This document contains an overview of typical exam questions for the FEM part of Unit 1 *Numerical Modeling of Construction Materials* of the module CIEM1110. Percentages indicate the work load, where 100% would be the size of a complete exam. The level of difficulty is representative for what you can expect in the actual exam. This collection is not exhaustive in the range of topics that may be covered in the exam.

Questions with solutions

[5%] 1. (a) Expand the B-matrix for a 4-node element in two-dimensional elasticity in terms of shape functions N_1 to N_4.

Solution:

$$\mathbf{B} = \begin{bmatrix} N_{1,x} & 0 & \cdots & N_{4,x} & 0 \\ 0 & N_{1,y} & \cdots & 0 & N_{4,y} \\ N_{1,y} & N_{1,x} & \cdots & N_{4,y} & N_{4,x} \end{bmatrix}$$

[5%] (b) Show the role of the Jacobian in the construction of the B-matrix for isoparametric elements.

Solution: The Jacobian linearizes the relation between (x, y) and (η, ξ)

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}$$

Derivatives with respect to global coordinates are obtained with the inverse of $\mathbf{J}$ as

$$\begin{bmatrix} N_{i,x} \\ N_{i,y} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} N_{i,\xi} \\ N_{i,\eta} \end{bmatrix}$$

These then go into the $\mathbf{B}$ matrix as shown under (a).

[5%] (c) When evaluating the stiffness matrix, where does the term det(J) appear in the formulation?

Solution: In the integral in front of $\mathrm{d}\xi\,\mathrm{d}\eta$

$$\iint \mathbf{B}^T\mathbf{D}\mathbf{B}\,\det(\mathbf{J})\,\mathrm{d}\xi\,\mathrm{d}\eta$$

[5%] (d) Explain why the term det(J) is needed.

Solution: It is there to correct for the fact that integration is performed over the reference element while the stiffness matrix is defined in the physical element domain

Note that the questions do not require an explicit expression of the shape functions.

2. A researcher is investigating the mass diffusion behavior of a new type of material and opts for performing a diffusion FE simulation with backward Euler time integration. However, the size and number of simulations to be performed start to create a computational bottleneck.

[5%] (a) The researcher decides to switch to a **forward Euler** scheme. For the same model and time step size, how would this choice impact execution time, model accuracy and numerical stability?

Solution: Execution time would be significantly faster due to the explicit nature of the scheme and because the system can be solved much faster through lumping the **M** matrix. Accuracy would scale with time step in a similar manner, but stability would now be much poorer.

[5%] (b) Another way to accelerate the simulations is to increase time step size while sacrificing as least as possible the accuracy of the scheme. With this in mind, what other time integrator could be recommended and why?

Solution: The trapezoidal scheme with $\theta = \frac{1}{2}$ would be a good choice here. It is second-order accurate with time step size (backward Euler is only first-order accurate) while still being unconditionally stable and therefore allowing one to increase time step size without stability concerns.

[25%] 3. Starting from the following pseudocode for computing the element stiffness matrix of a general isoparametric element:

```

    function get_element_stiffness (elem)
      k_ele = 0.0;

      % your code should come

      % between these lines

      return k_ele;

    end function


```

Use the coding elements below to complete the function:

- `for ip = 1..nip`
- `end for`
- `get_node_coordinates (arg)`
- `get_B_matrix (arg)`
- `get_ip_weight (arg)`
- `get_nodes_for_elem (arg)`
- `get_jacobian_at_ip (arg1, arg2)`
- `get_shapegrads_at_ip (arg)`
- `get_D_matrix ()`

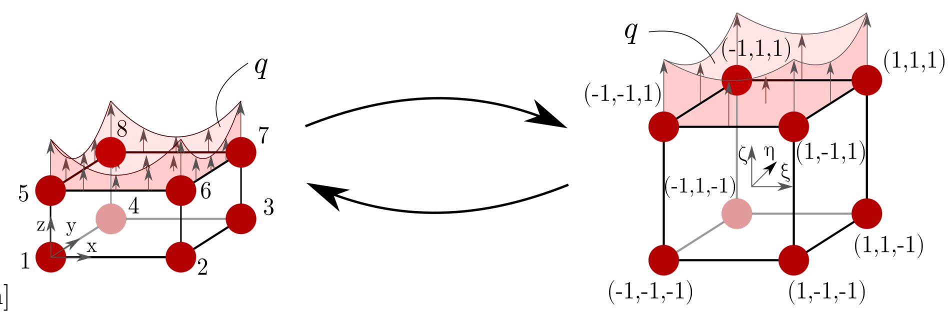
Note that some functions have arguments. When writing your answer, clearly indicate which arguments are passed to each function. Assume all functions return a single variable to which you can assign a name of your choice. You can then use these new variables as arguments for subsequent function calls. Make sure to use all of the above elements when writing your code, adding more lines with basic mathematical operations where necessary. You can also make use of the following operators:

- `matmul(A,b)`: matrix/vector product
- `matmul(A,B)`: matrix/matrix product
- `matmul(A,B,C)`: triple matrix product
- `a * B`: scalar product
- `inverse(A)`: matrix inverse
- `transpose(A)`: matrix transpose
- `determinant(A)`: matrix determinant

Solution:

```
    function get_element_stiffness (elem)
      k_ele = 0.0;

      nodes  = get_nodes_for_elem (elem);
      coords = get_node_coordinates (nodes);
      dmat   = get_D_matrix ();
      thick  = get_thickness ();

      for ip = 1..nip
        weight   = get_ip_weight (ip);
        jacobian = get_jacobian_at_ip (coords, ip);
        pgrads   = get_shapegrads_at_ip (ip);

        grads = matmul(inverse(jacobian), pgrads);
        bmat  = get_B_matrix (grads)

        k_ele = k_ele + matmul(transpose(bmat),matmul(dmat,bmat)) * weight * thick *
  determinant(jacobian)
      end for

      return k_ele;
    end function
```

[30%] 4. Consider the following 8-node brick element, to which a surface load is applied on the face defined by nodes 5, 6, 7 and 8:



[h]

Nodal coordinates in x-y-z space are given by:

Node	x	y	z
1	0.0	0.0	0.0
2	3.0	0.0	0.0
3	3.0	2.0	0.0
4	0.0	2.0	0.0
5	0.0	0.0	1.0
6	3.0	0.0	1.0
7	3.0	2.0	1.0
8	0.0	2.0	1.0

and the DOFs are arranged in the element vector in the order:

$$\mathbf{a}^{(e)} = \begin{bmatrix} u_{x1} & u_{y1} & u_{z1} & \cdots & u_{x8} & u_{y8} & u_{z8} \end{bmatrix}^{\mathrm{T}}$$

The magnitude of the surface load varies in space and can be described in isoparametric coordinates as:

$$q = 1 + \xi^2 + \eta^2$$

while shape functions can be given by the compact expression:

$$N_i = \frac{1}{8}\left(1 + \xi_i \xi\right)\left(1 + \eta_i \eta\right)\left(1 + \zeta_i \zeta\right)$$

Based on the information given, answer the following:

- **Compute the surface integral related to the load q** to find the equivalent nodal forces acting on the element. Perform the integration **numerically** and give the equivalent load vector computed with 1x1 Gauss integration. **Clearly show all steps taken**
- How can this integral be evaluated exactly with numerical integration? **Specify the number of integration points** that is needed and **show the procedure for performing the numerical integration** (you do not need to work out the math to arrive)

For integration points and weights, refer to the following definition of 1D Gauss integration:

# points	location(s)	weight(s)
1	0.0	2
2	$\pm\frac{1}{\sqrt{3}}$	1
3	$\pm\sqrt{\frac{3}{5}}$	$\frac{5}{9}$
	0	$\frac{8}{9}$

Solution:

The integral that gives the equivalent nodal forces for load **q** can be written as:

$$\mathbf{f} = \int_{\Gamma} \mathbf{N}^{\mathrm{T}} \mathbf{q} \, \mathrm{d}\Gamma \tag{1}$$

where **f** has size $[24 \times 1]$ and $\mathbf{q} = \begin{pmatrix} 0 & 0 & q \end{pmatrix}^{\mathrm{T}}$.

Noting that Γ corresponds to a Q4 element and q acts in the out-of-plane direction, we can evaluate this integral in a straightforward way by writing in parametric coordinates:

$$\bar{\mathbf{f}} = \int_{-1}^{+1} \int_{-1}^{+1} \overline{\mathbf{N}}^{\mathrm{T}} q j \, \mathrm{d}\xi \mathrm{d}\eta \tag{2}$$

where now the force has size $[4 \times 1]$, the load q becomes a scalar and $\overline{\mathbf{N}}$ is formed by arranging the shape functions evaluated at $\zeta = 1$ in a single row:

$$\overline{\mathbf{N}} = \begin{bmatrix} N_5 & N_6 & N_7 & N_8 \end{bmatrix} \tag{3}$$

and the shape functions are given by the compact expression:

$$N_i = \frac{1}{4}\left(1 + \xi_i \xi\right)\left(1 + \eta_i \eta\right) \tag{4}$$

In order to compute the integral, the determinant of the Jacobian j of the surface is needed:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{3}{2} & 0 \\ 0 & 1 \end{bmatrix} \quad \Rightarrow \quad j = \frac{3}{2} \tag{5}$$

The question asks for a numerical integration of $\bar{\mathbf{f}}$ with 1x1 Gauss integration. For that, we compute the integrand at $\xi = \eta = 0$ and multiply it by a weight $w = 4$ (two-dimensional integral), giving:

$$\bar{\mathbf{f}} \approx \overbrace{\frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}}^{\overline{\mathbf{N}}^{\mathrm{T}}(\xi=0,\eta=0)} \cdot \underbrace{(1+0+0)}_{q(\xi=0,\eta=0)} \cdot \overbrace{\frac{3}{2}}^{j} \cdot \underbrace{2 \cdot 2}_{w} = \begin{bmatrix} \frac{3}{2} \\ \frac{3}{2} \\ \frac{3}{2} \\ \frac{3}{2} \end{bmatrix} \tag{6}$$

which can then be placed in the full-size equivalent load vector for the brick element:

$$\mathbf{f} = \begin{bmatrix} \mathbf{0}_{[1\times 12]} & 0 & 0 & 3/2 & 0 & 0 & 3/2 & 0 & 0 & 3/2 \end{bmatrix}^{\mathrm{T}} \tag{7}$$

For exact numerical integration, we need to look at the polynomial degree of the integrand. Since $\overline{\mathbf{N}}$ has linear terms, q has quadratic terms and j is constant, the integrand is of order 3. With Gauss quadrature, it can be integrated exactly with 4 integration points (2x2):

$$\bar{\mathbf{f}} = \sum_{i}^{2}\sum_{j}^{2} \overline{\mathbf{N}}\left(\xi_i, \eta_j\right) q\left(\xi_i, \eta_j\right) j w_i w_j \tag{8}$$

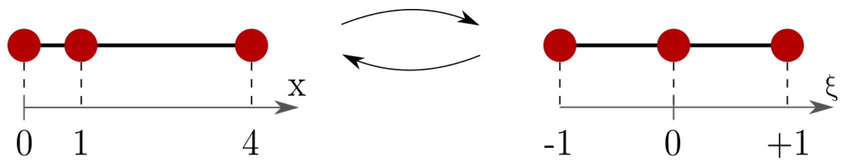
with point locations and weights given by:

ξ_i	η_j	w_i	w_j
$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	1	1
$\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	1	1
$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	1	1
$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	1	1

[25%] 5. Consider a 3-node line element for a bar problem with shape functions defined in the natural coordinate frame as:

$$N_1 = \frac{1}{2}\xi\left(\xi - 1\right) \quad N_2 = 1 - \xi^2 \quad N_3 = \frac{1}{2}\xi\left(\xi + 1\right)$$

The nodes of the element are positioned at $x_1 = 0$, $x_2 = 1$ and $x_3 = 4$, as shown in the figure below.



Demonstrate that this element can represent infinite strain inside the element with finite nodal displacements.[1]

Solution:

We start with the interpolation for strains:

$$\varepsilon = \frac{\partial \mathbf{N}}{\partial x}\mathbf{a}^e \tag{9}$$

since the shape functions are given in parametric coordinates, the chain rule must be applied:

$$\frac{\mathrm{d}\mathbf{N}}{\mathrm{d}x} = \frac{\mathrm{d}\mathbf{N}}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x} \tag{10}$$

From the definition of isoparametric mapping and the given coordinates we have:

$$x = N_i(\xi)x_i \quad \Rightarrow \quad x = 1 + \xi^2 + 2\xi \tag{11}$$

from which we can compute the jacobian:

$$j = \frac{\mathrm{d}x}{\mathrm{d}\xi} = 2\xi + 2 \tag{12}$$

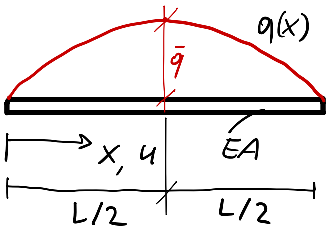
Going back to the definition of strains and substituting j and the derivatives of the shape functions with respect to ξ we have:

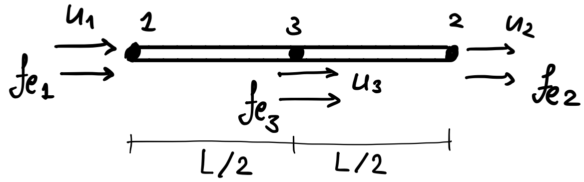
$$\varepsilon = \frac{1}{2\xi + 2}\begin{bmatrix} \xi - \frac{1}{2} & -2\xi & \xi + \frac{1}{2} \end{bmatrix}\begin{bmatrix} u_{x1} \\ u_{x2} \\ u_{x3} \end{bmatrix} \tag{13}$$

from which we can see that if $\xi \to -1$, the denominator $2\xi + 2 \to 0$ and consequently $\varepsilon \to \infty$, $\forall\ \mathbf{a}^e \neq \mathbf{0}$. This distorted element can therefore represent an infinite strain at its left node even when the nodal displacements are finite.

[35%] 6. Derive the expression of the vector of nodal equivalent forces corresponding to the parabolic distributed load $q\left(x\right)$ for the 3-node bar element depicted below starting from the differential equation of the bar. Use a 1- and 2-point quadrature rules in $[-1; 1]$ (location $\pm 1/\sqrt{3}$ and weights equal to 1 for the two-point rule).

[1] This question has been asked in an online open book exam. For a closed-book written exam, the question would likely be broken down in smaller steps





Solution: Strong form

$$-EAu_{,xx} = q(x)$$

Weak form equation

$$-\int_{-L/2}^{L/2} wEAu_{,xx}\,\mathrm{d}x = \int_{L/2}^{L/2} wq(x)\,\mathrm{d}x$$

Discretize and divide by amplitudes (focus on RHS)

$$\mathbf{f} = \int_{-L/2}^{L/2} \mathbf{N}^T q\,\mathrm{d}x$$

Shape functions

$$\mathbf{N}(x) = \left[-\tfrac{x}{L} + \tfrac{2x^2}{L^2}, \quad \tfrac{x}{L} + \tfrac{2x^2}{L^2}, \quad 1 - \tfrac{4x^2}{L^2}\right]$$

or

$$\mathbf{N}(\xi) = \left[\tfrac{1}{2}\xi^2 - \tfrac{1}{2}\xi, \quad \tfrac{1}{2}\xi^2 + \tfrac{1}{2}\xi, \quad -\xi^2 + 1\right]$$

or

$$\mathbf{N}(\xi) = \left[\tfrac{2}{L^2}x^2 - \tfrac{3}{L}x + 1, \quad \tfrac{2}{L^2}x^2 - \tfrac{1}{L}x, \quad \tfrac{4}{L^2}x^2 + \tfrac{4}{L}x\right]$$

Next, integration points have to be computed in x-coordinate if the first set of shape functions is used, and the jacobian determinant ($|j| = L/2$) should be added to the integral. The integral is expanded as

$$\mathbf{f} \approx \sum_{i=1}^{\mathrm{np}} \mathbf{N}^T(\xi_i) q(\xi_i) |j| w_i$$

Result with 1 point:

$$\mathbf{f} = \begin{bmatrix}0 & 0 & 1\end{bmatrix} \bar{q}L$$

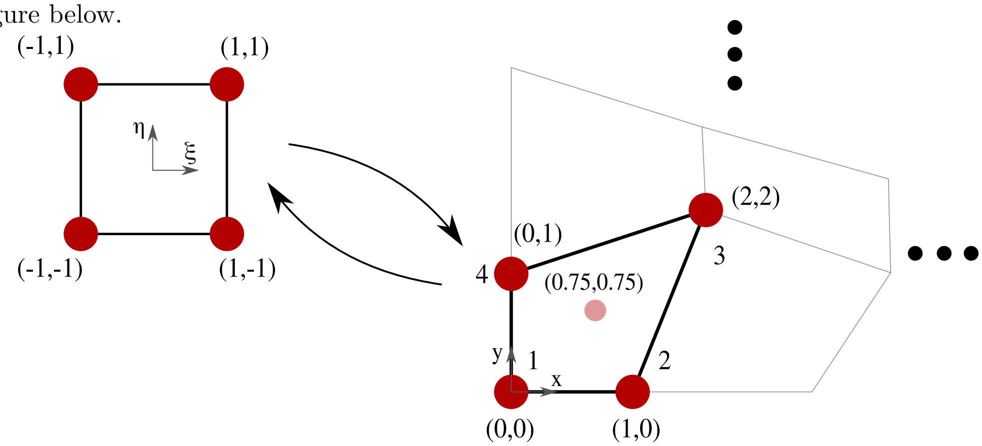
with 2 points:

$$\mathbf{f} = \begin{bmatrix}\tfrac{1}{9} & \tfrac{1}{9} & \tfrac{4}{9}\end{bmatrix} \bar{q}L$$

(for node numbering with mid-node as third; mid-node as second is also allowed, as long as the numbering is consistent)

Questions without solutions

1. An equilibrium FEM problem is solved with a mesh of distorted Q4 elements, some of which can be seen in the figure below.



The system is solved and a global DOF vector $\mathbf{a}$ is computed. From this global vector, the nodal displacements of the element highlighted in the figure can be extracted:

$$\mathbf{a}^{(\mathrm{e})} = \begin{bmatrix} 0 & 0 & 0.2 & 0 & 0.4 & 0.1 & 0 & 0.3 \end{bmatrix}^{\mathrm{T}}$$

where DOFs are ordered as $\mathbf{a}^{(\mathrm{e})} = \begin{bmatrix} a_{x1} & a_{y1} & \cdots & a_{x4} & a_{y4} \end{bmatrix}$. The shape functions for this element can be written in the parametric system $\xi - \eta$ as:

$$N_i = \frac{1}{4} \left(1 + \xi_i \xi\right) \left(1 + \eta_i \eta\right)$$

and the two coordinate systems are related through the Jacobian matrix and its inverse:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad \mathbf{J}^{-1} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix}$$

For this problem, the material is modeled as linear-elastic with stiffness matrix given by:

$$\mathbf{D} = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 500 \end{bmatrix}$$

[10%] (a) Using isoparametric mapping, prove that the centroid of the element in $\xi - \eta$ space corresponds to the centroid of the distorted element in physical $x - y$ space (shown in light red in the figure)

[10%] (b) Describe the procedure to compute the stress vector $\boldsymbol{\sigma}$ at an arbitrary location (ξ, η) inside the element. Employ the isoparametric mapping formalism and for now assume general values for node coordinates, $\mathbf{a}^{(e)}$ and $\mathbf{D}$.

[20%] (c) Use the procedure you just described to compute the value of the horizontal stress σ_{xx} at point $(0.75, 0.75)$ in physical space. You can simplify your calculations by taking advantage of the zero entries in the values given above.

2. The following pseudo-code snippets and questions relate to coding aspects of the Finite Element Method. Provide a short textual answer to each of them (do not write any code).

[5%] (a) Part of a loop for computing $\mathbf{K}^{(\mathrm{e})}$ can be seen below:

```
# ...

elmat = 0

for each integration point:
    B = compute_B_matrix (dN_dx)
    D = compute_D_matrix (youngs_modulus, poisson_ratio)
    thickness = get_thickness()

    elmat += transpose(B) * D * B * thickness

# ...
```

This code would not integrate a correct stiffness matrix. What is missing from it?

[5%] (b) Consider the following routine that implements the main loop of a FE solver:

```
# ...

K = 0
f = 0

for each element:
    K += element_stiffness  (element)
    f += element_bodyforces (element)

f += point_loads

solution = invert(K) * f

# ...
```

This code would not compute a valid solution. What step is missing?

[5%] 3. (a) With reference to the three-point Newton-Cotes integration rule (NC3), derive the missing information (?) in the table below.

	location ξ_i	weigth w_i
1	?	1/3
2	?	?
3	?	?

[5%] (b) Describe the procedure to perform numerical integration of a function in a generic one-dimensional domain.

[5%] (c) Integrate $I = \int_1^7 \frac{1}{x}\,\mathrm{d}x$ using NC3

[30%] 4. Consider the one-dimensional equation of heat conduction with unit conductivity:

$$\frac{\mathrm{d}^2\phi}{\mathrm{d}x^2} + Q = 0 \quad \text{for } 0 \leq x \leq L,$$

in which

$$Q = \begin{cases} 1 & 0 \leq x \leq L/2 \\ 0 & L/2 \leq x \leq L \end{cases}$$

and $\phi = 0$ at $x = 0$ and $x = L$.

(a) Derive the weak form at element level.

(b) Derive the discrete weak form at element level.

(c) Determine the value of ϕ at $x = L/2$ using a two-element discretization.

Note:

- To apply the boundary conditions $\phi = 0$ strike out corresponding rows and columns in the stiffness matrix. This should generate a 1×1 stiffness matrix.
- Consider a unit cross section.