

## Chapter 2

# Thermodynamics

Thermodynamics includes a *theoretical* and an *applied* part. The applied thermodynamics has its roots at the end of 19th century and it is used to calculate the temperature distribution in a continuum body. This aim is fulfilled by employing the balance of internal energy. We will study this approach in Sect. 2.1 for macroscopic systems and in Sect. 2.2 for microscopic systems. The difference between macroscopic and microscopic systems relies on the used constitutive equation.

The theoretical thermodynamics has started around 1950s. It has the goal of defining constitutive (material) equations that close the balance equations. By using thermodynamics we will derive the constitutive equations necessary in the computational reality. In Chap. 1 we have employed many constitutive equations with an ad-hoc method. In this chapter we will answer the question of how to derive these equations in a thermodynamically consistent manner. Concretely, in Sect. 2.3 we will analyze such an approach and derive the NAVIER–STOKES–FOURIER equations for a viscous fluid. Unfortunately, there are various methods in the literature for the thermodynamically consistent derivation of constitutive equations—we will not discuss the pros and cons of these different approaches. Herein, we present an engineering approach suitable for many simple material models. Although the method fails to cover some processes, it is general enough for determining all of the constitutive equations necessary for the simulated engineering applications in this book. Moreover, the necessary mathematical knowledge is fairly low. After having studied the method in Sect. 2.3 we will employ it in Sect. 2.4 for viscoelastic materials and in Sect. 2.5 for plastic deformations. Much use of the method will be made in the next chapter, too.

## 2.1 Temperature Distribution in Macromechanics

A *conductive* material possesses the ability of transferring thermal energy, *heat*, without mass transport. In other words, heat travels between particles where particles remain at their positions. If we hold one end of a steel spoon in hot water, heat conducts to the other end without any deformation of the spoon itself. Of course there is a small expansion due to the temperature change in the spoon, however, in this section we neglect this reversible deformation and assume the body as *rigid* throughout the simulation. The balance equations can be introduced in a material or open system. The local forms of the equations are identical in both systems.

We motivate the governing equations for a solid body by using a material system. Mass and momentum balances in current frame read

$$\left( \int_{\mathcal{B}} \rho \, dv \right)^{\cdot} = 0, \quad \left( \int_{\mathcal{B}} \rho v_i \, dv \right)^{\cdot} = \int_{\partial\mathcal{B}} \sigma_{ji} n_j \, da + \int_{\mathcal{B}} \rho f_i \, dv. \quad (2.1)$$

From the mass balance we calculate the mass density (or pressure) and from the momentum balance we acquire the displacement (or velocity). For the temperature calculation we will use another balance equation. In order to obtain this equation we start off with the balance of *total* energy in the current frame:

$$\left( \int_{\mathcal{B}} \rho e \, dv \right)^{\cdot} = \int_{\partial\mathcal{B}} F_j n_j \, da + \int_{\mathcal{B}} \rho s \, dv, \quad (2.2)$$

where  $\rho$ ,  $e$ ,  $F_i$ , and  $s$  denote the mass density,<sup>1</sup> the specific<sup>2</sup> total energy, the energy flux, and the specific energy supply, respectively. The total energy is a conserved quantity like mass and momentum; the balance of total energy lacks a production term. We can decompose total energy density:

$$\rho e = \rho u + \frac{1}{2} \rho v^2, \quad (2.3)$$

where the first term is the so-called *internal* energy density with the specific internal energy,  $u$ , and the second term is the *kinetic* energy density due to the velocity,  $v$ . Now, by inserting the mass balance into the balance of energy as well as into the balance of momentum, we obtain

$$\begin{aligned} \int_{\mathcal{B}} \rho v_i^{\cdot} \, dv &= \int_{\partial\mathcal{B}} \sigma_{ji} n_j \, da + \int_{\mathcal{B}} \rho f_i \, dv, \\ \int_{\mathcal{B}} \rho e^{\cdot} \, dv &= \int_{\partial\mathcal{B}} F_j n_j \, da + \int_{\mathcal{B}} \rho s \, dv. \end{aligned} \quad (2.4)$$

---

<sup>1</sup>Density means per volume.

<sup>2</sup>Specific means per mass.

After applying GAUSS's law, we acquire the local forms:

$$\rho v_i \dot{\phantom{x}} - \frac{\partial \sigma_{ji}}{\partial x_j} - \rho f_i = 0, \quad \rho e \dot{\phantom{x}} - \frac{\partial F_j}{\partial x_j} - \rho s = 0. \quad (2.5)$$

We observe a clear structure in the balance equations. The first terms indicate which term is balanced. The second terms are divergence of fluxes. The third terms are supply terms. On the right-hand side the production terms are written—momentum and total energy are conserved quantities. Equation (2.3) implies that the rate of specific energy can be rewritten by making use of the momentum balance:

$$\begin{aligned} e \dot{\phantom{x}} &= \left( u + \frac{1}{2} v_i v_i \right) \dot{\phantom{x}} = u \dot{\phantom{x}} + v_i \dot{\phantom{x}} v_i, \\ \rho e \dot{\phantom{x}} &= \rho u \dot{\phantom{x}} + v_i \left( \frac{\partial \sigma_{ji}}{\partial x_j} + \rho f_i \right). \end{aligned} \quad (2.6)$$

By using the latter in the balance of total energy we obtain

$$\begin{aligned} \rho u \dot{\phantom{x}} + v_i \frac{\partial \sigma_{ji}}{\partial x_j} - \frac{\partial F_j}{\partial x_j} + v_i \rho f_i - \rho s &= 0, \\ \rho u \dot{\phantom{x}} - \frac{\partial}{\partial x_j} (F_j - \sigma_{ji} v_i) - \rho (s - f_i v_i) &= \sigma_{ji} \frac{\partial v_i}{\partial x_j}. \end{aligned} \quad (2.7)$$

This equation has a structure of a balance equation. The first term denotes that the internal energy is balanced. The second term is a divergence of the internal energy flux and the third term is the internal energy supply. These terms are often abbreviated as

$$-q_j = F_j - \sigma_{ji} v_i, \quad r = s - f_i v_i, \quad (2.8)$$

where the so-called *heat flux*,  $q_i$ , and the *supply term*,  $r$ , needs to be defined or given. The minus sign in front of the heat flux is due to the convention that the heat pumped into the system has been seen as a positive quantity. The first power generators were using coal to burn and they did produce mechanical energy. Heat added into the system as well as the mechanical work taken out of the system were seen as *positive* quantities. We keep herein this convention and define the flux term of the internal energy as  $-q_i$ . The balance of internal energy reads

$$\rho u \dot{\phantom{x}} + \frac{\partial q_j}{\partial x_j} - \rho r = \sigma_{ji} \frac{\partial v_i}{\partial x_j}, \quad (2.9)$$

or in global form (after using the balance of mass and GAUSS's law)

$$\left( \int_{\mathcal{B}} \rho u \, dv \right) \dot{\phantom{x}} = - \int_{\partial \mathcal{B}} q_j n_j \, da + \int_{\mathcal{B}} \rho r \, dv + \int_{\mathcal{B}} \sigma_{ji} \frac{\partial v_i}{\partial x_j} \, dv. \quad (2.10)$$

Here we see again the effect of the minus sign in front of the flux term. The heat flux is defined as the rate of energy transported *into* the system across the boundary. Since the plane normal is directed outward, we need a minus sign in order to describe a transport into the system. In other words, the heat fluxes into the body against the direction of the plane normal,  $\mathbf{n}$ . Therefore, a minus sign is necessary to heat the system up, if the heat flux is positive. The second term on the right-hand side is called a radiation term,  $r$ . Actually, the name radiation might be misleading; this is not a thermal radiation, for example, radiation from the sun cannot be modeled with this term. This term is a specific (per mass) heat supply,  $r$ , used in the microwave oven or in a laser welding. We will call the term *internal heating* or heat supply in the following. The last term on the right-hand side denotes a production term. When a deformation occurs, this production term will alter the internal energy. We cannot simply neglect this term. As long as there is a deformation in the continuum body, internal energy will be produced. The internal energy is related to the temperature such that an increasing internal energy will imply a temperature increase. Therefore, for any process where a deformation occurs, there will be a change in temperature. The production term is also called an *internal friction*. In many systems this production term may be small, especially for slow deformations, such that we can assume that the process is isothermal. This justification has been used in the Chap. 1.

In this section we restrict the model for rigid bodies,  $v_i = 0$ , no deformations are allowed. Balances of mass and momentum are satisfied identically and the balance of internal energy simplifies to

$$\rho \dot{u} + \frac{\partial q_j}{\partial x_j} - \rho r = 0 . \quad (2.11)$$

For a rigid body the internal energy depends only on temperature,

$$u = u(T) . \quad (2.12)$$

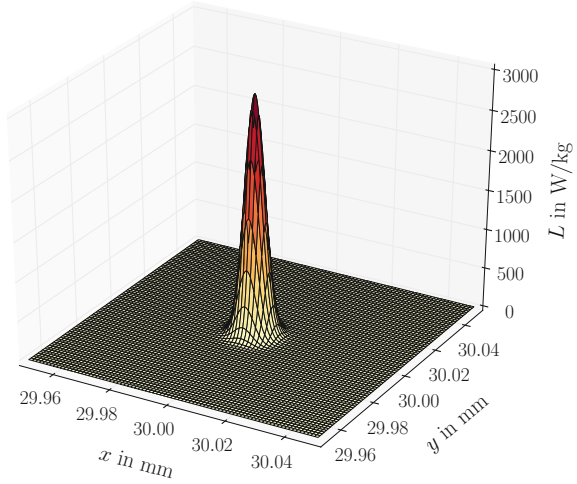
The internal heating or heat supply,  $r$ , increases the temperature by affecting all particles together. This term is a volumetric power; the food in the microwave oven heats up in each of its particles at the same time. In the example of a spoon in hot water, heat is transferred across the surface and then fluxes from one end to the other. If we put the spoon in a microwave oven, all of its particles heat up together due to the supplied heat supply,  $r$ . The same holds in case of a laser beam.<sup>3</sup> A laser beam supplies energy on a focused location. Suppose that a laser beam irradiates on one end of the spoon. All particles on that end are irradiated and they all heat up simultaneously. The laser radiates heat in such a way that it increases the temperature volumetrically. We model a laser welding via specific heat supply term,  $r$ .

We want to implement a laser welding process and choose a plate as geometry: A steel plate is welded with a laser beam. The energy supply comes as a laser beam in

---

<sup>3</sup>A laser (Light Amplification by Stimulated Emission of Radiation) generates a focused beam of photons in the same wavelength (coherent).

**Fig. 2.1** Laser beam distribution as the GAUSSian bell shape in  $xy$ -plane



a concentrated manner and heats up the whole thickness of the plate at once, roughly alike cylinder, but instead a circle cross section, a GAUSSian bell shape<sup>4</sup> should be modeled. Any circular GAUSSian bell shape in  $x_1x_2$ -plane can be expressed as

$$r = P \exp \left( - \left( (x_1 - \hat{x}_1)^2 + (x_2 - \hat{x}_2)^2 \right) \right), \quad (2.13)$$

such that the laser beam reaches its maximum value  $P$  at the position  $(\hat{x}_1, \hat{x}_2)$ , since  $r = P \exp(0) = P$ . The power becomes asymptotically zero for coordinates away from  $(\hat{x}_1, \hat{x}_2)$  owing to the minus sign. We want to simulate a laser beam evolving in time. The power (energy rate) of the laser reads

$$L(x_i, t) = P \exp \left( - 50000 \left( \left( x_1 - \frac{l}{2} \left( 1 + \frac{1}{2} \sin(2\pi\tau) \right) \right)^2 + (x_2 - v_L t)^2 \right) \right), \quad (2.14)$$

where the laser beam moves with a time parameter  $\tau = t/t_{\text{end}} = [0, \dots 1]$  in  $x_1$  sinusoidally and along  $x_2$  linearly with a constant speed  $v_L$ . Of course, this is a model problem. In reality, the path is already defined in design and programmed into a NC (Numeric Controller) laser welding machine. Here we want to implement a complicated path description in order to present how to implement such a process into the code. The power of the laser beam,  $P$ , can be found in the data sheet of the laser welding machine. In order to visualize the implemented laser beam distribution, we plot the distribution for  $\hat{x} = \hat{y} = 30 \text{ mm}$  and  $P = 3000 \text{ W/kg}$  in Fig. 2.1. Since we multiplied  $L$  in Eq. (2.14) with a huge number of 50 000, the laser beam affects only locally, as expected from a focused light.

<sup>4</sup>It is named for Carl Friedrich Gauß.

For a rigid body the deformation gradient equals to the KRONECKER delta such that current and initial frames are equal,  $x_i = X_i$ . The balance of internal energy reads

$$\rho \frac{\partial u}{\partial t} + \frac{\partial q_i}{\partial x_i} - \rho r = 0 . \quad (2.15)$$

For the internal heating we use the laser beam,  $r = L$ . Moreover, the internal energy depends only on the temperature,  $u = u(T)$ , in case of a rigid body. Hence we obtain

$$\rho \frac{\partial u}{\partial T} \frac{\partial T}{\partial t} + \frac{\partial q_i}{\partial x_i} - \rho L = 0 . \quad (2.16)$$

In Sect. 2.3 on p. 126 we will formally introduce and discuss the so-called *specific heat capacity*:

$$c = \frac{\partial u}{\partial T} , \quad (2.17)$$

and explain how to measure this material parameter. The specific heat capacity is constant for many engineering materials. We also need a constitutive (material) equation for the heat flux,  $q_i$ . The simplest relation is given by FOURIER's law:<sup>5</sup>

$$q_i = -\kappa \frac{\partial T}{\partial x_i} , \quad (2.18)$$

where the material parameter,  $\kappa$ , is referred to as *thermal conductivity*. We assume it being constant. The minus sign denotes that the heat flux conducts in the direction of decreasing temperature gradient. Hence, heat fluxes from higher to lower temperature. This phenomenon is known intuitively and its validity is a subject of theoretical thermodynamics. For the moment we take it for granted and insert FOURIER's law into the balance of internal energy

$$\rho c \frac{\partial T}{\partial t} - \kappa \frac{\partial^2 T}{\partial x_i \partial x_i} - \rho L = 0 . \quad (2.19)$$

This differential equation is called the *field* equation for  $T$ , where its solution leads to the temperature distribution in a rigid body. We acquire the variational form of the latter differential equation by discretizing in time and by using integration by parts. First we utilize a backward difference scheme for temporal discretization, as usual,

$$\frac{\partial T}{\partial t} = \frac{T(t) - T(t - \Delta t)}{\Delta t} = \frac{T - T^0}{\Delta t} . \quad (2.20)$$

---

<sup>5</sup>The constitutive equation is named after Jean-Baptiste Joseph Fourier.

Secondly, we multiply the field equation by the test function,  $\delta T$ , for spatial discretization

$$\int_{\mathcal{B}} \left( \frac{\rho c}{\Delta t} (T - T^0) - \kappa \frac{\partial^2 T}{\partial x_i \partial x_i} - \rho L \right) \delta T \, dv = 0. \quad (2.21)$$

The second term in the integral form possesses a second derivative in space, whereas the multiplied test function has no derivatives. Thus, the continuity conditions of  $T$  and  $\delta T$  are different. In the GALERKIN procedure we utilize the same function space for the primitive variable  $T$  and its test function  $\delta T$ . Hence it has to belong to class  $C^2$  at least. We can integrate by parts and acquire a form where both terms are differentiated once such that the continuity condition of  $T$ ,  $\delta T$  is weakened and they need to belong to  $C^1$ . After integrating by parts we acquire the weak form:

$$F_T = \int_{\mathcal{B}} \left( \frac{\rho c}{\Delta t} (T - T^0) \delta T + \kappa \frac{\partial T}{\partial x_i} \frac{\partial \delta T}{\partial x_i} - \rho L \delta T \right) dv - \int_{\partial \mathcal{B}} \kappa \frac{\partial T}{\partial x_i} \delta T n_i \, da. \quad (2.22)$$

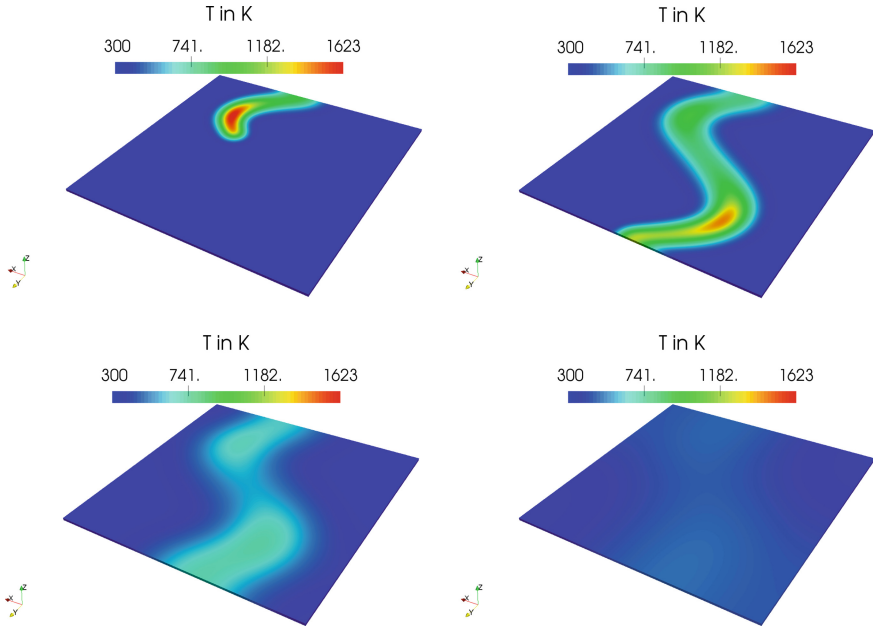
The integrand on the boundaries  $\partial \mathcal{B}$  shall be given. In Chap. 1 we have seen DIRICHLET boundary conditions, where the solution itself is given, and NEUMANN boundary conditions, where the gradient of the solution in the surface outward normal,  $n_i$ , is defined. For the case of the energy equation the heat flux,  $q_i$ , projected into the surface normal,  $q_i n_i$ , defines a NEUMANN condition. This condition is the heat exchange from the surface of the plate to the surrounding medium, probably air or some special kind of gases like Argon or Helium for a better welding. The heat exchange is due to the temperature difference between the surface and the surroundings. Surface becomes hot as a consequence of the energy delivered by the laser beam. The temperature of the surroundings might be set constant<sup>6</sup> to an ambient temperature,  $T_{\text{amb}}$ . Thus, we obtain the DIRICHLET condition  $T = T_{\text{amb}}$  on the boundary. Both conditions can be mixed together in order to create another type of boundary condition. We introduce a mixed boundary condition, to which is referred as a ROBIN boundary condition,<sup>7</sup> for the whole surface  $\partial \mathcal{B}$  as follows

$$q_i n_i = h(T - T_{\text{amb}}) \quad \forall x_i \in \partial \mathcal{B}, \quad (2.23)$$

where a (positive) *convective heat transfer coefficient*,  $h$  in  $\text{W}/(\text{m}^2 \text{K})$ , is introduced that depends on the material and state of the ambient. If the body is embedded in fluid the convection heat transfer coefficient is higher than in air. In case of a moving

<sup>6</sup>Constant temperature of the surroundings is a *warm bath idealization*. Suppose that there is so much water in a bath; a heat exchange with the body within the bath does not affect the temperature of the bath, at all. The water on the surface of the body remains at the same constant temperature all the time.

<sup>7</sup>It is named after Victor Gustave Robin.



**Fig. 2.2** Temperature distribution at 2, 5, 15, and 50 s in the steel plate during the laser welding

fluid that surrounds the body, the coefficient is even higher. This so-called natural convection in Eq. (2.23) provides a positive energy flux into the body if the ambient temperature is higher than the surface temperature,

$$-q_i n_i = h(T_{\text{amb}} - T) > 0 \text{ when } T_{\text{amb}} > T . \quad (2.24)$$

Additionally, for the heat flux we readily employ FOURIER's law in Eq. (2.18). Now the *linear* variational form reads

$$\begin{aligned} F_T = \int_{\mathcal{B}} \left( \frac{\rho c}{\Delta t} (T - T^0) \delta T + \kappa \frac{\partial T}{\partial x_i} \frac{\partial \delta T}{\partial x_i} - \rho L \delta T \right) dv + \\ + \int_{\partial \mathcal{B}} h(T - T_{\text{amb}}) \delta T \, da . \end{aligned} \quad (2.25)$$

The code below computes the temperature distribution transiently, which can be seen in Fig. 2.2.



```

1  """Computational reality 10, temperature distribution in a
    ↳ macroscopic body"""
2  __author__ = "B. Emek Abali"
3  __license__ = "GNU GPL Version 3.0 or later"
4  #This code underlies the GNU General Public License ,
    ↳ http://www.gnu.org/licenses/gpl-3.0.en.html
5
6  from fenics import *
7  import numpy
8  parameters["allow_extrapolation"]=True
9  parameters["form_compiler"]["cpp_optimize"] = True
10 set_log_level(ERROR)
11
12 rho=7860.0      # mass density of steel in kg/m^3
13 c=624.0         # heat capacity in J/(kg K)
14 kappa=30.1     # thermal conductivity in W/(m K)
15 h=18.0         # heat convection out of the surface into ambient
    ↳ in W/(m^2 K)
16 Ta=300.0       # ambient temperature in K
17 l=0.1          # length in x and y directions in m
18 thickness=0.001 # thickness of the plate in m
19 P=3.0e6        # laser power in W/kg
20 speed=0.02     # laser speed in m/s
21 # Create mesh and define function space
22 mesh = BoxMesh(Point(0.0,0.0,0.0), Point(1,1,thickness),
    ↳ 200,200,2)
23 Space = FunctionSpace(mesh, 'P', 1)
24 cells = CellFunction('size_t', mesh)
25 facets = FacetFunction('size_t', mesh)
26 da = Measure('ds', domain=mesh, subdomain_data=facets)
27 dv = Measure('dx', domain=mesh, subdomain_data=cells)
28 t=0.0
29 t_end=50.0
30 Dt=0.1
31 initial_T = Expression("Tini", Tini=Ta)
32 T0 = interpolate(initial_T, Space)
33 Laser = Expression("P*exp(-50000.0*(pow(x[0]-0.5*1*(1+0.5*sin
    ↳ (2*pi*t/t_e)), 2))+pow(x[1]-velo*t, 2))", P=P, t=0, t_e=
    ↳ t_end/10, l=1, velo=speed)
34 T = TrialFunction(Space)
35 del_T = TestFunction(Space)
36 Form = (rho*c/Dt*(T-T0)*del_T \
37        + kappa*T.dx(i)*del_T.dx(i) \
38        - rho*Laser*del_T) * dv \
39        + h*(T-Ta)*del_T*da
40
41 left=lhs(Form)
42 right=rhs(Form)
43
44 A = assemble(left) # non-changing by time stepping
45 b = None          # dynamically assembled acc. to time
46 T = Function(Space)
47 file_T = File("/calcul/CR10/T.pvd")

```

```

48 for t in numpy.arange(0, t_end, Dt):
49     print "Time ", t
50     Laser.t=t
51     b=assemble(right, tensor=b)
52     solve(A, T.vector(), b, 'cg')
53     if t==int(t): file_T << (T, t)
54     T0.assign(T)

```

## To-do

Temperature distribution in a macroscopic rigid body has been computed. In order to deepen the understanding of the implementation try to do the following steps:

- Since the integral form is linear we may have implemented as in the previous sections by using “Form” and “Gain.” Try to implement in this way, the results have to be identical.
- Change the boundary condition to a weak<sup>8</sup> DIRICHLET condition and then to adiabatic<sup>9</sup> boundaries by manipulating the parameter  $h$ .
- Search for approximate values of  $h$  for *natural* and *forced* convection. Which one has been established in the given code?
- Since heat escapes over the boundaries, sooner or later the temperature becomes the ambient temperature homogeneously in all body. Find the material parameters for a polymer and apply the same laser power. Is the body out of steel or polymer will reach the ambient temperature quicker?

## 2.2 Heat Transfer in Micromechanics

Heat propagation in a rigid body has been described by a parabolic differential equation in the last section. A mathematician recognizes the differential equations under two classes: *diffusion* and *wave* problems. A parabolic differential equation models a diffusion problem and a hyperbolic differential equation models a wave propagation. We refrain from using this terminology and point on the famous “paradox” due to the characteristics of the parabolic differential equation used in the heat transfer.<sup>10</sup> Consider FOURIER’s law describing the heat flux:

$$q_i(\mathbf{x}, t) = -\kappa G_i(\mathbf{x}, t), \quad G_i = \frac{\partial T}{\partial x_i}. \quad (2.26)$$

<sup>8</sup>We apply a DIRICHLET condition *strongly* by exchanging the solution with the given solution by using “DirichletBC” in the code. Instead of this method, we can satisfy the condition *weakly* by writing it under the boundary integral.

<sup>9</sup>An adiabatic boundary prevents heat transfer across boundaries.

<sup>10</sup>See [16] for some interesting explanations on the characteristics of the heat propagation.

Flux is the rate of energy exchange, in other words, it describes how quickly the heat energy travels between neighboring particles. Since the flux depends only on the temperature gradient, as long as there exists a temperature gradient the exchange may happen as quick as possible. Therefore, heat flux depending only on the temperature gradient results in an infinite propagation of the information. A typical example is a long bar excited on one end by a laser pulse. As expected, the temperature changes on the excited end. According to Eq. (2.26) at the very moment changed the temperature the neighboring element feels this change. The temperature change implies a heat flux instantaneously, thus whole bar “knows” this temperature change. In other words, the temperature starts changing in the whole bar instantaneously. For a bar in a macroscopic length scale, the temperature change is insignificant. Hence, FOURIER’s law is quite accurate for many engineering problems in macroscale. However, in a microscopic length scale and laser pulses in femtoseconds the accuracy of FOURIER’s law is inappropriate. There are even measurements in these scales suggesting a more sophisticated definition of the heat flux<sup>11</sup> than FOURIER’s law. In this section we will generalize the heat flux by adding a rate dependency, similar to ideas used in Sect. 1.4, and simulate in microscale.

The generalization of the heat flux can be introduced in many different ways. We are interested in applied thermodynamics; so we use an argumentation that the flux and temperature gradient occur in different time instants. It is challenging to choose cause and effect, thus, we basically introduce a time lag both to the heat flux and temperature gradient:

$$q_i(\mathbf{x}, t + \tau_q) = -\kappa G_i(\mathbf{x}, t + \tau_T) . \quad (2.27)$$

The parameters  $\tau_q$  and  $\tau_T$  defines the time-delay or relaxation times between the flux and its response, i.e., the temperature gradient. By setting  $\tau_T = 0$  we attain CATTANEO–VERNOTTE’s heat flux<sup>12</sup> Since we want to evaluate the heat flux in the current time,  $t$ , we expand the left and right sides of Eq. (2.27) around the current time by using a TAYLOR series with linear terms:

$$q_i(\mathbf{x}, t) + \tau_q \dot{q}_i^*(\mathbf{x}, t) = -\kappa G_i(\mathbf{x}, t) - \kappa \tau_T \dot{G}_i^*(\mathbf{x}, t) . \quad (2.28)$$

By suppressing the arguments the heat flux for micromechanics reads

$$q_i = -\tau_q \dot{q}_i^* - \kappa G_i - \kappa \tau_T \dot{G}_i^* . \quad (2.29)$$

---

<sup>11</sup>See, for example, [18].

<sup>12</sup>It is named for Carlo Cattáneo and Pierre Vernotte.

Since the body is rigid, there is no difference between the total and partial time rate. Time discretization delivers

$$\begin{aligned} q_i &= -\tau_q \frac{q_i - q_i^0}{\Delta t} - \kappa G_i - \kappa \tau_T \frac{G_i - G_i^0}{\Delta t}, \\ q_i &= \frac{\Delta t}{\Delta t + \tau_q} \left( \frac{\tau_q}{\Delta t} q_i^0 - \kappa \left( 1 + \frac{\tau_T}{\Delta t} \right) G_i + \frac{\kappa \tau_T}{\Delta t} G_i^0 \right). \end{aligned} \quad (2.30)$$

As we have established the field equation for a rigid body in the last section

$$\rho c \frac{\partial T}{\partial t} + \frac{\partial q_i}{\partial x_i} - \rho L = 0, \quad (2.31)$$

we can now implement the heat flux with relaxation times. After making the first term discrete in time, multiplying by the test function,  $\delta T$ , and then applying an integration by parts, we obtain the weak form:

$$\int_{\mathcal{B}} \left( \frac{\rho c}{\Delta t} (T - T_0) \delta T - q_i \frac{\partial \delta T}{\partial x_i} - \rho L \delta T \right) dv + \int_{\partial \mathcal{B}} \hat{q} \delta T da, \quad (2.32)$$

where  $\hat{q} = q_i n_i$  is the given boundary condition.

Every body consists of electronic particles and thus emits energy as a result of the changes in the electronic configurations of the atoms. This phenomenon happens above the absolute temperature, 0 K, for all times in form of radiation. Radiation is a volumetric supply and every particle of the body emits energy in the form of electromagnetic waves (photons), however, they are again absorbed from the neighboring particles. Only the particles building the surface emits energy such that this type of radiation is modeled as a surface phenomenon.<sup>13</sup> The maximum rate of energy emitted from a rigid body into a vacuum is given by the STEFAN–BOLTZMANN law:<sup>14</sup>

$$\tilde{q} = \sigma T^4, \quad (2.33)$$

where the STEFAN–BOLTZMANN constant,  $\sigma = 5.670 \cdot 10^{-8} \text{ W/m}^2$ , is a universal constant. In the vacuum, the radiation waves propagate with the speed of light,  $c_0 = 2.998 \cdot 10^8 \text{ m/s}$ . In air, they propagate with  $c_{\text{air}} = c_0/n$ , where the *refractive index* is  $n = 1.0008$  for air. The energy rate to the air is

$$\tilde{q} = n^2 \sigma T^4, \quad (2.34)$$

in other words, it is only 0.16% more than into the vacuum; this difference is negligible. The aforementioned relation holds for the idealized surface referred to as a

<sup>13</sup>For some intuitive explanations and examples of the thermal radiation, see [2, Chap. II, Sect. 9–4].

<sup>14</sup>The law is named after Josef Stefan and Ludwig Boltzmann.

*blackbody*. Real surfaces emit less energy than the blackbody and this rate is measured by the *emissivity*,  $\varepsilon$ , of the surface:

$$\tilde{q} = \varepsilon \sigma T^4 . \quad (2.35)$$

The emissivity is between zero and one. Even in vacuum this type of energy exchange occurs. If the body is surrounded by air at  $T_{\text{amb}}$  then the energy rate emitted from the body reads

$$\tilde{q} = \varepsilon \sigma (T^4 - T_{\text{amb}}^4) . \quad (2.36)$$

We can now combine the natural convection and the surface radiation in order to find out the boundary condition as follows

$$\hat{q} = q_i n_i = h(T - T_{\text{amb}}) + \varepsilon \sigma (T^4 - T_{\text{amb}}^4) , \quad (2.37)$$

for a body embedded in resting air. If the air possesses a velocity, as in case of a forced convection, the value of  $h$  increases and the heat exchange via convection dominates; the surface radiation can be completely neglected.

We acquire the following weak form for the computation:

$$\begin{aligned} \text{Form} = & \int_{\mathcal{B}} \left( \frac{\rho c}{\Delta t} (T - T_0) \delta T - q_i \frac{\partial \delta T}{\partial x_i} - \rho L \delta T \right) dv + \\ & + \int_{\partial \mathcal{B}} \left( h(T - T_{\text{amb}}) + \varepsilon \sigma (T^4 - T_{\text{amb}}^4) \right) \delta T da , \end{aligned} \quad (2.38)$$

with the heat flux,  $q_i$ , as in Eq. (2.30). Obviously, the weak form is nonlinear due to the thermal radiation in the boundary conditions.

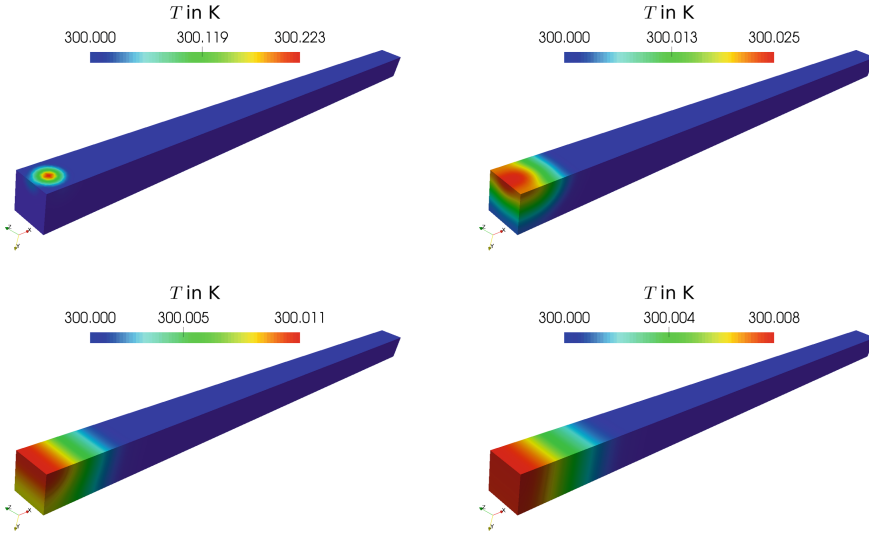
Suppose we have a tiny beam of  $100 \times 5 \times 5 \mu\text{m}$  excited on one end with a laser beam. Taken from [18] time lag parameters are given below in picoseconds:

$$\begin{aligned} & \text{for Cu , } \tau_T = 70.833 \text{ ps , } \tau_q = 0.4648 \text{ ps ,} \\ & \text{for Au or Ag , } \tau_T = 89.286 \text{ ps , } \tau_q = 0.7438 \text{ ps ,} \\ & \text{for Pb , } \tau_T = 12.097 \text{ ps , } \tau_q = 0.1720 \text{ ps .} \end{aligned} \quad (2.39)$$

We choose an appropriate unit system:

$$1 \text{ ps} = 10^{-12} \text{ s} , \quad 1 \mu\text{g} = 10^{-9} \text{ kg} , \quad 1 \mu\text{m} = 10^{-6} \text{ m} , \quad 1 \text{ nN} = 10^{-9} \text{ N} , \quad (2.40)$$

where energy is in femtoJoule,  $1 \text{ nN} \times 1 \mu\text{m} = 10^{-15} \text{ Nm} = 1 \text{ fJ}$ , and temperature in K as usual. A short pulse of laser on one end ignites a heat transport to the surface and toward to other end, see Fig. 2.3. The implementation with consistent units has been realized by the code below.



**Fig. 2.3** Temperature distribution at 1, 15, 50, and 100 ns in the gold bar, consider the change of the temperature scale for a better visualization

```

1 """ Computational reality 11, temperature distribution in a
   ↳ microscopic body """
2 --author-- = "B. Emek Abali"
3 --license-- = "GNU GPL Version 3.0 or later"
4 #This code underlies the GNU General Public License ,
   ↳ http://www.gnu.org/licenses/gpl-3.0.en.html
5
6 from fenics import *
7 import numpy
8 parameters["allow_extrapolation"]=True
9 parameters["form_compiler"]["cpp_optimize"] = True
10 set_log_level(ERROR)
11
12 rho=19.3E-6 # mass density of gold (Au) in mug/mum^3]
13 c=129.0E6 # heat capacity in fJ/(mug K)
14 kappa=318.0E-3 # thermal conductivity in fJ/(mum K ps)
15 tau_T = 89.286 # in ps
16 tau_q = 0.7438 # in ps
17 h=18.0E-9 # natural convection coefficient in fJ/(ps mum^2
   ↳ K)]
18 emis=0.47 # emissivity of gold not polished
19 sigma=5.670E-17 # Stefan-Boltzmann constant in fJ/(ps mum**2)
20 Ta=300.0 # ambient temperature in K
21 l=100.0 # length in mum
22 th=5.0 # thickness in mum
23 P=30.0E3 # laser power in fJ/(mum ps)
24
25 mesh = BoxMesh(Point(0.0,0.0,0.0), Point(l,th,th), 200,10,10)
26 Space = FunctionSpace(mesh, 'P', 1)
27 VectorSpace = VectorFunctionSpace(mesh, 'P', 1)
28 cells = CellFunction('size_t', mesh)

```

```

29 facets = FacetFunction('size_t', mesh)
30 da = Measure('ds', domain=mesh, subdomain_data=facets)
31 dv = Measure('dx', domain=mesh, subdomain_data=cells)
32 t=0.0
33 t_end=100000.0 #0.1 mus
34 Dt=1000.0
35 initial_T = Expression("Tini", Tini=Ta)
36 T0 = interpolate(initial_T, Space)
37 Laser = Expression("P*exp(-1.0*(pow(x[0]-2, 2)+pow(x[1], 2)+
    ↪ pow(x[2]-2.5, 2)))", P=P)
38 T = Function(Space)
39 del_T = TestFunction(Space)
40 dT = TrialFunction(Space)
41 q0 = Function(VectorSpace)
42 G = as_tensor(T.dx(i), (i,))
43 G0 = as_tensor(T0.dx(i), (i,))
44 q = as_tensor(Dt/(Dt+tau_q)*(tau_q/Dt*q0[i]-kappa*(1+tau_T/Dt
    ↪ )*G[i]+kappa*tau_T/Dt*G0[i]), (i,))
45 Form = (rho*c/Dt*(T-T0)*del_T \
46 - q[i]*del_T.dx(i) \
47 - rho*Laser*del_T ) * dv \
48 + (h*(T-Ta) \
49 + emis*sigma*(T**4-Ta**4) )*del_T*da
50 Gain = derivative(Form, T, dT)
51
52 file_T = File("/calcul/CR11/T.pvd")
53 for t in numpy.arange(0,t_end,Dt):
54     print "Time ",t
55     if t>=2000.: Laser.P=0
56     solve(Form==0, T, [], J=Gain, \
57           solver_parameters={"newton_solver":{"linear_solver":
    ↪ "mumps", "relative_tolerance": 1e-3}}, \
58           form_compiler_parameters={"cpp_optimize": True, "
    ↪ representation": "quadrature", "
    ↪ quadrature_degree": 2} )
59
60     if t==int(t): file_T << (T,t)
61     q0 = project(q, VectorSpace)
62     T0.assign(T)

```

## To-do

Temperature distribution with a time lag has been implemented.

- Change the boundary condition to a DIRICHLET boundary without radiation.
- Simulate the same problem with FOURIER's law.
- Produce a 2D plot with temperature versus  $x$ -coordinates. Plot the results with CATTANEO-VERNOTTE's and FOURIER's law on top of each other and analyse the difference.

### 2.3 Thermodynamics in a Nutshell

Theoretical thermodynamics concerns derivation of the constitutive equations, which we have already been using in the former sections. For some viscous fluids, for example water, we are certain that the NAVIER–STOKES equation is an adequate model to describe the flow behavior. The material model given by the constitutive equation has been attained phenomenologically (by using empirical research). In the early 1940s the concept of continuum mechanics has been redesigned under the name *rational mechanics*. Different scientific branches of mechanics: solid body mechanics, fluid mechanics, and applied thermodynamics, have been fused by using this concept. Such an abstraction of different studies has lead to theoretical thermodynamics<sup>15</sup> used to derive the constitutive equations.

Theoretical thermodynamics is a non-unique approach. At least four prominent methodologies can be listed: the COLEMAN–NOLL procedure, MULLER’s rational thermodynamics, and non-equilibrium thermodynamics.<sup>16</sup> They deliver the basic equations like NAVIER–STOKES’s equation such that we believe that all methodologies are correct. We omit an introduction and discussion of different methodologies and design a method using elements from all of them. The output is again the well-known equations for simple viscous fluids such that we can convince ourselves that the procedure is acceptable. The proposed method possesses some limitations that we will remark by presenting and applying the procedure in the following. Although the method has some weak points, it is relatively simple and allows to be generalized easily in order to involve electromagnetic interactions in Chap. 3.

We have introduced and implemented the following three balance equations in their local forms: the balance of mass, the balance of linear momentum, and the balance of internal energy, respectively:

$$\rho \dot{\bullet} + \rho \frac{\partial v_i}{\partial x_i} = 0, \quad \rho v_i \dot{\bullet} - \frac{\partial \sigma_{ji}}{\partial x_j} - \rho f_i = 0, \quad \rho u \dot{\bullet} + \frac{\partial q_i}{\partial x_i} - \rho r = \sigma_{ij} \frac{\partial v_j}{\partial x_i}, \quad (2.41)$$

in the current frame expressed in a Cartesian coordinate system. The first two has zero sources, i.e., zero right-hand sides and the balance of internal energy has a non-zero source. The source is a production term. We cannot eliminate or “shut off” the production term. For example the production term of the internal energy—known as the internal friction—alters the internal energy, as long as the material undergoes a motion. Internal energy fails to be a conserved quantity. Mass and linear momentum are conserved quantities, since they lack a production term.

From the balance equations we want to solve the mass density,  $\rho$ , the velocity,  $v_i$ , and the temperature,  $T$ . First we need to close the balance equations by defining the constitutive equations for the CAUCHY stress,  $\sigma_{ji}$ , for the specific internal energy,  $u$ ,

<sup>15</sup>Thermodynamics of irreversible processes started with papers of Carl Eckart, see [4], [5], [6], [7].

<sup>16</sup>The COLEMAN–NOLL procedure is named after Bernard D. Coleman and Walter Noll, see [3]. MULLER’s rational thermodynamics is named for Ingo Müller, see [11]. The non-equilibrium thermodynamics is introduced by Sybren Ruurds de Groot and Peter Mazur, see [8].



and for the heat flux,  $q_i$ . After having found the definitions, the formulation will be complete such that we can generate a weak form and solve the balance equations augmented with the constitutive equations. Theoretical thermodynamics has the aim of determining the constitutive equations. In this section we will perform the necessary steps leading to the adequate constitutive equations modeling a viscous fluid.

We want to compute  $(\rho, v_i, T)$  for a fluid in a EULERIAN frame,  $(x_i, t)$ . The unknowns  $(\rho, v_i, T)$  are referred to as *primitive* variables: their mathematical existence is accepted without any further investigation. Our goal is to compute the primitive variables by satisfying the balance equations. We can solve the balance equations if they are closed: All constitutive equations for  $\sigma_{ij}$ ,  $q_i$ , and  $u$  are given by functions depending on the primitive variables or their time and space derivatives.

The first assumption is that the specific total energy,  $e$ , is additive so that the specific internal energy,  $u$ , and specific kinetic energy,  $e^{\text{kin}}$ , are separable and independent. Hence, the internal energy fails to depend on the velocity. By following [14] we assume that the change of the internal energy<sup>17</sup> is recoverable. The kinetic energy is irreversible. There are many different formulations in the literature and none of them is wrong, because we cannot measure different parts of the energy separately; they are just various models approximating the behavior of the material with different accuracies. The accuracy of any formalism can only be tested by measurements. In the end, the primitive variables are calculated with adequate accuracy, if the material modeling is appropriate. Without discussing its limitations, we assume that the internal energy possesses only terms occurring a recoverable change. This assumption is of paramount importance and leads to a useful methodology described in the following.

We introduce mass density,  $\rho(x_i, t)$ , velocity,  $v_i(x_j, t)$ , and temperature  $T(x_i, t)$  as primitive variables. We axiomatically assume that they are independent functions.<sup>18</sup> Velocity fails to be an *objective* variable. If we perform a EUCLIDIAN coordinate transformation, velocity transforms other than a tensor of rank one. Fortunately, symmetric velocity gradient,  $d_{ij} = \partial v_i / \partial x_j$ , is an objective variable, it is a tensor of rank two.<sup>19</sup> Constitutive equations shall depend on objective variables.

We start the formulation by proposing an equation for the internal energy. Actually, this equation can be derived in various ways. We present here a method based on the balance of internal energy since we will make much use of it in the next chapter.<sup>20</sup> Consider a material of having the following constitutive equation:

$$\sigma_{ji} = {}^r\sigma_{ji} + {}^d\sigma_{ji} , \quad (2.42)$$

<sup>17</sup>Under the assumption that no phase changes occur.

<sup>18</sup>Of course they are coupled, however, independent. We can hold the temperature fixed and move the body or restrict any motion and change the temperature.

<sup>19</sup>The formulation holds for fluids with elasticity, too. Therefore, we need to introduce,  $d_{ij} = \dot{\varepsilon}_{ij}$ , for a fixed frame,  $w_i = 0$ . The proof of this identity is out of scope, therefore, we explain it in Appendix A.4 on p. 301.

<sup>20</sup>For another, more conventional derivation, see [1, Sect. 3].

where the reversible (recoverable) change is given by the first and irreversible (dissipative) change is described by the second term. The assumption of separating stress tensor additively is justified by the energy assumption, where we have also separated the reversible (recoverable) and irreversible (dissipative) terms into an internal and kinetic energy, respectively. For a fluid without elasticity, we introduce the reversible term by using pressure,  $p$ , as follows

$${}^r\sigma_{ji} = -p\delta_{ji} . \quad (2.43)$$

The dissipative term will be a function of  $d_{ij}$  leading to a flow in the system. The stress consists of the reversible term, if the fluid rests (zero velocity),<sup>21</sup> this state is called a mechanical *equilibrium*. In equilibrium, the pressure enables a reversible momentum transport, for example the sound waves in a fluid are transported by the pressure,  $p$ . This process is reversible since the motion of fluid particles are recovered after the sound wave has passed by. This small motion of fluid particles are neglected at all, we only calculate the velocity leading to a convection of the fluid. If the fluid rests, pressure still transports sound waves reversibly. Hence, we can call the pressure as a *hydrostatic* pressure since it is responsible for a momentum transport in the static or equilibrium state. In the mechanical equilibrium the stress reads

$$\sigma_{ji} \Big|_{\text{eq.}} = -p\delta_{ji} . \quad (2.44)$$

Formally, the decomposition in Eq. (2.42) is correct since we still have not defined  ${}^q\sigma_{ij}$ . Instead of searching for a definition of  $\sigma_{ij}$  we now have to search for  $p$  and  ${}^q\sigma_{ji}$ .

For a thermal equilibrium we introduce a quantity called a specific *entropy*,  $\eta$ , as

$$-\frac{\partial q_i}{\partial x_i} \Big|_{\text{eq.}} = q^\bullet \Big|_{\text{eq.}} , \quad \frac{1}{T} q^\bullet \Big|_{\text{eq.}} = \rho \eta^\bullet , \quad (2.45)$$

by following [8, Chap. XIV, Sect. 2]. The entropy is responsible for a reversible transport of energy, i.e., the process is in a thermal equilibrium. Thermal equilibrium is often explained as a slow temperature change, actually, it is just the reversible part of the process without any dissipation. Indeed a slow temperature change is undertaken by the reversible part. Similar to the previous case, we now have to determine a constitutive equation for  $\eta$  and  $q_i$ . The balance of internal energy:

$$\rho u^\bullet + \frac{\partial q_i}{\partial x_i} - \rho r = \sigma_{ij} \frac{\partial v_j}{\partial x_i} , \quad (2.46)$$

reads at thermal and mechanical equilibrium

---

<sup>21</sup>Since we use a EUCLIDEAN transformation to test the objectivity, a constant velocity is accepted, too. Consider a rigid body moving with a constant velocity, it actually *rests* in a coordinate system moving with this velocity. Hence, we can always introduce a constantly moving coordinate system, which is allowed in the EUCLIDEAN transformation, where the body rests.

$$\rho u^\bullet - \rho T \eta^\bullet = -p \frac{\partial v_i}{\partial x_i} , \quad (2.47)$$

since by existing internal heating (supply term), equilibrium cannot take place. We recall that the internal energy is fully recoverable. By introducing a specific volume,  $v = 1/\rho$ , a volume per mass, we can rewrite the balance of mass as follows

$$\begin{aligned} (v^{-1})^\bullet v &= -\frac{\partial v_i}{\partial x_i} , \\ \frac{v^\bullet}{v} &= \frac{\partial v_i}{\partial x_i} , \\ \rho v^\bullet &= \frac{\partial v_i}{\partial x_i} . \end{aligned} \quad (2.48)$$

Now by inserting the latter into the balance of internal energy we obtain

$$\begin{aligned} \rho u^\bullet - \rho T \eta^\bullet &= -\rho p v^\bullet , \\ u^\bullet &= T \eta^\bullet - p v^\bullet . \end{aligned} \quad (2.49)$$

Furthermore from the latter expression, we acquire so-called GIBBS's equation:

$$du = T d\eta - p dv , \quad (2.50)$$

under the condition that  $u$  has a first integral:

$$u = \int du . \quad (2.51)$$

Often, this condition is referred to as the 1st law of thermodynamics.<sup>22</sup> The integration is between two states. Suppose we start from the state,  $\{T = T_0, v = v_0\}$ , and end up in a state at another temperature and mass density (thus specific volume),  $\{T, v\}$ . Since the internal energy is a total differential we can obtain the energy by integrating from the state one to state two

$$u = \int_{(T_0, v_0)}^{(T, v)} du = u(T, v) - u(T_0, v_0) . \quad (2.52)$$

Only the first and last states are important, not the states in between. This condition is a limitation and the methodology herein with this limitation is called the *equilibrium* thermodynamics. Exactly this assumption is a deficiency on the way to a general

---

<sup>22</sup>For instance in [14] the internal energy is introduced as a full recoverable quantity such that the first integral is automatically justified. Either we can accept the axiom of existence of  $du$  as the 1st law, or the assumption that the internal energy is fully recoverable as the 1st law.

theory.<sup>23</sup> In other words, for some processes this assumption may lead to constitutive equations not capable of describing the process accurately. The fact that the total energy has a dissipative term only due to the kinetic energy might be too restrictive. The methodology presented here would not suffice for describing a process where temperature (or its rate) plays a dissipative role in the total energy. However, at least for all processes presented in this book, this restriction is admissible and the implemented method is reliable.

Usually GIBBS's equation is an axiomatic start; its validity is taken for granted. Herein, we have motivated it by using the balance of internal energy at the equilibrium. Since the internal energy is assumed to be recoverable, the differential relation holds for the non-equilibrium, too. By considering Eq. (2.50) we realize that the internal energy is a function of entropy and specific volume,

$$u = u(\eta, v) . \quad (2.53)$$

We have an inconsistency by defining energy depending on a variable,  $\eta$ , which is not yet defined. Better we shall find a constitutive equation for energy depending on the primitive variables or their derivatives, i.e., on the so-called *state* variables.<sup>24</sup> In this section the state or primary variables are  $\{T, v\}$ . Thus, we need an energy definition depending only on the primary variables. In order to obtain such an energy we introduce the so-called specific HELMHOLTZ free energy:

$$\psi = u - T\eta . \quad (2.54)$$

Its total differential is assumed to exist<sup>25</sup>

$$d\psi = du - \eta dT - T d\eta . \quad (2.55)$$

By inserting Eq. (2.50) into the latter we obtain

$$d\psi = -\eta dT - p dv . \quad (2.56)$$

From this differential form we realize that the free energy depends on the primary variables

$$\psi = \psi(T, v) . \quad (2.57)$$

---

<sup>23</sup>For a brief explanation of this deficiency, we refer to [13, Chap.2].

<sup>24</sup>Since state variables are derived from the primitive variables we can also name them as primary variables.

<sup>25</sup>This assumption is another weak point in the methodology, the 1st law of thermodynamics only states that the internal energy has a perfect differential,  $du$ , but not the free energy. Concretely, we have to make sure that  $d(T\eta)$  exists.

Obviously, we have the following relations:

$$d\psi = \frac{\partial\psi}{\partial T} dT + \frac{\partial\psi}{\partial v} dv, \quad \eta = -\frac{\partial\psi}{\partial T}, \quad p = -\frac{\partial\psi}{\partial v}. \quad (2.58)$$

Our goal is to determine the *dual* variables depending on the primary variables<sup>26</sup>

$$\eta = \eta(T, v), \quad p = p(T, v), \quad (2.59)$$

leading to the following differentials:

$$\begin{aligned} d\eta &= \frac{\partial\eta}{\partial T} dT + \frac{\partial\eta}{\partial v} dv = A dT + B dv, \\ dp &= \frac{\partial p}{\partial T} dT + \frac{\partial p}{\partial v} dv = C dT + D dv. \end{aligned} \quad (2.60)$$

We need to determine the material coefficients  $A, B, C, D$  as functions depending on the state variables. The partial derivatives are taken by holding the other arguments fixed. We skip an explicit notation about the dependency, since it is superfluous. The small change  $d(\cdot)$  is simply how we shall undertake the measurements.

We cannot measure the (specific) entropy,  $\eta$ , directly. Instead, heat (thermal energy) is measured in a calorimeter,  $\delta q = T d\eta$ . The notational difference,  $\delta q$ , is only due to the fact that we cannot form a total differential of the heat. In other words, it is necessary to integrate over the whole process that is an evolution, the knowledge of the start and end states is not sufficient for  $\delta q$ . Technically, we just measure the heat energy supplied to the system. By holding the specific volume constant,  $dv = 0$ , and by varying the temperature,  $dT$ , we obtain a change of heat,  $\delta q$ , which is measured as

$$\delta q = c dT \Rightarrow A = \frac{c}{T}, \quad (2.61)$$

where  $c$  is called the *specific heat capacity* by holding volume constant. It may depend on temperature and have to be measured for different specific volumes. Of course for different specific volumes, the numerical value of  $c$  may vary, too. Hence,  $c = c(T, v)$ , at least in principle. These measurements are difficult such that either heat capacity is assumed to be constant or solely the dependence on the temperature is found in the literature.

We can easily measure the coefficient  $D$  by fixing the temperature,  $dT = 0$ ; varying the specific volume,  $dv$ ; and measuring the pressure change,  $dp$ . It is possible to change the temperature and measure the pressure for a fixed (specific) volume for determining  $C$ . However, it is rather difficult to set the temperature constant and measure the heat exchange due to the variation in the specific volume. The problem is

---

<sup>26</sup>Although this is mathematically obvious that the dual variables have to depend on the same set of arguments of the free energy, namely on the state variables, this condition is called the *equipresence* principle, see [17, Sect. 293.η].

that we normally measure the heat change over the temperature measurement. There is a MULLER-calorimeter for this purpose but it is not appropriate for all materials.<sup>27</sup>

Fortunately, we can skip measurements for determining  $C$  coefficient because  $C = B$ . In order to see this relation, we write the dual variables in a matrix notation:

$$\begin{pmatrix} d\eta \\ dp \end{pmatrix} = \begin{pmatrix} c/T & B \\ C & D \end{pmatrix} \begin{pmatrix} dT \\ dv \end{pmatrix}. \quad (2.62)$$

The condition  $B = C$  is a *symmetry* condition in the matrix of coefficients. This condition can be seen readily by using the free energy depending on  $T$  and  $v$ ,

$$B = \frac{\partial \eta}{\partial v} = - \frac{\partial^2 \psi}{\partial v \partial T} = - \frac{\partial^2 \psi}{\partial T \partial v} = \frac{\partial p}{\partial T} = C. \quad (2.63)$$

This condition is known as MAXWELL's reciprocal relation.<sup>28</sup> For a viscous fluid we can integrate and find out the constitutive equations:

$$\eta = \int d\eta = \int \frac{c}{T} dT + \int B dv, \quad p = \int dp = \int B dT + \int D dv, \quad (2.64)$$

from a reference state  $\{T_{\text{ref}}, v_{\text{ref}}\}$  to the current state  $\{T, v\}$ . For a *linear* material the coefficients are constants such that we readily obtain

$$\begin{aligned} \eta &= c \ln \left( \frac{T}{T_{\text{ref}}} \right) + B(v - v_{\text{ref}}), \\ p &= B(T - T_{\text{ref}}) + D(v - v_{\text{ref}}). \end{aligned} \quad (2.65)$$

For an incompressible material  $v = v_{\text{ref}}$ , such that  $B$  and  $D$  fails to be measured by a variation of the specific volume, since  $dv = 0$ . GIBBS's equation becomes  $du = c dT$  and the specific entropy reads

$$\eta = c \ln \left( \frac{T}{T_{\text{ref}}} \right). \quad (2.66)$$

We calculate  $p$  from the balance of mass.

In order to derive the heat flux,  $q_i$ , and the dissipative part of the stress tensor,  $\phi_{ij}$ , we continue the methodology in the following. Since we have defined the internal energy, we can insert Eq. (2.49)<sub>2</sub>:

$$\dot{u} = T\dot{\eta} - p\dot{v} \quad (2.67)$$

<sup>27</sup>See [10] for a detailed explanation about the MULLER-calorimeter named after F. Horst Müller.

<sup>28</sup>It is named after James Clerk Maxwell.

into the balance of internal energy in Eq.(2.41)<sub>3</sub> and obtain

$$\begin{aligned} T\rho\eta^\bullet - p\rho v^\bullet + \frac{\partial q_i}{\partial x_i} - \rho r &= \sigma_{ij} \frac{\partial v_j}{\partial x_i}, \\ T\rho\eta^\bullet + \frac{\partial q_i}{\partial x_i} - \rho r &= {}^d\sigma_{ij} \frac{\partial v_j}{\partial x_i}, \end{aligned} \quad (2.68)$$

after having used the balance of mass as in Eq.(2.48). We can rewrite the latter equation further,

$$\begin{aligned} \rho\eta^\bullet + \frac{1}{T} \frac{\partial q_i}{\partial x_i} - \rho \frac{r}{T} &= \frac{1}{T} {}^d\sigma_{ij} \frac{\partial v_j}{\partial x_i}, \\ \rho\eta^\bullet + \frac{\partial}{\partial x_i} \left( \frac{q_i}{T} \right) - q_i \frac{\partial}{\partial x_i} \left( \frac{1}{T} \right) - \rho \frac{r}{T} &= \frac{1}{T} {}^d\sigma_{ij} \frac{\partial v_j}{\partial x_i}, \end{aligned} \quad (2.69)$$

in order to acquire a balance equation:

$$\rho\eta^\bullet + \frac{\partial}{\partial x_i} \left( \frac{q_i}{T} \right) - \rho \frac{r}{T} = - \frac{q_i}{T^2} \frac{\partial T}{\partial x_i} + \frac{1}{T} {}^d\sigma_{ij} \frac{\partial v_j}{\partial x_i}. \quad (2.70)$$

This balance equation is the balance of entropy:

$$\rho\eta^\bullet + \frac{\partial \Phi_i}{\partial x_i} - \rho \frac{r}{T} = \Sigma, \quad (2.71)$$

with the flux term,  $\Phi_i$ , and the production term,  $\Sigma$ , as follows

$$\Phi_i = \frac{q_i}{T}, \quad \Sigma = - \frac{q_i}{T^2} \frac{\partial T}{\partial x_i} + \frac{1}{T} {}^d\sigma_{ij} \frac{\partial v_j}{\partial x_i}. \quad (2.72)$$

The 2nd law of thermodynamics asserts that any process attains a positive entropy production:

$$\Sigma \geq 0. \quad (2.73)$$

This law restricts the possible constitutive relations. In other words,  $q_i, \sigma_{ji}$  have to be such that  $\Sigma \geq 0$  is assured for every possible processes. This restriction leads to the constitutive equations for  $q_i$  and  $\sigma_{ij}$  as presented in the following in three steps. First we introduce the following notation:

$$G_i = \frac{\partial T}{\partial x_i}, \quad d_{ij} = \frac{\partial v_{[i}}{\partial x_{j])}, \quad d_{|ij|} = d_{ij} - \frac{1}{3} d_{kk} \delta_{ij}. \quad (2.74)$$

Moreover, in non-polar materials the dissipative stress is symmetric,  ${}^d\sigma_{ij} = {}^d\sigma_{ji}$ . We exclude polar materials in this book.<sup>29</sup> Secondly, we rewrite the entropy production:

---

<sup>29</sup>Nematic fluids used in LCD (Liquid Crystal Display) is a prominent polar material.

$$\Sigma = -\frac{1}{T^2} q_i G_i + \frac{1}{3T} {}^d\sigma_{ii} d_{jj} + \frac{1}{T} {}^d\sigma_{|ij|} d_{|ij|} \geq 0. \quad (2.75)$$

This multiplication can be seen as *thermodynamical fluxes*:

$$\mathcal{F}^\alpha = \left\{ -q_i, {}^d\sigma_{ii}, {}^d\sigma_{|ij|} \right\}, \quad (2.76)$$

multiplied (by an inner product) with *thermodynamical forces*:

$$\mathcal{K}^\alpha = \left\{ \frac{G_i}{T^2}, \frac{1}{3T} d_{jj}, \frac{1}{T} d_{|ij|} \right\}, \quad (2.77)$$

as follows

$$\Sigma = \mathcal{F}^\alpha \cdot \mathcal{K}^\alpha \geq 0, \quad \alpha = 1, 2, 3. \quad (2.78)$$

The thermodynamical forces are independent among each other. Thirdly, each thermodynamical force is of another rank. For an isotropic material a thermodynamical flux may depend only on the same rank of the thermodynamical forces. Since otherwise under a coordinate transformation different rank tensors transform differently such that the dependency of one flux component on the force component changes. Hence, for isotropic materials the thermodynamical fluxes depend only on the thermodynamical forces of the same rank:

$$\mathcal{F}^1 = \mathcal{F}^1(\mathcal{K}^1), \quad \mathcal{F}^2 = \mathcal{F}^2(\mathcal{K}^2), \quad \mathcal{F}^3 = \mathcal{F}^3(\mathcal{K}^3). \quad (2.79)$$

This condition is known as the CURIE symmetry principle.<sup>30</sup> Consider the following relations:

$$-q_i = \bar{a} \frac{1}{T^2} G_i, \quad {}^d\sigma_{ii} = \bar{b} \frac{1}{3T} d_{jj}, \quad {}^d\sigma_{|ij|} = \bar{c} \frac{1}{T} d_{|ij|}, \quad (2.80)$$

where the material coefficients may depend on the thermodynamical forces.<sup>31</sup> If we insert the constitutive relations into the entropy production

$$\Sigma = \bar{a} \frac{1}{T^4} G_i G_i + \bar{b} \frac{1}{9T^2} d_{ii} d_{jj} + \bar{c} \frac{1}{T^2} d_{|ij|} d_{|ij|} \geq 0, \quad (2.81)$$

has to hold. We know that every term is a multiplication between different types of fluxes and forces, i.e., every term is independent. In order to demand  $\Sigma \geq 0$  for any process, the coefficients have to be

$$\bar{a} \geq 0, \quad \bar{b} \geq 0, \quad \bar{c} \geq 0, \quad (2.82)$$

<sup>30</sup>It is named after Pierre Curie.

<sup>31</sup>For the sake of clarity, the coefficients are functions of the invariants of thermodynamical forces.



since  $T > 0$  in K(elvin). This conclusion is quite general and the coefficients may be scalar functions of the thermodynamical forces, i.e., the thermodynamical fluxes may depend on the thermodynamical forces nonlinearly. If we rename the coefficients:

$$\bar{a} \frac{1}{T^2} = \kappa, \quad \bar{b} \frac{1}{3T} = 3\lambda + 2\mu, \quad \bar{c} \frac{1}{T} = 2\mu, \quad (2.83)$$

and simplify to *linear* materials by assuming that  $\kappa$ ,  $\lambda$ , and  $\mu$  are constant, then we end up in a NAVIER–STOKES–FOURIER fluid:

$$\begin{aligned} q_i &= -\kappa G_i, \\ \sigma_{ij} &= -p\delta_{ij} + {}^d\sigma_{ij} = -p\delta_{ij} + \frac{1}{3}(3\lambda + 2\mu)d_{kk}\delta_{ij} + 2\mu d_{[ij]} = \\ &= (-p + \lambda d_{kk})\delta_{ij} + 2\mu d_{ij}. \end{aligned} \quad (2.84)$$

Obviously the material constants are  $\kappa \geq 0$ ,  $\mu \geq 0$ , and  $3\lambda + 2\mu \geq 0$ , in order to fulfill the 2nd law. Since  $d_{ii} = \rho v^*$  for an incompressible fluid, commonly  $3\lambda + 2\mu = 0$  is used, which is known as STOKES's hypothesis. However, this hypothesis cannot be verified experimentally. Since  $d_{ii} = 0$  holds for an incompressible fluid flow, it is not possible to detect the value of  $3\lambda + 2\mu$ . We will never use this hypothesis. Certainly, by utilizing  $d_{ii} = 0$  we assume that the flow is incompressible. This interpretation is correct, incompressibility is not a material property; even water flows compressible under great pressure and temperature conditions.<sup>32</sup> By using the incompressibility we spare the mass balance for the computation of  $p$ . For a numerical stability  $\lambda$  shall be a large number.

For a non-linear fluid we can now quickly generalize the constitutive equations and propose

$$\bar{c} \frac{1}{T} = 2 \left( \mu_0 + \frac{k}{\pi \sqrt{II}} \arctan \left( \frac{\sqrt{II}}{B} \right) \right), \quad II = \frac{1}{2} d_{ij} d_{ij}, \quad (2.85)$$

as already employed in Sect. 1.8. For the thermodynamical consistency, material parameters,  $\mu_0$ ,  $k$ ,  $B$ , have positive values.

We want to compute the same channel problem as in Sect. 1.7, this time by computing not only the pressure and velocity but also the temperature change. We choose water, a viscous linear fluid and model with FOURIER–NAVIER–STOKES constitutive equations:

$$\sigma_{ij} = -p\delta_{ij} + {}^d\sigma_{ij}, \quad {}^d\sigma_{ij} = \lambda d_{kk}\delta_{ij} + 2\mu d_{ij}, \quad q_i = -\kappa \frac{\partial T}{\partial x_i}. \quad (2.86)$$

---

<sup>32</sup>However, if we use  $3\lambda + 2\mu = 0$  then we assume that the incompressibility is a property of the material. This is definitely not the case, see [9] and also [15].

Since water can be assumed as incompressible,  $dv = 0$ , we have the rate of internal energy from GIBBS's equation:

$$\rho u^* = \rho T \eta^* = \rho c T^* . \quad (2.87)$$

For a fixed<sup>33</sup> domain,  $\Omega$ , we obtain a weak form from Eq(2.41)<sub>1</sub> for computing the pressure:

$$F_p = \int_{\Omega} \left( \frac{\partial \rho}{\partial t} + v_i \frac{\partial \rho}{\partial x_i} + \rho \frac{\partial v_i}{\partial x_i} \right) \frac{\delta p}{\rho} dv . \quad (2.88)$$

This integral form is in the unit of power. For an incompressible flow the latter reduces to

$$F_p = \int_{\Omega} \frac{\partial v_i}{\partial x_i} \delta p dv . \quad (2.89)$$

Analogously the balance of linear momentum in Eq(2.41)<sub>2</sub> leads to a weak form for computing the velocity:

$$\begin{aligned} F_v = \int_{\Omega} \left( \rho \frac{\partial v_i}{\partial t} \delta v_i + \rho v_j \frac{\partial v_i}{\partial x_j} \delta v_i + \frac{\partial p}{\partial x_i} \delta v_i + {}^d\sigma_{ji} \frac{\partial \delta v_i}{\partial x_j} - \rho f_i \delta v_i \right) dv - \\ - \int_{\partial\Omega} {}^d\sigma_{ji} n_j \delta v_i da . \end{aligned} \quad (2.90)$$

This integral form is also in the unit of power. If we want to apply a NEUMANN boundary by defining a mechanical pressure applied on the boundary, then the traction vector:

$$\hat{t}_i = \sigma_{ji} n_j = -p n_i + {}^d\sigma_{ji} n_j , \quad (2.91)$$

is necessary. We can apply a mechanical pressure on left and right openings,  $\hat{p}$ . The mechanical pressure on a boundary reads

$$-\frac{1}{3} \sigma_{kk} = \hat{p} = -\frac{1}{3} ( -p \delta_{kk} + {}^d\sigma_{kk} ) . \quad (2.92)$$

For applying this pressure we need the traction vector inward the domain

$$\hat{t}_i = -n_i \hat{p} = \frac{n_i}{3} ( -p \delta_{kk} + {}^d\sigma_{kk} ) . \quad (2.93)$$

For an incompressible flow,  $d_{kk} = 0$ , the spherical dissipative stress vanishes such that we obtain

$$\hat{t}_i = -p n_i , \quad (2.94)$$

---

<sup>33</sup>In a fixed domain we simply write  $\frac{d(\cdot)}{dt}$  instead of  $(\cdot)^*$  and obtain the balance equations for open systems.

leading to

$${}^d\sigma_{ji}n_j = \hat{t}_i + pn_i = 0. \quad (2.95)$$

Due to the latter the boundary terms vanish in Eq. (2.91), i.e., in general for incompressible flows, the boundary terms in the weak form of velocity vanish.

In order to compute the temperature, we need a weak form in the unit of power. Either we generate it from the balance of internal energy as in Eq. (2.41)<sub>3</sub> by dividing by  $T$  and multiplying with  $\delta T$ , or from the balance of entropy as in Eq. (2.70) by multiplying with  $\delta T$ . For the first option we need to use  $\rho u^* = \rho c T^*$  and for the second option  $\eta^* = T^* c / T$ . The result is the same. We use the entropy balance and obtain

$$\int_{\Omega} \left( \frac{\rho c}{T} \frac{\partial T}{\partial t} + \frac{\rho c}{T} v_i \frac{\partial T}{\partial x_i} + \frac{\partial}{\partial x_i} \left( \frac{q_i}{T} \right) - \frac{\rho r}{T} - \frac{1}{T} \sigma_{ij} \frac{\partial v_j}{\partial x_i} + \frac{1}{T^2} q_i \frac{\partial T}{\partial x_i} \right) \delta T \, dv = 0. \quad (2.96)$$

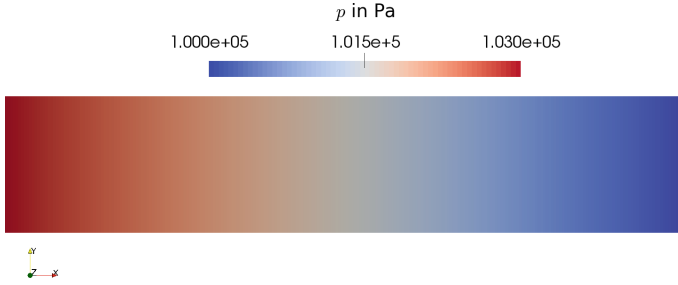
The term with the heat flux needs to be integrated by parts, since it consists a second gradient of temperature. Hence we acquire the weak form

$$\begin{aligned} F_T = \int_{\Omega} & \left( \frac{\rho c}{T} \frac{\partial T}{\partial t} \delta T + \frac{\rho c}{T} v_i \frac{\partial T}{\partial x_i} \delta T - \frac{q_i}{T} \frac{\partial \delta T}{\partial x_i} - \frac{\rho r}{T} \delta T - \right. \\ & \left. - \frac{1}{T} \sigma_{ij} \frac{\partial v_j}{\partial x_i} \delta T + \frac{1}{T^2} q_i \frac{\partial T}{\partial x_i} \delta T \right) dv + \int_{\partial\Omega} \frac{1}{T} \hat{q} \delta T \, da. \end{aligned} \quad (2.97)$$

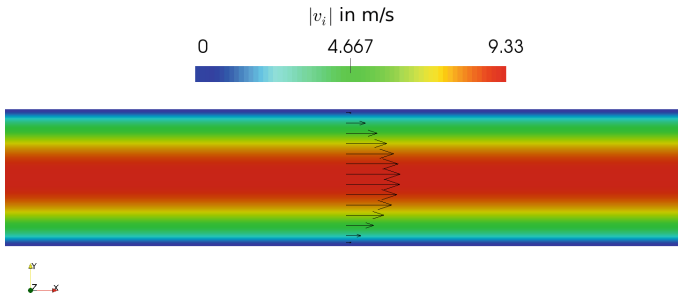
After utilizing the time discretization we can sum up all integral forms since they are all in the unit of power

$$\begin{aligned} \text{Form} = F_p + F_v + F_T = \int_{\Omega} & \left( v_{i,i} \delta p + \rho \frac{v_i - v_i^0}{\Delta t} \delta v_i + \rho v_j v_{i,j} \delta v_i + \right. \\ & + p_{,i} \delta v_i + {}^d\sigma_{ji} \delta v_{i,j} - \rho f_i \delta v_i + \frac{\rho c}{T} \frac{T - T^0}{\Delta t} \delta T + \frac{\rho c}{T} v_i T_{,i} \delta T - \frac{q_i}{T} \delta T_{,i} - \\ & \left. - \frac{\rho r}{T} \delta T - \frac{1}{T} \sigma_{ij} v_{j,i} \delta T + \frac{1}{T^2} q_i T_{,i} \delta T \right) dv + \int_{\partial\Omega} \frac{1}{T} \hat{q} \delta T \, da, \end{aligned} \quad (2.98)$$

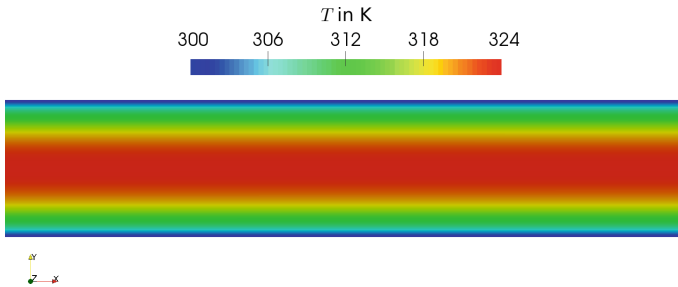
where we have employed the usual comma notation for a partial space derivative. This form can be solved by applying appropriate boundary conditions. In a 2D channel filled with water, the primitive variables,  $\{p, v_i, T\}$ , are computed. On top and bottom walls fluid rests at 300 K temperature. Due to the greater pressure on the left side than on the right hand side, water flows from left to right. The pressure on the left rises linearly in time such that the viscous fluid moves faster in time, with the typical parabolic flow profile. The production term increases the temperature because of the viscous flow. We keep on top and bottom at 300 K and on the left and right we implement adiabatic boundaries. In Figs. 2.4, 2.5 and 2.6 we present distributions of



**Fig. 2.4** Pressure distribution in channel flow at 300 s



**Fig. 2.5** Velocity distribution in channel flow at 300 s. Colors denote the magnitude of velocity



**Fig. 2.6** Distribution of temperature in channel flow at 300 s. Temperature increases due to the internal friction. It is forced to be 300 K on *top* and *bottom* boundaries

primitive variables after 5 min obtained with the code below. The viscous flow with a relatively high velocity causes a significant temperature change. We have used the code given below. By using a mixed function space for primitive variables we have computed all unknowns at once, i.e., monolithically. This method is necessary since we have inserted the balance equations in each other by obtaining the governing equations. Moreover, the field equations are coupled and nonlinear, so we are not allowed to solve them separately.

```

1  """Computational reality 12, channel flow of Navier-Stokes-
    ↪ Fourier fluid"""
2  __author__ = "B. Emek Abali"
3  __license__ = "GNU GPL Version 3.0 or later"
4  #This code underlies the GNU General Public License ,
    ↪ http://www.gnu.org/licenses/gpl-3.0.en.html
5
6  from fenics import *
7  import numpy
8  set_log_level(ERROR)
9
10 xlength = 0.5 # m
11 ylength = 0.1 # m
12 mesh=RectangleMesh(Point(0.0,- ylength/2.0), Point(xlength,
    ↪ ylength/2.0), 200,40)
13
14 TensorSpace = TensorFunctionSpace(mesh, 'P', 1)
15 VectorSpace = VectorFunctionSpace(mesh, 'P', 1)
16 ScalarSpace = FunctionSpace(mesh, 'P', 1)
17 # p, v, T
18 Space = MixedFunctionSpace([ScalarSpace, VectorSpace,
    ↪ ScalarSpace])
19
20 facets = FacetFunction('size_t', mesh)
21 cells = CellFunction('size_t', mesh)
22 da = Measure('ds', domain=mesh, subdomain_data=facets)
23 dv = Measure('dx', domain=mesh, subdomain_data=cells)
24
25 left = CompiledSubDomain('near(x[0],0) && on_boundary')
26 right = CompiledSubDomain('near(x[0],1) && on_boundary', l=
    ↪ xlength)
27 bottom = CompiledSubDomain('near(x[1],- y1/2.0)', y1=ylength)
28 top = CompiledSubDomain('near(x[1], y1/2.0)', y1=ylength)
29
30 v_noslip = Constant((0.0, 0.0))
31 pL = Expression('100000.0+10*t', t=0)
32 pR = Constant(100000.0)
33 Tini = 300. #K
34 bc1=DirichletBC(Space.sub(0), pL, left)
35 bc2=DirichletBC(Space.sub(0), pR, right)
36 bc3=DirichletBC(Space.sub(1), v_noslip, bottom)
37 bc4=DirichletBC(Space.sub(1), v_noslip, top)
38 bc5=DirichletBC(Space.sub(1).sub(1), 0.0, left)
39 bc6=DirichletBC(Space.sub(1).sub(1), 0.0, right)
40 bc7=DirichletBC(Space.sub(2), Tini, top)
41 bc8=DirichletBC(Space.sub(2), Tini, bottom)
42
43 bc=[bc1, bc2, bc3, bc4, bc5, bc6, bc7, bc8]
44 u_init = Expression(('p0', '0.0', '0.0', 'T0'), p0=100000.0, T0=
    ↪ Tini)
45
46 i, j, k, l = indices(4)
47 n = FacetNormal(mesh)

```

```

48 | t = 0.0
49 | Dt = 50.
50 | t_end = 500.
51 |
52 | test = TestFunction(Space)
53 | du = TrialFunction(Space)
54 | u0 = Function(Space)
55 | u = Function(Space)
56 | u0 = interpolate(u_init, Space)
57 | u = interpolate(u_init, Space)
58 | p0, v0, T0 = split(u0)
59 | p, v, T = split(u)
60 | delp, delv, delT = split(test)
61 | delta = Identity(2)
62 |
63 | #water approx. at 300 K
64 | rho = 995.7 #kg/m^3
65 | mu = 0.8 #Ns/m^2 = kg / s /m
66 | lambada = mu*1E5
67 | c = 4180. #J/(kgK)
68 | kappa = 0.58 #W/(m K)
69 | h=18.0 #W/(m^2 K)
70 | Tamb = Tini
71 |
72 | d = sym(grad(v))
73 | dsigma = as_tensor( lambada*d[k,k]*delta[i,j] + 2.0*mu*d[i,j
    ↪ ], (i,j))
74 | sigma = as_tensor( -p*delta[i,j] + dsigma[i,j], (i,j) )
75 | q = as_tensor( -kappa*T.dx(i), (i,) )
76 | f = Constant((0,0))
77 | r = Constant(0)
78 |
79 | Form = ( v[i].dx(i)*delp \
80 |         + rho*(v-v0)[i]/Dt*delv[i] \
81 |         + rho*v[j]*v[i].dx(j)*delv[i] \
82 |         + p.dx(i)*delv[i] \
83 |         + dsigma[j,i]*delv[i].dx(j) \
84 |         - rho*f[i]*delv[i]
85 |         + rho*c/T*(T-T0)/Dt*delT \
86 |         + rho*c/T*v[i]*T.dx(i)*delT \
87 |         - q[i]/T*delT.dx(i) \
88 |         - rho*r/T*delT \
89 |         - 1./T*sigma[i,j]*v[j].dx(i)*delT \
90 |         + 1./T**2*q[i]*T.dx(i)*delT \
91 |         )*dv
92 |
93 | Gain = derivative(Form, u, du)
94 |
95 | pwd='/calcul/CR12/'
96 | file_p = File(pwd+'pressure.pvd')
97 | file_v = File(pwd+'velocity.pvd')
98 | file_T = File(pwd+'temperature.pvd')
99 |

```

```

100 for t in numpy.arange(0,t_end,Dt):
101     print 'time:',t
102     pL.t = t
103     solve(Form==0, u, bc, J=Gain, \
104           solver_parameters={"newton_solver":{"linear_solver":
105                               ↪ "mumps", "relative_tolerance": 1e-5}}, \
106           form_compiler_parameters={"cpp_optimize": True, "
107                               ↪ representation": "quadrature", "
108                               ↪ quadrature_degree": 2} )
109     file_p << (u.split()[0], t)
107     file_v << (u.split()[1], t)
108     file_T << (u.split()[2], t)
109     u0.assign(u)

```

### To-do

We have implemented the same channel flow as in Sect. 1.7 by incorporating the temperature distribution caused by the viscous flow. For such high velocities, the temperature increase is significant. This outcome is partly due to the implemented adiabatic boundaries, which are actually not very realistic.

- Implement ROBIN boundary conditions for heat flux on all boundaries.
- Implement the code by using a material with a higher viscosity (search for properties of a polymer melt).

## 2.4 Thermoviscoelasticity

By considering the principles of thermodynamics in a EULERian frame, we have derived all of the necessary constitutive equations for a viscous fluid in the last section. For fluids we use an open system. In this section we will derive the constitutive equations for a deformable solid in a LAGRANGEan (reference) frame expressed in Cartesian coordinates. A material system is utilized for solids. As the reference frame we choose the initial frame, where the positions (coordinates) of particles are known. We start first by transforming the balance equations from the current frame to the initial frame. The following identities in a Cartesian coordinate system:

$$dv = J dV, \quad n_j da = (F^{-1})_{kj} J N_k dA, \quad (2.99)$$

have been derived in Sect. 1.4 for arbitrary coordinate systems. The balance equations of mass, momentum, and internal energy in the current frame for a material system:<sup>34</sup>

---

<sup>34</sup>A material system is a closed system possessing the same particles over time. In a material system no (mass) convection is allowed.

$$\begin{aligned}
\left( \int_{\mathcal{B}} \rho \, dv \right)^{\cdot} &= 0, \\
\left( \int_{\mathcal{B}} \rho v_i \, dv \right)^{\cdot} &= \int_{\partial \mathcal{B}} \sigma_{ji} n_j \, da + \int_{\mathcal{B}} \rho f_i \, dv, \\
\left( \int_{\mathcal{B}} \rho u \, dv \right)^{\cdot} &= - \int_{\partial \mathcal{B}} q_j n_j \, da + \int_{\mathcal{B}} \left( \rho r + \sigma_{ji} \frac{\partial v_i}{\partial x_j} \right) dv,
\end{aligned} \tag{2.100}$$

are transformed into the initial frame

$$\begin{aligned}
\left( \int_{\mathcal{B}_0} \rho J \, dV \right)^{\cdot} &= 0, \\
\left( \int_{\mathcal{B}_0} \rho v_i J \, dV \right)^{\cdot} &= \int_{\partial \mathcal{B}_0} \sigma_{ji} (\mathbf{F}^{-1})_{kj} J N_k \, dA + \int_{\mathcal{B}_0} \rho f_i J \, dV, \\
\left( \int_{\mathcal{B}_0} \rho u J \, dV \right)^{\cdot} &= - \int_{\partial \mathcal{B}_0} q_j (\mathbf{F}^{-1})_{kj} J N_k \, dA + \int_{\mathcal{B}_0} \left( \rho r + \sigma_{ji} \frac{\partial v_i}{\partial x_j} \right) J \, dV.
\end{aligned} \tag{2.101}$$

Initial frame is constant in time,  $(dV)^{\cdot} = 0$ , thus, the balance of mass in the initial frame reads

$$\rho_0 = \rho J. \tag{2.102}$$

The mass density in the initial state,  $\rho_0$ , is of course constant in time,  $\rho_0^{\cdot} = 0$ . By introducing fluxes in the initial frame:

$$P_{ki} = \sigma_{ji} (\mathbf{F}^{-1})_{kj} J, \quad Q_k = q_j (\mathbf{F}^{-1})_{kj} J, \tag{2.103}$$

and inserting the mass balance into the momentum balance and internal energy balance, we acquire

$$\begin{aligned}
\int_{\mathcal{B}_0} \rho_0 v_i^{\cdot} \, dV &= \int_{\partial \mathcal{B}_0} P_{ki} \, dA + \int_{\mathcal{B}_0} \rho f_i J \, dV, \\
\int_{\mathcal{B}_0} \rho_0 u^{\cdot} \, dV &= - \int_{\partial \mathcal{B}_0} Q_k N_k \, dA + \int_{\mathcal{B}_0} \left( \rho_0 r + J \sigma_{ji} \frac{\partial v_i}{\partial x_j} \right) dV.
\end{aligned} \tag{2.104}$$

After utilizing GAUSS's law on the boundary integrals, we write the balance equations in their local forms:

$$\rho_0 v_i^{\cdot} - \frac{\partial P_{ki}}{\partial X_k} - \rho_0 f_i = 0, \quad \rho_0 u^{\cdot} + \frac{\partial Q_k}{\partial X_k} - \rho_0 r = J \sigma_{ji} \frac{\partial v_i}{\partial x_j}. \tag{2.105}$$

We have written the production terms on the right-hand side. Since the formulation is in the initial frame, the partial derivative with respect to  $x_i$  needs to be reformulated as a differentiation in  $X_i$ . The velocity gradient in the current frame reads



$$\frac{\partial v_i}{\partial x_j} = \frac{\partial v_i}{\partial X_k} \frac{\partial X_k}{\partial x_j} = \frac{\partial v_i}{\partial X_k} (\mathbf{F}^{-1})_{kj} , \quad (2.106)$$

hence, we obtain

$$J \sigma_{ji} \frac{\partial v_i}{\partial x_j} = J \sigma_{ji} \frac{\partial v_i}{\partial X_k} (\mathbf{F}^{-1})_{kj} = P_{ki} \frac{\partial v_i}{\partial X_k} . \quad (2.107)$$

The second PIOLA–KIRCHHOFF stress tensor:

$$S_{ij} = (\mathbf{F}^{-1})_{jk} P_{ik} = (\mathbf{F}^{-1})_{jk} \sigma_{lk} (\mathbf{F}^{-1})_{il} J , \quad (2.108)$$

is more beneficial by obtaining constitutive equations. From the latter the nominal stress becomes

$$P_{ij} = F_{jl} S_{il} . \quad (2.109)$$

We further rewrite the production term. By starting with the right CAUCHY–GREEN deformation tensor,  $C_{ij} = F_{ki} F_{kj}$ , and its corresponding GREEN–LAGRANGE strain tensor,  $2E_{ij} = (C_{ij} - \delta_{ij})$ , we obtain

$$\begin{aligned} C_{ij} &= F_{ki} F_{kj} = F_{kj} F_{ki} = C_{ji} , \\ 2\dot{E}_{ij} &= \dot{C}_{ij} = 2F_{k(i} \dot{F}_{k j)} . \end{aligned} \quad (2.110)$$

In the initial frame we have the following identity:

$$\frac{\partial v_j}{\partial X_i} = \frac{\partial^2 x_j}{\partial X_i \partial t} = \frac{\partial^2 x_j}{\partial t \partial X_i} = F_{ji}^{\bullet} , \quad (2.111)$$

since  $x_i = x_i(t, X_j)$ . By using the aforementioned relations we acquire the following version of the production term:

$$P_{ij} \frac{\partial v_j}{\partial X_i} = F_{jl} S_{il} \frac{\partial v_j}{\partial X_i} = F_{jl} S_{il} F_{ji}^{\bullet} = F_{j(l} S_{il} F_{ji)}^{\bullet} = S_{il} \dot{E}_{il} , \quad (2.112)$$

for a symmetric stress tensor,  $S_{ij} = S_{ji}$ . In case of non-polar materials, the CAUCHY stress tensor is symmetric, leading to the symmetric second PIOLA–KIRCHHOFF stress tensor given in Eq. (2.108). For non-polar materials the balance of internal energy in the initial frame reads

$$\rho_0 \dot{u} + \frac{\partial Q_k}{\partial X_k} - \rho_0 r = S_{ij} \dot{E}_{ij} . \quad (2.113)$$

At equilibrium the balance of internal energy is

$$\rho_0 \dot{u} - \rho_0 T \dot{\eta} = S_{ij} \dot{E}_{ij} , \quad (2.114)$$

since the internal energy is fully recoverable and the stress tensor is decomposed into an elastic (reversible) term,  ${}^eS_{ij}$ , and into a dissipative (irreversible) term,  ${}^dS_{ij}$ , such that

$$S_{ij} = {}^eS_{ij} + {}^dS_{ij} . \quad (2.115)$$

We need constitutive equations for the specific entropy,  $\eta$ , for the heat flux,  $Q_i$ , and for the elastic and dissipative stress tensors,  ${}^eS_{ij}$ ,  ${}^dS_{ij}$ . By using the 1st law of thermodynamics we can rewrite the rate of internal energy as a differential form:

$$du = T d\eta + {}^eS_{ij} v dE_{ij} , \quad (2.116)$$

where the specific volume,  $v = 1/\rho_0$ , is a known quantity. The latter differential form is often introduced as GIBBS's equation.<sup>35</sup>

In Eq.(2.116) the internal energy is given as a function of  $\eta$  and  $E_{ij}$ . Having a function of the strains is adequate since the strains are given by the primitive variables (displacement). However, we have just introduced a variable called entropy,  $\eta$ , we lack a definition for it. We simply want to exchange the dependency from entropy to the temperature, which is one of the primitive variables. We transform<sup>36</sup> the differential form in Eq.(2.116) by introducing a free energy:

$$\psi = u - T\eta , \quad (2.117)$$

into the following form:

$$d\psi = du - \eta dT - T d\eta = -\eta dT + {}^eS_{ij} v dE_{ij} . \quad (2.118)$$

This differential form implies an energy depending on the temperature and strain,

$$\psi = \psi(T, E_{ij}) , \quad (2.119)$$

such that

$$-\eta = \frac{\partial \psi}{\partial T} , \quad {}^eS_{ij} v = \frac{\partial \psi}{\partial E_{ij}} . \quad (2.120)$$

The temperature and strain are called the primary or state variables. Since the energy depends on the primary variables, its derivatives depend on the same set of variables, too. So the derived, dual, or conjugate variables,  $\eta$ ,  ${}^eS_{ij}$ , depend on the primary variables

$$\begin{aligned} d\eta &= A dT + \bar{p}_{ij} dE_{ij} , \\ d({}^eS_{ij} v) &= p_{ij} dT + C_{ijkl} dE_{kl} . \end{aligned} \quad (2.121)$$

<sup>35</sup>For an alternative derivation of GIBBS's equation we refer to [12, Chap.8].

<sup>36</sup>Mathematicians call this transformation a LEGENDRE transformation named after Adrien-Marie Legendre.

We can readily apply the MAXWELL symmetry condition (reciprocal relation):

$$\bar{p}_{ij} = \frac{\partial \eta}{\partial E_{ij}} = -\frac{\partial^2 \psi}{\partial E_{ij} \partial T} = -\frac{\partial^2 \psi}{\partial T \partial E_{ij}} = -\frac{\partial {}^c S_{ij} v}{\partial T} = -v \frac{\partial {}^c S_{ij}}{\partial T} = -v p_{ij} . \quad (2.122)$$

The specific volume is a given function in space for heterogeneous materials or a constant value for homogeneous materials. It is coupled to the temperature through constitutive equations, however, it is independent on  $T$  so we have taken it out in the differentiation with respect to the temperature. The dual variables read

$$\begin{aligned} d\eta &= A dT - p_{ij} v dE_{ij} , \\ d {}^c S_{ij} &= p_{ij} dT + C_{ijkl} dE_{kl} . \end{aligned} \quad (2.123)$$

As in the previous section  $A = c/T$ , where the specific heat capacity,  $c$ , is measured by varying the temperature and recording the change of heat by fixed strains,  $dE_{ij} = 0$ . In other words, all of the boundaries are clamped and the temperature is varied. The *stiffness tensor*  $C_{ijkl}$  is measured on a constant temperature,  $dT = 0$ , by varying the strains  $dE_{ij}$  and recording the stress changes  $d {}^c S_{ij}$ . Since  $C_{ijkl}$  consists of many coefficients, we also need to establish various measurements. One of such measurements is the prominent tensile test. Throughout the experiment, the temperature is fixed such that the components of  $C_{ijkl}$  are valid for a specific temperature. One needs to redo the experiments in different temperatures for determining components as a function in  $T$ . The thermal pressure  $p_{ij}$  is the pressure occurring due to temperature variation by fixed strains,  $dE_{ij} = 0$ . The body tries to expand or shrink and applies a pressure on the clamped boundaries holding the strains fixed.

The values for the thermal pressure are difficult to find in the literature. Therefore, we introduce the coefficients of thermal expansion,  $\alpha_{ij}$ , which are measured by varying the temperature and measuring the strain change

$$dE_{ij} = \alpha_{ij} dT , \quad (2.124)$$

for a specific stress. Since such a measurement is realized by fixed stress,  $d {}^c S_{ij} = 0$ , we can observe from Eq. (2.123)<sub>2</sub>

$$\begin{aligned} 0 &= p_{ij} dT + C_{ijkl} dE_{kl} , \quad p_{ij} dT = -C_{ijkl} \alpha_{kl} dT \\ \Rightarrow p_{ij} &= -C_{ijkl} \alpha_{kl} . \end{aligned} \quad (2.125)$$

Now, the dual variables become

$$\begin{aligned} d\eta &= \frac{c}{T} dT + C_{ijkl} \alpha_{kl} v dE_{ij} , \\ d {}^c S_{ij} &= -C_{ijkl} \alpha_{kl} dT + C_{ijkl} dE_{kl} . \end{aligned} \quad (2.126)$$

For non-polar materials the stress tensor is symmetric, we assume that the elastic part is also symmetric,  ${}^c S_{ij} = {}^c S_{ji}$ . We restrict the formalism for linear materials such that

the stiffness tensor,  $C_{ijkl}$ , the coefficients of thermal expansion,  $\alpha_{ij}$ , and the specific heat capacity,  $c$ , are constants and we acquire the dual variables by integrating from the reference state,  $T = T_{\text{ref.}}$ ,  $E_{ij} = 0$ , to the current state

$$\begin{aligned}\eta &= c \ln \left( \frac{T}{T_{\text{ref.}}} \right) + C_{ijkl} \alpha_{kl} v E_{ij} , \\ {}^e S_{ij} &= -C_{ijkl} \alpha_{kl} (T - T_{\text{ref.}}) + C_{ijkl} E_{kl} .\end{aligned}\quad (2.127)$$

Often, thermal strains are introduced

$${}^{\text{th}} E_{kl} = \alpha_{kl} (T - T_{\text{ref.}}) , \quad (2.128)$$

such that the elastic stress is written as

$${}^e S_{ij} = C_{ijkl} (E_{kl} - {}^{\text{th}} E_{kl}) . \quad (2.129)$$

Finally, we have determined the GIBBS equation:

$$\begin{aligned}du &= T d\eta + {}^e S_{ij} v dE_{ij} = \\ &= c dT + T C_{ijkl} \alpha_{kl} v dE_{ij} + C_{ijkl} (E_{kl} - \alpha_{kl} (T - T_{\text{ref.}})) v dE_{ij} = \\ &= c dT + v C_{ijkl} \alpha_{kl} T_{\text{ref.}} dE_{ij} + v C_{ijkl} E_{kl} dE_{ij} ,\end{aligned}\quad (2.130)$$

solely depending on the temperature and displacement (over the known relation between strain and displacement). For a linear thermoelastic isotropic body, the material parameters reduce to

$$C_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu \delta_{ik} \delta_{jl} + \mu \delta_{il} \delta_{jk} , \quad \alpha_{ij} = \alpha \delta_{ij} , \quad (2.131)$$

thus, the internal energy rate reads

$$u^* = c T^* + v(3\lambda + 2\mu) \alpha T_{\text{ref.}} E_{ii}^* + v(\lambda \delta_{ij} E_{kk} + 2\mu E_{ij}) E_{ij}^* . \quad (2.132)$$

For deriving the heat flux,  $Q_i$ , and the dissipative stress,  ${}^d S_{ij}$ , we start with Eq. (2.116) in the following form:

$$\rho_0 u^* = \rho_0 T \eta^* + (S_{ij} - {}^d S_{ij}) E_{ij}^* , \quad (2.133)$$

and insert it into Eq. (2.113) in order to acquire the balance of entropy in the reference frame:

$$\begin{aligned}\rho_0 \eta^* + \frac{\partial}{\partial X_i} \left( \frac{Q_i}{T} \right) - \rho_0 \frac{r}{T} &= \frac{1}{T} {}^d S_{ij} E_{ij}^* + Q_i \frac{\partial}{\partial X_i} \left( \frac{1}{T} \right) , \\ \rho_0 \eta^* + \frac{\partial}{\partial X_i} \left( \frac{Q_i}{T} \right) - \rho_0 \frac{r}{T} &= \frac{1}{T} {}^d S_{ij} E_{ij}^* - \frac{1}{T^2} Q_i \frac{\partial T}{\partial X_i} .\end{aligned}\quad (2.134)$$

The right-hand side is the production term and it has to be positive according to the 2nd law of thermodynamics:

$$\Sigma = \frac{1}{T} {}^dS_{ij} E_{ij}^{\bullet} - \frac{1}{T^2} Q_i G_i \geq 0 , \quad (2.135)$$

where again for simplicity we have used the following notation:

$$G_i = \frac{\partial T}{\partial X_i} . \quad (2.136)$$

The stress tensor is symmetric for non-polar materials; we have employed a symmetric reversible term, the dissipative term has to be symmetric, too. A symmetric tensor of rank two can be decomposed into a spherical (volumetric) term and a deviatoric term. Multiplication of a volumetric with a deviatoric term vanishes such that the entropy production reads

$$\Sigma = \frac{1}{3T} {}^dS_{ii} E_{jj}^{\bullet} + \frac{1}{T} {}^dS_{[ij]} E_{[ij]}^{\bullet} - \frac{1}{T^2} Q_i G_i \geq 0 . \quad (2.137)$$

By introducing thermodynamical fluxes:

$$\mathcal{F}^{\alpha} = \left\{ Q_i , \quad {}^dS_{ii} , \quad {}^dS_{[ij]} \right\} , \quad (2.138)$$

and thermodynamical forces:

$$\mathcal{K}^{\alpha} = \left\{ -\frac{G_i}{T^2} , \quad \frac{1}{3T} E_{jj}^{\bullet} , \quad \frac{1}{T} E_{[ij]}^{\bullet} \right\} , \quad (2.139)$$

we can rewrite the 2nd law:

$$\Sigma = \mathcal{F}^{\alpha} \cdot \mathcal{K}^{\alpha} , \quad \alpha = 1, 2, 3 . \quad (2.140)$$

All of thermodynamical fluxes are of different type (tensors of different ranks). According to the CURIE principle thermodynamical fluxes depend only on their corresponding thermodynamical forces of the same rank such that we obtain

$$\mathcal{F}^1 = \mathcal{F}^1(\mathcal{K}^1) , \quad \mathcal{F}^2 = \mathcal{F}^2(\mathcal{K}^2) , \quad \mathcal{F}^3 = \mathcal{F}^3(\mathcal{K}^3) . \quad (2.141)$$

We can readily propose linear constitutive equations:

$${}^dS_{ii} = \mu_1 E_{ii}^{\bullet} , \quad {}^dS_{[ij]} = \mu_2 E_{[ij]}^{\bullet} , \quad Q_i = -\kappa G_i , \quad (2.142)$$

where  $\mu_1$ ,  $\mu_2$ , and  $\kappa$  are all positive constants such that  $\Sigma \geq 0$ . The viscous part of the stress reads

$$\begin{aligned}
{}^dS_{ij} &= \frac{1}{3} {}^dS_{kk} \delta_{ij} + {}^dS_{|ij|} = \frac{\mu_1}{3} E_{kk}^* \delta_{ij} + \mu_2 \left( E_{ij}^* - \frac{1}{3} E_{kk}^* \delta_{ij} \right) = \\
&= \frac{\mu_1 - \mu_2}{3} E_{kk}^* \delta_{ij} + \mu_2 E_{ij}^* .
\end{aligned} \tag{2.143}$$

Then by using the obtained elastic stress we acquire a linear thermoviscoelastic material model:

$$S_{ij} = C_{ijkl} (E_{kl} - \alpha_{ij} (T - T_{\text{ref}})) + \frac{\mu_1 - \mu_2}{3} E_{kk}^* \delta_{ij} + \mu_2 E_{ij}^* . \tag{2.144}$$

For a constant  $\kappa$  the constitutive equation:

$$Q_i = -\kappa \frac{\partial T}{\partial X_i} , \tag{2.145}$$

is called FOURIER's law in the LAGRANGEan frame.

In order to compute the displacement and temperature in a linear thermoviscoelastic body, we employ the balance of momentum and the balance of entropy:

$$\begin{aligned}
\rho_0 u_i'' - \frac{\partial P_{ji}}{\partial X_j} - \rho_0 f_i &= 0 , \\
\rho_0 \eta' + \frac{\partial}{\partial X_i} \left( \frac{Q_i}{T} \right) - \rho_0 \frac{r}{T} &= \frac{1}{T} {}^dS_{ij} E_{ij}^* - \frac{1}{T^2} Q_i \frac{\partial T}{\partial X_i} .
\end{aligned} \tag{2.146}$$

The primitive variables are displacement,  $u_i$ , and temperature,  $T$ . Hence we multiply the balance of linear momentum with  $\delta u_i$  and integrate over the continuum body for generating a form in the unit of energy. By multiplying the balance of entropy with  $\delta T$  and integrating over the body, we obtain a form in the unit of power. After discretizing in time, we can multiply the equation with  $\Delta t$  in order to acquire both forms in the unit of energy. Having forms in the same unit, we can sum them up. Furthermore, we apply GAUSS's law in order to weaken the forms and acquire

$$\begin{aligned}
\text{Form} &= \int_{\mathcal{B}_0} \left( \rho_0 \frac{u_i - 2u_i^0 + u_i^{00}}{\Delta t \Delta t} \delta u_i + P_{ji} \delta u_{i,j} - \rho_0 f_i \delta u_i + \right. \\
&+ \frac{\rho_0}{T} (\eta - \eta^0) \delta T - \Delta t \frac{1}{T} Q_i \delta T_{,i} - \Delta t \frac{\rho_0 r}{T} \delta T - \frac{1}{T} {}^dS_{ij} (E_{ij} - E_{ij}^0) \delta T + \\
&\left. + \Delta t \frac{1}{T^2} Q_i T_{,i} \delta T \right) dV + \int_{\partial \mathcal{B}_0} \left( \Delta t \frac{1}{T} \hat{Q} \delta T - \hat{t}_i \delta u_i \right) dA ,
\end{aligned} \tag{2.147}$$

where the comma notation has been used for a partial space derivative in  $X_i$ . We summarize the necessary relations:

$$F_{ij} = \frac{\partial u_i}{\partial X_j} + \delta_{ij} , \quad C_{ij} = F_{ki} F_{kj} , \quad E_{ij} = \frac{1}{2} (C_{ij} - \delta_{ij}) ,$$

$$\begin{aligned}
{}^e S_{ij} &= -C_{ijkl} \alpha_{kl} (T - T_{\text{ref.}}) + C_{ijkl} E_{kl} , \quad \eta = c \ln \left( \frac{T}{T_{\text{ref.}}} \right) + C_{ijkl} \alpha_{kl} v E_{ij} , \\
{}^d S_{ij} &= \frac{\mu_1 - \mu_2}{3} E_{kk}^* \delta_{ij} + \mu_2 E_{ij}^* , \quad Q_i = -\kappa \frac{\partial T}{\partial X_i} , \\
S_{ij} &= {}^e S_{ij} + {}^d S_{ij} , \quad P_{ij} = F_{jl} S_{il} .
\end{aligned} \tag{2.148}$$

For an isotropic body the stiffness tensor and coefficients of thermal expansion are

$$C_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu \delta_{ik} \delta_{jl} + \mu \delta_{il} \delta_{jk} , \quad \alpha_{ij} = \alpha \delta_{ij} . \tag{2.149}$$

Therefore, in case of an isotropic body we need seven material parameters, viz.,  $\lambda$ ,  $\mu$ ,  $\alpha$ ,  $\mu_1$ ,  $\mu_2$ ,  $\kappa$ , and  $c$ .

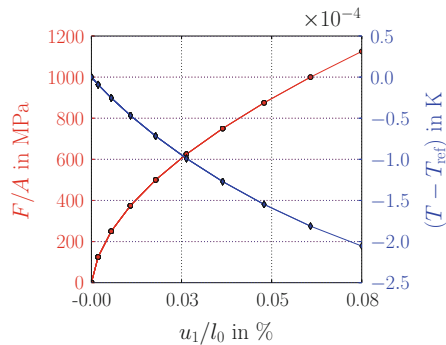
In a tensile testing we normally assume that the process is isothermal. By computing the reality where heat is produced due to the entropy production, we can validate this engineering assumption. The geometry is a beam along  $X_1$  and we use a ROBIN boundary condition for the heat flux over all boundaries:

$$\hat{Q} = h(T - T_{\text{amb}}) . \tag{2.150}$$

On the left side we hold the beam fixed and on the right side we pull with the force given by the traction vector  $\hat{t}_i = (800t, 0, 0)$  MPa linearly in time,  $t$ . The traction (force per area) is the controlled parameter, i.e., the machine is steered by the force. The tip displacement is measured, it is an observed quantity. Conveniently we plot stress versus strain, where the stress (on the right tip) is the (axial) traction and the (normal axial) strain,  $E_{11}$ , is the displacement divided by the initial length. The traction vector,  $\hat{t} = N_j P_{ji}$ , is given by the nominal or engineering stress,  $P_{ji}$ . The strain,  $E_{ij}$ , is called the engineering strain; we have introduced it as the GREEN-LAGRANGE strain measure.

We apply a mechanical load and measure the temperature in the middle of the beam as well as the stress and strain on the tip. In Fig. 2.7 the temperature change can be seen, it is clearly negligible. This is good news, because we measure the elasticity

**Fig. 2.7** Tensile testing and temperature change due to the deformation



components,  $\lambda$ ,  $\nu$  for isotropic materials by using a tensile testing and assume that the temperature remains constant. The code for the computation is given below.

```

1  """Computational Reality 13, thermoviscoelasticity"""
2  __author__ = "B. Emek Abali"
3  __license__ = "GNU GPL Version 3.0 or later"
4  #This code underlies the GNU General Public License ,
5      ↪ http://www.gnu.org/licenses/gpl-3.0.en.html
6
7  from fenics import *
8  import numpy
9  set_log_level(ERROR)
10 #units: mm, 1000 kg=ton, s, MPa, mJ, K
11 delta = Identity(3)
12 f = Constant((0.0, 0.0, -9810.))
13 r = 0.
14 Tref = 293.15 #in K
15 Tamb=Tref
16
17 # Material data of P265GH (St 45.8) from VDI Waermeatlas , at
18     ↪ 293.15 K
19 rho0 = 7850.0E-9 #in kg / mm^3
20 kappa = 57.0 #in mJ / (s mm K)
21 capacity = 430.0E6 #in mJ / (ton K)
22 alpha = 12.2E-6 #in 1/K at 373.15 K
23 EModul = 211.E+3 #in MPa
24 nu = 0.28
25 h = 10.E-3 #in mJ / (s m^2 K)
26 mu1 = 1.E+6 #in MPa / s
27 mu2 = 3.E+6 #in MPa / s
28
29 tMax = 5.0
30 Dt = 0.5
31 t = 0.0
32
33 xMin, xMax, xElements = 0.0, 100.0, 10
34 yMin, yMax, yElements = -10., +10., 10
35 zMin, zMax, zElements = +10., -10., 10
36 mesh = BoxMesh(Point(xMin,yMin,zMin), Point(xMax,yMax,zMax),
37     ↪ xElements,yElements,zElements)
38 N = FacetNormal(mesh)
39 length = abs(xMax-xMin)
40
41 T_Space = FunctionSpace(mesh, 'P', 1)
42 u_Space = VectorFunctionSpace(mesh, 'P', 1)
43 Space = MixedFunctionSpace([T_Space, u_Space])
44
45 cells = CellFunction('size_t', mesh)
46 facets = FacetFunction('size_t', mesh)
47 dA = Measure('ds', domain=mesh, subdomain_data=facets)
48 dV = Measure('dx', domain=mesh, subdomain_data=cells)
49
50 left = CompiledSubDomain('near(x[0],1) && on_boundary',l=xMin
51     ↪ )
52 right = CompiledSubDomain('near(x[0],1) && on_boundary',l=
53     ↪ xMax)

```



```

49 back = CompiledSubDomain('near(x[1],1) && on_boundary',l=yMin
    ↪ )
50 front = CompiledSubDomain('near(x[1],1) && on_boundary',l=
    ↪ yMax)
51 bottom = CompiledSubDomain('near(x[2],1) && on_boundary',l=
    ↪ zMin)
52 top = CompiledSubDomain('near(x[2],1) && on_boundary',l=zMax)
53
54 facets.set_all(0)
55 right.mark(facets, 1)
56 tHat = Expression(('A*t','0.','0.'), A=250., t=0.)
57 bc = [DirichletBC(Space.sub(1), Constant((0.0, 0.0, 0.0)),
    ↪ left),\
58 DirichletBC(Space.sub(1).sub(1), Constant(0.0), right),\
59 DirichletBC(Space.sub(1).sub(2), Constant(0.0), right)]
60
61 dunkn = TrialFunction(Space)
62 test = TestFunction(Space)
63 delT, delu = split(test)
64
65 unkn = Function(Space)
66 unkn0 = Function(Space)
67 unkn00 = Function(Space)
68
69 unkn_init = Expression(('T_ini','0','0','0'),T_ini=Tref)
70 unkn = interpolate(unkn_init,Space)
71 unkn0.assign(unkn)
72 unkn00.assign(unkn0)
73
74 T, u = split(unkn)
75 T0, u0 = split(unkn0)
76 T00, u00 = split(unkn0)
77
78 i, j, k, l = indices(4)
79 delta = Identity(3)
80 F = as_tensor(u[i].dx(j)+delta[i,j], (i,j))
81 F0 = as_tensor(u0[i].dx(j)+delta[i,j], (i,j))
82 C = as_tensor(F[k,i]*F[k,j], (i,j))
83 C0 = as_tensor(F0[k,i]*F0[k,j], (i,j))
84 E = as_tensor(1./2.*(C[i,j]-delta[i,j]), (i,j))
85 E0 = as_tensor(1./2.*(C0[i,j]-delta[i,j]), (i,j))
86 lambada = EModul * nu / (1. + nu) / (1. - 2. * nu)
87 mu = 0.5 * EModul / (1. + nu)
88 C_ = as_tensor(lambada*delta[i,j]*delta[k,l]+mu*delta[i,k]*
    ↪ delta[j,l]+mu*delta[i,l]*delta[j,k], (i,j,k,l))
89 alp = as_tensor(alpha*delta[i,j], (i,j))
90 eStress = as_tensor(-C_[i,j,k,l]*alp[k,l]*(T-Tref) + C_[i,j,k
    ↪ ,l]*E[k,l], (i,j))
91 dStress = as_tensor((mu1-mu2)/3.*(E-E0)[k,k]/Dt*delta[i,j] +
    ↪ mu2*(E-E0)[i,j]/Dt, (i,j))
92 S = as_tensor(eStress[i,j]+dStress[i,j], (i,j))
93 P = as_tensor(F[j,l]*S[i,l], (i,j))
94 eta = as_tensor(capacity*ln(T/Tref) + C_[i,j,k,l]*alp[k,l]

```

```

    ↪ ]*1./rho0*E[i,j] , ( ))
95 eta0 = as_tensor( capacity*ln(T0/Tref) + C_[i,j,k,l]*alp[k,l
    ↪ ]*1./rho0*E0[i,j] , ( ))
96 Q = as_tensor(-kappa*T.dx(i), (i,))
97
98 Form = (rho0*(u-2.*u0+u00)[i]/Dt/Dt*delu[i] + P[j,i]*delu[i].
    ↪ dx(j) - rho0*f[i]*delu[i] + rho0/T*(eta-eta0)*delT -
    ↪ Dt/T*Q[i]*delT.dx(i) - Dt*rho0*r/T*delT - 1./T*dStress
    ↪ [i,j]*(E-E0)[i,j]*delT + Dt/T**2*Q[i]*T.dx(i)*delT ) *
    ↪ dV + Dt/T*h*(T-Tamb)*delT*(dA(0)+dA(1)) - tHat[i]*delu
    ↪ [i]*dA(1)
99
100 Gain = derivative(Form, unkn, dunkn)
101
102 import matplotlib as mpl
103 mpl.use('Agg')
104 import matplotlib.pyplot as pylab
105 pylab.rc('text', usetex=True)
106 pylab.rc('font', family='serif', serif='cm', size=30)
107 pylab.rc('legend', fontsize=30)
108 pylab.rc(('xtick.major', 'ytick.major'), pad=15)
109 pylab.subplots_adjust(top=0.90)
110 pylab.subplots_adjust(bottom=0.17)
111 pylab.subplots_adjust(left=0.20)
112 pylab.subplots_adjust(right=0.8)
113
114 fig = pylab.figure(1, figsize=(14,10))
115 ax1 = fig.add_subplot(111)
116 ax1.grid(True, axis='x')
117 ax1.set_xlabel('$u_1/l_0$ in $\\%$')
118 ax1.set_ylabel('$F/A$ in MPa', color='r')
119 ax1.tick_params(axis='y', colors='r')
120 ax1.grid(True, axis='y', color='r')
121 ax2 = ax1.twinx()
122 ax2.set_ylabel('$ (T-T_{\\mathrm{ref}}) $ in K', color='b')
123 ax2.tick_params(axis='y', colors='b')
124 ax2.grid(True, axis='y', color='b')
125 ax2.ticklabel_format(style='sci', axis='y', scilimits=(-2,2))
126
127 pwd='/calcul/CR13/'
128 file_u = File(pwd+'displ.pvd')
129 file_T = File(pwd+'temp.pvd')
130 strain, stress, temp = [], [], []
131
132 while t < tMax:
133     print 'time: ', t
134     tHat.t = t
135     solve(Form==0, unkn, bc, J=Gain, \
136           solver_parameters={"newton_solver":{"linear_solver":
    ↪ "mumps", "relative_tolerance": 1e-5} }, \
137           form_compiler_parameters={"cpp_optimize": True, "
    ↪ representation": "quadrature", "
    ↪ quadrature_degree": 2} )

```

```

138     file_T << (unkn.split()[0], t)
139     file_u << (unkn.split()[1], t)
140
141
142     strain.append(unkn.split()[1](xMax,0.,0.)[0]/length*100.)
143     stress.append(tHat(xMax,0.,0.)[0])
144     temp.append(unkn.split()[0](xMax/2.,0.,0.)-Tref)
145     ax1.plot(strain, stress, 'o-', color='r')
146     ticks = numpy.linspace(numpy.array(strain).min(), numpy.
    ↪ array(strain).max(), 4)
147     ax1.set_xticks(ticks)
148     ax1.set_xticklabels(['%1.2f' % i_ticks for i_ticks in
    ↪ ticks])
149     ax2.plot(strain, temp, 'd-', color='b')
150     fig.savefig(pwd+'CompReal13_tensiletest.pdf')
151     unkn00.assign(unkn0)
152     unkn0.assign(unkn)
153     t += Dt

```

## To-do

We have employed the 1st and 2nd laws of thermodynamics, obtained constitutive (material) equations, and computed a coupled thermoviscoelastic problem. In a tensile testing the temperature change is negligible.

- Which term is responsible for the temperature change?
- Implement the code for a thermoelastic problem by setting  $\mu_1 = \mu_2 = 0$ , thus,  $^dS_{ij} = 0$ . Solve a laser welding application as in Sect. 2.1 and determine the deformations.
- Try to implement a bimetal and apply a thermal loading. Guess and inspect the occurring deformation.

## 2.5 Thermoplasticity

We have seen a methodology for deriving material equations from thermodynamical restrictions called the 1st and 2nd laws. Unfortunately, it is rather difficult to utilize this procedure for plasticity. There are numerous different suggestions but none of them is accepted by all communities. We present here a more or less widely accepted methodology—it is used in many commercial codes.<sup>37</sup> Within its derivation there occur many assumptions, hence, the method fails to rely on a sound thermodynamical formulation. From a pragmatic point of view, however, it works!

<sup>37</sup>We present a monolithic approach; however, many commercial codes still use a staggered schema. A staggered schema solves the field equations subsequently such that the results from each solution are used in the subsequent field equation. Such an approach is used in Sect. 1.9, where the balance equations are solved subsequently.

Balance equations of mass, momentum, and internal energy has been motivated in the last section. By neglecting big deformations with  $F_{ij} \approx \delta_{ij}$  they result in<sup>38</sup>

$$\rho_0 = \rho, \quad \rho u_i'' - \sigma_{ji,j} - \rho f_i = 0, \quad \rho u' + q_{i,i} - \rho_0 r = \sigma_{ij} \dot{\varepsilon}_{ij}. \quad (2.151)$$

We axiomatically assume that the production of internal energy consists of two parts: a reversible part including elastic and thermal strain, and an irreversible part due to the plastic strain:

$$\sigma_{ij} \dot{\varepsilon}_{ij} = \sigma_{ij} (\dot{\varepsilon}_{ij}^e + \dot{\varepsilon}_{ij}^p). \quad (2.152)$$

This assumption is by no means more restrictive than the assumption of decomposing the stress in the last sections. Again we assume that the reversible part remains at equilibrium such that we obtain from the balance of internal energy at equilibrium

$$\rho u' - T \eta' = \sigma_{ij} \dot{\varepsilon}_{ij}^e. \quad (2.153)$$

In this setting, GIBBS's equation reads

$$du = T d\eta + v \sigma_{ij} d\varepsilon_{ij}, \quad (2.154)$$

again with  $v = 1/\rho$  as a known quantity. The latter equation allows us to generate the material equations for the dual variables. However, this time  $\varepsilon_{ij}$  is not known. Hence, the chosen primary variables should be  $\{T, \sigma_{ij}\}$ . We use the same mathematical trick in order to transform the energy into a quantity depending on the primary variables by introducing the so-called specific GIBBS free energy:

$$g = u - T \eta - v \sigma_{ij} \varepsilon_{ij}, \quad (2.155)$$

with its differential:

$$dg = du - \eta dT - T d\eta - v \varepsilon_{ij} d\sigma_{ij} - v \sigma_{ij} d\varepsilon_{ij}, \quad (2.156)$$

and by inserting the latter in Eq. (2.154)

$$dg = -\eta dT - v \varepsilon_{ij} d\sigma_{ij}. \quad (2.157)$$

The assumption that the free energy possesses a first integral:

$$g = \int dg, \quad (2.158)$$

---

<sup>38</sup>We also use a linear strain measure,  $\varepsilon_{ij}$ , instead of  $E_{ij}$  in order to attain an identical formulation for plasticity as given in the literature.

is a weakness in the formulation. We take the latter as granted; under this assumption it is obvious that we can write

$$g = g(T, \sigma_{ij}) , \quad -\eta = \frac{\partial g}{\partial T} , \quad -v \varepsilon_{ij} = \frac{\partial g}{\partial \sigma_{ij}} . \quad (2.159)$$

GIBBS's free energy depends on the primary variables, viz., on  $T$  and  $\sigma_{ij}$ . The conjugated or dual variables,  $\eta$  and  $\varepsilon_{ij}$ , depend on the same set of arguments as the energy,

$$\begin{aligned} d\eta &= \frac{c}{T} dT + \bar{\alpha}_{ij} d\sigma_{ij} , \\ d\varepsilon_{ij} &= \alpha_{ij} dT + S_{ijkl} d\sigma_{kl} . \end{aligned} \quad (2.160)$$

Again we employ the MAXWELL relation:

$$\bar{\alpha}_{ij} = \frac{\partial \eta}{\partial \sigma_{ij}} = -\frac{\partial^2 g}{\partial \sigma_{ij} \partial T} = -\frac{\partial^2 g}{\partial T \partial \sigma_{ij}} = v \frac{\partial \varepsilon_{ij}}{\partial T} = v \alpha_{ij} , \quad (2.161)$$

since  $v = 1/\rho$  is a function of  $X_i$  but not of the temperature. For a *linear* material model the coefficients,  $c$ ,  $\alpha_{ij}$ ,  $S_{ijkl}$  are all constants. For a linear model we obtain the dual variables by integrating from the *ground* state  $\{T = T_{\text{ref.}}, \sigma_{ij} = 0\}$  without strain and entropy to the current state and obtain

$$\begin{aligned} \eta &= c \ln \left( \frac{T}{T_{\text{ref.}}} \right) + v \alpha_{ij} \sigma_{ij} , \\ \varepsilon_{ij} &= \alpha_{ij} (T - T_{\text{ref.}}) + S_{ijkl} \sigma_{kl} . \end{aligned} \quad (2.162)$$

The first term can be seen as thermal strains and the second term as elastic strains:

$$\begin{aligned} {}^{\text{th}}\varepsilon_{ij} &= \alpha_{ij} (T - T_{\text{ref.}}) , \quad {}^{\text{e}}\varepsilon_{ij} = S_{ijkl} \sigma_{kl} , \\ \varepsilon_{ij} &= {}^{\text{th}}\varepsilon_{ij} + {}^{\text{e}}\varepsilon_{ij} . \end{aligned} \quad (2.163)$$

Then the so-called HOOKE's law with DUHAMEL–NEUMANN supplemental term<sup>39</sup> for thermal strains can be deduced

$$\sigma_{kl} = C_{klij} ({}^{\text{e}}\varepsilon_{ij} - {}^{\text{th}}\varepsilon_{ij}) , \quad (2.164)$$

where the stiffness tensor,  $C_{ijkl}$ , is the inverse<sup>40</sup> of the compliance tensor,  $S_{ijkl}$ . We can even use the assumption already undertaken:

<sup>39</sup>It is named after Jean-Marie Constant Duhamel and Franz Ernst Neumann.

<sup>40</sup>For the inverse of a tensor of rank four we need an identity tensor of rank four. This method can be challenging. Instead of that, the inverse is found by using the VOIGT notation. For linear materials we can always rewrite the compliance tensor in the VOIGT notation, which is a  $6 \times 6$  matrix and its inverse is easy to determine. From the resulting  $6 \times 6$  matrix in the VOIGT notation, the stiffness tensor is obtained.

$$\varepsilon_{ij} = {}^e\varepsilon_{ij} + {}^p\varepsilon_{ij} , \quad (2.165)$$

in order to rewrite the material equation for stress:

$$\sigma_{ij} = C_{ijkl}(\varepsilon_{kl} - {}^p\varepsilon_{kl} - {}^{th}\varepsilon_{kl}) . \quad (2.166)$$

The rate of stress reads

$$\begin{aligned} \dot{\sigma}_{ij} &= C_{ijkl}(\dot{\varepsilon}_{kl} - \dot{{}^p\varepsilon}_{kl} - \dot{{}^{th}\varepsilon}_{kl}) , \\ \dot{\sigma}_{ij} &= C_{ijkl}(\dot{\varepsilon}_{kl} - \dot{{}^p\varepsilon}_{kl} - \alpha_{kl}T^*) , \end{aligned} \quad (2.167)$$

by using that  $C_{ijkl}$ ,  $\alpha_{ij}$ , as well as  $T_{\text{ref}}$  are constants. From the balance of internal energy in Eq. (2.151)<sub>3</sub> augmented by GIBBS's equation (2.154) we obtain

$$\begin{aligned} \rho T \dot{\eta} + \sigma_{ij} \dot{{}^e\varepsilon}_{ij} + q_{i,i} - \rho_0 r &= \sigma_{ij} \dot{{}^e\varepsilon}_{ij} , \\ \rho T \dot{\eta} + q_{i,i} - \rho_0 r &= \sigma_{ij} \dot{{}^p\varepsilon}_{ij} . \end{aligned} \quad (2.168)$$

Now by using the material equations for dual variables in Eq. (2.160) and in Eq. (2.166) we obtain the field equation for temperature:

$$\begin{aligned} \rho c T^* + T \alpha_{ij} \dot{\sigma}_{ij} + q_{i,i} - \rho_0 r &= \sigma_{ij} \dot{{}^p\varepsilon}_{ji} , \\ \rho c T^* + T \alpha_{ij} C_{ijkl}(\dot{\varepsilon}_{kl} - \dot{{}^p\varepsilon}_{kl} - \alpha_{kl}T^*) + q_{i,i} - \\ - \rho_0 r - C_{ijkl}(\varepsilon_{kl} - {}^p\varepsilon_{kl} - \alpha_{kl}(T - T_{\text{ref}})) \dot{{}^p\varepsilon}_{ji} &= 0 . \end{aligned} \quad (2.169)$$

The field equation for displacement is acquired from Eq. (2.151)<sub>2</sub> by augmenting Eq. (2.166) as follows

$$\begin{aligned} \rho u_i'' - \sigma_{ji,j} - \rho f_i &= 0 , \\ \rho u_i'' - C_{jikl}(\varepsilon_{kl} - {}^p\varepsilon_{kl} - \alpha_{kl}(T - T_{\text{ref}}))_{,j} - \rho f_i &= 0 . \end{aligned} \quad (2.170)$$

The field equations are nonlinear and coupled. We can solve them after having defined  $\dot{{}^p\varepsilon}_{ij}$ ,  $\dot{{}^e\varepsilon}_{ij}$ , and  $q_i$ .

Plasticity starts with the assumption that we can acquire the rate of plastic strain by using a *dissipation function*,  $\Phi$ , as follows

$$\dot{{}^p\varepsilon}_{ij} = \Lambda^* \frac{\partial \Phi}{\partial \sigma_{ij}} . \quad (2.171)$$

The associated plasticity proposes to use the flow potential,  $f$ , for the dissipation function,  $f \equiv \Phi$ . Modeling kinematic hardening has been discussed in Sect. 1.6.2, we use the same notation and skip the calculations undertaken there. The flow potential:

$$f = \frac{1}{2}(\sigma_{|ij|} - \beta_{ij})(\sigma_{|ij|} - \beta_{ij}) - \frac{1}{3}\sigma_Y^2, \quad (2.172)$$

results in

$$c\Lambda^* = \Gamma^*, \quad \Lambda^* = \frac{(\sigma_{|ij|} - \beta_{ij})\dot{\sigma}_{ij}}{\frac{2}{3}c\sigma_Y^2}. \quad (2.173)$$

With the help of Eq. (2.167) we write

$$\Lambda^* = \frac{(\sigma_{|ij|} - \beta_{ij})C_{ijkl}(\dot{\varepsilon}_{kl} - \dot{\varepsilon}_{kl}^{\text{th}})}{\frac{2}{3}c\sigma_Y^2}, \quad (2.174)$$

and by inserting the rate of plastic strain:

$$\dot{\varepsilon}_{kl}^{\text{p}} = \Lambda^* \frac{\partial f}{\partial \sigma_{kl}} = \Lambda^*(\sigma_{|kl|} - \beta_{kl}), \quad (2.175)$$

into the multiplier, we obtain

$$\begin{aligned} \Lambda^* \left( 1 + \frac{(\sigma_{|ij|} - \beta_{ij})C_{ijkl}(\sigma_{|kl|} - \beta_{kl})}{\frac{2}{3}c\sigma_Y^2} \right) &= \frac{(\sigma_{|ij|} - \beta_{ij})C_{ijkl}(\dot{\varepsilon}_{kl} - \dot{\varepsilon}_{kl}^{\text{th}})}{\frac{2}{3}c\sigma_Y^2}, \\ \Lambda^* &= \frac{(\sigma_{|ij|} - \beta_{ij})C_{ijkl}(\dot{\varepsilon}_{kl} - \dot{\varepsilon}_{kl}^{\text{th}})}{\frac{4}{9}H\sigma_Y^2 + (\sigma_{|ij|} - \beta_{ij})C_{ijkl}(\sigma_{|kl|} - \beta_{kl})}, \end{aligned} \quad (2.176)$$

where we have chosen  $c = 2/3H$  for an easier association of parameters.<sup>41</sup> By using the conditional parameter  $\langle \gamma \rangle$  from Eq. (1.216) we define the plastic strain rate:

$$\dot{\varepsilon}_{mn}^{\text{p}} = \langle \gamma \rangle \frac{(\sigma_{|ij|} - \beta_{ij})C_{ijkl}(\dot{\varepsilon}_{kl} - \dot{\varepsilon}_{kl}^{\text{th}})}{\frac{4}{9}H\sigma_Y^2 + (\sigma_{|ij|} - \beta_{ij})C_{ijkl}(\sigma_{|kl|} - \beta_{kl})}(\sigma_{|mn|} - \beta_{mn}). \quad (2.177)$$

This equation gives the evolution of the plastic strain, which is accumulated by the rate of plastic strain such that we can acquire it by integration,

$$\varepsilon_{ij}^{\text{p}} = \int \dot{\varepsilon}_{ij}^{\text{p}} dt. \quad (2.178)$$

<sup>41</sup>We use  $H$  for the plastic modulus instead of  $h$  as in Sect. 1.6 since we have started to use  $h$  for the convective heat transfer coefficient in the mixed boundary conditions for temperature.

For the heat flux we use FOURIER's law:

$$q_i = -\kappa T_{,i} . \quad (2.179)$$

Furthermore, we may test the validity of the evolution equation by employing the 2nd law of thermodynamics. By using Eq. (2.168)<sub>2</sub> we reformulate the balance of internal energy into the balance of entropy:

$$\begin{aligned} \rho T \dot{\eta} + q_{i,i} - \rho r &= \sigma_{ij} \dot{\varepsilon}_{ij}^p , \\ \rho \dot{\eta} + \left( \frac{q_i}{T} \right)_{,i} - \rho \frac{r}{T} &= -\frac{1}{T^2} q_i T_{,i} + \frac{1}{T} \sigma_{ij} \dot{\varepsilon}_{ij}^p , \end{aligned} \quad (2.180)$$

with the production term being positive:

$$\Sigma = -\frac{1}{T^2} q_i T_{,i} + \frac{1}{T} \sigma_{ij} \dot{\varepsilon}_{ij}^p \geq 0 . \quad (2.181)$$

Obviously, it is challenging to prove that the rate of plastic strain in Eq. (2.177) is thermodynamically admissible. Therefore, we rather have to “believe in” the formulation than to derive in a thermodynamically compatible way. For the moment a thermodynamically consistent formulation of plasticity is an unresolved issue and still heavily discussed in the literature.

We have obtained two coupled field equations for temperature and displacement from the balance equations of internal energy in Eq. (2.169) and of momentum in Eq. (2.170), respectively. In the initial frame the time derivatives are simply the partial time derivatives. From the balance of momentum and energy we generate the weak forms in the unit of energy. First we apply the usual time discretization. Secondly, we multiply the momentum balance with  $\delta u_i$  and the energy balance with  $\Delta t \delta T / T$  in order to rectify the unit of energy. Finally, we integrate by parts and obtain

$$F_u = \int_{\mathcal{B}_0} \left( \rho \frac{u_i - 2u_i^0 + u_i^{00}}{\Delta t^2} \delta u_i + \sigma_{ji} \delta u_{i,j} - \rho f_i \delta u_i \right) dV - \int_{\partial \mathcal{B}_0} \hat{t}_i \delta u_i dA , \quad (2.182)$$

as well as

$$\begin{aligned} F_T = \int_{\mathcal{B}_0} & \left( \frac{\rho c}{T} (T - T^0) \delta T + \Delta t \alpha_{ij} C_{ijkl} \left( \dot{\varepsilon}_{kl} - \dot{\varepsilon}_{kl}^p - \dot{\varepsilon}_{kl}^u \right) \delta T - \right. \\ & \left. - \Delta t q_i \left( \frac{\delta T}{T} \right)_{,i} - \Delta t \frac{\rho r}{T} \delta T - \frac{\Delta t}{T} \sigma_{ij} \dot{\varepsilon}_{ij}^p \delta T + \Delta t \frac{1}{T^2} q_i T_{,i} \delta T \right) dV + \\ & + \int_{\partial \mathcal{B}_0} \Delta t \hat{q} \frac{\delta T}{T} dA , \end{aligned} \quad (2.183)$$



with

$$\mathbf{F} = \mathbf{F}_u + \mathbf{F}_T , \quad (2.184)$$

where we implement the stress, the kinematic hardening, and the plastic strain according to the so-called incremental plasticity,

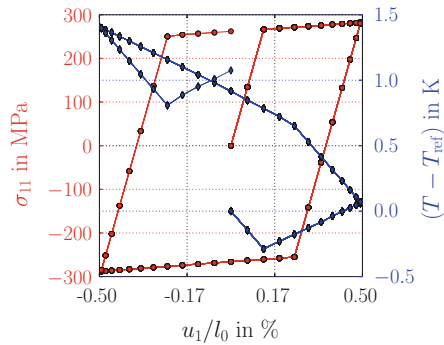
$$\begin{aligned} \dot{\sigma}_{ij} &= C_{ijkl}(\dot{\varepsilon}_{kl} - \dot{\varepsilon}_{kl}^p - \dot{\varepsilon}_{kl}^{th}) , \quad \sigma_{ij} = \sigma_{ij}^0 + \Delta t \dot{\sigma}_{ij} , \\ \dot{\varepsilon}_{kl}^{th} &= \alpha_{kl} \frac{T - T^0}{\Delta t} , \quad \dot{\beta}_{ij} = \frac{(\sigma_{|kl|}^0 - \beta_{kl}^0) \dot{\sigma}_{kl}}{\frac{2}{3} \sigma_Y^2} (\sigma_{|ij|}^0 - \beta_{ij}^0) , \\ \dot{\varepsilon}_{mn}^p &= \langle \gamma \rangle \frac{(\sigma_{|ij|}^0 - \beta_{ij}^0) C_{ijkl} (\varepsilon_{kl} - \varepsilon_{kl}^{th})}{\frac{4}{9} H \sigma_Y^2 + (\sigma_{|ij|}^0 - \beta_{ij}^0) C_{ijkl} (\sigma_{|kl|}^0 - \beta_{kl}^0)} (\sigma_{|mn|}^0 - \beta_{mn}^0) , \\ \beta_{ij} &= \beta_{ij}^0 + \Delta t \dot{\beta}_{ij} , \quad \varepsilon_{ij}^p = \varepsilon_{ij}^0 + \Delta t \dot{\varepsilon}_{ij}^p , \end{aligned} \quad (2.185)$$

and the heat flux as well as the strain as follows

$$q_i = -\kappa T_{,i} , \quad \varepsilon_{ij} = u_{(i,j)} , \quad \varepsilon_{ij}^0 = u_{(i,j)}^0 , \quad \dot{\varepsilon}_{ij} = \frac{1}{\Delta t} (\varepsilon_{ij} - \varepsilon_{ij}^0) . \quad (2.186)$$

Consider again a one-axial tensile testing, as in the previous section. By including plasticity we can capture an effect known from the daily life. If a cyclic loading with plastic deformation is utilized, the structure heats up. This phenomenon can clearly be seen in Fig. 2.8. A part of the energy has been stored such that the temperature decreases and increases. This part of the process is reversible and it is modeled by the entropy,  $\eta$ . Simultaneously, entropy is produced by  $\Sigma$ , which is an irreversible effect increasing the temperature further. In total, after one cycle of deformation, the temperature is increased approximately 1 K. The code is below including all realistic material parameters for a standard steel.

**Fig. 2.8** Tensile testing and temperature rise due to the plastic deformation



```

1  """Computational reality 14, thermoplasticity"""
2  __author__ = "B. Emek Abali"
3  __license__ = "GNU GPL Version 3.0 or later"
4  #This code underlies the GNU General Public License ,
5      ↪ http://www.gnu.org/licenses/gpl-3.0.en.html
6
7  from fenics import *
8  import numpy
9  set_log_level(ERROR)
10 #units: mm, 1000 kg=ton, s, MPa, mJ, K
11 Tref = 300. #in K
12 Tamb=Tref
13 # Material data of P265GH (St 45.8) from VDI Waermeatlas , at
14 ↪ 293.15 K
15 rho = 7.85E-9 #in tonne / mm^3
16 kappa = 57.0 #in mJ / (s mm K)
17 capacity = 430.0E6 #in mJ / (ton K)
18 alpha = 12.2E-6 #in 1/K at 373.15 K
19 EModul = 211.0E+3 #in MPa
20 nu = 0.28
21 H = 2600. #MPa
22 h = 10.E-3 #in mJ / (s mm^2 K)
23 sigmaY = Constant(250.0) #MPa
24
25 tMax = 10.0
26 Dt = 0.2
27 t = 0.0
28
29 xMin, xMax, xElements = 0.0, 100.0, 20
30 yMin, yMax, yElements = -5., +5., 2
31 zMin, zMax, zElements = -5., +5., 2
32 mesh = BoxMesh(Point(xMin,yMin,zMin), Point(xMax,yMax,zMax),
33 ↪ xElements,yElements,zElements)
34 N = FacetNormal(mesh)
35 length = abs(xMax-xMin)
36
37 Scalar = FunctionSpace(mesh, 'P', 1)
38 Vector = VectorFunctionSpace(mesh, 'P', 1)
39 Tensor = TensorFunctionSpace(mesh, 'P', 1)
40 Space = MixedFunctionSpace([Scalar, Vector])
41
42 cells = CellFunction('size_t', mesh)
43 facets = FacetFunction('size_t', mesh)
44 dA = Measure('ds', domain=mesh, subdomain_data=facets)
45 dV = Measure('dx', domain=mesh, subdomain_data=cells)
46
47 left = CompiledSubDomain('near(x[0],1) && on_boundary',l=xMin
48 ↪ )
49 right = CompiledSubDomain('near(x[0],1) && on_boundary',l=
50 ↪ xMax)
51 back = CompiledSubDomain('near(x[1],1) && on_boundary',l=yMin
52 ↪ )

```

```

47 front = CompiledSubDomain('near(x[1],1) && on_boundary',l=
    ↪ yMax)
48 bottom = CompiledSubDomain('near(x[2],1) && on_boundary',l=
    ↪ zMin)
49 top = CompiledSubDomain('near(x[2],1) && on_boundary',l=zMax)
50
51 facets.set_all(0)
52 displ = Expression(('0.5*sin(2.*pi*f*time)', '0.0', '0.0'),f
    ↪ =0.1,time=0)
53 bc1 = DirichletBC(Space.sub(1),displ,right)
54 bc2 = DirichletBC(Space.sub(1),Constant((0.0,0.0,0.0)),left)
55
56 bc = [bc1,bc2]
57
58 dunkn = TrialFunction(Space)
59 test = TestFunction(Space)
60 delT, delu = split(test)
61
62 unkn = Function(Space)
63 unkn0 = Function(Space)
64 unkn00 = Function(Space)
65
66 unkn_init = Expression(('Tini', '0', '0', '0'),Tini=Tref)
67 unkn = interpolate(unkn_init,Space)
68 unkn0.assign(unkn)
69 unkn00.assign(unkn0)
70
71 T, u = split(unkn)
72 T0, u0 = split(unkn0)
73 T00, u00 = split(unkn0)
74
75 i, j, k, l, m, n, o, p, r, s = indices(10)
76 delta = Identity(3)
77 lambada = EModul * nu / (1.+nu) / (1.-2.*nu)
78 mu = 0.5 * EModul / (1.+nu)
79 C = as_tensor(lambada*delta[i,j]*delta[k,l]+mu*delta[i,k]*
    ↪ delta[j,l]+mu*delta[i,l]*delta[j,k], (i,j,k,l))
80 alp = alpha*delta
81
82 peeps0 = Function(Tensor)
83 sigma0= Function(Tensor)
84 dev_sigma0 = as_tensor(sigma0[i,j]-1./3.*sigma0[k,k]*delta[i,
    ↪ j], (i,j))
85 beta0 = Function(Tensor)
86
87 eps = sym(grad(u))
88 eps0 = sym(grad(u0))
89 Deps = (eps-eps0)/Dt
90
91 teps = alp*(T-Tref)
92 Dteps = alp*(T-T0)/Dt
93
94 gamma = Function(Scalar)

```

```

95 Dpeps = as_tensor(gamma*(dev_sigma0-beta0)[i,j]*C[i,j,k,l]*(
    ↳ Deps-Dteps)[k,l]/(4./9.*H*sigmaY**2+(dev_sigma0-beta0)
    ↳ [m,n]*C[m,n,o,p]*(dev_sigma0-beta0)[o,p])*(dev_sigma0-
    ↳ beta0)[r,s], (r,s))
96
97 Dsigma = as_tensor(C[i,j,k,l]*(Deps-Dpeps-Dteps)[k,l], (i,j))
98 Dbeta = as_tensor(gamma*(dev_sigma0-beta0)[k,l]*Dsigma[k,l]
    ↳ ]/(2./3.*sigmaY**2)*(dev_sigma0-beta0)[i,j], (i,j))
99
100 sigma = sigma0 + Dt*Dsigma
101 beta = beta0 + Dt*Dbeta
102 peps = peps0 + Dt*Dpeps
103
104 dev_sigma = as_tensor(sigma[i,j]-1./3.*sigma[k,k]*delta[i,j],
    ↳ (i,j))
105 q = as_tensor(-kappa*T.dx(i), (i,))
106
107 f = Constant((0.0,0.0,0.0))
108 R = Constant(0.0)
109 qHat = h*(T-Tamb)
110
111 F_u = (rho*(u-2.*u0+u00)[i]/Dt/Dt*delu[i] + sigma[j,i]*delu[i]
    ↳ ].dx(j) - rho*f[i]*delu[i])*dV
112 F_T = (rho*capacity/T*(T-T0)*delT+ Dt*alp[i,j]*C[i,j,k,l]*(
    ↳ Deps-Dpeps-Dteps)[k,l]*delT - Dt*q[i]*(delT/T).dx(i) -
    ↳ Dt*rho*R/T*delT - Dt*sigma0[i,j]*Dpeps[j,i]*delT/T)*
    ↳ dV + Dt*qHat*delT/T*dA
113
114 Form = F_u + F_T
115 Gain = derivative(Form, unkn, dunkn)
116
117 import matplotlib as mpl
118 mpl.use('Agg')
119 import matplotlib.pyplot as pylab
120 pylab.rc('text', usetex=True)
121 pylab.rc('font', family='serif', serif='cm', size=30)
122 pylab.rc('legend', fontsize=30)
123 pylab.rc(('xtick.major', 'ytick.major'), pad=15)
124 pylab.subplots_adjust(top=0.90)
125 pylab.subplots_adjust(bottom=0.17)
126 pylab.subplots_adjust(left=0.20)
127 pylab.subplots_adjust(right=0.75)
128
129 fig = pylab.figure(1, figsize=(14,10))
130 ax1 = fig.add_subplot(111)
131 ax1.grid(True, axis='x')
132 ax1.set_xlabel(r'$u_1/l_0$ in %$')
133 ax1.set_ylabel(r'$\sigma_{11}$ in MPa', color='r')
134 ax1.tick_params(axis='y', colors='r')
135 ax1.grid(True, axis='y', color='r')
136 ax2 = ax1.twinx()
137 ax2.set_ylabel(r'$ (T-T_{\mathrm{ref}}) $ in K', color='b')
138 ax2.tick_params(axis='y', colors='b')
139 ax2.grid(True, axis='y', color='b')

```

```

140 ax2.ticklabel-format(style='sci', axis='y', scilimits=(-2,2))
141
142 pwd='/calcul/CR14/'
143 file_u = File(pwd+'displ.pvd')
144 file_T = File(pwd+'temp.pvd')
145 strain, stress, temp = [], [], []
146
147 while t < tMax:
148     print 'time: ', t
149     displ.time = t
150     solve(Form==0, unkn, bc, J=Gain, \
151           solver_parameters={"newton_solver":{"linear_solver":
152             ↳ "mumps", "relative_tolerance": 1e-5}}, \
153           form_compiler_parameters={"cpp_optimize": True, "
154             ↳ representation": "quadrature", "
155             ↳ quadrature_degree": 2} )
156
157     file_T << (unkn.split()[0], t)
158     file_u << (unkn.split()[1], t)
159
160     sigma_ = project(sigma, Tensor, solver_type="mumps", \
161           form_compiler_parameters={"cpp_optimize": True, "
162             ↳ representation": "quadrature", "
163             ↳ quadrature_degree": 2} )
164     sigma0.assign(sigma_)
165     beta_ = project(beta, Tensor, solver_type="mumps", \
166           form_compiler_parameters={"cpp_optimize": True, "
167             ↳ representation": "quadrature", "
168             ↳ quadrature_degree": 2} )
169     beta0.assign(beta_)
170     flow_ = project(1./2.*(dev_sigma0-beta0)[i,j]*(dev_sigma0
171             ↳ -beta0)[i,j] - 1./3.*sigmaY**2, Scalar,
172             ↳ solver_type="mumps", \
173           form_compiler_parameters={"cpp_optimize": True, "
174             ↳ representation": "quadrature", "
175             ↳ quadrature_degree": 2} )
176     flow_bool = flow_.vector().array() >= 0.
177     direction_ = project((dev_sigma0-beta0)[i,j]*Deps[i,j],
178             ↳ Scalar, solver_type="mumps", \
179           form_compiler_parameters={"cpp_optimize": True, "
180             ↳ representation": "quadrature", "
181             ↳ quadrature_degree": 2} )
182     direction_bool = 1./2.*(numpy.sign(direction_.vector().
183             ↳ array())+1.)
184     gamma.vector[:] = numpy.array(flow_bool*direction_bool,
185             ↳ dtype=int)
186
187     peps_ = project(peps, Tensor, solver_type="mumps", \
188           form_compiler_parameters={"cpp_optimize": True, "
189             ↳ representation": "quadrature", "
190             ↳ quadrature_degree": 2} )
191     peps0.assign(peps_)
192
193     unkn00.assign(unkn0)

```

```

176     unkn0.assign( unkn )
177
178     strain.append( unkn0.split() [1] (xMax,0.,0.) [0] / length
           ↪ * 100.)
179     stress.append( sigma0 (xMax/2.,0.,0.) [0] )
180     temp.append( unkn0.split() [0] (xMax/2.,0.,0.) - Tref )
181     ax1.plot( strain, stress, 'o-', color='r' )
182     ticks = numpy.linspace( numpy.array( strain ).min(), numpy.
           ↪ array( strain ).max(), 4 )
183     ax1.set_xticks( ticks )
184     ax1.set_xticklabels( [ '%1.2f' % i_ticks for i_ticks in
           ↪ ticks ] )
185     ax2.plot( strain, temp, 'd-', color='b' )
186     fig.savefig( pwd+'CompReal14_tensiletest.pdf' )
187
188     t += Dt

```

## To-do

Plastic deformation generates a temperature change in the system.

- Try to change the boundary conditions to adiabatic boundaries. Guess the result before starting the numerical calculation.
- Find out a stress/strain curve for another material and determine the plasticity modulus or hardening parameter  $H$  in MPa. What will be the result for a lower or higher  $H$  parameter?
- Find out the material behavior of aluminum and explain why the linear hardening model used in this section is inadequate for modeling an aluminum sample.

## References

1. Abali, B.E.: Thermodynamically compatible modeling, determination of material parameters, and numerical analysis of nonlinear rheological materials. Ph.D. thesis, Technische Universität, Berlin, Institute of Mechanics (2014)
2. Cengel, Y.A.: Introduction to Thermodynamics and Heat Transfer. McGraw Hill Higher Education Press, New York (2007)
3. Coleman, B., Noll, W.: The thermodynamics of elastic materials with heat conduction and viscosity. Arch. Ration. Mech. Anal. **13**(1), 167–178 (1963)
4. Eckart, C.: The thermodynamics of irreversible processes. i. The simple fluid. Phys. Rev. **58**, 267–269 (1940)
5. Eckart, C.: The thermodynamics of irreversible processes. ii. Fluid mixtures. Phys. Rev. **58**(3), 269–275 (1940)
6. Eckart, C.: The thermodynamics of irreversible processes. iii. Relativistic theory of the simple fluid. Phys. Rev. **58**(10), 919–924 (1940)
7. Eckart, C.: The thermodynamics of irreversible processes. iv. The theory of elasticity and anelasticity. Phys. Rev. **73**, 373–382 (1948)
8. de Groot, S.R., Mazur, P.: Non-Equilibrium Thermodynamics. Dover Publications, New York (1984)
9. Questions in fluid mechanics: Gad-el Hak, M. J. Fluids Eng. **117**, 3–5 (1995)

10. Kilian, H.G., Höhne, G.: The müller-calorimeter and applications. *Thermochimica Acta* **69**(1), 199–219 (1983)
11. Müller, I.: The coldness, a universal function in thermoelastic bodies. *Arch. Ration. Mech. Anal.* **41**(5), 319–332 (1971)
12. Müller, I.: *Thermodynamics*. Pitman Publishing, London (1985)
13. Müller, I., Ruggeri, T.: *Rational Extended Thermodynamics*. Springer, New York (1998)
14. Pauli, W.: *Pauli lectures on physics*. In: *Thermodynamics and the Kinetic Theory of Gases*, vol. 3. Dover (2000) repub. of MIT Press, Cambridge, Massachusetts (1973)
15. Rajagopal, K.R.: A new development and interpretation of the Navier-Stokes fluid which reveals why the "Stokes assumption" is inapt. *Int. J. Non-Linear Mechan.* **50**, 141–151 (2013)
16. Salazar, A.: Energy propagation of thermal waves. *Eur. J. Phys.* **27**(6), 1349 (2006)
17. Truesdell, C., Toupin, R.A.: *Principles of classical mechanics and field theory*. In: Flügge, S. (ed.), *Handbuch der Physik*, vol. III/1 (1960)
18. Tzou, D.Y.: Experimental support for the lagging behavior in heat propagation. *J. Thermophys. Heat Transf.* **9**(4), 686–693 (1995)

Computational Reality  
Solving Nonlinear and Coupled Problems in Continuum  
Mechanics

Abali, B.E.

2017, XVII, 308 p. 48 illus. in color., Hardcover

ISBN: 978-981-10-2443-6