Chapter 6
Iteration Unleashed. Computer Technology in Science

Johannes Lenhard

Abstract Does computer technology play a philosophically relevant role in science? The answer to this question is explored by focusing on the conception of mathematical modeling, how this conception is modified in computational modeling, and how this change is related to computer technology. The main claim states that computational modeling is geared towards iterative procedures which replace complicated or even intractable integrations. This shift is not a mere technicality, but presents a major conceptual transformation of modeling. At the same time, it is argued, the form and function of iterative procedures are dependent on the available computer technology.

A number of different cases, among them the Schrödinger equation in quantum chemistry and Markov chain Monte Carlo methods, are discussed that span iterative strategies in pre-computer, mainframe, and desktop-computer time. In methodological respect, bracketed iterations are discerned from exploratory iterations. Opacity and agnosticism are discussed as epistemic ramifications of the shift towards iterations.

1 Introduction

Computer technology is in widespread use in a great variety of contexts – scientific ones as well as non-scientific ones. Many people, asked to point out the signature technology of the recent decades, would single out computer technology – what makes it a prime example when one investigates the impact and importance of technology for science.

However, the term computer technology can be used in two senses. One is the technology of the computer as a machine, its hardware, like input and output devices, the integrated circuitry, or its principal design architecture etc. The other sense is technology that crucially depends on or entails computer technology in the first sense. Examples reach from the enormous Large Hadron Collider in particle

J. Lenhard (✉)
Philosophy Department, Bielefeld University, Pb. 100131, 33501 Bielefeld, Germany
e-mail: johannes.lenhard@uni-bielefeld.de

S.O. Hansson (ed.), *The Role of Technology in Science: Philosophical Perspectives*,
Philosophy of Engineering and Technology 18, DOI 10.1007/978-94-017-9762-7_6

physics over computer tomography in medicine to the tiny dimensions of DNA arrays. The sheer range of different technologies that rely on computer technology testifies its relevance. The present chapter will mainly deal with computer technology in this second sense, though it will take into account that the second sense is not independent of the first one.

The key question is: Granted the importance of computer technology what is its impact on science? This question receives its significance when compared to an influential view, one could call it the standard account, that acknowledges the *practical* importance of computer technology, but more or less denies that it has a *philosophically relevant* impact on science. It holds that computer technology is a versatile and powerful tool of science, a means to accelerate and to amplify research and development, and also to boost all sorts of applications. According to this account, however, science would not have changed much in essence. Basically, computers do carry out computations and store and handle data. Both activities have been present in modern science all the time, hence do not strictly require computer technology. What computers do, so the standard account, is merely to speed-up data-handling and computation to astonishing degrees. Such a viewpoint denies a *philosophically* relevant role of computer technology in science.[1]

The present chapter will argue that this standard view is misleading, because computer technology does not present merely a new costume for basic mathematical calculation, rather it changes the very conception of mathematical modeling. Hence it changes the entire game. Consider the impact of the printing press. It would be grossly misconstrued if one would take it as a mere acceleration of writing by hand. What follows in this chapter is the attempt to convince you that this analogy, though somewhat grandiose, is an apt one.

Basically, computer technology leads to the automation of algorithmic procedures in a fairly wide sense. In his analysis of industrial economy, Karl Marx has pointed out that the key driver of the industrial revolution was not so much the power supply by the steam engine, but rather automation of artisan work, like the Spinning Jenny had achieved. Computer technology seems to hold a comparable position insofar as it automates formally or mathematically described procedures. However, such a comparison has to be based on firmer ground. This chapter describes and analyzes the impact of computer technology. What are its characteristics? What are its ramifications concerning the methodology and epistemology of science?

Section 2 takes "Representing and Intervening" as a starting point, the two major activities that Ian Hacking (1983) discerned in science. Hacking, among others, did much to revalue the intervention part and to confirm that both parts are philosophically on a par. It is an uncontroversial observation that most interventions into the material world, and experiments in particular, rely on technologies. Thus technology participates in the status of interventions. The significance of technology

[1]For instance, computing technology has not received an entry in the rich compilation of Meijers (2009), whereas diverse chapters discuss relevant examples.

for science has surely reached a new level due to computer technology. However, I want to put forward a stronger claim that holds that computer technology is interweaving representation and intervention and hence calling into question the distinction between the two.

Mathematical representation will serve as an instance. The function and form of mathematization changes with computing technology and this change is of utmost importance for the philosophy of science. This claim will be supported by an analysis of computational models and of the ways such models are constructed and implemented. In particular, computational modeling presents a fundamental transformation of mathematical modeling. This transformation can be characterized by the methodological notions of iteration and exploration, and by the epistemological notions of opacity and agnosticism. This transformation affects the perceived structure of the scientific endeavor.

The thesis will be detailed by discussing iteration, exploration (in Sect. 3), opacity, and agnosticism (Sect. 4) as major features of modeling under the conditions of computer technology. The upshot of the analysis will be that computer technologies and the ways they are employed in producing data, identifying phenomena, analyzing them, and making predictions, are mutually re-enforcing. Computer technology thus canalizes the trajectories of science, not in the sense of determining them, rather making some of them attractive. Section 5 will address the significance of infrastructure for how science is organized. The concluding section will indicate potential lessons one might draw for the study of the sciences.

2 Representing and Intervening

In his classic introduction to philosophy of science, Ian Hacking discerns two major scientific activities, those of representing and intervening (1983). We will use this differentiation as a starting point to assess the significance of computer technology.

One of Hacking's main points is to show that the (then) current tendency in philosophy of science, namely to see the representation part as the philosophically more relevant part of science, is an unwarranted prejudice. Both parts – representation and intervention – play their indispensable role; Hacking's famous slogan for instrumental realism – "If one can spray them, they exist" – may suffice as an example where he stresses the relevance of the intervention side. Of course, Hacking was by far not the only philosopher of science who pointed that out. The whole movement of so-called 'new experimentalism' more or less gives the activity of intervening center stage in science. It is uncontroversial that technology plays a major part in interventions, be it on the scale of laboratory experiments or that of engineering. Hence the significance of technology for science will in general be positively correlated with the valuation of interventions.

In his book Hacking forcefully grants intervention its place, but a place still somewhat derived from representation. The overarching view is rather one of "homo

depictor " than the "homo faber".[2] The assessment of computer technology will lead us to place representation and intervention on a par, indeed to see both as deeply intertwined aspects of one activity. The following premise is surely justified: The significance of technology for science has reached a new level due to computer technology, because this type of technology is so widespread and versatile. So wherever technology comes in, say for example in a measurement apparatus, it is not unlikely that some computer technology is included. Admittedly, this claim leaves the question open what the characteristics of computer technology are. My point is not merely that this technology amplifies the share of the intervention part in science, rather computer technology questions the boundary between representing and intervening.

Representation has to do with identifying and displaying structures, and with creating a picture that helps to understand how phenomena come about and which factors contribute in what way to their dynamics. Computer technology, or more precisely, the use of computer technology in science, undermines this viewpoint. Computer models often do not represent what is out there, but play an essential role in transforming the subject matter and in constituting the phenomena. Sociologist of science Donald MacKenzie, for instance, supports a performativity thesis when he argues that computer models function in financial economics as: "An Engine, Not a Camera" (2006).

A classic topic in philosophy of science is that observations are always influenced by theory ('theory-ladenness'). Scholars like Paul Edwards rightly add that many data are 'model-laden' – his example are highly processed satellite data about the earth's climate (Edwards 2010). The present chapter argues that a similar view is valid in computational modeling more generally. The methodological and epistemological characteristics of computational modeling lead to an interweaving of representation and intervention.

3 Computational Modeling

Mathematical modeling has played an influential role in many modern sciences. A main impact of computer technology on science, or on scientific practice, was (and is) that it contributed to the birth of a new type of mathematical modeling, namely computational modeling. This new type is dependent on computational power and thus is tied to computer technology. There are further dependencies of technological nature, as we will see, insofar easy access to the machines, networked infrastructure, and the form of input and output interfaces matter.

[2]See Hacking 1983, 132. I would like to refer to Alfred Nordmann 2012 who argues about the relationship of representing in intervening in Hacking and about the broader significance for the philosophy of science and technology.

Computational modeling nowadays is employed in a great variety of sciences. It is quite obvious that the scope of mathematical modeling (in the dressing of computational modeling) has been enlarged to an astonishing degree. This observation urges us to ask how we can philosophically characterize computational modeling. There is a methodological and an epistemological part of the answer. Both parts point to the key role that technology plays in computational modeling.

3.1 Iteration

A preeminent feature of computational models is that they employ iterative strategies. Let us take quantum chemistry as an example and start with a look at the British mathematician and physicist Douglas R. Hartree (1897–1958). He was ahead of his time when he conceived computer technology and mathematical methods as twins. He combined great mathematical skills with a passion for tinkering and automating calculation procedures. He is mostly remembered for the anecdote, though one that actually happened, that he built a working copy of Vannevar Bush's Differential Analyzer out of Meccano parts that he diverted from his children's toy inventory. But he deserves a fuller appreciation in the context of our investigation. Hartree was not only an expert in computing technology in general, but was a very early and ardent follower of digital machines in particular. He conducted pioneering work with the ENIAC and the expertise he gained there was essential input for the development of the EDSAC at Cambridge, UK.

Early on, Hartree realized from his experiences with the ENIAC and its general-purpose programmability that digital computing would not merely make computation faster, but would demand to adapt mathematical modeling to the technology: "One thing which I learnt from practical experience of handling the ENIAC is the importance of taking what I might call a 'machine's-eye view' of the problems as furnished to it; and this is not so easy as it sounds." (1984 <1947>, 22) These problems covered technical issues like compiling a computer program that todays compilers and software languages solve automatically. The most important problem, however, was and is how mathematical problems should be formulated adequately. Although it is obvious that such formulations somehow have to reflect the technology available, it is not straightforward what that means for mathematical or computational modeling. High-speed machines were going to change methods and problems together, as Hartree expressed: "It is necessary not only to design machines for the mathematics, but also to develop a new mathematics for the machines." (1949, 115)

Hartree figured that iterative methods would suit particularly well to high-speed machines. Shockwaves, for instance, originate as singularities in solutions of a non-linear partial differential equation (PDE). If one replaces the PDE by a finite difference version, Hartree reasoned, one would need (under particular conditions) 200,000 multiplications to calculate an approximation to a practically useful degree. This condition had made such procedures impractical for extant devices. With the

electronic digital computer, however, such iterative procedures had become a viable option and even a favorable way to go as iteration is exactly what the electronic digital computer is good at. Hartree foresaw a canalizing effect – mathematical modeling would, and should, develop in a way that adapts to the strengths and weaknesses of the computer's capability (1984 <1947>, 31).

Hartree was especially prepared to acknowledge the prospects of iterative methods because he had pioneered them in a half-automated setting even prior to the digital computer. In general, iterative processes are very suitable for mechanization – repeating similar operations – and Hartree was inventive to design an iterative procedure that navigated between technological feasibility and modeling accuracy, drawing on mechanized and graphical integration methods. This procedure was the starting point for the now prominent Hartree-Fock method in quantum chemistry. It is worth to view at this method because it displays an iterative strategy in the context of a computing technology prior to the electronic computer, even if it became popular with the digital computer (for more historical detail as well as more quantum chemical context, cf. Park 2009).

Hartree's work in quantum chemistry, or chemical physics, can be regarded as important early outsider contribution. The key problem of quantum chemistry is to solve the Schrödinger equation which contains – in principle, at least – the information about the electronic structure of atoms and molecules. This equation, alas, is an epitome of complexity and analytical solutions are practically impossible. Hartree conceived the challenge as one of creating a different mathematical strategy, not oriented at analytical solution, but numerical adequacy. For him, extracting numerically adequate values out of the Schrödinger equation was a challenge to computational modeling. His strategy was to jointly develop a model and a procedure (including technology) that together would be practically operational. He devised and designed also a procedure (including graphical integration steps) that could be iterated semi-mechanically and that was fitted to a model. In fact, it was a mutual fit: It was a key conceptual step to consider procedure and model as mutually dependent on each other and to develop both jointly.

Let me briefly consider the iterative character. The Schrödinger equation entails an elevated level of computational complexity, because each electron interacts with all others, so that the values for electron potentials cannot be calculated independently one after the other. Hartree's approach placed a numerical handle on this problem by constructing a different kind of iteration in the following way.

One starts to calculate the value of the potential of one electron (counterfactually) assuming all others fixed as a given (ad hoc) field. In the next step, the first electron is assigned the value calculated in the first step and the second electron is regarded as variable, all others as fixed. Then the numerical procedure is iterated. The first series is completed when each electron has been singled out for one step. Then the whole series of iterations gets iterated, i.e. one starts anew a second round of iterations, now taking the results of the first round as initial conditions. At the end of the second series of iterations, the values of the potential of all electrons have been re-adapted. If they differ from the first series, the whole procedure is iterated again. And so on, until the values do not change anymore between two series of iterations.

This iterative procedure is now known as self-consistent field (SCF) approach. It creates a balance through iteration. Ingeniously, one has ignored different parts of the interaction potential in each step to find values that are mutually consistent, hoping that errors then cancel out. This balance substitutes the analytical solution and emerges through a co-ordination of technology and modeling.

One can criticize Hartree's approach as not principled (enough) and too artificial, as merely oriented at numerical and mechanical feasibility. For Hartree, numerical virtues outweighed the gaps in the theoretical justification of the model. Hartree's method, or more precisely its refined descendant, the Hartree-Fock method, got widely accepted when computers became a crucial instrument in quantum chemistry and when it turned out that the predictions obtained by this method were deemed good enough.

The SCF procedure does away with the interdependence of electrons that is the main obstacle for computational solution strategies. Thereby it deliberately ignores the factually existing mutual interdependence. From our point of view, computational modeling inevitably does some harm to the in-principle rigor of quantum – or any fundamental – theory. However, any modeling attempt has to do this at one or the other spot. This perspective ties in with the recent debate in philosophy of science about the role of models and especially about "models as autonomous mediators" (Morrison 1999). Computer models are not defined and specified solely in terms of some underlying theory – although the latter might be very important, as in Hartree's case. Rather model and computer technology are interwoven. Hartree's point was that he took numerical feasibility as a guiding criterion for model-and-instrument together. His major contribution in philosophical respect, I would like to argue, is not his mastering the challenge of numerical feasibility, but that he approached model and instrument as a compound.

Thus, we have seen the proposed canalizing effect at work. The particular example of Hartree was included because it started from a strong fundamental theory with a clear mathematical formalization, i.e. the Schrödinger equation, so that it is a 'hard' case for showing the relevance of computer technology. One might ask: Can't that interdependence be taken into account later – by way of error corrections? Yes, it can. But this addition then has to build on and improve the SCF-based result, i.e. it constitutes a modeling step that itself is based on the performance of a numerical procedure. Consequently, this correction would have to follow a more exploratory than principled reasoning. Today, a family of numerical strategies called post-Hartree-Fock methods follow this path. They came up much later however, when supplementing the iterative strategy by an exploratory component was a good idea. Exploration is related to technology, too.

3.2 Grappling with Exploration

Let me explain why I take the difference between the iterative and the exploratory component as important. Iteration is a straightforward operation and iterative

strategies are long known, they are for instance parts of the famous algorithms of Newton or Gauss. Computer technology can play out its strengths where modeling involves procedures that require so many iterative steps that they are infeasible without this technology. Additionally, there is also a basic difference in iterative strategies. Hartree's SCF method is *bracketed* in the sense that there is one command to conduct iterations until a certain state has been achieved which then is taken as result. Iterative strategies become *exploratory* when the procedures get interrupted and changed in light of the preliminary results. That is, one is not so much testing one single model, but rather testing and adapting a whole class of models in an exploratory manner. Therefore, feedback is an essential element in the modeling process. It is greatly enhanced by visual output, and it presupposes more or less easy accessibility so that researchers can afford to spend most of the time with preliminary versions of models.

My premise is: The exploratory add-on to iteration is a typical feature of smaller, i.e. highly available and cheap machines. Hence the exploratory-plus-iterative character of computational modeling is a feature of new computer technology since around 1990 that made these features available.[3] Let us approach the point of exploratory modeling strategies from an example where computer technology was not inviting. The next episode provides an illustration of how the digital computer opened up iterative strategies in quantum chemistry but at the same time also set limits regarding exploration.

Right from its invention in the late 1940s, the digital computer was related to the 'big science' of nuclear physics and the military. Peter Galison (1996), for instance, gives a vivid account of how Ulam, von Neumann, and others reasoned about computational strategies that would become tractable with a machine able to do high-speed iteration of elementary mathematical operations that otherwise would have demanded too high a number of human computers.

Other than physics, quantum chemistry was not 'big science' and to get access to computers was difficult in the 1950s. The case of Bernard Ransil who conducted the first so-called *ab initio* calculation might serve as an illustration. In 1956, Ransil worked at the University of Chicago in the quantum chemistry group of Robert Mulliken and Clemens Roothaan. At that time quantum chemical calculations normally followed a 'semi-empirical' approach, i.e. they inserted empirically determined values to simplify overly complex and difficult computations. For example, one would put in the value of a certain energy that is mathematically expressed in terms of an integral, if the value is known from experiment, but (too) hard to obtain by integration. Already very simple diatomic cases required impractically long computation times with extant mechanical devices – if one chose to refrain from the semi-empirical strategy.

Ransil's task was to design and implement the "first computer program to generate diatomic wavefunctions" (Bolcer and Hermann 1994, 8) without recourse to empirical values. This program was written in machine language for a UNIVAC

[3]This thesis is put into the context of a "culture of prediction" by Johnson and Lenhard (2011).

(Remington-Rand 1103) computer that was not in possession of the university, but of the military and was located at Wright Field Air Force Base in Dayton, Ohio. Mulliken had contracted excess computing time from the military. That meant a particularly inconvenient working arrangement: Ransil had to prepare the set of commands of his program, then travel from Chicago to Ohio with a stack of prepared punch cards, and work over night or the weekend with the UNIVAC (for more color to the story, see Mulliken 1989). The modification of the program was extremely tedious by today's standards due to the working conditions and also due to the fact that machine language programs will regularly require entirely new programming when the model is modified.

Nevertheless, at the end of the day, or many days, or rather many nights, the program ran. That gave reason for Mulliken and Roothaan to announce "Broken Bottlenecks and the Future of Molecular Quantum Mechanics", so the title of their (1959) paper. The broken bottleneck to which they refer was the complete automation of computing which made computational strategies feasible that avoided to bring in empirically determined values. They report about Ransil's machine program: "The importance of such a machine program is illustrated by the fact that the entire set of calculations on the N_2 molecule which took Scherr (with the help of two assistants) about a year, can now be repeated in 35 min ..." (Mulliken and Roothaan 1959, 396).

The speed of computation definitely mattered and speed was basically a matter of technology. At the same time, it was a demanding process to arrange all conditions so that one could profit from the speed. On the one hand, computer technology unleashed iterative modeling strategies. On the other hand, it erected also limiting conditions due to cost, availability, programming language, and (missing) feedback interfaces. In effect, exploratory work was hardly possible, because it would have demanded an infrastructure where going back-and-forth between testing the performance and modifying the model would be a practical option. This was clearly not the case at Ransil's time and, more general, exploration was hard during the time of relatively expensive and centrally maintained mainframe computers. They canalized modeling toward bracketed rather than exploratory iteration.

3.3 From Bracketed to Exploratory Iteration

Quantum chemistry is arguably only one case among many, picked out here because it covers both pre-computer and computer era.[4] Let us now look at a different case, not a particular scientific field, but a technique that is used in a wide array of sciences, namely the Monte Carlo (MC) method. It is discussed, for instance, by Peter Galison (1996) who wants to make the case for a "Tertium Quid", locating MC

[4]Lenhard (2014) gives a more detailed account of the development of computational quantum chemistry.

simulation alongside experiment and theory. The issue of simulation is discussed in Chap. 7 of this volume; in the following we focus on the iterative nature of MC methods.

Consider a simplistic example of how MC works: Calculating the area of a figure is often difficult because integration depends on the algebraic form of the 'boundary'. MC integration provides an elegant way out: Put the figure in a rectangular frame of known size and create – with the help of a (pseudo-)random number generator – points with random positions inside the frame. Some points will fall into the figure, some will lie inside the frame but outside the figure. If one iterates this procedure very often, i.e. if one creates very many random points, the mathematical law of large numbers will be at work and guarantee that the 'empirical' fraction of points in the figure relative to all points will converge to the relative size of the figure in the frame. Thus, the basic recipe of MC is to replace *analytic integration* by a *probabilistic* approach with massive *iteration* – a method only feasible based on computer technology.

Monte Carlo methods rest on a fundamental conceptual shift while on the algorithmic level, it works with brute force iteration. A well-known downside is the rather slow convergence rate, i.e. one needs often impractically long runs to reach the desired precision. Such methods have received great sophistication in the form of so-called Markov chain Monte Carlo (MCMC) methods. These methods have seen an enormous uptake and have become the standard approach to tackle complicated integrations in many branches of the sciences.

They combine MC with the theory of Markov chains, so let me first add a few words about Markov chains. Such chains are random processes that can be described as movements through a space according to probabilistic rules. Intuitively, think of a tourist's random walk through the streets of a town, choosing on each crossing between all streets with specified probabilities. After a couple of decisions, an observer can still guess where roughly the tourist started, but after many steps this will be less so. The basic mathematical theorem[5] posits that such a chain will converge to its unique equilibrium distribution, no matter where it started. A most astonishing fact to experts in the field is that convergence to equilibrium regularly happens very quickly! In typical cases, the speed of convergence is a fact observed from computer runs, not a general fact derived from Markov chain theory.

Markov chain Monte Carlo methods make use of this fact to simulate the equilibrium distribution. Again, the algorithm is easy to implement. Let the Markov chain move for a while, report where it is, then start over anew. If one repeats that often, the cumulated reported results will present a Monte Carlo picture of the equilibrium distribution. The point is that the convergence rate of this iterative procedure is much higher than for regular Monte Carlo. Thus it is easy to simulate the equilibrium distribution for a given Markov chain. The trick of MCMC is to start

[5]Matters are greatly simplified in our discussion. Only the discrete case is considered and questions of how the space is defined or which technical conditions have to be satisfied are ignored as they are not important for the illustrative task.

with a complicated object, like a high-dimensional integral, and then to construct a Markov chain that has this object as equilibrium distribution. This construction is often not difficult – mainly because one has the license to define the Markov process in a suitable space, like the space of configurations of the Ising model (discussed in a moment). The basic conceptual point of MCMC is the same as with the simpler Monte Carlo, i.e.

(i) complicated or analytically intractable integrals are transformed into the idiom of repeated (random) processes and thereby
(ii) the problem of integration is transformed into one of iteration.

MCMC has roots practically as old as the Monte Carlo method itself; it goes back to Metropolis et al. (1953) and was later systematized in the Metropolis-Hastings-algorithm (Hastings 1970). Although the method was specific enough to be called an 'algorithm', MCMC acquired great popularity only since around 1990. This more recent dynamics is based on the interplay of computer technology and modeling that changed MCMC from a somewhat curious invention into a method of extremely wide use. The key, or so will be argued, is the step from bracketed to exploratory iteration.

Let us illustrate this claim by a standard example, the Ising model of thermodynamics. Consider a two-dimensional grid of cells; each cell has a spin (up or down) and interacts with its four neighbors via a tendency to take up the same spin as the neighbor. This behavior is implemented as a stochastic process. Roughly, in each time step the spin at any location takes on the same value as the spin at neighboring locations, but only with a certain probability. And the probability is higher the more neighbors already show a particular spin. Local interactions are easily described while the resulting global distributions are famously intractable, because the state of the neighbor of a given cell depends again on the states of its neighbors etc.[6] One spectacular proof of success of MCMC was the solution of the Ising model's riddle. MCMC transformed the intractable problem of determining the equilibrium distribution into a question that was solvable with a surprisingly moderate effort.[7] MCMC became a standard method to tackle complicated probability distributions and multi-dimensional integrals that formerly were deemed intractable.

The iterative part is clearly fundamental to MCMC, but what is the role of the exploratory component? This role comes to the fore when one addresses the question whether the Markov chain has actually reached the equilibrium. It is known from 'experience' that such chains often converge surprisingly fast, but there is no known general mathematical result stating how fast a given chain converges and when it has reached its equilibrium. Diaconis (2008) acknowledges the tremendous

[6]The problem is similar to the one of electron interaction in the Schrödinger equation.

[7]Persi Diaconis, a leading probability theorist, vividly describes how astonished he was when he first saw the MCMC solving this task (2008). R.I.G. Hughes (1999) gives a highly readable account of the Ising model in the context of simulation.

impact of MCMC on recent sciences and he aptly points out that the convergence question is an important and urgent problem.

Though there are no strict solutions to this problem, there exist a bunch of strategies to explore and inspect the behavior. The software AWTY (Nylander et al. 2008) that has been designed for the special case of MCMC methods in Bayesian phylogenetics provides an illustration. The acronym expands in "Are we there yet?", i.e.: has the chain under consideration already reached equilibrium? The program provides visualizations of the chain's behavior that shall enable the researchers to explore the chain's behavior and to judge whether the dynamics looks like one near equilibrium. In effect, exploration with the help of visualization substitutes a theoretical convergence result. Regarding technology, this substitution process, and exploratory strategies in general, does not demand great computing power, rather it makes use of small and easily available computers that make such exploratory procedures a practical option. This observation indicates that the upswing of MCMC and the availability of (relatively small) computers at the same time are more than a mere coincidence.

4 Epistemic Ramifications

A good way to characterize the epistemic ramifications of computing technology is again by contrast. Mathematical modeling operationalizes all sorts of relationships and makes them amenable to systematic manipulation. Notwithstanding the fact that mathematical calculations might become complicated and demanding, mathematical modeling was oriented at, or rather bound to, a transparent outline. Arguably mathematics even served as a paradigm of epistemic transparency. This goal resonates with the metaphysical vision of simple fundamental laws that are expressed in mathematical form. It might even be a main factor in producing this vision.

Of course, mundane questions of tractability influenced mathematical model building. Consider systems of partial differential equations (PDEs) that come up in many instances when mathematical models express global relationships with the rationale of the differential and integral calculus. We have discussed already the Schrödinger equation; another famous example are the Navier-Stokes-equations in hydrodynamics. Such systems of PDEs remain largely intractable and hence the engineering sciences preferred approaches that circumvent such systems. With computer technologies, however, this attitude has reversed. Systems of PDEs now look very attractive because they can be treated by standard software packages that convert the analytically intractable into iterative schemes tractable by the computer.

However, the shift to computer technology has important epistemic ramifications that basically run counter to the epistemic transparency connected to mathematical modeling. Iterative procedures are fruitful exactly because iterations do not simply arrive at a point foreseeable in the beginning. Computer models, at least the more complex ones, perform a myriad of simple steps – every single one perfectly

conceivable and 'logical'. Over run time, however, the initial assumptions interact in an intricate way and normally this cumulative effect creates epistemic opacity in the sense that manifest features of the model's behavior often cannot be traced back to particular assumptions or input values. Hence, mathematical modeling – in the form of computational modeling – is moving away from its own former virtue of epistemic transparency![8]

We have seen in the case of the Ising model and the MCMC method that the solution is computed although a transparent understanding is not attained. We are at a loss for two reasons. First, the definition tells us everything about the behavior of the Markov chain in the next time step, but tell us little about the long-term behavior, so that we (or the computer) have to actually conduct the iterations. Second, the definition itself follows an instrumental reasoning: The chain is not derived from its significance in terms of the dynamics in question (Ising model), apart from the sole aspect of having the desired equilibrium distribution. Hence there are often no intuitive clues of how such chains evolve over time.

Let us contemplate a bit more on the notion of understanding and its relationship to transparency and opacity. True, the ideal of epistemic transparency seems to be hampered when computer technology is at work. But how strict a prerequisite is this ideal for understanding? The latter notion is a quite flexible one – what counts as intelligible has undergone changes in history as highlighted by Peter Dear (2006), among others. He discerns two branches of science – natural philosophy oriented at explanation and understanding on the one hand and, on the other hand, science as instrumentality that is oriented at intervention.

"Alongside science as natural philosophy, therefore, we have science as an operational, or instrumental, set of techniques used to do things: in short, science as a form of engineering, whether that engineering be mechanical, genetic, computational, or any other sort of practical intervention in the world." (Dear 2006, 2)

Dear diagnoses that the parts are normally taken to relate to each other in a circular way:

"Why are science's instrumental techniques effective? The usual answer is: by virtue of science's (true) natural philosophy. How is science's natural philosophy shown to be true, or at least likely? The answer: by virtue of science's (effective) instrumental capabilities. Such is the belief, amounting to an ideology, by which science is understood in modern culture. It is circular, but invisibly so." (2006, 6)

Hence, science belongs to two ideal types simultaneously. Modern science emerged as natural philosophy amalgamated with instrumentality. Does computer technology change the composition of this amalgam? Does it affect the very notion of understanding?

[8]Here, my argument takes up Paul Humphreys' account (2004, 2009) of simulation who points out that epistemic opacity is an important (if deplorable) aspect of simulation. I would like to maintain that opacity applies to the use of computer technology more generally. As this topic is also discussed in Chap. 7, I can be brief here and just point out the general significance in connection with computer technology.

Dear stresses that there are no timeless, ahistorical criteria for determining what will count as satisfactory to the understanding. Assertions of intelligibility can be understood only in the particular cultural settings that produce them. Intelligibility, for him, is ultimately an irreducible category (2006, 14). This is the clue to my claim about the broader epistemological significance of computing technology: basic epistemic notions are related to technology – in the sense that they will change in correspondence to changes in technology. In our case, computational modeling affects what is desired and accepted as understanding.

To a certain extent, it is the mathematical form of representations that enables scientists and engineers to draw conclusions and to make predictions. Traditionally, in the circular picture Dear shows us, understanding, explanation, and the capacity for intervention are correlated in a positive way, i.e. the more understanding and explanation, the more ability to predict and to manipulate – and vice versa. With computer technology and computational modeling, however, the ability to predict (and therefore to systematically intervene) are *negatively* correlated to epistemic transparency.

At least, the use of computing technology together with computational models in some cases does provide accurate results. And this might be seen as a sufficient criterion for understanding: If you want to prove that you understand how a problem can be solved – show the solution. From this pragmatic perspective, it is a minor aspect how 'transparent' your machinery is. Thus computer technology seems to fit to a pragmatic, intervention-oriented notion of understanding. The somewhat paradoxical diagnosis then is that this pragmatic understanding is rooted more in technology than in intellectual transparency.[9] If we accept this pragmatic notion then explanation would be decoupled from understanding, because explanation seems to demand more and is less open to a pragmatic account. At least this holds for most philosophical accounts of scientific explanation. I see it as an open question whether the pragmatic sense of understanding will prove to be a preliminary and deficient mode – or whether it will be accepted as the thing that computational science can achieve and that eventually will be adopted as a (non-deficient) notion of intelligibility.

Additionally, the epistemic impact of computer technology on science is fostered by methods that are data-driven and do largely ignore any theoretical structure, but are nevertheless effective in yielding predictions. In their paper on "Agnostic Science", Napoletani, Panza, and Struppa (2011) describe the "microarray paradigm", a methodological paradigm of data analysis. They "argue that the modus operandi of data analysis is implicitly based on the belief that if we have collected enough and sufficiently diverse data, we will be able to answer any relevant question concerning the phenomenon itself." (2011, 1)

A microarray, also known as DNA array, is a chip with thousands of manufactured short strands of DNA on it. If a sample is tested (washed over it), constituents of the sample bind to the chunks on the chip, depending on both the composition of the probe and the spatial distribution of the strands in the array.

[9]Cf. Lenhard 2009 for a more full-fledged argument about understanding and simulation.

Thus, the resulting pattern – created with the help of artificial coloring of the probe – somehow mirrors what the probe consists of. However, the exact shape of the patterns depends on complicated conditions that are hard to control, like the exact location and constituents of the single DNA strands. In sum, the resulting patterns contain a wealth of information, but at the same time a high degree of noise. Because of the sheer bulk of data, even a high level of noise leaves intact the chances to detect the signal, i.e. extract relevant information about the probe. Efron (2005) highlights a similar perspective when he sees the twenty-first century marked by data giganticism, and takes microarrays as paradigmatic. In this regard, DNA arrays are a typical example of a situation where new technological high-throughput devices deliver great amounts of data. These data, in turn, require the use of computer technology for analysis.

The point is that computer based data analysis might be able to detect signals and to make predictions, like: the patterns of the probe resemble patterns produced by tissue with a certain disease. In this way, data analysis can take advantage of the amount of data without specifying a model of the dynamics that produces the data. This is what Napoletani et al. call "agnosticism". This agnosticism refers to a characteristic of the mathematical methods: They work on resemblance of patterns but do not involve theoretical hypotheses or structural assumptions about how the patterns are produced. In this respect, the mathematical techniques of data analysis are indeed "agnostic". Consequently, the success of DNA arrays and other data-driven methods shows how computer technologies on the data side and on the analysis side work together in a way that features prediction and intervention.

I want to avoid a potential misunderstanding: It is not claimed that science, due to computer technology, can get rid of any structural understanding. Rather, a new problem occurs to integrate the combined results of computer technology-and-modeling with more traditional approaches. Consequently, Napoletani et al. take a careful stance in later passages of their text. They suggest that after having achieved predictive success in an "agnostic" manner, later steps, especially the incorporation into the body of scientific knowledge, may need a more structure-based understanding.

5 Infrastructure

In this section I want to very briefly address how computer technology, in particular networked infrastructure and software packages, affects the social organization of science. Of course, there is excellent scholarly work on 'big science' as it was organized in the Manhattan Project and later on. Not accidentally, computer technology is a core element in big science, see for instance Edwards (1996). Climate science provides a somewhat different type of example, also "big", and also making essential use of computer technology (cf. Edwards 2010). These cases will not be the issue here, however. Instead, I want to concentrate on small computers that have become part of everyday science culture, even of everyday culture.

Sociologist of science Sherry Turkle has studied how design and education in architecture and civil engineering have changed in the course of the proliferation of computer technology (Turkle 2009). Her study compares two timelines, around 1980 and around 2000, i.e. before and after the spread of small computers. By 1980 computer technology had been introduced, but many practitioners remained hesitant so that the technology was employed in a way that was crafted on the older tool-building tradition. However, by 2000, the situation had changed, so Turkle. Many users were working with these tools as a matter of course while persons from the tool-building tradition that could repair their instruments or check the code weren't available anymore. That is, developers and users of software had started to build different social groups.

Turkle reports disagreement among architects as well as civil engineers whether students should learn programming (2009, 19) – does one need to understand the instruments one uses? On the one side, the use of software packages, like for computer assisted design, offered advantages in terms of which tasks could be fulfilled by persons without long experience in the field. On the other side, there were concerns about growing dependency on tools that essentially had become opaque.

Moreover, also the possibility of a new sort of understanding comes up in Turkle's study, one that is based on the *explorative* mode of modeling: At the 1980 timeline, design was thought to follow or elaborate on a fundamental plan. This changed, as Turkle reports (2009, 23), because computer programs allowed to play with preliminary designs. The exploratory mode can be recognized also in software tools that are adapted to different environments. To grab code and to customize it had become usual practice by 2000. These somewhat unprincipled procedures are reflected in a relatively low status of any single program. To use several programs and models and to compare their results is widespread practice that is greatly enhanced by infrastructure of networked computers.

Without doubt, these developments have serious drawbacks. There is a trade off between pragmatically motivated exploration and theoretically founded certainty. It is not yet clear, I would like to argue, to which extent scientists can influence, are able to choose, or rather have to accept how weights are assigned in this trade.

6 Conclusion

Let us take stock. We have investigated several aspects of how computer technology and conceptions of computational modeling are interrelated. In the form of computational modeling, mathematical modeling has undergone a fundamental transformation, characterized by the features of iteration, exploration, opacity, and agnosticism. This transformation affects the perceived structure of the scientific endeavor.

We only briefly discussed how computer technology is involved in the production of data. Admittedly, this constitutes a highly relevant matter. Data are – contrary to

their etymological roots – not primarily 'given', but often construed by computer technology and computational models. Think of a CT scan, an image of a scanning tunnel microscope, a visualization of particle collisions at the LHC (Cern), or a map displaying rain clouds based on satellite data. All of them show highly processed data. Normally such data can only be produced and handled with the aid of computer technology. Data-driven methods have become a slogan in many branches of science. We have touched upon this issue during the discussion of agnosticism and the "DNA-array paradigm". The interplay of data production and analysis, made possible and mediated by computer technology, fosters our claim that technology and modeling are interwoven.

Finally, I would like to point out two issues that pose open questions. First, remind the metaphor of computer technology as the Spinning Jenny of computation. I take it as an open question whether science is inherently industrial or artisan, i.e. to which extent computer technology will change the fundamentals of scientific knowledge production. Does the exploratory mode yield only successes of a transient nature? Does the use of computer technology require to re-assess the notion of understanding in the way indicated in the preceding investigation? These questions have profound implications for how we understand science and our culture. Any answer will have to grant computer technology a central place in the analysis.

The second issue is a methodological one. If a study wants to elaborate on the previous argumentation, or to disprove it in some controversial points, it will profit when it combines philosophical, historical, and sociological aspects. I am convinced that attempts to elucidate the relationship between science and computer technology call for a re-assessment of how we study science and technology. There exist ongoing broader movements into that direction, like the recent 'practice turn' in philosophy of science, or integrated programs like "&HPS".[10] In my opinion, the explicit inclusion of technology and the technological sciences will be among the success conditions for this kind of study.

References

Bolcer, J. D., & Hermann, R. B. (1994). Chapter 1: The development of computational chemistry in the United States. In K. B. Lipkowitz & D. B. Boyd (Eds.), *Reviews in computational chemistry* (Vol. 5). New York: VCH Publishers.

Dear, P. (2006). *The intelligibility of nature. How science makes sense of the world*. Chicago: The University of Chicago Press.

Diaconis, P. (2008). The Markov chain Monte Carlo revolution. *Bulletin of the American Mathematical Society, 46*, 179–205.

Edwards, P. N. (1996). *The closed world. Computers and the politics of discourse in Cold War America*. Cambridge, MA: MIT Press.

[10] Among them also Nordmann (2012) who formulates a program for history, philosophy, and sociology of the technological sciences.

Edwards, P. N. (2010). *A vast machine. Computer models, climate data, and the politics of global warming*. Cambridge, MA: MIT Press.

Efron, B. (2005). Bayesians, frequentists, and scientists. *Journal of the American Statistical Association, 100*(469), 1–5.

Galison, P. (1996). Computer simulations and the trading zone. In P. Galison & D. J. Stump (Eds.), *The disunity of science: Boundaries, contexts, and power* (pp. 118–157). Stanford: Stanford University Press.

Hacking, I. (1983). *Representing and intervening. Introductory topics in the philosophy of natural science*. Cambridge: Cambridge University Press.

Hartree, D. R. (1949). *Calculating instruments and machines* (photographical reprint). Urbana: The University of Illinois Press.

Hartree, D. R. (1984). *Calculating machines. Recent and prospective developments and their impact on mathematical physics*. Cambridge, MA/London: MIT Press (Photographical reprint of Hartree's inaugural lecture 1947.)

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika, 57*(1), 97–109.

Hughes, R. I. G. (1999). The Ising model, computer simulation, and universal physics. In M. Morgan & M. Morrison (Eds.), *Models as mediators* (pp. 97–145). Cambridge: Cambridge University Press.

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

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

Johnson, A., & Lenhard, J. (2011). Towards a culture of prediction: Computational modeling in the era of desktop computing. In A. Nordmann, H. Radder, & G. Schiemann (Eds.), *Science transformed? Debating claims of an epochal break* (pp. 189–199). Pittsburgh: University of Pittsburgh Press.

Lenhard, J. (2009). The Great Deluge: Simulation modeling and scientific understanding. In H. de Regt, S. Leonelli, & K. Eigner (Eds.), *Scientific understanding. Philosophical perspectives* (pp. 169–186). Pittsburgh: University of Pittsburgh Press.

Lenhard, J. (2014). Disciplines, models, and computers. The path to computational quantum chemistry. *Studies in History and Philosophy of Science Part A, 48*, 89–96.

MacKenzie, D. (2006). *An engine, not a camera. How financial models shape markets*. Cambridge, MA/London: MIT Press.

Meijers, A. (2009). *Philosophy of technology and engineering sciences* (Handbook of the philosophy of science, Vol. 9). Amsterdam: Elsevier.

Metropolis, N., Rosenbluth, M. N., Rosenbluth, A. H., & Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics, 21*(6), 1087–1092.

Morrison, M. (1999). Models as autonomous agents. In M. Morgan & M. Morrison (Eds.), *Models as mediators* (pp. 38–65). Cambridge: Cambridge University Press.

Mulliken, R. S. (1989). *Life of a scientist*. New York: Springer.

Mulliken, R. S., & Roothaan, C. C. J. (1959). Broken bottlenecks and the future of molecular quantum mechanics. *Proceedings National Academy of Sciences, 45*, 394–398.

Napoletani, D., Panza, M., & Struppa, D. (2011). Agnostic science. Towards a philosophy of data analysis. *Foundations of Science, 16*(1), 1–20.

Nordmann, A. (2012). Im Blickwinkel der Technik. Neue Verhältnisse von Wissenschaftstheorie und Wissenschaftsgeschichte. *Berichte Wissenschaftsgeschichte, 35*, 200–216.

Nylander, J. A., Wilgenbusch, J. C., Warren, D. L., & Swofford, D. L. (2008). AWTY (are we there yet?): A system for graphical exploration of MCMC convergence in Bayesian phylogenetics. *Bioinformatics, 24*(4), 581–583.

Park, B. S. (2009). Between accuracy and manageability: Computational imperatives in quantum chemistry. *Historical Studies in the Natural Sciences, 39*(1), 32–62.

Turkle, S. (2009). *Simulation and its discontents*. Cambridge, MA/London: MIT Press.