, then S provides a core simulation of system R with respect to B" (Humphreys 2004, 110, emphasis added). The computational model comprises six different elements, each performing a specific function. These are the computational template, the construction assumptions of that model, the correction set, an interpretation, an initial justification, and the output representation (see Humphreys 2004, 102). The first element of this sextuple, i.e., the computational template, is the heart of the computational model and can essentially be understood as a set of computationally tractable equations (61).

Taking stock of these interpretations, simulation models are still obtained from mathematical models in varying degrees and fashions. With Winsberg, this comes through the hierarchical-inferential process that ultimately results in a model of the phenomena. For Humphreys, the unit of analysis for computational science is the *computational template*. Following his example, a simulation utilizing Newton's Second Law consists of a theoretical template that "describes a very general constraint on the relationship between any force, mass, and acceleration, but to use it in any given case, we need to specify a particular force

function, such as a gravitational force, an electrostatic force, a magnetic force, or some other variety of force" (Humphreys 2004, 60). A computational template emerges when "the resulting, more specific, equation form is computationally tractable" (60). Finally, Lenhard intends to balance the transformations of mathematical models introduced by the computer with the role of simulations as instruments that channel mathematical modeling.

One must then ask, to what extent are these interpretations aligned or misaligned with the notion of simulations as a way of approaching analytically intractable mathematics? While there is some evident overlap, there are also a handful of reasons to separate these two notions. For starters, simulation models are conceived as a richer structure than mathematical models by philosophers arguing for the novelty of simulation modeling (e.g., they use external databases, involve multiple layers of models). This also means that the goal of simulations has substantially shifted from finding solutions to a set of equations representing a complex target system. Finally, scientific research involving computer simulations does not necessarily reflect the same epistemic and methodological principles, social organization, and research questions as those involving mathematical models.

Climate simulations have made visible the rich and complex structure of simulations, primarily through the implementation of a plurality of models. In fact, many philosophers agree that model pluralism is an inherent and inevitable feature of simulation models. As Lenhard and Winsberg (2010, 261) put it, "pluralism is not a temporary failure that eventually will be overcome, but will remain for principled reasons of simulation modeling methodology." Parker has argued that "complex climate models generally are physically incompatible with one another—they represent the physical processes acting in the climate system in mutually incompatible ways and produce different simulations of climate" (Parker 2006, 350). Durán (2020) has reflected on the plurality of models in regard to the architecture of simulation models. There, simulation models *recast* a host of models pertaining to different kinds of representational values, methodological principles, and epistemic goals. The resulting architecture includes *kernel simulations*, understood as the implementation of each individual model in the formalism of a programming language, and *integration modules*—modules "which play two fundamental roles, namely, they integrate external databases, protocols, libraries and the like with [each kernel simulation], and ensure the synchronization and compatibility among [the kernel simulations]" (Durán 2020, 307). Computer simulations are therefore conceived as non-hierarchical, non-inferential, and non-homogeneous units of analysis.

3.3 Can simulations be autonomous from mathematical models?

The view that simulations are a "new type" of mathematical models tends to obscure the tension between acknowledging that simulation models both provide an unprecedented form of modeling and a forceful attempt to stay rooted in mathematical modeling. For instance, Winsberg introduced the idea of ad hoc modeling, understood as "relatively simple mathematical relationships designed to approximately capture some physical effect in nature. When 'coupled' to the more theoretical equations of a simulation, they allow the simulation to produce outputs that are more realistic than they could have been without some consideration of that physical effect" (Winsberg 1999, 282). Another distinctive methodological practice in simulation is "kludging," roughly understood as adding bits of code to simulation that are not principled in their design and whose purpose is to optimize the performance and improve the simulation in a "quick and dirty" way (Lenhard 2019).

But kludging is not the only distinctive methodological trick implemented in simulations. Fuzzy modularity (i.e., the piecemeal adjustment of models for their use in multiple simulations) and generative entrenchment (i.e., the multiple sources on which the model depends because they played a role in generating it) also cement claims about confirmatory holism and explain the failure of analytic understanding in climate models, for instance (Lenhard and Winsberg 2010, 256–257). Yet another interesting example is the so-called Arakawa operator, also discussed by Lenhard and Winsberg, which can be used to overcome the nonlinear instability of the mathematics in meteorological models. In this respect, Lenhard says: "[i]n my opinion, this was a decisive point: the discreteness of the model required artificial and also nonrepresentative elements in the simulation model whose dynamic effects could be determined only in a (computer) experiment" (Lenhard 2019, 36). Finally, parametrizations further engross the list as they are "pragmatic decisions that balance fidelity to what we know about the target system with the need for effective implementation" (Lenhard and Winsberg 2010, 256).

What does this alleged distinctive form of modeling mean for the representational merits of simulations? In principle, not much. Ad hoc modeling takes it that "more" modeling is added to the simulation for reasons of tractability, but there is no claim of added representational value. Kludging, fuzzy modularity, Arakawa-like operators, and parametrization are genuine simulation-inspired practices, but they are also "nonrepresentative" of the target system (Lenhard 2019, 36). Again, they are solely dedicated to making the simulation model tractable.

Interestingly, it is increasingly the case that mathematical and logical formalism is omitted in favor of readymade algorithmic structures. Researchers prefer to dispense with the trouble of first developing a mathematical model and then figuring out how to implement it as (part of) a simulation model by representing target systems directly into their codes. For instance, DeAngelis and Grimm (2014) and Peck (2012) show how a (total or partial) representation by the simulation model might take place directly at the level of algorithmic structures and without the mediation of any formal mathematical modeling. The representation is built from hypothesized relational structures abstracted from the target system and directly coded as the simulation model.

One could object at this point that readymade algorithmic structures are conducive to other forms of modeling. That the practice of dispensing with the writing of mathematical equations before coding the algorithm does not necessarily imply that there is no mathematical model underpinning the algorithm.[9] But the critical point here is that, on occasion, researchers encode forms of behavior of the target system that do not correlate with mathematical modeling. To put this idea somewhat differently: if we want to recreate the algorithm as a mathematical model, we would face the problem that specific structures and patterns of behavior relevant to the representation of the target system and encoded in the algorithm do not correspond to mathematical machinery. Durán (2020) explores this idea, arguing that programming languages allow researchers to encode into their simulation-specific structures and patterns of behavior of the target system. The key intuition here is that a given simulation might represent two non-trivially different target systems depending on the chosen programming language, code execution, and the like. Constraints on behavior and behavioral decisions are, on many occasions, conditional on circumstances. For example, *if-then statements* and other forms of programming conditionals might constrain the behavior of the simulation and, as such, configure non-trivially different target systems. Durán (2022) illustrates this with a simulation of spatiotemporal patterns of respiratory

anthrax infection in a population (see Cooper et al. 2004). In this simulation, the network of nodes and subnodes can be directly coded into the simulation through nested conditionals (i.e., no mathematical formalism is required). As such, and depending on the conditional executed, the simulation would represent different valid paths in the proliferation and spread of the infection, distinctive states of the infection at any given time, and the like.

Can it be assumed that programming languages and code execution constitute legitimate forms of representation that are not necessarily reliant on mathematical models? Some researchers seem to think so (Aronis et al. 2020). Simulation models also seem to allow this kind of philosophical speculation. Clearly, more research is needed in this direction. It remains an open question, whether kludging, Arakawa-like operators, and other computational-inspired practices have representational value or are solely instrumental to the tractability of the simulation model.

4. A new scientific methodology

Where can computer simulations be located in the methodological map? Famously, Rohrlich placed them somewhere intermediate between theoretical physical science and its empirical methods of experimentation and observation (Rohrlich 1990, 507). This view strikes now as too narrow, even for equation-based simulations. The prevailing view is that computer-based methodologies rather extend the class of tractable mathematics and representation and thereby broaden the ranges of modeling (Morgan 2003), observations (Beisbart 2017), predictions (Parker 2014), measurements (Morrison 2009; Tal 2011), and explanation of phenomena (Durán 2017), among several other scientific endeavors. That is to say, computer simulation is not just an intermediate between two familiar ends, but rather a scientific methodology in its own right. Furthermore, there are good reasons to believe that computer simulations raise new epistemological issues, arguably without a precedent in the philosophy of science. This point has forcefully been made by Humphreys and constitutes a central element of his understanding of computer-based methodologies. To be precise, Humphreys distinguishes between anthropocentric epistemologies, which "involve representational intermediaries that are tailored to human cognitive capacities" (Humphreys 2009, 617), and non-anthropocentric epistemologies, where "there now exist superior, non-human, epistemic authorities" (Humphreys 2009, 617). Computer simulations belong to the latter class.

In this context, the claim arises that computer simulations are *epistemically opaque* in that "no human can examine and justify every computational step performed by the computer, because those steps are too numerous" (Parker 2014).[10] What, more precisely, does *epistemic opacity* amount to? Humphreys discusses two related but distinct definitions. The first definition—sometimes referred to as *general epistemic opacity* (GEO) (Alvarado 2021; Beisbart 2021)—says that a given process is opaque to an agent to the extent that the said agent does not know (that is, cannot check, trace, or survey) all of the epistemically relevant elements of the process. Here, a process is broadly understood as the different methods, devices, systems, or instruments of interest. What constitutes an epistemically relevant element of the process will depend on the kind of process involved (Humphreys 2009, 618). For instance, a mathematical proof can be considered the process, and a given lemma is a relevant element in that process. The second definition specifies that a process is *essentially epistemically opaque* (EEO) to an agent if it is *impossible*, given the nature of the agent, to know all the epistemically relevant elements of the process. For instance, the weather

forecast for the next two years is impossible to predict by climatologists given their cognitive limitations to handle all the variables involved in such complex systems.

Philosophically speaking, there are a few distinctions of interest between GEO and EEO. For instance, the former is tailored to diverse contingencies, such as context, efforts, goals, and the current state of knowledge of the agent(s). In other words, GEO comes in degrees.[11] Consider Humphreys' own example: "for a mathematical proof, one agent may consider a particular step in the proof to be an epistemically relevant part of the justification of the theorem, whereas to another, the step is sufficiently trivial to be eliminable" (Humphreys 2009, 618). The first agent's knowledge of the proof might change over time, say, in light of a new piece of information. This agent then decides to join the second agent in that that particular step in the mathematical proof is utterly irrelevant. Context, goals, efforts, and the current state of an agent's (or agents') knowledge vary over time, as does practice, and the agents themselves. In contrast, EEO takes it that it is the very nature of agents that prevents knowing all the relevant elements of the process: "[m]any, perhaps all, of the features that are special to simulations are a result of this inability of human cognitive abilities to know and understand the details of the computational process" (Humphreys 2009, 618–619). In other words, a process is essentially epistemically opaque, not because the agent does not know a given relevant epistemic element in the process, but because the agent will never know, given their nature, any of the relevant epistemic elements in the process. EEO is not contingent upon the agent's epistemic context, goals, or efforts, but rather it is an absolute matter about the nature of the agent.

Here we should note that both GEO and EEO are understood from the agent-relative perspective. Whereas in GEO there might be a point in the future where a process ceases to be opaque (e.g., because the mathematician decides that the step is irrelevant for the proof), in EEO agents are by their constitutional nature unable to access the relevant elements of the process. This might either be because they are cognitively limited (e.g., a computer algorithm involves too many steps) or time-restricted (e.g., the algorithm would take long to compute). Agent-relative epistemic opacity is very much the way in which the literature has discussed this issue so far (Beisbart 2021; Durán and Formanek 2018), including the most recent and, sadly, last article on computer simulations by Humphreys (Humphreys, 2022). Interestingly, in this article, Humphreys extends the interpretation of "agent" to also include computer algorithms, with the result that, if we ask questions about ameliorating opacity, one could always think of a third-party algorithm fulfilling this role. This idea is extensively exploited in the literature on *transparency*, especially in the context of machine learning. This said, while trading human agents for algorithms does have some appeal, it does not come cheap. A particularly pressing issue is the *algorithmic regress* that transparency presupposes. To illustrate this, consider an algorithm A that is epistemically opaque. Suppose we make use of A_1, a third-party algorithm that can, presumably, provide knowledge on the relevant elements e in A. Given that A_1 is by definition also epistemically opaque, we are not yet in a position to claim knowledge of e. For this, we need to turn to a second algorithm, A_2 for dealing with the opacity of A_1. The regress continues until either we reach a simple algorithm A_n of which we know all the relevant elements or we abruptly decide to stop the regress.

In a later work, Alvarado challenges the agent-based view on opacity on the basis that "there are instances of epistemic opacity that are either neutral to and/or independent from the limitations of agents. That is, they arise in virtue of factors that are not responsive to or are not related to agential resources" (Alvarado 2021, 9). Whereas Alvarado admits that this

description of agent neutrality remains close to agent-based viewpoints (e.g., "as far as accounts of epistemic opacity go, agent-neutral instances of opacity can still be formulated in relation to agential limitations" (10)), agent independency poses an interesting departure from both standard views. According to Alvarado, "an account of agent-independent opacity must include both the fact that the opacity does not arise in virtue of anything related to an agent *and* the fact that it is not responsive to agential resources and/or efforts" (13). In other words, a process is EEO to an agent if it is impossible, given the nature of the *process*, to know all its epistemically relevant elements.[12] Borrowing Alvarado's example, we can say that a stochastic process is agent-independent opaque in virtue of "the combination of its stochasticity (the randomness of paths chosen) and the vast overdetermination (the fact that many—too many—different paths lead to the same outcome) [which makes] inquiry into the actual paths taken (the relevant epistemic elements of the process) inaccessible" (Alvarado 2021, 14).

This more nuanced, process-centered approach to EEO proposed by Alvarado is a welcome addition to the literature, particularly because it offers a way to account for cases where opacity cannot be explained by the cognitive limitations of agents. However, more needs to be said. For instance, it remains unexplained on what grounds a process is to be considered inherently opaque. Without this, it is difficult to distinguish between processes that permanently remain opaque from those that might cease to be opaque at some point in the future. Furthermore, an argument must be provided such that it excludes non-human agents (e.g., algorithms) from accessing inherently opaque processes. Indeed, Alvarado's argument doesn't seem to work if the agent is non-human. Let us recall that Humphreys accepts that algorithms can channel insight into the epistemically relevant elements of a process (Humphreys, 2004, p. 150).

Complementary to these debates are attempts to deal with opacity. Above, I mentioned *transparency*, nowadays gaining significant traction in philosophical debates over machine learning. The core idea of transparency is to make algorithms accessible by showing the inner workings and properties of the algorithm (e.g., Creel 2020). The opposing view is *computational reliabilism*, understood as a set of methods and practices that credit reliability to an algorithm under conditions of opacity (Durán and Formanek 2018; Humphreys 2022; Durán, forthcoming). In other words, whereas transparency makes efforts to grant (human) access to algorithms, computational reliabilism accepts their opacity and focuses instead on the conditions for epistemically trusting them.

There is still plenty of room for further philosophical debate on epistemic opacity and the different specific conceptions of it that figure in debates over computer simulations. But perhaps the greatest contribution of these debates to our understanding of computer simulations (and machine learning) is to bring to the fore their merits as units of philosophical analysis in their own right.

Acknowledgments

Many thanks to Edoardo Datteri, Florian Boge, Ramón Alvarado, and Nico Formanek for comments on earlier versions of this chapter. Their acute insights have made their way into this chapter. This chapter is in great debt to the editors of this Handbook, who read and carefully commented on the several versions of the chapter. Thank you also for your encouragement and patience. All errors are my sole responsibility.

This work is partially supported by the EU program under the scheme "INFRAIA 2020-2024-SoBigData++: European Integrated Infrastructure for Social Mining and Big

Data Analytics," grant agreement 871042. It is also partially supported by the EU program under the scheme "ICT48 Humane AI Net," grant agreement 952026. Their support is gratefully acknowledged.

Notes

1 Although nowadays Schelling's model is implemented using computers, Schelling himself warned against their use for understanding the model. Instead, he used coins or other elements to show how segregation occurred. In this respect, Schelling says: "I cannot too strongly urge you to get the nickels and pennies and do it yourself. I can show you an outcome or two. A computer can do it for you a hundred times, testing variations in neighborhood demands, overall ratios, sizes of neighborhoods, and so forth. But there is nothing like tracing it through for yourself and seeing the thing work itself out. In an hour you can do it several times and experiment with different rules of behavior, sizes and shapes of boards, and … subgroups of dimes and pennies that make different demands on the color compositions of their neighborhoods" (Schelling 1971, 85). Schelling's warning against the use of computers is an amusing anecdote that illustrates how scientists could sometimes fail in predicting the role of computers in their own respective fields.

2 Schelling also introduced a 1-D version, with a population of 70 agents, with the four nearest neighbors on either side, the preference consists of not being minority, and the migration rule is that whoever is discontented moves to the nearest point that meets her demands (Schelling 1971, 149).

3 For a more thorough review of kinds of computer simulations, see (Durán chap. 1).

4 Here, a *mathematical model* is a generic term covering any scientific, non-physical model, such as theoretical models, data models, phenomenological models, and the like (Frigg and Hartmann 2020).

5 Thanks to Florian Boge for pressing on this point.

6 In her view, computer simulations are the "result of applying a particular kind of discretization to the theoretical/mathematical model […] There are several reasons for characterizing this type of investigation as an experiment, or more properly, a computer experiment" (Morrison 2015, 219). Thanks to Ramón Alvarado for this reminder.

7 The flexibility of a model is measured as the capacity to implement "generic structures" and the associated possibility of reusing the model in different contexts.

8 Autonomy is attributable to the scarcity of data rather than being a methodological principle of models and modeling.

9 Thanks to Edoardo Datteri for pressing on this point.

10 There is a burgeoning literature that discusses other forms of opacity, such as social opacity (Longino 1990), methodological opacity (Beisbart 2021), corporate opacity (Burrell 2016), and representational opacity (Humphreys 2022), just to mention a few.

11 In Humphrey's words, "[i]t is obviously possible to construct definitions of 'partially epistemically opaque' and 'fully epistemically opaque'" (Humphreys 2009, n. 5).

12 Alvarado provides his own working definition; see (Alvarado 2021, 13).

References

Alvarado, Ramón. 2021. "Explaining Epistemic Opacity." Forthcoming in *The Science and Art of Simulation II*, edited by Andreas Kaminski, Michael Resch, and Petra Gehring. Berlin: Springer Verlag. (References in the text are to the 2021 online preprint version.)

Aronis, John M., Jeffrey P. Ferraro, Per H. Gesteland, Fuchiang Tsui, Ye Ye, Michael M. Wagner, and Gregory F. Cooper. 2020. "A Bayesian Approach for Detecting a Disease that is not Being Modeled." *PLoS One* 15(2):1–15. https://doi.org/10.1371/journal.pone.0229658.

Beisbart, Claus. 2014. "Are We Sims? How Computer Simulations Represent and What This Means for the Simulation Argument." *The Monist* 97(3): 399–417.

———. 2017. "Are Computer Simulations Experiments? And If Not, How Are They Related To Each Other?" *European Journal for Philosophy of Science* 8(2): 171–204.

———. 2021. "Opacity Thought Through: On the Intransparency of Computer Simulations." *Synthese* 199: 11643–11666.
Beisbart, Claus and John D. Norton. 2012. "Why Monte Carlo Simulations are Inferences and Not Experiments." *International Studies in the Philosophy of Science* 26(4): 403–422.
Boge, Florian J. 2019. "Why Computer Simulations Are Not Inferences, and in What Sense They Are Experiments." *European Journal for Philosophy of Science* 9(13). https://doi.org/10.1007/s13194-018-0239-z.
———. 2020. "How to Infer Explanations from Computer Simulations." *Studies in History and Philosophy of Science Part A* 82: 25–33.
Burrell, Jenna 2016. "How the Machine 'Thinks': Understanding Opacity in Machine Learning Algorithms." *Big Data & Society* 3(1): 1–12.
Cooper, Gregory F., Denver Dash, John Levander, Weng-Keen Wong, William R. Hogan, and Michael M. Wagner. 2004. "Bayesian Biosurveillance of Disease Outbreaks." In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, ed. Christopher Meek, Max Chickering and Joseph Halpern, 94–103. Arlington, VA: AUAI Press.
Creel, Kathleen A. 2020. "Transparency in Complex Computational Systems." *Philosophy of Science* 87(4): 568–589.
DeAngelis, Donald L. and Volker Grimm. 2014. "Individual-based Models in Ecology after Four Decades." *F1000Prime Reports* 6(39): 1–6.
Durán, Juan M. 2017. "Varying the Explanatory Span: Scientific Explanation for Computer Simulations." *International Studies in the Philosophy of Science* 31(1): 27–45.
———. 2018 *Computer Simulations in Science and Engineering. Concept, Practices, Perspectives*. Switzerland: Springer.
———. 2020. "What is a Simulation Model?" *Minds and Machines* 30: 301–323.
———. 2022. "Models, Explanation, Representation, and the Philosophy of Computer Simulations." In *Philosophy of Computing* (*Philosophical Studies* 143), ed. Björn Lundgren and Nancy A. Nuñez Hernández. Berlin: Springer pages 221–249.
Durán, Juan M. and Nico Formanek. 2018. "Grounds for Trust: Essential Epistemic Opacity and Computational Reliabilism." *Minds and Machines* 28(4): 645–666.
Durán, Juan M (forthcoming) "Beyond transparency: computational reliabilism as an externalist epistemology for algorithms" ed Juan M. Durán and Giorgia Pozzi. Synthese Library. Springer Cham.
El Skaf, Rawad and Cyrille Imbert. 2013. "Unfolding in the Empirical Sciences: Experiments, Thought Experiments and Computer Simulations." *Synthese* 190: 3451–3474.
Fox-Keller, Evelyn. 2003. "Models, Simulations, and 'Computer Experiments'." In *The Philosophy of Scientific Experimentation*, ed. Hans Radder, 198–215. Pittsburgh: University of Pittsburgh Press.
Frigg, Roman and Stephan Hartmann. 2020. "Models in Science." In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta, https://plato.stanford.edu/archives/spr2020/entries/models-science/.
Frigg, Roman and Julian Reiss. 2009. "The Philosophy of Simulation: Hot New Issues or Same Old Stew?" *Synthese* 169(3): 593–613.
Gilbert, Nigel and Klaus G. Troitzsch. 2005. *Simulation for the Social Scientist*, 2nd edn. Maidenhead; New York: Open University Press.
Goldman, Nir. 2014. "A Virtual Squeeze on Chemistry." *Nature Chemistry* 6(November): 1033–1034.
Grüne-Yanoff, Till and Paul Weirich. 2010. "The Philosophy and Epistemology of Simulation: A Review." *Simulation & Gaming* 41(1): 20–50.
Guala, Francesco. 2002. "Models, Simulations, and Experiments." In *Model-based Reasoning: Science, Technology, Values*, edited by Lorenzo Magnani and Nancy J. Nersessian, 59–74. Boston, MA: Springer US.
Hartmann, S. 1996. "Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View." In *Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View*, ed. Rainer Hegselmann, Ulrich O. Mueller, and Klaus G. Troitzsch, 77–100. Berlin: Springer.
Humphreys, Paul W. 1990. "Computer Simulations." *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association* 2: 497–506.
———. 2004. *Extending Ourselves: Computational Science, Empiricism, and Scientific Method*. Oxford: Oxford University Press.

———. 2009. "The Philosophical Novelty of Computer Simulation Methods." *Synthese* 169(3): 615–626.
———. 2022 "Epistemic Opacity and Epistemic Inaccessibility." Accessed 21 Sep 2022 https://wordpress.its.virginia.edu/Paul_Humphreys_Home_Page/files/2016/02/epistemic-opacity-and-epistemic-inaccessibility.pdf
Krohs, Ulrich. 2008. "How Digital Computer Simulations Explain Real-World Processes." *International Studies in the Philosophy of Science* 22(3): 277–292.
———. 2019. *Calculated Surprises*. Oxford: Oxford University Press.
Lenhard, Johannes and Eric Winsberg. 2010. "Holism, Entrenchment, and the Future of Climate Model Pluralism." *Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics* 41(3): 253–262.
Lesne, Annick. 2007. "The Discrete versus Continuous Controversy in Physics." *Mathematical Structures in Computer Science* 17(2): 185–223.
Longino, Helen E. 1990. *Science as Social Knowledge: Values and Objectivity in Scientific Inquiry*. Princeton, NJ: Princeton University Press.
Morgan, Mary S. 2003. "Experiments without Material Intervention: Model Experiments, Virtual Experiments and Virtually Experiments." In *The Philosophy of Scientific Experimentation*, ed. Hans Radder, 216–235. Pittsburgh, PA: University of Pittsburgh Press.
———. 2004. "Simulation: The Birth of a Technology to Create 'Evidence' in Economics." *Revue d'histoire des sciences* 57(2): 339–375.
Morrison, Margaret. 2009. "Models, Measurement and Computer Simulation: The Changing Face of Experimentation." *Philosophical Studies* 143(1): 33–57.
———. 2015. *Reconstructing Reality: Models, Mathematics, and Simulations*. Oxford: Oxford University Press.
Parker, Wendy S. 2006. "Understanding Pluralism in Climate Modeling." *Foundations of Science* 11: 349–368.
———. 2009. "Does Matter Really Matter? Computer Simulations, Experiments, and Materiality." *Synthese* 169(3): 483–496.
———. 2014. "Computer Simulation." In *The Routledge Companion to Philosophy of Science*, ed. Martin Curd and Stathis Psillos 135–145. New York: Routledge.
———. 2020. "Evidence and Knowledge from Computer Simulation." *Erkenntnis* 87: 1521–1538.
Peck, Steven L. 2012. "Agent-based Models as Fictive Instantiations of Ecological Processes." *Philosophy and Theory in Biology* 4: 1–12.
Rohrlich, Fritz. 1990. "Computer Simulation in the Physical Sciences." *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association* 2: 507–518.
Schelling, Thomas C. 1971. "On the Ecology of Micromotives." *National Affairs* 25: 61–98.
Tal, Eran. 2011. "From Data to Phenomena and Back Again: Computer-simulated Signatures." *Synthese* 182(1): 117–129.
Vichniac, Gérard Y. 1984. "Simulating Physics with Cellular Automata." *Physica D: Nonlinear Phenomena* 10(1–2): 96–116.
Weisberg, Michael. 2013. *Simulation and Similarity: Using Models to Understand the World*. Oxford: Oxford University Press.
Winsberg, Eric. 1999. "Sanctioning Models: The Epistemology of Simulation." *Science in Context* 12(2): 275–292.
———. 2001. "Simulations, Models, and Theories: Complex Physical Systems and Their Representations." *Philosophy of Science* 68(3): S442–S454.
Wolfram, Stephen. 1984. "Universality and Complexity in Cellular Automata." *Physica* 10D: 1–35.
Woolfson, Michael M. and Geoffrey J. Pert. 1999. *An Introduction to Computer Simulation*. Oxford: Oxford University Press.