Finite Element Analysis

Questions and Answers

Daniel Zahnd

May 22, 2025

Contents

1 Preliminaries

1. **Q** — What are the basic requirements of a good finite elements simulation?

A — In order to obtain a good finite element model, the geometry of the problem, the involved forces and the material properties need to be known and clear. For example, the geometry of a beam, the applied stress (forces) and the material properties of the beam need to be known to properly simulate the bending under application of some force.

2. **Q** — What is the finite element method?

A — The finite element method is a way to numerically solve partial differential equations. It works for any shape of the domain. The basic idea is to break the domain down into very small - but finite - elements, in which everything is constant or linear. There is also the non-linear finite element method, but it is much more complicated.

2 1D and 2D discrete systems

3. **Q** — How can an example of a static 1D discrete system be solved using the finite element method?

A — Looking at the diagram fig. 1, one can set up the connectivity table as seen in table 1. The equation for finite element i describing the forces linked to the displacements is given by

$$\begin{pmatrix} {}^ik & -{}^ik \\ -{}^ik & {}^ik \end{pmatrix} \begin{pmatrix} {}^iu_1 \\ {}^iu_2 \end{pmatrix} = \begin{pmatrix} {}^if_1 \\ {}^if_2 \end{pmatrix} \quad \Leftrightarrow \quad {}^i\mathbf{k}{}^i\mathbf{u} ={}^i\mathbf{f}. \tag{2.1}$$

Element	n1, dof1	n2, dof2
1	1	1
2	1	2
3	2	3

Table 1: Connectivity table for the nodes of problem 1.

Writing the global stiffness matrix as $\mathbf{K}$, the global displacement vector as $\mathbf{U} = (U_1, U_2, U_3)^\top$ and the global force vector as $\mathbf{F} = (F_1, F_2, F_3)^\top$, one can write the matrix equation describing the static system as

$$\underbrace{\begin{pmatrix} {}^1k + {}^2k & -{}^1k - {}^2k & 0 \\ -{}^1k - {}^2k & {}^1k + {}^2k + {}^3k & -{}^3k \\ 0 & -{}^3k & {}^3k \end{pmatrix}}_{= \begin{pmatrix} K_{11} & K_{12} & 0 \\ K_{21} & K_{22} & K_{23} \\ 0 & K_{32} & K_{33} \end{pmatrix}} \begin{pmatrix} U_1 \\ U_2 \\ U_3 \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix} \tag{2.2}$$

Looking at the problem fig. 1, one can see that the global displacements U_1 and U_3 are constrained to $U_1 = U_3 = 0$. Furthermore, what is known is the applied force $F = F_2$ to node 2. The unknown quantities are the displacement U_2 and the reaction forces F_1 and F_3. Implementing this into the matrix equation above leads to

$$\begin{pmatrix} K_{11} & K_{12} & 0 \\ K_{21} & K_{22} & K_{23} \\ 0 & K_{32} & K_{33} \end{pmatrix} \begin{pmatrix} 0 \\ U_2 \\ 0 \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix}. \tag{2.3}$$

This leads to th equation $K_{22}U_2 = F_2$, which is solved by

$$U_2 = \frac{F_2}{K_{22}} = \frac{F_2}{{}^1k + {}^2k + {}^3k}. \tag{2.4}$$

Finally, the reaction forces can be calculated using the displacement U_2 as

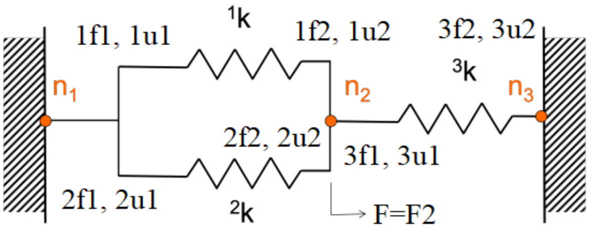


Figure 1: Visualization of problem 1.

$$F_1 = K_{12}\frac{F_2}{K_{22}} = -F_2\frac{{}^1k + {}^2k}{{}^1k + {}^2k + {}^3k}, \qquad F_3 = K_{32}\frac{F_2}{K_{22}} = -F_2\frac{{}^3k}{{}^1k + {}^2k + {}^3k}. \tag{2.5}$$

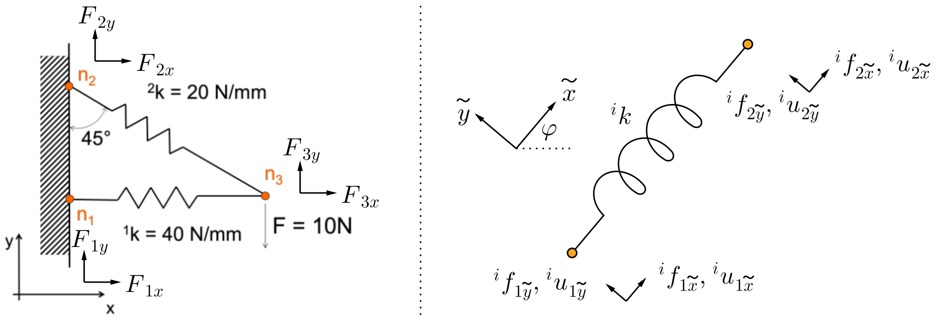

F_{2y}
F_{2x}
n2
²k = 20 N/mm
45°
F_{3y}
n3
F_{3x}
n1
F_{1y}
¹k = 40 N/mm
F = 10N
y
x
F_{1x}
$\tilde{y}$
$\tilde{x}$
φ
^ik
$^if_{2\tilde{x}}, {}^iu_{2\tilde{x}}$
$^if_{2\tilde{y}}, {}^iu_{2\tilde{y}}$
$^if_{1\tilde{y}}, {}^iu_{1\tilde{y}}$
$^if_{1\tilde{x}}, {}^iu_{1\tilde{x}}$

Figure 2: Visualization of problem 2.

4. **Q** — How can an example of a static 2D discrete system be solved using the finite element method?

A — Considering fig. 2, the connectivity table can be set up as seen in table 2. Consider first of all the spring element drawn in the right panel of fig. 2. In the reference frame $(\tilde{x}, \tilde{y})$ of the spring element, one can set up the internal force equation as

$$^i\tilde{\boldsymbol{f}} = {}^i\tilde{\mathbf{k}}{}^i\tilde{\boldsymbol{u}} \quad \Leftrightarrow \quad \begin{pmatrix} ^if_{1\tilde{x}} \\ ^if_{1\tilde{y}} \\ ^if_{2\tilde{x}} \\ ^if_{2\tilde{y}} \end{pmatrix} = \begin{pmatrix} ^ik & 0 & -^ik & 0 \\ 0 & 0 & 0 & 0 \\ -^ik & 0 & ^ik & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} ^iu_{1\tilde{x}} \\ ^iu_{1\tilde{y}} \\ ^iu_{2\tilde{x}} \\ ^iu_{2\tilde{y}} \end{pmatrix}. \tag{2.6}$$

Using the rotation matrix (clockwise rotation about angle φ)

$$\mathbf{R} = \begin{pmatrix} \cos(\varphi) & \sin(\varphi) & 0 & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 & 0 \\ 0 & 0 & \cos(\varphi) & \sin(\varphi) \\ 0 & 0 & -\sin(\varphi) & \cos(\varphi) \end{pmatrix}, \tag{2.7}$$

one can relate $^i\tilde{\boldsymbol{f}}$ and $^i\tilde{\boldsymbol{u}}$ to their counterpart in the global reference frame (x, y) as

$$^i\tilde{\boldsymbol{f}} = \mathbf{R}{}^i\boldsymbol{f}, \qquad ^i\tilde{\boldsymbol{u}} = \mathbf{R}{}^i\boldsymbol{u}. \tag{2.8}$$

Using these equations, one can write

$$^i\boldsymbol{f} = \mathbf{R}^\top{}^i\tilde{\boldsymbol{f}} = \mathbf{R}^\top{}^i\tilde{\mathbf{k}}\mathbf{R}{}^i\boldsymbol{u} = {}^i\mathbf{k}{}^i\boldsymbol{u} \tag{2.9}$$

with $^i\mathbf{k} \doteq \mathbf{R}^\top{}^i\tilde{\mathbf{k}}\mathbf{R}$. Setting up now the connectivity table for the 2 nodes and 2 degrees of freedom for each node as shown in table 2, one can construct the total global stiffness matrix $\mathbf{K}$ by means of $^1\mathbf{k}$ and $^2\mathbf{k}$. Defining the total global force vector $\boldsymbol{F} = (F_{1x}, F_{1y}, F_{2x}, F_{2y}, F_{3x}, F_{3y})^\top$ and the total global displacement vector as $\boldsymbol{U} = (U_{1x}, U_{1y}, U_{2x}, U_{2y}, U_{3x}, U_{3y})^\top$, we can write

$$\mathbf{K}\boldsymbol{U} = \boldsymbol{F}. \tag{2.10}$$

The boundary conditions are $F_{1x} = F_{1y} = F_{2x} = F_{2y} = F_{3x} = U_{1x} = U_{1y} = U_{2x} = U_{2y} = 0$, $F_{3y} = -10\,\mathrm{N}$ and $U_{3x}, U_{3y} \neq 0$. The problem is then solved by appropriate software tools. The solution of the problem is

$$\boldsymbol{F} = \begin{pmatrix} -10 \\ 0 \\ 10 \\ 10 \\ 0 \\ -10 \end{pmatrix} \mathrm{N}, \qquad \boldsymbol{U} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.25 \\ -1.25 \end{pmatrix} \mathrm{mm}. \tag{2.11}$$

Element	n1, dof1	n1, dof2	n2, dof3	n2, dof4
1	1	2	5	6
2	3	4	5	6

Table 2: Connectivity table for the nodes of problem 2.

3 1D continuous system

5. **Q** — How does an example application of the static 1D continuous FEM method work?

A — Consider a 1D problem of heat transfer given by the equation

$$q(x) = -k\frac{\mathrm{d}^2T(x)}{\mathrm{d}x^2}, \tag{3.1}$$

where $T(x)$ is the temperature at location x, $q(x)$ is a source function and k is the thermal conductivity.

Let a particular problem be defined by the boundary conditions $T(0) = T(l) = 0$ and

$$q(x) = \begin{cases} q_0 & 0 \leq x \leq l/2 \\ 0 & l/2 < x \leq l \end{cases}. \tag{3.2}$$

Let $\delta T(x)$ be a weighting function, which satisfies the same boundary conditions as $T(x)$. The weak formulation of the differential equation with $\delta T(x)$ is then given by

$$\int_0^l \left[k\frac{\mathrm{d}^2T(x)}{\mathrm{d}x^2} + q(x)\right]\delta T(x)\,\mathrm{d}x \quad \forall\ \delta T(x). \tag{3.3}$$

By the partial integration rule

$$\int f(x)g'(x)\,\mathrm{d}x = f(x)g(x) - \int f'(x)g(x)\,\mathrm{d}x \tag{3.4}$$

and defining

$$f(x) \doteq k\delta T(x), \qquad g(x) \doteq \frac{\mathrm{d}T(x)}{\mathrm{d}x}, \tag{3.5}$$

we obtain from eq. (3.3) the equation

$$\left[k\delta T(x)\frac{\mathrm{d}T(x)}{\mathrm{d}x}\right]_0^l - \int_0^l k\frac{\mathrm{d}T(x)}{\mathrm{d}x}\frac{\mathrm{d}\delta T(x)}{\mathrm{d}x}\,\mathrm{d}x = -\int_0^l q(x)\delta T(x)\,\mathrm{d}x = 0. \tag{3.6}$$

The first term vanishes, because we assume the same boundaries for $\delta T(x)$ as for $T(x)$, namely $\delta T(0) = \delta T(l) = 0$. Furthermore, the last integral can be simplified to

$$\int_0^l q(x)\delta T(x)\,\mathrm{d}x = q_0\int_0^{l/2} \delta T(x)\,\mathrm{d}x \tag{3.7}$$

due to the definition of $q(x)$. With that, we finally obtain

$$\int_0^l k\frac{\mathrm{d}T(x)}{\mathrm{d}x}\frac{\mathrm{d}\delta T(x)}{\mathrm{d}x}\,\mathrm{d}x = q_0\int_0^{l/2} \delta T(x)\,\mathrm{d}x \qquad \forall\ \delta T(x). \tag{3.8}$$

Consider the finite element analysis task as depicted in fig. 3. Let now an individual element ${}^e\Omega$ be considered. The approximate solution ${}^eT^h(x)$ on this element shall be the sum of products between element shape functions and element unknowns; in vector form, this can be written as

$${}^eT^h(x) = {}^e\boldsymbol{H}(x){}^e\boldsymbol{q}, \tag{3.9}$$

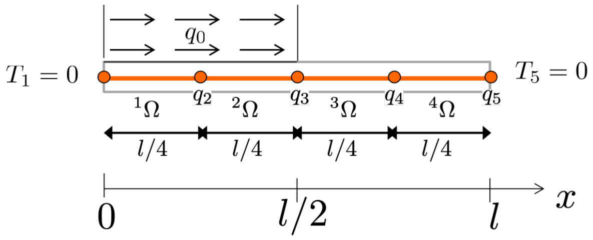


Figure 3: Visualization of problem 2.

where ${}^e\boldsymbol{H}(x) = ({}^eh_1(x), \ldots, {}^eh_{{}^ep}(x))$ is a vector of element shape functions, where ${}^ep$ is the number of nodes for the element e. Furthermore, ${}^e\boldsymbol{q} = ({}^eq_1, \ldots, {}^eq_{{}^ep})^\top$ is a vector of unknowns for each element e. The construction of the approximated solution on the complete space - not just on ${}^e\Omega$ - is achieved by assembling the element approximations. This requires a mapping between the global node numbering $\boldsymbol{q}$ and the local node vector ${}^e\boldsymbol{q}$. This is done by the localization matrix ${}^e\mathbf{L}$, which establishes the relation

$$ {}^e\boldsymbol{q} = {}^e\mathbf{L}\boldsymbol{q}, \qquad {}^eL_{ij} = \begin{cases} 1, & \text{global node } j \text{ equal to local node } i \\ 0, & \text{otherwise} \end{cases}. \tag{3.10} $$

One can now write $T^h(x)$ and $\delta T^h(x)$ as

$$ T^h(x) = \sum_{e=1}^{m} {}^eT^h(x) = \sum_{e=1}^{m} {}^e\boldsymbol{H}(x){}^e\boldsymbol{q} = \sum_{e=1}^{m} {}^e\boldsymbol{H}(x){}^e\mathbf{L}\boldsymbol{q} \tag{3.11} $$

and

$$ \delta T^h(x) = \sum_{e=1}^{m} {}^e\delta T^h(x) = \sum_{e=1}^{m} {}^e\boldsymbol{H}(x){}^e\delta\boldsymbol{q} = \sum_{e=1}^{m} {}^e\boldsymbol{H}(x){}^e\mathbf{L}\delta\boldsymbol{q}. \tag{3.12} $$

Inserting this into eq. (3.8) leads to

$$ \sum_{e=1}^{m} \left[\int_{{}^e\Omega} k \frac{\mathrm{d}{}^eT^h(x)}{\mathrm{d}x} \frac{\mathrm{d}{}^e\delta T^h(x)}{\mathrm{d}x} \, \mathrm{d}x \right] = \sum_{e=1}^{m} \left[\int_{{}^e\Omega} {}^eq(x){}^e\delta T^h(x) \, \mathrm{d}x \right], \tag{3.13} $$

where ${}^eq(x)$ is the function $q(x)$ on each element. Using the defined equations and quantities, one obtains after several simplifications

$$ \sum_{e=1}^{m} \delta\boldsymbol{q}^\top {}^e\mathbf{L}^\top \left[\int_{{}^e\Omega} k{}^e\boldsymbol{B}^\top {}^e\boldsymbol{B} \, \mathrm{d}x \right] {}^e\mathbf{L}\boldsymbol{q} = \sum_{e=1}^{m} \delta\boldsymbol{q}^\top {}^e\mathbf{L}^\top \left[\int_{{}^e\Omega} {}^eq{}^e\boldsymbol{H}^\top \, \mathrm{d}x \right], \tag{3.14} $$

where ${}^e\boldsymbol{B} = \frac{\mathrm{d}{}^e\boldsymbol{H}}{\mathrm{d}x}$. Because the factor $\delta\boldsymbol{q}^\top$ is arbitrary and the same for both sides of the equation, it can be drawn out of the sum and subsequently left away, hence one obtains

$$ \mathbf{K}\boldsymbol{q} = \boldsymbol{r}, \qquad \mathbf{K} = \sum_{e=1}^{m} {}^e\mathbf{L}^\top {}^e\mathbf{K}{}^e\mathbf{L}, \qquad \boldsymbol{r} = \sum_{e=1}^{m} {}^e\mathbf{L}^\top {}^e\boldsymbol{r}, \tag{3.15} $$

where

$$ {}^e\mathbf{K} = \int_{{}^e\Omega} k{}^e\boldsymbol{B}^\top {}^e\boldsymbol{B} \, \mathrm{d}x, \qquad {}^e\boldsymbol{r} = \int_{{}^e\Omega} {}^eq{}^e\boldsymbol{H}^\top \, \mathrm{d}x. \tag{3.16} $$

An element ${}^eK_{ij}$ of the local stiffness matrix ${}^e\mathbf{K}$ is given by

$$ {}^eK_{ij} = \int_{{}^e\Omega} k \frac{\mathrm{d}{}^eh_i(x)}{\mathrm{d}x} \frac{\mathrm{d}{}^eh_j(x)}{\mathrm{d}x} \, \mathrm{d}x, \tag{3.17} $$

and a component ${}^{e}r_i$ of the vector ${}^{e}\boldsymbol{r}$ is calculated as

$$ {}^{e}r_i = \int_{{}^{e}\Omega} {}^{e}q(x)\,{}^{e}h_i(x)\,\mathrm{d}x. \tag{3.18}$$

In order to perform those integrals, the element shape functions ${}^{e}h_i(x)$ need to be defined. In the case of the problem at hand, the element shape functions are given according to fig. 4. There are two nodes per element, hence there are two shape functions

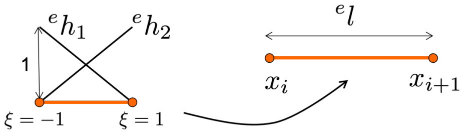


Figure 4: Shape functions for one element in problem 2.

$$ {}^{e}h_1(\xi) = \frac{1}{2}(1-\xi), \qquad {}^{e}h_2(\xi) = \frac{1}{2}(1+\xi) \tag{3.19}$$

defined using the local coordinate ξ. These shape functions ensure, that the function ${}^{e}h_1$ is one at the first node and zero at the second node, and the opposite for ${}^{e}h_2$. The transformation $\xi \to x$ is given by the equation

$$ \xi(x) = \frac{2x - x_i - x_{i+1}}{{}^{e}l}, \qquad {}^{e}l = x_{i+1} - x_i, \tag{3.20}$$

where x_i denotes the global location of the global node i. Hence, ${}^{e}h_1(x)$ and ${}^{e}h_2(x)$ are given by

$$ {}^{e}h_1(x) = \frac{x_{i+1} - x}{x_{i+1} - x_i}, \qquad {}^{e}h_2(x) = \frac{x - x_i}{x_{i+1} - x_i}. \tag{3.21}$$

With this, or using the chain rule as

$$ \frac{\mathrm{d}\,{}^{e}h_i(\xi(x))}{\mathrm{d}x} = \frac{\partial\,{}^{e}h_i(\xi(x))}{\partial \xi}\frac{\mathrm{d}\xi(x)}{\mathrm{d}x}, \quad i \in \{1,2\}, \tag{3.22}$$

the integrals eq. (3.17) and eq. (3.18) can be performed. In fact, we have

$$ {}^{e}K_{11} = \int_{{}^{e}\Omega} k \left(\frac{\partial\,{}^{e}h_1}{\partial\xi}\right)^2 \left(\frac{\mathrm{d}\xi}{\mathrm{d}x}\right)^2 \mathrm{d}x = \frac{k}{{}^{e}l}\int_{{}^{e}\Omega}\mathrm{d}x = \frac{k}{{}^{e}l}\int_{-1}^{1}\frac{{}^{e}l}{2}\,\mathrm{d}\xi = \frac{k}{{}^{e}l}, \tag{3.23}$$

$$ {}^{e}K_{12} = {}^{e}K_{21} = \int_{{}^{e}\Omega} k \left(\frac{\partial\,{}^{e}h_1}{\partial\xi}\right)\left(\frac{\partial\,{}^{e}h_2}{\partial\xi}\right) \left(\frac{\mathrm{d}\xi}{\mathrm{d}x}\right)^2 \mathrm{d}x = -\frac{k}{{}^{e}l}\int_{{}^{e}\Omega}\mathrm{d}x = -\frac{k}{{}^{e}l}\int_{-1}^{1}\frac{{}^{e}l}{2}\,\mathrm{d}\xi = -\frac{k}{{}^{e}l}, \tag{3.24}$$

$$ {}^{e}K_{22} = \int_{{}^{e}\Omega} k \left(\frac{\partial\,{}^{e}h_2}{\partial\xi}\right)^2 \left(\frac{\mathrm{d}\xi}{\mathrm{d}x}\right)^2 \mathrm{d}x = \frac{k}{{}^{e}l}\int_{{}^{e}\Omega}\mathrm{d}x = \frac{k}{{}^{e}l}\int_{-1}^{1}\frac{{}^{e}l}{2}\,\mathrm{d}\xi = \frac{k}{{}^{e}l}. \tag{3.25}$$

Hence, the local stiffness matrix ${}^{e}\mathbf{K}$ is given by

$$ {}^{e}\mathbf{K} = \frac{k}{{}^{e}l}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \qquad {}^{e}l = l/4. \tag{3.26}$$

Using $\mathbf{K} = \sum_{e=1}^{4} {}^{e}\mathbf{L}^{\top}\,{}^{e}\mathbf{K}\,{}^{e}\mathbf{L}$, one obtains the global stiffness matrix $\mathbf{K}$ as

$$ \mathbf{K} = \frac{4k}{l}\begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}. \tag{3.27}$$

Taking into account the information present in fig. 3, the components ${}^e r_i$ for $i \in \{1, 2\}$ are similarly calculated as

$$
\begin{aligned}
{}^1 r_1 &= \int_{{}^1\Omega} {}^1 q(x) {}^1 h_1(x) \,\mathrm{d}x = \frac{q_0}{2} \int_0^{l/4} \left(2 - \frac{8x}{l}\right) \mathrm{d}x = \frac{q_0 l}{8}, && (3.28) \\
{}^1 r_2 &= \int_{{}^1\Omega} {}^1 q(x) {}^1 h_2(x) \,\mathrm{d}x = \frac{q_0}{2} \int_0^{l/4} \left(\frac{8x}{l}\right) \mathrm{d}x = \frac{q_0 l}{8}, && (3.29) \\
{}^2 r_1 &= \int_{{}^2\Omega} {}^2 q(x) {}^2 h_1(x) \,\mathrm{d}x = \frac{q_0}{2} \int_{l/4}^{l/2} \left(4 - \frac{8x}{l}\right) \mathrm{d}x = \frac{q_0 l}{8}, && (3.30) \\
{}^2 r_2 &= \int_{{}^2\Omega} {}^2 q(x) {}^2 h_2(x) \,\mathrm{d}x = \frac{q_0}{2} \int_{l/4}^{l/2} \left(-2 + \frac{8x}{l}\right) \mathrm{d}x = \frac{q_0 l}{8}, && (3.31) \\
{}^3 r_1 &= {}^3 r_2 = {}^4 r_1 = {}^4 r_2 = 0. && (3.32)
\end{aligned}
$$

Using $\boldsymbol{r} = \sum_{e=1}^4 {}^e\mathbf{L}^\top {}^e\boldsymbol{r}$, the global vector $\boldsymbol{r}$ is given by

$$
\boldsymbol{r} = \frac{q_0 l}{8} \begin{pmatrix} 1 \\ 2 \\ 1 \\ 0 \\ 0 \end{pmatrix}. \tag{3.33}
$$

In total therefore, the system $\mathbf{K}\boldsymbol{q} = \boldsymbol{r}$ to be solved amounts to

$$
\frac{k}{q_0 l^2} \begin{pmatrix} 32 & -32 & 0 & 0 & 0 \\ -32 & 64 & -32 & 0 & 0 \\ 0 & -32 & 64 & -32 & 0 \\ 0 & 0 & -32 & 64 & -32 \\ 0 & 0 & 0 & -32 & 32 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 0 \\ 0 \end{pmatrix}. \tag{3.34}
$$

Since the boundary conditions imply $q_1 = q_5 = 0$, we can leave away the first and last row and column of the problem; thus we need to solve

$$
\frac{k}{q_0 l^2} \begin{pmatrix} 64 & -32 & 0 \\ -32 & 64 & -32 \\ 0 & -32 & 64 \end{pmatrix} \begin{pmatrix} q_2 \\ q_3 \\ q_4 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix}, \tag{3.35}
$$

from which the solution is obtained as

$$
\begin{pmatrix} q_2 \\ q_3 \\ q_4 \end{pmatrix} = \left[\frac{k}{q_0 l^2} \begin{pmatrix} 64 & -32 & 0 \\ -32 & 64 & -32 \\ 0 & -32 & 64 \end{pmatrix} \right]^{-1} \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix}. \tag{3.36}
$$

The temperature $T(q_i)$ at node i with $i \in \{1, 2, \ldots, 5\}$ is then given by means of $T(q_i) = q_i$. Using this and knowing the coordinates of node i in the mesh, one obtains an approximate solution for the temperature (field) $T(x)$. The obtained solution for the developed finite element model (FEM) and the associated analytical solution to problem 2 are depicted in fig. 5.

4 2D continuous system

6. **Q** — How does an example application of the static 2D continuous FEM method work?

A — Consider the problem as seen in fig. 6. The problem consists of a square plate with uniform

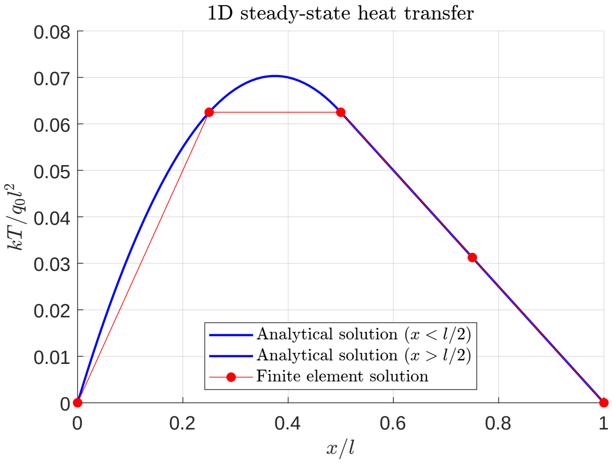


Figure 5: Analytical and FEM solutions to problem 2 as depicted in fig. 3.

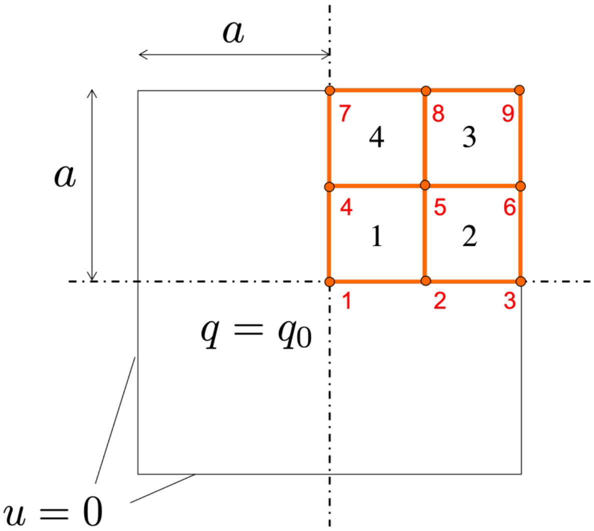


Figure 6: Visualization of the problem to be solved using the finite element method.

conductivity k. There is a constant heat source $q(x, y) = q_0$ all over the plate, and the edges are held at constant temperature zero.

The governing equation for this problem is given by

$$k\boldsymbol{\nabla}^2 u(x, y) + q(x, y) = 0, \tag{4.1}$$

where it is assumed that the conductivity k is uniform and constant over the domain of interest. Hereby, $u(x, y)$ is the temperature field to be approximated using the finite element method in 2D.

For this purpose, the weak form of the strong form eq. (4.1) has to be derived first. The strategy is the same as in 1D; one multiplies the strong form equation by a weight function $\delta u(x, y)$ and integrates it over the domain Ω of interest:

$$\int_\Omega \left[k\boldsymbol{\nabla}^2 u(x, y) + q(x, y)\right] \delta u(x, y)\,\mathrm{d}x\,\mathrm{d}y = 0, \qquad \forall\ \delta u(x, y). \tag{4.2}$$

This equation must be valid for any $\delta u(x, y)$ for it to be equivalent to the strong form eq. (4.1) and thus to be a sound weak form of the problem. First of all, recall the Gauss theorem

$$\int_\Omega (\boldsymbol{\nabla} \cdot \boldsymbol{F})\,\mathrm{d}V_\Omega = \int_{\partial\Omega} \boldsymbol{F} \cdot \boldsymbol{n}_\Omega\,\mathrm{d}S_\Omega, \tag{4.3}$$

where Ω is the m-dimensional domain, $\partial\Omega$ is its boundary, $\mathrm{d}V_\Omega$ is the infinitesimal volume element of Ω, $\mathrm{d}S_\Omega$ is the infinitesimal surface element of Ω and $\boldsymbol{n}_\Omega$ is a unit vector pointing everywhere perpendicularly outwards on $\partial\Omega$. Now, for some functions $f(x,y)$ and $g(x,y)$ we obtain $\boldsymbol{\nabla}(fg) = (\boldsymbol{\nabla} f)g + f(\boldsymbol{\nabla} g)$ by the product rule. Hence, we also obtain $f(\boldsymbol{\nabla} g) = \boldsymbol{\nabla}(fg) - (\boldsymbol{\nabla} f)g$. Defining $f \doteq \delta u$ and $g = k\boldsymbol{\nabla} u$, we thus get

$$\delta u k \boldsymbol{\nabla}^2 u = \boldsymbol{\nabla}\left(\delta u k \boldsymbol{\nabla} u\right) - (\boldsymbol{\nabla}\delta u)k\boldsymbol{\nabla} u. \tag{4.4}$$

Using this result and the Gauss theorem stated above, one can perform the simplifications

$$\begin{aligned} 0 &= \int_\Omega \left[k\boldsymbol{\nabla}^2 u + q\right] \delta u \,\mathrm{d}x\,\mathrm{d}y \\ &= \int_\Omega \boldsymbol{\nabla}\left(\delta u k \boldsymbol{\nabla} u\right)\,\mathrm{d}x\,\mathrm{d}y - \int_\Omega k(\boldsymbol{\nabla} u)(\boldsymbol{\nabla}\delta u)\,\mathrm{d}x\,\mathrm{d}y + \int_\Omega q\delta u\,\mathrm{d}x\,\mathrm{d}y \\ &= -\int_\Omega k(\boldsymbol{\nabla} u)(\boldsymbol{\nabla}\delta u)\,\mathrm{d}x\,\mathrm{d}y + \int_\Omega q\delta u\,\mathrm{d}x\,\mathrm{d}y, \end{aligned} \tag{4.5}$$

where

$$\int_\Omega \boldsymbol{\nabla}\left(\delta u k\boldsymbol{\nabla} u\right)\,\mathrm{d}V_\Omega = \int_{\partial\Omega} (\delta u k \boldsymbol{\nabla} u)\cdot \boldsymbol{n}_\Omega\,\mathrm{d}S_\Omega = 0, \qquad \mathrm{d}V_\Omega = \mathrm{d}x\,\mathrm{d}y \tag{4.6}$$

because of the boundary condition $u(x,y) = 0$ and hence also $\delta u(x,y) = 0$ at the boundary $\partial\Omega$ of the domain Ω. Exploiting the constant conductivity k and the boundary condition $q(x,y) = q_0$ one obtains the weak form of the problem as

$$k\int_\Omega \left[\boldsymbol{\nabla} u(x,y)\right]\left[\boldsymbol{\nabla}\delta u(x,y)\right]\,\mathrm{d}x\,\mathrm{d}y = q_0\int_\Omega \delta u(x,y)\,\mathrm{d}x\,\mathrm{d}y \qquad \forall\ \delta u(x,y). \tag{4.7}$$

Now, the function $u(x,y)$ is approximated by a sum of nodal constant unknowns q_i and shape functions $h_i(x,y)$ multiplied by them, namely

$$u(x,y) \approx u^h(x,y) = \sum_{i=1}^{p} h_i(x,y)q_i = \boldsymbol{H}(x,y)\boldsymbol{q}, \tag{4.8}$$

where $\boldsymbol{H} = (h_1(x,y),\ldots,h_p(x,y))$. Similarly, we have

$$\delta u(x,y) \approx \delta u^h(x,y) = \sum_{i=1}^{p} h_i(x,y)\delta q_i = \boldsymbol{H}(x,y)\boldsymbol{\delta q}, \tag{4.9}$$

where $\boldsymbol{\delta q} = (\delta q_1,\ldots,\delta q_p)^\top$. Note that for a vector $\boldsymbol{b} = (b_1,\ldots,b_p)$ and a constant vector $\boldsymbol{a} = (a_1,\ldots,a_p)^\top$ it holds that

$$\boldsymbol{\nabla}(\boldsymbol{a}^\top\boldsymbol{b}^\top) = \boldsymbol{a}^\top(\boldsymbol{\nabla}\boldsymbol{b})^\top, \tag{4.10}$$

where $\boldsymbol{\nabla}\boldsymbol{b}$ is a $d\times p$ matrix, if d is the dimensionality of the space, i.e. $\boldsymbol{\nabla} = (\partial_1,\ldots,\partial_d)^\top$. Inserting now eq. (4.8) and eq. (4.9) into eq. (4.7) yields

$$\int_\Omega k\boldsymbol{\nabla}\left(\boldsymbol{Hq}\right)\boldsymbol{\nabla}\left(\boldsymbol{H\delta q}\right)\,\mathrm{d}x\,\mathrm{d}y = \int_\Omega q_0\boldsymbol{H\delta q}\,\mathrm{d}x\,\mathrm{d}y. \tag{4.11}$$

Since $\boldsymbol{Hq}$ and $\boldsymbol{H\delta q}$ are scalars, it holds that

$$\boldsymbol{Hq} = (\boldsymbol{Hq})^\top = \boldsymbol{q}^\top\boldsymbol{H}^\top, \qquad \boldsymbol{H\delta q} = (\boldsymbol{H\delta q})^\top = \boldsymbol{\delta q}^\top\boldsymbol{H}^\top \tag{4.12}$$

and furthermore

$$\boldsymbol{\nabla}(\boldsymbol{q}^\top\boldsymbol{H}^\top) = \boldsymbol{q}^\top(\boldsymbol{\nabla H})^\top, \qquad \boldsymbol{\nabla}(\boldsymbol{\delta q}^\top\boldsymbol{H}^\top) = \boldsymbol{\delta q}^\top(\boldsymbol{\nabla H})^\top, \qquad \boldsymbol{\nabla}(\boldsymbol{Hq}) = (\boldsymbol{\nabla H})\boldsymbol{q}. \tag{4.13}$$

Therefore, we can rewrite the above equation as

$$\delta \boldsymbol{q}^\top \left[\int_\Omega k(\boldsymbol{\nabla H})^\top (\boldsymbol{\nabla H}) \, \mathrm{d}x \, \mathrm{d}y \right] \boldsymbol{q} = \delta \boldsymbol{q}^\top \int_\Omega q_0 \boldsymbol{H}^\top \, \mathrm{d}x \, \mathrm{d}y. \tag{4.14}$$

Redefining $\boldsymbol{\nabla H} \doteq \mathbf{B}$ and leaving away the constant factor $\delta \boldsymbol{q}^\top$ both on the right hand and left hand sides of the above equation yields

$$\underbrace{\left[\int_\Omega k \mathbf{B}^\top \mathbf{B} \, \mathrm{d}x \, \mathrm{d}y \right]}_{\doteq \mathbf{K}} \boldsymbol{q} = \underbrace{\int_\Omega q_0 \boldsymbol{H}^\top \, \mathrm{d}x \, \mathrm{d}y}_{\doteq \boldsymbol{r}} \quad \Leftrightarrow \quad \mathbf{K} \boldsymbol{q} = \boldsymbol{r}. \tag{4.15}$$

One can write the above equation also using local shape functions instead of global ones; then the above equation reads as

$$\underbrace{\left[\int_{{}^e\Omega} k \, {}^e\mathbf{B}(x,y)^\top \, {}^e\mathbf{B}(x,y) \, \mathrm{d}x \, \mathrm{d}y \right]}_{\doteq {}^e\mathbf{K}} {}^e\boldsymbol{q} = \underbrace{\int_{{}^e\Omega} q(x,y) \, {}^e\boldsymbol{H}(x,y)^\top \, \mathrm{d}x \, \mathrm{d}y}_{\doteq {}^e\boldsymbol{r}} \tag{4.16}$$

with $\mathbf{K} = \sum_{e=1}^m {}^e\mathbf{L}^\top \, {}^e\mathbf{K} \, {}^e\mathbf{L}$ and $\boldsymbol{r} = \sum_{e=1}^m {}^e\mathbf{L}^\top \, {}^e\boldsymbol{r}$, where m is the total element number in the discretized system and ${}^e\mathbf{L}$ is the localization matrix of element e defined as

$$ {}^e\boldsymbol{q} = {}^e\mathbf{L}\boldsymbol{q}, \qquad {}^eL_{ij} = \begin{cases} 1, & \text{global node } j \text{ equal to local node } i \\ 0, & \text{otherwise} \end{cases}. \tag{4.17}$$

Consider fig. 7. The local shape functions $h_j(\xi, \eta)$ of such an element are given by

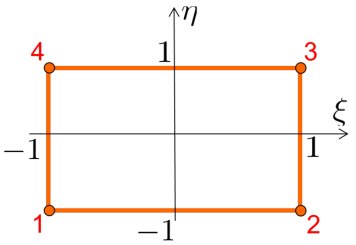


Figure 7: Local node numbering and coordinate system of a 2D rectangular finite element.

$$\begin{aligned} h_1(\xi,\eta) &= \frac{1}{4}(1-\xi)(1-\eta), \qquad & h_2(\xi,\eta) &= \frac{1}{4}(1+\xi)(1-\eta), \\ h_3(\xi,\eta) &= \frac{1}{4}(1+\xi)(1+\eta), \qquad & h_4(\xi,\eta) &= \frac{1}{4}(1-\xi)(1+\eta). \end{aligned} \tag{4.18}$$

These shape functions each define a plane in 3D space, where $h_j(\xi,\eta)$ is one at node j and zero at all the other nodes of the element. Now, the transformation ${}^eT$ of an element's coordinates to the global coordinates is given by

$$ {}^eT : \mathbb{R}^2 \to \mathbb{R}^2, \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} {}^ex(\xi,\eta) = \sum_{i=1}^4 h_i(\xi,\eta) x_i \\ {}^ey(\xi,\eta) = \sum_{i=1}^4 h_i(\xi,\eta) y_i \end{pmatrix} \tag{4.19}$$

where x_i and y_i denote the global coordinates of the local node i of element e.

We now calculate ${}^1T$ for the first element as shown in fig. 6. We have

$$ {}^1T : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} {}^1x(\xi,\eta) = h_2(\xi,\eta)\frac{a}{2} + h_3(\xi,\eta)\frac{a}{2} = \frac{a}{4}(1+\xi) \\ {}^1y(\xi,\eta) = h_3(\xi,\eta)\frac{a}{2} + h_4(\xi,\eta)\frac{a}{2} = \frac{a}{4}(1+\eta) \end{pmatrix}. \tag{4.20}$$

Now, the Jacobian $\mathbf{J}_{{}^eT}$ of the transformation ${}^eT$ is given by

$$\mathbf{J}_{{}^eT} = \begin{pmatrix} \frac{\partial^e x}{\partial \xi} & \frac{\partial^e y}{\partial \xi} \\ \frac{\partial^e x}{\partial \eta} & \frac{\partial^e y}{\partial \eta} \end{pmatrix}, \qquad \begin{pmatrix} \partial_\xi \\ \partial_\eta \end{pmatrix} = \begin{pmatrix} \frac{\partial^e x}{\partial \xi} & \frac{\partial^e y}{\partial \xi} \\ \frac{\partial^e x}{\partial \eta} & \frac{\partial^e y}{\partial \eta} \end{pmatrix} \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix} = \mathbf{J}_{{}^eT} \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix} \tag{4.21}$$

and hence also

$$\begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix} = \mathbf{J}^{-1}_{{}^eT} \begin{pmatrix} \partial_\xi \\ \partial_\eta \end{pmatrix}. \tag{4.22}$$

We have in particular

$$\frac{\partial^1 x}{\partial \xi} = \frac{a}{4}, \quad \frac{\partial^1 x}{\partial \eta} = 0, \quad \frac{\partial^1 y}{\partial \xi} = 0, \quad \frac{\partial^1 y}{\partial \eta} = \frac{a}{4} \tag{4.23}$$

and thus the Jacobian

$$\mathbf{J}_{{}^1T} = \begin{pmatrix} \frac{a}{4} & 0 \\ 0 & \frac{a}{4} \end{pmatrix}, \qquad \mathbf{J}^{-1}_{{}^1T} = \begin{pmatrix} \frac{4}{a} & 0 \\ 0 & \frac{4}{a} \end{pmatrix}, \qquad \det\left[\mathbf{J}_{{}^1T}\right] = \frac{a^2}{16}. \tag{4.24}$$

Thus, the differential area element $\mathrm{d}x\,\mathrm{d}y$ transforms as $\mathrm{d}x\,\mathrm{d}y = \det\left[\mathbf{J}_{{}^eT}\right]\mathrm{d}\xi\,\mathrm{d}\eta$ for the coordinate transformation from global coordinates (x, y) to the local coordinates (ξ, η) of element e.

Now, the B-matrix ${}^e\mathbf{B}$ is defined as

$$\boldsymbol{\nabla}^e\boldsymbol{H} = \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix} (h_1(x, y), \ldots, h_4(x, y)) = \begin{pmatrix} \partial_x{}^e h_1 & \partial_x{}^e h_2 & \partial_x{}^e h_3 & \partial_x{}^e h_4 \\ \partial_y{}^e h_1 & \partial_y{}^e h_2 & \partial_y{}^e h_3 & \partial_y{}^e h_4 \end{pmatrix}, \tag{4.25}$$

where the notation ${}^a h_i = h_i(\xi, \eta)$ and ${}^e h_i = h_i(x, y)$ was adopted. We can further write

$$\partial_x{}^e h_i = \frac{\partial^a h_i}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial^a h_i}{\partial \eta}\frac{\partial \eta}{\partial x}, \qquad \partial_y{}^e h_i = \frac{\partial^a h_i}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial^a h_i}{\partial \eta}\frac{\partial \eta}{\partial y}. \tag{4.26}$$

For the element $e = 1$, the derivatives are $\partial_x \xi = \frac{4}{a}$, $\partial_x \eta = 0$, $\partial_y \xi = 0$ and $\partial_y \eta = \frac{4}{a}$. Calculating furthermore

$$\begin{array}{rclcrcl} \frac{\partial^a h_1}{\partial \xi} &=& -\frac{1}{4}(1-\eta) & \qquad & \frac{\partial^a h_1}{\partial \eta} &=& -\frac{1}{4}(1-\xi) \\ \frac{\partial^a h_2}{\partial \xi} &=& \frac{1}{4}(1-\eta) & & \frac{\partial^a h_2}{\partial \eta} &=& -\frac{1}{4}(1+\xi) \\ \frac{\partial^a h_3}{\partial \xi} &=& \frac{1}{4}(1+\eta) & & \frac{\partial^a h_3}{\partial \eta} &=& \frac{1}{4}(1+\xi) \\ \frac{\partial^a h_4}{\partial \xi} &=& -\frac{1}{4}(1+\eta) & & \frac{\partial^a h_4}{\partial \eta} &=& \frac{1}{4}(1-\xi) \end{array} \tag{4.27}$$

one can assemble the B-matrix in local coordinates as

$${}^a\mathbf{B} = \frac{1}{a}\begin{pmatrix} -(1-\eta) & (1-\eta) & (1+\eta) & -(1+\eta) \\ -(1-\xi) & -(1+\xi) & (1+\xi) & (1-\xi) \end{pmatrix}. \tag{4.28}$$

Recall, that the element stiffness matrix ${}^e\mathbf{K}$ for the problem at hand is given as

$${}^e\mathbf{K} = \int_{{}^e\Omega} k\,{}^e\mathbf{B}^\top{}^e\mathbf{B}\,\mathrm{d}x\,\mathrm{d}y = \int_{{}^a\Omega} k\,{}^a\mathbf{B}^\top{}^a\mathbf{B}\det\left[\mathbf{J}_{{}^eT}\right]\mathrm{d}\xi\,\mathrm{d}\eta. \tag{4.29}$$

For a product of matrices $\mathbf{XY}$ the entry $[\mathbf{XY}]_{ij}$ is given by $[\mathbf{XY}]_{ij} = \sum_k X_{ik}Y_{kj}$. Considering the element $e = 1 = a$, we calculate now the elements ${}^1K_{11}$ and ${}^1K_{13}$. The former element is calculated as

$$\begin{aligned} {}^1K_{11} &= \int_{{}^a\Omega} k\left[{}^a\mathbf{B}^\top{}^a\mathbf{B}\right]_{11}\det\left[\mathbf{J}_{{}^1T}\right]\mathrm{d}\xi\,\mathrm{d}\eta = \frac{ka^2}{16}\int_{\xi=-1}^{1}\int_{\eta=-1}^{1}\left[{}^a\mathbf{B}^\top{}^a\mathbf{B}\right]_{11}\mathrm{d}\xi\,\mathrm{d}\eta \\ &= \frac{ka^2}{16}\int_{\xi=-1}^{1}\int_{\eta=-1}^{1}\left[\sum_{k=1}^{2}{}^aB_{k1}{}^aB_{k1}\right]\mathrm{d}\xi\,\mathrm{d}\eta = \frac{ka^2}{16}\int_{\xi=-1}^{1}\int_{\eta=-1}^{1}\left[{}^aB_{11}^2 + {}^aB_{21}^2\right]\mathrm{d}\xi\,\mathrm{d}\eta \\ &= \frac{ka^2}{16}\int_{\xi=-1}^{1}\int_{\eta=-1}^{1}\left[\frac{1}{a^2}(1-\eta)^2 + \frac{1}{a^2}(1-\xi)^2\right]\mathrm{d}\xi\,\mathrm{d}\eta \\ &= \frac{k}{16}\int_{\eta=-1}^{1}\left(\frac{14}{3} - 4\eta + 2\eta^2\right)\mathrm{d}\eta = \frac{2k}{3}. \end{aligned} \tag{4.30}$$

The element ${}^1K_{13}$ is then calculated as

$$
\begin{aligned}
{}^1K_{13} &= \int_{{}^a\Omega} k \left[{}^a\mathbf{B}^\top{}^a\mathbf{B}\right]_{13} \det\left[\mathbf{J}_{{}^1T}\right] \mathrm{d}\xi\,\mathrm{d}\eta = \frac{ka^2}{16}\int_{\xi=-1}^{1}\int_{\eta=-1}^{1} \left[{}^a\mathbf{B}^\top{}^a\mathbf{B}\right]_{13} \mathrm{d}\xi\,\mathrm{d}\eta \\
&= \frac{ka^2}{16}\int_{\xi=-1}^{1}\int_{\eta=-1}^{1} \left[\sum_{k=1}^{2} {}^aB_{k1}{}^aB_{k3}\right] \mathrm{d}\xi\,\mathrm{d}\eta \\
&= -\frac{k}{16}\int_{\xi=-1}^{1}\int_{\eta=-1}^{1} \left[(1-\eta)(1+\eta)+(1-\xi)(1+\xi)\right] \mathrm{d}\xi\,\mathrm{d}\eta = -\frac{k}{3}.
\end{aligned}
\tag{4.31}
$$

Calculating all the elements of ${}^1\mathbf{K}$ in the same way one obtains

$$
{}^1\mathbf{K} = \frac{k}{6}\begin{pmatrix} 4 & -1 & -2 & -1 \\ -1 & 4 & -1 & -2 \\ -2 & -1 & 4 & -1 \\ -1 & -2 & -1 & 4 \end{pmatrix}. \tag{4.32}
$$

One can see that the formerly calculated elements ${}^1K_{11}$ and ${}^1K_{13}$ coincide with the elements in that matrix.

Recall from above, that the element load vector is defined as

$$
{}^e\boldsymbol{r} = \int_{{}^e\Omega} q(x,y){}^e\boldsymbol{H}(x,y)^\top \,\mathrm{d}x\,\mathrm{d}y = \int_{{}^a\Omega} {}^aq(\xi,\eta){}^a\boldsymbol{H}(\xi,\eta)^\top \det\left[\mathbf{J}_{{}^eT}\right] \mathrm{d}\xi\,\mathrm{d}\eta. \tag{4.33}
$$

The vector ${}^a\boldsymbol{H}(\xi,\eta)$ is given by

$$
{}^a\boldsymbol{H}(\xi,\eta)^\top = \frac{1}{4}\begin{pmatrix} (1-\xi)(1-\eta) \\ (1+\xi)(1-\eta) \\ (1+\xi)(1+\eta) \\ (1-\xi)(1+\eta) \end{pmatrix}. \tag{4.34}
$$

for the element $e = 1 = a$. Hence, the component ${}^1r_2$ of the load vector ${}^1\boldsymbol{r}$ is calculated as

$$
\begin{aligned}
{}^1r_2 &= \frac{q_0a^2}{64}\int_{{}^a\Omega}(1+\xi)(1-\eta)\,\mathrm{d}\xi\,\mathrm{d}\eta = \frac{q_0a^2}{64}\int_{\eta=-1}^{1}\int_{\xi=-1}^{1}(1+\xi)(1-\eta)\,\mathrm{d}\xi\,\mathrm{d}\eta \\
&= \frac{q_0a^2}{64}\int_{\xi=-1}^{1}(2+2\xi)\,\mathrm{d}\xi = \frac{q_0a^2}{16}.
\end{aligned}
\tag{4.35}
$$

Calculating all the other components ${}^1r_i$ for $i \in \{1,2,3,4\}$ one arrives at the element load vector

$$
{}^1\boldsymbol{r} = \frac{q_0a^2}{16}(1,1,1,1)^\top. \tag{4.36}
$$

Now, the global load vector $\boldsymbol{r}$ will have nine components, as there are nine global nodes. In order to assemble the global load vector, we build a connectivity table: Based on this connectivity

Element	GN for LN1	GN for LN2	GN for LN3	GN for LN4
1	1	2	5	4
2	2	3	6	5
3	5	6	9	8
4	4	5	8	7

Table 3: Connectivity table the heat transfer problem as visualized in fig. 6. The table shows the global node (GN) associated to local nodes (LN).

table, one can now build the global load vector $\boldsymbol{r}$ as

$$\boldsymbol{r} = \frac{q_0 a^2}{16} \begin{pmatrix} 1 \\ 1+1 \\ 1 \\ 1+1 \\ 1+1+1+1 \\ 1+1 \\ 1 \\ 1+1 \\ 1 \end{pmatrix} = \frac{q_0 a^2}{16} \begin{pmatrix} 1 \\ 2 \\ 1 \\ 2 \\ 4 \\ 2 \\ 1 \\ 2 \\ 1 \end{pmatrix}. \tag{4.37}$$

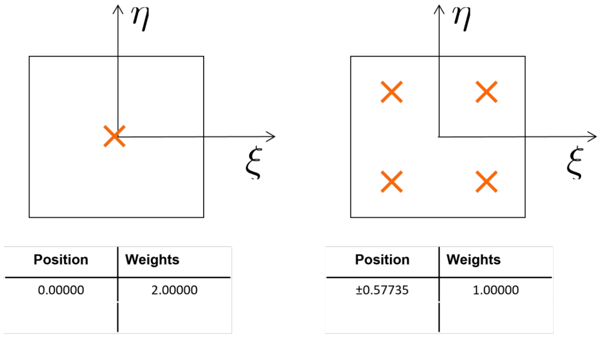
Using Gauss integration, one can approximate the integral to calculate the element stiffness matrix ${}^e\mathbf{K}$ and element load vector ${}^e\boldsymbol{r}$ as

$$\begin{aligned} {}^e\mathbf{K} &= \int_{{}^a\Omega} k\,{}^a\mathbf{B}^\top(\xi,\eta)\,{}^a\mathbf{B}(\xi,\eta)\det\left[\mathbf{J}_{{}^eT}(\xi,\eta)\right]\,\mathrm{d}\xi\,\mathrm{d}\eta \\ &\approx \sum_{j=1}^{s} \omega(j)\,{}^a\mathbf{B}^\top[\xi(j),\eta(j)]\,{}^a\mathbf{B}[\xi(j),\eta(j)]\det\left(\mathbf{J}_{{}^eT}[\xi(j),\eta(j)]\right) \end{aligned} \tag{4.38}$$

and

$$\begin{aligned} {}^e\boldsymbol{r} &= \int_{{}^a\Omega} {}^a q(\xi,\eta)\,{}^a\boldsymbol{H}^\top(\xi,\eta)\det\left[\mathbf{J}_{{}^eT}(\xi,\eta)\right]\,\mathrm{d}\xi\,\mathrm{d}\eta \\ &\approx \sum_{j=1}^{s} \omega(j)\,{}^a q\left[\xi(j),\eta(j)\right]\,{}^a\boldsymbol{H}^\top[\xi(j),\eta(j)]\det\left(\mathbf{J}_{{}^eT}[\xi(j),\eta(j)]\right), \end{aligned} \tag{4.39}$$

where $\omega(j)$, $\xi(j)$ and $\eta(j)$ are the weights and element coordinate values at the integration point $j \in \{1,\ldots,s\}$ for each element. What is now needed in order to implement the calculations is expressing the B-matrix, the H-vector and the Jacobian in terms of local coordinates $(\xi(j),\eta(j))$ at the integration points. Furthermore, the weights $\omega(j)$ need to be known for a chosen geometry. For using one or four integration points per element, the weights are given as seen in fig. 8.



Position	Weights
0.00000	2.00000

Position	Weights
±0.57735	1.00000

Figure 8: Weights for using one or four integration points for a 2D-rectangular element.

The coordinate transformation ${}^eT$ can be written in matrix form as

$$[x(\xi,\eta), y(\xi,\eta)] = [h_1(\xi,\eta),\ldots,h_p(\xi,\eta)] \begin{pmatrix} {}^e x_1 & {}^e y_1 \\ \vdots & \vdots \\ {}^e x_p & {}^e y_p \end{pmatrix} = {}^a\boldsymbol{H}(\xi,\eta)({}^e\boldsymbol{x}, {}^e\boldsymbol{y}), \tag{4.40}$$

where ${}^e\boldsymbol{x}$ and ${}^e\boldsymbol{y}$ are the coordinates of the p nodes of element e in the global coordinate system.

The Jacobian $\mathbf{J}_{{}^eT}(\xi,\eta)$ can then be calculated as

$$\begin{aligned}\mathbf{J}_{{}^eT}(\xi,\eta) &= \begin{pmatrix} \frac{\partial {}^e x(\xi,\eta)}{\partial \xi} & \frac{\partial {}^e y(\xi,\eta)}{\partial \xi} \\ \frac{\partial {}^e x(\xi,\eta)}{\partial \eta} & \frac{\partial {}^e y(\xi,\eta)}{\partial \eta} \end{pmatrix} = \begin{pmatrix} \partial_\xi \\ \partial_\eta \end{pmatrix} [x(\xi,\eta), y(\xi,\eta)] = \begin{pmatrix} \partial_\xi \\ \partial_\eta \end{pmatrix} {}^a\boldsymbol{H}(\xi,\eta)({}^e\boldsymbol{x}, {}^e\boldsymbol{y}) \\ &= \underbrace{\begin{pmatrix} \frac{\partial h_1(\xi,\eta)}{\partial \xi}, \dots, \frac{\partial h_p(\xi,\eta)}{\partial \xi} \\ \frac{\partial h_1(\xi,\eta)}{\partial \eta}, \dots, \frac{\partial h_p(\xi,\eta)}{\partial \eta} \end{pmatrix}}_{\doteq \mathbf{A}(\xi,\eta)} ({}^e\boldsymbol{x}, {}^e\boldsymbol{y}) = {}^a\mathbf{A}(\xi,\eta)({}^e\boldsymbol{x}, {}^e\boldsymbol{y}). \end{aligned} \tag{4.41}$$

The B-matrix ${}^a\mathbf{B}(\xi,\eta)$ can be calculated as

$$\begin{aligned}{}^a\mathbf{B}(\xi,\eta) &= \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix} {}^e\boldsymbol{H}(x,y) = \mathbf{J}_{{}^eT}^{-1} \begin{pmatrix} \partial_\xi \\ \partial_\eta \end{pmatrix} {}^a\boldsymbol{H}(\xi,\eta) = \mathbf{J}_{{}^eT}^{-1} \begin{pmatrix} \frac{\partial h_1(\xi,\eta)}{\partial \xi}, \dots, \frac{\partial h_p(\xi,\eta)}{\partial \xi} \\ \frac{\partial h_1(\xi,\eta)}{\partial \eta}, \dots, \frac{\partial h_p(\xi,\eta)}{\partial \eta} \end{pmatrix} \\ &= \mathbf{J}_{{}^eT}^{-1}{}^a\mathbf{A}(\xi,\eta). \end{aligned} \tag{4.42}$$

With these results and the weights from fig. 8, one can now calculate the finite element model using Gauss integration for $s = 1$ or $s = 4$ integration points per element. The results obtained by Gauss integration are visualized in fig. 9.

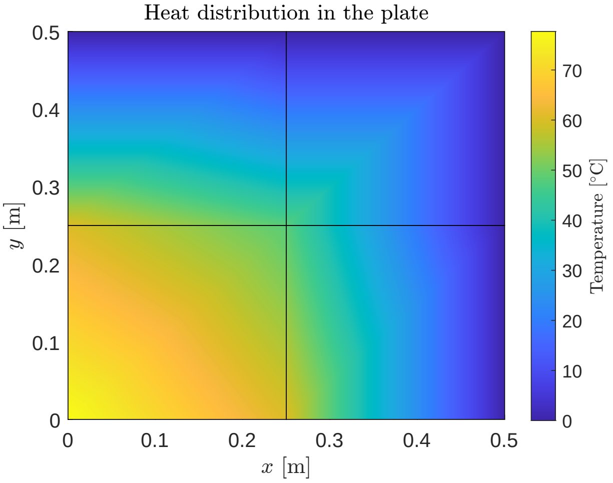


Figure 9: Solution obtained by Gauss integration for the heat conduction problem.

7. **Q** — What is the principle of Gauss integration (Gauss quadrature) used in finite element solvers?

A — The principle of Gauss integration - also known as Gauss quadrature - is based on approximating an integral using optimally chosen integration points and corresponding weights. In general, one would like to evaluate an integral of the form

$$I = \int_{-1}^{1} f(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} w_i f(x_i). \tag{4.43}$$

Now, consider that $f(x)$ may be approximated very well using a polynomial $p_{2n-1}(x)$ of degree $2n-1$, where $n \in \mathbb{N}$. This is possible for almost all smooth functions, given sufficiently high order n. Now, Gauss integration relies on intelligently and carefully choosing the integration points x_i and weights w_i and is thus yields exact results for polynomials up to degree $2n-1$ using n integration points x_i and correspondingly n weights w_i.

Consider now the space of polynomials with the basis $\{1, x, x^2, x^3, \dots\}$ and the inner product defined as

$$\langle p, q \rangle = \int_{-1}^{1} p(x)q(x)\,\mathrm{d}x. \tag{4.44}$$

The so-called Legendre polynomials $L_n(x)$ are obtained by the Gram-Schmidt orthogonalization procedure as a basis of the space of polynomials, hence the Legendre polynomials up to and including order n form a basis of space of polynomials up to n'th order and are thus equivalent to the basis $\{1, x, x^2, x^3, \dots\}$. But the Legendre polynomials have the property

$$\langle L_i, L_j \rangle = \int_{-1}^{1} L_i(x)L_j(x)\,\mathrm{d}x = 0, \qquad i \neq j \tag{4.45}$$

by the orthogonal property of Legendre polynomials. This also means, that the Legendre polynomial $L_n(x)$ is orthogonal to any polynomial $p_{n-1}(x)$ of order $n-1$, namely

$$\langle L_n, p_{n-1} \rangle = \int_{-1}^{1} L_n(x)p_{n-1}(x)\,\mathrm{d}x = 0. \tag{4.46}$$

At this point it is important to realize, that any polynomial $p_{2n-1}(x)$ of order $2n-1$ can be written in the form

$$p_{2n-1}(x) = q_{n-1}(x)L_n(x) + r_{n-1}(x), \tag{4.47}$$

where $q_{n-1}(x)$ and $r_{n-1}(x)$ are polynomials up to order $n-1$. This follows from polynomial division

$$\frac{p_{2n-1}(x)}{L_n(x)} = q_{n-1}(x) + \frac{r_{n-1}(x)}{L_n(x)}. \tag{4.48}$$

Calculating now the integral for $p_{2n-1}(x)$ yields

$$\int_{-1}^{1} p_{2n-1}(x)\,\mathrm{d}x = \int_{-1}^{1} q_{n-1}(x)L_n(x)\,\mathrm{d}x + \int_{-1}^{1} r_{n-1}(x)\,\mathrm{d}x = \int_{-1}^{1} r_{n-1}(x)\,\mathrm{d}x, \tag{4.49}$$

where the first term has vanished because of the orthogonality property of the Legendre polynomial $L_n(x)$ with any other polynomial of maximal degree $n-1$. Another important property of Legendre polynomials is, that the each Legendre polynomial $L_n(x)$ has exactly n roots and all of these roots lie in the interval $[-1, 1]$. Recalling the goal to find optimal weights w_i and evaluation points x_i to approximate the integral $I = \int_{-1}^{1} f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i)$. Now, choosing the roots $x_1, \dots, x_n$ of the Legendre polynomial $L_n(x)$ as evaluation points for the polynomial $p_{2n-1}(x)$, we have the situation

$$\sum_{i=1}^{n} w_i p_{2n-1}(x_i) = \sum_{i=1}^{n} w_i q_{n-1}(x_i)L_n(x_i) + \sum_{i=1}^{n} w_i r_{n-1}(x_i) = \sum_{i=1}^{n} w_i r_{n-1}(x_i), \tag{4.50}$$

because $L_n(x_i) = 0$ for all $i \in \{1, \dots, n\}$. Now, one can choose the weights w_i in such a way, that

$$\int_{-1}^{1} p_{2n-1}(x)\,\mathrm{d}x = \sum_{i=1}^{n} w_i p_{2n-1}(x_i) = \sum_{i=1} w_i r_{n-1}(x_i) = \int_{-1}^{1} r_{n-1}(x)\,\mathrm{d}x \tag{4.51}$$

holds exactly. This is done by solving the linear system of equations

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \dots & x_n^{n-1} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} \int_{-1}^{1} 1\,\mathrm{d}x \\ \int_{-1}^{1} x\,\mathrm{d}x \\ \int_{-1}^{1} x^2\,\mathrm{d}x \\ \vdots \\ \int_{-1}^{1} x^{n-1}\,\mathrm{d}x \end{pmatrix} \tag{4.52}$$

The n weights obtained in such a way ensure, that all powers of x up to order $n-1$ are integrated exactly over the interval $[-1,1]$ by the sum $\sum_{i=1}^{n} w_i x_i^k$ for $k \in \{0,1,\dots,n-1\}$. Recall, that the x_i are the roots of the n'th Legendre polynomial. That is to say, choosing the weights as described ensures

$$\int_{-1}^{1} x^k \,\mathrm{d}x = \sum_{i=1}^{n} w_i x_i^k, \qquad k \leq n-1. \tag{4.53}$$

Writhing the polynomial $r_{n-1}(x)$ as

$$r_{n-1}(x) = c_{n-1}x^{n-1} + c_{n-1}x^{n-2} + \cdots + c_1 x^1 + c_0 = \sum_{j=0}^{n-1} c_j x^j \tag{4.54}$$

therefore yields

$$\begin{aligned}\int_{-1}^{1} r_{n-1}(x)\,\mathrm{d}x &= \int_{-1}^{1} \sum_{j=1}^{n-1} c_j x^j \,\mathrm{d}x = \sum_{j=0}^{n-1} c_j \int_{-1}^{1} x^j \,\mathrm{d}x = \sum_{j=0}^{n-1} c_j \sum_{i=1}^{n} w_i x_i^j = \sum_{i=1}^{n} w_i \sum_{j=0}^{n-1} c_j x_i^j \\ &= \sum_{i=1}^{n} w_i r_{n-1}(x_i).\end{aligned} \tag{4.55}$$

Putting everything together we have

$$\int_{-1}^{1} p_{2n-1}(x)\,\mathrm{d}x = \sum_{i=1}^{n} w_i p_{2n-1}(x_i), \tag{4.56}$$

if the weights w_i are obtained by solving eq. (4.52) and if the x_i are the roots of the n'th Legendre polynomial $L_n(x)$.

Recall now that the initial goal was to calculate $I = \int_{-1}^{1} f(x)\,\mathrm{d}x$ by a weighted sum $\sum_{i=1}^{n} w_i f(x_i)$. Let for the following material w_i always be the weights obtained by solving eq. (4.52) and x_i the roots of the n'th degree Legendre polynomial $L_n(x)$. Now if $f(x)$ is identical to a polynomial of degree $2n-1$, we have $f(x) = p_{2n-1}(x)$ and thus by the above result

$$\int_{-1}^{1} f(x)\,\mathrm{d}x = \sum_{i=1}^{n} w_i f(x_i). \tag{4.57}$$

If $f(x)$ is not identical, but very similar to a polynomial of degree $2n-1$, we have $f(x) \approx p_{2n-1}(x)$ and hence the approximate result

$$\int_{-1}^{1} f(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} w_i f(x_i) \tag{4.58}$$

holds. Note, that the

As an example, we consider the procedure for $n=3$, allowing for exact integration of polynomials up to order $2n-1=5$ using Gauss integration. For this, we need the Legendre polynomial $L_3(x)$ and its roots given by

$$L_3(x) = \frac{1}{2}\left(5x^3 - 3x\right), \qquad x_1 = -\sqrt{3/5},\ x_2 = 0,\ x_3 = \sqrt{3/5}. \tag{4.59}$$

Now, eq. (4.52) yields the system of equations

$$\begin{pmatrix} 1 & 1 & 1 \\ -\sqrt{3/5} & 0 & \sqrt{3/5} \\ 3/5 & 0 & 3/5 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 2/3 \end{pmatrix} \tag{4.60}$$

to be solved. Solving this yields $w_1 = 0.55555556$, $w_2 = 0.88888889$, $w_3 = 0.55555556$. Given this scheme, one can integrate functions $f(x)$ that are well approximated by a fifth order polynomial extremely well; much better and faster than with the trapezoidal rule for example.

Two remarks remain to be made: First, if one has a general interval $[a, b]$ to integrate over, namely $\int_a^b f(x)\,\mathrm{d}x$, one can simply use change of variables formula

$$\int_a^b f(x)\,\mathrm{d}x = \int_{y(a)}^{y(b)} f(x(y))\frac{\mathrm{d}x(y)}{\mathrm{d}y}\,\mathrm{d}y. \tag{4.61}$$

In our case we use $y(x) = \frac{2x-(a+b)}{b-a}$ leading to

$$\int_a^b f(x)\,\mathrm{d}x = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}y + \frac{a+b}{2}\right)\mathrm{d}y. \tag{4.62}$$

As a second remark it shall be noted, that the Legendre polynomials can be calculated recursively as

$$L_0(x) = 1, \quad L_1(x) = x, \quad L_{n+1}(x) = \frac{(2n+1)L_n(x) - nL_{n-1}(x)}{n+1} \tag{4.63}$$

and the weights used for Gauss integration can be found directly via the formula

$$w_i = \frac{2}{(1-x_i^2)\,[L_n'(x_i)]^2}, \tag{4.64}$$

where x_i is the i'th root of $L_n(x)$ and $L_n'(x)$ is the derivative of $L_n(x)$ with respect to x.

5 3D continuous system

8. **Q** — How does an example application of the static 3D continuous FEM method work?

A — The finite element method for static problems in 3D works exactly analogous to the 2D method, with the difference that the finite elements are now in 3D rather than in 3D. This makes a slight reshaping of the H-vector containing the element shape functions necessary. Furthermore, the transformation T from global coordinates $\boldsymbol{x} = (x, y, z)^\top$ to element coordinates $\boldsymbol{\xi} = (\xi, \eta, \zeta)^\top$ is given The transformation from element coordinates $\boldsymbol{\xi}$ to global coordinates by

$${}^eT : \boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi}) = \sum_{i=1}^{{}^ep} {}^eh_i(\boldsymbol{\xi})\,{}^e\boldsymbol{x}_i = {}^e\mathbf{H}(\boldsymbol{\xi})\,{}^e\boldsymbol{x}, \tag{5.1}$$

where ${}^e\boldsymbol{x}$ is defined as

$${}^e\boldsymbol{x} = ({}^ex_1, {}^ey_1, {}^ez_1, \ldots, {}^ex_{{}^ep}, {}^ey_{{}^ep}, {}^ez_{{}^ep}) \tag{5.2}$$

and ${}^e\mathbf{H}(\boldsymbol{x})$ is given by

$$\begin{aligned} {}^e\mathbf{H}(\boldsymbol{x}) &= ({}^e\mathbf{H}_1(\boldsymbol{x}), \ldots, {}^e\mathbf{H}_{{}^ep}(\boldsymbol{x})) \\ &= \begin{pmatrix} {}^eh_1(\boldsymbol{x}) & 0 & 0 & {}^eh_2(\boldsymbol{x}) & 0 & 0 & \ldots & {}^eh_{{}^ep}(\boldsymbol{x}) & 0 & 0 \\ 0 & {}^eh_1(\boldsymbol{x}) & 0 & 0 & {}^eh_2(\boldsymbol{x}) & 0 & \ldots & 0 & {}^eh_{{}^ep}(\boldsymbol{x}) & 0 \\ 0 & 0 & {}^eh_1(\boldsymbol{x}) & 0 & 0 & {}^eh_2(\boldsymbol{x}) & \ldots & 0 & 0 & {}^eh_{{}^ep}(\boldsymbol{x}) \end{pmatrix}. \end{aligned} \tag{5.3}$$

The function $\boldsymbol{u}(\boldsymbol{x})$ to find by the finite element method is given by the approximation $\boldsymbol{u}(\boldsymbol{x}) \approx \boldsymbol{u}^h(\boldsymbol{x})$. The approximated solution $\boldsymbol{u}^h(\boldsymbol{x})$ can be further decomposed into element solutions ${}^e\boldsymbol{u}^h(\boldsymbol{x})$ by means of

$$\boldsymbol{u}^h(\boldsymbol{x}) = \sum_{e=1}^{m} {}^e\boldsymbol{u}^h(\boldsymbol{x}), \tag{5.4}$$

where $m \in \mathbb{N}$ is the number of elements in the FEM system. The element solutions ${}^e\boldsymbol{u}^h(\boldsymbol{x})$ are approximated by the sum of element shape functions ${}^eh_i(\boldsymbol{x})$ multiplied by the nodal unknowns ${}^e\boldsymbol{q}_i$ for all nodes $i \in \{1, \ldots, {}^ep\}$ with ${}^ep \in \mathbb{N}$ the number of nodes for element e. That is, we have

$$ {}^e\boldsymbol{u}^h(\boldsymbol{x}) = \sum_{i=1}^{{}^ep} {}^eh_i(\boldsymbol{x}) {}^e\boldsymbol{q}_i = \sum_{i=1}^{{}^ep} {}^e\mathbf{H}_i(\boldsymbol{x}) {}^e\boldsymbol{q}_i = {}^e\mathbf{H}(\boldsymbol{x}) {}^e\boldsymbol{q}, \tag{5.5} $$

where ${}^e\boldsymbol{q} = ({}^e\boldsymbol{q}_1, \ldots, {}^e\boldsymbol{q}_{{}^ep})^\top$.

6 General continuous and time-dependent systems

9. **Q** — How can an example of a general case of a time-dependent partial differential equation (PDE) be solved using the finite element method?

A — Let us consider a one-dimensional, time- and space-dependent boundary value problem governed by the partial differential equation (PDE)

$$ \frac{\partial u(x,t)}{\partial t} - \frac{\partial}{\partial x}\left(k(x) \frac{\partial u(x,t)}{\partial x} \right) = f(x,t), \quad x \in [0, L] = \Omega, \quad t > 0 \tag{6.1} $$

subject to Dirichlet boundary conditions

$$ u(0,t) = u_{x0}, \quad u(L,t) = u_{xL}, \quad t > 0, \quad u(x,0) = u_{t0}(x), \tag{6.2} $$

where the last equation is an initial condition rather than a boundary condition. This is the strong form of the problem. In order to obtain a numerical solution to the PDE, we must convert it to the weak form. This is done by multiplying both sides of the PDE with weighting function $\delta u(x,t)$ and integrating over the domain Ω as

$$ \int_\Omega \frac{\partial u(x,t)}{\partial t} \delta u(x,t) \,\mathrm{d}x - \int_\Omega \frac{\partial}{\partial x} \left(k(x) \frac{\partial u(x,t)}{\partial x} \right) \delta u(x,t) \,\mathrm{d}x = \int_\Omega f(x,t) \delta u(x,t) \,\mathrm{d}x. \tag{6.3} $$

where we require the weighting function (variation) $\delta u(x,t)$ to be zero at the boundary of the integration domain Ω. In order for the above weak form to be equivalent to the strong form, the formulation must hold for any weighting function $\delta u(x,t)$. Using integration by parts of the second term in the above weak form formulation and exploiting that the weighting function $\delta u(x,t)$ goes to zero at integration boundaries, we obtain

$$ \int_\Omega \frac{\partial u(x,t)}{\partial t} \delta u(x,t) \,\mathrm{d}x + \int_\Omega k(x) \frac{\partial u(x,t)}{\partial x} \frac{\partial \delta u(x,t)}{\partial x} \,\mathrm{d}x = \int_\Omega f(x,t) \delta u(x,t) \,\mathrm{d}x. \tag{6.4} $$

Now, let the function $u(x,t)$ be approximated by a hypothesis function $u^h(x,t)$, such that $u(x,t) \approx u^h(x,t)$. The hypothesis function shall be defined element-wise as

$$ u^h(x,t) = \sum_{e=1}^{m} {}^eu^h(x,t), \tag{6.5} $$

where the element-wise function ${}^eu^h(x,t)$ shall be given as the product of the element shape functions ${}^eh_i(x)$ and the nodal unknowns ${}^eq_i(t)$ as

$$ {}^eu^h(x,t) = \sum_{i=1}^{{}^ep} {}^eh_i(x) {}^eq_i(t) = {}^e\boldsymbol{H}(x) {}^e\boldsymbol{q}(t), \quad {}^e\delta u^h(x,t) = \sum_{i=1}^{{}^ep} {}^eh_i(x) {}^e\delta q_i(t) = {}^e\boldsymbol{H}(x) {}^e\delta\boldsymbol{q}(t) \tag{6.6} $$

where ${}^e\boldsymbol{H}(x) = ({}^e h_1(x), \dots, {}^e h_{{}^e p}(x))^\top$ with ${}^e p$ being the number of nodes per element e and ${}^e\boldsymbol{q}(t) = ({}^e q_1(t), \dots, {}^e q_{{}^e p}(t))^\top$. Also the weighting function $\delta u(x,t)$ is approximated in the same way using the element shape functions and nodal unknowns, as shown above. The connection of the element system to the global system is established via the localization matrix ${}^e\mathbf{L}$ given by a connectivity table such that

$$ {}^e\boldsymbol{q} = {}^e\mathbf{L}\boldsymbol{q}, \tag{6.7} $$

which means that we have

$$ \begin{aligned} u^h(x,t) &= \sum_{e=1}^{m} {}^e u^h(x,t) = \sum_{e=1}^{m} {}^e\boldsymbol{H}(x){}^e\boldsymbol{q} = \sum_{e=1}^{m} {}^e\boldsymbol{H}(x){}^e\mathbf{L}\boldsymbol{q}, \\ \delta u^h(x,t) &= \sum_{e=1}^{m} {}^e u^h(x,t) = \sum_{e=1}^{m} {}^e\boldsymbol{H}(x){}^e\delta\boldsymbol{q} = \sum_{e=1}^{m} {}^e\boldsymbol{H}(x){}^e\mathbf{L}\delta\boldsymbol{q}. \end{aligned} \tag{6.8} $$

One can also define the system on global scale, not with element-wise defined shape functions. In this case, we have just

$$ u^h(x,t) = \boldsymbol{H}(x)\boldsymbol{q}(t), \quad \delta u^h(x,t) = \boldsymbol{H}(x)\delta\boldsymbol{q}(t), \tag{6.9} $$

where $\boldsymbol{H}(x) = (h_1(x), \dots, h_n(x))$ now contains all shape functions for the whole system, also defined on the whole domain Ω; and $\boldsymbol{q}(t) = (q_1(t), \dots, q_n(t))^\top$ contains all n nodal unknowns of the system. Inserting eq. (6.9) into eq. (6.4) leads after some simplifications to

$$ \int_\Omega \boldsymbol{H}(x)^\top \boldsymbol{H}(x)\,\mathrm{d}x \cdot \frac{\mathrm{d}\boldsymbol{q}(t)}{\mathrm{d}t} + \int_\Omega k(x)\frac{\mathrm{d}\boldsymbol{H}(x)^\top}{\mathrm{d}x}\frac{\mathrm{d}\boldsymbol{H}(x)}{\mathrm{d}x}\,\mathrm{d}x \cdot \boldsymbol{q}(t) = \int_\Omega f(x,t)\boldsymbol{H}(x)^\top\,\mathrm{d}x. \tag{6.10} $$

Defining the mass matrix $\mathbf{M}$, the stiffness matrix $\mathbf{K}$ and the load vector $\boldsymbol{r}(t)$ as

$$ \mathbf{M} \doteq \int_\Omega \boldsymbol{H}(x)^\top \boldsymbol{H}(x)\,\mathrm{d}x, \quad \mathbf{K} \doteq \int_\Omega k(x)\frac{\mathrm{d}\boldsymbol{H}(x)^\top}{\mathrm{d}x}\frac{\mathrm{d}\boldsymbol{H}(x)}{\mathrm{d}x}\,\mathrm{d}x, \quad \boldsymbol{r}(t) \doteq \int_\Omega f(x,t)\boldsymbol{H}(x)^\top\,\mathrm{d}x, \tag{6.11} $$

we can write the system to be solved as

$$ \mathbf{M}\frac{\mathrm{d}\boldsymbol{q}(t)}{\mathrm{d}t} + \mathbf{K}\boldsymbol{q}(t) = \boldsymbol{r}(t). \tag{6.12} $$

One can also express everything element-wise as

$$ \mathbf{M} = \sum_{e=1}^{m} {}^e\mathbf{L}^\top {}^e\mathbf{M} {}^e\mathbf{L}, \quad \mathbf{K} = \sum_{e=1}^{m} {}^e\mathbf{L}^\top {}^e\mathbf{K} {}^e\mathbf{L}, \quad \boldsymbol{r}(t) = \sum_{e=1}^{m} {}^e\mathbf{L}^\top {}^e\boldsymbol{r}(t). \tag{6.13} $$

and

$$ {}^e\mathbf{M} \doteq \int_{{}^e\Omega} {}^e\boldsymbol{H}(x)^\top {}^e\boldsymbol{H}(x)\,\mathrm{d}x, \quad {}^e\mathbf{K} \doteq \int_{{}^e\Omega} k(x)\frac{\mathrm{d}^e\boldsymbol{H}(x)^\top}{\mathrm{d}x}\frac{\mathrm{d}^e\boldsymbol{H}(x)}{\mathrm{d}x}\,\mathrm{d}x, \quad {}^e\boldsymbol{r}(t) \doteq \int_{{}^e\Omega} f(x,t){}^e\boldsymbol{H}(x)^\top\,\mathrm{d}x. \tag{6.14} $$

Now, once one has assembled the global system eq. (6.12), it needs to be solved by application of the boundary and initial conditions. Now, the initial condition $\boldsymbol{q}(t=0)$ is given by $u(x,t=0) = u_{t0}(x)$, which is specified. Hence, time integration of the system eq. (6.12) is necessary. There are several ways to do this, where the most straightforward methods are the explicit and implicit Euler methods.

Both methods start with approximating the derivative $\frac{\mathrm{d}\boldsymbol{q}(t)}{\mathrm{d}t}$ as

$$ \left.\frac{\mathrm{d}\boldsymbol{q}(t)}{\mathrm{d}t}\right|_{t=j\Delta t} \approx \frac{\boldsymbol{q}^{j+1} - \boldsymbol{q}^j}{\Delta t} \tag{6.15} $$

by discretizing $\boldsymbol{q}(t)$ into discrete timesteps Δt, where $\boldsymbol{q}^j = \boldsymbol{q}(t^j) = \boldsymbol{q}(j\Delta t)$ and $\boldsymbol{r}^j = \boldsymbol{r}(t^j) = \boldsymbol{r}(j\Delta t)$. Now, the explicit Euler method approximates the differential equation eq. (6.12) by

$$\mathbf{M}\frac{\boldsymbol{q}^{j+1} - \boldsymbol{q}^j}{\Delta t} + \mathbf{K}\boldsymbol{q}^j = \boldsymbol{r}^j, \tag{6.16}$$

whereas the implicit method approximates it as

$$\mathbf{M}\frac{\boldsymbol{q}^{j+1} - \boldsymbol{q}^j}{\Delta t} + \mathbf{K}\boldsymbol{q}^{j+1} = \boldsymbol{r}^{j+1}. \tag{6.17}$$

The implicit formulation ensures that stiffness and mass effects are treated at the future time t^{j+1}, leading to better stability as compared to the explicit method. However, the implicit method requires solving a linear system of equations at each timestep, whereas the explicit method provides a simple equation to compute at each timestep. That is, the explicit method eq. (6.16) is rearranged to

$$\boldsymbol{q}^{j+1} = \boldsymbol{q}^j + \Delta t\mathbf{M}^{-1}\left(\boldsymbol{r}^j - \mathbf{K}\boldsymbol{q}^j\right), \tag{6.18}$$

whereas the implicit method eq. (6.17) amounts to

$$\left(\mathbf{M} + \Delta t\mathbf{K}\right)\boldsymbol{q}^{j+1} = \mathbf{M}\boldsymbol{q}^j + \Delta t\boldsymbol{r}^{j+1}. \tag{6.19}$$

10. **Q** — How can as simple FEM example with time integration be calculated?

A — Consider the 1D heat equation, which is of the form

$$\frac{\partial T(x,t)}{\partial t} - \frac{\partial}{\partial x}\left(k(x)\frac{\partial T(x,t)}{\partial x}\right) = f(x,t). \tag{6.20}$$

We assume $f(x,t) = 0$ and use the boundary and initial conditions

$$T(x=0,t) = 0\,\mathrm{K}, \quad T(x=L,t) = 0\,\mathrm{K}, \quad T(x,t=0) = T_0\sin(\pi x) \tag{6.21}$$

with $T_0 = 1\,\mathrm{K}$ and $L = 1\,\mathrm{m}$. The thermal conductivity is furthermore assumed to be constant $k(x) = k = 1\,\mathrm{Wm^{-1}K^{-1}}$. We set the number of nodes $n = 7$ and hence the number of elements to $m = 6$ and use linear elements with two nodes and the shape functions

$${}^e h_1(\xi) = \frac{1}{2}(1-\xi), \qquad {}^e h_2(\xi) = \frac{1}{2}(1+\xi). \tag{6.22}$$

Following the treatment outlined above, with ${}^e\boldsymbol{H}(x) = ({}^e h_1(x), {}^e h_2(x))$ the element mass matrix ${}^e\mathbf{M}$ is given by

$${}^e\mathbf{M} = \int_{{}^e\Omega} {}^e\boldsymbol{H}(x)^\top {}^e\boldsymbol{H}(x)\,\mathrm{d}x = \int_{-1}^{1} {}^e\boldsymbol{H}(x(\xi))^\top {}^e\boldsymbol{H}(x(\xi))\left|\frac{\mathrm{d}x(\xi)}{\mathrm{d}\xi}\right|\mathrm{d}\xi, \tag{6.23}$$

the element stiffenss matrix ${}^e\mathbf{K}$ by

$${}^e\mathbf{K} = \int_{{}^e\Omega} k\frac{\mathrm{d}{}^e\boldsymbol{H}(x)^\top}{\mathrm{d}x}\frac{\mathrm{d}{}^e\boldsymbol{H}(x)}{\mathrm{d}x}\,\mathrm{d}x = \int_{-1}^{1} k\frac{\mathrm{d}{}^e\boldsymbol{H}(x(\xi))^\top}{\mathrm{d}\xi}\frac{\mathrm{d}{}^e\boldsymbol{H}(x(\xi))}{\mathrm{d}\xi}\left(\frac{\mathrm{d}\xi(x)}{\mathrm{d}x}\right)^2\left|\frac{\mathrm{d}x(\xi)}{\mathrm{d}\xi}\right|\mathrm{d}\xi \tag{6.24}$$

and the element load vector ${}^e\boldsymbol{r}(t)$ as

$${}^e\boldsymbol{r}(t) = \int_{{}^e\Omega} f(x,t){}^e\boldsymbol{H}(x)\top\,\mathrm{d}x = \int_{-1}^{1} 0\cdot{}^e\boldsymbol{H}(x(\xi))\left|\frac{\mathrm{d}x(\xi)}{\mathrm{d}\xi}\right|\mathrm{d}\xi. \tag{6.25}$$

These quantities are calculated to be

$$ {}^{e}\mathbf{M} = \frac{L}{32}\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \qquad {}^{e}\mathbf{K} = \frac{6k}{L}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \qquad {}^{e}\boldsymbol{r}(t) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{6.26} $$

The assembled quantities are furthermore given by

$$ \mathbf{M} = \frac{L}{32}\begin{pmatrix} 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{pmatrix}, \quad \mathbf{K} = \frac{6k}{L}\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}, \quad \mathbf{r}(t) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{6.27} $$

Now, the nodal unknowns $\boldsymbol{q}(t) = (q_1(t), \dots, q_7(t))^\top$ are calculated using the explicit Euler method. The initial condition is given by sampling from $T(x, t = 0) = T_0 \sin(\pi x)$ at the locations $x_1, \dots, x_7$ of the seven nodes. This leads to the nodal unknowns vector $\boldsymbol{q}^0$ at time $t = 0$.

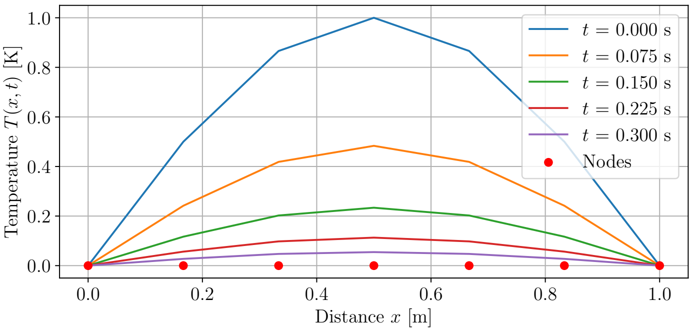


Figure 10: Result for the presented example.

Using the explicit Euler method, each time iteration step is now given by

$$ \boldsymbol{q}^{j+1} = \boldsymbol{q}^{j} + \Delta t \mathbf{M}^{-1}\left(\boldsymbol{r}^{j} - \mathbf{K}\boldsymbol{q}^{j}\right), \tag{6.28} $$

where Δt is the timestep. An example python implementation with the mentioned parameters leads to the result as shown in fig. 10. The code to produce the presented example is found below.

```python
import numpy as np
import matplotlib.pyplot as plt

# --- Parameters ---
L = 1.0           # Length of the domain (in meters)
n_elements = 6    # Number of finite elements
T = 0.3           # Final time (in seconds)
dt = 0.001        # Time step (in seconds)
k = 1.0           # Thermal conductivity (W/m K)
T_0 = 1.0         # Initial temperature (K)

# Derived quantities
n_nodes = n_elements + 1  # Number of nodes (one more than elements)
dx = L / n_elements      # Element size (length of each finite element)
n_steps = int(T / dt)    # Number of time steps based on final time and time step size

# --- Shape function derivatives for linear elements ---
# For a 2-node element on [0, dx], the derivatives of the shape functions are constant
dN_dx = np.array([[-1/dx, 1/dx]])

# --- Element Matrices (constant for uniform mesh) ---
# Element stiffness matrix (Ke) for linear elements
Ke = k * (dN_dx.T @ dN_dx) * dx  # Element stiffness matrix (2x2)

# Element mass matrix (Me) using analytical integration for linear elements
Me = (dx / 6) * np.array([[2, 1],  # Element mass matrix (2x2)
                          [1, 2]])

# --- Assembly of Global Matrices ---
# Initialize global stiffness and mass matrices with zeros
K = np.zeros((n_nodes, n_nodes))  # Global stiffness matrix
M = np.zeros((n_nodes, n_nodes))  # Global mass matrix
```

```python

# Assemble the global stiffness and mass matrices
for e in range(n_elements):
    # Local-to-global mapping for each element
    idx = [e, e + 1]

    # Assemble K and M by adding the contributions from each element
    for i in range(2):
        for j in range(2):
            K[idx[i], idx[j]] += Ke[i, j]  # Add element stiffness to global K
            M[idx[i], idx[j]] += Me[i, j]  # Add element mass to global M

# --- Initial condition: u(x, 0) = sin(pi*x) ---
# Initialize the node positions (x) and the initial temperature distribution
x = np.linspace(0, L, n_nodes)  # Position of each node
q = T_0 * np.sin(np.pi * x)           # Initial temperature profile

# Apply Dirichlet boundary conditions: u=0 at both ends (x=0 and x=L)
q[0] = 0
q[-1] = 0

# --- Time integration using Explicit Euler ---
q_all = [q.copy()]  # List to store the solution at all time steps

# Precompute M inverse (or lumped inverse for efficiency/stability)
# Lumped mass matrix: diagonal with row-sums of M
M_lumped_inv = 1 / M.sum(axis=1)

# Time-stepping loop (Explicit Euler method)
for step in range(n_steps):
    # Compute the right-hand side for the time update: r^j = -K @ q^j
    rhs = -K @ q

    # Explicit Euler update: q^{j+1} = q^j + dt * M^{-1} * (-K q^j)
    q_new = q + dt * (M_lumped_inv * rhs)

    # Enforce boundary conditions (u=0 at both ends)
    q_new[0] = 0
    q_new[-1] = 0

    # Update the solution and store it
    q = q_new
    q_all.append(q.copy())  # Append the solution at the current time step

# --- Plotting ---
from matplotlib import cm

q_all = np.array(q_all)  # Convert the list of solutions into a numpy array

# Sample 5 time indices evenly spaced through the time steps
time_indices = np.linspace(0, n_steps, 5, dtype=int)  # Choose 5 time indices
time_values = np.linspace(0, T, n_steps + 1)[time_indices]  # Corresponding time values

# Plot the temperature profile at each selected time step
plt.figure(figsize=(8, 4))  # Set the figure size

for idx, t_idx in enumerate(time_indices):
    plt.plot(x, q_all[t_idx], label=f"$t$ = {time_values[idx]:.3f} s")  # Plot each time step

# Label the axes
plt.xlabel("Distance $x$ [m]")
plt.ylabel("Temperature $T(x,t)$ [K]")

# Highlight the nodes with red markers
plt.scatter(x, np.zeros_like(x), color='red', zorder=5, label='Nodes')

# Add legend and grid
plt.legend()
plt.grid(True)

# Adjust layout and display the plot
plt.tight_layout()
plt.savefig('resultexampletimeint.pdf')
plt.show()
```

7 FEM software

11. **Q** — What is a usual workflow with FEM software such as Abaqus or Comsol?

A — A typical workflow with FEM software such as Abaqus or COMSOL begins with the pre-processing stage. Here, the geometry of the model is defined, either through CAD tools or by importing from external files. Material properties such as Young's modulus E, Poisson's ratio ν, and density are specified. The geometry is then discretized into finite elements by meshing, and boundary conditions, including constraints, loads, and initial conditions, are applied.

In the solution stage, the software assembles the global system of equations, typically of the form

$$\mathbf{Ku} = \mathbf{f} \tag{7.1}$$

where $\mathbf{K}$ is the global stiffness matrix, $\mathbf{u}$ is the displacement vector, and $\mathbf{f}$ is the external force vector. The solver computes the displacements, from which stresses and strains are derived.

The final stage is post-processing. Results such as displacements, stress fields, strain distributions, and other derived quantities are visualized and interpreted. Verification and validation steps follow, including mesh refinement, convergence studies, and comparison with experimental or analytical results.

12. **Q** — What can be problems with linear shape functions in quadrilateral elements?

A — Linear shape functions in quadrilateral elements can suffer from a numerical problem known as shear locking. This issue becomes significant in bending-dominated problems when using linear shape functions. As a result of using linear shape functions, the deformation is artificially restricted and the numerical solution underestimates the displacements. The shear locking effect can be reduced or even eliminated by using higher-order shape functions or a finer mesh.

13. **Q** — What is hourglassing?

A — Hourglassing is a numerical artifact that occurs in under-integrated elements, particularly those using reduced integration. These elements may exhibit zero-energy deformation modes, called hourglass modes, in which the shape of the element distorts in a pattern that does not contribute to the internal strain energy. Hourglassing can be reduced or even eliminated by using full integration instead of reduced integration, or by using higher-order shape functions for the finite elements.

14. **Q** — What are the main requirements for a model to be simplified in terms of the geometry?

A — In terms of geometry, the main requirement for a model to be simplified is that the region of interest for the finite element simulation is far from the simplified part of the geometry. Simplifying the geometry is important especially in larger models to save computation time. However, it is important that object and geometries of interest are drawn in the necessary complexity and also meshed properly.

15. **Q** — What is submodeling and what are challenges with it?

A — Submodeling is when has a global coarse model and a finer local model within or on top of it. The coarse model determines the boundary conditions for the finer model on top of it. This approach is also heavily used for larger models, where one can save a lot of computational resources doing this.

16. **Q** — What is full integration and reduced integration with respect to Gauss integration used in finite element solvers?

A — Consider a polynomial $p_{2n-1}(x)$ of order $2n - 1$. Gauss integration allows to integrate this polynomial exactly using at least n integration points. That is, in order to integrate a polynomial function $p_k(x)$ exactly and thus have a full integration, one needs to have at least $(k + 1)/2$ integration points. As an example, full integration of a polynomial of order 7 requires at least $(7 + 1)/2 = 4$ integration points. If one uses less than 4 integration points, one has a so-called reduced integration, which may still be quite good, but not exact.

Let us provide an example with respect to the finite element method. Consider a 2D quadrilateral element as visualized in fig. 11. First, lets consider linear shape functions ${}^{e}h_i \sim f(1, \xi, \eta, \xi\eta)$.

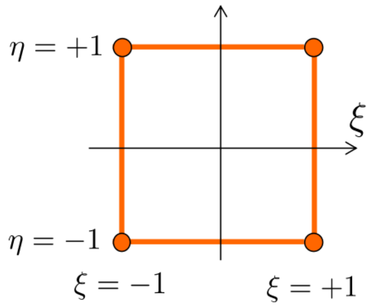


Figure 11: 2D quadrilateral element.

A derivative of the shape function with respect to local coordinates is of order $\partial_{\xi,\eta}{}^{e}h_i \sim f(1, \xi, \eta)$. The computation of the element stiffness matrix ${}^{e}\mathbf{K}$ involves products of the derivatives of the shape functions, hence one has a function $\partial_{\xi,\eta}{}^{e}h_i \partial_{\xi,\eta}{}^{e}h_j \sim f(1, \xi, \eta, \xi^2, \eta^2, \xi\eta)$. The highest order terms to integrate via Gauss quadrature are hence quadratic, that is $k = 2$. Therefore we need at least $(k + 1)/2 = (2 + 1)/2 = 3/2$ and hence 2 integration points per axis for full integration. That is, in the case of a quadrilateral 2D element with linear shape functions, one has to have $2 \times 2 = 2$ integration points per element to have full (exact) integration using Gauss quadrature. In case of quadratic shape functions for a 2D quadrilateral element, one will find that the highest order terms to integrate are of oder $k = 4$. Hence one needs at least $(k+1)/2 = (4+1)/2 = 5/2$ and hence 3 integration point per axis and therefore $3 \times 3 = 9$ integration points per element to have exact integration.

8 Various questions

17. **Q** — What are contact problems?

A — A contact problem is a situation in which there are two bodies with different properties interacting with each other. At the contact surface, the material properties are discontinuous. The challenge in contact problems is that contact may only occur in certain special circumstances; but if contact is established, many properties of the modelled system can change. For example, the conductance properties suddenly change when contact between surfaces is established.

Contact between two bodies typically feature

- contact pressure resisting penetration,
- frictional stress resisting sliding aswell as
- electrical and thermal interactions.

Recall that stress is defined as internal force per area and strain is defined as deformation in the loaded condition relative to the unloaded situation.

18. **Q** — How does the FEM formulation to solve contact problems look like?.

19. **Q** — Consider the usual formulation of a finite element problem:

$$\mathbf{K}\boldsymbol{u} = \boldsymbol{F}, \tag{8.1}$$

where $\boldsymbol{F}$ is the external force vector, $\boldsymbol{u}$ is the vector of nodal displacements and $\mathbf{K}$ is the stiffness matrix of the problem. The total potential energy for such a system involving contact surfaces is given by

$$W(\boldsymbol{\lambda}, \boldsymbol{u}) = \frac{1}{2}\boldsymbol{u}^\top \mathbf{K}\boldsymbol{u} - \boldsymbol{u}^\top \boldsymbol{F} + \boldsymbol{\lambda}^\top \left(\boldsymbol{u}\mathbf{G}^\top - \boldsymbol{g}_0\right), \tag{8.2}$$

where $\mathbf{G}\boldsymbol{\lambda}$ are additional internal forces due to contact constraints, $\boldsymbol{\lambda}$ are Lagrange multipliers interpreted as contact forces and $\boldsymbol{g}_0$ is the desired gap between contact surfaces (usually 0 for perfect contact). Equilibrium conditions now require the gradients of the energy functional $W(\boldsymbol{\lambda}, \boldsymbol{u})$ to be zero, hence

$$\boldsymbol{\nabla}_{\boldsymbol{u}} W(\boldsymbol{\lambda}, \boldsymbol{u}) = \mathbf{K}\boldsymbol{u} - \boldsymbol{F} + \mathbf{G}\boldsymbol{\lambda} = 0, \quad \boldsymbol{\nabla}_{\boldsymbol{\lambda}} W(\boldsymbol{\lambda}, \boldsymbol{u}) = \mathbf{G}^\top \boldsymbol{u} - \boldsymbol{g}_0 = 0. \tag{8.3}$$

Combining both equations one gets

$$\mathbf{K}\boldsymbol{u} + \mathbf{G}\boldsymbol{\lambda} = \boldsymbol{F}, \quad \mathbf{G}^\top \boldsymbol{u} = \boldsymbol{g}_0 \tag{8.4}$$

and thus in matrix form

$$\begin{pmatrix} \mathbf{K} & \mathbf{G} \\ \mathbf{G}^\top & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F} \\ \boldsymbol{g}_0 \end{pmatrix}. \tag{8.5}$$

One can see that it yields a new matrix equation of the same form as before: Stiffness matrix times displacement vector equals force vector.

20. **Q** — What are possible problems associated with solving contact problems?

A — Solving contact problems with the finite element method is challenging for a number of reasons. First of all, contact between bodies of different material composition introduces geometrical aswell as material nonlinearities. Second, solutions tend to become unstable and iterative solvers may not converge easily anymore, because of non-smooth constraints. At a contact surface, the material properties for example can make jumps and thus the constraints are non-smooth. Lastly, the accuracy and robustness of the contact formulation can be highly mesh-dependent.

21. **Q** — What is the Lagrangian formulation of a finite element methods problem?

A — The Lagrangian formulation is such that the nodes are fixed within the material and thus follow material deformation. The elements therefore deform as the material deforms.

Let $\boldsymbol{X}$ be the initial position of particles in a material and let t be the time. Let furthermore $\boldsymbol{x}$ be the position of a particle initially at $\boldsymbol{X}$ at time $t = 0$. In the Lagrangian formulation, the position of a particle at a later time t is given by

$$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{X}, t) \tag{8.6}$$

and thus the displacement is

$$\boldsymbol{u}(\boldsymbol{X}, t) = \boldsymbol{x}(\boldsymbol{X}, t) - \boldsymbol{X}. \tag{8.7}$$

This corresponds to the coordinates (nodes) following material deformation.

22. **Q** — What is the Eulerian formulation of a finite element methods problem?

A — The Eulerian formulation is such that the nodes are fixed in space and thus do not follow material deformation. The elements thus do not deform as the material deforms, rather the material flows through the elements.

Let $\boldsymbol{X}$ be the initial position of particles in a material and let t be the time. Let furthermore $\boldsymbol{x}$ be the position of a particle initially at $\boldsymbol{X}$ at time $t = 0$. In the Eulerian formulation, the position of a particle at a later time t is given by

$$\boldsymbol{X} = \boldsymbol{X}(\boldsymbol{x}, t) \tag{8.8}$$

and thus the displacement is

$$\boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{x} - \boldsymbol{X}(\boldsymbol{x}, t). \tag{8.9}$$

This corresponds to the coordinates (nodes) being fixed in space and the material moving through the elements.

23. **Q** — What are the advantages and disadvantages of the Eulerian vs. the Lagrangian finite element method formulations?

A — In the Lagrangian formulation, the independent variables are the fixed positions $\boldsymbol{X}$ and the time t. The advantages of the Lagrangian formulation hence lie in accurate tracking of material interfaces, which is good for solid mechanics. When there are large deformations however, the mesh can distort tragically, because the coordinate system comoves with material deformation.

In the Eulerian formulation, the independent variables are the material $\boldsymbol{x}$ positions and the time t. The advantage of the Eulerian formulation hence is that there is no mesh distortion, since elements stay fixed. The Eulerian formulation is therefore suitable for fluid flow and very large deformations. However, it is difficult to track material boundaries with this setup.

24. **Q** — What is hyperelasticity? What is the difference to normal elasticity?

A — E normal elastic material is characterized by a linear relationship between stress σ and strain ε. Recall that stress is defined as internal force per area and strain is defined as deformation in the loaded condition relative to the unloaded situation. So in a normal elastic situation, one usually has a linear relationship $\sigma = E\varepsilon$, where E is often Young's modulus, which is defined as the ratio of axial stress to axial strain. This situation is known as elastic condition and it is sufficient, if one only has relatively small deformations.

Hyperelasticity however occurs, when one has large elastic deformations such as in rubber or biological tissues. Energy is stored in the material, when it is being deformed. Hyperelasticity no longer features a linear relationship between stress and strain, which is the main difference between elasticity and hyperelasticity.

25. **Q** — What is parameter identification and how does a typical workflow using FEM software to do this look like?

A — Parameter identification refers to a technique involving finite element software to determine the material properties (parameters) of a material. Consider some material and a model that is parametrized with material parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$. In order to determine the material parameters $\boldsymbol{\theta}$, one has to design an experiment, which can also be modelled in finite element software.

Let now u be the observed quantity in the experiment. In the finite element model, the observed quantity is dependent upon the material parameters $\boldsymbol{\theta}$, therefore we write $u^{fem}(\boldsymbol{\theta})$. Furthermore, $u_i^{fem}(\boldsymbol{\theta})$ shall denote the value of the quantity $u(\boldsymbol{\theta})$ at the i'th measurement point out of N in total. The measured experimental data we denote as u_i^{exp}.

Now, one defines initial parameters $\boldsymbol{\theta}_0$ and calculates the outcome of the experiment in a region $[\boldsymbol{\theta}_0 - \delta\boldsymbol{\theta}, \boldsymbol{\theta}_0 + \delta\boldsymbol{\theta}]$ using the finite element model. In order to have a measure on how appropriate the chosen parameters $\boldsymbol{\theta}$ are, one can define an objective function as

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{N} \left(u_i^{fem}(\boldsymbol{\theta}) - u_i^{exp} \right)^2 . \tag{8.10}$$

This objective function is to be minimized with respect to the model parameters $\boldsymbol{\theta}$. Calculating the gradient $\boldsymbol{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ can help to see in which direction the parameters $\boldsymbol{\theta}_i \to \boldsymbol{\theta}_{i+1}$ should be updated. Reiterate this procedure until convergence. In terms of a workflow, the procedure can be formulated as follows:

(1) Define a parametrized material model giving predicted data $u_i^{fem}(\boldsymbol{\theta})$.

(2) Conduct the simulated experiment in real live and gather data u_i^{exp}.

(3) Calculate predictions with finite element model using initial values $\boldsymbol{\theta}_0$.

(4) Reiterate model parameters $\boldsymbol{\theta}_j$ until one has $J(\boldsymbol{\theta}_{j+1}) - J(\boldsymbol{\theta}_j) \to 0$.

26. **Q** — What is a natural frequency?

A — Consider the equation of motion for an undamped harmonic oscillator without forcing terms:

$$m\ddot{x}(t) + kx(t) = 0. \tag{8.11}$$

Here, m is the mass of the oscillating point, $x(t)$ is the position of it and k is the elasticity of the spring material. Now, assuming a solution of the form $x(t) = Ae^{i\omega t}$, the solution needs to fulfill

$$-m\omega^2 Ae^{i\omega t} + kAe^{i\omega t} = 0 \quad \Leftrightarrow \quad (k - m\omega^2)Ae^{i\omega t} = 0. \tag{8.12}$$

The amplitude A and the exponential cannot be zero, hence the solution must fulfill

$$\omega = \pm\sqrt{\frac{k}{m}} \doteq \omega_0, \tag{8.13}$$

which is called the natural frequency of the oscillator. This is the frequency at which this undamped and forcing-free oscillates. It can be shown that exciting a system at one of its natural frequencies leads to resonance, i. e. to ideal absorption of energy and hence to catastrophic outcomes on the system.

27. **Q** — What is a resonance frequency?

A — In the context of mechanical or structural dynamics, the resonance frequency is the frequency at which a system oscillates with the maximum amplitude in response to an external periodic excitation. This phenomenon occurs when the frequency of the external force matches the system's natural frequency, leading to constructive interference and energy accumulation over time.

Let us derive the governing equation for the amplitude of a one-degree-of-freedom damped harmonic oscillator under harmonic forcing, and explain how resonance emerges.

The starting point is the equation of motion for a mass-spring-damper system subjected to a sinusoidal driving force:

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = F_0\cos(\omega t), \tag{8.14}$$

where m is the mass, c is the damping coefficient, k is the stiffness, F_0 is the amplitude of the driving force, ω is the driving frequency and $x(t)$ is the displacement of the mass.

To find the steady-state solution, we assume a particular solution of the form

$$x_p(t) = A(\omega)\cos(\omega t - \phi), \tag{8.15}$$

where $A(\omega)$ is the frequency-dependent amplitude and ϕ is the phase shift.

To solve this, we use the complex exponential form of the forcing function:

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = \Re\{F_0 e^{i\omega t}\}. \tag{8.16}$$

Assuming a complex solution of the form $x(t) = \Re\{X(\omega)e^{i\omega t}\}$, where $X(\omega) \in \mathbb{C}$, substitution into the equation of motion leads to

$$-m\omega^2 X(\omega)e^{i\omega t} + ic\omega X(\omega)e^{i\omega t} + kX(\omega)e^{i\omega t} = F_0 e^{i\omega t} \tag{8.17}$$

which is simplified to

$$X(\omega) = \frac{F_0}{k - m\omega^2 + ic\omega}. \tag{8.18}$$

The amplitude is the modulus of this complex function:

$$A(\omega) = |X(\omega)| = \left| \frac{F_0}{k - m\omega^2 + ic\omega} \right| = \frac{F_0}{\sqrt{(k - m\omega^2)^2 + (c\omega)^2}}. \tag{8.19}$$

Define the undamped natural frequency $\omega_0 = \sqrt{\frac{k}{m}}$ and damping ratio $\beta = \frac{c}{2m}$, the amplitude becomes:

$$A(\omega) = \frac{F_0/m}{\sqrt{(\omega_0^2 - \omega^2)^2 + (2\beta\omega)^2}}. \tag{8.20}$$

The resonance frequency ω_r is the frequency at which this amplitude $A(\omega)$ is maximal. For weak damping ($\beta \ll \omega_0$), the maximum occurs approximately at

$$\omega_r \approx \omega_0 = \sqrt{\frac{k}{m}}. \tag{8.21}$$

At this frequency, the system absorbs energy most efficiently, resulting in oscillations of maximum amplitude. In practice, slight damping shifts the resonance peak to a frequency slightly less than ω_0, and limits the amplitude from becoming unbounded.

28. **Q** — How is the governing equation to calculate resonance frequencies for a FEM problem derived?

A — Let us consider a one-dimensional, time- and space-dependent structural problem governed by the second-order partial differential equation (PDE)

$$\rho(x)\frac{\partial^2 u(x,t)}{\partial t^2} - \frac{\partial}{\partial x}\left(E(x)\frac{\partial u(x,t)}{\partial x}\right) = f(x,t), \quad x \in [0, L] = \Omega, \quad t > 0 \tag{8.22}$$

subject to Dirichlet boundary conditions

$$u(0,t) = u_{x0}, \quad u(L,t) = u_{xL}, \quad t > 0, \tag{8.23}$$

and initial conditions

$$u(x,0) = u_{t0}(x), \qquad \frac{\partial u(x,0)}{\partial t} = v_{t0}(x). \tag{8.24}$$

This is the strong form of the structural dynamics problem. In order to obtain a numerical solution, we must convert it to the weak form. This is done by multiplying both sides of the PDE with a weighting function $\delta u(x,t)$ and integrating over the domain Ω:

$$\int_\Omega \rho(x)\frac{\partial^2 u(x,t)}{\partial t^2}\delta u(x,t)\,\mathrm{d}x - \int_\Omega \frac{\partial}{\partial x}\left(E(x)\frac{\partial u(x,t)}{\partial x}\right)\delta u(x,t)\,\mathrm{d}x = \int_\Omega f(x,t)\delta u(x,t)\,\mathrm{d}x. \tag{8.25}$$

Using integration by parts on the second term and assuming $\delta u(x,t)$ vanishes at the boundaries, we obtain

$$\int_\Omega \rho(x)\frac{\partial^2 u(x,t)}{\partial t^2}\delta u(x,t)\,\mathrm{d}x + \int_\Omega E(x)\frac{\partial u(x,t)}{\partial x}\frac{\partial \delta u(x,t)}{\partial x}\,\mathrm{d}x = \int_\Omega f(x,t)\delta u(x,t)\,\mathrm{d}x. \tag{8.26}$$

Next, we approximate the displacement $u(x,t)$ by a finite-dimensional hypothesis function $u^h(x,t)$:

$$u^h(x,t) = \boldsymbol{H}(x)\boldsymbol{q}(t), \quad \delta u^h(x,t) = \boldsymbol{H}(x)\delta\boldsymbol{q}(t), \tag{8.27}$$

where $\boldsymbol{H}(x) = (h_1(x), \dots, h_n(x))$ are global shape functions and $\boldsymbol{q}(t) = (q_1(t), \dots, q_n(t))^\top$ are the nodal unknowns. Inserting this into the weak form yields

$$\int_\Omega \rho(x)\boldsymbol{H}(x)^\top \boldsymbol{H}(x)\,\mathrm{d}x \cdot \ddot{\boldsymbol{q}}(t) + \int_\Omega E(x)\frac{\mathrm{d}\boldsymbol{H}(x)^\top}{\mathrm{d}x}\frac{\mathrm{d}\boldsymbol{H}(x)}{\mathrm{d}x}\,\mathrm{d}x \cdot \boldsymbol{q}(t) = \int_\Omega f(x,t)\boldsymbol{H}(x)^\top\,\mathrm{d}x. \tag{8.28}$$

We define the global system matrices and load vector as

$$\begin{aligned} \mathbf{M} &\doteq \int_\Omega \rho(x)\boldsymbol{H}(x)^\top \boldsymbol{H}(x)\,\mathrm{d}x, \quad \mathbf{K} \doteq \int_\Omega E(x)\frac{\mathrm{d}\boldsymbol{H}(x)^\top}{\mathrm{d}x}\frac{\mathrm{d}\boldsymbol{H}(x)}{\mathrm{d}x}\,\mathrm{d}x, \\ \boldsymbol{r}(t) &\doteq \int_\Omega f(x,t)\boldsymbol{H}(x)^\top\,\mathrm{d}x, \end{aligned} \tag{8.29}$$

so the resulting semi-discrete system becomes

$$\mathbf{M}\ddot{\boldsymbol{q}}(t) + \mathbf{K}\boldsymbol{q}(t) = \boldsymbol{r}(t). \tag{8.30}$$

This equation represents the equation of motion for the 1D structural dynamics problem in the finite element framework. Time discretization of eq. (8.30) leads to various explicit or implicit time-stepping schemes, depending on accuracy and stability requirements.

29. **Q** — How are resonance frequencies calculated using the eigenvalue method in FEM problems?

A — Recall, that in the case one wants to calculate resonance frequencies for some structural dynamics problem with the finite element method, the problem to solve is

$$\mathbf{M}\ddot{\boldsymbol{q}}(t) + \mathbf{K}\boldsymbol{q}(t) = \boldsymbol{r}(t), \tag{8.31}$$

where $\boldsymbol{r}(t)$ is the external forcing vector - which comprises for example of effects such as energy supplied to the oscillating system -, $\mathbf{K}$ is the stiffness matrix and $\mathbf{M}$ is the mass matrix of the problem. Furthermore, $\boldsymbol{q}(t)$ is the vector containing all position information for all nodes in the problem.

Now, in the case of calculating the natural frequencies - also known as the eigenfrequencies - of the system, the external forcing term is set to zero, i. e. $\boldsymbol{r}(t) = 0$. If that were not be the case, the resulting oscillating frequency would be determined or at leas influenced by the periodicity of $\boldsymbol{r}(t)$. The resulting equation to solve is hence

$$\mathbf{M}\ddot{\boldsymbol{q}}(t) + \mathbf{K}\boldsymbol{q}(t) = 0. \tag{8.32}$$

Either, one solves this system via time-integration using explicit or implicit methods and extracts the frequency components via Fourier transformation from the obtained solution $\boldsymbol{q}(t)$; or one performs eigenvalue calculation. The latter strategy shall be briefly explained below.

Assume, that $\boldsymbol{q}(t)$ as a solution of the form

$$\boldsymbol{q}(t) = \boldsymbol{\phi}e^{i\omega t}, \tag{8.33}$$

where ω is a possible natural frequency of the system. Now, inserting this into eq. (8.32) leads to

$$-\omega^2\mathbf{M}\boldsymbol{\phi}e^{i\omega t} + \mathbf{K}\boldsymbol{\phi}e^{i\omega t} = 0 \quad \Leftrightarrow \quad \left(\mathbf{K} - \omega^2\mathbf{M}\right)\boldsymbol{q}(t) = 0. \tag{8.34}$$

One can now go about using the Cholesky decomposition. The Cholesky decomposition theorem states, that any symmetric and positive-definite matrix $\mathbf{A}$ can be decomposed as

$$\mathbf{A} = \mathbf{B}\mathbf{B}^\top, \tag{8.35}$$

where $\mathbf{B}$ is a unique lower triangular matrix. A matrix $\mathbf{A}$ is symmetric, if $\mathbf{A} = \mathbf{A}^\top$ and it is furthermore positive-definite, if all eigenvalues λ_i of $\mathbf{A}$ are positive, i. e. $\lambda_i > 0$. Now, the mass matrix $\mathbf{M}$ is symmetric by definition and in most cases also positive-definite. Assuming the necessary conditions, we write

$$\mathbf{M} = \mathbf{L}\mathbf{L}^\top \tag{8.36}$$

for a lower triangular matrix $\mathbf{L}$. Writing now $\boldsymbol{q}(t) = (\boldsymbol{L}^\top)^{-1}\boldsymbol{y}$ for a vector $\boldsymbol{y}$, this leads to the equation

$$\mathbf{K}(\boldsymbol{L}^\top)^{-1}\boldsymbol{y} = \omega^2\mathbf{M}(\boldsymbol{L}^\top)^{-1}\boldsymbol{y} = \omega^2\mathbf{L}\boldsymbol{y} \tag{8.37}$$

because of $\mathbf{M} = \mathbf{L}\mathbf{L}^\top$ and $(\mathbf{L}^\top)^{-1}\mathbf{L}^\top = \mathbf{L}^\top(\mathbf{L}^\top)^{-1} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Multiplying the above equation with $\mathbf{L}^{-1}$ from the left and defining $\mathbf{L}^{-\top} \doteq (\mathbf{L}^\top)^{-1}$, one obtains

$$\mathbf{L}^{-1}\mathbf{K}\mathbf{L}^{-\top}\boldsymbol{y} = \omega^2\mathbf{I}\boldsymbol{y}. \tag{8.38}$$

Defining $\mathbf{Q} \doteq \mathbf{L}^{-1}\mathbf{K}\mathbf{L}^{-\top}$ and $\lambda \doteq \omega^2$, one can see that the above equation is the general eigenvalue calculation equation, namely

$$\mathbf{Q}\boldsymbol{y} = \lambda\boldsymbol{y}. \tag{8.39}$$

The eigenvalues λ_i are calculated by finding the roots of the characteristic polynomial $\det(\mathbf{Q} - \lambda_i\mathbf{I})$, namely by calculating

$$\lambda_i \quad \text{such that} \quad \det(\mathbf{Q} - \lambda_i\mathbf{I}) = 0. \tag{8.40}$$

Due to the presupposed positive-definiteness of $\mathbf{M}$, all eigenvalues are positive, which means that the i eigenfrequencies (or natural frequencies) ω_i are calculated form the eigenvalues λ_i as $\omega_i = \sqrt{\lambda_i}$.

30. **Q** — What is Young's modulus?

A — Young's modulus E measures material stiffness, defined as the ratio of axial stress to axial strain under elastic conditions

$$E = \frac{\sigma}{\varepsilon}, \tag{8.41}$$

where σ is axial stress and ε is axial strain. Higher E indicates stiffer materials. Its unit is Pascal (Pa).

31. **Q** — What is the von Mises stress and how is it calculated?

A — The von Mises stress σ_{vM} predicts yielding under complex loading. It is derived from distortion energy theory and is expressed as

$$\sigma_{vM} = \sqrt{\frac{1}{2}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right]}. \tag{8.42}$$

Or in terms of Cartesian stress components we can write

$$\sigma_{vM} = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2 - \sigma_x\sigma_y - \sigma_y\sigma_z - \sigma_z\sigma_x + 3(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2)}. \tag{8.43}$$

Yielding occurs when $\sigma_{vM} \geq \sigma_y$, where σ_y is the yield stress.

32. **Q** — What is Poisson's ratio and its significance?

A — Poisson's ratio ν describes the relationship between axial and transverse strain under uniaxial loading

$$\nu = -\frac{\varepsilon_{\text{transverse}}}{\varepsilon_{\text{axial}}}. \tag{8.44}$$

It typically ranges from 0 to 0.5.

33. **Q** — What are stress and strain and how are they defined?

A — Stress σ is the internal force per unit area

$$\sigma = \frac{F}{A}. \tag{8.45}$$

In 3D, stress is represented by the stress tensor:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix}. \tag{8.46}$$

Strain ε measures relative deformation:

$$\varepsilon = \frac{\Delta L}{L}. \tag{8.47}$$

The strain tensor for 3D includes normal and shear deformations:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} & \gamma_{xy}/2 & \gamma_{xz}/2 \\ \gamma_{xy}/2 & \varepsilon_{yy} & \gamma_{yz}/2 \\ \gamma_{xz}/2 & \gamma_{yz}/2 & \varepsilon_{zz} \end{bmatrix}. \tag{8.48}$$

Stress represents internal resistance, while strain quantifies deformation. Their relationship is governed by laws like Hooke's law.