Faculty of Civil Engineering and Geosciences
28 June 2023, 13:30–16:30
CIEM5110 Unit 2: Finite Elements for Structural Analysis

General instructions
All pages of your answer sheets must contain your name and your student number. Number each page of your answer sheets and indicate the total number of pages used. Write clearly and in English. Points for each question are in the left margin brackets.

This question sheet **must be returned** before leaving.

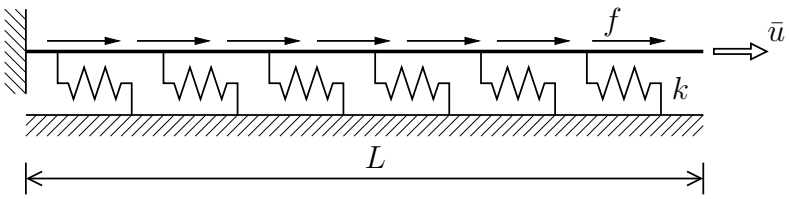
Questions

[40%] 1. Consider the 1D problem of an elastically supported bar, which can be described with the following equation:

$$-EA\frac{\mathrm{d}^2u}{\mathrm{d}x^2} + ku = f$$

where EA is the axial stiffness of the bar, k is the stiffness of the support and f is a distributed load. The following boundary conditions are applied:

$$u = 0 \quad \text{at} \quad x = 0$$
$$u = \bar{u} \quad \text{at} \quad x = L$$



(a) Construct the weak form for this problem.

(b) Derive an expression for the discrete system of equations for a general 1D finite element discretization.

(c) Discuss how the two boundary conditions are dealt with when solving this problem with the finite element method.

(d) In Gauss integration n integration points can integrate polynomials of the order $2n-1$ exactly. How many Gauss integration points are needed to integrate the stiffness matrix of the elastically supported bar problem exactly when 3-node quadratic elements are used?

[15%] 2. Suppose you are running a steady-state **nonlinear finite element simulation** in `pyJive` with a combination of `NonlinModule` and `SolidModel`, with boundary conditions being taken care of by other models which can be left out of consideration for answering this question. Assume the model does not contain body/source loads.

Under such scenario, the goal of `NonlinModule` is to compute the solution **a** that corresponds to a residual **r** whose norm is close enough to zero. The `SolidModel` is responsible for implementing the finite element relations specific for continuum mechanics problems.

Regarding the relative roles of `NonlinModule` and `SolidModel` and the **communication** that happens between them, answer the following questions:

(a) What does `NonlinModule` need to **send** and **receive** to/from `SolidModel` in order to solve the FE problem? For each of these quantities, specify if they are scalars, vectors or matrices and provide a short explanation for why they are needed by the receiving party. **TIP**: At least three items are needed for a complete answer.

(b) Suppose you are solving a problem with a two-dimensional stress equilibrium problem with a total of 1620 trianglular finite elements with one integration point each and a total of 1048 nodes. Give the size of all the variables you listed in (a).

[25%] 3. Consider the PDE for transient heat conduction:

$$\rho c \dot{u} + \nabla \cdot \mathbf{q} = f$$

with ρc being the specific heat capacity, u a temperature field, f a temperature source/sink term and $\mathbf{q}$ a heat flux given by:

$$\mathbf{q} = -\boldsymbol{\kappa} \nabla u$$

where $\boldsymbol{\kappa}$ is a conductivity matrix (linear). This PDE is solved for initial and boundary conditions given by:

$$u = g \text{ at } \Gamma_g, \quad u(t=0) = u_0, \quad -\mathbf{q} \cdot \mathbf{n} = h \text{ at } \Gamma_h$$

with $\mathbf{n}$ being the normal vector to the surface Γ_h

Assume you would like to solve transient heat problems with a semi-discrete FEM approach. This leads to a system of equations of the form:

$$\mathbf{M}\dot{\mathbf{a}} + \mathbf{K}\mathbf{a} = \mathbf{f}$$

with:

$$\mathbf{M} = \int_\Omega \mathbf{N}^\mathrm{T} \rho c \mathbf{N} \, \mathrm{d}\Omega \quad \mathbf{K} = \int_\Omega \mathbf{B}^\mathrm{T} \boldsymbol{\kappa} \mathbf{B} \, \mathrm{d}\Omega \quad \mathbf{f} = \int_\Omega \mathbf{N}^\mathrm{T} f \, \mathrm{d}\Omega + \int_{\Gamma_h} \mathbf{N}^\mathrm{T} h \, \mathrm{d}\Gamma$$

and a Backward Euler time stepper is adopted to solve the problem in time:

$$\mathbf{a}_{n+1} = \mathbf{a}_n + \Delta t \dot{\mathbf{a}}_{n+1}$$

Imagine your task is to implement this new simulation setup in `pyJive` for 2D/3D continuum problems. Given the following list of `Models` and `Modules`,

Model/module	Functionality
`BarModel`	Assembly of 1D equilibrium elements in extension
`FrameModel`	Assembly of Timoshenko elements with extension
`SolidModel`	Assembly of isoparametric equilibrium elements
`DirichletModel`	Handles Dirichlet-type boundary conditions
`NeumannModel`	Handles Neumann-type boundary conditions
`SolverModule`	Solves linear quasi-static FE systems
`NonlinModule`	Newton-Raphson solver in load/displacement control
`ExplicitTimeModule`	Solves dynamic systems with Central Difference
`NewmarkModule`	Solves linear dynamic systems with the Newmark time stepper
`ArclenModule`	Newton-Raphson solver with arc-length control
`LinBuckModule`	Computes buckling loads and modes
`ModeShapeModule`	Computes natural frequencies and modes
`NLNewmarkModule`	A Newmark time stepper with a Newton-Raphson loop

answer the following questions regarding the new implementations you would need to make:

(a) Given the Backward Euler expression and the semi-discrete system of equations above, and assuming $\mathbf{a}_n$ and $\mathbf{f}_{n+1}$ are known, derive a linear system of equations for time step $n+1$ of the form

$$\hat{\mathbf{K}}\mathbf{a}_{n+1} = \hat{\mathbf{f}}$$

and arrive at expressions for $\hat{\mathbf{K}}$ and $\hat{\mathbf{f}}$ as a function of $\mathbf{K}$, $\mathbf{M}$ and other known quantities.

(b) Pick **two** items from the list above that you believe would be best suitable as a **starting point** for your new implementations (*i.e.* existing code with related functionality). Provide a brief explanation for your choices and a short description of what you would need to change with respect to the existing code.

(c) Suppose you have successfully implemented code for transient heat conduction in `pyJive`. Apart from the newly-implemented code, which other models/modules from the table above should be used in combination with your new tools in order to set up the complete FE problem mentioned above? Provide a short motivation for your choices.

[20%] 4. Consider again the list of `pyJive` components from the previous question. For each of the modeling applications below, pick **one module** that would be appropriate for use in combination with `FrameModel`. Provide a short motivation for your choices.

(a) A two-story frame subjected to vertical loads for which you would like to determine at which load level static stability would be lost

(b) A roof frame, to determine the sensitivity of static stability to small geometric imperfections and/or load misalignments

(c) The deck of a pedestrian bridge, to determine whether typical pedestrian load signals would lead to high deformations through dynamic resonance

(d) The mechanical response of a frame under high-velocity impact loads

(e) A shallow shell modeled as a frame under vertical loads, for which the complete load-displacement path including several snap-throughs and snap-backs should be obtained