“Materiality arguments” around the machine

From the program verification debate to the debates about computer simulations empiricity

Franck Varenne
Associate Professor of Philosophy of Science
University of Rouen & ERIAC (Normandy, France)
Research Associate at IHPST (Paris)
Two observations

1. A persistence of similar “materiality arguments” in different philosophical debates regarding the epistemic power - and limits - of computer programs and computer outputs.

2. Some of them are remarkably parallel or successive but few are explicitly related to each other.
   - In particular:
     - Most philosophers of computer simulations do not quote Fetzer nor any of his opponents
     - Although most of them use - or refer to - a similar “materiality argument”
     - What’s more: some of them even use as explicit argument the existence of “program verification” techniques
Research hypothesis

• Such persistence and multiple occurrences of similar “materiality arguments” around the digital computer in different philosophical contexts suggests that something really central - and difficult to think at the same time - is here at stake.

• Hence, it could be informative and revealing to carefully analyze and characterize the varieties of forms of these persisting “materiality arguments” from a comparative standpoint.

• This comparative approach, more than a selective or direct one, could help us to have a more-fine grained analysis by crossing different approaches of the same puzzle as much as different ways of coping with it, i.e. the puzzling hybrid nature of the programmed computer as an epistemic instrument.
Research question

• What can we learn about "what a machine and a program are?" (ANR PROGRAM scope) from a comparative analysis of:
  – the persistence
  – the similarities
  – the differences

between these “materiality arguments”? 
Outline

I. Methodological preamble
   I. Method, scope and feasibility of this study
   II. Broad characterization of what I will call a “materiality argument” in this context

II. The materiality argument in the “Formal Verification” debate

III. The materiality argument in the debate about the empiricity of computer simulations

Conclusion
I. METHODOLOGICAL PREAMBLE
I. Methodological preamble

Method, limited scope and feasibility of this study

• **Method**: historical, comparative and philosophical (interpretative)

• **Limited scope**:
  - Problem: a huge number of publications is related to each of the two debates
  - Hence: a choice of publications (the most significant, the turning-point ones)
  - Advancement: today, I will present only the first developments of this comparative research

• **Feasibility**, i.e. compatibility between this method and my philosophical position:
  - My intellectual position toward these debates is external if not completely neutral: my claim about the empiricity of computer simulations is that it does not essentially rely on the consideration of materiality (Varenne, *Du modèle à la simulation informatique*, 2007; Phan & Varenne, “Agent-Based Models…”, 2008; Varenne, “Chains of Reference in Computer Simulations”, 2013; Varenne, *From Models to Simulations*, Routledge, 2018, Chap. 8 and “Conclusion”)

  - Up to now, this position has been quite rare, but it is spreading among “computer science minded” philosophers of science. It is not far from other positions which are developed by Juan Duran (*Computer Simulations in Science and Engineering*, Springer, 2018).
I. Methodological preamble

Broad characterization of a “materiality argument”

• A “materiality argument” is:

any argument that

[in a

• context of a debate about the nature or about the degree of the epistemic power
• of a device containing - or of a technical or scientific inquiry making use of - at least one programmed digital computer]

refers to the materiality of this computer

– Variant of the term “materiality”: “physicality”

– Directly derived argument: “causality”
II. THE “MATERIALITY ARGUMENT” IN THE “FORMAL VERIFICATION” DEBATE
II. The “materiality argument” in the “formal verification” debate

Hoare’s claim

• “Computer programming is an exact science in that all the properties of a program and all the consequences of executing it in any given environment can, in principle, be found out from the text of the program itself by means of purely deductive reasoning.”

II. The “materiality argument” in the “formal verification” debate

Fetzer’s claim

- “The formal approach to program verification appears to trade upon an equivocation. While it is indeed possible to construct a formal proof of program correctness with respect to an abstract model of a machine, the significance of that proof depends upon the features of that model. When the model is an abstract model (the axioms of which can be given as stipulations) that does not represent a specific physical machine, a formal proof of correctness can then guarantee that a program will perform as specified (unless mistakes have been made in its construction). Otherwise, it cannot.”

II. The “materiality argument” in the “formal verification” debate

• Some reminders and comments:
  – The rare violence of this debate mainly was due to “bad faith” in each parts
  – At the beginning, particularly in (Fetzer, 1988), it seemed as if he wanted to deny any interest or relevance to formal program verification techniques. But he was ambiguous (perhaps voluntarily to make the “buzz”). And this attitude understandably caused violent reactions.
  – Fetzer finally but reluctantly (1991, 1998) rejoined a more balanced standpoint (that was trivial right from the start according to many) which, in the meantime, had been more clearly expressed by Jon Barwise (1989):
    • Formal program verification techniques surely give more evidence and increase our reliance in the good functioning both of the formal program and of the programmed material computer.
    • But it is true that the sole formal verification of the program as abstract construct doesn’t permit to completely and a priori guarantee the good functioning of any material device comprised of this verified program - once loaded – together with its implementing hardware.
    • So testing is useful also and cannot be completely replaced.
  – Even Hoare himself later made some amends (2009):
    • “My basic mistake [in 1969] was to set up proof in opposition to testing, where in fact both of them are valuable and mutually supportive ways of accumulating evidence of the correctness and serviceability of programs”, Communications of the ACM, 52(10), October 2009, p. 31.
II. The “materiality argument” in the “formal verification” debate

• The role of materiality in the final Fetzer’s theses (Fetzer, “Philosophical Aspects of Program Verification”, 1991, pp. 199-200):

Analogies with pure mathematics (Fetzer 1991 interpreting Hoare 1969)

<table>
<thead>
<tr>
<th>Objects of inquiry</th>
<th>Mathematics</th>
<th>Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theorems</td>
<td>Programs</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods of inquiry</th>
<th>Mathematics</th>
<th>Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proofs</td>
<td>Verifications</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 1 A plausible analogy
formal proof of program correctness \(\cong\) formal proof of theorem

<table>
<thead>
<tr>
<th>Domain</th>
<th>Mathematics</th>
<th>Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Premises</td>
<td>Input</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Mathematics</th>
<th>Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rules of inference</td>
<td>Programs</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Range</th>
<th>Mathematics</th>
<th>Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theorems</td>
<td>Outputs</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2 A more plausible analogy
“programs can be viewed as functions from initial states (inputs) to final states (outputs)”
II. The “materiality argument” in the “formal verification” debate

• An alternative comparison (Fetzer, 1991):
  
  mathematical proofs / scientific theories / programs

• and two working arguments
  
  – Semantical significance (in the sense of a **material**, non formal, semantic)
  – Materiality (along with causal capability)

• 1. Programs differ from Theorems:
  “Computer programs seem to differ from mathematical theorems insofar as they are intended to possess a **semantical significance** that mathematical theorems (within pure mathematics) do not possess, a difference arising because programs, unlike theorems, are **instructions for machines**.”


• 2. Programs and Scientific theories are similar in their semantics (this analogy permits to implicitly establish the semantical considerations on some material - non formal - semantic, contrary to Hoare)
  “The specific commands that constitute a program, for example, **stand for** corresponding operations by means of **computing machines**, while the generalization that constitute a theory **stand for lawful properties of the physical world**”


• 3. But Programs differ from Scientific theories in nature:
  “Yet even scientific theories [classical mechanics, special relativity, quantum mechanics], do not possess the causal capabilities of computer programs, which can affect the performance of those machines when they are **loaded then executed**.”

II. The “materiality argument” in the “formal verification” debate

• The alternative table (Fetzer, 1991):

<table>
<thead>
<tr>
<th></th>
<th>Mathematical Proofs</th>
<th>Scientific Theories</th>
<th>Computer Programs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntactic Entities</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Semantic Significance</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Causal Capability</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Fig. 3 A more general comparison
II. The “materiality argument” in the “formal verification” debate

• “[Barwise, 1989] claimed that I had failed to sufficiently differentiate ‘programs’ as (abstract) types and ‘programs’ as (causal) instances, a conception fundamental to my position. I therefore responded to his objection by distinguishing programs-as-texts (unloaded) from programs-as-causes (loaded), where (human) verification involves the application of deductive methods to programs-as-texts:

‘Hoare and I both assume that programs-as-causes are represented by programs-as-texts. The difference is that Hoare assumes that programs-as-causes are always appropriately represented by programs-as-texts, an assumption that I challenge’ (Fetzer, 1989)”

II. The “materiality argument” in the “formal verification” debate

- The final - reconstructed – table after (Fetzer, 1998)

<table>
<thead>
<tr>
<th></th>
<th>Mathematical Proofs</th>
<th>Scientific Theories</th>
<th>Computer Programs-as-texts (unloaded)</th>
<th>Computer Programs-as-causes (loaded)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntactic Entities</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Semantic Significance</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Causal Capability</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
II. The “materiality argument” in the “formal verification” debate

- The final - reconstructed – table after (Fetzer, 1998)

<table>
<thead>
<tr>
<th></th>
<th>Mathematical Proofs</th>
<th>Scientific Theories</th>
<th>Computer Programs-as-texts (unloaded)</th>
<th>Computer Programs-as-causes (loaded)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntactic Entities</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Semantic Significance</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Causal Capability</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

 Appropriately represents (Hoare)

 Approximately represents (Fetzer)
II. The “materiality argument” in the “formal verification” debate

- The final - reconstructed – table after (Fetzer, 1998)

<table>
<thead>
<tr>
<th></th>
<th>Mathematical Proofs</th>
<th>Scientific Theories</th>
<th>Computer Programs-as-texts (unloaded)</th>
<th>Computer Programs-as-causes (loaded)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntactic Entities</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Semantic Significance</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Causal Capability</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- But there is a remaining ambiguity in the term “represent”: Scientific modeling relation for Fetzer ≠ Formal modeling relation for Hoare
II. The “materiality argument” in the “formal verification” debate

A brief summary (1/2)

• Due to the necessary use of a material (non abstract) machine:
  – We have to distinguish between programs as texts and programs as causes
  – There is a relation of representation between the two
  – According to Fetzer, this is a relation of approximate modeling just like the one pertaining to applied mathematics formulae in scientific theories
  – According to him, the parts of the loaded programs (programs-as-causes) are subject to causal relationships
  – According to Hoare (understood by Fetzer), this is a relation of formal modeling within the domain of pure mathematics.
II. The “materiality argument” in the “formal verification” debate

A brief summary (2/2)

• Due to the necessary use of a material (non abstract) machine, :
  – For Hoare, one single execution of the program on a material computer (provided that its electronics works well: hence his reservation through the expression “in principle” in his quotation) is only an instantiation of the program as text
  – Whereas, for Fetzer, the materiality either (1) of the program or of (2) an aspect of the program or (3) of one type of program (this point remains ambiguous) leads to identify the activity of programming with the building of scientific theories
  – Hence, for Fetzer, programs, just like scientific theories, can be compared with scientific conjectures and their testing can be compared with attempts to refute them (in this respect, Fetzer explicitly refers to Popper)
III. THE “MATERIALITY ARGUMENT” IN THE DEBATE ABOUT THE EMPIRICITY OF COMPUTER SIMULATIONS
Warnings

• Contrary to the “program verification debate”, the debate on the role of the materiality in the assessment of the epistemic power of computer simulations (CSs) has been explicit
• But it has raged much later (officially, from 2009 on).
• In this context, the role of materiality has been directly discussed and debated
• Many different authors have defended many different claims (sometimes only slightly differing)
• There already exists some syntheses of this debate: (Varenne, 2001), (Morrison, 2015), (Duran, 2018)
The “pre-history” of the debate
Galison 1996, 1997

Three arguments in favor of the assimilation of CSs to experiments

• **The epistemic argument**
  – If you consider a CS as a black box, the ins and outs of this black box are of the same nature and are treated the same way as the ins and outs of real experiments

• **The stochasticist argument or more generally the essentialist, structuralist or realist argument**
  – There exists a structural “correspondence [of the discretization imposed by the CS] to nature” (1996, p. 146);
  – The “Monte Carlo method” has a “mimetic power” (1996, p. 148)

• **The pragmatist argument**
  – CSs are just another - most effective - numerical method applied to more realist (i.e. continuous formulae) representations
The “pre-history” of the debate
Galison 1996

The epistemic argument in favor of the empiricity of CSs

• “All the forms of assimilation of Monte Carlo to experimentation that I have presented so far (stability [robustness], error tracking [round-offs problems], variance reduction [statistical estimation, strategies to bias simulated samples and algebraically compensate this bias to accelerate the convergence to the estimate], replicability [publication of CSs’ programs’ lists, portability of CSs thanks to the recourse of universal languages and to the diffusion of data recordings, modularity of programming style, these modules being “analogous to the routine elements of the practice in theoretical physics such as Laplace’s equation, group theory, vector algebra, or the tensor calculus”, ibid. p. 140], and so on) have been fundamentally epistemic. That is, they are all means and practices by which the researchers can argue toward the validity and robustness of their conclusions.”

The “pre-history” of the debate
Galison 1997

The epistemic argument in favor of the empiricity of CSs

“It is preferable to think of a large scale calculation as a numerical experiment, with the program as the apparatus, and to employ all the methodology which has previously been established for real experiments (notebooks, control experiments, error estimates and so on)”


And Galison adds a comment:

“Computational errors were uncertainties of this sort [of the sort of uncertainties occurring in the resolution of the measure of a really empirical events such as a spark in a spark chamber]; they had their origin in the resolution of the device, but the ‘device’ was now a piece of software”

A critic of the “physicality claim”
Barberousse, Franceschelli, Imbert, 2009

• “Whereas computer simulations involve no direct physical interaction between the machine they are run on and the physical systems they are used to investigate, they are often used as experiments and yield data about these systems. It is commonly argued that they do so because they are implemented on physical machines [ = “physicality claim”, see e.g. Norton, Suppe, 2001]. We claim that physicality is not necessary for their representational and predictive capacities and that the explanation of why computer simulations generate desired information about their target system is only to be found in the detailed analysis of their semantic levels [and from their hierarchy going from the physical level to the high-level language level]. We provide such an analysis and we determine the actual consequences of physical implementation for simulations”

A critic of the “physicality claim”
Barberousse, Franceschelli, Imbert, 2009

• “The physicality claim seems to underlie the metaphor of the computer as ‘a stand in for’, or ‘a probe of’ the system upon which experiments are made”, Ibid., p. 559

1. First, Barberousse et al. assume that the epistemic argument itself is more or less grounded on some structuralist argument

2. Then, they show that there is no simple nor stable isomorphism between the structure of the target system and the different structures of the physical states of the computer

3. But they maintain the character of empiricity for some CSs while not really explaining it, despite their parallel invocation and explanation of semantic levels

4. The semantic levels they invoke are multiple but all are internal to the computing system and refer to formalisms only: this is due to their focus on numerical simulation of equation-based models only

5. Finally they claim that “physical implementation matters only in so far as it enables us to perform huge computations and thereby mechanically unfold what was contained in the discrete dynamical equations”, ibid., p. 573.

So, for them, materiality seems to be, but really is not decisive: it only permits a change in degree and not in nature
A turning-point: Parker 2009


Main claim:
- Computer Simulations “do not qualify per se as experiments but typical scientific studies involving computer simulations models do” (p. 2)
- And, in that respect, materiality matters
A turning-point: Parker 2009

- Section 2.1: Objections to Guala’s proposed distinction

For Guala (2005), there is an “ontological difference” between an experiment and a simulation:

“The difference lies in the kind of relationship existing between, on the one hand, an experiment and its target system, and on the other, a simulation and its target. In the former case, the correspondence holds at a “deep,” “material” level, whereas in the latter, the similarity is admittedly only abstract and formal. … In a genuine experiment, the same material causes as those in the target system are at work; in a simulation, they are not, and the correspondence relation (of similarity or analogy) is purely formal in character. (Guala 2005, 214-215).”

- So, for Guala: E (material correspondence) TS ≠ S (formal correspondence) TS

Parker’s “note 8”: “A formal similarity consists in the sharing of some formal property, where a formal property is a property that can only be instantiated by a relation between/among other properties.”

Objection to Guala by Parker (2009):

Parker objects to the too restrictive distinction made by Guala (2002 ; 2005): contrary to Guala, the terms “simulation” and “experiment” share some part of their extensions.

Some simulations are experiments and some experiments are simulations.
A turning-point: Parker 2009

• Against Guala, Parker objects:

  – 1\textsuperscript{st} that an experiment remains an experiment even when eventually those are not the same material causes that really are at work in it (not the same as the experimentalist thinks they are).
    • E.g. a biologist can make an experiment with a new drug on a rat (having humans as target systems): this drug eventually can trigger physiological effects that are not the same as the ones which would be at work if the experiment had been made on humans. Still, this still unrecognized error of attribution of causality doesn’t prevent the whole procedure to be called an experiment on the rat with human as target systems.

•

  – 2\textsuperscript{nd} that some physical simulations exist which also are experiments
    • E.g. simulation on a real car with simulated winds in a wind tunnel (i.e. simulation of natural winds thanks to winds caused by artificial means).
    • The same material causes will be at work on the car but the whole procedure can be called a simulation.
    • Similarly, any experiment conducted on a scale model also is a simulation.

Hence it is wrong to say that, in any simulation, the material causes never can be the same as the ones at work in the target system.
Parker’s characterizations of a simulation and of an experiment in general

- **Simulation:** “I characterize a simulation as a time-ordered sequence of states that serves as a representation of some other time-ordered sequence of states; at each point in the former sequence, the simulating system’s having certain properties represents the target system’s having certain properties.”

  - Note 6: “Note that the sequence of states that constitutes a simulation is something defined and chosen by the scientist—as a simulating system evolves, neither all of the states that it occupies nor all of its properties at any given time must be assumed to represent states or properties of the target system.”

  - “In virtue of what does one entity represent another? Not in virtue of there being some particular mind-independent relationship between the two entities; rather, it is just that some agent intends that the one entity stand for the other [...] In practice, scientists typically select a simulating system on the basis of its being hoped or believed to be similar to the target system in ways deemed relevant, given the goals of the simulation study. For the kinds of goals typical in scientific studies (e.g. prediction or explanation), the aim is usually for there to be particular formal similarities between the simulating and target systems.” (p. 7)

- **Experiment:** “An experiment can be characterized as an investigative activity that involves intervening on a system in order to see how properties of interest of the system change, if at all, in light of that intervention”

- **Intervention:** “An intervention is, roughly, an action intended to put a system into a particular state, and that does put the system into a particular state, though perhaps not the one intended.”
• “A computer simulation is a sequence of states undergone by a digital computer, with that sequence representing the sequence of states that some real or imagined system did, will or might undergo. So computer simulations do qualify as simulations, as expected. But precisely because computer simulations consist of sequences of computer states—not activities undertaken by inquiring agents—they clearly do not qualify as experiments” (ibid)
A turning-point: Parker 2009

Short interlude and comments on Parker’s characterization of a CS

For Parker, a CS is “a type of representation”, “a sequence of computer states representing other sequences of states”.

• I suggested that a CS is a different kind of thing. This ambiguous word (“simulation”) either denotes a specific kind of process or the result of this specific kind of process.
  – As a process, a simulation is not directly nor essentially a representation. It is a way of operating with symbols: it is “a strategy of symbols processing which is always at least partly based on the power of exemplification of some of these symbols” (Phan, Varenne, 2008; Varenne, From Models to Simulations, 2007/2018).
  – First and foremost, the most important (prevalent) meaning of the term “simulation” leads us to say that it is always a type of process operated on symbols and not always a type of symbol or representation (even if it is true that some but not all r-simulation can be representations of states sequences).
  – Parker’s characterization wrongly merges the p-simulation and the r-simulation. More precisely:
    • 1. it wrongly conflates the process and the product;
    • 2. it also wrongly assumes that an r-simulation always has to be temporal;
    • 3. finally it also wrongly assumes that an r-simulation always has to directly represent at least some parts or some aspects or some time-slices of an external target system.
A turning-point: back to Parker 2009
On “Computer Simulation Studies”

• Nevertheless, for Parker, as a kind of representation, a CS cannot qualify as an experiment. But, for her, if we take into account the kind of “broader activity” she calls “computer simulation study”, it can qualify as an experiment.

• Because a “computer simulation study” includes a series of activities and interventions:
  – 1. “setting the initial state of the digital computer from which a simulation will involve” [which is akin to preparing and constraining the initial conditions of the experimented system before an experimentation stricto sensu]
  – 2. “triggering the evolution by starting the computer program that generates the simulation”
  – 3. “then collecting information regarding how various properties of the computing system, such as the values stored in various locations in its memory or the colors displayed on its monitor, evolve in light of the earlier intervention (i.e. the intervention that involves setting the initial state of the computing system and triggering its subsequent evolution)”

As we can see, Parker’s claim is very similar to the argument about CSs identified as the “epistemic argument” by Galison (1996).
A turning-point: Parker 2009

On “Computer Simulation Studies”

Hence Parker goes on:

• “So defined, a computer simulation study does qualify as an experiment—an experiment in which the system intervened on is a programmed digital computer. The importance of looking at computer simulation studies in this way—as experiments on real, material/physical systems—will be the focus of the next section.”
According to Parker (2009), CSs are not “nonmaterial experiments”: i.e. they are not experiments on mathematical models (Morgan’s claim).

Let’s recall Morgan, 2003:

“Question 1: What counts as a material intervention? [...] How far may we stretch the notion of what counts as "material" in experiments? I will outline the character of two types of experiments: "virtually" experiments, ones in which we have nonmaterial experiments on (or with) semimaterial objects, and "virtual" experiments, ones in which we have nonmaterial experiments but which may involve some kind of mimicking of material objects.

Question 2: How do the results of these vicarious experiments relate to the world? The issues of experimental validity are many, but I suggest in this context that the validity of experimental results also hinges on how the objects and models used in experiments relate to the things they are supposed to tell us about in the world. Models used in experiments represent the material world in different ways, and this has implications for the range of inference that such vicarious experiments can support”, Mary Morgan, “Experiments without Material Intervention: Model Experiment, Virtual Experiments and Virtually Experiments”, in Hans Radder (ed.), The Philosophy of Scientific Experimentation, Pittsburgh, Pittsburgh University Press, 2003: p. 217.

About virtually experiments or experiments on semimaterial objects (e.g. discretized real bone): “These are experiments [...] using a mathematical model as an experimental instrument, not experiments on the mathematical model to find out how it behaves.”, ibid., p. 223.
First, let’s recall Morgan, 2003, p. 233 (continued):

“I have taken materiality to be one of the prime characteristics of interest in these hybrids and suggest that when faced with such a hybrid experiment, we need to look carefully at where and how much materiality is involved, and where it is located, before we can say much more about its validity.”
Importance of materiality and “degrees of materiality” in Morgan (2003)

For Morgan (2003), in a real classical experiment, both the object and the intervention on the object have to be material, physical (p. 218).

• According to Morgan, one can say that there exist experiments on mathematical models (as in classic mathematical modeling in economics) because such practices with models use some kind of external “intervention” (ibid., p. 218). But those external “interventions” are not material. Those are the questions asked to the models by exploring it with different parameters or initial values. These “experiments on models” are said by Morgan to be “extended verbal thought experiment”. There processes are deductive and mathematical – nonmaterial in this sense - even if they use the computer. They only can surprise but not confound us (2003, 2005).

• Note here that, contrary to Parker (2009), an intervention consisting in the action to set some initial conditions and to then trigger the model is not considered material by Morgan: hence, it’s not a real intervention.

• As a consequence, her analysis of experiments cannot be purely epistemic: the nature of the process at stake remains very important. For her, the materiality of the objects acted upon and of the objects used for the intervention is crucial. This is the very reason why, for Morgan and contrary to Parker, an “experiment on a mathematical model” is far away from a real experiment on material systems. Morgan says that this is because, contrary to experiments on “semimaterial objects”\(^1\), it has a very low “degree of materiality”.

\(^1\)“because [this type of semimaterial object] may contain things in the structure that the scientist did not ‘know’ about” (ibid., p. 224).
Parker, 2009:

“To some extent [Morgan’s] characterization of computer simulation studies as nonmaterial experiments (e.g., ibid., 217) seems to depend on their being understood as experiments on mathematical models or mathematical objects (ibid., 225, and 232) rather than on real, physical stuff. [questionable !] It is this rather common way of thinking of computer simulation that I want to take issue with in this section.

In fact, I want to stress the importance of instead understanding computer experiments as, first and foremost, experiments on real material systems. The experimental system in a computer experiment is the programmed digital computer—a physical system made of wire, plastic, etc.”
Then Parker shows that it is not plausible to say that a CS is an experiment on a mathematical model, because:

1\textsuperscript{st} - we experiment not on the model but on a \textbf{realization} of the mathematical model: the “\textbf{physical computing system}”

   » Note 14: “Following Norton and Suppe (2001, 105), system B is a \textbf{realization} of system A when there is a \textbf{many-one behavior-preserving mapping} from the states of B onto the states of A. Note, however, that by ‘states’ here I mean states of the system as defined by the scientists conducting the study, which will pick out only some system variables and only at particular times or intervals of time (see note 6)”. [implicit assumption: in a realization, similarly to what happens in a simulation, a temporal sequence has to partially but iconically represent another temporal sequence]

2\textsuperscript{nd} – moreover, through that, we do not experiment on a direct realization of the initial equations of the model but on some realizations of some discretized equations that approximate the initial equations

3\textsuperscript{rd} – as a consequence of this indirectness, we need “justifications” to validly infer from the CS results to the “mathematical system specified”, and finally to the target system
A turning-point: back to Parker 2009

Material verification and materially grounded chain of inferences

This is precisely where Parker decides to invoke the importance of the material “verification” practices (≡ Fetzer 1989!)

“In many fields, there is even a special (and somewhat misleading) name—verification—given to the process of collecting evidence that a computer simulation can indicate solutions that are close enough to those of the programmed equations and/or the preferred model equations. Verification activities are concerned not only with such things as the adequacy of the numerical methods that are employed to estimate solutions to the programmed equations, but also with the design and actual implementation of the computer program that is run to generate the simulation (see e.g. Oberkampf et al. 2004).

Possible errors in the coding of the program are considered important and are investigated as part of the verification process. And if the computer hardware itself is not working properly or as expected, that too can sometimes be revealed through verification activities [Fetzer’s argument].”

The material part of the process of program verification is essential to this verification or justification and to this “chain of inference”:

“With that kind of approach [verification], the aim is often to justify a chain of inferences, from simulation results to conclusions about solutions to the programmed equations, from conclusions about solutions to the programmed equations to conclusions about solutions to the preferred model equations, and from conclusions about solutions to the preferred model equations to conclusions about the real-world target system.”
A turning-point: back to Parker 2009

In the absence of such “chain of inferences”, Parker (2009) still invokes the necessity of some material control on “hardware changes” and on “additions of new codes”

There exist some CSs which result can be directly compared to their target system without using such a process of chain of inference.

Parker gives the example of predictive modeling in meteorology: these CSs are “improved in light of repeated testing against meteorological observations”.

“Such evaluations involve comparison of the observed behavior of one material/physical system (the programmed computer(s) that produces the forecasts) with the observed behavior of another material system (the atmosphere), and there is a keen awareness of this materiality among the scientists who manage and develop the forecasting system; they recognize the potential for such things as hardware changes and addition of new code to impact the accuracy of their forecasts, and they watch for and try to avoid problems that might arise from such changes”
Parker draws the main conclusion of her article from this invocation of the **materiality** of the procedure either of verification or of control of CSs:

CSs are **material** experiments on **material** systems, that is,

on “programmed digital computers”

- “In a very straightforward sense, computer simulation studies are **material** modeling exercises and **material** experiments”

- “The **mathematical system** specified by the preferred model equations is no less an **external target system** than the real-world material system that motivated the computer simulation study in the first place. Inferences about these and any other target systems will take as their starting point the observed behavior of a material system— the **programmed digital computer** —and accounts of the epistemology of computer simulation should reflect this.”
Objection of the “same stuff”

To answer this objection (inspired by Morgan, 2002), Parker gives some examples (problems of feasibility, of dimensionality, Reynolds number, etc.) where one can see that:

“It is not always the case that inferences about target systems are more justified when experimental and target systems are made of the “same stuff” than when they are made of different materials.”

My comment on “more justified”: in 1948 (in “The role of the digital procedure in reducing the noise level”), Von Neumann already had shown this. See Varenne, 2001; 2009; 2018)
Hence, for Parker, the focus on materiality must not be misunderstood.

“The focus on materiality is somewhat misplaced because what is ultimately of interest when it comes to justifying inferences about target systems is not materiality, but relevant similarity. To justify such an inference, scientists need evidence that the experimental system is similar to the target system in whatever respects are relevant, given the particular question they want to answer about the target system. The relevant similarities might be formal similarities, material similarities or some combination of the two, depending on the type of experiment and the target question at hand. But, crucially, having experimental and target systems made of the same materials does not guarantee that all of the relevant similarities obtain.”
Trying to sum up Parker 2009

• In 6 successive arguments:
  – 1. Epistemic argument on CSs as black boxes (Galison, 1996)
  – Objection: the black box is not material
  – 2. Materiality of the black box;
  – Objection: the object upon which we experiment in a CS is not material. It is a mathematical model. So it is an abstract system.
  – 3. Material program verification argument: in a CS, you never directly experiment on the abstract system because the behavior of the “physical computing system” is at least partly autonomous from its corresponding mathematical model [= Fetzer, 1989]
  – 4. With CSs, also, material direct comparisons between 2 different material systems are possible
  – Objection: hence you implicitly assume that your materiality argument itself is grounded on some kind of hidden, deep and unexplainable connivance between different material systems qua material, don’t you?
  – 5. No. To have the “same stuff” is not necessary: materiality matters but not in the sense you mean:
  – Objection: if this is not materiality qua materiality that matters, what does matter? That is: in which sense does materiality really matter, if it does not matter qua materiality?
  – 6. Materiality matters in the sense that it can participate to the definition of the “relevant similarities” in a particular question asked [perspectivism]
  
  Question: with this noun “similarity”, you named the problem, didn’t you?
First conclusions (1/2)

1. In both categories of argument (Inductive Program Verification, Empiricity of CSs), philosophers assume right from the start that it is more relevant to take into account the whole system, i.e. the “physical computing system”: the software + the hardware

2. From this on, their common and main argument is grounded on the claim that either the knowledge of the program as text or of the mathematical model implemented for the CS is not sufficient to predict the behavior of the “physical computing system” as a whole

3. Note that Fetzer’s argument can be seen as only a subpart of the global materiality argument in favor of the empiricity of CSs
First conclusions (2/2)

4. Parker adds the idea that the mathematical model is external to the computing system, comparing it with the traditional external target system. So the “programmed computer” is a tertium quid which is material and upon which one experiments.

5. The problem with this claim is that it creates a new gap between the model and the CS upon which we experiment: how to justify the persisting relevance of the CS results for the formal model? The notion of “similarity” is used in the answer: but it seems very vague.

6. My diagnosis: The notion of similarity dramatically reemerges at the end of the arguments (just as in the final opposition between “appropriately represents” and “approximately represents” in Hoare/Fetzer’s debate) because, at the beginning of the reflection, the concepts of “program”, “simulation” and “realization” have all been interpreted in terms of some relation of “representation” without further analysis of this rich and ambiguous notion.
THANK YOU!


Varenne (2009), Qu’est-ce que l’informatique?, Paris, Vrin.