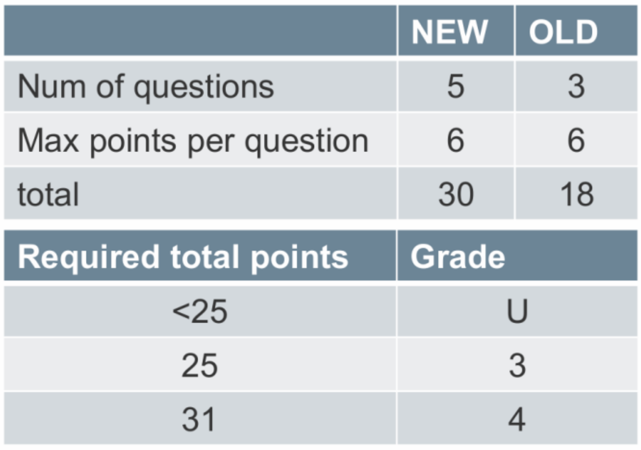
Examination

MHA021 Finite Element Method

Date and time:	2025-08-25, 14.00-18:00
Instructors:	Jim Brouzoulis (phone 2253) and Knut Andreas Meyer (phone 1495). An instructor will visit the exam around 14:30 and 16:30.
Solutions:	Example solutions will be posted within a few days after the exam on the course homepage.
Grading:	Will be posted on Canvas on September 15, the latest.
Review:	Tuesday September 16, 12-13, in meeting room Newton, third floor, Mechanical Engineering building.
Permissible aids:	Chalmers type approved pocket calculator. **Note**: A formula sheet is available as a pdf-file alongside with this exam thesis.

This is a modified version of the re-exam to match the exam structure that will be used for the academic year 25/26. This exam has 5 problems with maximum 6 points per problem. You can have up to 15 points from the hand-in assignments



	NEW	OLD
Num of questions	5	3
Max points per question	6	6
total	30	18

Required total points	Grade
<25	U
25	3
31	4

Exam instructions

All exam problems require a hand-in on paper. For some of the problems, it may be convenient to also use Python including CALFEM. However, note that a complete solution requires deriving and stating all equations hand-written (or computer formatted, not plain text) separately from the code. If you use Python and CALFEM as part of your solutions, you must make sure to also hand in any Python code you have written yourself. You do this by saving your files under `C:\__Exam__\Assignments\` in the appropriate sub-directories created for each problem.

Each Python file that you create must contain your anonymous code as a comment on the first line, e.g.

Code snippet 1 Replace with **your** anonymous code

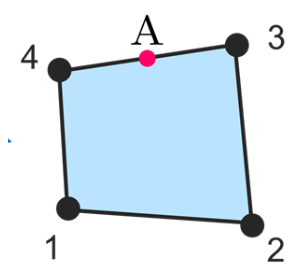
```
# MHA021-1234-XYZ
```

Note that to get Python up and running on the exam computer additional steps are required. Please see the instructions in the folder `Python setup`.

You can utilize these files by copying appropriate files into the sub-directories for the problem where they are needed. Should you need to refer to the CALFEM manual, you can find this also (excluding the examples section) under the folder `Calfem documentation`.

Finally, remember to save any open Python files before you log-out from the computer when you are finished with the exam.

Problem 1



Consider a bilinear quadrilateral element

a) The coordinates for the four nodes (on the element level) are

$$\boldsymbol{x}_1^e = [0.014, 0.010]^T \text{ m}, \quad \boldsymbol{x}_2^e = [0.021, 0.009]^T \text{ m}, \quad \boldsymbol{x}_3^e = [0.018, 0.018]^T \text{ m} \quad \boldsymbol{x}_4^e = [0.012, 0.016]^T \text{ m}$$

The point A is exactly in the middle between nodes 3 and 4. What are the local, $\boldsymbol{\xi}_A$ and global, $\boldsymbol{x}_A$, coordinates of point A? (1.5p)

b) Given an FE-solution with the following nodal displacements of the element (given as 10^{-5} m),

$$\boldsymbol{u}_1^e = [4.1, 1.0]^T, \quad \boldsymbol{u}_2^e = [3.5, 1.6]^T, \quad \boldsymbol{u}_3^e = [3.8, 1.7]^T, \quad \boldsymbol{u}_4^e = [3.7, 1.5]^T$$

Calculate the displcement vector at point A (1.5p)

c) The total potential energy of an FE approximation is usually overestimated compared to the exact solution. What does this imply in terms of the response of the structure? Motivate your answer. (1.5p)

d) Consider a piecewise linear approximation of the unknown field u_h in 1D. If the element size of all elements are halfed, by what factor is the error in the approximation u_h expected to decrease? (1.5p)

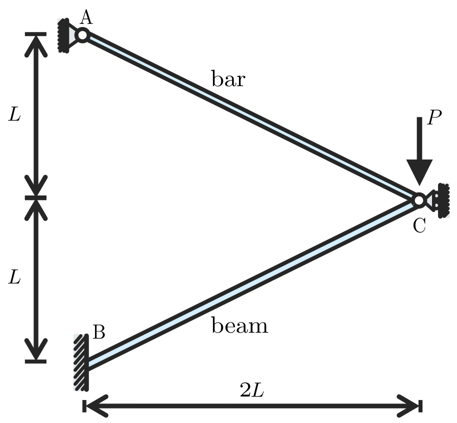
Problem 2

Consider a structure consisting of one bar and one beam element. All properties are here assumed to be equal to 1: E=A=I=rho=P=1

a) Determine the vertical displcement of node C (4.0p)

b) Determine the lowest eigenfrequency in Hertz and draw the corresponding vibration mode (2.0p)

Use the functions bar2m and beam2m to compute element mass matrices (these functions were added to mha021.py 221225)



Problem 3

Consider the bar in Figure 1 which is loaded by distributed axial load, with linearly increasing intensity, given as $b(x) = b_0 \frac{x}{L}$ [N/m]. The strong form for this problem is given as: find the axial displacement $u(x)$ such that

$$\begin{cases} -\dfrac{\mathrm{d}}{\mathrm{d}x}\left[EA\dfrac{\mathrm{d}u}{\mathrm{d}x}\right] = b \quad 0 < x < L \\ u(0) = 0 \\ N(L) = EA\dfrac{\mathrm{d}u}{\mathrm{d}x}(L) = 0 \end{cases}$$

where E is Young's modulus, A is the cross-sectional area.

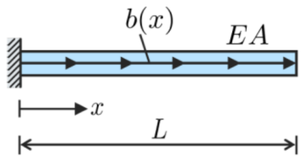


Figure 1: Clamped bar studied in Problem 1.

Tasks:

a) From the strong form, derive the weak form. (1.0p)

b) From the weak form, derive the global finite element formulation $\mathbf{K}\,\mathbf{a} = \mathbf{f}$. (1.0p)

c) Consider the bar discretized into three equally long linear elements. Determine the explicit system of equations $\mathbf{K}\,\mathbf{a} = \mathbf{f}$. Let $E = 200\,\mathrm{GPa}$ and $A = 2 \times 10^{-5}\,\mathrm{m}^2$, $b_0 = 10\,\mathrm{kN/m}$ and $L = 1\,\mathrm{m}$. Approximate the load as piecewise constant over each element. (3.0p)

d) The exact solution to the strong form is $u(x) = \frac{b_0\,x(3L^2-x^2)}{6EAL}$ and the corresponding normal force is $N(x) = \frac{b_0(L^2-x^2)}{2L}$.

Determine the error in normal force at $x = 0$ for the discretization above (1.0p)

Problem 4

Assume a plane stress mechanical FE simulation has been performed of a component with thickness, $t = 0.025\,\mathrm{m}$. The material is assumed linear elastic with Young's modulus, $E = 210\,\mathrm{GPa}$, and Poisson's ratio, $\nu = 0.3$. The constitutive matrix is then given as

$$\boldsymbol{D} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}$$

The simulation used a discretization of bilinear isoparametric quadrilateral elements as can be seen in Figure 4. In this problem, you will only consider a single element (among those used in the FE simulation). The coordinates for the four nodes (on the element level) are

$$\boldsymbol{x}_1^{\mathrm{e}} = [0.014, 0.010]^{\mathrm{T}}\ \mathrm{m}, \quad \boldsymbol{x}_2^{\mathrm{e}} = [0.021, 0.009]^{\mathrm{T}}\ \mathrm{m}, \quad \boldsymbol{x}_3^{\mathrm{e}} = [0.018, 0.018]^{\mathrm{T}}\ \mathrm{m} \quad \boldsymbol{x}_4^{\mathrm{e}} = [0.012, 0.016]^{\mathrm{T}}\ \mathrm{m}$$

a) Use numerical integration with 1x1 and 2x2 Gauss quadrature points to calculate the element area, $A^{\mathrm{e}} = \int_{\Omega} \mathrm{d}\Omega$. Present the two values for the area. (4.0p)

b) Outline the steps to compute the stress in each of the integration points. You do not need to compute the stresses but you should provide a pseudo code showing all necessary steps (write on paper or in Python) (2.0p)

Problem 5

In this problem, we shall solve the heat equation,

$$\boldsymbol{\nabla}^{\mathrm{T}} \boldsymbol{q} = 0 \text{ in } \Omega \tag{2.1}$$

for the inhomogeneous plate with a hole, Ω, shown below.

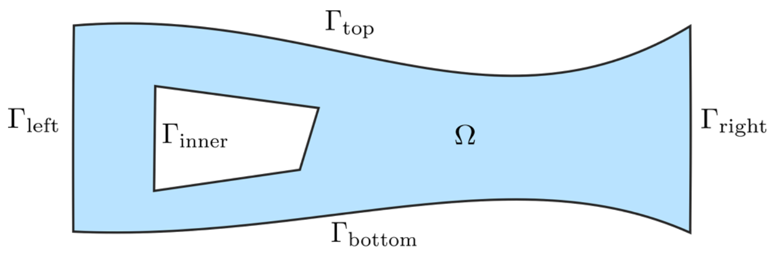


Figure 2: The plate with a hole to be analyzed in this problem.

On the left and right boundaries, Γ_{left} and Γ_{right}, the temperatures, T_{left} and T_{right}, are $0\,°\text{C}$. The inner boundary, Γ_{inner}, is heated by $2\,\text{W/m}^2$ (going into the body). The material is isotropic with heat conductivity, $k = 1\,\text{W/(m°C)}$, such that the heat flux vector can be written $\boldsymbol{q} = -k\,\boldsymbol{\nabla} T$.

To solve the problem, you are given a mesh heat_flow.pkl (shown below) of Ω with linear triangle elements consisting of the following parts:

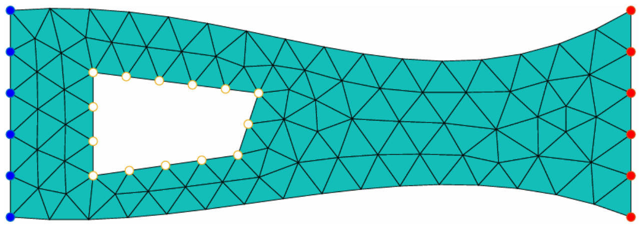


Figure 3: Markers show the nodes included in the boundary sets.

Tasks:

a) Based on the description above, state the boundary conditions on equation form for every part of the boundary **(0.5p)**

b) Derive the weak form of the heat equation, using the problem-specific boundary conditions **(0.5p)**

c) Derive the global FE form of the heat equation for the problem at hand **(0.5p)**

d) Use Python to solve the FE problem, and calculate the norm of the temperature degree of freedoms. **(3.0p)**
Hint 1: You may use the CALFEM function `flw2te` to calculate the element cont (4.5p)
Hint 2: If you are no the function flow2t_Ke_fe ro Neumann boundary conditions, you mav set $T_{\text{right}} = 10\,°\text{C}$ instead of $0\,°\text{C}$ to keep working with this problem. Please state clearly

Use the starter code on the next page

Starter code for Problem 5

```
# Load the mesh from file
import pickle


filename = "heat_flow.pkl"
with open(filename, "rb") as f:
    mesh = pickle.load(f)


fig = plot_mesh(mesh.nodes, mesh.elements, mesh.edges,
show_node_ids=True)
fig.show()
displayvar("\\text{edge nodes in mesh}",mesh.edges)


# Input data
t = 1 # thickness [m]
k = 1 # conductivity [W/m]
T0 = 0 # Prescribed temperature on edges
h = 2 # prescribed heat flux on the inner boundary [W/m^2]


# WRITE YOUR SOLUTION BELOW
```

Solutions

Problem 1: solution

a) The local coordinate of point A is, $\xi_A = [0, 1]^T$, which gives with the shape functions in the formula sheet for a 4-noded quadrilateral,

$$\mathbf{x}_A = \sum_{\alpha=1}^{4} \mathbf{x}_\alpha \hat{N}_\alpha(\xi_A) = \begin{bmatrix} 0.0150 \\ 0.0170 \end{bmatrix} \text{ m} \tag{3.1}$$

b) Given the displacement vector, we transform this into the dof vector as $\mathbf{a}^e = [u^e_{x,1}, u^e_{y,1}, u^e_{x,2}, \cdots]^T$, and then we have the vectorized shape functions,

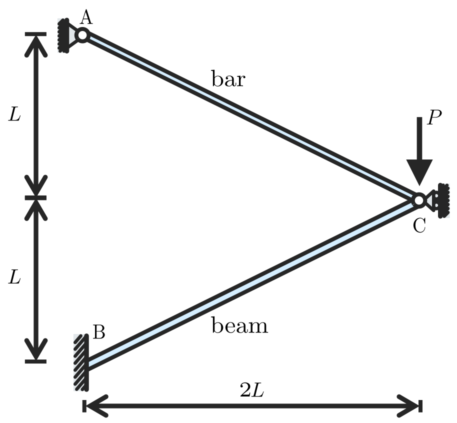
$$\mathbf{N}^e = \begin{bmatrix} M_1 & 0 & M_2 & \cdots \\ 0 & M_1 & 0 & \cdots \end{bmatrix} \tag{3.2}$$

and the displacement at the point $\mathbf{x}_A$ is then given as

$$\mathbf{u}_A = \mathbf{N}^e \mathbf{a}^e = \begin{bmatrix} 0.0375 \\ 0.0160 \end{bmatrix} \text{ mm} \tag{3.3}$$

c) & d) See lecture notes

Problem 2 solution



```
import numpy as np
from mha021 import *
# 2a
E = A = P = L = I = rho = 1.0
A2 = A
Ex = np.array([
    [0, 2*L],
    [2*L, 0]
])
Ey = np.array([
    [0,  L],
    [L, 2*L]
])
fig = draw_discrete_elements(Ex, Ey, annotate='both')
fig.show()
num_dof = 8
num_el = 2
K = np.zeros((num_dof,num_dof))
f = np.zeros(num_dof)
ex1 = Ex[0, :]
ey1 = Ey[0, :]
ex2 = Ex[1, :]
ey2 = Ey[1, :]
dofs1 = [1, 2, 3, 4, 5, 6]
dofs2 = [4, 5, 7, 8]

K1 = beam2e(ex1, ey1, E, A, I=I)
K2 = bar2e(ex2, ey2, E, A=A)
assem(K, K1, dofs=dofs1)
assem(K, K2, dofs=dofs2)

# supports at A and B, vertical load at C:
bc_dofs = [1, 2, 3, 4, 7, 8]  # fix appropriate DOFs
bc_vals = [0, 0, 0, 0, 0, 0]  # all should be zero
f[5-1] += -P # vertical DOF at C
a, r = solve_eq(K, f, bc_dofs, bc_vals)
displayvar("a", a, accuracy=4)
```

```
# 2b

M1 = beam2m(ex1, ey1, rho, A)
M2 = bar2m(ex2, ey2, rho, A)
M = np.zeros((num_dof,num_dof))
assem(M, M1, dofs=dofs1)
assem(M, M2, dofs=dofs2)


free_dofs = [5, 6]
K_red = extract_block(K, free_dofs)
M_red = extract_block(M, free_dofs)
displayvar("M", M_red, accuracy=3)


omega2, phi = eigh(K_red, M_red)
f = np.sqrt(omega2)/(2*np.pi)
displayvar("f_{min}", f[0])
displayvar("\phi", phi[:, 0])
```

Problem 3 solution

3a

Multiply the differential equation with an (arbitrary) test function $v(x)$ and integrate over the interval

$$-\int_0^L v\frac{d}{dx}\left[EA\frac{du}{dx}\right]\,\mathrm{d}x = \int_0^L v\,b\,\mathrm{d}x$$

IBP. of the left-hand side gives (after moving the boundary terms to the right-hand side)

$$\int_0^L EA\,\frac{dv}{dx}\,\frac{du}{dx}\,\mathrm{d}x = \int_0^L v\,b\,\mathrm{d}x + v(L)\left[EA\,\frac{du}{dx}\right]_{x=L} - v(0)\left[EA\,\frac{du}{dx}\right]_{x=0}$$

At $x = L$, we can substitute the the boundary term given in the strong form ($EA\,\dfrac{\mathrm{d}u}{\mathrm{d}x}(0) = 0$) $\Rightarrow$ the weak form:

Find u such that

$$\int_0^L EA\,\frac{dv}{dx}\,\frac{du}{dx}\,\mathrm{d}x = \int_0^L v\,b\,\mathrm{d}x - v(0)\,\underbrace{\left[EA\,\frac{du}{dx}\right]_{x=0}}_{\text{reaction force}=R} = \int_0^L v\,b\,\mathrm{d}x - v(0)\,R$$

where $u(0) = 0$.

3b

Approximate u using a linear combination of shape functions: $u \approx u_h = \sum_i N_i(x)a_i = \mathbf{N}\,\mathbf{a}$, where $N = [N_1(x)\ N_2(x)\ldots N_n(x)]$ is a row vector with the shape functions and $a = \begin{bmatrix}a_1\ a_2\ \ldots\ a_n\end{bmatrix}^{\mathrm{T}}$ are the degrees of freedom. Use the same shape functions for the test function $v(x) = \sum_i N_i(x)c_i = \mathbf{N}\,\mathbf{c}$. Inserting into the weak form while using the notation $\dfrac{\mathrm{d}}{\mathrm{d}x}\mathbf{N}(x) = \mathbf{B}$ gives

$$\mathbf{c}^{\mathrm{T}}\int_0^L EA\,\mathbf{B}^{\mathrm{T}}\,\mathbf{B}\,\mathrm{d}x\,\mathbf{a} = \mathbf{c}^{\mathrm{T}}\int_0^L \mathbf{N}^{\mathrm{T}}\,b\,\mathrm{d}x - \mathbf{c}^{\mathrm{T}}\,\mathbf{N}^{\mathrm{T}}(L)R$$

which should hold for arbitrary $\mathbf{c}$ $\Rightarrow$...

$$\int_0^L EA\,\mathbf{B}^{\mathrm{T}}\,\mathbf{B}\,\mathrm{d}x\,\mathbf{a} = \int_0^L \mathbf{N}^{\mathrm{T}}\,b\,\mathrm{d}x - \mathbf{N}^{\mathrm{T}}(0)R$$

or simply $\mathbf{K}\,\mathbf{a} = \mathbf{f}_l + \mathbf{f}_{\mathrm{b}}$ with

$$\mathbf{K} = \int_0^L EA\,\mathbf{B}^{\mathrm{T}}\,\mathbf{B}\,\mathrm{d}x,\quad \mathbf{f}_{\mathrm{l}} = \int_0^L \mathbf{N}^{\mathrm{T}}\,b\,\mathrm{d}x,\quad \mathbf{f}_{\mathrm{b}} = -\mathbf{N}^{\mathrm{T}}(0)R$$

3c

Assembly of $\mathbf{K}$ and $\mathbf{f}_l$ (using the mid-point of each element to evaluate b) gives the system of equations

$$\begin{bmatrix} 12000000. & -12000000. & 0. & 0. \\ -12000000. & 24000000. & -12000000. & 0. \\ 0. & -12000000. & 24000000. & -12000000. \\ 0. & 0. & -12000000. & 12000000. \end{bmatrix} \begin{bmatrix} 0 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} 277.8 \\ 1111.1 \\ 2222.2 \\ 1388.9 \end{bmatrix} + \begin{bmatrix} -R \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

3d

Using linear shape functions: $N_1^e = -\frac{1}{L}(x - x_{i+1})$ and $N_2^e = \frac{1}{L}(x - x_i)$ gives

$$\mathbf{N}^e = [N_1^e, N_2^e] \quad \Rightarrow \mathbf{B}^e = \frac{\mathrm{d}}{\mathrm{d}x}\mathbf{N}^e = \frac{1}{L^e}[-1, 1]$$

The normal force is computed as $N_h^e = EA\,\mathbf{B}^e\,\mathbf{a}^e$ which gives for element 1:

$$N_h^1 = EA\,\mathbf{B}^1\,\mathbf{a}^1 = EA\frac{1}{L^e}[-1, 1]\begin{bmatrix} 0 \\ a_2 \end{bmatrix} \approx 4722\,\mathrm{N}$$

which gives an error $e_N = \frac{N(0) - N_h^1}{N(0)} \approx -5.6\,\%$

Problem 4 solution

a)

Calculating the integral implies that we have

$$\int_{\Omega^e} \mathrm{d}\Omega \approx \sum_{\alpha=1}^{4} w_\alpha \det(\mathbf{J}(\xi_\alpha)) = 50\,\mathrm{mm}^2$$

See Python code for computing the integral. You should get the same value regardless of number of integration points.

b) See lecture notes and/or study the function used in CA2 to compute stress

Problem 5 solution

a)

Since nothing is given on the top and bottom boundaries, we assume that these are insulated, i.e. $q_n = 0$. We then set up the boundary conditions formally as

$$T(\mathbf{x}) = 0\,°\mathrm{C}, \quad \mathbf{x} \text{ on } \Gamma_{\text{left}} \tag{2.1}$$

$$T(\mathbf{x}) = 0\,°\mathrm{C}, \quad \mathbf{x} \text{ on } \Gamma_{\text{right}} \tag{2.2}$$

$$\mathbf{q}^{\mathrm{T}}(\mathbf{x})\mathbf{n} = 0\,\mathrm{W/m^2}, \quad \mathbf{x} \text{ on } \Gamma_{\text{bottom}} \tag{2.3}$$

$$\mathbf{q}^{\mathrm{T}}(\mathbf{x})\mathbf{n} = 0\,\mathrm{W/m^2}, \quad \mathbf{x} \text{ on } \Gamma_{\text{top}} \tag{2.4}$$

$$\mathbf{q}^{\mathrm{T}}(\mathbf{x})\mathbf{n} = 2\,\mathrm{W/m^2}, \quad \mathbf{x} \text{ on } \Gamma_{\text{inner}} \tag{2.5}$$

b)

We multiply the Partial Differential Equation (PDE) by an arbitrary scalar test function, $v(\mathbf{x})$, and integrate over the domain, Ω, resulting in

$$\int\limits_{\Omega} v\ \nabla^{\mathrm{T}}\mathbf{q}\ \mathrm{d}\Omega = 0 \tag{2.6}$$

Applying Green-Gauss theorem to the left hand side, results in

$$\int\limits_{\Gamma} v\ \mathbf{n}^{\mathrm{T}}\mathbf{q}\ \mathrm{d}\Gamma - \int\limits_{\Omega} [\nabla v]^{\mathrm{T}}\,\mathbf{q}\ \mathrm{d}\Omega = 0 \tag{2.7}$$

Finally, we split the boundary terms into the Dirichlet ($\Gamma_g = \Gamma_{\text{left}} \cup \Gamma_{\text{right}}$) and Neumann ($\Gamma_h = \Gamma_{\text{top}} \cup \Gamma_{\text{bottom}}$ and Γ_{inner}) parts, insert the constitutive law, $\mathbf{q} = -k\nabla T$, and re-arrange to get

$$\int\limits_{\Omega} [\nabla v]^{\mathrm{T}}\,[k\ \nabla T]\ \mathrm{d}\Omega = -\int\limits_{\Gamma_h} v\ q_{\mathrm{n}}\ \mathrm{d}\Gamma - \int\limits_{\Gamma_{\text{inner}}} v\ q_{\mathrm{n}}\ \mathrm{d}\Gamma - \int\limits_{\Gamma_g} v\ \mathbf{n}^{\mathrm{T}}\mathbf{q}\ \mathrm{d}\Gamma \tag{2.8}$$

Still subject to the constraint, $T(\mathbf{x}) = 0\,°\mathrm{C}$ for x on Γ_{g}. Inserting that $q_{\mathrm{n}} = 0$ on Γ_h, we get

$$\int\limits_{\Omega} [\nabla v]^{\mathrm{T}}\,[k\ \nabla T]\ \mathrm{d}\Omega = -\int\limits_{\Gamma_{\text{inner}}} v\ q_{\mathrm{n}}\ \mathrm{d}\Gamma - \int\limits_{\Gamma_g} v\ \mathbf{n}^{\mathrm{T}}\mathbf{q}\ \mathrm{d}\Gamma \tag{2.9}$$

c)

To get the FE form, we introduce the approximation of the test function, $v(\mathbf{x}) \approx v_h(\mathbf{x}) = \sum_{i=1}^{N_d} N_i(\mathbf{x}) c_i = \mathbf{N}\,\mathbf{c}$, where c_i are arbitrary weights, that are coordinate independent. This results in

$$\sum_{i=1}^{N_d} c_i \underbrace{\left[\int_\Omega [\nabla N_i(\mathbf{x})]^\mathrm{T} [k\ \nabla T]\ \mathrm{d}\Omega - \left[-\int_{\Gamma_\text{inner}} N_i(\mathbf{x})\ q_\text{n}\ \mathrm{d}\Gamma - \int_{\Gamma_g} N_i(\mathbf{x})\ \mathbf{n}^\mathrm{T}\mathbf{q}\ \mathrm{d}\Gamma\right]\right]}_{r_i} = 0 \tag{2.10}$$

Since c_i are arbitrary coefficients, and this has to hold for any c_i, this implies that $r_i = 0$ has to hold. We can see this by considering the case that $c_\alpha = 1$, and $c_i = 0$ if $i \neq 0$. Then $r_\alpha = 0$ follows, and this has to hold for any $1 \leq \alpha \leq N_d$. We thus obtain the FE form,

$$\int_\Omega [\nabla N_i(\mathbf{x})]^\mathrm{T} [k\ \nabla T]\ \mathrm{d}\Omega = -\int_{\Gamma_\text{inner}} N_i(\mathbf{x})\ q_\text{n}\ \mathrm{d}\Gamma - \int_{\Gamma_g} N_i(\mathbf{x})\ \mathbf{n}^\mathrm{T}\mathbf{q}\ \mathrm{d}\Gamma \tag{2.11}$$

and can finally insert the FE-approximation for the temperature, $T(\mathbf{x}) \approx T_h(\mathbf{x}) = \sum_{i=1}^{N_d} N_i(\mathbf{x}) a_i = \mathbf{N}\mathbf{a}$, to obtain,

$$\underbrace{\int_\Omega [\nabla N_i(\mathbf{x})]^\mathrm{T} [k\ \nabla N_j]\ \mathrm{d}\Omega}_{K_{ij}}\ a_j = \underbrace{-\int_{\Gamma_\text{inner}} N_i(\mathbf{x})\ q_\text{n}\ \mathrm{d}\Gamma - \int_{\Gamma_g} N_i(\mathbf{x})\ \mathbf{n}^\mathrm{T}\mathbf{q}\ \mathrm{d}\Gamma}_{f_i} \tag{2.12}$$

$$T_h(\mathbf{x}) = 0\,{}^\circ\mathrm{C} \text{ on } \Gamma_\text{left} \tag{2.13}$$

$$T_h(\mathbf{x}) = 0\,{}^\circ\mathrm{C} \text{ on } \Gamma_\text{right} \tag{2.14}$$

where the reaction flux, $q_n(\mathbf{x})$ is unknown on Γ_g and $q_n(\mathbf{x}) = 2\,\mathrm{W/m^2}$ on Γ_inner.

d) See Pythoon code