3 Basic 2D and 3D finite element methods
Contents

1. Modelling principles and boundary value problems in engineering sciences
2. Energy methods and basic 1D finite element methods
   - bars/rods, beams, heat diffusion, seepage, electrostatics
3. Basic 2D and 3D finite element methods
   - heat diffusion, seepage
4. Numerical implementation techniques of finite element methods
5. Abstract formulation and accuracy of finite element methods
6. Finite element methods for Euler–Bernoulli beams
7. Finite element methods for Timoshenko beams
10. Finite element methods for 2D and 3D elasticity
11. Extra lecture: other finite element applications in structural engineering
3 Basic 2D and 3D finite element methods

Contents

1. 2D weak form (based on the principle of virtual work)
2. 2D finite element method (based on the weak form)

Learning outcome

A. Understanding of the main principles behind the 2D finite element method
B. Ability to formulate and apply the finite element method for 2D model problems

References

Lecture notes: chapters 4.1–6, 5.1–4
Text book: chapters 2.1–4
3.0 Generalizing 1D finite element methods to 2D or 3D intuitively

How straightforward it might be to generalize 1D finite element methods for 2D or 3D problems?
### 3.1 2D weak form – model problem

**Stationary isotropic heat diffusion (conduction) problem in 2D:** Let us consider heat diffusion in *isotropic* material. *Fourier law* builds a constitutive relation between the *heat flux* \( q \) and the *temperature* \( T \) through the *thermal conductivity* \( k \) as

\[
\begin{pmatrix}
q_x(x, y) \\
q_y(x, y)
\end{pmatrix}
= q(x, y) = q = -k \nabla T = -k(x, y) \nabla T(x, y) = -k(x, y)
\frac{\partial T(x, y)}{\partial x}
\frac{\partial T(x, y)}{\partial y}
\]
Stationary isotropic heat diffusion (conduction) problem in 2D: Let us consider heat diffusion in isotropic material. Fourier law builds a constitutive relation between the heat flux $q$ and the temperature $T$ through the thermal conductivity $k$ as

$$\begin{pmatrix} q_x(x, y) \\ q_y(x, y) \end{pmatrix} = q(x, y) = q = -k \nabla T = -k(x, y) \nabla T(x, y) = -k(x, y) \begin{pmatrix} \frac{\partial}{\partial x} T(x, y) \\ \frac{\partial}{\partial y} T(x, y) \end{pmatrix}$$

The first law of thermodynamics, or the principle of conservation of energy, combined with the stationary state assumption, implies the following diffusion equation describing the problem:

$$\frac{\partial q_x(x, y)}{\partial x} + \frac{\partial q_y(x, y)}{\partial y} = \nabla \cdot q(x, y) = \nabla \cdot q = f = f(x, y), \quad (x, y) \in \Omega \subset \mathbb{R}^2$$

$$\Rightarrow -\nabla \cdot (k(x, y) \nabla T(x, y)) = f(x, y), \quad (x, y) \in \Omega; \quad \text{Notation:} \quad \text{div } q = \nabla \cdot q$$
3.1 2D weak form – model problem

Euler equations, or the strong form, i.e., the partial differential equation and the boundary conditions, of the problem is the following:

(1) \(-\nabla \cdot (k(x, y)\nabla T(x, y)) = f(x, y), \ (x, y) \in \Omega\)

(2a) \(T(x, y) = T_0(x, y), \ (x, y) \in \Gamma_T\)

(2b) \(q(x, y) \cdot n = q_0(x, y), \ (x, y) \in \Gamma_q\)
Euler equations, or the strong form, i.e., the partial differential equation and the boundary conditions, of the problem is the following:

(1) \[-\nabla \cdot (k(x, y)\nabla T(x, y)) = f(x, y), \quad (x, y) \in \Omega\]
(2a) \[T(x, y) = T_0(x, y), \quad (x, y) \in \Gamma_T\]
(2b) \[q(x, y) \cdot n = q_0(x, y), \quad (x, y) \in \Gamma_q\]

Here:
- \(T\) temperature (unknown function)
- \(k\) thermal conductivity (given material data)
- \(f\) heat supply (given loading data), \(\Omega\) domain (given geometrical data)
- \(\Gamma_T\) boundary part for given temperature (given boundary data)
- \(\Gamma_q\) boundary part for given heat flux (given boundary data)
- \(T_0\) temperature on the boundary (given essential, Dirichlet, boundary data)
- \(q_0\) heat flux on the boundary (given natural, Neumann, boundary data).
1. Multiply the differential equation (1) by a (smooth) test function (specified later):

\[- \nabla \cdot (k \nabla T) = f \ \Rightarrow \ - \nabla \cdot (k \nabla T) \hat{T} = f \hat{T}, \ \text{in} \ \Omega \subset \mathbb{R}^2 = R \times R\]
3.1 2D weak form – model problem

1. Multiply the differential equation (1) by a (smooth) test function (specified later):

\[- \nabla \cdot (k \nabla T) = f \quad \Rightarrow \quad - \nabla \cdot (k \nabla T) \hat{T} = f \hat{T}, \quad \text{in} \quad \Omega \subset R^2 = R \times R\]

2. Integrate over the domain:

\[\Rightarrow - \int_{\Omega} \nabla \cdot (k \nabla T) \hat{T} \, dA = \int_{\Omega} f \hat{T} \, dA, \quad dA = dx \, dy\]
3.1 2D weak form – model problem

1. Multiply the differential equation (1) by a (smooth) test function (specified later):

\[-\nabla \cdot (k \nabla T) = f \Rightarrow -\nabla \cdot (k \nabla T) \hat{T} = f \hat{T}, \quad \text{in } \Omega \subset R^2 = R \times R\]

2. Integrate over the domain:

\[\Rightarrow -\int_{\Omega} \nabla \cdot (k \nabla T) \hat{T} \, dA = \int_{\Omega} f \hat{T} \, dA, \quad dA = dx \, dy\]

3. Integrate by parts (the left hand side):

\[\Rightarrow -\int_{\partial \Omega} n \cdot (k \nabla T) \hat{T} \, ds + \int_{\Omega} (k \nabla T) \cdot \nabla \hat{T} \, dA = \int_{\Omega} f \hat{T} \, dA\]
3.1 2D weak form – model problem

1. Multiply the differential equation (1) by a (smooth) test function (specified later):

\[- \nabla \cdot (k \nabla T) = f \quad \Rightarrow \quad - \nabla \cdot (k \nabla T) \hat{T} = f \hat{T}, \quad \text{in } \Omega \subset R^2 = R \times R\]

2. Integrate over the domain:

\[\Rightarrow - \int_{\Omega} \nabla \cdot (k \nabla T) \hat{T} \, dA = \int_{\Omega} f \hat{T} \, dA, \quad dA = dx \, dy\]

3. Integrate by parts (the left hand side):

\[\Rightarrow - \int_{\partial \Omega} \underline{n} \cdot (k \nabla T) \hat{T} \, ds + \int_{\Omega} (k \nabla T) \cdot \nabla \hat{T} \, dA = \int_{\Omega} f \hat{T} \, dA\]

4. Utilize the natural boundary condition (2b):

\[\underline{q}(x, y) \cdot \underline{n} = q_0(x, y), \quad (x, y) \in \Gamma_q\]
3.1 2D weak form – model problem

1. Multiply the differential equation (1) by a (smooth) test function (specified later):

\[- \nabla \cdot (k \nabla T) = f \quad \Rightarrow \quad - \nabla \cdot (k \nabla T) \hat{T} = f \hat{T}, \quad \text{in } \Omega \subset R^2 = R \times R\]

2. Integrate over the domain:

\[\Rightarrow - \int_{\Omega} \nabla \cdot (k \nabla T) \hat{T} \, dA = \int_{\Omega} f \hat{T} \, dA, \quad dA = dx \, dy\]

3. Integrate by parts (the left hand side):

\[\Rightarrow - \int_{\partial \Omega} n \cdot (k \nabla T) \hat{T} \, ds + \int_{\Omega} (k \nabla T) \cdot \nabla \hat{T} \, dA = \int_{\Omega} f \hat{T} \, dA\]

4. Utilize the natural boundary condition (2b): \[\mathbf{q}(x, y) \cdot \mathbf{n} = q_0(x, y), \quad (x, y) \in \Gamma_q\]

5. Set a zero essential boundary condition (2a) for the test function: \[\hat{T} = 0 \quad \text{on } \Gamma_T\]

\[\Rightarrow \int_{\Omega} (k \nabla T) \nabla \hat{T} \, dA = \int_{\Omega} f \hat{T} \, dA - \int_{\Gamma_q} q_0 \hat{T} \, ds\]
3.1 2D weak form – model problem

**Weak form.** Find \( T = T(x, y) \) such that it satisfies \( T|_{\Gamma} = T_0, \int_{\Omega} |\nabla T|^2 \, dA < \infty \),

\[
\int_{\Omega} (k \nabla T) \cdot \nabla \hat{T} \, dA = \int_{\Omega} f \hat{T} \, dA - \int_{\Gamma_q} q_0 \hat{T} \, ds
\]

for all \( \hat{T} = \hat{T}(x, y) \) satisfying \( \hat{T}|_{\Gamma} = 0, \int_{\Omega} |\nabla \hat{T}|^2 \, dA < \infty \).
3.1 2D weak form – model problem

Weak form. Find $T = T(x, y)$ such that it satisfies \( T|_{\Gamma_T} = T_0 \), \( \int_\Omega |\nabla T|^2 dA < \infty \),
\[
\int_\Omega (k \nabla T) \cdot \nabla \hat{T} \, dA = \int_\Omega f \hat{T} \, dA - \int_{\Gamma_q} q_0 \hat{T} \, ds
\]
for all $\hat{T} = \hat{T}(x, y)$ satisfying $\hat{T}|_{\Gamma_T} = 0$, \( \int_\Omega |\nabla \hat{T}|^2 \, dA < \infty \).

Remark. Note that the solution and the test function, respectively, have to satisfy the boundary conditions \( T|_{\Gamma_T} = T_0 \), \( \hat{T}|_{\Gamma_T} = 0 \) and the regularity conditions
\[
\int_\Omega |\nabla T|^2 dA < \infty, \quad \int_\Omega |\nabla \hat{T}|^2 dA < \infty.
\]

Then they are called kinematically admissible.
3.1 2D weak form – model problem

**Weak form.** Find \( T = T(x, y) \) such that it satisfies \( T_{|\Gamma_T} = T_0, \int_{\Omega} |\nabla T|^2 dA < \infty, \)
\[
\int_{\Omega} (k\nabla T) \cdot \nabla \hat{T} \, dA = \int_{\Omega} f \hat{T} \, dA - \int_{\Gamma_q} q_0 \hat{T} \, ds
\]
for all \( \hat{T} = \hat{T}(x, y) \) satisfying \( \hat{T}_{|\Gamma_T} = 0, \int_{\Omega} |\nabla \hat{T}|^2 \, dA < \infty. \)

**Remark.** Note that the solution and the test function, respectively, have to satisfy the boundary conditions \( T_{|\Gamma_T} = T_0, \hat{T}_{|\Gamma_T} = 0 \) and the **regularity conditions**
\[
\int_{\Omega} |\nabla T|^2 dA < \infty, \int_{\Omega} |\nabla \hat{T}|^2 dA < \infty.
\]

Then they are called **kinematically admissible.**

**Remark.** Starting from the weak form we could correspondingly derive the strong form (integrating by parts ”backwards”).
Abstract weak form. Find $T \in S$ such that

$$a(T, \hat{T}) = l(\hat{T}), \quad \forall \hat{T} \in V,$$

with the definitions

$$a(T, \hat{T}) := \int_{\Omega} (k \nabla T) \cdot \nabla \hat{T} \, dA,$$

$$l(\hat{T}) := \int_{\Omega} f \hat{T} \, dA - \int_{\Gamma_q} q_0 \hat{T} \, ds.$$

where $a : S \times V \rightarrow \mathbb{R}$ is a bilinear form, $l : V \rightarrow \mathbb{R}$ is a functional.
Abstract weak form. Find $T \in S$ such that

$$a(T, \hat{T}) = l(\hat{T}) \quad \forall \hat{T} \in V,$$

with the definitions

$$a(T, \hat{T}) := \int_{\Omega} (k\nabla T) \cdot \nabla \hat{T} \ dA,$$

$$l(\hat{T}) := \int_{\Omega} f \hat{T} \ dA - \int_{\Gamma_q} q_0 \hat{T} \ ds.$$

where $a : S \times V \rightarrow \mathbb{R}$ is a bilinear form, $l : V \rightarrow \mathbb{R}$ is a functional and $S, V$ are the function spaces for the trial and test functions, respectively.

$$S := \left\{ T \mid T|_{\Gamma_T} = T_0, \int_{\Omega} |\nabla T|^2 \ dA < \infty \right\},$$

$$V := \left\{ \hat{T} \mid \hat{T}|_{\Gamma_T} = 0, \int_{\Omega} |\nabla \hat{T}|^2 \ dA < \infty \right\}.$$
3.1 2D weak form – 2D and 3D generalizations

3D heat diffusion:

\[
\int_{\Omega} (k \nabla T) \cdot \nabla \hat{T} \, dA = \int_{\Omega} f \, \hat{T} \, dA \quad \rightarrow \quad \int_{\Omega} (k \nabla T) \cdot \nabla \hat{T} \, dV = \int_{\Omega} f \, \hat{T} \, dV
\]

\[T = T(x, y), \quad \hat{T} = \hat{T}(x, y)\]

\[T = T(x, y, z), \quad \hat{T} = \hat{T}(x, y, z)\]

\[
\nabla T = \begin{pmatrix}
\frac{\partial T}{\partial x} \\
\frac{\partial T}{\partial y}
\end{pmatrix}
\]

\[
\nabla T = \begin{pmatrix}
\frac{\partial T}{\partial x} \\
\frac{\partial T}{\partial y} \\
\frac{\partial T}{\partial z}
\end{pmatrix}
\]
3.1 2D weak form – 2D and 3D generalizations

3D heat diffusion:

\[ \int_{\Omega} (k \nabla T) \cdot \nabla \hat{T} \, dA = \int_{\Omega} f \, \hat{T} \, dA \quad \rightarrow \quad \int_{\Omega} (k \nabla T) \cdot \nabla \hat{T} \, dV = \int_{\Omega} f \, \hat{T} \, dV \]

\[ T = T(x, y), \quad \hat{T} = \hat{T}(x, y) \quad \rightarrow \quad T = T(x, y, z), \quad \hat{T} = \hat{T}(x, y, z) \]

\[ \nabla T = \begin{pmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{pmatrix} \quad \rightarrow \quad \nabla T = \begin{pmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{pmatrix} \]

2D or 3D seepage (flow of water through soils):

\[ \int_{\Omega} (k \nabla p) \cdot \nabla \hat{p} \, d\Omega = \int_{\Omega} f \, \hat{p} \, d\Omega, \]

2D: \[ p = p(x, y), \quad \hat{p} = \hat{p}(x,y), \quad \nabla = \nabla_{xy}, \quad d\Omega = dx\,dy \]

3D: \[ p = p(x,y,z), \quad \hat{p} = \hat{p}(x,y,z), \quad \nabla = \nabla_{xyz}, \quad d\Omega = dx\,dy\,dz \]
3.1 2D weak form – 2D and 3D generalizations

**Unisotropic conductivity:** Let us consider heat diffusion (or seepage) in *unisotropic* material. *Fourier law* (*Darcy law*) builds a constitutive relation between the *heat flux* \( q \) (*velocity* \( v \)) and the *temperature* \( T \) (*pressure* \( p \)) through the *thermal* (*hydraulic*) conductivity (or permeability) matrix \( k \) as

\[
\begin{pmatrix}
q_x(x, y) \\
q_y(x, y)
\end{pmatrix}
= q(x, y) = q = -k \nabla T
= -\begin{bmatrix}
k_{11}(x, y) & k_{12}(x, y) \\
k_{21}(x, y) & k_{22}(x, y)
\end{bmatrix}
\begin{pmatrix}
\frac{\partial}{\partial x} T(x, y) \\
\frac{\partial}{\partial y} T(x, y)
\end{pmatrix}
\]
3.1 2D weak form – 2D and 3D generalizations

Unisotropic conductivity: Let us consider heat diffusion (or seepage) in unisotropic material. Fourier law (Darcy law) builds a constitutive relation between the heat flux $q$ (velocity $v$) and the temperature $T$ (pressure $p$) through the thermal (hydraulic) conductivity (or permeability) matrix $k$ as

$$
\begin{pmatrix}
q_x(x, y) \\
q_y(x, y)
\end{pmatrix}
= q(x, y) = q = -k \nabla T
= -\begin{bmatrix}
k_{11}(x, y) & k_{12}(x, y) \\
k_{21}(x, y) & k_{22}(x, y)
\end{bmatrix}
\begin{pmatrix}
\frac{\partial T(x, y)}{\partial x} \\
\frac{\partial T(x, y)}{\partial y}
\end{pmatrix}
$$

which can be generalized to 3D with the conductivity matrix

$$
k = \begin{bmatrix}
k_{11}(x, y, z) & k_{12}(x, y, z) & k_{13}(x, y, z) \\
k_{21}(x, y, z) & k_{22}(x, y, z) & k_{23}(x, y, z) \\
k_{31}(x, y, z) & k_{31}(x, y, z) & k_{33}(x, y, z)
\end{bmatrix}.
$$
3.1 2D weak form – 2D and 3D generalizations

**Unisotropic conductivity**

Let us consider heat diffusion (or seepage) in unisotropic material. *Fourier law (Darcy law)* builds a constitutive relation between the heat flux $\mathbf{q}$ (velocity $\mathbf{v}$) and the temperature $T$ (pressure $p$) through the thermal (hydraulic) conductivity (or permeability) matrix $k$ as

$$
\begin{pmatrix}
q_x(x, y) \\
q_y(x, y)
\end{pmatrix} = \mathbf{q}(x, y) = \mathbf{q} = -k \nabla T = - \begin{bmatrix}
k_{11}(x, y) & k_{12}(x, y) \\
k_{21}(x, y) & k_{22}(x, y)
\end{bmatrix}
\begin{pmatrix}
\frac{\partial}{\partial x} T(x, y) \\
\frac{\partial}{\partial y} T(x, y)
\end{pmatrix}
$$

which can be generalized to 3D with the conductivity matrix

$$
k = \begin{bmatrix}
k_{11}(x, y, z) & k_{12}(x, y, z) & k_{13}(x, y, z) \\
k_{21}(x, y, z) & k_{22}(x, y, z) & k_{23}(x, y, z) \\
k_{31}(x, y, z) & k_{32}(x, y, z) & k_{33}(x, y, z)
\end{bmatrix}.
$$

**Remark.** For *homogeneous* materials, $k_{ij}$‘s are constants.
3.1 2D weak form

(i) For function \( v(x) = x \), defined on interval \( \Omega = (0,1) \subset \mathbb{R} \), calculate

\[
\int_{\Omega} v^2 \, dx \quad \text{and} \quad \int_{\Omega} (v')^2 \, dx
\]

(ii) For function \( v(x,y) = x+y \), defined on domain \( \Omega = (0,1) \times (0,1) \subset \mathbb{R}^2 \), calculate

\[
\int_{\Omega} v^2 \, dA \quad \text{and} \quad \int_{\Omega} |\nabla v|^2 \, dA
\]
3.2 2D finite element method – model problem

1. Divide the solution area (domain) into \( n \) subdomains, elements \( e_i \) (triangles, quadrangles, ...) with nodes \( x_j = (x_j, y_j) \) and element size \( h_i = \text{diam}(e_i) \):

\[
\begin{align*}
x_1 & \quad \ldots \quad x_j & \quad \ldots & \quad x_m \\
\end{align*}
\]
3.2 2D finite element method – model problem

1. Divide the solution area (domain) into \( n \) subdomains, elements \( e_i \) (triangles, quadrangles, ...) with nodes \( x_j = (x_j, y_j) \) and element size \( h_i = \text{diam}(e_i) \):

\[
\begin{align*}
\text{x}_1 & \quad \text{x}_j & \quad \text{x}_m \\
\text{x}_1 & \quad \text{x}_j & \quad \text{x}_m \\
\text{x}_1 & \quad \text{x}_j & \quad \text{x}_m
\end{align*}
\]
3.2 2D finite element method – model problem

1. Divide the solution area (domain) into $n$ subdomains, elements $e_i$ (triangles, quadrangles, ...) with nodes $x_j = (x_j, y_j)$ and element size $h_i = \text{diam}(e_i)$:

![Diagram of divided solution area with subdomains and nodes]

2. Choose a trial function for the finite element approximation as a sum

$$T_h(x) = \phi_0(x)d_0 + \phi_1(x)d_1 + \cdots + \phi_m(x)d_m = \sum_{j=0}^{m} \phi_j(x)d_j$$
3.2 2D finite element method – model problem

1. Divide the solution area (domain) into $n$ subdomains, elements $e_i$ (triangles, quadrangles, …) with nodes $x_j = (x_j, y_j)$ and element size $h_i = \text{diam}(e_i)$:

2. Choose a trial function for the finite element approximation as a sum

$$T_h(x) = \phi_0(x)d_0 + \phi_1(x)d_1 + \cdots + \phi_m(x)d_m = \sum_{j=0}^{m} \phi_j(x)d_j$$

with suitable local basis functions $\phi_i$ of some polynomial order (now linear or bilinear):
3.2 2D finite element method – model problem

1. Divide the solution area (domain) into $n$ subdomains, elements $e_i$ (triangles, quadrangles, …) with nodes $x_j = (x_j, y_j)$ and element size $h_i = \text{diam}(e_i)$:

2. Choose a trial function for the finite element approximation as a sum

$$T_h(x) = \phi_0(x)d_0 + \phi_1(x)d_1 + \cdots + \phi_m(x)d_m = \sum_{j=0}^{m} \phi_j(x)d_j$$

with suitable local basis functions $\phi_i$ of some polynomial order (now linear or bilinear):
3.2 2D finite element method – model problem

1. Divide the solution area (domain) into \( n \) subdomains, elements \( e_i \) (triangles, quadrangles, …) with nodes \( x_j = (x_j, y_j) \) and element size \( h_i = \text{diam}(e_i) \):

![Diagram showing division of solution area into subdomains]

2. Choose a trial function for the finite element approximation as a sum

\[
T_h(x) = \phi_0(x)d_0 + \phi_1(x)d_1 + \cdots + \phi_m(x)d_m = \sum_{j=0}^{m} \phi_j(x)d_j
\]

with suitable local basis functions \( \phi_i \) of some polynomial order (now linear or bilinear):

![Diagram showing trial functions and basis functions]

The unknown scalar values \( d_i = T_h(x_i) \) are called the degrees of freedom.
Ensure that the trial function satisfies the **essential boundary conditions**:

\[ T_0(x_j) = T_h(x_j) = \sum_{j=0}^{m} \phi_j(x_j)d_j = d_j \quad \forall x_j = (x_j, y_j) \in \Gamma_T \quad (\text{say, } p \ j's) \]
Ensure that the trial function satisfies the essential boundary conditions:

\[ T_0(x_j) = T_h(x_j) = \sum_{j=0}^{m} \phi_j(x_j) d_j = d_j \quad \forall x_j = (x_j, y_j) \in \Gamma_T \quad \text{(say, } p \ j' \text{'s)} \]

3. Choose a test function of a similar form (Galerkin method) with the corresponding (but zero) condition:

\[
\hat{T}_h(x) = \phi_0(x)c_0 + \phi_1(x)c_1 + \cdots + \phi_m(x)c_m = \sum_{j=0}^{m} \phi_j(x)c_j \\
0 = \hat{T}_h(x_j) = \sum_{j=0}^{m} \phi_j(x_j)c_j = c_j \quad \forall x_j \in \Gamma_T
\]
3.2 2D finite element method – model problem

Ensure that the trial function satisfies the essential boundary conditions:

\[ T_0(x_j) = T_h(x_j) = \sum_{j=0}^{m} \phi_j(x_j)d_j = d_j \quad \forall x_j = (x_j, y_j) \in \Gamma_T \quad \text{(say, } p \text{ } j's) \]

3. Choose a test function of a similar form (Galerkin method) with the corresponding (but zero) condition:

\[ \hat{T}_h(x) = \phi_0(x)c_0 + \phi_1(x)c_1 + \cdots + \phi_m(x)c_m = \sum_{j=0}^{m} \phi_j(x)c_j \]

\[ 0 = \hat{T}_h(x_j) = \sum_{j=0}^{m} \phi_j(x_j)c_j = c_j \quad \forall x_j \in \Gamma_T \]

4. Insert the functions – trial and test – into the weak form:

\[ \int_{\Omega} (k\nabla T) \cdot \nabla \hat{T} \, d\Omega = \int_{\Omega} f \hat{T} \, d\Omega - \int_{\Gamma_q} q_0 \hat{T} \, ds \]
3.2 2D finite element method – model problem

Ensure that the trial function satisfies the essential boundary conditions:

\[ T_0(x_j) = T_h(x_j) = \sum_{j=0}^{m} \phi_j(x_j)d_j = d_j \quad \forall x_j = (x_j, y_j) \in \Gamma_T \quad (\text{say, } p \ j's) \]

3. Choose a test function of a similar form (Galerkin method) with the corresponding (but zero) condition:

\[ \hat{T}_h(x) = \phi_0(x)c_0 + \phi_1(x)c_1 + \cdots + \phi_m(x)c_m = \sum_{j=0}^{m} \phi_j(x)c_j \]

\[ 0 = \hat{T}_h(x_j) = \sum_{j=0}^{m} \phi_j(x_j)c_j = c_j \quad \forall x_j \in \Gamma_T \]

4. Insert the functions – trial and test – into the weak form:

\[
\int_{\Omega} (k \nabla T) \cdot \nabla \hat{T} \, d\Omega = \int_{\Omega} f \hat{T} \, d\Omega - \int_{\Gamma_q} q_0 \hat{T} \, ds \quad \Rightarrow \\
\int_{\Omega} (k \sum_{j=0}^{m} \nabla \phi_j d_j) \cdot \sum_{i=0}^{m} \nabla \phi_i c_i \, d\Omega = \int_{\Omega} f \sum_{i=0}^{m} \phi_i c_i \, d\Omega - \int_{\Gamma_q} q_0 \sum_{i=0}^{m} \phi_i c_i \, ds
\]
This results in a simple equation system

\[ K \mathbf{d} = \mathbf{f} \]

with the \textit{stiffness matrix} (computable for \( i,j = 1, \ldots, m-p \)), \textit{force vector} (computable for \( i = 1, \ldots, m-p \)) and the \textit{displacement vector} (unknown for \( i = 1, \ldots, m-p \)):

\[
K = \begin{bmatrix} K_{ij} \end{bmatrix}, \quad K_{ij} = \int_{\Omega} (k\nabla \phi_j) \cdot \nabla \phi_i \, d\Omega,
\]

\[
f = \begin{bmatrix} f_i \end{bmatrix}, \quad f_i = \int_{\Omega} f_\phi \, d\Omega - \int_{\Gamma_q} q_0 \phi_i \, ds - \sum_{x_j \in \Gamma_T} \int_{\Omega} (k\nabla \phi_j T_0(x_j)) \cdot \nabla \phi_i \, d\Omega,
\]

\[
\mathbf{d} = \begin{bmatrix} d_j \end{bmatrix}.
\]
3.2 2D finite element method – model problem

This results in a simple equation system

\[ Kd = f \]

with the stiffness matrix (computable for \( i,j = 1, \ldots, m-p \)), force vector (computable for \( i = 1, \ldots, m-p \)) and the displacement vector (unknown for \( i = 1, \ldots, m-p \)):

\[ K = \begin{bmatrix} K_{ij} \end{bmatrix}, \quad K_{ij} = \int_{\Omega} (k \nabla \phi_j) \cdot \nabla \phi_i \, d\Omega, \]

\[ f = \begin{bmatrix} f_i \end{bmatrix}, \quad f_i = \int_{\Omega} f \phi_i \, d\Omega - \int_{\Gamma_q} q_0 \phi_i \, ds - \sum_{x_j \in \Gamma_T} \int_{\Omega} (k \nabla \phi_j \ T_0(x_j)) \cdot \nabla \phi_i \, d\Omega, \]

\[ d = \begin{bmatrix} d_j \end{bmatrix}. \]

details on blackboard or exercises

✓ the general case
✓ examples with few elements
This results in a simple equation system

\[ K d = f \]

with the **stiffness matrix** (computable for \( i,j = 1, \ldots, m-p \)), **force vector** (computable for \( i = 1, \ldots, m-p \)) and the **displacement vector** (unknown for \( i = 1, \ldots, m-p \)):

\[
K = \begin{bmatrix} K_{ij} \end{bmatrix}, \quad K_{ij} = \int_{\Omega} (k\nabla \phi_j) \cdot \nabla \phi_i \, d\Omega,
\]

\[
f = \begin{bmatrix} f_i \end{bmatrix}, \quad f_i = \int_{\Omega} f \phi_i \, d\Omega - \int_{\Gamma_q} q_0 \phi_i \, ds - \sum_{x_j \in \Gamma_T} \int_{\Omega} (k\nabla \phi_j \cdot T_0(x_j)) \cdot \nabla \phi_i \, d\Omega,
\]

\[
d = \begin{bmatrix} d_j \end{bmatrix}.
\]

details on blackboard or exercises

✓ the general case
✓ examples with few elements
✓ implications of quadratic basis functions
This results in a simple equation system

\[ K \mathbf{d} = f \]

with the **stiffness matrix** (computable for \( i,j = 1, \ldots, m-p \)), **force vector** (computable for \( i = 1, \ldots, m-p \)) and the **displacement vector** (unknown for \( i = 1, \ldots, m-p \)):

\[ K = [K_{ij}], \quad K_{ij} = \int_{\Omega} (k\nabla \phi_j) \cdot \nabla \phi_i \, d\Omega, \]

\[ f = [f_i], \quad f_i = \int_{\Omega} f \phi_i \, d\Omega - \int_{\Gamma_q} q_0 \phi_i \, ds - \sum_{x_j \in \Gamma_T} T_0(x_j) \int_{\Omega} (k\nabla \phi_j) \cdot \nabla \phi_i \, d\Omega, \]

\[ \mathbf{d} = [d_j]. \]

**Remark.** The stiffness matrix is (very often) **symmetric** (due to derivative orders) and its entries are concentrated in a narrow diagonal band forming a **band matrix** (due to local trial and test functions). These features can be utilized in computer implelementation – implying small amounts of **memory needs** and quick **processing**.
This results in a simple equation system

\[ K \mathbf{d} = \mathbf{f} \]

with the stiffness matrix (computable for \( i,j = 1, \ldots, m-p \)), force vector (computable for \( i = 1, \ldots, m-p \)) and the displacement vector (unknown for \( i = 1, \ldots, m-p \)):

\[ K = [K_{ij}], \quad K_{ij} = \int_\Omega (k \nabla \phi_j) \cdot \nabla \phi_i \, d\Omega, \]

\[ \mathbf{f} = [f_i], \quad f_i = \int_\Omega f \phi_i \, d\Omega - \int_{\Gamma_q} \mathbf{q}_0 \phi_i \, ds - \sum_{\mathbf{x}_j \in \Gamma_T} T_0(\mathbf{x}_j) K_{ij}, \]

\[ \mathbf{d} = [d_j]. \]

**Remark.** The stiffness matrix is (very often) symmetric (due to derivative orders) and its entries are concentrated in a narrow diagonal band forming a band matrix (due to local trial and test functions). These features can be utilized in computer implementation – implying small amounts of memory needs and quick processing.

**Remark.** Test and trial functions have to be (only) once differentiable (and will be then integrated over the domain) and (only) evaluable on the boundary.
Remark. The stiffness matrix and force vector above can be also written by using the abstract formalism introduced earlier:

\[
K_{ij} = \int_{\Omega} (k \nabla \phi_j) \cdot \nabla \phi_i \, d\Omega,
\]
\[
= a(\phi_j, \phi_i)
\]
\[
f_i = \int_{\Omega} f \phi_i \, d\Omega - \int_{\Gamma_q} q_0 \phi_i \, ds - \sum_{x_j \in \Gamma_T} T_0(x_j)K_{ij},
\]
\[
= l(\phi_i) - \sum_{x_j \in \Gamma_T} T_0(x_j)a(\phi_j, \phi_i)
\]
Remark. The stiffness matrix and force vector above can be also written by using the abstract formalism introduced earlier:

\[
K_{ij} = \int_{\Omega} (k \nabla \phi_j) \cdot \nabla \phi_i \, d\Omega,
\]

\[= a(\phi_j, \phi_i)\]

\[
f_i = \int_{\Omega} f \phi_i \, d\Omega - \int_{\Gamma_q} q_0 \phi_i \, ds - \sum_{x_j \in \Gamma_T} T_0(x_j) K_{ij},
\]

\[= l(\phi_i) - \sum_{x_j \in \Gamma_T} T_0(x_j)a(\phi_j, \phi_i)\]

Remark. All the integrals above are fairly local due to local basis functions.
Remark. The stiffness matrix and force vector above can be also written by using the abstract formalism introduced earlier:

\[
K_{ij} = \int_{\Omega} (k \nabla \phi_j) \cdot \nabla \phi_i \, d\Omega, \\
= a(\phi_j, \phi_i)
\]

\[
f_i = \int_{\Omega} f \phi_i \, d\Omega - \int_{\Gamma_q} q_0 \phi_i \, ds - \sum_{x_j \in \Gamma_T} T_0(x_j)K_{ij}, \\
= l(\phi_i) - \sum_{x_j \in \Gamma_T} T_0(x_j)a(\phi_j, \phi_i)
\]

Remark. All the integrals above are fairly local due to local basis functions.

Remark. In 3D finite element methods, \(\mathbf{x} = (x, y)\) is replaced by \(\mathbf{x} = (x, y, z)\), and, accordingly, triangles are replaced by *tetrahedra* and quadrangles by *hexahedra* – and element edges by element *facets*.

The main principles and techniques related to the finite element formulations remain the same, however.
QUESTIONS?

ANSWERS”

LECTURE BREAK!