Computer modeling and simulation: towards epistemic distinction between verification and validation

V.S. Pronskikh

Fermi National Accelerator Laboratory

Introduction

Verification and validation of computer codes and models used in simulation are two aspects of the scientific practice of high importance and have recently been discussed by philosophers of science. While verification is predominantly associated with the correctness of the way a model is represented by a computer code or algorithm, validation more often refers to model's relation to the real world and its intended use. It has been argued that because complex simulations are generally not transparent to a practitioner, the Duhem problem can arise for verification and validation due to their entanglement; such an entanglement makes it impossible to distinguish whether a coding error or model's general inadequacy to its target should be blamed in the case of the model failure. I argue that in order to disentangle verification and validation, a clear distinction between computer modeling (construction of mathematical computer models of elementary processes) and simulation (construction of models of composite objects and processes by means of numerical experimenting with them) needs to be made. Holding on to that distinction, I propose to relate verification (based on theoretical strategies such as inferences) to modeling and validation, which shares the common epistemology with experimentation, to simulation. To explain reasons of their intermittent entanglement I propose a weberian ideal-typical model of modeling and simulation as roles in practice. I suggest an approach to alleviate the Duhem problem for verification and validation generally applicable in practice and based on differences in epistemic strategies and scopes.

Modeling and simulation

In order to disentangle verification and validation one needs first to disentangle and explicitly define modeling and simulation. Among approaches to their definition one can identify two most frequently occurring ones in literature: those terms are either used interchangeably without assigning them unambiguous and explicit definitions or in a way that simulation represents a subset of more universal modeling practices. For example, the definition of simulation by E.Winsberg is manifold, he describes it both as “the kind of “theorizing” […] – the construction of local, representative models,” and experimenting with computer (Winsberg, 2010); he also associates simulation with a model itself as used in simulation. In (Keller 2003, 204) it is also argued that “computer simulation is … directed toward eliciting the implications of well-formulated theoretical models”, relating it to a more extent to theory. P. Humphreys gives computer simulation a definition of numerical experimenting (which is close to one of Winsberg's characterizations): "a computer simulation is any computer-implemented method for exploring the properties of mathematical models"; he also calls simulation a computational device producing solutions to the model (Humphreys, 1991). S.Peck argues that "simulation can be viewed as another kind of experimental system" (Peck, 2004, 530). M.Morrison (Morrison, 2009, 55) also relates simulation to numerical experimentation and contends that "computer plus simulation programme functions as apparatus". Some accounts even characterize computer simulations as "material experiments in a straightforward sense" (Parker, 2009, 495) because they are performed on a digital computer (which is a material system) while others argue that the physicality of processes in a computer does not explain why simulations generate new data (Barberousse, 2009, 573).

There is also an ambiguity in definitions of modeling. For example, (Morrison, 2009, 47) claims that "the computational resources of simulation [...] make[] it different from modelling" and ascribes simulation to "a type of "enhanced" modelling". M.Weisberg delineates modeling as "the indirect study of real-world systems via the construction and analysis of models" (Weisberg, 2013). Here modeling stands not only for building models but also for their analysis, which can also be understood as exploring their properties. In the former case it has much in common with the way Winsberg defines simulation, and in the latter – how both Winsberg and Humphreys characterize simulation in the sense of numerical experimentation. This latter interpretation is also supported by Weisberg's description of simulation as "computing the behaviour of the model using a particular set of initial conditions" (Weisberg, 2013).

One of the first and most cited denotations of simulation (not necessary computational) was given by S.Hartmann, who wrote that "a simulation imitates one process by another process" (Hartmann, 1996). While this definition apparently encompasses both modeling and simulation as described by the aforementioned authors, one can clearly envision that in order to accomplish such a simulation it is necessary to both construct the imitating process (or a code (a computer-implemented algorithm) in the case of computer simulation) and explore its behavior (run the computer code with a particular set of input parameters or perform many runs with parameters covering all the parameter space of the problem) ((Winsberg, 2014) also describes simulation as one run of a computer code). Even if we look into the construction of a computational code of the imitating process we can clearly discriminate between elementary processes (like interaction of a particle with a nucleus) and composite processes (heat release in an irradiated water tank, radiation propagation and attenuation in matter) that are constructed by embedding low-level and more universal elementary ones into a more general framework.

For instance, in particle physics, a model of particle interaction with another particle or a nucleus is more elementary than that of its interaction with a block of material, where the particle encounters sometimes hundreds of other particles and takes part in a multitude of interactions of different kinds. The reason of this relative simplicity is that an elementary process model and occurs at another scale of system organization or different (lower) level of it. Building higher-level structural models can be recognized as a separate kind of activity and expertise (epistemic scope) from both running simulation code and elementary process model building; however, from the point of view of scientific practice, those practitioners who run simulations can also either build structural models or adjust existing ones to their needs as their different role. This allows to consider modelers and simulationists as ideal types in the weberian sense as will be described below. That is one of the reasons I shall discuss the higher-level structural model construction as a part of simulations.

One possible way to support the necessary distinction is to consider the epistemic scope of a practitioner in the field. Counterintuitively, an increase in the level of model organization does not always entail a respective increase in complexity and scope of required knowledge but changes the scope of that knowledge; such an increase usually implies alterations of the scope. A process modeler, who is supposed to build his or her models from the very basic principles, needs to be familiar and able to apply all mathematical structures pertinent to models he or she builds down to the level of the most elementary processes and basic laws. On the contrary, simulationists who are not process modelers and thus build and apply higher-level models of composite objects often are not required to have an extensive acquaintance with the basic structures of models that the underlying elementary processes are based upon; their concern is that the models they use be well verified by the modelers who create them[1]. Elementary models are usually provided to them in the form of ready-to-use computational procedure units suitable for incorporating them into more complex composite models. Therefore, simulationists have to envision the structure and macroscopic designs of the complex system they intend to construct. Simulationists also have a general understanding how the relevant model parameters affect behavior of the modeled system and explore influence of those parameters.

Thus, there exists an apparent controversy in the discussed above definitions of modeling and simulation leaving room for a more rigorous characterizations of both domains capable of answering the question whether simulation is construction of a model, computing its behavior or both (the entire computational study of a particular system). I argue that such controversy can nevertheless be resolved provided one considers differences in both aspects of the practices (construction of elementary models, construction of composite models, their explorations) and epistemic scopes of corresponding practitioners (knowledge how to construct elementary process model on the one hand and knowledge how to construct a real-world target model based on a set of pre-built embedded elementary models and numerical experimentation). This latter construction of a "real-world" higher-level target model can comprise carrying out many individual runs of a simulation code supplying it with different sets of input parameters. Based on the discussion above I propose to resolve the controversy by defining modeling as creation of computational mathematical models of elementary objects and processes and simulation as creation of composite computational models (those embedding elementary ones) by numerical experimentation.

[1] Much alike the use of a TV set or a phone does not require the knowledge of its internal organization.

A study of magnet placement in experiments with the Tevatron accelerator can serve an example of the distinction. In order to describe and study propagation of protons in a complex magnetic fields of the accelerator (including second-order effects such as so-called fringe fields) one needs to ascribe magnetic fields to individual magnets, dipoles, quadruples etc. which altogether constitute the entire magnetic system of the accelerator and create its magnetic optics. The magnets are characterized by shapes, sizes, relative arrangements and altogether represent the organization of accelerator's technical real-world in its highest-order scale (I do not discuss here the social dimension). Construction and study of the accelerator's computational model can thus be regarded as simulation. In the course of such simulation (model creation and particle propagation numerical study) practitioners usually run computer models a multitude of times supplying it with various sets of parameters covering all the parameter space under scrutiny in order to meet an optimal regime. As I shall discuss in the next paragraphs, such simulation is grounded in strategies which possess many features of experimental practice rather than theoretical and such an ascription of experimental strategies to simulation *rather than modeling* constitutes an essential part of my further argument. However, construction of a composite accelerator model relies in turn on incorporated in it elementary models of magnetic field creation by charged particles that are governed by Maxwellian equations. Maxwellian low is the lowest, the most elementary level of the accelerator system organization and its computational implementation serves a building block of the higher-level simulation model of the entire accelerator. On the other hand, a representation and solution of Maxwellian equations are analytical and constitutes an inference pertinent to theoretical strategies. Therefore, development of computational procedures calculating solutions of equations of electrodynamics for an arbitrary set of initial conditions is deemed as modeling for the purpose of my argument.

Applicability of the epistemology of experimentation

Based on the distinction made above between simulation and modeling an ascription of experimental strategies (Franklin 2012) to simulation rather than modeling in the aforementioned sense can be made. Simulationists as higher-level model designers often use common sense considerations to verify that their results are consistent; however, more often they "benchmark" their results (or outputs) against experimental or observational "real-world" data as well as other simulation methods (computer codes). For example, a complex magnetic field produced by a complex accelerator structure can sometimes be also measured experimentally and compared to a simulation output. Nevertheless, matching their outputs to analytical solutions is not generally available to them due to both complexity and opacity (Humphreys, 2004) of the systems they simulate and difference in their epistemic scope with modelers. It is, however, possible for modelers, who create models of elementary processes, for instance, to obtain an analytical solution of a lower-level problem of electrodynamics for a simplest magnetic structure and then verify how its computational representation is programmed; for such models either other computational models or analytical solutions usually exist to compare to.

Another method frequently used to increase confidence that apparatus works properly is to vary one of the parameters of the system under scrutiny, for instance, adding ink to a sample and observing the predicted color change in a microscope. (Winsberg, 2010) discusses that simulationists also vary parameters of the model and check whether the system responds in accordance with their expectations. However, by virtue of the distinction between the two scales (modeling and simulation ones) one can see that such approach is possible only for simulationists working with high-level models of composite objects and processes, and in this respect it is similar to the conventional experimentation. For instance, a simulationist can vary distances between individual dipole and quadruple magnets in an accelerator arrangement to see the response, for example, whether the agreement with the measured field strength becomes better or worse (I shall refer to this example when discussing validation experiments). One more such example is varying density or material composition in a model sample irradiated by certain particles and matching simulated energy release to that measured in a calorimetric experiment. On the other hand, models of elementary processes (eventually leading to a heat production) are verified by modelers like theories in a way different than object ones, i.e. parameter variations cannot suffice to argue for their validity. Again, the latter strategy is conceivable for elementary (low-level) process modeling at the stage of its computer implementation in order to verify if the model is suitably coded. That stage, nevertheless, cannot be referred to as the simulation (high-level) model construction itself. Here, simulation is considered not only as "enhanced modelling" but also a domain of different scale and scope than the computer modeling.

One more Franklin's experimental epistemic strategy (Winsberg 2010, 44) finds similar to that used in simulations is measuring the same observable with a different kind of apparatus; in simulations that strategy correlates with simulating the same system using two or more different models. According to the distinction between simulation and modeling, such a strategy cannot be applied to modeling of elementary processes and elaborating such models as another instrumental theoretical model in order to be as valid as the previous one is supposed to reproduce the same set of empirical data as the first one and not necessarily its predictions outside the relevant data range. Rather than modeling, parameters of higher-level models in simulation are varied exactly in the way as it is done in experimentation, assuming different models to be different "apparatuses". The same accelerator can be simulated by independent simulation codes, for example, MAD and Synergia, which exploit completely different high-level concepts and assumptions (and can be associated with two different apparatuses in experimentation). Nevertheless, all codes used in the field reveal identical understanding of the low-level Maxwellian electrodynamics, which belongs to the scope of elementary process modeling.

In the case of different models of the same "real-world" object, for instance, accelerator (or thunderstorm) different codes supplied with sets of input parameters ("lattices") are different representations of the same "real-world" object; however, they refer to different model objects –

sets of structural models implemented based on different assumptions. The correspondence between these different models and the "real-world" one is not obvious as the simulations models can employ various abstractions and idealizations (Humphreys, 2004) and, more importantly, models of processes may contain many different fictitious assumptions (like the artificial viscosity model (Winsberg, 2010, 14)) or even be in a contradiction with experience and underlying physical laws (like the Arakawa operator (Lenhard, 2007)). From this point of view, an important way to increase confidence in simulation results is to investigate models based on as different as possible or at least independent model representations of the process under scrutiny, as being different approaches to description of the same reference process by an imitating one, neither of the models can be thought of as *per se* more relevant. That is why the experimental strategy of comparing simulations employing different higher-level model representations and different sets of incorporated low-level models of processes is imperative to increase reliability of simulation results.

Modeler and simulationist as ideal types

Despite due to substantial differences in epistemic scope and strategies as well as their relation to different organizational scales, simulation and modeling are evidently distinct they are often interwoven in the scientific practice. That implies that individual practitioners are often engaged in both kinds of activities. In order to represent this, the weberian theory of ideal types (Weber, 1949, 49) can be involved. Let us assume simulationist and modeler to be two ideal types whose differences on epistemic and ontological grounds are extensively discussed throughout this paper. One more important ideal type is an IT expert, whose expertise comprises computer programming and competent operation. Virtually, a practitioner can belong solely to any of these ideal types, however, more often his or her function encompasses all the three domains in one way or another.

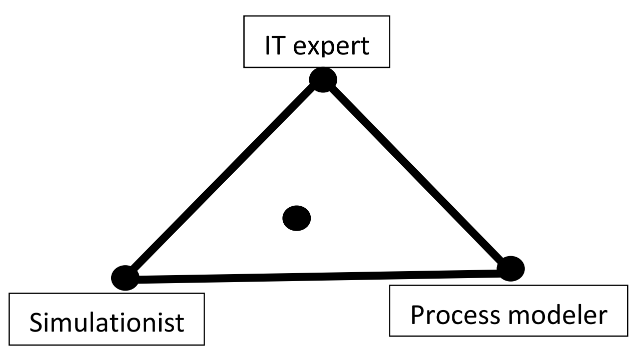

Figure 1. Triangle of ideal types in the simulation practice (vertices of the triangle) and the expertise of an actual practitioner (the dot inside the triangle).

There is, therefore, one more significant similarity between experiments and simulations, which results from the distinction discussed above. Experimentation requires a detailed knowledge of the instrumental and technical theories on which the functioning of apparatus is based for the sake of interpretation of data in terms of high-level theories. However, being encoded in computational procedures elementary process models can serve examples of procedural knowledge; they do not represent knowledge *why* something is to be calculated in a particular way (or proofs and inferences pertinent to theory), but rather a recipe *how* one can calculate a quantity, i.e. an instruction how to obtain an answer to a particular question by means of either applying it computationally or supplying it as a set of input instructions to a computer code.

These procedural models are circulated between modelers and simulationists, and therefore, simulationists in order to investigate a model in simulations need, on the one hand, to choose models of processes, and construe out of them composite models (simulations) of a "real-world" object whose properties they intend to explore. Thus, the epistemic scope of simulation of a thunderstorm does not necessary encompass interactions of individual molecules in a cloud, or knowledge required to simulate interactions of a particle within a chunk of material does not necessarily encompass that of interactions of individual particles with individual nuclei, provided simulationists possess necessary elementary models as pre-existing elaborated computational procedures. This explains why a simulationist can successfully practice higher-level simulations despite being "ignorant of aspects of how [lower-level procedure] was programmed or how it works" (Parke, 2014).

Once we distinguish modelers of processes from simulationists who numerically experiment with those models as well as notice that the boundaries between these roles tend to blur in practice, one can try to draw a schematic representation of the roles involved in the production of simulation results (Figure 1). In simulation, having acquired all necessary models (codes, lattices, and input decks), an advanced IT user can start experimenting with them producing first new results. I define here an IT expert as one whose computer literacy is sufficient to engage with computer systems – codes, programming languages, and operating systems; that is usually comprehensible by an experienced practitioner from a technical professional field who for a particular reason demands to turn to simulations; that can be someone, for instance, without a background in meteorology simulating a thunderstorm, or a background in particle interactions attempting to simulate particle propagation in matter.

However, there is a long way between applying ready sets of procedures and a competent experimenting with models – simulation – because one needs to understand what kind of process models exerts which effect on output in the course of simulations. Such an understanding is crucial for simulationists in order to be able to adequately interpret outputs. That is why on the way from an IT user to an experienced simulationist a practitioner needs to communicate with modelers. Modelers provide them models of elementary processes with access to a limited parameter space of variables not implying knowledge of models' internal mathematical structures. They also can create and provide beginner simulationists example decks (sets of model parameters for process models used) and lattices (structural object models) representing solutions of simple problems, which serves them as an aid in learning how to understand and use models. An actual practitioner (see Figure 1) can be represented by a simulationist A instructed by a modeler B and an IT expert C, all the three being roles. B provides A "low-level" process models (usually in the form of procedures) and instructions how to use them, and C provides A supplementary computer codes (scripts) and instructions how to employ them.

Another path in Figure 1, that from an IT user to a process modeler usually lies through more specialized education and communication with modelers. Structure modeling skills require design thinking and geometric imagination as discussed above and can be acquired through practice as well as a more formal education. On the other hand, process modelers in order to develop their procedures are usually required certain IT skills. In the course of their everyday's practice they often reconcile low-level process model construction with applying those models to construction of composite simulations of higher-level "real-world" systems. In order to accomplish that certain IT expertise is also demanded. Thus, an actual low-level process modeler is often also a "higher-level" simulationist whereas a simulationist even starting as a pure ideal type usually acquires certain interactional expertise (Collins, 2010) in understanding of low-level models through communication with process modelers. Nevertheless, even concurrent and alternate practicing "low-level" modeling and "higher-level" simulation roles does not entail their epistemic entanglement and therefore keeping in mind that distinction is essential for differentiation between verification and validation.

Verification and its relation to modeling

Verification (code verification) is usually understood as either code verification, i.e. search for and fixing mistakes in a computer code or solution verification (estimating solution errors and accuracy of the code input and output. AIAA standard (AIAA, 1998) defines verification as "the process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution to the model". What is actually verified according to that definition is that an already constructed model is correctly implemented in the code (accurately solved) as the code is a computational representation of such a model (its conceptual description). Due to the discussed above applicability of inference to low-level elementary process models, in most cases analytical checks of both the algorithm implementation and the solution are available.

Requirement of inference turns out to be in agreement with a practitioner's statement that "verification deals with mathematics" (Oberkampf, 2004). Even in the cases when elementary process theories are semi-phenomenological and are based on empirical data, those data are acquired and low-level process model adjustments are performed in separate and independent studies, outside the context and scope of a particular high-level simulation under scrutiny. Availability of such analytical checks and the inference strategy supports association of verification with modeling in the sense discussed in this work. Referring to an example discussed above, solutions of Maxwellian electrodynamics equations can step by step be matched with outputs of the computational procedure and thus its implementation verified. Therefore, I suggest that the conventional verification is applicable to the "low-level" elementary process model building defined as modeling in this work.

Validation and its connections to simulation

Rather than verification, validation is defined by AIAA as "the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model." (AIAA, 1998) Such a description implies that the reference is a "real-world" object and analytical solution is not generally available because the system under scrutiny belongs to the higher level of system organization (accelerator or thunderstorm). One more important feature of validation is its relation to experimental data. For such system as accelerator, simulation outputs are matched with empirical results for the target system itself or its smaller copy (prototype). In the latter case experimenting with such a copy with the aim to obtain data for simulation code validation is called validation experiment. A definition of physics as an experimental science (i.e. heavily relying on empirical data) allows to understand practitioners' statement that "validation deals with physics" (Oberkampf, 2004). Simulation defined in this work as numerical experimentation with composite models of "real-world" objects involving experimentation with parameters of the computational code and other strategies common with experimentation (as discussed above) has certain connections with validation.

Referring to an example discussed in previous paragraphs, in order to simulate heat release in a composite object irradiated by various particles, one needs to validate the simulation code versus data obtained in measurements of heat release in simple objects made of pure materials using certain particles with well-defined energies and distributions (validation experiments). Low-level elementary process models in the form of pre-built procedures invoked by the higher-level simulation code are not tested at the simulation stage (including validation), belong to a different epistemic scope, and are not altered at that stage. Its relation to the higher-level objects, reliance on experimental data as well as applicability of experimental strategies allows us to associate simulation as defined in this work with validation. Such a correlation implies that higher-level simulation codes are validated rather than verified in the conventional sense. This does not exclude searches and fixes of algorithmic errors, however, unavailability of analytic solutions and inferences makes them insufficient in absence of experimental strategies.

Role of calibration

Calibration or "the process of adjusting numerical or physical modeling parameters in the computational model for the purpose of improving agreement with real-world data" (AIAA, 1998) is one of experimental strategies of exceptional importance in simulations requires particular attention given the definitions of simulation and modeling proposed in this work. In simulation as well as conventional experimentation Franklin's description of calibration as is also applicable: "a legitimate and important factor, [which] may even be decisive, in determining the validity of an experimental result" [Franklin, 1994]. From practitioners' viewpoint, it is employed when "validation is not feasible or practical," (Oberkampf, 2004) and purports adjusting computational model parameters so that its outputs matched empirical data for a well-understood (or standard) case (a validation experiment can serve an example). An agreement of model outputs with the standard case suggests that the use of that model in a novel context may also be conceivable. Bearing on the distinction in system organization levels and epistemic scopes discussed here, one has to clearly differentiate simulation (higher-level) model parameters, which are subject to alterations in the course of calibration, and elementary process (low-level) ones, which are not. Oversight of this rule in certain cases can entail entanglement discussed below.

Entanglement arguments

A number of arguments has been arisen that support the viewpoint that verification and validation are entangled. For instance, (Jebeile, 2012) argues that verification and validation are "two phases [that] cannot be performed distinctively" and thus are entangled. One of examples that can illuminate such an entanglement can be considered examined by (Lenhard, 2007) implementation of the "Arakawa operator" (this example was shortly discussed above). (Jebeile, 2012; Lenhard, 2007) point out to a possibility of introducing distortions in the system behavior through so-called discretization schemes[2], when differential equations of a mathematical model are converted to difference (algebraic) ones with the aim of a more convenient programming of a computer code. Given the distinction discussed in this work, a plausible approach to alleviate harm of the discretization and similar errors is to separate construction of such discretization schemes (low-level epistemic scope of modeling) from application of such schemes for simulation of higher-level models. I do not assert that it is always practically feasible, and concede that in certain practical cases such and entanglement can take place, however, I maintain that when scopes and strategies are separable and distinct, verification and validation can be differentiated. Distinction of scopes allows to prevent "model success due to piecemeal adjustment" (Winsberg, 2010), which causes the entanglement or the Duhem problem for verification and validation.

Another part of the entanglement argument is that "computer simulations are not open to direct inspection" (J.Jebeile, 2012) or epistemic opacity (Humphreys, 2004) of simulation. I suggest that elementary process models, their computer implementation, and code verification are open to direct inspection by modelers (theorists); also in the majority of practically relevant cases methods exist to estimate numerical solution error at this stage. On the other hand, I concur that simulation is epistemically opaque, but is a numerical-experimental practice (different epistemic scope), and proceeds through Franklin's epistemic strategies of experimentation as discussed above, and thus do not need to be open but rather properly calibrated comparably to an experimental apparatus. Therefore, epistemic opacity claim is not relevant for modeling as defined in this work and,

[2] In accelerator beam dynamics simulations similar uncertainties are often associated with so-called "symplecticity".

although true for simulation, does not bring about a verification and validation entanglement with necessity in general.

Conclusion

In this paper argue that verification and validation entanglement is not indispensable and although can arise in certain practically relevant cases is not true as a generalization. To justify that I assert that a distinction between modeling (defined here as construction of low-level mathematical computer models of elementary processes) and simulation (construction of higher-level models of composite objects and processes assisted by numerical experimenting with them) needs to be recognized. Those are different in epistemic scope, the former being theorizing involving inference and relying on analytical solutions, and the latter being numerical experimentation based on Franklin's epistemic strategies. I contend that such a distinction whenever practically feasible can alleviate implications of the Duhem problem for verification and validation. I show that although epistemically distinct, in practice modeling and simulation constitute roles and ideal types and can be performed by the same practitioners and, therefore, if not acknowledged by them as roles can call for a "role entanglement". I suggest that for this distinction to hold and to mitigate the verification and validation entanglement as well as the "piecemeal adjustment" of models undermining their reliability, low-level elementary process models undergone verification must not be altered afterwards in the course of higher-level validation simulations.

References

(AIAA, 1998) The American Institute of Aeronautics and Astronautics (AIAA), Committee on Standards in Computational Fluid Dynamics, 1998, URL = <http://www.aiaa.org/>.

(Barberousse, 2009) Barberousse, Anouk, Franceschelli Sara, Imbert, Cyrille, Computer simulations as experiments, Synthese (2009) 169:557–574.

Franklin, Allan, 2004. A. Franklin, "Experiment in Physics", *The Stanford Encyclopedia of Philosophy* (Winter 2012 Edition), Edward N. Zalta (ed.), URL = <http://plato.stanford.edu/archives/win2012/entries/physics-experiment/>.

(Hartmann, 1996) Hartmann, Stephen, The World as a Process. Simulations in the Natural and Social Sciences, In Hegselmann, R., Mueller, U. and Troitzch, K., eds. 1996. *Modelling and Simulation in Natural Sciences from the Philosophy of Science Point of View*, Dordrecht, Kluwer Academic Publishing, 1996, pp. 77-100.

(Humphreys, 1991) Humphreys, Paul, "Computer Simulations." In Fine, Forbes, and Wessels eds., PSA Volume 2, East Lansing, 1990, pp. 497-506.

(Humphreys, 2004) Humphreys, Paul, *Extending Ourselves. Computational Science, Empiricism, and Scientific Method*, Oxford University Press, 2004.

(Jebeile, 2012) Jebeile, Julie. Verification & Validation of Computer Simulations: A Philosophical Analysis, In: Proceedings of the International Conference Models and Simulations MS5, Helsinki, 2012, p.54.

(Keller, 2003) Keller, Evelyn Fox, "Models, Simulation and Computer Experiments". In Radder ed. *The Philosophy of Scientific Experimentation*, The University of Pittsburgh Press, 2003, 200.

(Lenhard, 2007) Lenhard, Johannes, Computer Simulation: The Cooperation between Experimenting and Modeling, Philosophy of Science, Vol. 74, No. 2 (April 2007), pp. 176-194

(Morrison, 2009) Morrison, Margaret, Models, measurement and computer simulation: the changing face of experimentation, Philos Stud (2009) 143:33–57.

(Parke, 2014) Parke, Emily C. Experiments, Simulations, and Epistemic Privilege, Philosophy of Science, Vol. 81, No. 4 (October 2014), pp. 516-536

(Parker, 2009) Parker, Wendy S. Does matter really matter? Computer simulations, experiments, and materiality, Synthese (2009) 169:483–496.

(Peck, 2004) Peck, Steven L. Simulation as experiment: a philosophical reassessment for biological modeling, TRENDS in Ecology and Evolution Vol.19 No.10 October 2004.

(Weber, 1949) Weber, Max, 'Objectivity' in social science and social policy. In *The methodology of the social sciences*, trans. and ed. by E.A. Shils and H.A. Finch, New York: Free Press., 1949, pp. 49-112.

(Weisberg, 2013) Weisberg, Michael. *Simulation and Similarity. Using Models to Understand the World*, Oxford University Press, 2013.

(Winsberg, 2010) Winsberg, Eric B.. *Science in the Age of Computer Simulations*, The University of Chicago Press, Chicago and London, 2010.

(Winsberg, 2014) Winsberg, Eric. "Computer Simulations in Science", The Stanford Encyclopedia of Philosophy (Fall 2014 Edition), Edward N. Zalta (ed.), URL = <http://plato.stanford.edu/archives/fall2014/entries/simulations-science/>.

(Oberkampf, 2004) Oberkampf, William L. Verification and Validation in Computational Simulation, URL = <http://www.psfc.mit.edu/ttf/2004/talks/oberkampf.pdf>