Solid mechanics and linear Finite Element Analysis

1. Introduction

In this practical we study the mechanical behaviour of a solid from a continuum mechanics point of view. In the first part we discuss basic continuum mechanical concepts as e.g. fundamental notations, kinematics and balance laws. We then derive governing equations for the linear elastic behaviour of solids. Commonly used notation can be reviewed in Appendix A. The second part treats the numerical solution of elasticity problems with finite element analysis. The finite element method (FEM) has grown in popularity over the last decades and is nowadays the standard numerical solution method for (elliptical) partial differential equations (PDEs). There are various commercial programs available that are used in science and industry. All of them, however are based on the same algorithmic structure, which we derive and numerically implement in a simple 1D code in this practical.

2. Derivation of the governing differential equations

2.1. Kinematics

In the following we study the motion and deformation of a simply connected, compact body $\mathcal{B}(t)$ as a function of time t in the Euclidean space. The boundary (i.e. the surface) of the body is denoted by $\partial\mathcal{B}(t)$. We assume that the reference configuration of the body $\mathcal{B}_0 = \mathcal{B}(t=0)$ is stress-free. During the deformation process the spatial configuration of the body w.r.t. to the chosen euclidean coordinate system is changing to the so-called current configuration $\mathcal{B}(t)$. We denote the material points of the body in the reference and the current configuration by the position vectors $\boldsymbol{X} \in \mathcal{B}_0$ and $\boldsymbol{x}(t) \in \mathcal{B}(t)$, respectively (cf. Fig. 1). The current position and therefore the movement of a material point within the body is described by the function $\boldsymbol{\chi}$, depending on the reference position and time:

$$\boldsymbol{x} = \boldsymbol{\chi}(\boldsymbol{X}, t). \tag{1}$$

The displacement $\boldsymbol{u}$ can then be defined as the difference between current and reference position,

$$\boldsymbol{u}(\boldsymbol{X}, t) = \boldsymbol{\chi}(\boldsymbol{X}, t) - \boldsymbol{X}. \tag{2}$$

The velocity of a material point is the total derivative of the displacement with respect to time

$$\boldsymbol{v}(\boldsymbol{X}, t) = \frac{\mathrm{d}\boldsymbol{\chi}(\boldsymbol{X}, t)}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{u}(\boldsymbol{X}, t)}{\mathrm{d}t} = \dot{\boldsymbol{u}}(\boldsymbol{X}, t). \tag{3}$$

The acceleration is defined as the total derivative of the velocity with respect to time

$$\boldsymbol{a}(\boldsymbol{X}, t) = \frac{\mathrm{d}\boldsymbol{v}(\boldsymbol{X}, t)}{\mathrm{d}t} = \dot{\boldsymbol{v}}(\boldsymbol{X}, t). \tag{4}$$

In the following, we consider only small deformations and choose as deformation measure the *infinitesimal strain tensor*, which is the symmetric part of the displacement gradient

$$\boldsymbol{\varepsilon}(\boldsymbol{X}, t) = \frac{1}{2}\left(\frac{\partial \boldsymbol{u}(\boldsymbol{X},t)}{\partial \boldsymbol{X}} + \left(\frac{\partial \boldsymbol{u}(\boldsymbol{X},t)}{\partial \boldsymbol{X}}\right)^{\mathsf{T}}\right). \tag{5}$$

Most quantities in the following section are introduced in the reference configuration; therefore, if we do not write the position vector $\boldsymbol{X}$ we always imply that the point of evaluation is in the reference configuration.

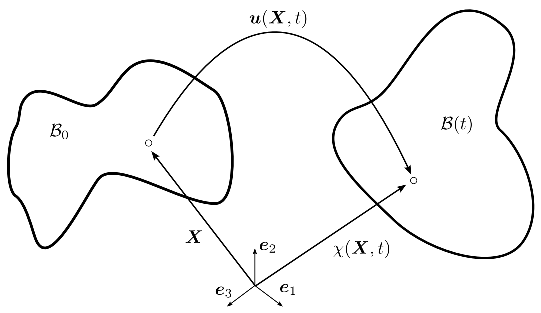

Figure 1: Schematic representation of the kinematic relations. $\mathcal{B}_0$ is the body in the reference configuration, $\mathcal{B}(t)$ denotes the body in the deformed configuration after all points of $\mathcal{B}$ underwent the displacement defined by $\boldsymbol{u}$. $\boldsymbol{X}$ and $\boldsymbol{x} = \boldsymbol{\chi}(\boldsymbol{X},t)$ are position vectors of the same point in both configurations.

2.2. Balance equations

Consider an arbitrary body in the Euclidean space. Newton postulated that "The alteration of motion is ever proportional to the motive force impress'd; and is made in the direction of the right line in which that force is impress'd." [*] Interpreted in a modern sense we can say " The rate of change of the momentum $\boldsymbol{p}$ of a body is equal to the force $\boldsymbol{F}$ acting on it."[†] Mathematically this can be written as

$$\boldsymbol{F} = \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = \frac{\mathrm{d}(m\boldsymbol{v})}{\mathrm{d}t} = \frac{\mathrm{d}(m\dot{\boldsymbol{u}})}{\mathrm{d}t}, \tag{6}$$

where m denotes the mass and $\boldsymbol{v}$ the velocity of each material point of the body. If the mass stays unchanged in time, we get the well-known equation

$$\boldsymbol{F} = m\frac{\mathrm{d}(\boldsymbol{v})}{\mathrm{d}t} = m\boldsymbol{a}, \tag{7}$$

with $\boldsymbol{a}$ the acceleration of the body. The net force $\boldsymbol{F}$ is the sum of the forces acting on the surface of the body, $\boldsymbol{F}_{\partial\mathcal{B}}$, and the forces acting on the material points in the volume of the body, $\boldsymbol{F}_{\mathcal{B}}$,

$$\boldsymbol{F} = \sum \boldsymbol{F}_{\partial\mathcal{B}} + \sum \boldsymbol{F}_{\mathcal{B}}. \tag{8}$$

As an illustrative example imagine a book lying on a table. Here, the book is not in motion; hence, the change of momentum is zero, $\dot{\boldsymbol{p}} = 0$. Thus, all surface and volume forces together are in equilibrium. In this case, the gravitational force acting on the book is in equilibrium with the contact force on the bottom side of the book. If we describe a continuum body (e.g. the book) as the infinite sum of continuously distributed, infinitely small volume

[*]The Mathematical Principles of Natural Philosophy (1846) by Isaac Newton, translated by Andrew Motte
[†]The Feynman lectures on physics, Volume 1 (2005), Chapter 9 by Feynman, Richard P.; Leighton, Robert B.; Sands, Matthew.

elements $\mathrm{d}V$ and surface elements $\mathrm{d}A$, the equations for the net force and the momentum can be written as

$$\boldsymbol{F} = \int_{\partial\mathcal{B}} \boldsymbol{t}\,\mathrm{d}A + \int_{\mathcal{B}} \rho\boldsymbol{b}\,\mathrm{d}V, \tag{9}$$

$$\boldsymbol{p} = \int_{\mathcal{B}} \rho\dot{\boldsymbol{u}}\,\mathrm{d}V, \tag{10}$$

with the mass density ρ, the body force density $\boldsymbol{b}$ and the traction vector $\boldsymbol{t}$.

Inserting equations (9) and (10) into (6) we get the balance equation for the linear momentum in integral form,

$$\int_{\mathcal{B}} \rho\ddot{\boldsymbol{u}}\,\mathrm{d}V = \int_{\partial\mathcal{B}} \boldsymbol{t}\,\mathrm{d}A + \int_{\mathcal{B}} \rho\boldsymbol{b}\,\mathrm{d}V. \tag{11}$$

The traction $\boldsymbol{t}$ acting on the surface element of an arbitrary cut with normal $\boldsymbol{n}$ implicitly defines the Cauchy stress tensor $\boldsymbol{\sigma}$ through

$$\boldsymbol{t}\,\mathrm{d}A = \boldsymbol{\sigma}\cdot\boldsymbol{n}\,\mathrm{d}A. \tag{12}$$

With the aid of the divergence theorem (cf. Appendix A) we obtain from the balance equation

$$\int_{\mathcal{B}} \rho\ddot{\boldsymbol{u}}\,\mathrm{d}V = \int_{\mathcal{B}} \mathrm{div}\,(\boldsymbol{\sigma})\;\mathrm{d}V + \int_{\mathcal{B}} \rho\boldsymbol{b}\,\mathrm{d}V, \tag{13}$$

or in local form for a material point the governing differential equation of the deformation $\boldsymbol{u}$ of a solid

$$\boxed{\rho\ddot{\boldsymbol{u}} = \mathrm{div}\,(\boldsymbol{\sigma}) + \rho\boldsymbol{b}}. \tag{14}$$

Note: The balance of angular momentum,

$$\int_{\mathcal{B}} \boldsymbol{x}\times\rho\ddot{\boldsymbol{u}}\,\mathrm{d}V = \int_{\partial\mathcal{B}} \boldsymbol{x}\times\boldsymbol{t}\,\mathrm{d}A + \int_{\mathcal{B}} \boldsymbol{x}\times\rho\boldsymbol{b}\,\mathrm{d}V, \tag{15}$$

can be derived in the same way and leads to the symmetry of the Cauchy stress tensor,

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^{\mathsf{T}}. \tag{16}$$

In addition to the balance of linear and angular momentum in a closed mechanical system the total amount of mass in conserved. This the governed by the continuity equation for mass density, which is given by

$$\tfrac{\partial\rho}{\partial t} + \mathrm{div}\,(\rho\dot{\boldsymbol{u}}) = 0. \tag{17}$$

Task 1.1: Simplify the local form (14) in case of statics (no acceleration) and give an example of body forces other than the gravitational force. Describe a case where we can neglect the body forces.

2.3. Constitutive equations

The kinematic relations and balance equations are valid regardless of the material under consideration. Taking a closer look at the equations, one observes that we are not able to solve the balance equations: in the isothermal case without dissipative effects there are altogether 13 unknowns (the components of $\boldsymbol{\sigma}$ and $\boldsymbol{u}$ and additionally ρ), but only 7 equations (3 equations each for the balance of linear and angular momentum and one equation for the balance of mass). Therefore, additional equations are required for solution. These are obtained by providing a link between the balance equation and the kinematic equation, through a constitutive equation that describes the stress tensor $\boldsymbol{\sigma}$ as a function of the strain tensor $\boldsymbol{\varepsilon}$:

$$\boldsymbol{\sigma} = f(\boldsymbol{\varepsilon}). \tag{18}$$

In general the function f can be non-linear, depends on the material properties of the solid and may even consider the deformation history. In the following we use a simple function - a linear mapping between $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$, which is called the stiffness tensor $\mathbb{C}$. This material law is known as *Hooke's law*:

$$\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon}. \tag{19}$$

Task 2.1: Write eq. (19) in index notation and denote for which indices the Einstein summation convention applies. How many components does $\mathbb{C}$ have in general in 3 dimensions? Show that if one uses that the stress and strain tensors are symmetric ($\boldsymbol{\sigma} = \boldsymbol{\sigma}^{\mathsf{T}}$; $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{\mathsf{T}}$), the number of independent components reduces to 36.

Above, we noted that f is a 'constitutive' function. This implies that f cannot be derived from e.g. fundamental balance laws. To determine f one usually has to make assumptions concerning the material behaviour. One commonly used possibility is to assume the existence of a strain energy density function of the form

$$W = \frac{1}{2}\boldsymbol{\sigma} : \boldsymbol{\varepsilon}, \tag{20}$$

from which the stiffness tensor can be derived by

$$\mathbb{C} = \frac{\partial^2 W}{\partial \boldsymbol{\varepsilon} \partial \boldsymbol{\varepsilon}}. \tag{21}$$

Task 2.2: Show that the stiffness tensor has 21 independent components, if we take the symmetries from Task 2.1 *and* eq. (21) into account. (*Hint*: use Schwarz's theorem)

In the isotropic case the components of the stiffness tensor are given by

$$C_{ijkl} = \lambda \delta_{ij}\delta_{kl} + \mu \left(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right), \tag{22}$$

where δ_{ij} is the Kronecker symbol, which is 1 for $i = j$ and 0 otherwise, cf. Appendix A. The more common engineering material parameters Young's Modulus E and Poisson ratio ν are related to the Lamé Constants μ, λ by

$$E = 2\mu(1+\nu), \tag{23}$$

$$\nu = \frac{\lambda}{2(\lambda+\mu)}. \tag{24}$$

Task 2.3: For this task we consider a one dimensional bar of length l, which is clamped at $x = 0$ and statically loaded by a prescribed force F applied at $x = l$. We assume a fully isotropic material and a constant mass density $\rho = \text{const.}$. Furthermore, for this one dimensional problem, we assume that the cross section A of the bar is constant and also does not change under load (implying $\nu = 0$) (cf. Fig. 2). Show that the *local balance equation* of linear momentum in one dimension is reduces to

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(EA\frac{\mathrm{d}u}{\mathrm{d}x}\right) + A\rho b(x) = 0. \tag{25}$$

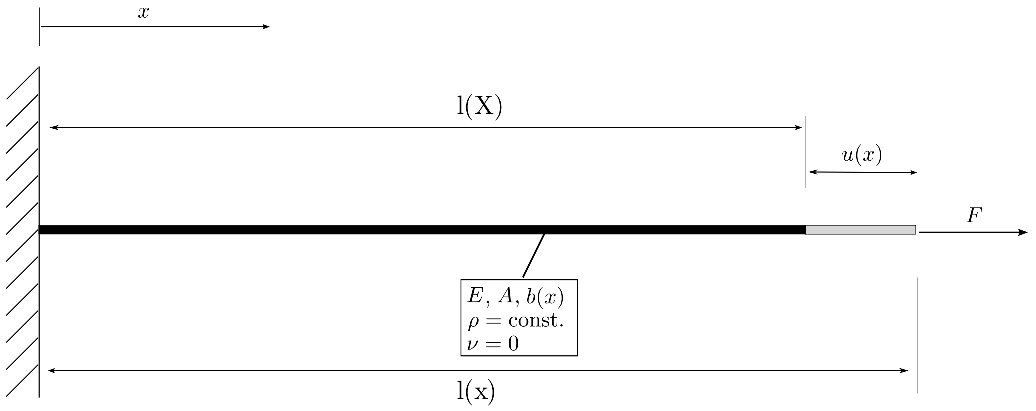

Figure 2: Clamped bar in reference configuration (black) and current configuration (grey) under load as an example of a 1D elastostatic problem.

3. Weak form

As a starting point we consider the system

$$
\begin{aligned}
div\,(\boldsymbol{\sigma}) + \rho \boldsymbol{b} &= \boldsymbol{0} && \text{in } \mathcal{B}, && (26a)\\
\boldsymbol{t} &= \bar{\boldsymbol{t}} && \text{on } \Gamma_{\mathrm{t}} \subset \partial\mathcal{B}, && (26b)\\
\boldsymbol{u} &= \bar{\boldsymbol{u}} && \text{on } \Gamma_{\mathrm{u}} \subset \partial\mathcal{B}, && (26c)
\end{aligned}
$$

with the von Neumann boundary Γ_{t} and the Dirichlet boundary Γ_{u} and the respective prescribed boundary tractions $\bar{\boldsymbol{t}}$ and boundary displacements $\bar{\boldsymbol{u}}$. The displacement vector $\boldsymbol{u}$ has to be twice continuously differentiable (recall eq. (19) and (5)), a strong requirement when it comes to the numerical implementation. Instead of demanding that eq. (26) holds in an absolute sense (e.g. in every material point) we will derive the so-called weak formulation, where this equation only holds for specific test functions over a domain. Here we take a rather formal approach based on the Lax-Milgram Theorem[*]. To derive the weak form we first multiply eq. (26a) with a test function $\delta\boldsymbol{u}$ which satisfies the property $\delta\boldsymbol{u} = 0$ on Γ_{u}. Subsequent integration leads to

$$\int_{\mathcal{B}} div\,(\boldsymbol{\sigma}) \cdot \delta\boldsymbol{u}\,\mathrm{d}V + \int_{\mathcal{B}} \rho\boldsymbol{b} \cdot \delta\boldsymbol{u}\,\mathrm{d}V = 0. \tag{27}$$

With the relation

$$div\,(\boldsymbol{\sigma} \cdot \delta\boldsymbol{u}) = div\,(\boldsymbol{\sigma}) \cdot \delta\boldsymbol{u} + \boldsymbol{\sigma}^{\mathsf{T}} : grad\,(\delta\boldsymbol{u}) \tag{28}$$

it follows that

$$\int_{\mathcal{B}} div\,(\boldsymbol{\sigma} \cdot \delta\boldsymbol{u})\;\mathrm{d}V - \int_{\mathcal{B}} \boldsymbol{\sigma}^{\mathsf{T}} \cdot grad\,(\delta\boldsymbol{u})\;\mathrm{d}V + \int_{\mathcal{B}} \rho\boldsymbol{b} \cdot \delta\boldsymbol{u}\,\mathrm{d}V = 0. \tag{29}$$

For the first part of eq. (29) we use the divergence theorem and the property $\delta\boldsymbol{u} = 0$, which holds on the Dirichlet boundary Γ_u, to obtain

$$
\begin{aligned}
\int_{\mathcal{B}} div\,(\boldsymbol{\sigma} \cdot \delta\boldsymbol{u})\;\mathrm{d}V = \int_{\partial\mathcal{B}} (\boldsymbol{\sigma} \cdot \delta\boldsymbol{u}) \cdot \boldsymbol{n}\,\mathrm{d}A &= \int_{\Gamma_{\mathrm{t}}} (\boldsymbol{\sigma} \cdot \boldsymbol{n}) \cdot \delta\boldsymbol{u}\,\mathrm{d}A + \int_{\Gamma_{\mathrm{u}}} \boldsymbol{\sigma} \cdot \underbrace{\delta\boldsymbol{u}}_{=0} \cdot \boldsymbol{n}\,\mathrm{d}A\\
&= \int_{\Gamma_{\mathrm{t}}} \bar{\boldsymbol{t}}\delta\boldsymbol{u}\,\mathrm{d}A.
\end{aligned}
\tag{30}
$$

With $\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon}$ and substitution into eq. (29) this leads to

$$\boxed{\int_{\mathcal{B}} (\mathbb{C} : \boldsymbol{\varepsilon}) : grad\,(\delta\boldsymbol{u})\;\mathrm{d}V = \int_{\mathcal{B}} \rho\boldsymbol{b} \cdot \delta\boldsymbol{u}\,\mathrm{d}V + \int_{\Gamma_{\mathrm{t}}} \bar{\boldsymbol{t}} \cdot \delta\boldsymbol{u}\,\mathrm{d}A}, \tag{31}$$

which is the *weak form* of the system (26).

The resulting integral is not only easier to solve numerically, but additionally $\boldsymbol{u}$ only needs to be once continuously differentiable. Now the solution holds only in an averaged sense (due to the integral) instead of absolutely in every material point and we can easily split the integral into subdomains for discretization purposes.

Task 3.1: Show that the weak form of the example problem (25) from Task 2.3 is given by

$$\int_0^l \frac{\mathrm{d}u(x)}{\mathrm{d}x} EA \frac{\mathrm{d}\delta u(x)}{\mathrm{d}x}\,\mathrm{d}x = \int_0^l A\rho b(x)\delta u(x)\,\mathrm{d}x + \underbrace{\bar{t}A}_{F}\,\delta u(l). \tag{32}$$

[*]This approach is valid for any PDE, but for specific cases (e.g. solid mechanics) there also exists a more physical derivation from the principle of virtual power (a variational approach)

4. Discretization and shape functions

For numerical solution of the weak form we need to discretize the system and find an appropriate approximation of the integrals. This is done by dividing the domain into discrete parts (or 'elements') as can be seen in Fig. 3 for the example problem. As we saw in Section 2 the balance law is also valid for arbitrary subdomains. Because the weak form eq. (31) is equivalent to the partial differential equation we can solve it on subdomains (i.e.elements). In one dimension we can decompose the domain into n_{el} line elements e_k with element boundary ∂e_k, so that the solution of eq. (25) is given by

$$\sum_{k=1}^{n_{el}} \left\{ \int_{e_k} \frac{\mathrm{d}u(x)}{\mathrm{d}x} EA \frac{\mathrm{d}\delta u(x)}{\mathrm{d}x}\,\mathrm{d}x - \int_{e_k} A\rho b(x)\delta u(x)\,\mathrm{d}x + \bar{t_e} A\delta u(x)\Big|_{\partial e_k} \right\} = 0, \tag{33}$$

where $\bar{t_e}$ denotes the von Neumann boundary conditions on each element and $\bar{t}(l)A(l) = F$ the global von Neumann boundary condition. Note that the Dirichlet boundary conditions are implicitly defined through the test function by $\delta u = 0$ on Γ_{u} (cf. Section 3).

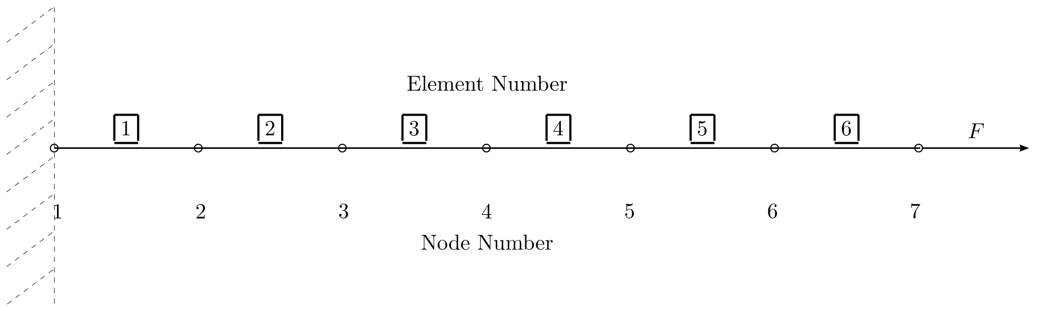

Figure 3: Approximation of the bar with constant cross section by one-dimensional rods (dashed) and discretization into 6 elements e with altogether 7 nodes

4.1. Interpolation

For the discretization of the displacement field we assume that the discrete values $\hat{u}_i$ at nodes n_i are equal to the values of the displacement field at this position

$$\hat{u}_i = u(x(n_i)). \tag{34}$$

Subsequently we abbreviate the position of a node n_i by $x_i = x(n_i)$. For the values between two nodes (inside the element) we need an approximation. Because polynomials are convenient in terms of differentiation and integration we will use such a scheme for interpolating in between the elements' nodes. An easy way to derive an interpolation function is the use of Lagrange basis polynomials $l_i(x)$. They have the properties to be 1 on the position of the corresponding node n_i and 0 on all other nodes n_j, which can be expressed in terms of the Kronecker symbol

$$l_i(x_j) = \delta_{ij}. \tag{35}$$

A polynomial with k given function values (e.g. 'support points') x_i can be conveniently written as the following product

$$l_i(x) := \prod_{\substack{1\le m\le k\\ m\ne i}} \frac{x - x_m}{x_i - x_m} = \frac{(x-x_1)}{(x_i-x_1)} \cdots \frac{(x-x_{i-1})}{(x_i-x_{i-1})} \frac{(x-x_{i+1})}{(x_i-x_{i+1})} \cdots \frac{(x-x_k)}{(x_i-x_k)}. \tag{36}$$

In finite element terminology these interpolations inside an element are called 'shape functions' denoted by N, where $N_i(x) = l_i(x)$. For piece-wise interpolations inside each element the *number of support points is the number of nodes of the element.* Therefore the chosen polynomial degree of the shape functions dictates the number of nodes N_{max} belonging to the element. For example a so-called linear element in 1D has two nodes n_1 and n_2. It is called linear because the interpolation between the nodes is a linear polynomial. The related piecewise (linear) shape functions are N_1 and N_2 (cf. Fig.4). The interpolation of the displacement field can now easily be obtained by multiplication of the discrete value at a node with the corresponding shape function. Because of property (35) we get the exact values at the nodes with a polynomial interpolation in between. Summing up the resulting products leads then to the approximation $\tilde{u}$ of the field values in each element

$$\tilde{u}(x) = \sum_{i=1}^{N_{max}} N_i(x)\hat{u}_i. \tag{37}$$

For the numerical implementation it is beneficial to write $\tilde{u}(x)$ as a vector operation between the shape functions and the discrete node values

$$\tilde{u}(x) = \sum_{i=1}^{N_{max}} N_i(x)\hat{u}_i = \begin{bmatrix} N_1(x) & N_2(x) & \dots & N_i(x) \end{bmatrix} \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_i \end{bmatrix} = \boldsymbol{N}(x) \cdot \hat{\boldsymbol{u}}. \tag{38}$$

Note that the numbering of the vector elements is somewhat arbitrary and also depends on the element type (i.e. the number of nodes). Figure 4 shows on the right side the resulting interpolation for a linear element.

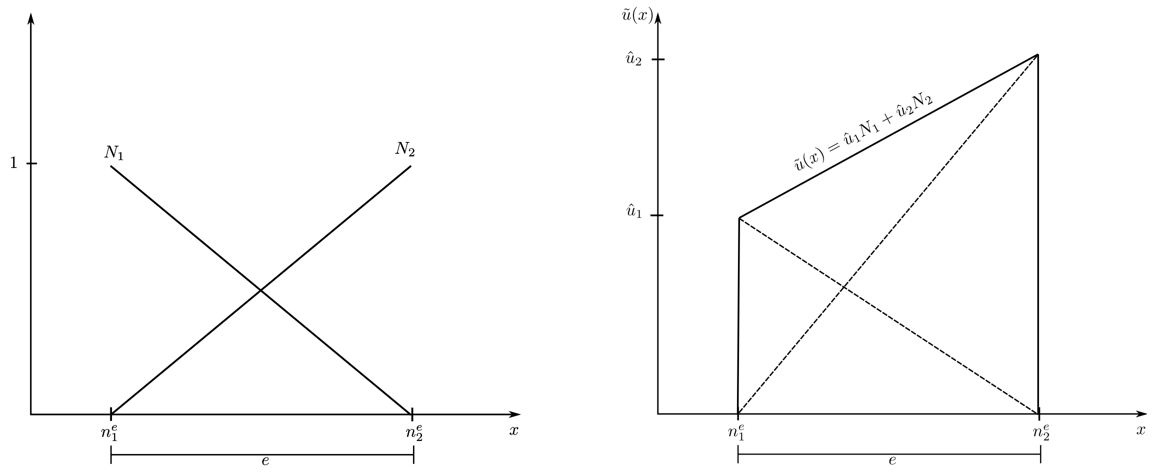

Figure 4: Shape functions N (left) and resulting interpolation $\tilde{u}$ (right) for a linear element e with node values $\hat{u}$

Task 4.1: Sketch the 6 elements that were used in Fig. 3 and draw the two linear shape functions N_1 and N_2 for each element on top of it.

Implement the shape functions for *one element* by use of eq. (38) with MatLab or Python and test the approximation with 3 different functions of your choice. Name those functions that can be exactly represented by the numerical approximation.

How does this result affect the representations of the displacement and stress fields on the element level?

Hint: The input arguments of the function should be two nodes (e.g. the element nodes n_1^e, n_2^e) with their respective function values (e.g. $\hat{u}_1, \hat{u}_2$) and positions (e.g. x_1, x_2) and in addition a list of evaluation points $\{\xi_1, \xi_2, \dots, \xi_i\} \in [x_1, x_2]$ in between. The function should compute the interpolation between the two points and return the interpolated function values at the evaluation points, e.g. $\{\tilde{u}(\xi_1), \tilde{u}(\xi_2), \dots, \tilde{u}(\xi_i)\}$.

Task 4.2: The use of shape functions allows for an easy implementation of the displacement gradient. Taking the spatial derivative of eq. (37) leads to

$$\begin{aligned}\frac{\mathrm{d}\tilde{u}(x)}{\mathrm{d}x} &= \sum_{i=1}^{N_{\max}} \frac{\mathrm{d}(N_i(x)\hat{u}_i)}{\mathrm{d}x} = \sum_{i=1}^{N_{\max}} \underbrace{\frac{\mathrm{d}N_i(x)}{\mathrm{d}x}}_{B_i(x)} \hat{u}_i \\ &= \sum_{i=1}^{N_{\max}} B_i(x)\hat{u}_i,\end{aligned} \tag{39}$$

where B_i are the spatial derivatives of the shape functions N_i.

Implement the derivatives in the same way as you implemented the shape functions in Task 4.1. Take as input again two nodes and an evaluation point and return the interpolated gradient value.

Plot them and test them again with the functions from Task 4.1.

To summarize, we can write the approximations for the displacement and the spatial derivative in an element in vector form as

$$\boxed{\tilde{u}(x) = \boldsymbol{N}(x)\cdot\hat{\boldsymbol{u}} \quad \text{and} \quad \frac{\mathrm{d}\tilde{u}(x)}{\mathrm{d}x} = \boldsymbol{B}(x)\cdot\hat{\boldsymbol{u}},} \tag{40}$$

with the row vectors $\boldsymbol{N}$ and $\boldsymbol{B}$ containing the shape functions and their derivatives and the column vector $\hat{\boldsymbol{u}}$ containing the discrete values at the nodes. For the approximation of the test function (and its derivative) we choose the same shape functions as for the displacement. Inserting the approximations into the one-dimensional weak form (29) leads to

$$\int_e \frac{\mathrm{d}u(x)}{\mathrm{d}x} EA \frac{\mathrm{d}\delta u(x)}{\mathrm{d}x}\,\mathrm{d}x - \int_e A\rho b(x)\delta u(x)\,\mathrm{d}x + \bar{t}_e A\delta u(x)\Big|_{\partial e} = 0 \tag{41}$$

$$\Rightarrow \delta\hat{\boldsymbol{u}}\cdot\left(\int_e \boldsymbol{B}^{\mathsf{T}}(x)EA\boldsymbol{B}(x)\,\mathrm{d}x\right)\cdot\hat{\boldsymbol{u}} - \delta\hat{\boldsymbol{u}}\cdot\left(\int_e \boldsymbol{N}^{\mathsf{T}}(x)A\rho b(x)\,\mathrm{d}x + \bar{t}_e A\Big|_{\partial e}\right) = 0, \tag{42}$$

which holds for each element e. Equation (42) holds for arbitrary test functions $\delta\hat{\boldsymbol{u}}$, so that

$$\underbrace{\int_e \boldsymbol{B}^\mathsf{T}(x)EA\boldsymbol{B}(x)\,\mathrm{d}x\cdot\hat{\boldsymbol{u}}}_{\boldsymbol{K}^e} - \underbrace{\int_e \boldsymbol{N}^\mathsf{T}(x)A\rho b(x)\,\mathrm{d}x}_{\boldsymbol{f}^e_{\mathcal{B}}} + \underbrace{\bar{t}_e A\Big|_{\partial e}}_{\boldsymbol{f}^e_{\Gamma}} = 0 \tag{43}$$

follows. This can be written in compact form as a system of linear equations

$$\boldsymbol{K}^e\cdot\hat{\boldsymbol{u}} = \boldsymbol{f}^e = \boldsymbol{f}^e_{\mathcal{B}} + \boldsymbol{f}^e_{\Gamma}, \tag{44}$$

with the element stiffness matrix $\boldsymbol{K}^e$, the displacement at the nodes $\hat{\boldsymbol{u}}$ and the right hand side $\boldsymbol{f}^e$ which can be decomposed into a volume part $\boldsymbol{f}^e_{\mathcal{B}}$ and a surface part $\boldsymbol{f}^e_{\Gamma}$.

Note that the element stiffness matrix has the dimension $N_{\max}\times N_{\max}$. The components of the matrix are integrals over the element domain e. In one dimension this is the line integral of the element length with the integrands A, E and $\boldsymbol{B}$.

Obviously the displacement vectors $\hat{\boldsymbol{u}}$ of the elements are not independent from each other: recall the discretization of the 1D bar from Fig. 3. Because two adjacent elements share a node in between them, also the displacements at this node must be the same.

As a result we can not solve the system of linear equations for *each element* independently, but we have to solve a system of coupled linear equation *for all elements*. The coupled system is obtained by a so-called assembly routine $\boldsymbol{\Lambda}^{n_{\mathrm{el}}}_{k=1}$, which takes the connectivity/ordering of the elements and the local numbering of the nodes into account. It assembles the element values $(\boldsymbol{K}^{e_k}, \hat{\boldsymbol{u}}^{e_k}, \boldsymbol{f}^{e_k})$ of each element e_k for all n_{el} elements into the global stiffness matrix $\boldsymbol{K}$, the global displacement vector $\boldsymbol{d}$, and the global right hand side vector $\boldsymbol{f}$:

$$\underbrace{\boldsymbol{\Lambda}^{n_{\mathrm{el}}}_{k=1}\boldsymbol{K}^{e_k}}_{\boldsymbol{K}}\cdot\underbrace{\boldsymbol{\Lambda}^{n_{\mathrm{el}}}_{k=1}\hat{\boldsymbol{u}}^{e_k}}_{\boldsymbol{d}} = \underbrace{\boldsymbol{\Lambda}^{n_{\mathrm{el}}}_{k=1}\boldsymbol{f}^{e_k}}_{\boldsymbol{f}}. \tag{45}$$

How the assembling of the elements is done in one dimension will be discussed in Section 6.

5. Numerical Integration

Until now, we know how to solve the problem of discretization and interpolation between discrete values. But to solve the discrete system (43) we still need to evaluate the integrals. Although analytical solutions of the occurring integrals are in principle possible, this is in more general situations (e.g. 2D or 3D) relatively complicated. Therefore those integrals are usually solved numerically. Here we only briefly motivate what numerical integration (also called numerical quadrature) is, and which integration schemes are usually used.

In general the approximation of an integral (in 1D) in the interval $[x_i, x_{i+1}]$ can be described as a weighted sum of n function values at specific integration (quadrature) points ζ_j,

$$\int_{x_i}^{x_{i+1}} f(x)\,\mathrm{d}x \approx (x_{i+1}-x_i)\sum_{j=1}^{n}\lambda_j f(\zeta_j), \tag{46}$$

where λ_j denotes the integration weights. The position of the integration points depends on the chosen integration scheme. E.g. for the trapezoidal rule (cf. Fig. 5) they are located on the boundaries of the element, $f(\zeta_1) = f(x_i)$, $f(\zeta_2) = f(x_{i+1})$, and are equally weighted with 0.5:

$$\int_{x_i}^{x_{i+1}} f(x)\,\mathrm{d}x \approx (x_{i+1}-x_i)\sum_{j=1}^{2}\lambda_j f(\zeta_j) = \frac{(x_{i+1}-x_i)}{2}(f(x_i)+f(x_{i+1})). \tag{47}$$

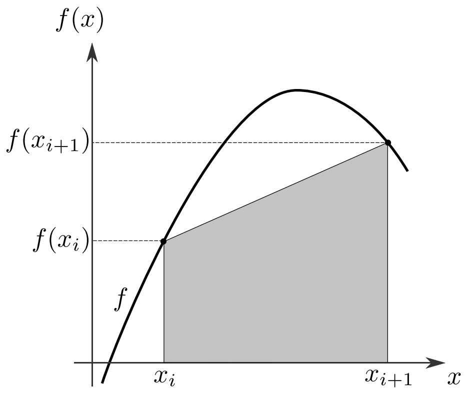

Figure 5: Sketch of the numerical integration with the trapezoidal rule of a function f.

Task 5.1: Implement a numerical integration scheme, based on eq. (46) that takes as arguments two points x_1 and x_2 which define the integration domain, and $n \geq 2$ integration points ζ and weights λ. For example in Python the function header could look like

def quadrature(x1, x2, n, lambda, zeta):...

Note, that the dimension of lambda and zeta is n. In the first step implement only the dimension $n = 2$ (the discussed trapezoidal rule) and an exception handling for $n \neq 2$.

Task 5.2: The trapezoidal rule is a special case of the so-called 'closed Newton-Cotes quadrature' formulas. Table 1 shows the first three degrees of the Newton-Cotes formulas with their common names. An important property of all closed Newton-Cotes formulas is that their integration points are always equidistant.

Implement the higher order Newton-Cotes schemes from Table 1 by using the implementation of Task 5.1. Test you integration with polynomials of different order. To which polynomial degree is each integration scheme exact? How many integration points are needed if the numerical integration should integrate linear/quadratic shape functions exactly?

Remark: By choosing different, not necessarily equidistant, weights and integration points one can optimize the numerical integration for a specific problem. Modern polynomial integration schemes are usually based on the so-called 'Gauss quadrature'.

Now we have all tools to solve the system on the element level: discretization, interpolation and numerical integration.

n	Common name	Weights λ	Integration Points ζ
2	Trapezoidal rule	$\frac{1}{2}, \frac{1}{2}$	x_i, x_{i+1}
3	Simpson rule	$\frac{1}{6}, \frac{4}{6}, \frac{1}{6}$	$x_i, \frac{x_i + x_{i+1}}{2}, x_{i+1}$
4	3/8 rule	$\frac{1}{8}, \frac{3}{8}, \frac{3}{8}, \frac{1}{8}$	$x_i, \frac{2x_i + x_{i+1}}{3}, \frac{x_i + 2x_{i+1}}{3}, x_{i+1}$

Table 1: Closed Newton Cotes formulas

Task 5.3: The final task on the element level is to solve the weak form (29) for *one element*. The steps on the element level are:

1. Approximate the displacement field and its gradient with discrete values on the element nodes and (linear) shape functions in between.
2. Solve the resulting integrals (43) for the components of the stiffness matrix and potentially the right hand side with numerical integration.

To test your integration and interpolation consider the clamped 1D bar as discussed in Section 2.

Assume that the bar is fully isotropic and homogeneous with a constant Young's modulus of $E = 210000\,\mathrm{N/mm^2}$ and that the body force is negligible ($b = 0$). The cross section area should be constant $A = \text{const.} = 25\,\mathrm{mm}^2$ and the length of the bar is given as $l = 50\,\mathrm{mm}$. Discretize the bar with *one element* (two nodes). For the Dirichlet boundary condition at the left side $u(x = 0) = 0$, modify your stiffness matrix by hand. The von Neumann boundary condition at $x = l$ is directly implemented as the right hand side of the system and has the magnitude $F = 5\,\mathrm{N}$.

Write down by hand the resulting system of linear equations for the two element node values $\hat{u}_1$ and $\hat{u}_2$ in matrix-vector form ($\boldsymbol{d} = [\hat{u}_1 \hat{u}_2]^{\mathrm{T}}$). Compare your handwritten solution to your implementation.

Choose an appropriate solver for the system of linear equations and *explain your choice*. You can either take an existing one (e.g. from Practical 1) or program one on you own. Solve the system and compare it to the analytical solution.

6. Assembly of the system

So far we are able to solve one finite element. For a better approximation more elements are needed. As already mentioned in Section 4 several connected elements with common nodes lead to a coupled system of linear equations.

6.1. Assembly of a spring system

To better understand the coupling and the assembly of a finite element system we first consider a simple one dimensional system of massless springs, as shown in Fig. 6. The force F_s of a linear spring is given by

$$F_\mathrm{s}\boldsymbol{e}_x = c\Delta u\boldsymbol{e}_x \tag{48}$$

with the spring constant c and the change in length of the spring Δu. Note, that internally the reaction force R_1 is induced by the Dirichlet boundary condition due to the displacement constrain at node 1. For example, for node 1 the force equilibrium equation depends on the spring constant c and the displacements u_1 and u_2 of the nodes 1 and 2, respectively:

$$-R_1 + F_1 = 0, \tag{49}$$
$$-R_1 + c(u_2 - u_1) = 0. \tag{50}$$

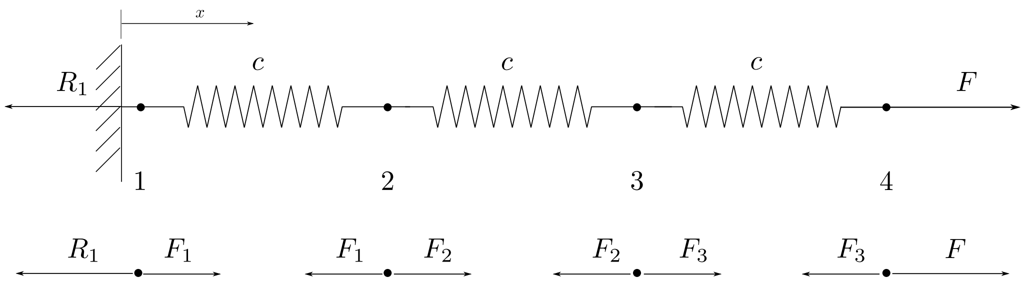

Figure 6: One dimensional spring system (top) with node numbering (middle) and the force equilibrium for each node (bottom)

Task 6.1: Derive the equilibrium equations for the other nodes and show that the global system can be written as

$$\underbrace{c\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}}_{\boldsymbol{K}} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}}_{\boldsymbol{d}} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ -F \end{bmatrix}}_{\boldsymbol{f}} - \underbrace{\begin{bmatrix} R_1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\boldsymbol{R}} \tag{51}$$

Note that the internal forces F_i at the elements cancel out, except for the reaction forces $\boldsymbol{R}$ at nodes with Dirichlet boundaries.

6.2. Assembly of the finite element system

For the assembly of the finite element system we proceed in analogy to the spring system. Similar to the spring system we have nodes in the finite element system, which connect structures with a specific stiffness. The node values are discrete values ($\hat{\boldsymbol{u}}$) that describe the system behaviour. In the spring system the structures are the springs with stiffness

c, in the finite element system this corresponds to the elements with the stiffness $\boldsymbol{K}^e$ (cf. Section 4). This motivates why the assembly procedure is the same for the spring system and the finite element system. In both cases we take the element/spring stiffness matrix, which is set up as the system of equations for the force equilibrium at the corresponding nodes, and copy it block-wise into the the global stiffness matrix $\boldsymbol{K}$. The assembly of the right hand side is straight forward. We only have to consider the external von Neumann boundary conditions (recall that the internal forces cancel out). Therefore the 'assembly' is only to insert the external forces at the right positions.

Task 6.2: Implement similar to Task 6.1 an assembly routine for linear elements. It should take a given element matrix (e.g. the spring stiffness or the calculated element matrix from Task 5.5) and the corresponding nodes as input parameter. The routine should block-wise copy the element stiffness matrices into the global stiffness matrix at the correct positions. If done correctly, you should be able to loop over all elements and fill the global matrix iteratively with the element matrices.

Task 6.3: Print out your global system matrix and compare it for $c = 1$ (or $\boldsymbol{K}^e = 1$) with eq. (51). What are the properties of the global stiffness matrix? Why can we not yet solve our global system? What is the physical explanation/interpretation of this?

7. Boundary conditions

7.1. Dirichlet boundary conditions for our specific problem

So far we are not able to solve the global system $\boldsymbol{K} \cdot \boldsymbol{d} = \boldsymbol{f}$ (cf. Tasks 6.2, 6.3), because Dirichlet boundary conditions were not yet considered in the implementation. The easiest approach to solve this problem is to modify by hand those equilibrium equations, which are dependent on boundary nodes. For example for the spring system of Section 6.1 we know that the displacement at node $\boxed{1}$ is zero, $u_1 = 0$. So we can rewrite the system in analogy to Task 6.1 as

$$\underbrace{c \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}}_{\tilde{\boldsymbol{K}}} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}}_{\boldsymbol{d}} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ F \end{bmatrix}}_{\boldsymbol{f}}. \qquad (52)$$

This system of linear equations can also be obtained by inserting the known displacement values ($u_1 = 0$) into eq. (49). Written in matrix form the internal forces are

$$\boldsymbol{R} = \begin{bmatrix} R_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = c \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}. \qquad (53)$$

Inserting $\boldsymbol{R}$ into eq. (51) and reformulation gives the same result as eq. (52). Since u_1 is already known, the first row can be eliminated from the system. Subsequently we can reduce the system by deleting all rows and columns that are directly affected by Dirichlet boundary conditions:

$$\underbrace{c\begin{bmatrix}2 & -1 & 0\\ -1 & 2 & -1\\ 0 & -1 & 1\end{bmatrix}}_{\boldsymbol{K}_\mathrm{r}}\underbrace{\begin{bmatrix}u_2\\ u_3\\ u_4\end{bmatrix}}_{\boldsymbol{d}_\mathrm{r}} = \underbrace{\begin{bmatrix}0\\ 0\\ F\end{bmatrix}}_{\boldsymbol{f}_\mathrm{r}}. \tag{54}$$

The resulting reduced stiffness matrix $\boldsymbol{K}_\mathrm{r}$ is no longer singular, so that we are able to solve the reduced system for the reduced global displacement vector $\boldsymbol{d}_\mathrm{r}$ and the reduced right hand side $\boldsymbol{f}_\mathrm{r}$.

7.2. Dirichlet boundary conditions in general

A general procedure for implementing Dirichlet boundary conditions is to first split the displacements into a known part $\boldsymbol{d}_\Gamma$ on Γ_u and an unknown part $\check{\boldsymbol{d}}$. In the next step we can modify the right hand side with the reaction forces $\boldsymbol{R}$ which are induced by $\boldsymbol{d}_\Gamma$:

$$\boldsymbol{d} = \boldsymbol{d}_\Gamma + \check{\boldsymbol{d}} \tag{55}$$

$$\Rightarrow \boldsymbol{K}\cdot\boldsymbol{d} = \boldsymbol{K}\cdot(\boldsymbol{d}_\Gamma + \check{\boldsymbol{d}}) = \boldsymbol{f} \tag{56}$$

$$\Rightarrow \boldsymbol{K}\cdot\check{\boldsymbol{d}} = \boldsymbol{f} - \underbrace{\boldsymbol{K}\cdot\boldsymbol{d}_\Gamma}_{\boldsymbol{R}}. \tag{57}$$

Afterwards we reduce the system by deleting rows and columns. This can be done directly or via a permutation matrix $\boldsymbol{P}$ and matrix vector multiplications. The permutation matrix maps the full displacement vector to the reduced vector

$$\boldsymbol{d}_\mathrm{r} = \boldsymbol{P}\cdot\boldsymbol{d}, \tag{58}$$

$$\boldsymbol{d}_\mathrm{r} = \boldsymbol{P}\cdot\check{\boldsymbol{d}}. \tag{59}$$

The permutation matrix satisfies the property [†]

$$P_{ik}P_{jk} = \delta_{ij}. \tag{60}$$

It has for n nodes with m Dirichlet boundary condition the dimension $(n-m)\times n$, so that $i,j \in 1,2,\ldots,(n-m)$ and $k \in 1,2,\ldots,n$. With these properties we can write

$$\boldsymbol{P}^\mathsf{T}\cdot\boldsymbol{d}_\mathrm{r} = \check{\boldsymbol{d}}. \tag{61}$$

and by multiplying eq. (57) with $\boldsymbol{P}$ from the left side,

$$\left(\boldsymbol{P}\cdot\boldsymbol{K}\cdot\boldsymbol{P}^\mathsf{T}\right)\cdot\boldsymbol{d}_\mathrm{r} = \boldsymbol{P}\cdot(\boldsymbol{f} - \boldsymbol{K}\cdot\boldsymbol{d}_\Gamma), \tag{62}$$

$$\Leftrightarrow$$

$$\boldsymbol{K}_\mathrm{r}\cdot\boldsymbol{d}_\mathrm{r} = \boldsymbol{f}_\mathrm{r}, \tag{63}$$

we get a reduced system that can be solved. After solving, the full displacement vector can be calculated as

$$\boldsymbol{d} = \boldsymbol{P}^\mathsf{T}\cdot\boldsymbol{d}_\mathrm{r} + \boldsymbol{d}_\Gamma. \tag{64}$$

[†]Note that P is neither orthogonal nor invertible and $P_{ki}P_{kj} \neq \delta_{ij}$.

Task 7.1: Write down by hand the permutation matrix for a 4×4 system with Dirichlet boundary conditions acting on the first node.

Task 7.2: Program a scheme that allows to modify the global system with Dirichlet boundary conditions. Use either the straightforward row and column deleting or the permutation matrix method. Your program should take as input the node number and the displacement of the constrained node. Be careful in particular with the implementation of non-zero displacements.

Test your implementation with the global stiffness matrix from the spring system (Task 6.1), imposing $u_1 = 0$. As a second test case substitute the von Neumann boundary condition with a second Dirichlet boundary condition, which should not be zero ($u_4 \neq 0$).

8. Summary and postprocessing

The implementation of Dirichlet boundary conditions was the last step for the one dimensional finite element code. As already mentioned, the extension to more dimensions is similar to 1D (although the node connectivity and shape functions are more complicated). To sum up, the required steps are always as follows:

1. Derive the governing PDEs.
2. Formulate the weak form of the problem.
3. Discretize the system with finite elements.
4. Choose an interpolation scheme (shape functions) and an appropriate numerical integration scheme.
5. Assemble the element stiffness matrices and the right hand side into a global system.
6. Include the boundary conditions and reduce the system.
7. Solve the reduced (sparse) system with a matching solver.
8. Postprocess your results and verify it via analytical solutions.

We are often interested in the approximate stress field $\tilde{\boldsymbol{\sigma}}$. This can be obtained in a postprocessing step: since we already have the interpolated strains $\tilde{\boldsymbol{\varepsilon}}$ via the displacement gradients, we can easily compute the stresses as

$$\tilde{\boldsymbol{\sigma}}(x) = E\tilde{\boldsymbol{\varepsilon}}(x) = E\boldsymbol{B}(x) \cdot \hat{\boldsymbol{u}}. \tag{65}$$

Task 8.1: Use all existing subroutines to perform a linear, one dimensional finite element analysis. As an example problem, use the system from Task 2.3. Discretize it with $n = 6$ elements and compute the approximate stress field. Plot the results and also investigate different boundary conditions on different positions. For the material properties and the geometry you can take the values from Task 5.5.

Feel free to work in groups, but *everyone* has to send a *working and commented* code to shucheta.shegufta@fau.de and of course do not forget your handwritten solutions.

Appendix A. Notations

We use the common continuum mechanical notation with bold letters. Small, bold Latin letters like $\boldsymbol{u}, \boldsymbol{n}$ denote *vectors* if not stated otherwise. *Second order tensors* are denoted with bold Latin capitals or Greek letters, e.g. $\boldsymbol{A}, \boldsymbol{\sigma}, \boldsymbol{\varepsilon}$. *Tensors of order four* use the notation $\mathbb{C}, \mathbb{A}, \dots$. *Scalar values* are identified with non-bold letters (e.g. f, a, α). We make use of Einstein's summation convention which states that, when an index variable appears twice in a single term, this implies summation of that term over all values of the index. By use of a orthonormal base in 3D with basis vectors $\boldsymbol{e}_i (i = 1, 2, 3)$, vectors and tensors can be written as

$$\boldsymbol{a} = \sum_{i=1}^{3} a_i \boldsymbol{e}_i = a_i \boldsymbol{e}_i, \tag{66}$$

$$\boldsymbol{A} = \sum_{i=1}^{3} \sum_{j=1}^{3} A_{ij} \boldsymbol{e}_i \otimes \boldsymbol{e}_j = A_{ij} \boldsymbol{e}_i \otimes \boldsymbol{e}_j. \tag{67}$$

For single and double contractions we use a single and a double dot,respectively

$$\boldsymbol{A} \cdot \boldsymbol{a} = A_{ij} a_j \boldsymbol{e}_i, \tag{68}$$

$$\boldsymbol{A} \cdot \boldsymbol{B} = A_{ik} B_{kj} \boldsymbol{e}_i \otimes \boldsymbol{e}_j, \tag{69}$$

$$\boldsymbol{A} : \boldsymbol{B} = A_{ij} B_{ij}, \tag{70}$$

$$\mathbb{C} : \boldsymbol{A} = C_{ijkl} A_{kl} \boldsymbol{e}_i \otimes \boldsymbol{e}_j. \tag{71}$$

The cross product in 3 dimensions between two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ is given by

$$\boldsymbol{u} \times \boldsymbol{v} = (u_2 v_3 - u_3 v_2)\boldsymbol{e}_1 + (u_3 v_1 - u_1 v_3)\boldsymbol{e}_2 + (u_1 v_2 - u_2 v_1)\boldsymbol{e}_3. \tag{72}$$

The gradient of a vector $\boldsymbol{a}$ is defined as

$$\nabla \boldsymbol{a} = grad\,(\boldsymbol{a}) = \tfrac{\partial a_i}{\partial x_j} \boldsymbol{e}_i \otimes \boldsymbol{e}_j, \tag{73}$$

the divergence of a tensor field $\boldsymbol{A}$ as

$$\nabla \cdot \boldsymbol{A} = \mathrm{d}iv\,(\boldsymbol{A}) = \tfrac{\partial A_{ij}}{\partial x_j} \boldsymbol{e}_i, \tag{74}$$

where $\boldsymbol{x}$ denotes in both cases the spatial position.

The divergence theorem states that the outward flux of a tensor field $\boldsymbol{\sigma}$ through a closed surface A with outward normal $\boldsymbol{n}$ is equal to the volume integral of the divergence over the region inside (V) the surface,

$$\iiint_V \mathrm{d}iv\,(\boldsymbol{\sigma})\, \mathrm{d}V = \iint_A \boldsymbol{\sigma} \cdot \boldsymbol{n}\, \mathrm{d}A. \tag{75}$$

As an abbreviation of the total derivative with respect to time we use a dot on top of the derived variable, e.g.

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \dot{\boldsymbol{x}}. \tag{76}$$

In addition we define the Kronecker symbol as

$$\delta_{ij} = \begin{cases} 1, & \text{if } j = i \\ 0, & \text{if } j \neq i. \end{cases} \tag{77}$$