$$F_1 x_1 + F_2 x_2 \leq F \tag{1.43}$$

$$P_1 x_1 + P_2 x_2 \leq P \tag{1.44}$$

$$S_1 x_1 + S_2 x_2 \rightarrow \max \tag{1.45}$$

Here, Equation 1.41 expresses the fact that the farmer cannot plant a negative area, Equation 1.42 the fact that no more than the given A square kilometers of farm land can be used, Equations 1.43 and 1.44 express the fertilizer and insecticide limits, respectively, and Equation 1.45 is the required revenue maximization. Taking Equations 1.41–1.45 as M, the system S as the farm land and the question Q, "How many square kilometers should be planted with wheat versus barley to maximize the revenue?", a mathematical model (S, Q, M) is obtained. For any set of parameter values for A, F, P, ..., the problem can again be easily solved using *Maxima*. This is done in the *Maxima* program `Farm.mac` which you find in the book software (see Appendix A). Let us look at the essential commands of this code:

```
1: load(simplex);
2: U:[x1>=0
3: ,x2>=0
4: ,x1+x2<=A
5: ,F1*x1+F2*x2 <=F
6: ,P1*x1+P2*x2<=P];
7: Z:S1*x1+S2*x2;
8: maximize_lp(Z,U);
```

(1.46)

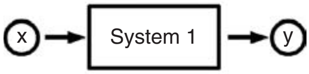
Line 1 of this code loads a package required by *Maxima* to solve linear programming problems. Lines 2–6 define the inequalities, corresponding to Equations 1.41–1.44 above. Note that lines 2–6 together make up a single command that stores the list of inequalities in the variable `U`. Line 7 defines the function `Z` that is to be maximized, and the problem is then solved in line 8 using *Maxima*'s `maximize_lp` command. Based on the parameter settings in `Farm.mac`, *Maxima* produces the following result:

```
[100, [x2 = 50, x1 = 0]]
```

This means that a maximum revenue of 100 is obtained if the farmer plants barley only (50 square kilometers).

1.5.6
Modeling a Black Box System

In Section 1.3 it was mentioned that the systems investigated by scientists or engineers typically are "input–output systems", which means they transform the given input parameters into output parameters. Note that the previous examples were indeed referring to such "input–output systems". In the tin example, the radius and height of the tin are input parameters and the surface area of the tin is



(a)

x (N)	10	20	30	40	50
y (cm)	3	5	11	12	16

(b)

Fig. 1.8 (a) System 1 with input x (N) and output y (cm). (b) System 1 data (file `spring.csv` in the book software).

the output parameter. In the plant growth example, the growth rate of the plant and its initial biomass is the input and the resulting time–biomass curve is the output (details in Chapter 3). In the tank example, the geometrical data of the tank and the concentration distribution are input parameters while the mass of the substance is the output. In the linear programming examples, the areas planted with wheat or barley are the input quantities and the resulting revenue is the output. Similarly, all systems in the examples that will follow can be interpreted as input–output systems.

The exploration of an example input–output system in some more detail will now lead us to further important concepts and definitions. Assume a "system 1" as in Figure 1.8 which produces an output length y (centimeters) for every given input force x [N]. Furthermore, assume that we do not know about the processes inside the system that transform x into y, that is, let this system be a "black box" to us as described above. Consider the following problem:

> Q: Find an input x that generates an output $y = 20$ cm.

This defines the question Q of the mathematical model (S, Q, M) that we are going to define. S is the "system 1" in Figure 1.8a, and we are now looking for an appropriate set of mathematical statements M that can help us to answer Q.

All that the investigator of system 1 can do is to produce some data using the system, hoping that these data will reveal something about the processes occurring inside the "black box". Assume that the data in the file `spring.csv` (which you find in the `PhenMod/LinReg` directory of the book software, see Appendix A) have been obtained from this system, see Figure 1.8b. To see what happens, the investigator will probably produce a plot of the data as in Figure 1.9a. Note that the plots in Figure 1.9 were generated using the scatter plot option of *OpenOffice.org Calc* (see Appendix A on how you can obtain this software). Figure 1.9a suggests that there is an approximately linear dependence between the x- and y-data. Mathematically, this means that the function $y = f(x)$ behind the data is a straight line:

$$f(x) = ax + b \tag{1.47}$$

Now the investigator can apply a statistical method called *linear regression* (which will be explained in detail in Section 2.2) to determine the coefficients a and b of this equation from the data, which leads to the "regression line"

$$f(x) = 0.33x - 0.5 \tag{1.48}$$

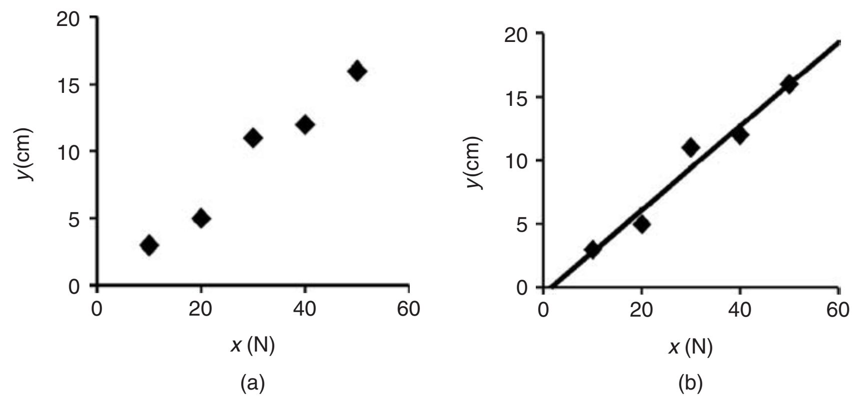


Fig. 1.9 (a) Plot of the data in `spring.csv`. (b) System 1 data with regression line. Both plots generated using *Calc*, see Section 2.1.1.1.

Figure 1.9b shows that there is a good coincidence or, in statistical terminology, a good ''fit'' between this regression line and the data. Equation 1.48 can now be used as the M of a mathematical model of system 1. The question Q stated above (''Which system input x generates a desired output $y = 20$ cm? '') can then be easily answered by setting $y = f(x) = 20$ in Equation 1.48, that is,

$$20 = 0.33x - 0.5 \tag{1.49}$$

which gives $x \approx 62.1$ N. Of course, this is just an *approximate result* for several reasons. First of all, Figure 1.9 shows that there are some deviations between the regression line and the data. These deviations may be due to measurement errors, but they may also reflect some really existing effects. If the deviations are due to measurement errors, then the precise location of the regression line and hence, the prediction of x for $y = 20$ cm is affected by these errors. If, on the other hand, the deviations reflect some really existing effects, then Equation 1.48 is no more than an approximate model of the processes that transform x into y in system 1, and hence, the prediction of x for $y = 20$ cm will be only approximate. Beyond this, predictions based on data such as the data in Figure 1.8b are always approximate for principal reasons. The y-range of these data ends at 16 cm, and system 1 may behave entirely different for y-values beyond 16 cm which we would not be able to see in such a data set. Therefore, the experimental validation of predictions derived from mathematical models is always an indispensable part of the modeling procedure (see Section 1.2). See also Chapter 2 for a deeper discussion of the quality of predictions obtained from black box models.

The example shows the *importance of statistical methods* in mathematical modeling. First of all, statistics itself is a collection of mathematical models that can be used to describe data or to draw inferences from data [19]. Beyond this, statistical methods provide a necessary link between nonstatistical mathematical models and the real world. In mathematical modeling, one is always concerned with experimental data, not only to validate model predictions, but also to develop hypotheses about the system, which help to set up appropriate equations. In the example, the data led us to the hypothesis that there is a linear relation between x and y. We have used a plot of the data (Figure 1.9) and the regression method to find the coefficients in Equation 1.48. These are methods of descriptive statistics, which can be used to summarize or describe data. Beyond this, inferential statistics provides methods that allow conclusions to be drawn from data in a way that accounts for randomness and uncertainty. Some important methods of descriptive and inductive statistics will be introduced below (Section 2.1).

Note 1.5.12 Statistical methods provide the link between mathematical models and the real world.

The reader might say that the estimate of x above could also have been obtained without any reference to models or computations, by a simple tuning of the input using the real, physical system 1. We agree that there is no reason why models should be used in situations where this can be done with little effort. In fact, we do not want to propose any kind of a fundamentalist ''mathematical modeling and simulation'' paradigm here. A *pragmatic approach* should be used, that is, any problem in science and engineering should be treated using appropriate methods, may this be mathematical models or a tuning of input parameters using the real system. It is just a fact that in many cases the latter cannot be done in a simple way. The generation of data such as in Figure 1.8 may be expensive, and thus, an experimental tuning of x toward the desired y may be inapplicable. Or, the investigator may be facing a very complex interaction of several input and output parameters, which is rather the rule than the exception as explained in Section 1.1. In such cases, the representation of a system in mathematical terms can be the only efficient way to solve the problem.

1.6 Even More Definitions

1.6.1 Phenomenological and Mechanistic Models

The mathematical model used above to describe system 1 is called a *phenomenological model* since it was constructed based on experimental data only, treating the system as a black box, that is, without using any information about the internal processes occurring inside system 1 when x is transformed into y. On the other hand, models that are constructed using information about the system S are called *mechanistic models*, since such models are virtually based on a look into the internal mechanics of S. Let us define this as follows [11]:

Definition 1.6.1 (Phenomenological and mechanistic models) A mathematical model (S, Q, M) is called
- *phenomenological*, if it was constructed based on experimental data only, using no a priori information about S,
- *mechanistic*, if some of the statements in M are based on a priori information about S.

Phenomenological models are also called *empirical models, statistical models, data-driven models* or *black box models* for obvious reasons. Mechanistic models for which all necessary information about S are available are also called *white box models*. Most mechanistic models are located somewhere between the extreme black and white box cases, that is, they are based on some information about S while some other important information is unavailable. Such models are sometimes called *gray box models* or *semi-empirical models* [20].

To better understand the differences between phenomenological and mechanistic models, let us now construct an alternative mechanistic model for system 1 (Figure 1.8). Above, we have treated system 1 as a black box, that is, we have used no information about the way in which system 1 transforms some given input x into the output y (Figure 1.8). Let us now assume that the internal mechanics of system 1 looks as shown in Figure 1.10, that is, assume that system 1 is a mechanical spring, x is a force acting on that spring, and y is the resulting elongation. This is now an *a priori information* about system 1 in the sense of Definition 1.6.1 above, and it can be used to construct a mechanistic mathematical model based on elementary physical knowledge. As is well known, mechanical springs can be described by Hooke's law, which in this case reads

$$x = k \cdot y \tag{1.50}$$

where k is the spring constant (newtons per centimeter), a measure of the elasticity of the spring. The parameter k is either known (e.g. from the manufacturer of the spring), or estimated based on data such as those in Figure 1.8. Now the following mechanistic mathematical model (S, Q, M) is obtained:
- S: System 1
- Q: Which system input x generates a desired output of $y = 20\,\text{cm}$?
- M: Equation 1.50

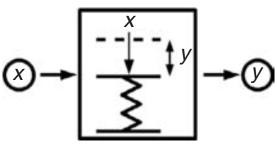


Fig. 1.10 Internal mechanics of system 1.

Based on this model, question Q can be answered as before by setting $y = 20$ cm in the model equation (1.50), which leads to

$$x = k \cdot 20 \tag{1.51}$$

that is, we can answer the question Q depending on the value of the spring constant, k. For example, assuming a value of $k \approx 3.11\ \text{N cm}^{-1}$ for the spring constant, we would get the same estimate $x \approx 62.1$ N as above. The mechanistic model of system 1 has several important advantages compared to the phenomenological model, and these advantages are *characteristic advantages of the mechanistic approach*. First of all, mechanistic models generally allow better predictions of system behavior. The phenomenological model equation (1.48) was derived from the data in Figure 1.8. These data involve forces x between 10 and 50 N. As mentioned below in our discussion of regression methods, this means that one can expect Equation 1.48 to be valid only close to this range of data between 10 and 50 N. The mechanistic model equation (1.50), on the other hand, is based on the well-established physical theory of a spring. Hence, we have good reason to expect its validity even outside the range of our own experimental testing.

Mechanistic models do also allow *better predictions* of modified systems. Assume for example that system 1 in Figure 1.10 is replaced by a system 2 that consists of two springs. Furthermore, assume that each of these system 2 springs has the same spring constant k as the system 1 spring. Then, in the phenomenological approach, the model developed for system 1 would be of no use, since we would not know about the similarity of these two systems (remember that the phenomenological approach assumes that no details are known about the internal mechanics of the system under consideration). This means that a new phenomenological model would have to be developed for system 2. A new data set similar to Figure 1.8 would be required, appropriate experiments would have to be performed, and afterwards, a new regression line similar to Figure 1.9 would have to be derived from the data. In the mechanistic approach, on the other hand, Hooke's law would immediately tell us that in the case of two springs the appropriate modification of Equation 1.50 is

$$x = 2k \cdot y \tag{1.52}$$

Another advantage of mechanistic models is the fact that they usually involve *physically interpretable parameters*, that is, parameters which represent real properties of the system. To wit: the numerical coefficients of the phenomenological model equation 1.47 are just numbers which cannot be related to the system. The parameter k of the mechanistic model equation 1.50, on the other hand, can be related to system properties, and this is of particular importance when we want to optimize system performance. For example, if we want smaller forces x to be required for a given elongation y, then in the phenomenological approach we would have to test a number of systems 2, 3, 4, . . . , until we would eventually arrive at some system with the desired properties. That is, we would have to apply a trial-and-error method. The mechanistic model, on the other hand, tells us exactly what we have to do: we have to replace the system 1 spring with a spring having a smaller spring constant k, and this will reduce the force x required for a given elongation y. In this simple example, it may be hard to imagine that someone would really use the phenomenological approach instead of Hooke's law. But the example captures an essential difference between phenomenological and mechanistic models, and it tells us that we should use mechanistic models if possible.

So, if mechanistic models could be set up easily in every imaginable situation, we would not have to talk about phenomenological models here. However, in many situations, it is not possible or feasible to use mechanistic models. As an essential prerequisite, *mechanistic models need a priori knowledge of the system*. If nothing is known about the system, then we are in the ''black box'' situation and have to apply phenomenological models. Suppose, for example, we want to understand why some roses wilt earlier than others (this example will be explained in more detail in Section 2.3). Suppose we assume that this is related to the concentrations of certain carbohydrates that can be measured. Then we cannot set up a mechanistic model as long as we do not know all the relevant processes that connect those carbohydrate concentrations with the observed freshness of the rose. Unless these processes are known, all we can do is to produce some data (carbohydrate concentration versus some appropriate measure of rose freshness) and analyze these data using phenomenological models.

This kind of situation where little is known about the system under investigation is rather the rule than the exception, particularly at early stages of a scientific investigation, or at the early stages of a product development in engineering. We may also be in a situation where we principally know enough details about the system under investigation, but where the system is so complex that it would take too much time and resources to setup a mechanistic model. An example is the optimization of the wear resistance of composite materials: Suppose that a composite material is made of the materials $M_1, M_2, \ldots, M_n$, and we want to know how the relative proportions of these materials should be chosen in order to maximize the composite materials resistance to wear. Then, the wear resistance of the composite material can depend in an extremely complex way on its composition. The author has investigated a situation of this kind where mechanistic modeling attempts failed due to the complexity of the overall system, and where a black box-type phenomenological neural network approach (see Section 2.5) was used instead [21]. An important *advantage of phenomenological models* is that they can be used in black box situations of this kind, and that they typically require much less time and resources. Pragmatic considerations should decide which type of model is used in practice. A mechanistic model will certainly be a bad choice if we need three weeks to make it work, and if it does not give substantially better answers to our question Q compared to a phenomenological model which can be set up within a day.

Note 1.6.1 (Phenomenological vs. mechanistic) *Phenomenological models* are universally applicable, easy to set up, but limited in scope. *Mechanistic models* typically involve physically interpretable parameters, allow deeper insights into system performance and better predictions, but they require a priori information on the system and often need more time and resources.

1.6.2
Stationary and Instationary models

It was already mentioned above that the question Q is an important factor that determines the appropriate mathematical model (S, Q, M). As an example, we have considered the alternative treatment of mechanical problems with the equations of classical or relativistic mechanics depending on the question Q that is investigated. In the system 1 example, we have used Q: "Which system input x generates a desired output of $y = 20\,\text{cm}$? ". Let us now modify this Q in order to find other important classes of mathematical models. Consider the following question:

Q: If a constant force x acts on the spring beginning with $t = 0$, what is the resulting elongation $y(t)$ of the spring at times $t > 0$?

This question cannot be answered based on the models developed above. The phenomenological model (Equation 1.48) as well as the mechanistic model (Equation 1.50) both refer to the so-called stationary state of system 1. This means that the elongation y expressed by these equations represents the time-independent (= stationary) state of the spring which is achieved after the spring has been elongated into the state of equilibrium where the force x exactly matches the force of the spring. On the other hand, the above question asks for the instationary (i.e. time-dependent) development of the elongation $y(t)$, beginning with time $t = 0$ when the force x is applied to the spring. To compute this $y(t)$, an instationary mathematical model (S, Q, M) is needed where the mathematical statements in M involve the time t. Models of this kind can be defined based on ordinary differential equations (details in Chapter 3). To make this important distinction between stationary and instationary models precise, let us define

Definition 1.6.2 (Stationary/instationary models) A mathematical model (S, Q, M) is called

- *instationary*, if at least one of its system parameters or state variables depends on time and
- *stationary* otherwise.

1.6.3
Distributed and Lumped models

Suppose now that the spring in system 1 broke into pieces under normal operational conditions, and that it is now attempted to construct a more robust spring. In such

3
Mechanistic Models I: ODEs

3.1
Distinguished Role of Differential Equations

As was explained, *mechanistic models* use information about the internal ''mechanics'' of a system (Definition 1.6.1). Referring to Figure 1.2, the main difference between phenomenological models (discussed in Chapter 2) and mechanistic models lies in the fact that phenomenological models treat the system as a black box, while in the mechanistic modeling procedure one virtually takes a look inside the system and uses this information in the model. This chapter and the following Chapter 4 treat *differential equations*, which is probably the most widely used mathematical structure of mechanistic models in science and engineering. Differential equations arise naturally, for example, as mathematical models of physical systems. Roughly speaking, differential equations are simply ''equations involving derivatives of an unknown function''. Their distinguished role among mechanistic models used in science and engineering can be explained by the fact that both scientists and engineers aim at the understanding or optimization of processes within systems.

The word ''process'' itself already indicates that a process involves a situation where ''something happens'', that is, where some quantities of interest change their values. Absolutely static ''processes'' where virtually ''nothing happens'' would be hardly of any interest to scientists or engineers. Now if it is true that some quantities of interest relating to a process under consideration change their values, then it is also true that such a process involves rates of changes of these quantities, which means in mathematical terms that it involves derivatives – and this is how ''equations containing derivatives of an unknown function'' or differential equations come into play. In many of the examples treated below it will turn out that it is natural to use rates of changes to formulate the mathematics behind the process, and hence to write down differential equations, while it would not have been possible to find appropriate equations without derivatives.

Note 3.1.1 (Distinguished role of differential equations)
1. Mechanistic models consider the processes running inside a system.
2. Typical processes investigated in science and engineering involve rates of changes of quantities of interest.
3. Mathematically, this translates into equations involving derivatives of unknown functions, i.e. differential equations.

Differential equations are classified into *ordinary and partial differential equations*. It is common to use *ODE* and *PDE* as abbreviations for ordinary and partial differential equations, respectively. This section is devoted to ODEs that involve derivatives with respect to only one variable (time in many cases), while PDEs (treated in Chapter 4) involve derivatives with respect to more than one variable (typically, time and/or space variables). In Section 3.2, mechanistic modeling is introduced as some kind of ''systems archaeology'', along with some first simple ODE examples that are used throughout this chapter. The procedure to set up ODE models is explained in Section 3.4, and Section 3.5 provides a theoretical framework for ODEs. Then, Sections 3.6–3.8 explain how you can solve ODEs either in closed form (i.e. in terms of explicit formulas) or using numerical procedures on the computer. ODE models usually need to be fitted to experimental data, that is, their parameters need to be determined such that the deviation of the solution of the ODE from experimental data is minimized (similar to the regression problems discussed in Chapter 2). Appropriate methods are introduced in Section 3.9, before a number of additional example applications are discussed in Section 3.10.

3.2 Introductory Examples

3.2.1 Archaeology Analogy

If one wants to explain what it really is that makes mechanistic modeling a very special and exciting thing to do, then this can hardly be done better than by the ''archaeology analogy'' of the French twentieth century philosopher Jacques Derrida [94]:

Note 3.2.1 (Derrida's archaeology analogy) ''Imagine an explorer arrives in a little-known region where his interest is aroused by an expanse of ruins, with remains of walls, fragments of columns, and tablets with half-effaced and unreadable inscriptions. He may content himself with inspecting what lies exposed to his view, with questioning the inhabitants (. . .) who live in the vicinity, about what tradition tells them of the history and meaning of these archaeological remains, and with noting what they tell him – and he proceeds upon his journey. But he may act differently. He may have brought picks, shovels, and spades with him, and he may set the inhabitants to work with these implements. Together with them he may start upon the ruins, clearing away rubbish and, beginning from the visible remains, uncover what is buried. If his work is crowned with success, the discoveries are self-explanatory: the ruined walls are part of the ramparts of a palace or a treasure house; fragments of columns can be filled out into a temple; the numerous inscriptions, which by good luck, may be bilingual, reveal an alphabet and a language, and, when they have been deciphered and translated, yield undreamed-of information about the events of the remote past . . .''

Admittedly, one may not necessarily consider archaeology as an exciting thing to do, particularly when it is about sitting for hours at inconvenient places, scratching dirt from pot sherds, and so on. However, what Derrida describes is what might be called the exciting part of archaeology: revealing secrets, uncovering the buried, and exploring the unknown. And this is exactly what is done in mechanistic modeling. A mechanistic modeler is what might be called a *system archaeologist*. Looking back at Figure 1.2, he is someone who virtually tries to break up the solid system box in the figure, thereby trying to uncover the hidden internal system mechanics. A phenomenological modeler, in contrast, just walks around the system, collecting and analyzing the data which it produces. As Derrida puts it, he contents himself ''with inspecting what lies exposed to view''.

The exploration of subsurface structures by archeologists based on ground-penetrating radar provides a nice allegory for the procedure in mechanistic modeling. In this method, the archaeologist walks along a virtual x axis, producing scattered data along that x axis similar to a number of datasets that are investigated below. In the phenomenological approach, one would be content with an explanation of these data in terms of the input signal sent into the soil, for example, using appropriate methods from Chapter 2, and with no attempt toward an understanding of the soil structures generating the data. What the archaeologist does, however, is mechanistic modeling: based on appropriate models of the measurement procedure, he gains information about subsurface structures. Magnetic resonance imaging (MRI) and computed tomography (CT) are perhaps the most fascinating technologies of this kind – everybody knows these fantastically detailed pictures of the inside of the human body.

Note 3.2.2 (Objective of mechanistic modeling) Datasets contain information about the internal mechanics of the data-generating system. Mechanistic modeling means to uncover the hidden internal mechanics of a system similar to an archaeologist, who explores subsurface structures using ground-penetrating radar data.

3.2.2 Body Temperature

Now let us try to become system archaeologists for ourselves, starting with simple data sets and considerations. To begin with, suppose you do not feel so good today and decide to measure your body temperature. Using a modern clinical thermometer, you will have the result within a few seconds, usually indicated by a beep signal of your thermometer. You know that your thermometer needs these few seconds to bridge the gap between room and body temperature. Modern clinical thermometers usually have a display where this process of adjustment can be monitored, or better: *could* be monitored if you could see the display during measurement. Be that as it may, the dataset in Figure 3.1a shows data produced by the author using a clinical thermometer. The figure was produced using the dataset `fever.csv` and the Maxima program `FeverDat.mac` from the book software (see the description of the book software in Appendix A). `FeverDat.mac` does two things: it reads the data from `fever.csv` using *Maxima*'s `read_nested_list` command, and then it plots these data using the `plot2d` command (see `FeverDat.mac` and *Maxima*'s help pages for the exact syntax of these commands).

3.2.2.1 Phenomenological Model

Remembering what we have learned about phenomenological modeling in the previous chapter, it is quite obvious what can be done here. The data points follow a very simple and regular pattern, and hence it is natural to use an explicit function $T(t)$ describing that pattern, which can then be fitted to the data using *nonlinear regression* as described in Section 2.4. Clearly, the data in Figure 3.1a describe an essentially exponential pattern (imagine a 180° counterclockwise rotation of the data). Mathematically, this pattern can be described by the function

$$T(t) = T_b - (T_b - T_0) \cdot e^{-r \cdot t} \tag{3.1}$$

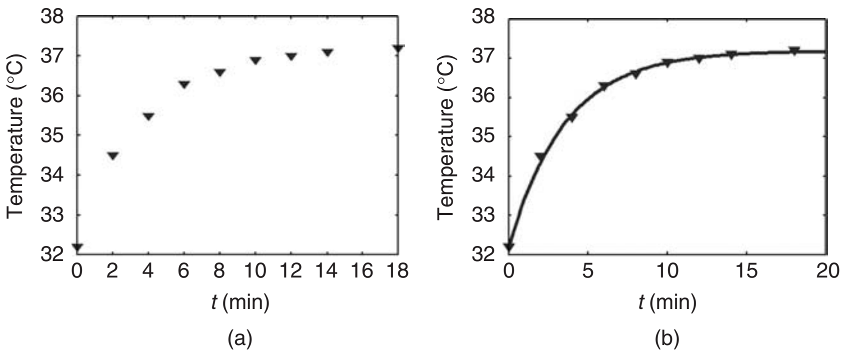


Fig. 3.1 (a) Body temperature data. (b) Body temperature data (triangles) and function $T(t)$ from Equation 3.4.

The parameters of this function have natural interpretations: T_0 is the initial temperature since $T(0) = T_0$, T_b is the body temperature since $\lim_{t\to\infty} T(t) = T_b$, and r controls the rate of temperature adjustment between T_0 and T_b. As Figure 3.1a shows, the values of T_0 and T_b should be slightly above 32 and 37 °C, respectively. Based on `fever.csv`, let us set $T_0 = 32.2$ and $T_b = 37.2$. To estimate r, we can, for example, substitute the datapoint ($t = 10$, $T = 36.9$) from `fever.csv` in Equation 3.1

$$36.9 = 37.2 - 5 \cdot \mathrm{e}^{-r \cdot 10} \tag{3.2}$$

which leads to

$$r = -\frac{\ln(0.06)}{10} \approx 0.281 \tag{3.3}$$

Note that Equation 3.3 can also be obtained using *Maxima*'s `solve` command, see the code `FeverSolve.mac` in the book software. Similar to the code (1.7) discussed in Section 1.5.2, the `solve` command produces several solutions here. Nine of these solutions are complex numbers, while one of the solutions corresponds to Equation 3.3. Using Equation 3.3, $T(t)$ can now be written as

$$T(t) = 37.2 - 5 \cdot \mathrm{e}^{\ln(0.06)/10 \cdot t} \tag{3.4}$$

Plotting this function together with the body temperature data from Figure 3.1a, Figure 3.1b is obtained. Again, this plot was generated using *Maxima*: see `FeverExp.mac` in the book software. As the figure shows, the function $T(t)$ fits the data very well. Remember our discussion of nonlinear regression in Section 2.4 where a quantity called *pseudo-*R^2 was introduced in formula (2.35) as a measure of the quality of fit. Here, the *Maxima* program `FeverExp.mac` computes an pseudo-R^2 value of 99.8%, indicating an almost perfect fit of the model to the data. Comparing Equation 2.35 with its implementation in `FeverExp.mac`, you will note that $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ is realized in the form `(y-yprog).(y-yprog)`, where the "`.`" denotes the scalar product of vectors, which multiplies vectors with components $y_i - \hat{y}_i$ in this case (see the *Maxima* help pages for more details on *Maxima*'s vector operation syntax). Note that the parameters of the model $T(t)$ have been obtained here using heuristic arguments. Alternatively, they could also be estimated using the nonlinear regression procedure described in Section 2.4.

3.2.2.2 **Application**

The model in Equation 3.4 can now be used to answer all kinds of questions related to the body temperature data. For example, it could be used to estimate the variation of the total measurement time (i.e. the time until the final measurement value is achieved) with varying starting temperatures of the thermometer. Or, it could be used to accelerate the measurement procedure using estimates of T_b based on the available data, and so on. Remember our definition of mathematical models in Section 1.4 above: a mathematical model is a set of mathematical statements that

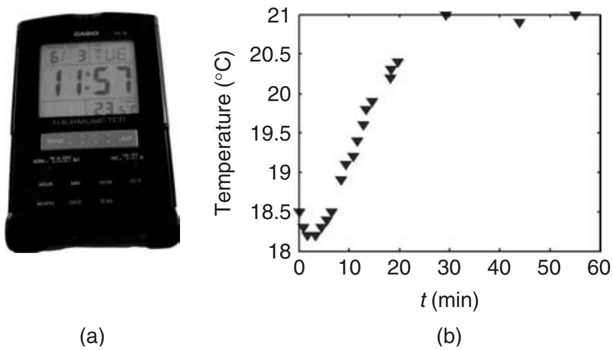


Fig. 3.2 (a) Alarm clock with temperature sensor. (b) Room temperature data.

can be used to answer a question which we have related to a system. As was pointed out there and in Note 1.2.2, the best mathematical model is the smallest and simplest set of mathematical statements that can answer the given question. In this sense, we can say that the phenomenological model (3.4) is the best mathematical model of the body temperature data probably with respect to most questions that we might have regarding the body temperature data. In Section 3.4.1, however, we will see that Equation 3.4 can also be derived from a mechanistic modeling approach.

3.2.3 Alarm Clock

Let us consider now a data set very similar to the body temperature data, but with a little complication that will lead us beyond the realms of phenomenological modeling. Suppose you enter a warm room with a temperature sensor in your hand, and you write down the temperature output of that sensor beginning with time $t = 0$ corresponding to the moment when you enter the warm room. At a first glance, this is a situation perfectly similar to the body temperature measurement, and you would probably expect a qualitative pattern of your data similar to Figure 3.1a. Now suppose that your data look as shown in Figure 3.2b; that is, your data are qualitatively different from those in Figure 3.1a, showing an initial decrease in the temperature even after you entered the warm room at time 0. Figure 3.2b has been produced using the *Maxima* code `RoomDat.mac` and the data `room.csv` in the book software (similar to `FeverDat.mac` discussed in Section 3.2.2).

3.2.3.1 Need for a Mechanistic Model

In principle, these data could be treated using a phenomenological model as before. To achieve this, we would just have to find some suitable function $T(t)$, which exhibits the same qualitative behavior as the data shown in Figure 3.2b. For example, a polynomial could be used for $T(t)$ (see the polynomial regression example in Section 2.2.6) or $T(t)$ could be expressed as a combination of a function similar to Equation 3.1 with a second-order polynom. Afterwards, the parameters of $T(t)$ would have to be adjusted such that $T(t)$ really matches the data, similar to our treatment of the body temperature data. As before, the function $T(t)$ could then be used, for example, to estimate the total measurement time depending on the starting temperature, and so on. However, it is obvious that any estimate obtained in this way would be relatively uncertain as long as we do not understand the initial decrease in the temperature in Figure 3.2b. For example, if we would use the phenomenological model $T(t)$ to estimate the total measurement time for a range of starting temperatures, then we would implicitly assume a similar initial decrease in the temperature for the entire range of starting temperatures under consideration – but can this be assumed? We do not know unless we understand the initial decrease in the temperature.

The initial decrease in the temperature data shown in Figure 3.2b contains information about the system that should be used if we want to answer our questions regarding the system with a maximum of precision. The data virtually want to tell us something about the system, just as ground-penetrating radar data tell the archaeologist something about subsurface structures. To construct a phenomenological model of temperature data, only the data themselves are required, that is, one virtually just looks at the display of the device generating the temperature data. Now we have to change our point of view toward a look at the data-generating device itself, and this means we shift toward mechanistic modeling.

Note 3.2.3 (Information content of ''strange effects'') Mechanistic models should be used particularly in situations where ''strange effects'' similar to the alarm clock data can be observed. They provide a means to explain such effects and to explore the information content of such data.

3.2.3.2 Applying the Modeling and Simulation Scheme

Figure 3.2a shows the device that produced the data in Figure 3.2b: an alarm clock with temperature display. The data in Figure 3.2b were produced when the author performed a test of the alarm clock's temperature sensor, measuring the temperature inside a lecture room. Initially, he was somewhat puzzled by the decrease in the measurements after entering the warm lecture room, but of course the explanation is simple. The alarm clock was cheap and its temperature sensor an unhasty and lethargic one – an unbeatable time span of 30 min is required to bridge the gap between 18 and 21 °C in Figure 3.2b. Before the author entered the lecture room at time $t = 0$, he and the alarm clock were outdoors for some time at an ambient temperature around 12 °C. The initial decrease in the temperature measurements, thus, obviously meant that the author had disturbed the sensor inside the alarm clock when it still tried to attain that 12 °C.

Now to set up a mechanistic mathematical model that can describe the pattern of the data in Figure 3.2b, we can follow the steps of the *modeling and simulation scheme* described in Note 1.2.3 (Section 1.2.2). This scheme begins with the *definitions step*, where a question to be answered or a problem to be solved is defined. Regarding Figure 3.2b, a natural question would be

Q_1: How can the initial decrease of the temperature data be explained?

Alternatively, we could start with the problem

Q_2: Predict the final temperature value based on the first few data points.

In the *systems analysis step* of the scheme in Note 1.2.3, we have to identity those parts of the system that are relevant for Q_1 and Q_2. Remember the car example in Section 1.1, where the systems analysis step led us from the system ‘‘car’’ in its entire complexity to a very simplified car model comprising only tank and battery (Figure 1.1). Here, our starting point is the system ‘‘alarm clock’’ in its entire complexity, and we need to find a simplified model of the alarm clock now in the systems analysis step, guided by our questions Q_1 and Q_2. Obviously, any details of the alarm clock not related to the temperature measurement can be skipped in our simplified model – just as any details of the car not related to the problem ‘‘The car is not starting’’ were skipped in the simplified model of Figure 1.1.

Undoubtedly, the temperature sensor is an indispensable ingredient of any simplified model that is expected to answer our questions Q_1 and Q_2. Now remember that we stated in Note 1.2.2 that the simplest model is the best model, and that one, thus, should always start with the simplest imaginable model. We have the simplest possible representation of the temperature sensor in our model if we just consider the temperature T_s displayed by the sensor, treating the sensor's internal construction as a black box. Another essential ingredient of the simplified model is the ambient temperature T_a that is to be measured by the sensor. With these two ingredients, we arrive at the simplified alarm clock model in Figure 3.3, which we call *Model A*. Note that Model A is not yet a mathematical model, but rather what we have called a *conceptual model* in Section 1.2.5. Model A represents an intermediate step that is frequently used in the development of mathematical models. It identifies state variables T_s and T_a of the model to be developed and it provides an approximate sketch of their relationship, with T_s being drawn inside a

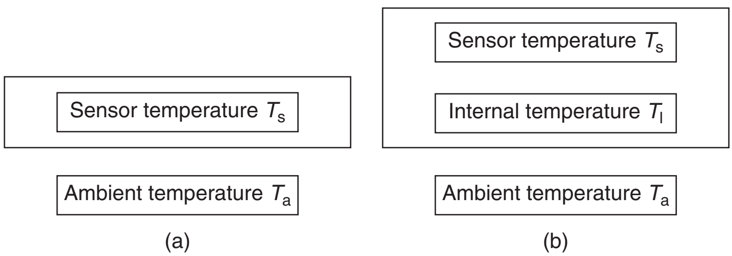


Fig. 3.3 Simplified models of the alarm clock: (a) Model A and (b) Model B.

rectangle symbolizing the alarm clock, T_a outside that rectangle. But although we already know the system S and the question Q of the mathematical model to be developed, Model A is still not a complete description of a mathematical model (S, Q, M) since mathematical statements M that could be used to compute the state variables are missing.

3.2.3.3 Setting Up the Equations

In this case, it is better to improve Model A a little bit before going into a formulation of the mathematical statements, M. Based on Model A, the only thing that the sensor (represented by T_s) "sees" is the ambient air temperature. But if this were true, then the initial decrease in the temperature data in Figure 3.2b would be hard to explain. If the sensor "sees" only the ambient air temperature, then its temperature should increase as soon as we enter the warm room at time $t = 0$. The sensor obviously somehow memorizes the temperatures of the near past, and this *temperature memory* must be included into our alarm clock model if we want to reproduce the data of Figure 3.2b. Now there are several possibilities how this temperature memory could be physically realized within the alarm clock. First of all, the temperature memory could be a consequence of the temperature sensor's internal construction. As a first, simple idea one might hypothesize that the temperature sensor always "sees" an old ambient temperature $T_a(t - t_{\text{lag}})$ instead of the actual ambient temperature $T_a(t)$. If this were true, the above phenomenological model for temperature adaption, Equation 3.1, could be used as follows. First, let us write down the ambient temperature T_a for this case

$$T_a(t) = \begin{cases} T_{a_1} & t < t_{\text{lag}} \\ T_{a_2} & t \geq t_{\text{lag}} \end{cases} \tag{3.5}$$

Here, T_{a_1} is the ambient temperature before $t = 0$, that is, before we enter the warm room. T_{a_2} is the ambient temperature in the warm room. Since we assume that the temperature sensor always sees the temperature at time $t - t_{\text{lag}}$ instead of the actual temperature at time t, Equation 3.5 describes the ambient temperature as seen by the temperature sensor. In Equation 3.1, $T_a(t)$ corresponds to the body temperature, T_b. This means that for $t < t_{\text{lag}}$ we have

$$T_1(t) = T_{a_1} - (T_{a_1} - T_0) \cdot e^{-r \cdot t} \tag{3.6}$$

and, for $t \geq t_{\text{lag}}$

$$T_2(t) = T_{a_2} - (T_{a_2} - T_1(t_{\text{lag}})) \cdot e^{-r \cdot (t - t_{\text{lag}})} \tag{3.7}$$

The parameters in the last two equations have the same interpretations as above in Equation 3.1. Note that $T_1(t_{\text{lag}})$ has been used as the initial temperature in Equation 3.7 since we are shifting from T_1 to T_2 at time t_{lag}, and $T_1(t_{\text{lag}})$ is the actual temperature at that time. Note also that $t - t_{\text{lag}}$ appears in the exponent of

Equation 3.7 to make sure that we have $T_2(t_{\text{lag}}) = T_1(t_{\text{lag}})$. The overall model can now be written as

$$T(t) = \begin{cases} T_1(t) & t < t_{\text{lag}} \\ T_2(t) & t \geq t_{\text{lag}} \end{cases} \tag{3.8}$$

3.2.3.4 **Comparing Model and Data**

Some of the parameters of the last equations can be estimated based on Figure 3.2b and the corresponding data in `Room.csv`:

$$T_0 \approx 18.5 \tag{3.9}$$

$$t_{\text{lag}} \approx 2.5 \tag{3.10}$$

$$T_{a_2} \approx 21 \tag{3.11}$$

In principle, the remaining parameters (T_{a_1} and r) can now either be determined by heuristic arguments as was done above in the context of Equation 3.1, or by nonlinear regression methods as described in Section 2.4. However, before this is done, it is usually efficient to see if reasonable results can be achieved using hand-tuned parameters. In this case, a hand tuning of the remaining parameters T_{a_1} and r shows that no satisfactory matching between Equation 3.8 and the data can be achieved, and thus any further effort (e.g. nonlinear regression) would be wasted. Figure 3.4 shows a comparison of Equation 3.8 with the data of Figure 3.2b based on the hand-fitted values $T_{a_1} = 16.7$ and $r = 0.09$. The figure has been produced using the *Maxima* code `RoomExp.mac` and the data `room.csv` in the book software. Looking at `RoomExp.mac`, you will note that the `if...then` command is used to implement Equation 3.8 (see *Maxima*'s help pages for more information on this and other "conditional execution" commands).

`RoomExp.mac` computes a coefficient of determination $R^2 = 92.7\%$, reflecting the fact that data and model are relatively close together. Nevertheless, the result

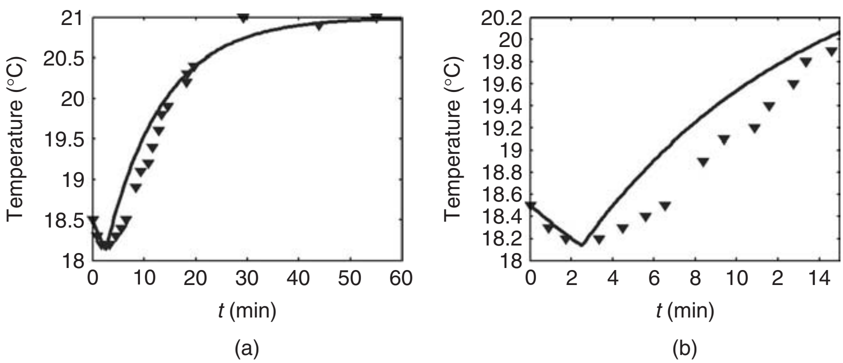


Fig. 3.4 (a) Comparison of Equation 3.8 (line) with the data of Figure 3.2b (triangles) using $T_{a_1} = 16.7$ and $r = 0.09$. (b) Same picture on a different scale, showing a dissimilarity between model and data.

is unsatisfactory since the qualitative pattern of the model curve derived from Equation 3.8 differs from the data. This is best seen if model and data are plotted for $t < 14$ as shown in Figure 3.4b. As can be seen, there is a sharp corner in the model curve, which is not present in the data. Also, the model curve is bent upward for $t > 3$ while the data points are bent downward there. Although the coefficient of determination is relatively high and although we might hence be able to compute reasonable temperature predictions based on this model, the qualitative dissimilarity of the model and the data indicates that the modeling approach based on Equation 3.8 is wrong. As it was mentioned in Section 1.2.2, the qualitative coincidence of a model with its data is an important criterion in the validation of models.

3.2.3.5 **Validation Fails – What Now?**

We have to reject our first idea of how temperature memory could be included into the model. Admittedly, it was a very simple idea to assume that the sensor sees ''old'' temperatures $T_a(t - t_{\text{lag}})$ shifted by a constant time t_{lag}. One of the reasons why this idea was worked out here is the simple fact that it led us to a nice example of a model rejected due to its qualitative dissimilarity with the data. Equation 3.8 also is a nice example showing that one cannot always distinguish in a strict sense between phenomenological and mechanistic models. On the one hand, it is based on the phenomenological model of temperature adaption, Equation 3.1. On the other hand, Equation 3.1 has been used here in a modified form based on our mechanistic considerations regarding the temperature memory of the system. As was already mentioned in Section 1.5, models of this kind lying somewhere between phenomenological and mechanistic models are also called *semiempirical* or *gray-box* models.

Before going on, let us spend a few thoughts on what we did so far in terms of the modeling and simulation scheme (Note 1.2.3). Basically, our systems analysis above led us to the conclusion that our model needs some kind of temperature memory. Equation 3.8 corresponds to the modeling step of Note 1.2.3, the simulation and validation steps correspond to Figure 3.4. After the validation of the model failed, we are now *back in the systems analysis step*. Principally, we could go now into a more detailed study of the internal mechanics of the temperature sensor. We could, for example, read technical descriptions of the sensor, hoping that this might lead us on the right path. But this would probably require a considerable effort and might result into unnecessarily sophisticated models. Before going into a more detailed modeling of the sensor, it is better to ask if there are other, simple hypotheses that could be used to explain the temperature memory of the system.

Remember that Note 1.2.2 says that the simplest model explaining the data is the best model. If we find such a simple hypothesis explaining the data, then it is the best model of our data. This holds true even if we do not know that this hypothesis is wrong, and even if the data could be correctly explained *only* based on the temperature sensor's internal mechanics. If both the models – the wrong model based on the simple hypothesis and a more complex model based on the temperature sensor's internal mechanics – explain the data equally and indistinguishably well, then we should, for all practical purposes, choose the simple model – at least based on the data, in the absence of any other indications that it is a wrong model.

3.2.3.6 A Different Way to Explain the Temperature Memory

Fortunately, it is easy to find another, simple hypothesis explaining the temperature memory of the system. In contrast to our first hypothesis above (''temperature sensor sees old temperatures''), let us now assume that the qualitative difference between the data in the body temperature and alarm clock examples (Figures 3.1a and 3.2b) is *not* a consequence of differences in the internal mechanics of the temperature sensors, but let us assume that the temperature sensors used in both the examples work largely the same way. If this is true, then the difference between the data in the body temperature and alarm clock examples must be related to differences in the construction of the clinical thermometer and the alarm clock as a whole, namely to differences in their construction related to temperature measurements. There is indeed an obvious difference of this kind: when the clinical thermometer is used, the temperature sensor is in direct contact with the body temperature that is to be measured. In the alarm clock, on the other hand, the temperature sensor sits somewhere inside, not in direct contact with the ambient temperature that is to be measured. This leads to the following.

Hypothesis:
The temperature memory of the alarm clock is physically realized in terms of the temperature of its immediate surroundings inside the alarm clock, for example, as the air temperature inside the alarm clock or as the temperature of internal parts of the alarm clock immediately adjacent to the temperature sensor.

To formulate this idea in mathematical terms, we need one or more state variable(s) expressing internal air temperature or the temperatures of internal parts immediately adjacent to the temperature sensor. Now it was emphasized several times that we should start with the simplest approaches. The simplest thing that one can do here is to use an *effective internal temperature* T_i, which can be thought of as some combination of internal air temperature and the temperature of relevant internal parts of the alarm clock. A more detailed specification of T_i would require a detailed investigation of the alarm clock's internal construction, and of the way in which internal air and internal parts' temperatures affect the temperature sensor. This investigation would be expensive in terms of time and resources, and it would hardly improve the results, which is achieved below based on T_i as a largely unspecified ''black box quantity''.

Note 3.2.4 (Effective quantities) Effective quantities expressing the cumulative effects of several processes (such as T_i) are often used to achieve simple model formulations.

Introducing T_i as a new state variable in Model A, we obtain *Model B* as an improved conceptual model of the alarm clock (Figure 3.3). As a next step, we now need mathematical statements (the M of the mathematical model (S, Q, M) to be developed) that can be used to compute the state variables. Since ODEs are required here, we will go on with this example at the appropriate place below (Section 3.4.2). It will turn out that Model B explains the data in Figure 3.2b very well.

3.2.3.7 **Limitations of the Model**

Remember that as a mechanistic modeler you are a ''systems archaeologist'', uncovering the internal system mechanics from data similar to an archaeologist who derives subsurface structures from ground-penetrating radar data. Precisely in this ''archaeological'' way, Figure 3.3 was derived from the data in Figure 3.2b by our considerations above. Our starting point was given by the data in Figure 3.2b with the puzzling initial decrease in the temperatures, an effect that obviously ''wants to tell us something'' about internal system mechanics. Model B now represents a hypothesis of what the data in Figure 3.2b might tell us about the internal mechanics of the alarm clock during temperature measurement.

Note that Model B is a really brutal simplification of the alarm clock as a real system (Figure 3.2a), similar to our brutal simplification of the system ''car'' in Section 1.1. In terms of Model B, nothing remains of the initial complexity of the alarm clock except for the three temperatures T_s, T_i, and T_a. It must be emphasized that Model B represents an *hypothesis* about what is going on inside the alarm clock during temperature measurement. It may be a right or wrong hypothesis. The only thing we can say with certainty is that Model B probably represents the simplest *hypothesis* explaining the data in Figure 3.2b. Model B may fail to explain more sophisticated temperature data produced with the alarm clock, and such more sophisticated data might force us to go into a more detailed consideration of the alarm clock's internals; for example, into a detailed investigation of the temperature sensor, or into a more detailed modeling of the alarm clock's internal temperatures, which Model B summarizes into one single quantity T_i.

As long as we are concerned with the data only in Figure 3.2b, we can be content with the rather rough and unsharp picture of the alarm clock's internal mechanics provided by Model B. The uncertainty remaining when mechanistic models such as Model B are developed from data is a principal problem that cannot be avoided. A mechanistic model represents a hypothesis about the internal mechanics of a system, and it is well known that it is, as a matter of principle, impossible to prove a scientific hypothesis based on data [95]. Data can be used to show that a model is wrong, but they can never be used to prove its validity. From a practical point of view, this is not a problem since we can be content with a model as long as it explains the available data and can be used to solve our problems.