Chapter 1
Introduction

What are models and how do we obtain and assess them? How do abstract models turn into tangible simulation results? What are the ever increasing number of "simulators" doing exactly, what constraints apply to their activities and how can their results be validated? These and other questions are discussed in the first chapter of our book. It is designed to be a general introduction as well as a separate introduction to each of the four subsequent parts. The first section of this chapter provides the general terms and definitions that apply to simulation and introduces the so-called simulation pipeline. In sections two and three we provide the basic foundations of modeling and simulation, respectively.

1.1 The Simulation Pipeline

The notion of simulation is quite ambiguous and requires clarification. In the context of this book, two of its interpretations are of particular relevance. In a broader sense, simulation is the complete process of the forecasting or replication of a certain scenario. Since such simulations are nowadays performed almost exclusively computer-based, we will not—as it is oftentimes seen elsewhere—refer to it as computer simulation. In a tighter sense (and in the title of this book), simulation only refers to the central part of this process, i.e., the actual computation—a classical case of a "pars pro toto". In the following, we will make use of both interpretations and only provide explicit clarification if the respective interpretation is not given implicitly.

In a broader scope, therefore, simulations are nothing other than "virtual experiments" on the computer. This remains unchanged by the fact that in most application areas served by simulation (for example, physics, chemistry or mechanics), the respective representatives of the "computational guild" are typically allocated to the theoreticians.

H.-J. Bungartz et al., *Modeling and Simulation*, Springer Undergraduate Texts in Mathematics and Technology, DOI 10.1007/978-3-642-39524-6_1,

The attractiveness of these virtual experiments is obvious. For a multitude of cases, "real" experiments are simply impossible due to the underlying time and spatial scales, for example. To illustrate, one needs only to consider astrophysics: no matter how hard-working, it is impossible for any physicist to devote the necessary billions of years at the telescope to study the life cycle of a galaxy; or geophysics—it may be possible to create experimental earthquakes, i.e., artificial earthquakes in a James Bond production, but those are not practical in real life. Moreover, not all that is possible in principle is actually desirable—one only needs to consider the testing of nuclear weapons, animal experiments, or genetic engineering. The former took their leave just at the time when the respective nations reached the ability to execute them in a completely virtual manner on the computer. The ethical component—nuclear bombs do not become friendlier if they are "brought to perfection" through simulations—must not be left out here, but as well will not be discussed further. And even in the remaining set of the feasible and justifiable, the effort is often the limiting factor: The static of buildings, the vulnerability of the HIV virus, the evacuation of a fully filled soccer stadium, economical or military strategies, etc. etc.–all these are not tested quickly, not even in the lab; not to mention the effort that fundamental experiments require in modern physics in the context of the Large Hadron Collider. Thus, there is no way to go without simulation and it is therefore worthwhile to take a closer look at its methodology. However, it is indisputable: Simulations *complement* theoretical analyses and experiments, they do not *replace* them.

The goals pursued by a simulation can be very diverse. Oftentimes, one wants to reconstruct a scenario which is well-known in principle in order to better understand it. This applies for example to catastrophies of a technical or natural kind. Why has an earthquake developed, why at this particular place, why at this particular instant in time? Why did one of the large traffic bridges across the Mississippi River in the US state of Minnesota collapse in August 2007? How could the tsunami in south-east Asia in late December 2004 develop such a devastating effect? The goal to predict unknown scenarios is also knowledge driven, but in general even more challenging. This applies not only to the catastrophies mentioned above (and for possible repetitions, resp.) as well as to urgent questions concerning climate change or the propagation of the world population, but also to many technical questions (properties of new alloys or composite materials). Besides discovery, another goal pertains to improvement, i.e., the optimization of a known scenario. Prominent examples include the (route) scheduling of airlines, the efficiency factor of chemical reactors and the efficiency of heat exchangers or the data throughput in a computer network.

Here, a simulation in the broader sense is not an intergral act, but rather a highly complex process consisting of a sequence of several steps which are traversed several times in various feedback loops.

To this end, the picture of a "simulation pipeline" has been established (see Fig. 1.1). We summarize the essential steps:

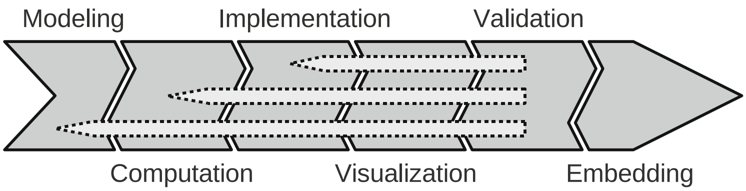

Fig. 1.1 The "simulation pipeline"

- The *Modeling*: At the very beginning we need a model, i.e., a simplified formal description of a suitable extract from the item of interest, which will then serve as the basis for the subsequent computations.
- The *computation* or *simulation in the tighter sense*, resp.: The model will be preprocessed (e.g., discretized) so that it is compatible with a computer platform. The solution of this preprocessed model requires the identification of efficient algorithms.
- The *implementation* (or more generally the *software-development*): The computational algorithms previously determined must be implemented efficiently (with respect to computational time and storage complexities, parallelization issues, etc.) on the target architecture or architectures. Currently, this step significantly exceeds the implementation in the classical sense: It is no longer sufficient to produce runnable code, but software must be designed and developed on a big scale and by every trick in the book.
- The *visualisation* (or more generally the *data exploration*): The data resulting from a simulation run must be interpreted. In some cases—e.g., for scalar quantities such as the drag coefficient in aero dynamics—this will be easy, in others—e.g., for highdimensional data sets—extracting the relevant information from the flood of numbers is a science of its own.
- The *validation*: Very important—how reliable are the results? Sources for errors lurk in the model, in the algorithm, in the code or in the interpretation of the results. Therefore, it is important to compare different models, different algorithms, and different codes, resp., as well as simulation results with inkind experiments. Depending on the source of the error, the process has to be restarted at the respective step and the pipeline has to be traversed once more starting from this point.
- The *embedding*: Simulations take place in a context—e.g., a development or production process—and should be integrated accordingly. This requires the definition of interfaces, a reasonable software engineering, simple testing environments, etc.

Let us take a look at a descriptive example—a little preview of part IV of this book. The subject of our interest is the automobile in a wind tunnel—or, better, the virtual automobile in the virtual wind tunnel. We want to figure out the wind resistance of the vehicle that, in technical terms, is denoted by the drag coefficient c_w. For subsonic aerodynamics, the suitable physical or mathematical model, resp., is given by the Navier–Stokes equations, a system of nonlinear partial differential equations. Their discretization in space and time can, for example, be carried out through finite elements or finite volumes. The resulting large and sparse linear systems of equations can be solved efficiently using multigrid methods. In view of the method's large computational time requirement, the method typically must be implemented on a parallel computer. The visualization of the three-dimensional velocity field will require techniques that exceed the well-known two-dimensional pictures using arrows, and the drag coefficient must be computed in a suitable way from the millions of computed discrete velocity and pressure values. The validation will rely on comparative calculations with other programs as well as experiments using prototypes in a wind tunnel.

The proper way to convey the results of an aerodynamics simulation to the design department (e.g., suggestions pertaining to details involving change in the shape of fenders) is an exciting task of embedding. Or said differently: How can simulation results be directly integrated into the CAD-model without causing the very time intensive design process to be restarted?

It is clearly illustrated through this example: The comprehensive solution of a simulation problem requires far more than "a little bit of computation"—all six steps pose an abundance of challenges to the different fields of science. In order to avoid an immediate misunderstanding: The figure depicting the pipeline should illustrate the diversity and sequential flow of the intermediate tasks, but it should on no account suggest that the individual steps can be worked separately by entirely different experts in a manner much like an assemply line production. In fact, everything is closely interwoven. For example, beginning early on in the design of numerical algorithms, one must keep an eye on an efficient implementation - this typically takes the target hardware into account one way or another.

The first two steps of the simulation pipeline—modeling and simulation in the tighter sense—are certainly of central importance. For this reason, and since they are absolutely necessary for an introduction to the topic, they will be covered in this book.

1.2 Introduction to Modeling

We will direct our attention to the first step of the simulation pipeline, the (mathematical) modeling, by discussing in sequence the following questions: What is actually a model, and what is its purpose? How does one obtain a suitable model? How can mathematical models be assessed? What is the difference between models, and how can they be classified? And finally—does there exist "the correct model"?

1.2.1 General Prerequisites

In general, a model is a (simplified) image of a (partial) reality. In this context, the models are always meant to be abstract, i.e., formal descriptions given mostly (but not always) through the methodological apparatus provided by mathematics or computer science. In the following, when talking about models, we will almost always think of mathematics or computer science models.

Mathematical modeling denotes the process of the formal derivation and analysis of a mathematical model for an effect, a phenomenon, or a technical system. The starting point is in general an informal description of the respective subject of modeling, for example in prose. This is typically converted next into a semiformal description, the model of the science application, using the tools of the application discipline. Finally, an additional step is required to derive a strictly formal model (i. e, unambiguous, consistent)—the mathematical model.

A simple example to illustrate this is the management of a timetable and the room scheduling of a school: To start, we have the textual description of the problem. This leads to the classical arrangement of index cards on the wall of the teachers' room which helps to avoid double bookings of rooms, but is limited in its ability to recognize possibilities for optimization. This situation changes when the problem is "mathematized", for instance being formulated as a graph-based scheduling problem. Once such an abstraction is performed, one can now apply appropriate methods for the problem's analysis and optimization.

The extent that mathematical modeling is suggestive and established differs greatly in various areas of science. It has a very long tradition in the exact natural sciences. Several formulations of theoretical physics, for example, are per se mathematically rooted and in many areas recognized as such. This holds in particular where they have been validated through experimental data (as, e.g., in classical continuum mechanics). A completely different situation prevails for national economic policy. In view of the substantial impact the psychology of the moment contributes, the extent that mathematical models are supportable is questionable. And even if there is consensus, the choice of the "right" model is by no means obvious. Further, depending on the choice of model, one can derive economic-political rules of action that are diametrically opposite. As an example, one may think of the perpetual argument between the "monetarists", who call for strict budgetary discipline even in times of recession, in contrast to the "Keynesians", who hold in high esteem the "deficit spending" of their idol John Maynard Keynes and therefore demand national investment programs in times of recession. Needless to say, both camps refer to models!

But even among scientists, consensus does not necessarily exist or, at best, it is an arduous process. This is exemplified by the on-going discussion about climate change and global warming.

Game theory, which we will cover in more detail later, is another example that illustrates the difficulty in finding the right model. John von Neumann's models for 2-person zero-sum games which embrace conservative min-max strategies—in short: always play so that the worst case loss is minimized assuming optimal performance by the opponent—may be an appropriate model to use for the head of a family on his one-time meander to the casino. Without doubt, however, these strategies do not suffice for the serious gambler.

Where do we model nowadays? A few important applications have already been mentioned, but no list can altogether be complete. However, one should be aware how commonplace the use of models, and thus modelling, have nowadays become:

- In *astrophysics*, one works to explain the origin and evolution of the universe as well as the life cycles of stars and galaxies.
- In *geophysics*, researchers want to understand processes which eventually lead to earthquakes.
- A central topic in *plasma physics* is fusion.
- The analysis of the spatial structure and therefore the funtionality of proteins is a focus in *protein research*.
- *Theoretical chemistry* investigates the causes for certain material behavior on the atomic level.
- *Drug design* is concerned with the systematic design of agents having an exactly specified functionality.
- Also *medicine* increasingly utilizes models, e.g., in the research of aneurysms or the optimization of implants.
- The discussions in *climate research*, which are driven by models, have had a high audience appeal—ranging from global warming, holes in the ozone layer or the future of the gulf stream.
- On a shorter term, but no less current, is *weather forecasting*, which relies on a mix of computations and measurements.
- The *automobile industry* yields a whole wealth of examples: Whether one considers crash tests (structural mechanics), deep drawing (structure optimization), aerodynamics or air conditioning (fluid dynamics), sound emission (aeroacoustics), fuel injection (combustion), vehicle dynamics (optimal control) or sensor and actuator technologies (coupled systems, micro-electro-mechanical systems)—models are always involved.
- Also the *semiconductor industry* depends completely on models and simulations—examples are given by device simulation (transistors etc.), process simulation (production of highly purified crystals), circuit simulation as well as questions regarding the optimization of chip layout.
- Furthermore, several models have been and are being developed in *economics*—for the business cycle, for the economic and fiscal policy and for pricing mechanisms. The fact that the experts in the government's advisory boards are not necessarily always of one opinion, illustrates how far one is still from a consensus model.
- For *banks and insurance companies*, models are highly important for the assessment (i.e., pricing) of financial derivates such as options.

- *Traffic technology* requires quite diverse models—for example for the formation, dissolution and avoidance of traffic congestion, for the long term planning of traffic routes and for evacuation scenarios.
- *Providers* such as energy companies require load models in order to design their networks as fail-safe as possible.
- *Shipping companies* rely on model-based fleet management.
- Population models are used by *city planners*, *governments* (thinking of China's "one-child-policy") as well as by *epidemiologists* (how do famines spread?).
- Without models, the statements of *pollsters* would ressemble the prophecies of fortune tellers.
- And finally, what would *computer games* and *computer movies* be without illumination or animation models?

One sees: There are quite diverse application areas for modeling that exist with both "hard" (i.e., mathematical-formula heavy) and "soft" (i.e., more descriptive) models. And by now one suspects that these require a corresponding wide range of tools from mathematics and computer science. As a result, the following central questions arise in the context of modeling:

1. How does one obtain a suitable model?
2. Which descriptive tools will be used?
3. How does one subsequently assess the quality of the derived models?

We will now direct our attention to these questions.

1.2.2 Derivation of Models

We will begin with the task of deriving a model—which often appears to be a miracle to the newcomer. This derivation typically occurs in several steps.

At first, one has to determine what exactly should be modeled and thereafter simulated. Whoever is inclined to answer "well, the weather", is not thinking on a grand enough scale: the weather in what time frame, the weather in what region and in what spatial resolution, resp., the weather in what precision? Several additional examples may serve as illustrations: Is one interested in a rough estimate of the efficiency of a car catalyst or rather in the detailed interior reaction processes, i.e., does the domain have to be resolved or not? Is one interested in the population growth in Kairo, in Egypt, or in all of Africa? Which resolution is therefore suitable or adequate? Should the throughput of a computer network or the average throughput time of a packet of data be determined by simulation, i.e., should data sets be considered separately as discrete entities, or is it sufficient to take averaged data flow quantities?

Next, one needs to determine which quantities play a *qualitative* role and how significant is their *quantitative* impact. We will illustrate this with a few examples: The optimal trajectory of the space shuttle is influenced by the gavitation of the moon, the gravitation of Pluto and the gravitation of this book, but not all of these are relevant for the computation of the trajectory. The development of the Dow Jones index may depend on statements of the director of the American central bank, statements of the authors of this book as well as the bankruptcy of the Sultanate of Brunei. However, investors do not have to consider these statements on an equal footing. One sees that we can sometimes grasp causality and relevance intuitively, but in general (in particular for highly complex systems) these are all but obvious. Oftentimes, there are—despite the relevant expertise and ample material data—only hypotheses. And typically these early decisions significantly influence the later simulation results.

Once the set of relevant quantities has been determined, we have to direct our attention to the network of relationships among the model parameters that have been deemed to be important. Once again there is the qualitative aspect (logical dependencies of the type "if, then" or signs of derivatives) and the quantitative aspect (particular factors, magnitudes of derivatives). Typically, these networks of relationships are complex and multi-layered: In general, the CPU performance of a computer has a strong influence on the computational time of a job; however, in the case of strong thrashing or a low cache hit rate, it only plays a minor role. Such fluctuating dependencies have to be incorporated into the model as well.

Now, what is a suitable instrument to best formalize the interactions and dependencies previously identified? Mathematics and computer science provide a wealth of descriptive tools and instruments:

- *algebraic equalities* or *inequalities* to describe laws ($E = mc^2$) or constraints ($w^T x$);
- systems of *ordinary differential equations* (differential equations with only a single independent variable, typically the time t), for example for the description of growth behavior ($\dot{y}(t) = y(t)$);
- systems of *partial differential equations* (i.e., differential equations with several independent variables, such as different spatial directions or space and time), for example for the description of deformation of a clamped diaphragm under load ($\Delta u = f$) or for the description of wave propagation ($u_t = u_{xx}$);
- *automata* and *state transition diagrams*, for example for the modeling of queues (filling levels as states, arrival and service end as transitions, resp.), of text recognition (previous text structures as the states and the new symbols as the transitions) or of growth processes with cellular automata (overall occupancy situation as the states with rule-based transitions);
- *graphs*, for example for the modeling of round trip problems (problem of the traveling salesman with places as vertices and paths as edges), of ordering problems (partial jobs as the vertices, dependencies in time through the edges), of computing systems (components as the vertices, connecting paths as the edges) or of processes (data flows, workflows);
- *probability distributions*, to describe arrival processes in a queue, error terms as well as white noise or the approval of the government's policy on the unemployment rate;

- *rule-based systems* or *fuzzy logic* for the modeling of control problems;
- *neural networks* for modeling learning;
- *languange concepts* such as UML, in order to model complex software systems;
- *algebraic structures*, for example groups in quantum mechanics or finite fields in cryptology.

In this book we will on occasion see that finding a "best description" is quite a lofty goal, and that most cases do not require *the* model but rather *a suitable* model.

Finally, a perceptively trivial question whose answer is nevertheless important for goal-oriented modeling and simulation:: What is the concrete task? Shall we find *an arbitrary* solution of the model; shall we find *the only* solution of the model; shall we find *a particular* solution (which is optimal with respect to a certain criterion or which satisfies certain boundary conditions or constraints, respectively); shall we find a critical region (e.g., a bottleneck); or shall we show that at least one or even several solutions exist?

Once a model has been identified, it needs to be assessed. This will be the topic of the following section.

1.2.3 Analysis of Models

The analysis and assessment of models deals with the derivation of statements in terms of their manageability and usefulness, resp.

Here, the question of *solvability* takes center stage: Does a certain model have one or several solutions, or none?

In population dynamics, for example, one is interested in whether or not a certain model has a stationary limit state and whether or not it actually attains its limit value. In ordering problems, i.e., when, for example, a set of tasks has to be completed on a set of machines, the question arises whether or not the precedence graph, which describes possible constraints in the ordering of tasks, contains any cycles. In minimization problems, it is crucial to determine whether the target function actually assumes a minimum, or possibly only contains saddle points (minimal with respect to a subset of directions while maximal with respect to the remaining directions).

In the case of solvability, the subsequent question concerns the *uniqueness* of solutions: Is there exactly one solution, is there exactly one global minimum? Is there a stable limit state or rather oscillations, as we will later see in predator–prey-scenarios in population dynamics, or different pseudo-stable states, among which the solution jumps back and forth (such as in the form of spatial configurations or convolutions in proteins)? In the case of several solutions, are these all of equal weighting, or are there preferred solutions?

A third aspect is likely to be less obvious: Does the solution depend *continuously on the input data* (initial values, boundary values, material parameters, constraints, etc.), or could rather small changes in the input lead to a completely different behavior in the solution? The idea of continuous dependency corresponds to a notion of the *sensitivity* or *conditioning* of a problem, resp.

In 1923, Hadamard chose the following three aspects as a launching point for his definition of a *well-posed problem*: A problem is therefore called well-posed if a solution exists, is unique, and furthermore depends continuously on all input data. However, Tikhonov and John have subsequently shown that this definition is quite restrictive—unfortunately, problems are mostly ill-posed. Prime examples for ill-posed problems which illustrate this notion are the so-called *inverse problems*. With an inverse problem, the result is essentially given and one is looking for the initial configuration: How should a pressing tool be configured so that a metal sheet is worked to produce the desired result? How much does carbon dioxide emission have to be reduced in the next ten years in order to avoid certain undesirable developments? In politics, what must be done today in order to reduce the percentage of unemployed persons below 10 % within three years? How do the components of a computer network have to be configured in order to guarantee a certain minimum throughput? Even if the corresponding forward problem is continuous (a marginal change in the corporate tax rate will hardly lead to a big jump in unemployment), the continuity typically no longer holds in the opposite direction: It cannot be expected that a slightly smaller unemployment rate can simply be obtained through a slightly reduced tax rate—even if one would like to believe this!

As one may already anticipate, such inverse problems are no rarity in practice—oftentimes a goal to be reached is given, and one is looking for a suitable way to do so. Even if this is an ill-posed problem, there are possible ways to "rescue" the model. A first option is the (meaningful) trial and adjustment, i.e., the solution of a sequence of forward problems. Here, the skill consists of reaching convergence quickly. An alternative approach is the so-called *regularization*. Here, one solves a related, well-posed (regularized) problem instead of the original problem. A generally helpful trick: If the problem is unpleasant, change it slightly!

We still have to discuss a fourth aspect of model assessment—one that is in fact often neglected by the pure modelers or at best treated as a distant relative: How difficult will it be to continue the processing of the model (i.e., the simulation)? In fact, the modeling is not done as an end in itself but rather as a means to perform simulations. This raises a couple of additional questions: Is the availability and quality of the required input data sufficient? In the end, what is the purpose of an ever so elegant model if I do not have access to the input data? For the solution of the model, do algorithms exist, and if so, what are their computational and storage complexities? Given this, is a solution realistic, in particular when keeping in mind the real-time requirements? After all, tomorrow's weather forecast must be completed before tomorrow. Do we have to anticipate or expect fundamental problems during the solution process (ill-conditioning, chaotic behavior)? Is the model competitive, or do there exist models with possibly a better price-performance ratio? And finally, how involved is the expected implementation effort? These all are questions that by far exceed the pure modeling effort but must nonetheless be considered at this stage.

If all these questions could be addressed satisfactorily, one could then approach the simulation stage. Before doing this, however, we will attempt to bring some structure in the midst of the flood of existing and conceivable models.

1.2.4 Classification of Models

From the multitude of possibilities for classification we will take a closer look at two: discrete vs. continuous models as well as deterministic vs. stochastic models.

In modeling, *discrete models* exploit discrete or combinatorial descriptions (binary or integer quantities, state transitions in graphs or automata), while in contrast *continuous models* are based on real-valued or continuous descriptions (real numbers, physical quantities, algebraic equations, differential equations). Obviously, discrete models are naturally used to model discrete phenomena, whereas continuous models are employed for continuous phenomena. However, this is by no means mandatory as demonstrated by the example of traffic simulation which will also be studied later in this book. Here, the traffic flow through a city can be modeled discretely (single cars as entities which wait at lights, etc.) as well as continuously (densities, flows through channels). The approach deemed more suitable depends on the actual problem setting.

Examples of *deterministic models* include systems of classical differential equations which manage without random components. Ever more frequently, however, the systems contain probabilistic components—whether to integrate error terms (noise), or to account for uncertainties, or to explicitly build in randomness (stochastic processes). Once again, there is no mandatory correlation between the character of the process being modeled and the instrument employed. Non-deterministic experiments such as the roll of a die represent a probabilistic reality and are modeled as such; Crashtests are strictly causal-deterministic and are generally modeled deterministically. The weather forecast becomes more interesting: In a sense, everything happens strictly deterministically, obeying the laws of thermodynamics and fluid dynamics. However, several turbulence models contain stochastic components. Finally, the (hopefully) deterministic incoming order of jobs for a printer is mostly modeled via stochastic processes—from the point of view of the printer, the jobs arrive randomly, and furthermore, at least for system design, one is rather interested in average quantities (means) and not in the individual fates of printing jobs.

1.2.5 Scales

One idea should be quickly dismissed, namely the one involving the "correct" model. Modeling is rather a question of consideration of complexity, or cost and accuracy. The more details and single effects one integrates in a model, the higher the precision one naturally expects for the attainable results—however, at the expense of increasing simulation cost. Phenomena always take place on certain *scales*—spatial (from nanometer up to the light year) and temporal (from the femtosecond up to billions of years), and models and simulations themselves are always based on certain scales. In principle, each molecule in the air makes a contribution to the weather—at the same time it would be crazy to consider all molecules individually when forecasting the weather. But one cannot neglect spatial resolution completely: A statement of the kind "tomorrow will be nice in Europe" is usually not very helpful. Thus the question pertaining to the level of detail (spatial or temporal) or resolution arises, i.e., which scales are appropriate—first in view of the desired accuracy of the result and second in view of the required solution cost.

A few examples may illustrate this. Let us begin with true high-technology. The heating of water in a cylindrically formed pot on a stove can be modeled and simulated in one spatial dimension (temperature as a function of time and height in the pot—after all, the pot is cylindrical and its content—water—is homogenous), in two spatial dimensions (temperature as a function of time, the height in the pot and the radial distance to the middle of the pot—after all, the room air is cooling the pot from the outside) or even in three spatial dimensions (additional dependence of the temperature on the circular angle—after all, no stove heats in perfect rotational symmetry); what is the appropriate approach? Or in population dynamics: Typically, the development of a species is described as a purely time-dependent process. However, this could not yield a reasonable description of the development of a population such as in the USA in the middle of the nineteenth century when the strong "go west!" drive prevailed for the migration of settlers.

The simulation of circuits provides another example. For many years, this simulation was performed as purely time-dependent (system simulators based upon Kirchhoff's circuit laws). The increase in integration density leads to a growing occurrence of parasitic effects (current through a conductor induces current through a nearby other conductor) which are local phenomena and require a spatial model component. Finally the catalytic converter in our cars: Do I really need to resolve in detail the geometry of the catalytic converter for the computation of macroscopic quantities such as the degree of efficiency?

The previous question leads us to another aspect, the interplay of scales. Often, we have to deal with a so-called "multiscale property". In this case, the scales cannot be separated without an unacceptable loss of accuracy because of shared interdependency. A classical example is given by turbulent flows. Phenomenological to turbulent flows, one has to deal with strong, erratic vortices of varying magnitudes—from tiny to very large. The flow is unsteady and inherently three-dimensional. Here, a strong energy transport takes place in all directions and between the scales. Depending on the viscosity of the fluid, one needs to compute the tiniest vortices even in larger domains in order to avoid incorrect results. The dilemma, therefore, is that for reasons of efficiency, one cannot resolve all that is needed to be resolved for reasons of accuracy. A remedy is found in turbulence models: They try to pack the fine-scale influence into suitable parameters of the large scale—through averaging (with respect to space or time) or through

Table 1.1 A hierarchy of possible simulations on different scales

Problem setting	Level of consideration	Possible model
Population increase globally	Countries/regions	Population dynamics
Population increase locally	Individuals	Population dynamics
Human physiology	Circuits/organs	System simulator
Blood circulation	Pump/canals/valves	Network simulator
Blood stream in the heart	Blood cells	Continuum mechanics
Cellular transport processes	Macromolecules	Continuum mechanics
Function of macromolecules	Atoms	Molecular dynamics
Atomic processes	Electrons, ...	Quantum mechanics

homogenization. Naturally, such multiscale phenomena set particular requirements to the models and simulations.

In view of the wide spectrum of relevant scales, one often encounters entire model hierarchies. As an illustration, we consider such a hierarchy centered around humans: Each level can represent certain things, but not others, and the models and simulation techniques differ from level to level (Table 1.1).

1.3 Introduction to Simulation

1.3.1 General Remarks

Our aim is not to derive and employ models just for the description of a circumstance, but for the subsequent simulation based on these models. To this end, the models have to be solved in concrete scenarios—for example differential equations plus initial and boundary conditions. This can be done through various methods.

An *analytic solution* not only includes existence and uniqueness proofs, but also the formal analytic construction of the solution—using "paper and pencil", as it is referred to in mathematics. This is insofar the preferred case since no further simplifications or approximations are required. However, this approach works almost exclusively only in very simple (and thus mostly hopelessly unrealistic) special cases. For example, one can directly write down the solution $y = c\,\mathrm{e}^t$ of the simplistic growth law $\dot{y}(t) = y(t)$ without any magic. A little less obvious, but still no trick, is the direct solution of the one-dimensional heat equation $u_{xx}(x,t) = u_t(x,t)$; here, a so-called separation approach yields $u(x,t) = \sin(cx)\mathrm{e}^{-c^2 t}$. Finally, in mini-graphs, one can detect a shortest path through a simple exhaustive search. But what alternatives exist when an analytical solution is not feasible? Irrelevant here is whether this is due to fundamental reasons or due to the limited capabilities of the person working the problem.

The *heuristic solution approach* offers a first alternative in which, beginning with plausibility arguments, one uses certain strategies to get closer to the unknown solution. Such heuristics are wide-spread primarily for problems in combinatorical or discrete simulation and optimization (e.g., greedy heuristics which always choose the best local alternative). In the knapsack problem, for example, one packs the item with the respective best weight-value-relation into the knapsack until nothing else fits. This does not necessarily lead to the best solution, and even if it does, it may take excessively long. But such a procedure, however, is still good as a heuristic.

In the *direct-numerical approach*, a numerical algorithm provides the exact solution modulo round-off error. The simplex algorithm for problems in linear optimization of the kind "solve $\max_x c^T x$ under the constraint $Ax \leq b$" is an example of this approach. For the *approximate-numerical approach*, however, one refers to an approximation method in order to approximate the solution of the model as accurately as possible. This task splits into two parts: first, the *discretization* of the continuous problem, and second, the *solution* of the discrete problem. Both parts are concerned with the question of convergence. The discretization should be of the type such that an increase in effort (i.e., an increase in resolution) will lead to asymptotically better approximations, and the (mostly iterative) solution technique for the discretized problem should first of all converge as well as converge rapidly to its solution.

The approximate-numerical approach is certainly the most important one for problems in numerical simulation; we will encounter it repeatedly in the following chapters.

1.3.2 Assessment

Of central significance in a simulation is the assessment of the computed results. The goal of *validation* is to determine whether we have used the correct (or rather, a suitable) model ("Do we solve relevant equations?"). By contrast, *verification*, takes a look at the algorithm and software program with the purpose to determine whether the given model has been solved correctly ("Do we solve the given equations correctly?"). Even in the case of two affirmative "Yes!" answers, the examination of aspects concerning the accuracy of the result as well as the invested effort is still remaining.

There are several possibilities for validating the computed simulation results. The classical procedure is the *comparison with experimental tests*—whether these are 1:1-experiments, as for example in crash tests, or scaled laboratory experiments, for example tests in a wind tunnel with downsized prototypes. Sometimes, however, this approach is prohibited for feasibility reasons or the required effort. But even when experiments can be performed one should use caution: First of all, it is easy for small differences to arise between the simulated and the experimental scenarios; secondly, one has to be very cautious with respect to the scaling of quantities (it is possible that certain effects do not appear on small scales); and thirdly, there may occur sporadic and systematic mistakes in measuring—computers and their operators do not have a monopoly on bugs!

A-posteriori observations provide an additional (and in general very inexpensive) possibility for validation—true to the motto that "one is always smarter afterwards". *Reality tests* compare the predicted with the actual result; one may think of the weather, the stock market or military scenarios. *Satisfaction tests* determine whether the desired result has materialized to a sufficient degree. Examples for applications are systems for traffic control as well as illumination and animation models in computer graphics.

In contrast, the *plausibility tests* remain on a purely theoretical level, as frequently encountered in physics. Here, one checks whether the simulation results are in contradiction to other, previously verified theories. Naturally, one must not be too conservative—possibly, the common doctrine errs and the simulator is correct!

Finally, there exists the option to perform a *model comparison*, i.e., to compare the results of simulations which are based on different models.

No matter how one proceeds—one needs to use caution before one draws conclusions from validations. There are various sources for errors; pears literally wait to be compared to apples, and Muenchhausen has been known to pull himself out of the swamp by his own hair ...

The topic of verification leads to convergence proofs, etc. for the employed algorithms on one hand and to correctness proofs for the designed programs on the other hand. While the former are well established and are known to be a favorite pasttime for numerical analysts, the later are still in their infancy. It is not so much the case that computer science has not achieved anything in this respect. It is rather the case that the simulation business—quite opposite to other software-intensive fields—is positioned extremely shirt-sleeved (not systematic in its approach): Here, one usually programs but hardly ever develops software. The pain threshhold seems to have been reached only recently, and requests for a formal framework (and thus for better possibilities for verification) become louder.

Even the aspect of *accuracy* is more complex than it appears at first sight. The first coming to mind is accuracy with respect to the quality of input data. If the input data is available in the form of measured data with an accuracy of three decimal places, then one cannot expect the result to be accurate to eight decimal places. In addition, one needs to keep an eye on the accuracy in relation to the problem—which can at times be problematic. In many cases, a model which produces errors below one per cent is considered completely sufficient. But in an election poll, for example, being half a percent off the mark can turn everything upside down—and thus render the modeling and simulation completely useless! Another factor is the need for security: Can one live with statements that reflect averaged values, or is it necessary to reflect a guaranteed worst-case-bound?

And finally the *cost question*—what effort (wrt. time for implementation, memory, computation or response time) was invested to reach the simulation result? In this context, it is important to neither consider the obtained benefit (e.g., the accuracy of the result) nor the invested effort individually, but rather in relation to each other. Basically, it is neither the best nor the cheapest car that one wants to buy, but rather the one with the best price-performance ratio.

http://www.springer.com/978-3-642-39523-9

Modeling and Simulation
An Application-Oriented Introduction
Bungartz, H.-J.; Zimmer, S.; Buchholz, M.; Pflüger, D.
2014, XIV, 413 p. 158 illus., Hardcover
ISBN: 978-3-642-39523-9