

## Chapter 14

# Heat Transfer

The molecular dynamics simulations discussed in Chap. 13 required the use of *ordinary differential equations (ODE)* when describing particle trajectories. For these, only one independent variable, in this case the time, was needed. But there also exist a very large range of physical problem settings in which the modeling process is assisted by *partial differential equations (PDE)* in a way that is somewhere in the area between obvious, suitable or necessary. An example is *structural mechanics*, which among other things considers the deformation of structures under the influence of forces. Such analyses are relevant in entirely different scenarios—from the construction of bridges to the construction of micro-electromechanical sensors and actuators (MEMS). Another important example is *fluid dynamics* which we will be concerned with in the next chapter. From the perspective of the simulation, the problems which can be modeled with PDEs are interesting and challenging, in particular since the most modern methods and computers are typically indispensable for their efficient numerical solution.

But first things first: A problem setting that is relatively easy to derive is described by the *heat equation* and is be considered in this chapter as a prototype of a phenomena that is based on *balancing processes*. In *thermodynamics* as well as in many other application areas it is important to be able to make statements pertaining to the propagation of heat. Frequently, the objective is to either divert heat as quickly as possible (e.g., at air conditioning systems and cooling devices) or to deliver it as quickly and loss-free as possible (e.g., at cooktops). However, other applications can be concerned with analyzing the distribution of heat in a body in order to recognize places that, for example, are threatened with the possibility of overheating.

The following will first briefly introduce the physical basics of heat transfer. The choice of dimension is also considered since ultimately the complexity of the simulation depends very much on the number of independent variables. In this case, we only consider the temperature, therefore the complexity depends on the number of dimensions considered in the model (space and time). From here, we naturally consider the simulation of the (steady) heat equation. This includes on the one hand

the discretization while on the other the solution of the resulting linear system of equations.

Given the tools discussed in Chap. 2 this requires not only analysis but the comprehensive parts from the section on numerical analysis. For obvious reasons, our treatment is once again relatively terse. A more comprehensive reference for the physical description as well as for the discretization is found in [57].

## 14.1 Derivation of the Heat Equation

The derivation of partial differential equations for the description of physical processes can be very complicated. Before modeling, one should first restrict the application area of interest as much as possible: The more scenarios and effects that are to be included into the model, the more complex the model obviously becomes. The point of departure are the basic physical laws, often in form of *conservation laws*. In this chapter, these are considerations concerning the amount and transfer of heat. Typically, such laws include differential operators (as a simple example from classical mechanics one can think of the relationship between the distance traveled and velocity or acceleration) in which the mathematical treatment leads to systems of differential equations that often are connected with algebraic equations that, e.g., describe constraints (maximal displacement, etc.).

For the diffusion of heat in an object, hereforth called the *body*, the desired quantity is the *temperature*  $T(\mathbf{x}; t)$  that is dependent upon the location  $\mathbf{x}$  and the time  $t$ . The time and space coordinates must be treated separately in the following derivation: For the operators such as  $\nabla$  (*gradient*),  $\operatorname{div}$  (*divergence*) and  $\Delta = \operatorname{div} \cdot \nabla$  (*Laplace operator*), the partial derivatives only apply to the spatial coordinates.

For the derivation of the heat equation we consider the amount of heat stored in the respective body from which we will obtain a conservation equation. We assume that this amount of heat per unit volume is proportional to the temperature whereby the proportionality factor is the product of the *density*  $\varrho$  of the body and the *specific heat capacity*  $c$  (a material property of the substance which describes how much energy is required in order to increase the temperature by a certain amount in a certain amount of substance). One then obtains the amount of heat  $Q$  that is stored per unit volume  $V$  (reference volume) of the body by integration over the reference volume:

$$Q = \int_V c\varrho T(\mathbf{x}; t) \, d\mathbf{x}. \quad (14.1)$$

In the following we assume that the body is *homogeneous* so that  $\varrho$  and  $c$  are (positive real) constants; hence, they will not play any particular role in the following.

The temperature distribution—and thus the amount of heat—changes over time. We thus derive an equation that describes the heat transfer across the surface

$\partial V$  of our reference volume. The heat transport is driven through the balancing processes (here due to temperature differences within the body) that we assume to be proportional to the *normal derivative*  $\frac{\partial T}{\partial n}$  at the surface, i.e., the scalar product of the gradient  $\nabla T$  and the outward directed normal vector  $\mathbf{n}$  (a positive normal derivative therefore describes the case of heat transfer into the volume  $V$ ). Here, the proportionality factor is provided by the so-called *thermal conductivity*  $k > 0$  (in a substance that conducts heat well, e.g., in a metal,  $k$  is large, whereas  $k$  is only slightly greater than zero for a substance that insulates heat). We assume the thermal conductivity to be constant within the entire body and in addition *isotropic*, i.e., it is not dependant on direction. We now obtain the inflow of heat into the volume  $V$  across its surface and therefore the temporal change in the amount of heat (here, other impacts such as chemical reactions within  $V$ , for example, are excluded) as a *surface integral* over the temperature gradient ( $k$  is only a constant), which can be transformed into a *volume integral* through use of the *divergence theorem*:

$$\frac{dQ}{dt} = \int_{\partial V} k \nabla T(\mathbf{x}; t) \cdot d\vec{S} = \int_V k \Delta T(\mathbf{x}; t) d\mathbf{x}. \quad (14.2)$$

Differentiation of (14.1) with respect to  $t$  and substitution into (14.2) yields

$$\int_V c\varrho \frac{\partial T}{\partial t}(\mathbf{x}; t) d\mathbf{x} = \int_V k \Delta T(\mathbf{x}; t) d\mathbf{x}.$$

Since this equation is satisfied for every arbitrary reference volume  $V$ , the two integrands must coincide:

$$c\varrho \frac{\partial T(\mathbf{x}; t)}{\partial t} = k \Delta T(\mathbf{x}; t).$$

With  $\kappa = k/(c\varrho)$  one immediately obtains the *heat equation*

$$\frac{\partial T(\mathbf{x}; t)}{\partial t} = \kappa \Delta T(\mathbf{x}; t), \quad (14.3)$$

that, for the case of three spatial coordinates  $x$ ,  $y$  and  $z$ , can be written explicitly as

$$\frac{\partial T(x, y, z; t)}{\partial t} = \kappa \left( \frac{\partial^2 T(x, y, z; t)}{\partial x^2} + \frac{\partial^2 T(x, y, z; t)}{\partial y^2} + \frac{\partial^2 T(x, y, z; t)}{\partial z^2} \right).$$

This is a PDE of *parabolic type* (see Sect. 2.4.6), for which *boundary-initial value problems* are typical: In this case, the temperature distribution  $T(\mathbf{x}; t_0)$  is given at the start  $t_0$  of the considered time interval as well as conditions for the surface of the overall body for all  $t > t_0$  (e. g., *Dirichlet boundary conditions* that specify the temperature, or *Neumann boundary conditions* that specify the heat flow). The differential equation itself states how the temperature distribution changes under these conditions.

Without restricting boundary conditions solutions for this equation can sometimes be determined without the need for numerical calculations, i.e., analytically or “with paper and pencil”. For example, one easily verifies that for arbitrary  $\nu_x, \nu_y, \nu_z \in \mathbb{N}$ , the functions

$$T(x, y, z; t) = e^{-\kappa(\nu_x^2 + \nu_y^2 + \nu_z^2)t} \sin(\nu_x x) \sin(\nu_y y) \sin(\nu_z z)$$

satisfy the heat equation.

The previous representation of the heat equation depends on three spatial variables and one temporal variable. In the following, however, we will only consider one or two spatial dimensions. The procedure does not differ from the three-dimensional case in principle. However, the two-dimensional case is much easier to conceive.

In addition, one is often not interested in the temporal development of the temperature distribution but only in a distribution reached in the limit *steady state* as  $t \rightarrow \infty$ . Such a distribution is characterized (in analogy to the equilibrium points for ODEs) by the fact that no more temporal change takes place. It therefore holds that  $T_t = 0$  so that the solution satisfies the *steady heat equation*

$$\kappa \Delta T(\mathbf{x}) = 0 . \quad (14.4)$$

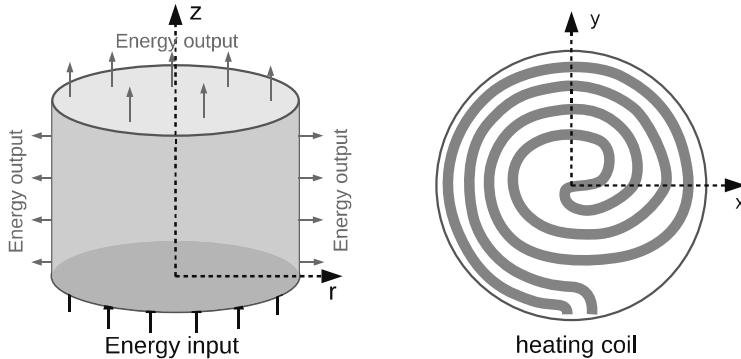
This is a differential equation of *elliptic type*; there no longer are initial values but only boundary conditions. One therefore solves a classical *boundary value problem*. In order to keep the effort of the subsequent discretization low, we will restrict ourselves here to the steady state problem.

### 14.1.1 Number of Dimensions

The number of spatial dimensions that have to be considered in a model and its subsequent simulation depends essentially on four things: on the shape of the object, on the boundary conditions, on the desired accuracy, and on the available computing capacities. Even though all items of daily use are three-dimensional objects, sometimes a simplified representation is entirely sufficient. The heat diffusion in a rod can be represented with only one spatial dimension while a piece of sheet metal requires two. For this, it is important that the sheet is insulated relatively well on the top and bottom surfaces since otherwise heat is exchanged in this third, not modeled dimension.

The required accuracy cannot be determined through the simulated object alone. As an example we consider the simulation of the heat diffusion in a cooking pot as it is illustrated schematically in Fig. 14.1

We make the simplifying assumption that the heat diffusion in the pot behaves according to the formula derived above. The main heat input comes from below via



**Fig. 14.1** Schematic representation of a cooking pot (*left*) and cooktop (*right*)

the cooktop and diffuses in the  $z$ -direction. For the amateur cook who needs only a rough idea of the temperature at a given height, it suffices to consider this dimension alone. Admittedly, the pot loses heat through the exterior sides. For the manufacturer of the pot it would make sense to consider this in the modeling regardless. Thus, the radius is used as an additional coordinate. As illustrated schematically on the right side in Fig. 14.1, the heat distribution is also inhomogeneous on the cooktop itself. For a manufacturer of cooktops it would be important to take such effects into consideration in the modeling of heat diffusion in a pot as well via the angle as a third spatial coordinate.

## 14.2 Discretization

In this section we assume that the model equations are given so that their discretization can now be discussed. Here, we will use *finite differences* which have already appeared in several places of this book. We will review the discretization since we are also interested in the efficient solution of the resulting linear systems of equations.

In the derivation of the heat equation we considered the case involving the *steady state* (14.4). The solution of this steady state equation requires numerical techniques different from those for the solution of the transient heat equation. In the *transient* case, both the space as well as the time must be discretized whereas in the steady state case only the spatial coordinates are discretized. For example, if one wants to analyse the distribution of heat in a material that is connected to a heat source on one side, one may consider this either as a steady state problem or as a transient problem. One could simulate the material for a certain period of time in order to determine the heat distribution at the end of this time period. In this case, the time has to be discretized and the problem becomes transient. Since one is not interested in the development over time, but rather in the final result, one may instead solve a

steady state problem in which one is looking for the heat distribution which satisfies the heat equation subject to the given initial conditions.

These two approaches impose quite different requirements to the respective numerical methods. Without numerics we cannot proceed in either case since the equations, together with their boundary and initial conditions, hardly ever have analytic solutions. We restrict our attention in the following to the steady state heat equation. In Sect. 2.4.6, we have introduced several discretization techniques for PDEs and for our problem at hand we select the method of finite differences. We briefly review the discretization through the construction of both the *3-point-stencil* (one spatial dimension) and *5-point-stencil* (two spatial dimensions), resp.

### 14.2.1 3-Point-Stencil

The underlying principle of finite differences is based on the use of difference quotients to approximate derivatives. As an example, for the first derivative one can approximate a tangent line using a secant line as illustrated in Fig. 14.2.

When approximating first derivatives, one distinguishes between *forward difference quotients*,

$$T'(x_0) \doteq \frac{T(x_0 + h) - T(x_0)}{h}, \quad (14.5)$$

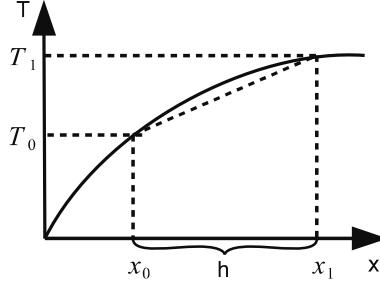
in which the current and following grid points are used, and the *backward difference quotients*,

$$T'(x_0) \doteq \frac{T(x_0) - T(x_0 - h)}{h}, \quad (14.6)$$

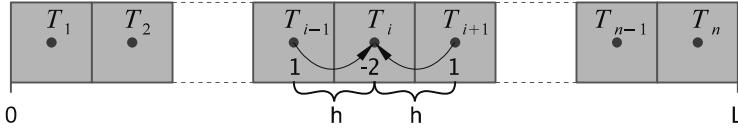
in which the current and previous grid points are used instead. In order to discretize the second derivatives appearing in the heat equation, one can once more construct difference quotients from (14.5) and (14.6). This procedure leads to

$$\begin{aligned} T''(x_0) &\doteq \frac{\frac{T(x_0+h)-T(x_0)}{h} - \frac{T(x_0)-T(x_0-h)}{h}}{h} \\ &= \frac{T(x_0 + h) - 2T(x_0) + T(x_0 - h)}{h^2}. \end{aligned} \quad (14.7)$$

To perform numerical computations we need to determine a suitable value for the *grid width*  $h$ . On the one hand, this value will depend on the required computational accuracy (it is hoped that the closer the grid points are to each other, the higher the accuracy—at least asymptotically). On the other hand, however, we have to keep in mind the computing resources available. In the case of the one-dimensional



**Fig. 14.2** Approximation of the derivative by the secant



**Fig. 14.3** Discretization of a bar

heat equation (for example, in the computation of the heat distribution in a rod), one needs to decide the number of discrete *grid points* at which to determine the temperature. The greater this number, the higher the expected accuracy as well as the computational effort. Figure 14.3 shows a section of a rod that has been divided into  $n$  intervals of equal length. In this example, we approximate the temperature,  $T_i$ ,  $i = 1, \dots, n$ , at the midpoints of the intervals. These midpoints therefore have a distance  $h = \frac{L}{n}$  to their neighboring midpoints.

We can now apply the discretization to the one-dimensional heat equation. For a given gridpoint  $i$ , using (14.4) in (14.7) implies

$$\kappa \cdot \frac{T_{i+1} - 2T_i + T_{i-1}}{h^2} = 0,$$

which simplifies to

$$T_{i+1} - 2T_i + T_{i-1} = 0.$$

The equation for gridpoint  $i$  requires the temperature  $T_i$  at point  $i$  as well as the values  $T_{i-1}$  and  $T_{i+1}$  at the neighboring grid points. Such a discretization of the second derivative is called a *3-point-stencil* since three neighboring points are involved in the computation at point  $i$ . The discrete equation has to be satisfied over the entire (discretized) domain, in this case for all  $i$  from 1 to  $n$ . However, the two boundary points do not have any exterior neighbors. We will treat the boundary conditions in more detail in Sect. 14.2.3. Here, we only note that we will have fixed,

known values  $T_0$  and  $T_{n+1}$ , at the boundary. Therefore, we obtain the following *linear system of equations* with  $n$  equations in  $n$  unknowns  $T_1, \dots, T_n$ :

$$\begin{pmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & & \vdots \\ 0 & 1 & -2 & \ddots & 0 \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \dots & 0 & 1 & -2 \end{pmatrix} \cdot \begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_{n-1} \\ T_n \end{pmatrix} = \begin{pmatrix} -T_0 \\ 0 \\ \vdots \\ 0 \\ -T_{n+1} \end{pmatrix}.$$

### 14.2.2 5-Point-Stencil

In view of the Laplace operator, the solution of (14.4) requires the two second order partial derivatives  $\frac{\partial^2 T}{\partial x^2}$  and  $\frac{\partial^2 T}{\partial y^2}$ . Analogous to (14.7), we approximate these derivatives at  $(x, y)$  using a uniform grid width  $h$  for both the  $x$ - and  $y$ -directions by,

$$\begin{aligned} \frac{\partial^2 T}{\partial x^2} &\doteq \frac{T(x+h, y) - 2T(x, y) + T(x-h, y)}{h^2}, \\ \frac{\partial^2 T}{\partial y^2} &\doteq \frac{T(x, y+h) - 2T(x, y) + T(x, y-h)}{h^2}. \end{aligned}$$

This in turn yields the approximation

$$\kappa \left( \frac{T(x+h, y) + T(x, y+h) - 4T(x, y) + T(x-h, y) + T(x, y-h)}{h^2} \right) = 0$$

of (14.4), and after simplification yields

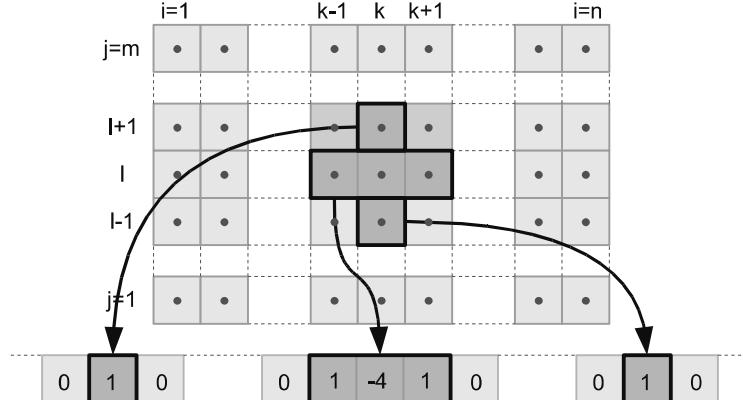
$$T(x+h, y) + T(x, y+h) - 4T(x, y) + T(x-h, y) + T(x, y-h) = 0.$$

Analogous to the one-dimensional case, we can subdivide the two-dimensional simulation domain into a grid of discrete cells and denote the temperature at the midpoint of cell  $(i, j)$  by  $T_{i,j}$ . In doing so we obtain the equation,

$$T_{i+1,j} + T_{i,j+1} - 4T_{i,j} + T_{i-1,j} + T_{i,j-1} = 0.$$

This discretisation scheme is called a *5-point-stencil*.

In the one-dimensional case, the left and right neighbors were needed for the computation of a cell. The individual temperature values were ordered in a vector which is important for setting up the system of equations as well as storing the values in the implementation. In order to set up a linear system of equations in the



**Fig. 14.4** Access to the required elements of the *5-point-stencil*

two-dimensional case, we need to determine an ordering in which the temperature values will be ordered in a vector. We choose a row-wise ordering: Neighbors which lie in the same row, defined in a geometric sense of the two-dimensionally ordered cells, will also be neighbors in the vector. Those neighbors which lie in the same column are separated from each other in the vector. Whereas this procedure appears straightforward and enjoys great popularity, it turns out to hinder an efficient implementation—at least if we would like to or even have to exploit the full capacity of the available computer hardware. (The so-called *spatial locality* is important for *cache-hierarchies* of modern processors: Items that are spatially close to each other should be treated promptly to ensure the availability of required quantities in the cache and to avoid expensive accesses to the main memory). Figure 14.4 illustrates the *5-point-stencil* for the computation of the element  $(k, l)$  as well as the ordering of the respective cells in the vector.

With these considerations, we now assemble a system of equations for the two-dimensional case. For elements at the boundary we proceed in a similar manner as in the one-dimensional case. For a very small test domain with only three cells per direction (without boundary), one obtains the following system of equations:

$$\left( \begin{array}{ccc|cc} -4 & 1 & & 1 & \\ 1 & -4 & 1 & & 1 \\ & 1 & -4 & & 1 \\ \hline 1 & & & -4 & 1 \\ & 1 & & 1 & -4 & 1 \\ \hline 1 & & & 1 & -4 & 1 \\ & 1 & & 1 & 1 & -4 \\ \hline & 1 & & 1 & 1 & -4 \\ & & 1 & & 1 & -4 \\ & & & 1 & & 1 \end{array} \right) \cdot \begin{pmatrix} T_{1,1} \\ T_{1,2} \\ T_{1,3} \\ T_{2,1} \\ T_{2,2} \\ T_{2,3} \\ T_{3,1} \\ T_{3,2} \\ T_{3,3} \end{pmatrix} = \begin{pmatrix} -T_{0,1} - T_{1,0} \\ -T_{0,2} \\ -T_{0,3} - T_{1,4} \\ -T_{2,0} \\ 0 \\ -T_{2,4} \\ -T_{3,0} - T_{4,1} \\ -T_{4,2} \\ -T_{4,0} - T_{3,4} \end{pmatrix}. \quad (14.8)$$

The matrix for this system is divided into nine blocks. The number of blocks always equals the square of the number of rows in the discretized domain, and the number of entries (including zero entries) per block equals the square of the number of columns. Solving the linear system of equations yields the (numerical, or discrete) solution of the two-dimensional steady-state heat equation.

### 14.2.3 Boundary Treatment

In Chap. 2, we have learned that for PDEs there exist different possibilities to define conditions at the boundary of the domain in order to guarantee the existence of a unique solution to the problem. The intuitive choices in the previous two sections were *Dirichlet boundary conditions*. Here, the temperature values are given at all boundary cells but we still need to consider a few additional details. For example, one has to determine what constitutes the boundary. In our case, we divided the domain into cells and computed temperatures at the cell midpoints (see Fig. 14.4). This opens up two possibilities: Either one fixes the temperature for the outermost layer of cells within the domain, or one adds another layer of cells around the domain in which one fixes the temperature. Here, we will pursue the second option, resulting in indices 0 and  $n + 1$ . Instead of computing the temperatures at the cell midpoints, which appears reasonable for illustrative purposes, one could also have chosen the vertices of the cells (or intersections of gridlines, resp.). Using vertices then implies that the outermost values would lie directly on the boundary of the domain which would subsequently simplify the boundary treatment. This would also be more meaningful in the sense that a given temperature usually holds directly at the boundary and not half a cell width away from it.

These considerations illustrate the amount of work which is required for the transition from the assignment of Dirichlet boundary conditions in the model to a numerical solution technique and its implementation. The question of boundary conditions will again reappear in Chap. 15, Computational Fluid Dynamics.

## 14.3 Numerical Solution of the PDE

The discretization for either the one- as well as the two-dimensional case produces a large *linear system of equations* with a corresponding *sparse* coefficient matrix—in the  $k$ 'th row, there can be nonzero entries only in positions that are connected to the grid point of this row via the discretization stencil, and there are at most three or five such positions, resp. In principle, this structure remains the same for three dimensions; there will be two additional off-diagonal entries since now the local grid point also has direct neighbors in the  $z$ -direction. The special structure of these systems of equations suggests the use of *iterative methods* for their solution as introduced in Sect. 2.4.4. Direct solvers such as Gaussian elimination are usually