On the Epistemology of Computer Simulations

Leon van der Grient

Bachelor thesis
Supervisor: Dr. Janneke van Lith
Utrecht University

Contents

Chapter 1

Introduction

1.1 The philosophy of computer simulations

The philosophy of computer simulations has received a lot of attention over the last two decades. This is of course closely linked to the emergence of computer simulations in many of the natural sciences, as well as in the economic and social sciences. They had a great impact on disciplines as is mentioned by the computer scientist Michael Heymann: "Computer processing and simulation transformed fields like meteorology and climate research into computer sciences."[1] The advances in technology made the computer, and consequently the computer simulation, a common scientific tool.

Although the *scientific* impact might be clear, we may ask ourselves about the *philosophy* of computer simulations. According to Heymann: "The computer has not only enabled science to explore complex system behaviour, it [...] caused epistemic shifts."[2] And the philosopher Paul Humphreys writes: "[P]owerful new currents sweeping through the sciences bring with them philosophical challenges that older modeling frameworks cannot address."[3]

These challenges can be linked to four different areas of philosophy and, thus, address different questions and problems. The philosophy of computer simulations can focus on:

1. *Metaphysics:* Do computer simulations create some sort of 'parallel worlds'?
2. *Semantics:* How do models (and thus simulations) relate to the real world?
3. *Methodology:* What kind of activity is running a computer simulation? Especially, is it an experiment?
4. *Epistemology:* How do we *know* propositions or facts that we obtain from a computer simulation?

This paper focuses on the epistemological question and will only touch upon some of the other issues. Before we proceed, a little bit more needs to be said about epistemology itself.

1.2 Epistemology

Epistemology, the philosophy of knowledge, addresses a variety of questions concerning the nature and structure of knowledge. Before we can discuss the epistemology of computer simulations, we first need to specify what we understand as 'epistemological' questions. I will briefly discuss the most prominent issues in epistemology.

[1] Heymann 2010, p. 195.
[2] Heymann 2010, p. 194.
[3] Humphreys 2009, p. 625.

1.2.1 A definition of 'knowledge'

One of the tasks of epistemology is to provide a definition of 'knowledge', which proves to be a very difficult task. A first plausible attempt is to define knowledge as *justified true belief* (or JTB for short). However, this attempt was shown to be unsuccessful by Edmund Gettier in the 1960s.[4] Modifications were made by numerous philosophers (see for example Nozick [1981] and Goldman [1976]) who, for a large part, mainly tried to tighten the link between the justification and the truth of the belief. Recently some philosophers have taken a different approach—for instance by making knowledge context dependent (contextualism, DeRose [2009]) or subject dependent (subject sensitive invariantism, Hawthorne [2004]).

Although giving a precise definition of knowledge is bound to face objections, I think it is safe to say that knowledge is 'some sort of' JTB. This means that to know p one has to belief p,[5] be justified to belief p,[6] and p has to be true.[7]

A notational remark

For convenience sake, let us introduce the following notation for a knowing subject S that knows p and uses justification method $\mathcal{M}$: $p_S^{\mathcal{M}}$. Let us also define ω as a proposition (or fact) yielded by a computer simulation and Ω as a set of such propositions (or facts).

1.2.2 Justification

Another task of epistemology, which follows directly from the JTB definition of knowledge, is defining what a proper justification consists of. For every proposition we belief, we can ask ourselves '*how* do we *justify* this belief?' (rendering it knowledge if p were true). In our case we are especially interested in the question 'how does a computer simulation justify our belief in ω?' Can we perhaps reduce it to a familiar form of justification—like observation or mathematical inference? We will attend to these questions in chapter 3 and 4.

However, there are actually *two* 'levels' of justifying knowledge: first, there is the *justification of the belief*; and second, there is *the justification of the method* we use to justify a belief. For instance, say $p_S^{\mathcal{M}_{\text{seeing}}}$ where p is 'There is a pencil on the desk', S is a person with good visible abilities, and $\mathcal{M}_{\text{seeing}}$ would be 'Seeing the pencil on the desk'. Although there can be some discussion about whether or not $\mathcal{M}_{\text{seeing}}$ is a sufficient method of justifying p[8] the point here is that there is a *second* level of justification—namely justifying $\mathcal{M}_{\text{seeing}}$ itself. There is a reason why we accept $\mathcal{M}_{\text{seeing}}$ as a (component of) justification of p—for instance because the method has proven to be reliable.

When we talk about computer simulations we have to keep this distinction in mind: on the one hand we have the justification of ω which is $\mathcal{M}_{\text{CS}}$, and on the other hand we need to justify $\mathcal{M}_{\text{CS}}$ itself.

[4]See Gettier 1963.

[5]Imagine saying 'I know that it snows outside, but I don't belief it.' This can mean two things: either you do not *really* not belief it (you just say you do not to express your surprise of the fact that it snows); or you do not really know it.

[6]Imagine saying 'I know I am going to win this match.' without having any justification for that claim—would we qualify this as knowledge (even if I actually do win the match)? I do not think this would be plausible, since it would made guessing a case of knowledge every time the you guess correctly. What this justification exactly *implies* is a question open for debate.

[7]Imagine saying 'I know the earth is the center of the universe.' and let us suppose I have proper justification for this claim. It is commonly agreed that we cannot know a false fact. If we could it would entail that we could know both p and $\neg p$, and thus q since any proposition can be derived from a contradiction.

[8]However, we could easily add more methods of justification to the example.

1.3 Outline of this paper

Before we look at the epistemological issues concerning computer simulations, we will first need to have an idea about what a computer simulation actually consists of (chapter 2). We therefore will consider some semantic questions about models (§2.1) and try to identify the features of a simulation (§2.2). This will give us the ability to define a computer simulation (§2.3). We will find that computer simulations can be equation-based or agent-based; both will be discussed when we take a look at the epistemological issues (§3.1.1–2). Beside the two different issues, associated with the two different kinds of computer simulation, there are two issues that apply to all computer simulations: truncation errors and functions by approximation (§3.1.3–4).

In chapter 4 we will turn to some recent debates about the philosophy of computer simulations. One of these debates is about the claim that computer simulations give rise to *new* philosophical problems (§4.1). After a brief introduction and some comments on *general* issues (§4.1.1) I will argue that there are in fact new philosophical problems, but that these problems are less numerous than claimed by most advocates (§4.1.3). As a final point we will look at a recent claim that computer simulations are 'just' *opaque* thought experiments (§4.2). If so, this could mean that the philosophy of computer simulations is not novel after all. However, I will argue that computer simulations are similar to thought experiments *in use only* and differ at the level of epistemic justification.

Chapter 2

Models and simulations

2.1 Semantic considerations

The physical sciences (but other sciences as well) make extended use of models to get insight in their theories. But what exactly *is* a model and how does it relate to theory? Furthermore, what is the relationship between the model (a hypothetical system) and the system it represents (what is called the *target-system*)?

Let us examine a simple kind of model: throwing a die. We will introduce a function f such that $\langle f \rangle = 3.5$ and $f : \mathbb{R} \to \{1, 2, 3, 4, 5, 6\}$. Now we claim that f represents throwing a die: you can only get $1, 2, 3, 4, 5$ or 6 and the average of your throws will converge towards 3.5.

These results can be generalized. As pointed out by Frigg [2010] there are two steps to be distinguished: first I present a hypothetical system (the function f in the above), then I make the claim that the hypothetical system represents the target-system (the claim that f represents throwing a die). The second step needs justification, which we provided by pointing out that the relevant aspects of throwing a die are captured in the hypothetical system. The importance of this step was stressed by Guala [2002] as well: "[M]odels must be put at work by means of a 'theoretical hypothesis', stating that they stand in a certain relation [...] with real-world entities or systems."[1] Although there are some far more comprehensive theories of models (e.g. Frigg [2010]) we can suffice with the following definition:

> **Model**: a hypothetical system that is claimed to represent a certain target-system.

Although this does explain the relationship between hypothetical system en target-system, it does not tell us anything about the hypothetical system *itself*. What does a hypothetical system consist of? I would say it consists of 'theories' and 'assumptions', that is, from theory we build some hypothetical system and we adapt it to the target-system by making assumptions. According to Guala, the semantic view of theories "identifies theories with sets of models."[2] But this seems at odds with our definition of a model, for when we claim that the hypothetical system represents a certain target-system we defend this claim often on theory. If theory is a set of models, the defend would seem somewhat circular although this actually does not have to be the case—the relation between theories and models can be coherentistic in structure. In any case it would be far beyond the scope of this paper to investigate this issue, as Grüne-Yanoff and Weirich [2010] rightly noted: "The relation between models and theory is at least as problematic as that between models and simulation, but in contrast to the latter, the former has been the subject of an ongoing debate for the past 30 or so years."[3]

[1] Guala 2002, p. 61.
[2] Guala 2002, p. 60.
[3] Grüne-Yanoff and Weirich 2010, p. 24.

2.2 Simulation: features

Giving a unequivocal definition is difficult, perhaps impossible since many philosophers seem to have there own notion of 'simulation.' Since we are in no need of a sharp definition I will just discuss some features that I find essentially part of every definition of simulation.

Simulations and models are strongly linked. Nevertheless, as is mentioned earlier, the relation is problematic. As for their relation to theory "simulations in their construction are autonomous in the sense that they are not merely put together from theory and in that they function as investigative tools independently from theories."[4] This agrees with Humphreys remark that it is important "that the simulation is actually carried out on a concrete device,"[5] therefore resembling a investigative tool rather than abstract representations. According to Hartmann [1996] a simulation is a *dynamic* model: a model that evolves through time. This in contrast to a static model (e.g. a map). Another feature of simulations is that they are agent-dependent for "simulations are not in nature, it is us who 'see' them and often build them according to our purposes."[6] To sum up the features, simulations are

- investigative tools;
- dynamic models;
- agent-dependent.

Of course this is not a complete list, but it shows the key features of simulations that are the least controversial.

2.3 Computer simulation: a definition

A computer simulation is a simulation where *all* parts are carried out on a computer. Humphreys makes a distinction between the *core simulation* (which is essentially related to the computer) and the *representation* of the output (which is part of any simulation). In Humphreys [2004] he gives a definition of a computer simulation that we will adopt here:[7]

> System S provides a core simulation of an object 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.
>
> [...]
>
> When both a core simulation of some behaviour and a correct representation of the output from that core simulation are present, we have a full computer simulation of that behaviour.[8]

This definition nicely captures all the features a computer simulation should have. Since S is a concrete computational device, all parts are carried out on a computer; via a temporal process, which agrees with Hartmann's dynamic model. That the representation of B is either dynamically or statically does not mean that we can use a static model. It means that the *model* needs to be dynamically, but the *representation* can be either dynamically or statically. The most important part of the definition is that S produces 'solutions to a computational model.' However, we can distinguish between two types of computational models: *equation-based* and *agent-based*,[9] which I will each discuss below.

[4] Grüne-Yanoff and Weirich 2010, p. 24.

[5] Humphreys 2004, p. 109.

[6] Guala 2002, p. 62.

[7] This is an adaptation of his *working definition* from Humphreys [1990, p. 501]: "A computer simulation is any computer-implemented method for exploring the properties of mathematical models *where analytic methods are unavailable*."(my emphasis) We will NOT adopt this restriction—i.e. computational models can be based on analytically unsolvable *and* solvable equations.

[8] Humphreys 2004, p. 110–111.

[9] I owe this distinction to Grüne-Yanoff and Weirich [2010].

2.3.1 Equation-based models

Equation-based models "describe the dynamics of a target system with the help of equations that capture the deterministic features of the whole system."[10] This means that the model describes the *macro* properties of the system. Let $f(u, v, w)$ be an equation (analytically solvable or unsolvable) that represents (properties of) a target-system T with properties U, V, W. The equation-based simulation will yield values for these properties by solving f for given boundary conditions.

For example, consider the simple pendulum[11] that is showed in Fig. 2.1 where L is the length of the pendulum, ϕ the angle, m the mass and g the gravitational constant. We know from Newton's second law that $F_\phi = ma_\phi = mL\ddot{\phi}$ and we also know that $F_\phi = -mg\sin\phi$. The differential equation which represents the motion of the simple pendulum is therefore

$$\ddot{\phi} + \frac{g}{L}\sin\phi = 0. \tag{2.1}$$

This equation cannot be solved analytically without making the approximation $\sin\phi \approx \phi$. The example shows that an equation-based model consist of an equation that represents the target-system $\phi(t)$ (Eq. 2.1) with properties L and m, and boundary conditions ϕ_0 and $\dot{\phi}_0$. Although analytically unsolvable, Eq. 2.1 *does* specify the target-system at any future state.

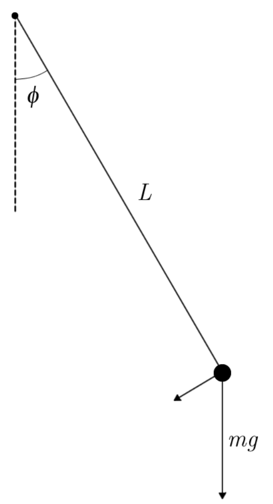

Figure 1: The simple pendulum

2.3.2 Agent-based models

The dynamics of a system can be *described* by a model (equation-based) or be *generated* by a model. In the latter case, the model describes the dynamics of the constituent parts of the system that, taken together, generate the dynamics of the system as a whole. These models are called 'agent-based' because they define behavioural rules that certain agents are bound to. An example of a computer simulation that is based on an agent-based model is the 'natural selection' of a given species. We can imagine a highly advanced computer simulation that simulates how a certain species evolves over time. The model of the simulation only defines a member of the species (what is preferable and what not, what a member needs to stay alive, etc.) and some rules of interaction between members (share food with relatives only, how to procreate, etc.). What is important is that the actual evolution of the system is *not* defined in this model (as was the case in Eq. 2.1 of

[10] Grüne-Yanoff and Weirich 2010, p. 30.

[11] For more background on this example I refer to Taylor [2005] chapter 1 (especially example 1.2 on pp. 30–33 and exercise 1.50 on p. 41) and chapter 5. Other examples of equation-based models can be found in exercise 2.43 on p. 79 and in exercise 5.39 on p. 212.

the previous section). The only way to see how the system develops is to run it.

The recognition of the distinction between equation-based and agent-based computer simulations is important if we want to grasp the opacity of the process. The computer does not only solve an equation (albeit incredibly fast), it solves *many* equations, simultaneously, and uses the output as input for different equations or even to alter equations. The propositions that we obtain from running an agent-based simulation can, if properly justified, give us knowledge that used to be completely inaccessible to us.

Chapter 3

Epistemological issues

We now face the challenge of providing proper justification for our belief in propositions yielded by computer simulations. In doing so one can distinguish two steps: the first is justifying the *modelling assumptions*—the models used in the simulation, the theory on which the models are built, the idealizations, etcetera. The second step is to justify the numerical methods—the work done by the CPU. We will focus on the second step here, since only this step is unique for computer simulations.

3.1 Epistemic opacity

The use of a computer to tackle numerical problems causes the process between input and output of the simulation to become opaque. Although we can often inspect a *single* step in the process, we are incapable of inspecting *all* the steps, reducing our understanding of the process. According to Humphreys a "process is essentially epistemically opaque to X if and only if it is impossible, given the nature of X, for X to know all of the epistemically relevant elements of the process."[1] This means that, since we cannot justify every step in the process, we have to justify the modules that make up the process. In other words, we have to justify the individual *modules*, instead of the individual elements.[2] Frigg and Reiss [2009] comment on the justification of numerical methods that "they are purely mathematical problems".[3] I agree that the justification of the numerical part of computer simulations is mostly mathematical, since it involves calculations and abstractions, but this does not mean that it is purely a mathematical issue. As I have noted above (and will again in chapter 4) the justification of using numerical methods (in computer simulations) is, in the end, a philosophical matter as it renders Ω genuine knowledge—i.e. it justifies $\mathcal{M}_{\text{CS}}$ which is a philosophical, epistemological, matter.

3.1.1 Module structure

At first sight the justification of $\mathcal{M}_{\text{CS}}$ looks very similar to that of a mathematical fact.[4] In the latter case we justify the knowledge by assuming the ability to provide a proper inference if one was pressed to do so. In the case of a computer simulation we cannot infer *every* step and so we must look for other means to justify the use of $\mathcal{M}_{\text{CS}}$. Fortunately we can rely on the deterministic logic of the computer:[5] we do not need to inspect all the steps if we can inspect all the modules. If a computer simulation does nothing else than

[1] Humphreys 2008, p. 8.

[2] A *module* is a (piece of) code or a (set of) command(s) that transform(s) a given input. A very simplistic example is the command 'add 2'. In general, given a module M, a set of inputs I and a set of outputs O we could write $M[I] = O$.

[3] Frigg and Reiss 2009, p. 602.

[4] I will discuss this similarity in more detail in chapter 4.

[5] I take for granted that given a set of instructions, a computer cannot do otherwise than follow these instructions.

applying a module M on a set of input I and we know $M[*]$ where $*$ is a dummy input, than we may say that we are justified in believing $M[I] = O$ even if we do not inspect every $M[I_i]$.[6] So the first step is to analyse the modules that make up the computer simulation to make sure there exist an O_i for every I_i. To be sure, we do not have to check *every* individual I_i, for this is impossible—-it suffices to inspect the structure of M for undefined values and other pitfalls.

3.1.2 The justification of agent-based models

We may be able to inspect the structure of M in simulations that are equation-based, but it is not yet clear whether we can inspect the structure of an agent-based model—and, more important, whether this results in proper justification at all. Imagine an agent-based model that simulates the economy of some fictional society (the particular details of the simulation are irrelevant). The model, for it being agent-based, has no module describing the *whole* system, but only modules that describe the *constituent* parts that *produce* the system once the simulation is run. For instance, there is a module that describes a human being (i.e. the relevant aspects) and a module that describes an economic good. However, there is no module that describes the society's economy.

In the case of a equation-based model we can inspect the modules A, B, C,... that together describe system S—i.e. $A + B + C = S$. Thus by inspecting the modules, we inspect the system. This differs from an agent-based model where the modules A and B produce a system S that is not simply the sum of the modules. Therefore, by inspecting A and B we do not inspect the (whole) system. However, it seems we cannot do more than inspect the modules we use (and hope for the best). When we consider the economy simulation, we might conclude, on the basis of that simulation, that scarcity of a product increases its value (call this ω). Can we say that we *know* ω (granted that it is true)? We need to be careful here, for we could have missed a relevant aspect that would have resulted in a different output. For example, we might not have included the spontaneous creativity of humans (which would change the outcome of the simulation to $\neg\omega$), as well as the human desire of status (which would change the outcome to ω again). This problem is, however, not specific to computer simulations but to simulations (and models) in general.

3.1.3 Truncation errors

A third and different kind of opacity, is the occurrence of so called 'truncation errors': errors that result from the storage of a finite number of digits. A computer can only store a finite number of digits in its memory and by doing so it automatically rounds off the output. To give an idea of this error consider the following modules $M_1 : x \mapsto \frac{x}{3}$ and $M_2 : x \mapsto 3x$ and the input $I = 1$. Now we apply $M_1[I] = 0.333 = O_1$ where the computer can only store three digits (or four if you count the 0). Next we apply $M_2[O_1] = 0.999$ which, of course, should return the original input value 1. Although the difference in this example is just 0.1% and a real computer can store a lot more digits, it does show how the storage of a finite number of digits can result in an error in the output. We also have to keep in mind that truncation errors will become even more vivid when non-linear modules or super-iteration modules, that contain huge amounts of iterated steps (whereby the output is used as input), are used.

Epistemically relevant is that truncation errors can cause scenarios similar to the infamous Gettier scenarios (Gettier 1963). Say we accept the use of a computer simulation as a reliable method for justifying our belief in ω. Consider a system S that simulates a target-system T on a computer. Let t and s be propositions that are derived from respectively T and S. (For instance we might consider a weather-simulation, S, and the

[6] There is already a pitfall we need to avoid, i.e. there needs to be a well-defined value for every $M[I_i]$. Consider the module that maps $x \mapsto \frac{1}{x}$ that is undefined for the input value 0. However, I think this is a matter of careful analysis of the modules one uses. It could become problematic if a computer simulation uses many modules that interact with each other and, thus, we do not know if every input element has a well-defined value in M. This could be the case when a model is agent-based.

actual weather, T). To illustrate the point, let s be a proposition *without* truncation error and let $\bar{s}$ be 'the same' proposition[7] *with* truncation error. It could now be the case that $t = \bar{s}$ and that $t \neq s$, which means that (if truncation error is not considered) we can know $\bar{s}$ on basis of S even if S is not a reliable system. For example, I might conclude on basis of S that 'it will rain today' (which it will) but I would have concluded 'it will snow today' (which it will not) if truncation errors were absent.

The problem we face is that *all* $\omega \in \Omega$ are based on variables and constants that suffer from truncation errors. We therefore either need to redefine our definition of knowledge or, perhaps less radical, we must specify a constraint on ω.

> **Constraint on** ω: we can justifiably belief ω iff S, the system ω belongs to, will also yield ω if *individually*, *some combination of*, and *all* of the variables and constants are slightly adjusted.

What is meant by *slightly adjusted* depends on the situation and is a problem for the mathematical sciences.[8]

3.1.4 Functions by approximation

There is a similar problem to that of truncation errors that is also invisible to most users of computer simulations. This is the problem of creating functions by approximation, which, to be sure, differs from the problems of error analysis and approximation methods in models. A function is not always available at the most fundamental level of a computer language. A processor can perform logical and arithmetical operations, but is not able to calculate trigonometric functions without using some kind of approximation techniques. An example of such a technique is Taylor expansion whereby a function f is represented by an infinite sum of terms that are calculated from derivatives of f (see Eq. 3.1).

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x - a^n). \tag{3.1}$$

Of course a computer can only calculate this sum to some finite term k such that there exists an error Δ between the *calculated* and *actual* answer. Another example, that is often used to calculate hyperbolic and trigonometric functions, is the so called CORDIC algorithm.[9] This technique makes use of iteration to create functions and is thus an approximation similar to the Taylor expansion.

To justify $\mathcal{M}_{\mathrm{CS}}$ we must make sure that Δ will not change ω. Therefore we need a second constraint on ω:

> **Constraint on** ω: we can justifiably belief ω iff S, the system ω belongs to, will also yield ω if *individually*, *some combination of*, and *all* of the functions are expanded to a higher degree.

How this is done is, again, a problem for the mathematical sciences.

3.2 Conclusion

The epistemological issues we discussed in the previous section all had in common that they caused the epistemic justification to become opaque. This epistemic opacity, as put forward by Humphreys, is claimed to be a novel feature of computer simulations and therefore asks for a new epistemology (this claim is defended in the next chapter). Before we proceed, however, we must say something about the different meanings of the notion

[7]Of course s can differ from $\bar{s}$, which is crucial for making the point. What I mean is that s and $\bar{s}$ are about the same proposition t. This leads to $(s \wedge \bar{s}) \rightarrow (s = \bar{s})$.

[8]If $N \in \mathbb{N}$ and N_i is the ith digit of N, then we must adjust $N_n \rightarrow N_n + 1$, where n is the last digit that is stored.

[9]See Volder 1959.

'epistemic opacity'. For we have been using this notion in a broad way and there is a slight difference between Humphreys idea of opacity and mine. To be more precise, we have been using 'epistemic opacity' to designate two different phenomena.

The first designation is similar to Humphreys's and entails that something is epistemically opaque to X if it is impossible for X to know all the relevant steps involved (see the exact definition at the beginning of §3.1). This opacity has its origin in the nature of X, or, perhaps more correct, in the discrepancy between the cognitive ability of X and the calculating ability of the computer. The consequence of this discrepancy is that we are not able to inspect *all* the relevant steps of the process and, thus, cannot rely on traditional justification methods. We used this notion of 'epistemic opacity' when we discussed module structures (§3.1.1) and agent-based models (§3.1.2).

We also used 'epistemic opacity' to point out the opaque background steps that are not visible to the end-users. This opacity is not a consequence of the nature of X but rather of the nature of the computer. The steps are opaque as far as they are not part of the simulation itself but belong to the simulating device. We used this notion of opacity when we discussed truncation errors (§3.1.3) and functions by approximation (§3.1.4). It is worth noting that this is NOT a problem that belongs to the sciences but to philosophy (although mathematical tools may be needed). We must distinguish between fundamental properties of the simulating device and properties of the simulation itself. The discussed issues are all fundamental properties of the computer—unlike, for example, measuring errors or approximation methods *within* the model.

Chapter 4

Recent debates

4.1 The novelty question

We have already seen that computer simulations have a huge impact as a scientific tool. It extends our ability to calculate and visualize properties of, and relations between entities dramatically. It may come as no surprise that computer simulations give rise to a number of new problems for scientist to deal with. One problem is the effective use of processor resources; another question is what numerical methods can provide the desired accuracy without consuming too much CPU power. These problems are new, genuine difficulties that need the proper attention of scientists. However, they are not philosophical problems and therefore of little interest to us now. We have focussed on epistemological issues concerning computer simulations and we can in fact ask ourselves the question: are the epistemological problems coined by computer simulations *novel* problems?

Although the majority view is that computer simulations *do* raise new epistemological questions (see for example Humphreys [2004], Humphreys [2009] and Lenhard and Winsberg [2010]) there are philosophers who defend the opposite view. In a recent article Frigg and Reiss [2009] argue that *all* philosophical questions—i.e. metaphysical, methodological, semantic and epistemological—concerning computer simulations are instances of more general discussions in the field of philosophy of science. "[W]e agree that simulations introduce something new and exciting into *science*, but we doubt that this requires us to rewrite the *philosophy of science*."[1]

However, I agree with Humphreys that "computational science introduces new issues into the philosophy of science"[2] and in this section I will defend this claim against its rivals. Hereby I will focus, again, on the *epistemological* issues (some of these issues have already been discussed in chapter 3). Since some of the arguments come from misunderstanding the essentially *computer* simulation part, I will first discuss some features of computer simulations that *do not* raise novel philosophical questions.

4.1.1 General issues

Modelling assumptions

Although, as we have seen in chapter 2, there is no unequivocal definition of a computer simulation, we *can* distinguish between the more general 'simulation' features and the essentially 'computer' features. Examples of simulation features are modelling assumption and representation between simulation and target-system. These features are present in *any* simulation and even in other parts of scientific tools like models. Thus they do not raise any new philosophical problems when present in computer simulations.

Frigg and Reiss [2009] criticise the novelty of questions relating modelling assumptions as well—not only for computer simulations, but for simulations in general. They discuss

[1] Frigg and Reiss 2009, p. 595, original emphases.
[2] Humphreys 2009, p. 615.

the claim made by Winsberg [2001] that the epistemology of simulation is distinct because it is *downward*, *autonomous* and *motley*.[3] They conclude that "[t]he conclusion that a different epistemology is needed could have been reached by studying the practice of much of science apart from computer applications."[4] And although there are novel and simulation specific questions concerning computational models, "they belong [. . .] to the class of problems that arise in connection with complex models in general[.]"[5]

Kludging and path-dependency

So far, we have seen that our inability to keep up with the computer, truncation errors, and functions by approximation are the origin of the epistemic opacity (chapter 3). There is, however, yet another source of opacity that goes by the name 'kludging.' The word has several proposed etymologies[6] and is described in the dictionary as

> **Kludge** *Computer Slang*
> a software or hardware configuration that, while inelegant, inefficient, clumsy, or patched together, succeeds in solving a specific problem or performing a particular task.[7]

Where a theory must be fundamental, eloquent and simple—a computer simulation must simulate a process with the available resources[8] and in a given time. This often forces software engineers to build inelegant, but functional, pieces of program. Kludging can be path dependent as is noted by Lenhard and Winsberg [2010]: "A kludge is built to optimize the performance of the overall model as it exists at that particular time, and with respect to the particular measures of performance that are in use right then."[9]

However, the opacity caused by kludges differs not fundamentally from the opacity caused by complex designed experiments. Any advanced, complex experiment is designed to be effective and is optimized according to the requirements and available methods of a particular time. Furthermore, building and developing an experiment is as path-dependent as a computer simulation. Both depend strongly on the choices of the designer. Therefore I do not think kludging will raise epistemological issues that did not already emerge from the philosophy of experiments.

4.1.2 New issues

There are, however, more specific problems that seem to belong to computer simulations alone. They obviously belong to the 'computer' part and lean strongly on the distinct feature of computers, namely the incredible computational power. I will start with a feature of the epistemology of computer simulations that is defended by Humphreys.[10]

Dehumanizing epistemology

As a result of introducing computational methods into the sciences humans are pushed away from the centre of scientific practice. Humphreys makes a distinction between a) a *hybrid* scenario where science is carried out partly by machines; and b) an *automated* scenario where science is entirely automated. We are currently in the hybrid scenario and in this situation, "the representational devices, which include simulations, are constructed to balance the needs of the computational tools and the human consumers."[11] According to Humphreys, "[c]omputational science introduces new issues into the philosophy of science

[3]See for a detailed discussion Frigg and Reiss 2009, p. 598–601.
[4]Frigg and Reiss 2009, p. 601.
[5]Frigg and Reiss 2009, p. 601.
[6]See for some examples http://en.wikipedia.org/wiki/Kludge (2 February 2011).
[7]http://dictionary.reference.com/browse/kludge (2 February 2011).
[8]Resources are CPU power and memory, but also include the source code, programming language, CPU architecture, etc. . .
[9]Lenhard and Winsberg [2010, p. 257].
[10]See Humphreys 2004 and Humphreys 2009 for a more comprehensive discussion.
[11]Humphreys 2009, p. 617.

because it uses methods that push humans away from the centre of the epistemological enterprise."[12] This is not so much a new issue in the current epistemology as it is a new epistemology altogether.[13]

Epistemic opacity

We have already encountered epistemic opacity in chapter 3 as an epistemological issue concerning computer simulations. To recap: "[a] process is essentially epistemically opaque to X if and only if it is impossible, given the nature of X, for X to know all of the epistemically relevant elements of the process."[14] This led to the epistemological problem of justifying $\mathcal{M}_{\mathrm{CS}}$.

Why is this a *novel* epistemological problem? According to Humphreys this is because "prior to the 1940s, theoretical science had not been able to automate the process from theory to applications in a way that made the details of parts of that process completely inaccessible to humans."[15] In no other part of epistemology is there an opacity of the justification due to our inability to follow *all* the subsequent steps. This raises a new and important epistemological problem: if ω is a proposition based on a computer simulation, how can we *know* ω? How can we justify our belief in ω when we do not understand all the subsequent steps that led to ω? That is, how can we justify $\mathcal{M}_{\mathrm{CS}}$, the method we use to justify ω?[16] As is noted by Humphreys, epistemic opacity "is not mentioned by Frigg and Reiss"[17] at all.

We can nevertheless legitimately ask whether knowing ω is principally different from knowing a scientific or mathematical fact for which the knowing subject cannot provide an inference—thus being opaque. For example, consider the following well known mathematical fact:

$$x_{\pm} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \tag{4.1}$$

as a solution to an equation of the form $ax^2 + bx + c = 0$. Although we know Eq. 4.1 we probably cannot infer it right away;[18] thus we cannot justify our belief in Eq. 4.1 on the self-evidence of the individual steps of the inference. This problem of justification seems similar to the problem of justification when using a computer simulation. In the case of Eq. 4.1 the method for justifying our belief is $\mathcal{M}_{\mathrm{R}}$, *remembering*; we know Eq. 4.1 because we can remember that we proved it once or that we can prove it if we took the effort. In the case of $\omega_S^{\mathcal{M}_{\mathrm{CS}}}$, where $\mathcal{M}_{\mathrm{CS}}$ is *using a computer simulation*, we know ω because a computer simulation is a reliable method. In both cases we know some fact p on the basis of $\mathcal{M}$ that can only be considered genuine knowledge insofar we can properly justify $\mathcal{M}$ itself.

[12]Humphreys 2009, p. 616.

[13]While this is an interesting thought, I do not agree with Humphreys that an automated science will push humans away from the centre of epistemology. The main point is that I find it hard to imagine a non human epistemology. Machines can, of course, perform tasks and, in some cases, this will be scientific tasks. But to say that, therefore, a machine does *science*—even if the task or the classes of tasks are generated by the machine itself—is to undervalue the human role in the scientific enterprise. In terms of knowledge, we need a 'knowing subject', i.e. a subject that we can subscribe a *belief* to; I do not think we can properly subscribe a belief to a machine and, thus, it seems rather odd to talk about a non human epistemology. It would be a different story if the machine was a rational agent—a human epistemology can make the shift to a more general *rational* epistemology. Nevertheless, I do not think we can speak of 'rational machines' as of yet, neither do I think this was the idea Humphreys had in mind. He sees the dehumanization in the shift of scientific work from human to machine. We disagree whether this implies a new epistemology is needed. Unfortunately, to defend my views on these matters in more details goes beyond the scope of this paper.

[14]Humphreys 2008, p. 8.

[15]Humphreys 2009, p. 618.

[16]As is mentioned in chapter 1, there are *two* levels of justification involved when it comes to knowledge. First of all there is the *justification of the belief*, and secondly, there is *the justification of the method* we use to justify a belief.

[17]Humphreys 2009, p. 618.

[18]If you can, it would not be hard to come up with a different example that you cannot infer but nevertheless do know.

While both $\mathcal{M}_{\mathrm{R}}$ and $\mathcal{M}_{\mathrm{CS}}$ are reliable methods of justification, the underlying justification of the methods *themselves* is very different. We justify $\mathcal{M}_{\mathrm{R}}$ (remembering a mathematical fact to be true) because we suppose that, in principle, we could go through the inference. The fact that we *do not* is because we find it time-consuming, not because we *cannot* do it.[19] This is not the case for $\mathcal{M}_{\mathrm{CS}}$ where we cannot go through the steps that led to ω because, and this is important, *there are just too many.* Although this does not look like an 'in principle' different, it is more than just a practical difference; I think *we humans* cannot, in principle, go through *all* the steps that led to ω. We do not get old enough or we lack the cognitive abilities.

At first glance this may look odd—how does it come that age can make the inference of a mathematical fact principally different from the inference of ω? However, this is exactly why traditional methods *do not* suffice for computer simulations; they presuppose the ability to go through the inference (however impractical they presume it can be done in a life-time). Where we should draw the line between impractical and not possible is a question open for debate, but it is not relevant for our current discussion. What matters is that, with the emergence of computer simulations, we crossed the line.

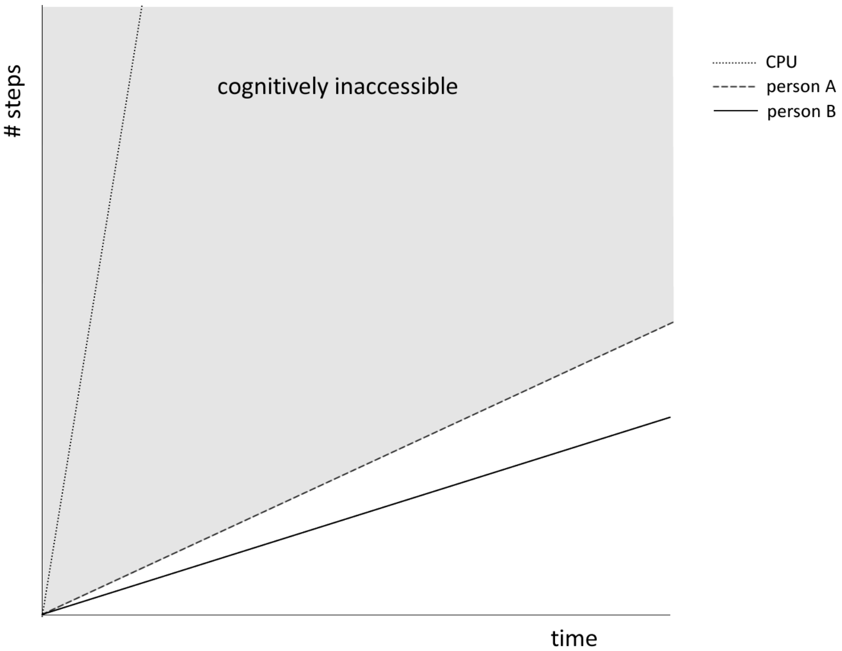

Figure 2: The availability function

Fig. 4.1 shows the number of steps as a function of the time for two persons with different cognitive abilities, and a CPU. Although both persons have a different cognitive limit—due to cognitive abilities and/or lack of time—the grey area is inaccessible to *all* humans.

While this can seem obvious it may be worth emphasizing that the cognitive limit is not a *fixed* limit and is *person-dependent.* It depends on the steps available (SA) to infer p as well as the time available (TA) to do so. This can be written for a person s as an 'availability' function $a(s,t) = \frac{\mathrm{SA}(s)}{\mathrm{TA}(s)}t$. In Fig. 4.1 $a(B) < a(A) < a(\textsc{cpu})$. The cognitive ability of s can be written as $\mathrm{CA}(s) = \dot{a}(s,t) = \frac{\mathrm{SA}(s)}{\mathrm{TA}(s)}$. In Fig. 4.1 this means that $\mathrm{CA(B)} < \mathrm{CA(A)} < \mathrm{CA}(\textsc{cpu})$, and, more general, that $\mathrm{CA}(s) < \mathrm{CA}(\textsc{cpu})$ for any person s.

A last remark must be made for we have presupposed something without making it explicit. We have accepted as a method of justifying mathematical factual knowledge the requirement of being able to reproduce the inference. We said that $\mathcal{M}_R$ was a valid

[19] Although consulting a book to guide you through the steps may be necessary.

method because we could provide the proof if we took the effort. This, however, is not the only available justification of $\mathcal{M}_R$—a different requirement can be that *society as a whole* can provide the inference. This can mean two things: either that the justification 'belongs' to society (in a kind of inter-subjectal manner), or, that we must not compare $a(\text{CPU})$ with $a(s)$ but with $\sum_{s\in S} a(s)$, the sum over all people of a society. If we take the first interpretation, it would dissolve the similarity with computer simulations altogether, as computer simulations are not inter-subjectally available to a society. For the second interpretation, I think the only way out is to posit that $\sum_{s\in S} a(s) < a(\text{CPU})$.

However, I do not think this necessary leads to serious problems. Firstly, the above mentioned (social) justification of $\mathcal{M}_R$ is itself challenged by the (individual) justification we tacitly applied at the beginning of this section. So at least for those subscribing to the individual justification computer simulations are in need of a novel justification. Secondly, the posit that $\sum_{s\in S} a(s) < a(\text{CPU})$ is a realistic one, considering the huge calculating power of today's processors. If we take in mind the incredible technological advances that are being made, we can safely say that even if $\sum_{s\in S} a(s) > a(\text{CPU})$ at the moment, the inequality sign will reverse in the near future.

4.1.3 Conclusion

A few concluding remarks need to be made. First of all we must not exaggerate the amount of new epistemological problems that have arisen from computer simulations. Most of the issues—e.g. kludging—are instances of more general discussions in the philosophy of science. Others, like the dehumanization of epistemology, are still open for debate.

A second point is that we must not overstate the novelty of the few new epistemological problems that *have* arisen. We have seen in §3.1 and §4.1.2 that a slight adjustment of familiar methods might suffice for the epistemology of computer simulations. Epistemic opacity, although new and in need of a adaptation of epistemology, does not necessarily cause a *radical* epistemic paradigm shift. This position is in between Frigg and Reiss's and Humphreys's and can therefore be called the 'moderate novelty claim.'

4.2 Thought experiments: a comparison

There are philosophers who compare computer simulations with thought experiments (see for example Di Paolo 2000). Both, they claim, "are tools with which to explore the consequences of a theoretical position."[20] If this is so, the epistemology of computer simulations might not be novel after all. However, as I will argue, this similarity is in *use* only and there are some differences in epistemology.

When we think of a computer simulation as a process that yields some output for a given input, we are tempted to conclude that "it is ultimately just a computer program that rearranges symbols in a logical fashion, and as such cannot arrive at new knowledge."[21] That is, a computer simulation only elucidates what is already enclosed in the input. We might notice that this epistemological problem is very similar to the case of thought experiments. Norton [2004] regards thought experiments as "draw[ing] upon what we already know of it"[22] since "[a]ll pure thought can do is transform what we already know."[23,24]

[20] Di Paolo 2000, p. 1.
[21] Di Paolo 2000, p. 1.
[22] Norton 2004, p. 45.
[23] Norton 2004, p. 49.
[24] Although this view is the most common, there are philosophers who adhere a different position. Brown [2004], for instance, defends a platonic view of thought experiments: "we can have a kind of perception, an intuition, of abstract entities."[Brown, 2004, p. 32] Thought experiments transcend empiricism because it uses this intuition to "intuit laws of nature as well."[Brown, 2004, p. 34] Even though this debate is interesting in itself, we will ignore views like Brown's since if they are correct, the similarity between thought experiments and computer simulations would not been made in the first place.

Nevertheless, there are two differences between the epistemology of computer simulations and that of thought experiments that I would like to point out. They belong to *justification* and *closure*. We have to keep in mind that the comparison is between p_S^{TE} en p_S^{CS}—i.e. a person S who uses a thought experiment or a computer simulation to justify his belief in p.

4.2.1 A difference in justification

The process of a computer simulation and a thought experiment might look similar—both proceed through logical steps from input to output—but, as is already mentioned and extensively discussed in the previous chapter, the process of a computer simulation is to a large extent inaccessible to us. This is exactly why computer simulations in Di Paolo [2000] are identified as *opaque* thought experiments.

Even though the opacity does not affect the *use* of both tools—typically we are only interested in the input and the output—it does affect the (epistemological) *justification*. In the case of thought experiments every step is completely evident, i.e. transparent; and therefore the justification of the process is essentially the justification of the individual steps. This can be done by traditional methods as induction, deduction or abduction.

The epistemic opacity of computer simulations delivers us novel problems concerning justification, as we have seen and discussed in the preceding chapter. The main issue is the inability to destruct the process in individual steps. I think this rescinds the similarity, at least epistemologically. A thought experiment is necessarily transparent, for its function is to show that some premise P follows from some antecedent A: its epistemological function is to provide a justification for concluding P from A, and this is precisely what is absent from a computer simulation.[25] The function of a computer simulation is *also* to show that A follows from P, but the justification of $\mathcal{M}_{\text{CS}}$ differs from the the justification of $\mathcal{M}_{\text{TE}}$. This means that *once justified* the *use* of both thought experiments and computer simulations is similar.

One might object that the difference that we just pointed out is covered by the adjective 'opaque'. I think we can answer to this objection in two ways: 1) a thought experiment is, as mentioned before, necessarily transparent which makes the notion of an 'opaque thought experiment' a contradiction in terms; 2) a opaque thought experiment, if not a contradiction, looses one of its most characteristic features, viz. transparency. Granting that computer simulations are opaque thought experiments, we could still argue that this does not imply that no new epistemological questions can be raised.

4.2.2 A difference in closure

There is another difference that is strongly linked to the opacity of the process of computer simulations. If we accept closure under known implication—which is a plausible although contested claim (see for a discussion Dretske 1970 and Nozick 1981)—computer simulations, in contrast to thought experiments, *can* yield us new knowledge.

I do not agree with Norton when he says that "pure thought cannot conjure up new knowledge"[26] but I think he is right when we replace 'pure thought' with 'operations of the mind.' The difference lies in the fact that I think pure thought, i.e. the mind, *can* conjure up new knowledge[27] but not when used to operate on existing knowledge. To illustrate this point, consider a set of initial facts F and the set of logical operators LO. If $f_1, f_2 \in F$ en $L \in LO$ then we can perform $L(f_1, f_2)$ which can yield some fact $f' \notin F$.[28] Now, why is f' not considered *new* knowledge? Let us call the set of our knowledge elements K, and let $F \subset K$. Since f' was not considered new knowledge $f' \in K$, or, which amounts

[25] That is, a thought experiment *provides* a justification, while a computer simulation is *in need of* one.

[26] Norton 2004, p. 50.

[27] There are some facts that seem to spring from pure thought alone. As an example, consider the thought 'I think' or any equivalent self-aware thought. It does not appear to originate in experience (if this seems mistaken, try to pin it to one of the senses) but from somewhere else; the mind(s-eye), the self, ... Be that as it may, the key point is to separate the 'source' function from the 'operating' function.

[28] Of course f' could be an element of F but not *necessarily* so.

to the same thing, $L(f_1, f_2) \in K$. Since we can carry out $L(f_1, f_2)$, *which is justified by self-evidence*, we can say that if $f_1, f_2, L \in K$ then $L(f_1, f_2) \in K$. It is important to stress that the reason that this if–then statement is valid is that the application of the operator L is self-evident.

This is precisely what differs in the case of a computer simulation: the operations of the processor are not self-evident due to the opacity. This means that, for a non self-evident operator $\bar{L}$, if $f_1, f_2, \bar{L} \in K$ this does not imply $\bar{L}(f_1, f_2) \in K$ and, consequently, that the output of a computer simulation *can* be considered new knowledge. Di Paolo [2000] write that "phenomena can never be *discovered* through building simulations because these are always based on existing theoretical knowledge[.]"[29] I think this view is too simplistic; on the contrary, the computer simulation *dis*covers in the literal sense of the word. It makes visible what was previously hidden or inaccessible to us.

4.2.3 Conclusion

Although computer simulations and thought experiments have a lot of similar features, they differ in *justification* and, when accepting closure under known implication, *closure*. The epistemological difference between computer simulations and thought experiments is caused by the epistemic opacity; however, as we argued, the addition of the adjective 'opaque' does not close the gap. Thus we can conclude that computer simulations are opaque thought experiments *in use only*.

[29]Di Paolo 2000, p. 6, original emphasise.

Chapter 5

Conclusion

We can now draw some general conclusions about the epistemology of computer simulations. The aim of this paper was twofold: on the one side to provide an introduction to the topic and to discuss the epistemological issues; on the other side to critically discuss two recent debates, both centred around the novelty claim. The first aim was mainly about the justification of $\mathcal{M}_{\text{CS}}$, the second was broadly about showing the novelty of $\mathcal{M}_{\text{CS}}$.

A few concluding comments. First of all, we have seen that although there is no unequivocal definition of 'model' and 'simulation', we were able to define a computer simulation and its characteristic features. The foremost property is of course the calculating power, causing epistemic opacity. We also observed a translucency between some fundamental computing device properties and the end-user interface. This resulted in two constraints on ω such that Gettier-like scenarios are avoided.

A large part of the paper was devoted to defending the novelty claim, i.e. that the epistemological problems related to computer simulations are in need of a *new* philosophy. We can now conclude that computer simulations *do* raise new issues in epistemology—though a few remarks need to be made: first of all, we need to separate the problems related to the core simulations (the computer part) from the problems associated with simulations. Secondly, there were some general issues concerning the core simulation (e.g. kludging) that are very similar to other problems in the philosophy of science. Further investigation must be made. More needs to be said on the philosophy of simulation and the philosophy of models; also, the argumentation against Humphreys dehumanization of epistemology needs to be further developed.

Furthermore, the novelty of the epistemology of computer simulations is sometimes overblown (dehumanizing epistemology, radical epistemic shifts,. . .). I would therefore propose a 'moderate' position that holds that *the method of using a computer simulation to obtain knowledge needs a justification that is not available from similar methods in epistemology,* however, *the epistemology of computer simulations is not radically different from the current epistemology and will not cause a shift in epistemology.*

Bibliography

Brown, J. R. (2004). *Why Thought Experiments Transcend Empiricism*, chapter 1, pages 23–43. Blackwell Publishing Ltd.

DeRose, K. (2009). *The Case For Contextualism*. Oxford University Press.

Di Paolo, E. A. e. a. (2000). Simulation models as opaque thought experiments. In et al., M. A. M., editor, *Proceedings of the Seventh International Conference on Artificial Life*, pages 497–506. Cambridge: MIT Press.

Dretske, F. (1970). Epistemic operators. *The Journal of Philosophy*, 67:1007–1023.

Frigg, R. (2010). Models and fiction. *Synthese*, 172:251–268.

Frigg, R. and Reiss, J. (2009). The philosophy of simulation: hot new issues or same old stew? *Synthese*, 169:593–613.

Gettier, E. (1963). Is justified true belief knowledge? *Analysis*, 23:121–123.

Goldman, A. I. (1976). What is justified belief? In Pappas, G., editor, *Justification and Knowledge*, pages 1–23. Dordrecht: D. Reidel.

Grüne-Yanoff, T. and Weirich, P. (2010). The philosophy and epistemology of simulation: A review. *Simulation Gaming*, 41:20–50.

Guala, F. (2002). Models, simulations, and experiments. In Magnani, L. and Nersessian, N., editors, *Model-Based Reasoning: Science, Technology, Values*, pages 59–74. Kluwer Academic/Plenum Publishers.

Hartmann, S. (1996). The world as a process. In Hegselmann, R., Mueller, U., and Troitzsch, K. G., editors, *Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View*, pages 77–100. Kluwer Academic Publishers.

Hawthorne, J. (2004). *Knowledge and Lotteries*. Oxford University Press.

Heymann, M. (2010). Understanding and misunderstanding computer simulation: The case of atmospheric and climate science - an introduction. *Studies in History and Philosophy of Modern Physics*, 41(3):193–200.

Humphreys, P. (1990). Computer simulations. In *Proceedings of the biennial meeting of the Philosophy of Science Association*, pages 497–506.

Humphreys, P. (2004). *Extending ourselves: computational science, empiricism, and scientific method*. Oxford University Press.

Humphreys, P. (2008). ZiF Mitteilungen. Technical report, Zentrum für interdisziplinäre Forschung.

Humphreys, P. (2009). The philosophical novelty of computer simulation methods. *Synthese*, 169(3):615–626.

Lenhard, J. and Winsberg, E. (2010). Holism, entrenchment, and the future of climate model pluralism. *Studies in History and Philosophy of Modern Physics*, 41(3):253–262.

Norton, J. D. (2004). *Why Thought Experiments do not Transcend Empiricism*, chapter 2, pages 44–66. Blackwell Publishing Ltd.

Nozick, R. (1981). *Philosophical Explanations*. Cambridge MA: Harvard University Press.

Taylor, J. R. (2005). *Classical Mechanics*. University Science Books.

Volder, J. E. (1959). The CORDIC trigonometric computing technique. *Electronic Computers, IRE Transactions on*, EC-8:330–334.

Winsberg, E. (2001). Simulations, models, and theories: Complex physical systems and their representations. *Philosophy of Science*, 68:442–454.