FERMILAB-PUB-19-361
ACCEPTED
February 2019

COMPUTER MODELING AND SIMULATION: INCREASING RELIABILITY BY DISENTANGLING VERIFICATION AND VALIDATION[1]

Vitaly S. Pronskikh[2]
Fermi National Accelerator Laboratory, Batavia, IL 60510, USA[3]

Abstract

Verification and validation (V&V) of computer codes and models used in simulations are two aspects of the scientific practice of high importance that recently have been discussed widely 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 the model's relation to the real world and its intended use. Because complex simulations are generally opaque to a practitioner, the Duhem problem can arise with verification and validation due to their entanglement; such an entanglement makes it impossible to distinguish whether a coding error or the model's general inadequacy to its target should be blamed in the case of a failure. I argue that a clear distinction between computer modeling and simulation has to be made to disentangle verification and validation. Drawing on that distinction, I suggest to associate modeling with verification and simulation, which shares common epistemic strategies with experimentation, with validation. To explain the reasons for their entanglement in practice, I propose a Weberian ideal-typical model of modeling and simulation as roles in practice. I examine an approach to mitigate the Duhem problem for verification and validation that is generally applicable in practice and is based on differences in epistemic strategies and scopes. Based on this analysis, I suggest two strategies to increase the reliability of simulation results, namely, avoiding alterations of verified models at the validation stage as well as performing simulations of the same target system using two or more different models. In response to E.Winsberg's claim that verification and validation are entangled I argue that deploying the methodology proposed in this work it is possible to mitigate inseparability of V&V in many if not all domains where modeling and simulation are used.

Keywords: modeling; computer simulations; verification; validation; experimentation

[1] Accepted to Minds and Machines

[2] Email: vspron@fnal.gov

[3] Fermi National Accelerator Laboratory is operated by the Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

Introduction

Computer modeling and simulations, their epistemological value (Frigg/Reiss, 2009) as well as their analogies and differences with experimentation long have been the subject of lively debates in the philosophy of science (Humphreys, 2004; Winsberg, 2010; Weisberg, 2013; Parke, 2014; Parker, 2009; Morrison, 2009; Morrison, 2015; Barberousse, 2009). Simulation often has been regarded as being an intermediate between experimenting and theorizing and even displaced from the center of knowledge production as well as enhancing human cognitive capacities (Durán 2018). Many of the strategies experimentalists use to obtain reliable results also have been found in simulationists' practice. At the same time, some authors argue that simulation is a type of theorizing or construction of representative models. Based on a view (Suppes 1960, 290) of mathematical models as sets of objects and relations as well as their operations, I consider a distinction between the former in simulation, which I call objects or object structures, and the latter I call processes. I suggest that a similar distinction can also be made between simulation and modeling that is actually reflected in practice. In such a view, simulation of an application can be related to experimenting with structures representing ideal objects; in such a sense, simulation is a type of experimenting, and all experimental strategies (Franklin, 2012) apply to it directly. However, in modeling, the focus is on creating the embedded models of processes down to the level of the most elementary ones occurring in the target system. In this sense, modeling is a type of theorizing, and all the strategies applicable to the development of theories are employed in modeling. Thought experiments can be attributed to a special epistemologically interesting case of theoretical work; however, this kind of epistemic practice is methodologically different from the creation of the mental model (which is how theorizing is understood in this work). We concede that computer simulations, while methodologically similar to thought experiments, nevertheless can be deemed epistemically superior than the latter (Chandrasekharan, 2012). It seems plausible that, particularly due to the distinction between modeling and simulation made above, a distinction of the same kind should be made between the verification and the validation of computer models, the former being grounded on practices and strategies of modeling, while the latter is grounded in simulation.

Among approaches to definitions of modeling and computer simulations, one can identify two most frequently occurring definitions in the literature; those terms either are 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 a computer (Winsberg, 2010); he also associates simulation with a model itself as those 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 greater 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 the 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 ambiguity in the definitions of modeling. For example, (Morrison, 2009, 47) suggests that "the computational resources of simulation […] make[] it different from modeling" and ascribes simulation to "a type of "enhanced" modeling" (Morrison, 2015, 226). I concur that "we have no justifiable reason to assume that, in these types of cases, experiment and simulation are methodologically or epistemically different" (Morrison, 2009, 43) including in the sense that simulations can surprise as akin to experiments (Parke, 2014), however, in this paper I attempt to contest the view that similation can be seen as modeling. M. Weisberg delineates modeling as "the indirect study of real-world systems via the construction and the 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 are similar. This latter interpretation is also supported by Weisberg's description of simulation as "computing the behavior 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 clearly can envision that it is necessary to both construct the imitating process to accomplish such a simulation (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 the parameters covering all the parameter space of the problem). ((Winsberg, 2010) also describes simulation as one run of a computer code, which is similar to our understanding of simulation). Even if we look into the construction of a computational code of the imitating process, we can clearly discriminate between embedded elementary (or lower-level) processes (like the 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 ones into a more general framework.

Rather than (Hartmann 1996, 83), who defines a process as solely some object or system whose state changes in time, I distinguish between a simulated object as a model of a complex concrete or ideal system in a particular state (used in a particular application) and the elementary (or lower-level) embedded processes in it as evolving in time consequences of states of its subsystems. In such a case, the process in a concrete complex system under scrutiny (application) is imitated by another process in simulation, while the imitating process model and constituting it entrenched components are created by modelers; thus, simulationists design models of concrete applications out of pre-verified lower-level models-components and experimentally explore the behavior of such imitating models. In this approach, verification is associated with modeling, i.e., the creation of models, while validation is associated with experimenting with the applications premised on implemented more elementary models. Additionally, contrary to (Weisberg 2013, 24), who distinguishes concrete models, mathematical models, and computational models, in our view, every mathematical model can be associated with a computational representation, whereas there cannot be a computational model without a possibility to present it in a mathematical structural form. Here, I will consider computational mathematical models, i.e., mathematical models and their computational representations that are used in simulations; both types of models described above — the composite and the lower-level elementary ones — belong to this type of model.

Modelers and simulationists as roles in scientific practice

As an example from particle physics, a model of particle interaction with another particle or a nucleus is more elementary (associated with a lower level) 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 for this relative simplicity is that an elementary process occurs at a lower level of system organization. Building higher-level structural models can be recognized as a separate kind of activity and expertise (epistemic scope) from both running simulation code and the construction of models of elementary processes; however, from the point of view of scientific practice, those practitioners who run simulations can also be involved either in the building of structural models or adjusting existing ones to their needs in their different roles. This allows considering modelers and simulationists as ideal types in the Weberian sense, as will be described below, which is one of the reasons I discuss the higher-level structural model construction as a part of simulations.

One possible way to elucidate the necessary distinction is to scrutinize epistemic scopes of practitioners in the field. Counterintuitively, an increase in the level of model organization does not always entail a respective increase in complexity and the scope of required knowledge but instead shifts the scope of that knowledge; such an increase usually implies alterations in the scope. A modeler of processes, who is supposed to build his or her lower-level models, has to be familiar and able to apply all the mathematical structures pertinent to the models he or she builds down to the level of the most elementary laws (which can be phenomenological). In contrast, simulationists who are not process modelers and thus construct and apply higher-level (simulation) models of composite applications often are not required to have an extensive acquaintance with the lower-level 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[4]. Lower-level models are usually provided to them in the form of ready-to-use computational procedure units suitable for incorporation into more complex composite models. Examples can be those modelers who create models of interactions of particles with other particles and nuclei (for particle beam interaction codes), developers of models of dark matter particles for cosmological simulations or creators of cloud models for climate simulations. Rather than that, simulationists have to envision the structure and macroscopic designs of the complex system they intend to construct. Simulationists also have a general comprehension of how the relevant model parameters affect the behavior of the modeled system and explore the influence of those parameters. Examples can serve those who develop energy deposition codes for accelerator applications (that incorporate models for interactions of individual particles with nuclei), simulations of evolution of galaxies (that incorporate interactions of dark matter particles), or climate, for instance, weather front simulations (that incorporate models of clouds). Therefore, the epistemologically hierarchical relation between simulationists and modelers resembles that of experimentalists and instrumentalists in experimental practices (Pronskikh, 2018).

Thus, there exists an apparent controversy in the previously discussed definitions of modeling and simulation, leaving room for more rigorous characterizations of both domains capable of addressing whether simulation is the construction of a model, is computing behavior, or both (the entire computational study of a particular system). I argue that such controversy nevertheless can be resolved provided that one considers differences in both aspects of the practices (construction of elementary models, construction of composite models, their explorations) and the epistemic scopes of the corresponding practitioners (knowledge of how to construct elementary process models on the one hand and knowledge of how to construct a real-world target model based on a set of prebuilt embedded elementary models and numerical experimentation on the other hand). 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 suggest a resolution of the controversy by defining modeling as a creation of computational mathematical models of elementary processes and by defining simulation as a creation of composite computational models (those embedding elementary models) in the course of numerical experimentation.

A study of radiation effects, which required the modeling of quadrupole magnets and their field in the LHC accelerator (LHC, 2004), can serve an example of this distinction. To describe and study the propagation of protons in a complex magnetic field of an accelerator, one needs to ascribe magnetic fields to individual magnets, dipoles, quadruples, etc. that altogether constitute the entire magnetic system of the accelerator and create its magnetic optics. The magnets are characterized by shapes, sizes, and relative arrangements and altogether represent the organization of the accelerator’s technical real-world at its highest-order scale. The construction and study of the accelerator’s computational model thus can be regarded as simulation. In the course of such a simulation (model creation and particle propagation numerical study), practitioners usually run computer models a multitude of times and supply them with various sets of parameters to cover all the parameter space under scrutiny and thus to meet an optimal regime. As I shall discuss in the next paragraphs, such a simulation is grounded in strategies that possess many features of experimental rather than theoretical practice, 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 incorporating elementary models of magnetic field creation by charged particles that are governed by Maxwellian equations. Maxwellian law is the lowest, the most elementary level of the accelerator system organization, and its computational implementation serves as a building block of the higher-level simulation model of the entire accelerator. However, a representation and solution of Maxwellian equations constitute inferences pertinent to theoretical strategies. Therefore, development of computational procedures calculating solutions of equations of electrodynamics for an arbitrary set of initial conditions is deemed modeling for the purpose of my argument. In the practitioners’ sense, the expertise required for creating models for arbitrary magnetic fields is different from that utilizing the models of fields to simulate particle transport in concrete accelerators both epistemologically (different epistemic scopes), and methodologically.

Applicability of the epistemology of experimentation

[4] Similar to how the use of a TV set or a phone does not require knowledge of its internal organization.

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 employ common sense considerations to verify that their results are consistent; however, more often, they validate (or "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 sometimes can also be measured experimentally and compared to a simulation output. Nevertheless, matching their outputs to analytical solutions is not generally available to them due to both the complexity and opacity (Humphreys, 2004) of the systems they simulate and the differences 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 simple magnetic structure and then verify how its computational representation is programmed; for such models, either other computational models or analytical solutions usually exist for comparison.

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

One more way that Franklin's experimental epistemic strategy (Winsberg 2010, 44) is 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 elementary processes and elaborating such models as another instrumental theoretical model 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 the predictions outside the relevant data range. Rather than in 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". Particle beam dynamics in accelerators, for example, can be simulated by independent simulation codes, such as MAD (Deniau 2018), which is used to simulate beam dynamics and optimize beam optics, or Synergia (Spentzouris 2004), which simulates collective effects, including space charge and wake fields in the machine. These codes exploit completely different high-level concepts and assumptions (and can be associated with two different apparatuses in experimentation). Nevertheless, all the codes used in the field reveal identical understandings of the low-level Maxwellian electrodynamics, which belong to the scope of modeling of lower-level processes.

In the case of different models of the same "real-world" object, such as different accelerator (or climate) codes supplied with sets of input parameters (lattices or subgrids), these 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" model is not obvious, as the simulation 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 (such as 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 many differences as possible, or at least an independent model, including representations of the process under scrutiny, through different approaches to describing the same reference process by 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 a common practice to increase the reliability of simulation results. Agreement of the simulation results obtained with two or more different models of the same target process or system can serve as an indication of the adequacy of the simulation and its relevance and reliability.

Modeler and simulationist as ideal types

Despite 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 intertwined in the scientific practice. This outcome implies that individual practitioners often are engaged in both kinds of activities. The Weberian theory of ideal types (Weber, 1949, 49) can be involved to represent this. Let us assume that the simulationist and modeler are two ideal types whose differences on epistemic and ontological grounds are discussed extensively 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 three domains in one way or another.

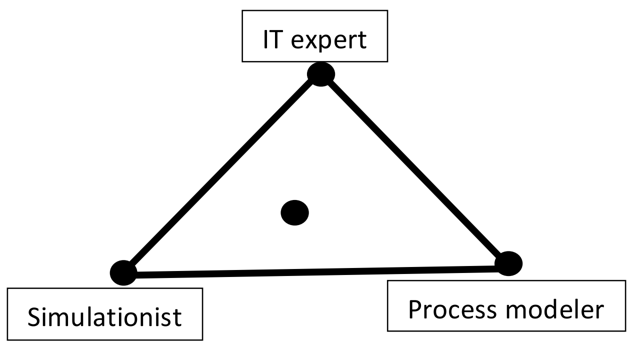

Figure 1. Triangle of the ideal types in simulation practice (vertices of the triangle) and the expertise of an actual practitioner (the dot inside the triangle).

There is, therefore, one more significant similarity between experiments and simulations, which results from the distinction discussed above. Experimentation requires detailed knowledge of the underlying instrumental and technical theories on which the functioning of the apparatus is based for the sake of data interpretation regarding high-level theories. However, being encoded in computational procedures, elementary process models can serve as examples of procedural knowledge; in addition to their representational role, they can function as recipes of *how* one can calculate a quantity, i.e., an instruction of 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; therefore, simulationists, to investigate a model of simulations, have to choose models of processes and composite models (simulations) and construct out of them as a composite application whose properties they intend to explore. Thus, the epistemic scope of the simulation of a thunderstorm does not necessarily encompass interactions of individual molecules in a cloud, and the 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 prebuilt and pre-verified by modelers elaborated computational procedures. This explains why a simulationist can practice higher-level simulations successfully 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 the necessary models (codes, lattices, and input decks), an advanced IT user can start experimenting with them and producing 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 — which is usually comprehensible by an experienced practitioner from a technical professional field who, for a particular reason, demands turning to simulations; this could 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 distance between applying ready sets of procedures and competently experimenting with models — simulation — because one needs to comprehend what kind of models of processes exert the effects found in outputs in the course of simulations. Such an understanding is crucial for simulationists to be able to adequately interpret outputs, that is, why a practitioner needs to communicate with modelers on the way from an IT user to an experienced simulationist. Modelers provide models of lower-level processes with access to a limited parameter space of variables, not implying the knowledge of models' internal mathematical structures. Modelers can also create and provide beginner simulationists with example decks (sets of model parameters for process models used), lattices and subgrids (structural object models), representing solutions of simple problems, which serve as aids 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, with all three being separate roles. B provides A with "low-level" process models (usually in the form of procedures) and instructions on how to use them, and C provides A with supplementary computer codes (scripts) and instructions on how to employ them.

Another path in Figure 1, that from an IT user to a process modeler, usually occurs 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 more formal education. However, modelers of processes are usually required certain IT skills to develop their procedures. In the course of their everyday practice, they often engage in construction of low-level models of processes and incorporate those models in the composite simulations of higher-level "real-world" systems and applications. Certain IT expertise is needed to accomplish this outcome. Thus, an actual modeler of low-level processes often is also a "higher-level" simulationist, whereas a simulationist, even starting as a pure ideal type, usually acquires certain interactional expertise (Gorman, 2010) in understanding low-level models through communication with modelers of processes. Nevertheless, even concurrent and alternate practicing "low-level" modeling and "higher-level" simulation roles do not entail epistemic entanglement; therefore, note that distinction is essential for differentiation between verification and validation.

Verification and its relation to modeling

Verification can be defined as the process of determining whether a computational representation approximates the solutions to the differential equations of the mathematical model of a process (in the case of Monte Carlo models, in particular). Verification (code verification) usually is understood as either code verification, i.e., search for and fix mistakes in a computer code or solution verification (estimating solution errors and accuracy of the code input and output). The 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 actually is verified, according to that definition, is that an already constructed model is implemented correctly in the code (accurately solved), as the code is a computational representation of such a model (its conceptual description). Due to the previously discussed applicability of inference to low-level elementary process models, analytical checks of both the algorithm implementation and the solution are available in most cases. However, the definition of verification can be extended. To verify model's implementation, in addition to comparing it to an analytical solution, one can also verify correctness of the computational code using methods of software engineering and (in the cases when analytical solutions are unavailable) comparing the results produced by the model algorithm at different stages to the solutions produced by independent implementations in other computational systems which are verified independently. The example of the latter can be the verification of how calculation of certain integrals is implemented in the model code by comparing intermediate results to the integrals calculated, for instance, to those obtained by applying the conventional tools, like the Mathematica code.

The inference requirement turns out to be in agreement with a practitioner's statement that "verification addresses 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 context under scrutiny. The availability of such checks and tools as well as the inference strategy supports the association of verification with modeling in the sense discussed in this work. Referring to the aforementioned studies, solutions of Maxwellian electrodynamics equations can be matched step by step with outputs of the computational procedure; thus, its implementation is 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 the verification that is defined as the process of determining whether or not the output of simulation approximates the correct solutions to the differential equations of the original model, validation can be described as determining whether the chosen model is a good representation of the real-world system (Winsberg, 2010, 19). These two activities are separable not only in practice but also on epistemic grounds. With the above definition, verification clearly falls within the scope of modeling, which is a kind of theorizing. In contrast, validation refers to the study of how well a model implemented by modelers in a particular code is capable of describing a real-world system. For example, matching outputs to analytical (or any sort of independent solutions) solutions is not generally available to simulationists due to the complexity of the systems they simulate and the difference in their epistemic scopes with modelers. This outcome is possible for modelers, however, who work on models of elementary processes; for such models, either other models or analytical solutions or conventional tools often exist for comparison.

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 that an analytical solution is not generally available because the system under scrutiny belongs to the higher level of system organization (for example, accelerator or climate). There is one additional important feature of validation is its relation to experimental data. For such a system as an 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 a validation experiment. A definition of physics as an experimental science (i.e., heavily relying on empirical data) allows understanding the practitioners' statement that "validation addresses physics" (Oberkampf, 2004). Simulation, defined in this work as numerical experimentation with composite models of "real-world" objects, involving experimentation with parameters of computational code and other strategies common with experimentation (as discussed above), has certain connections with validation.

Referring to an example discussed in previous paragraphs, one needs to validate the simulation code versus the data obtained in the measurements of heat release in simple objects made of pure materials using certain particles with well-defined energies and distributions (validation experiments) to simulate heat release in a composite object irradiated by various particles. The 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), belonging to a different epistemic scope, and are not modified at that stage. Its relation to higher-level objects and its reliance on experimental data as well as the 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, the unavailability of analytic solutions, inferences, and independent conventional analogs makes them insufficient in the 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 the experimental strategies of exceptional importance in simulations and 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 is also applicable: "a legitimate and important factor, [which] may even be decisive, in determining the validity of an experimental result" [Franklin, 1994]. From the practitioners' viewpoint, it is employed when "validation is not feasible or practical," (Oberkampf, 2004) and purports adjusting computational model parameters so that the outputs match 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 differentiate simulation (higher-level) model parameters clearly, which are subject to alterations in the course of calibration, and elementary process (low-level) parameters, which are not. Oversight of this rule in certain cases can entail entanglement as discussed below.

Entanglement arguments

A number of arguments have arisen that support the viewpoint that verification and validation are entangled (Winsberg, 2014). For instance, (Jebeile, 2012) argues that verification and validation are "two phases [that] cannot be performed distinctively" and thus are entangled. One of the examples that can illuminate such an entanglement has been examined by (Lenhard, 2007) as implementation of the "Arakawa operator" (this example was shortly discussed above). (Jebeile, 2012; Lenhard, 2007) note a possibility of introducing distortions in the system behavior through so-called discretization schemes[5], when differential equations of a mathematical model are converted to difference (algebraic) equations with the aim of 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 methodologically separate the construction of such discretization schemes (low-level epistemic scope of modeling) from the application of such schemes for the simulation of higher-level models*. I do not assert that it is always practically feasible and concede that such entanglement can occur in certain practical cases; however, I maintain that verification and validation can be differentiated when scopes and strategies are separable and distinct. Distinction of scopes allows preventing "model success due to piecemeal adjustment" (Winsberg, 2010), which causes the entanglement or the Duhem problem for verification and validation. As an example, when a simulationist embarks upon a simulation of climate, for example, a thunderstorm, they have to have a stable model of clouds beforehand; that model of clouds and its tuning parameters should not be modified (or tinkered with in any possible way) in order to achieve a better agreement with any particular climate context. Otherwise, even if such a tinkering gives better agreement for a certain set of empirical data (the one used for model tuning), in other contexts the simulation that embeds such tuned parameters will perform worse or even fail. Also, in nuclear and particle physics, if one simulated energy deposition by a particle beam in a collimator of an accelerator, one should not tweak the parameters of an embedded lower level models (for example, nuclear density or spallation model) in order to achieve a better agreement with the heating of a particular collimator. Otherwise, the simulation code tuned in such a way will not be applicable to other tasks like the energy deposition by other particles, their energies or in other types of targets. The models of clouds or nuclei in the examples above were verified (and, sometimes, validated) using relevant procedures and data sets with well-defined uncertainties, and their modification in the context of a particular higher-level application is unacceptable not to compromise the overall quality of the simulation code.

Ascribing validation of a computational mathematical model of a process to modeling as establishing and proving a mutual relationship between a mathematical model and its computational representation (such as the quality of the discretization), one can distinguish it from validation as experimenting with objects involving process models as provided by modelers. I concede that while the elementary processes occurring in the course of simulation are epistemically opaque (Humphreys 2004, 147-148), to simulationists, they do produce new knowledge that is not contained in the underlying process models (and, therefore, can surprise us (Parke, 2014); however, all the details of the processes taking place in a simulated system are not analytically predictable. However, the implementation of an equation in a computer code can be verified either analytically or using conventional tools or software

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

engineering approaches, even if the equation in question is physically unrealistic (Arakawa operator). Software engineering methods can be applied to verify model's computer implementation, which is an unambiguous translation of a mathematical structure to an algorithmic programming language. As the examples above suggest, simulation and modeling are different methodologically and are not necessarily entangled; the problem to distinguish whether the model or its computational representation that fails the case of a discrepancy between a simulation output and the real-world data can be addressed with a clearer epistemic and methodological distinction between them.
I suggest that models of lower-level processes, their computer implementation, and their code verification are open to direct inspection by modelers (theorists); practically relevant methods also exist to estimate numerical solution errors at this stage. However, 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 does 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, although true for simulation, does not bring about a verification and validation entanglement with necessity in general.

Separability of stages in Complex Simulations

To argue for the inseparability of the verification and validation stages—as well as looping in modifications of lower-level model elements, on the one hand, and, on the other hand, model adjustments to empirical data—, Winsberg (157) suggested the following life cycle for a typical complex model in climate sciences:

1. M1 (Model 1) is created based on well-established physical principles.
2. M1 is discretized and implemented computationally using S1 (Scheme 1).
3. Outputs of S1 are compared to empirical data.
4. If there is a lack of agreement, either M1 or S1 are changed based on parametrization, physical intuition, phenomenology, etc.
5. Model M2 is discretized with S2.
6. The process is repeated starting from Step 3.
7. Eventually, the model is sanctioned.

Several points can now be clarified based on the arguments developed in this work. First, both M1 and S2 can be verified in the sense that their computational implementations can be checked separately by modelers using analytical solutions, conventional tools, or software engineering methods, as suggested in the previous sections. Second, the above scheme suggests that M2 and M1 are independent despite being embedded in the same larger system. At one stage, a model of the smallest individual constituents and their interactions with others is developed; at the next stage, these constituents are embedded into models of interactions on a larger scale. This is also easy to imagine in the context of climate science. A model of a weather front (M2) must embed models of individual clouds (M1). Meanwhile, individual measurements taken by many meteorologists are required for the very existence of the concept of a weather front (Galison 2003). This paper's argument does not deny that entanglement may arise in certain cases; it does, however, suggest that the parametrizations for M1 and M2 are different and, in several discussed contexts, can be validated (i.e., their parameters empirically chosen) separately, without resorting to piecemeal adjustments. To accomplish this, step 6, which involves such adjustments and entanglement—an enforced procedure—must be avoided.
For instance, weather simulations can comprise two scales in which a small grid size allows for a description of each possible cloud system on the globe, cloud system models are embedded within a simulation of the weather that we experience on another scale (for instance, a thunderstorm). The same logic applies to multiscale nanoscience simulations where higher level models rely on embedded lower level ones. It can also be applied to cosmological simulations in which models of particles of dark matter are created at the lower level (each of which represents, for instance, 200 billion solar masses), and the evolution of large-scale galaxies in the universe containing 8.6 billion such particles is simulated at the higher level (Trujillo-Gomez 2011). Hence, in most domains where large scale simulations are used—nuclear and particle physics, nanoscience, cosmology, climate science, etc.: 1) M1 and M2 are not independent but are either included in a larger model or embedded in one

another, 2) M1 and M2 can be verified separately and validated using separate sets of empirical data, and 3) The "back and forth" model (in which the empirical sample intended for M2 validation is used to adjust M1) can lead to entanglement and should be avoided whenever possible. To return to the climate science example, one should use the empirical data of individual clouds to validate cloud modes and then use weather front data to validate the weather front model. However, while adjusting weather front model parameters, one should not tweak the individual cloud model parameters obtained at the previous step *even if* that improves the overall agreement with the data. This is because, while it can improve the fit to the available data, it will generate worse results when applied beyond the scope of the available empirical data set.
This paper does not argue that such separate validations are always sufficient to provide sets of parameters that ensure the best fit for all available data sets; such a separate parameter determination may not be universal. Nevertheless, I argue that because of alleviated entanglement, such parametrizations can be more robust. Although piecemeal adjustments yield better sets of entangled parameters that allow for a description of the systems used for validating the models, separate validations of M1 and M2 (i.e., the absence of Step 6 in the aforementioned scheme) can offer parametrizations that permit models to be extrapolated to the range of parameter values that go beyond those used to validate the models.

Hence, the proposed life cycle of a complex simulation (which, in most cases, can be realized practically, as described, for example, in the section "Entanglement Arguments") is as follows:

1. M1 is created on well-established lower-level physical principles.
2. M1 is verified (the code is checked and any available intermediate analytical solutions are taken advantage of).
3. M1 is discretized and implemented using S1.
4. S1 is verified (the code is checked and any available intermediate analytical solutions are taken advantage of).
5. The outputs of S1 are compared to the empirical data set E1 (validated).
6. M2 is created (sometimes by embedding pre-validated M1) (in the form of S1).
7. M2 is verified (the code is checked and any available intermediate analytical solutions are taken advantage of).
8. M2 is discretized and implemented using S2 (embedded S1 remains intact).
9. S2 is verified (the code is checked and any available intermediate analytical solutions are taken advantage of).
10. Outputs of S2 are compared to the empirical data set E2 (validated).
11. M2 is modified to better accommodate E2. Return to Step 6.
12. Eventually, the model is sanctioned.

Although in the practice of simulation in constructing models often follows the "back and forth" model in many domains—in the sense that parametrizations, phenomenology and intuitions are added at each iteration repeatedly over the course of the construction—I argue that separating the stages of model creation as well as the well-established, lower-level elements of models, and the levels that are variable and adjustable to particular contexts, is not only possible but also methodologically necessary. The apparent separation can often be achieved in most large-scale simulations despite many of them relying either on various grid parametrizations, like cloud models in climate science or parameterizations of stars in cosmological models, or on deploying fictions akin to force softening and viscosity because validation of different scales relies on non-overlapping sets of data. My proposal is consistent with Hasse and Lenhard (Hasse 2017), who maintain that "the adjustment of parameters limits the applicability of models. The model will often only be useful for describing scenarios which are not "too far away" from the scenarios that were used for the fit." They argue that "the question how far a model carries beyond the range where it was parameterized is closely related to the quality of the theory behind it ". Separate parametrization of different levels such as M1 and M1 suggested in this work is intended not to compromise the quality of the theory by excessive parametrizations through piecemeal adjustments.

Conclusion

This paper shows that verification and validation entanglement is not indispensable and, although it can arise in certain practically relevant cases, not universal. To justify this argument, I suggest that a distinction be drawn between modeling (defined here as construction of low-level mathematical computer models of processes) and simulation (construction of higher-level models of composite objects and processes assisted by numerical experimentation). These two aspects differ in their epistemic scope, the former implying theorizing with the involvement of inference and reliance on analytical approaches, conventional tools, or software engineering methods, and the latter suggesting numerical experimentation based on Franklin's epistemic strategies. I contend that such a distinction, whenever practically feasible, can mitigate the implications of the Duhem problem for verification and validation. I show that despite being epistemically distinct, modeling and simulation constitute roles and ideal types in practice, and can be performed by the same practitioners. The latter often entails their "role entanglement." I suggest that for this distinction to hold, as well as to mitigate both the verification and validation entanglement and "piecemeal adjustment" of models undermining reliability, low-level elementary process models that have undergone verification must not be altered in the course of higher-level validation.
I argue that such a separability can be achieved in many domains wherein simulation and modeling are deployed. Because of the methodological affinity of simulation to experimentation, simulation in the sense proposed in this work (i.e., running simulation code) with two or more different models, which serve different representations of the same target system, can be considered tantamount to different experiments with the same target system. Therefore, simulations with several different models that yield consistent results can be considered a strategy aimed at increasing the reliability of results.

Acknowledgements

The author is indebted to two anonymous reviewers for their careful reading of his manuscript and their many insightful comments and suggestions that helped improve the paper. I would like to thank the audience at Models and Simulations 6 conference for useful comments and discussions. I am grateful to Dr. Eric Winsberg and Dr. Arkadiy Lipkin for valuable feedback on the earlier versions of the manuscript. Fermi National Accelerator Laboratory is operated by the Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

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.

(Brunning, 2002) O. Bruning, R. Cappi, R. Garoby, O. Grobner, W. Herr, T. Linnecar, R. Ostojic, K. Potter, L. Rossi, F. Ruggiero (editor), K. Schindl, G. Stevenson, L. Tavian, T. Taylor, E. Tsesmelis, E. Weisse and F. Zimmermann, LHC Luminosity and Energy Upgrade: A Feasibility Study. CERN-LHC-Project-Report- 626, CERN.

(Chandrasekharan, 2012) Chandrasekharan, S., Nersessian, N.J., & Subramanian, V., Computational Modeling: Is this the end of thought experimenting in science? In J. Brown, M. Frappier, & L. Meynell, eds. *Thought Experiments in Philosophy, Science and the Arts.* (London: Routledge). pp. 239-260.

(Gorman 2010) Gorman M.E. Trading zones and interactional expertise. Creating new kinds of collaboration / M.E.Gorman (ed.). – Cambridge: MIT Press. – 2010.

(Deniau, 2018) Deniau Laurent, Grote Hans, Roy Ghislain, Schmidt Frank. The MAD-X Program (Methodical Accelerator Design) Version 5.04.02 User's Reference Manual, European Laboratory for Particle Physics, 2018, URL= http://madx.web.cern.ch/madx/.

(Durán 2018) Durán Juan M. Computer simulations as a technological singularity in the empirical sciences, *The Technological Singularity - A Pragmatic Perspective*. S. Armstrong, J. Miller, R. Yampolskiy, V. Callaghan (eds.). Springer Verlag.

(Franklin, 1994) Franklin, Allan. How to Avoid the Experimenters' Regress, Stud. Hist. Phil. Sci., Vol. 25, N. 3, pp. 463-491.

(Franklin, 2012) Franklin, Allan. "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/>.

(Hasse, 2017) Hasse H., Lenhard J. Boon and Bane: On the Role of Adjustable Parameters in Simulation Models. In: Lenhard J., Carrier M. (eds) Mathematics as a Tool. Boston Studies in the Philosophy and History of Science, vol 327. Springer, Cham

(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

(LHC, 2004) Roeck A. De, Gianotti F., Morsch A., Pokorski W., Simulation physics requirements from the LHC experiments, CERN-LCGAPP-2004-02, 2004.
(Morrison, 2009) Morrison, Margaret, Models, measurement and computer simulation: the changing face of experimentation, Philos Stud (2009) 143:33–57.

(Morrison, 2015) Morrison, Margaret. Reconstructing Reality: Models, Mathematics, and Simulations New York: Oxford University Press, 2015.

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

(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.

(Pronskikh, 2018) Pronskikh, Vitaly. Linguistic Privilege and Justice: What can we learn from STEM?, Fermilab publication FERMILAB-PUB-18-013.

(Frigg/Reiss:2009) Frigg, Roman and Julian Reiss. 2009. The philosophy of simulation: hot new issues or same old stew?, Synthese 169:593-613 .

(Suppes, 1960) Suppes, Patrick. "A Comparison of the Meaning and Use of Models in the Mathematical and the Empirical Sciences, " Synthese 12: 287-300.

(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.

(Trujillo-Gomez, 2011) Trujillo-Gomez Sebastian, Klypin Anatoly, Primack Joel, and Romanowsky Aaron J. Galaxies in ΛCDM with Halo Abundance Matching: luminosity-velocity relation, baryonic mass-velocity relation, velocity function and clustering, arXiv:1005.1289v3 [astro-ph.CO], 23 Sep 2011.

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

(Winsberg, 2009) Winsberg, Eric. Computer Simulation and the Philosophy of Science. Philosophy Compass, 4: 835-845. doi: 10.1111/j.1747-9991.2009.00236.x.

(Winsberg, 2010) Winsberg, Eric. *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/>.

(Winsberg, 2018) Winsberg, Eric. Climate Science and Philosophy. Cambridge University Press.