NSSC–II

Exercise 4

Finite Element Method

H. Pettermann, M. Schasching

Institute of Lightweight Design and Structural Biomechanics

Vienna University of Technology, Vienna, Austria

2022 05 12

0 General Information

- **Submission is due on Thursday, June 9th of 2022.** Please submit the material via TUWEL **by one of the group members**.
- The number of points corresponding to each task of the exercises is indicated within square brackets, e.g. [2 Points].

1 Introduction

To keep the FEM overhead and data handling to a minimum but to show some essential features of FEM, a straightened problem will be considered.

Complex pre-processing of meshes is avoided by working with a fixed mesh topology. However, the nodal coordinates will change for different tasks. Along the same line, a fixed set of Dirichlet

(essential) and Neumann (natural) boundary conditions is chosen. Isoparametric elements are NOT used to keep the implementation simple. Note, that in real life FEM analyses, all the above features would be used to allow for full exploitation of the advantages of FEM.

Direct and detailed software support cannot be given. According to the curriculum, the students have sufficient knowledge in programing. It is suggested to use Python, and exploit the features of the `numpy` and `matplotlib` packages. These tools have been employed by the lecturers in the development of this example. The problem is well suited for object oriented programming, however, a procedural approach is possible as well.

2 Problem Statement — Steady State Heat Conduction

An FEM program is to be developed and implemented to solve plane, steady state heat conduction problems. Guidelines, required equations, and hints are given below.

The example follows closely the text book by Zienkiewic et al. [1] which is available on-line for TU Wien students. Chapter 5 gives all required information and the derived equations as well as pre-computed variables.

Steady state heat conduction (see [1], Chpt. 5) is described by the partial differential equation,

$$-\frac{\partial}{\partial x_i} q_i + Q = 0 \quad , \tag{1}$$

together with the constitutive law,

$$q_i = -k_{ij} \frac{\partial}{\partial x_j} T \quad , \tag{2}$$

where q_i is the heat flux vector, Q the power density, T the temperature, and k_{ij} the conductivity tensor. As boundary conditions, only heat flux across the boundary and prescribed temperatures at the boundary, respectively, will be considered.

For plane heat conduction problems $i, j = 1, 2$, which implies $(\partial/\partial x_3)T = 0$ and $q_3 = 0$. Nevertheless, three-dimensional situations with a certain thickness are considered in terms of physics

(at least in the mind set of an engineer).

Isotropic material properties are treated, i.e. $k_{11} = k_{22}$ and $k_{12} = k_{21} = 0$, which are temperature independent but location (element) dependent.

It is recommended to consider physical units in the simulations (e.g. SI units) which allows for plausibility checks.

3 Finite Element Formulation

The Finite Elements to be implemented are linear triangles described in detail in [1], Sect. 5.1.3.1. The element "stiffness" matrix, $\mathbf{H^e}$, and the element "force" vector are derived in [1], Example 5.1. For $\mathbf{H^e}$ see page 125 (top) and set $k_{xx} = k_{yy}, k_{xy} = 0$ for isotropic material. The geometrical values for b_i and c_i as well as the area of the element, Δ, are given at page 120 and are illustrated in Fig. 5.1.

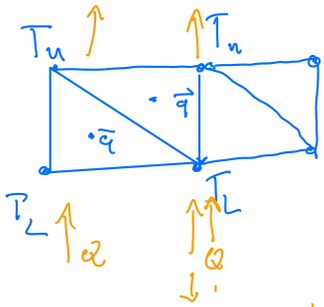
The "mass" matrix, $\mathbf{C^e}$, and the "force" vector, $\mathbf{s^e}$, are not needed for the present example.

Note that for this type of problem, each node has only one degree of freedom, i.e. the temperature. This is in contrast to mechanical problems with three degrees of freedom for each node, i.e. the displacements.

4 Single/Double Element Tests

Based on the procedures described in the next sections, it is strongly recommended to start with very simple cases for testing and verification, first. Choose configurations for which the results are known (and trivial).

- Model two triangles forming a square
- Prescribe all nodal temperatures
- Solve for the "reaction heat"
- Compare, check, assess, ...



- Model variations thereof by size, shape, bi-material, ...

5 Finite Element Model

5.1 Mesh Topology

For the "full scale" model use a mesh topology which is based on a regular grid of nodes as,

N91	N92	...	N100
..	..	..	..
N11	N12	...	N20
N1	N2	...	N10

element 1 formed by N1, N2, N11

with nodal coordinates N1(0,0), N10(L,0), N100(L,L).

Next step in building the FEM mesh is the definition of the element connectivities, i.e. giving for every element the nodes by which it is constituted. Corresponding to the equations in [1] (and as a widely applied convention in FEM), counterclockwise nodal assignment has to be used. Employ an element connectivity scheme where every element "En" is assigned a list of "its" three nodes as

E1(N1,N2,N11)	E2(N2,N12,N11)
E3(N2,N3,N12)	E4(N3,N13,N12)
...	
E17(N9,N10,N19)	E18(E10,E20,E19)
...	
...	E162(N90,N100,N99)

and fill the entire node grid following this scheme. The resulting mesh is shown in Fig. 1.

5.2 Total Stiffness Matrix

To obtain the desired system of equations, the total stiffness matrix, H_{ab}, has to be built up (*assembled*) from the element stiffness matrices. To do so, add the element stiffness contributions,

table / scheme → local element refers to the global

H^e_{ij}, to the corresponding H_{ab} elements following/respecting the "element connectivities".

Develop a data structure which allows for (algorithmically) easy access to the required values.

5.3 Boundary Conditions

A unique set of nodal boundary conditions (BCs) will be used throughout this tutorial example, as

- N91 – N100: Dirichlet BCs (depending on the "load case") $T = ...$
- N1 – N10: Neumann BCs (depending on the "load case") $P = ...$; note, that they follow from $q_y(x, y = 0) = ...$
- otherwise: Neumann BCs, $P = 0$

where P is the assembled load ("force") vector entry in the sense of a point-wise source of heat per time.

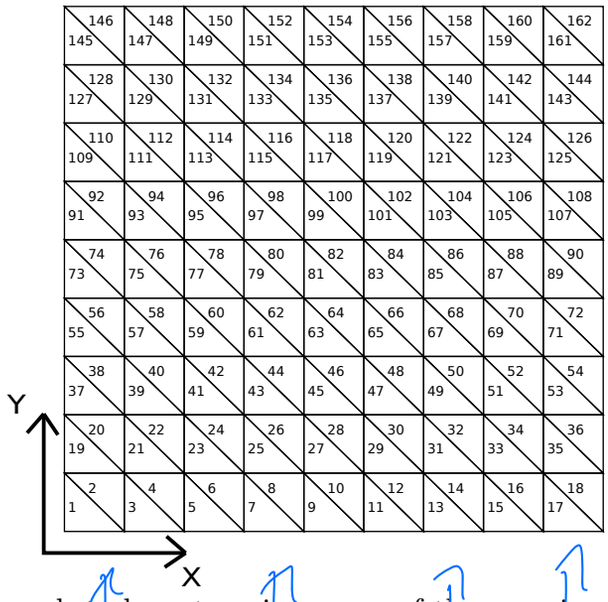


Figure 1: Regular mesh to be set up in course of the exercise and element numbers.

5.4 Solution

The total linear system of equations consists of sub-matrices and sub-vectors with known and unknown entries, respectively,

$$\begin{pmatrix} H_{1..90,1..90} & H_{91..100,1..90} \\ H_{1..90,91..100} & H_{91..100,91..100} \end{pmatrix} \begin{pmatrix} T_{1..90} \\ T_{91..100} \end{pmatrix} = \begin{pmatrix} P_{1..90} \\ P_{91..100} \end{pmatrix} , \tag{3}$$

where $(T_{1..90})$ and $(P_{91..100})$ are unknown, and $(T_{91..100})$ and $(P_{1..90})$ are given. The total stiffness matrix, $H_{ab} = H_{ba}$, is symmetric and known as it has been assembled before.

First, one can bring the known $(T_{91..100})$ to the right hand side, giving the sub-system,

$$(H_{1..90,1..90})\,(T_{1..90}) = (P_{1..90}) - (H_{91..100,1..90})\,(T_{91..100}) \quad , \tag{4}$$

which can be solved for $(T_{1..90})$ by some standard solver.

Then the remaining unknown "reaction forces", $(P_{91..100})$, can be computed.

Note, that the chosen set of BCs gives a system of equation which is ordered corresponding to known and unknown variables and, hence, allows for straightforward solution. In general cases, however, the system of equations has to reordered accordingly.

Finally, compute the local (element) temperature gradients as well as flux fields and store for each element. Because of the chosen element formulation the flux is constant within each element. Consequently, the values should be related to the geometrical center of the elements. Following [1], eqn. (5.18), with the gradient matrix derived at page 124 (bottom), gives the temperature gradient as,

$$\begin{pmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{pmatrix} = \frac{1}{2\Delta} \begin{pmatrix} b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} \quad , \tag{5}$$

with the indices 1,2,3 denoting the nodes of the element under consideration.

Subsequently, the heat flux follows form eqn. (2) of this document.

5.5 Post-Processing

Post-processing in the context of FEM, typically, includes the graphical presentation of the simulation results. In this example various variables are asked to be presented in form of suitable plots and graphs. Physical units are to be included. As tool, the use of `matplotlib` is suggested.

- Plot the temperature field $T(x, y)$ as contour plot, with the nodal temperatures as primary values and (bi-linear) interpolation between.
- Plot the temperature gradients and fluxes at the element centroids as vector plots (and, optionally, their components as contour plots without interpolation, i.e. constant for each element).
- Compare the fluxes in elements attached to the boundary $y = 0$ to the applied Neumann BC values (applied via nodal forces $P_{1..10}$).

For the regular square model, the following observations should be made.

- The temperature gradient must be constant in the entire model and equal to the overall gradient $\Delta T/\Delta y$.
- The flux must be constant in the entire model and must be equal to the applied flux. The latter must be equal to the sum of the applied nodal "forces" divided by the corresponding area.
- The temperature gradient and the flux, together with eqn. (2) must yield the input for the conductivity k

For irregular meshes and BCs the three latter observations will only hold approximatively.

For teaching reasons, the groups are asked to deliver numerical output values in a given file format. A Python function to do this is provided which has to be used. When other programming languages are used, the exact format specifier from the Python function has to be used to generate the required output.

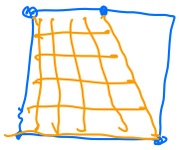
6 Variations

if the setup is correctly, then these variations must be straightforward

Once the above scheme is implemented and tested successfully, a set of model variations can be simulated to highlight some features of FEM. For each variation, named "V1" to "V4", the same post-processing steps as listed above are asked for. Note, that for some examples the approximative character of FEM will appear, however, mesh refinement would be a viable measure to reduce the error.

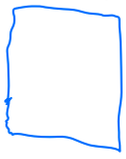
N100

V1



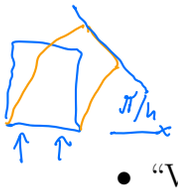
- Modify the coordinates
  - "V1": for trapezoidal shape of the model, set N100 to the coordinates $(\frac{L}{2}, L)$
  - "V2": to keep the quadratic shape but with a biased mesh, apply a modification of the x-coordinates,

mesh intensity changes



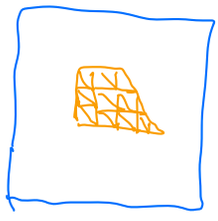
$$x^{\text{bias}} = x^{\text{regular}}[\frac{B}{L}x^{\text{regular}} - B + 1] \quad \text{with} \quad B = \frac{1}{2L}y \quad , \tag{6}$$

to keep the mesh at $y = 0$ regular (for easy application of nodal "forces") and for a gradual increase to $y = L$



  - "V3": for an annulus section of $\pi/4$ with center $(2L,0)$, outer radius $2L$ and inner radius L
- "V4": Modify in a set of elements in the center for $k^{\text{mod}} = ck$ as well as $k^{\text{mod}} = k/c$ (element numbers and the factor c will be specified)

inner element have different properties



7 Hints

Think about the data you need and an appropriate data structure (many ways are possible and feasible, there is no "unique best" way).

Even if a plane problem is treated, physics takes place in 3D; just the temperature gradient in out of plane direction is zero. All physical units are in "3D" they can (and should) be used for plausibility checks.

In Python the direct equation solver `numpy.linalg.solve(A,b)` is suitable for the present example.

For the Neumann BCs, the flux has to be converted to corresponding nodal "forces", P. First, do the considerations element wise (along the loaded edge). Then, sum up for nodes which belong to two elements.

8 Group Specific Input Values

Input variables for each group will be provided by an ASCII-text file. This contains

- geometry, L
- element thickness, h_z, see [1]
- thermal conductivity, k
- domain with modified properties, i.e. corresponding element numbers and factor c
- temperature BCs
- flux BCs

9 Group Specific Output and Points

A python function `print_HTP(H, T, P)` will be provided to create a txt-file containing formatted output for the determined overall assembled stiffness matrix H, the nodal temperature vector T and nodal force vector P. Make sure, that your system of equations is sorted by ascending node numbers, i.e., N1 N2 ... N100. The output-files for the basic case and the variations specified in Section 6 are asked for. If you are using a different programming language, please check pythons format function for how to prepare your output.
Furthermore, the results are to be plotted (see Section 5.5) and summarized in a short report in PDF-format, including some physical interpretation (do not go to much into detail).

Checklist for submission:

just one code, all the modifications are just inputchanges

- Group number.
- FEM-Code incl. README file to explain the content and the usage of the different files **in a single .zip-file**. [4 Points]
- Output-files for each variation - Basis, V1, V2, V3, V4a, V4b (see Section 6) **in a single .zip-file**. [1 Point each, output + part of report]
- Report **as a .pdf-file** - Containing post-processing results as plots and physical interpretation.

References

[1] O. Zienkiewicz, R. Taylor, and J. Zhu, "Chapter 5 - field problems: A multidimensional finite element method," in *The Finite Element Method: its Basis and Fundamentals (Seventh Edition)* (O. Zienkiewicz, R. Taylor, and J. Zhu, eds.), pp. 115 – 149, Oxford: Butterworth-Heinemann, seventh edition ed., 2013.