Example 31 (Quadratic functionals).

Analogy $\longrightarrow$ parabola:

$$\begin{array}{ccccc} J(v) & = & \frac{1}{2}a(v,v) & - & f(v) \\ \updownarrow & & \updownarrow & & \updownarrow \\ f(x) & = & ax^2 & + & bx \end{array}$$

quadratic functional $\mathbb{R}^2 \mapsto \mathbb{R}$ $\quad \triangleright$

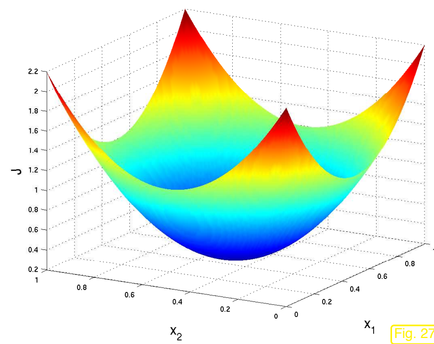


Fig. 27

3 The Finite Element Method (FEM)

Problem :	scalar second-order elliptic boundary value problem
Perspective :	variational interpretation in Sobolev spaces $\longrightarrow$ Sect. 2.6.2
Objective :	algorithm for the computation of an approximate numerical solution

3.1 Fundamentals

Moot point ($\longrightarrow$ Sect. 1.2): any computer can only handle a finite amount of information (reals)

Variational boundary value problem $\xrightarrow{\text{DISCRETIZATION}}$ System of a finite number of equations for real unknowns

3.1.1 Galerkin discretization

Abstract discussion: start from linear variational problem (see Sect. 2.6, (2.6.1))

$$u \in V: \quad a(u,v) = f(v) \quad \forall v \in V \ , \tag{2.6.1}$$

V = Hilbert space with norm $\|\cdot\|_V$, $a(\cdot,\cdot)$ continuous bilinear form, f continuous linear form.

Norm of $a(\cdot,\cdot)$:
$$C_A := \sup_{v \in V\setminus\{0\}} \sup_{u \in V\setminus\{0\}} \frac{|a(u,v)|}{\|u\|_V \|v\|_V} < \infty$$

Assumption: V-ellipticity of $a(\cdot,\cdot)$, see Def. 2.6.6:

$$\exists \gamma > 0: \quad |a(u,u)| \geq \gamma \|u\|_V^2 \quad \forall u \in V \ . \tag{2.8.1}$$

Remark. If $a(\cdot,\cdot)$ symmetric ($\longrightarrow$ inner product, see Def. 2.6.3) and $\|\cdot\|_V$ = energy norm $\|\cdot\|_A$

$\blacktriangleright$ $\gamma, C_A = 1$ (prove with Cauchy-Schwarz inequality)

Idea of Galerkin discretization

Replace V in (2.6.1) with a finite dimensional subspace V_N (discrete trial/test space).

Notation: N = formal index, tagging "discrete entities" ($\longrightarrow$ "finite amount of information")

Discrete variational problem

$$u_N \in V_N: \quad a(u_N, v_N) = f(v_N) \quad \forall v_N \in V_N \ . \tag{3.1.1}$$

Lax-Milgram Lemma Thm. 2.6.7 $\Rightarrow$ Existence & Uniqueness of solution $u_N \in V_N$, stability

$$\|u_N\|_V \leq \frac{1}{\gamma} \sup_{v_N \in V_N\setminus\{0\}} \frac{|f(v_N)|}{\|v_N\|_V} \ .$$

Issues:

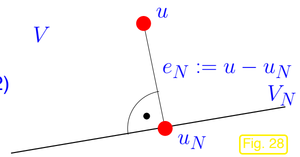
1. How "accurate" is the Galerkin solution u_N?
   (a) What measure for accuracy?
   (b) How to assess accuracy?
2. How to convert (3.1.1) into (linear) system of equations?

Ad 1(a): Focus on norm $\|\cdot\|_V$ (and $\|\cdot\|_A$, if $a(\cdot,\cdot)$ inner product)

Galerkin orthogonality

$$a(u - u_N, v_N) = 0 \quad \forall v_N \in V_N \ . \tag{3.1.2}$$

[Geometric meaning for inner product $a(\cdot,\cdot) \rightarrow$]



► Discretization error $e_N := u - u_N$ "$a(\cdot,\cdot)$-orthogonal" to discrete trial/test space V_N

Remark 32. If $a(\cdot,\cdot)$ is inner product on V: "Phythagoras' theorem" → Fig. 28

$$\|u - u_N\|_A^2 = \|u\|_A^2 - \|u_N\|_A^2 \ . \tag{3.1.3}$$

(3.1.3) ➣ simple formula for computation of energy norm of Galerkin discretization error in numerical experiments with known u.

△

Theorem 3.1.1 (Cea's lemma). *If* $a(\cdot,\cdot)$ *= continuous,* V*-elliptic, bilinear form,* $V_N \subset V$ *finite dimensional subspace,* $u \in V$ / $u_N \in V_N$ *solve* (2.6.1)/(3.1.1), *then*

$$\|u - u_N\|_V \leq \frac{C_A}{\gamma} \inf_{v_N \in V_N \setminus \{0\}} \|u - v_N\|_V$$

Proof. By Galerkin-orthogonality (3.1.2), for all $v_N \in V_N$

$$\begin{aligned} \gamma \|u - u_N\|_V^2 &\leq |a(u - u_N, u - u_N) + a(u - u_N, u_N - v_N)| \\ &= |a(u - u_N, u - v_N)| \leq C_A \|u - u_N\|_V \|u - v_N\|_V \ . \\ \Rightarrow \quad \|u - u_N\|_V &\leq \frac{C_A}{\gamma} \inf_{v_N \in V_N \setminus \{0\}} \|u - v_N\|_V \ , \end{aligned}$$

because v_N arbitrary. □

Quasi-optimality of Galerkin solutions: with $C > 0$ *independent* of u, V_N

$$\underbrace{\|u - u_N\|_V}_{\uparrow} \leq C \underbrace{\inf_{v_N \in V_N} \|u - v_N\|_V}_{\uparrow} \ , \tag{3.1.4}$$

(norm of) discretization error best approximation error (3.1.5)

► 1(b): To assess accuracy of Galerkin solution: study capability of V_N to approximate u!

► "Monotonicity" of best approximation

Trial test spaces $V_N, V_N' \subset V$: $V_N \subset V_N' \quad \Rightarrow \quad \inf_{v_N \in V_N'} \|u - v_N\|_V \leq \inf_{v_N \in V_N} \|u - v_N\|_V$.

Enhance accuracy by enlarging ("refining") trial space.

Reminder:

Definition 3.1.2 (Linear operator). *Let* V, W *be real vector spaces. A mapping* $\mathsf{T} : V \mapsto W$ *is a linear operator, if*

$$\mathsf{T}(\alpha u + \beta v) = \alpha \mathsf{T}(u) + \beta \mathsf{T}(v) \quad \forall u, v \in V, \forall \alpha, \beta \in \mathbb{R} \ .$$

Reminder:

Definition 3.1.3 (projection). *A linear operator* $\mathsf{P} : V \mapsto V$ *on a vector space* V *is a projection, if* $\mathsf{P}^2 = \mathsf{P}$.

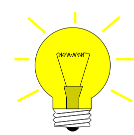
Definition 3.1.4 (Galerkin projection). *Under the assumptions of Cea's lemma Thm. 3.1.1 the Galerkin projection* $\mathsf{P}_N : V \mapsto V_N \subset V$ *is defined by*

$$a(\mathsf{P}_N u, v_N) = a(u, v_N) \quad \forall v_N \in V_N \ .$$

[Lax-Milgram Lemma Thm 2.6.7 $\Rightarrow$ P_N well defined and continuous]

3.1.2 The (linear) algebraic setting

[Now we tackle issue 2. (conversion of (3.1.1) into system of equations)]



I. Introduce (ordered) basis $\mathfrak{B}_N$ of V_N:

$$\mathfrak{B}_N := \{b_N^1, \ldots, b_N^N\} \subset V_N \quad , \quad V_N = \mathrm{Span}\,\{\mathfrak{B}_N\} \quad , \quad N := \dim(V_N) \ .$$

II. Basis representations
$$\begin{aligned} u_N &= \mu_1 b_N^1 + \cdots + \mu_N b_N^N \ , \quad \mu_i \in \mathbb{R} \\ v_N &= \nu_1 b_N^1 + \cdots + \nu_N b_N^N \ , \quad \nu_i \in \mathbb{R} \end{aligned}$$
in (3.1.1).

(3.1.1): $a(u_N, v_N) = f(v_N) \quad \forall v_N \in V_N$

$$a(\mu_1 b_N^1 + \dots + \mu_N b_N^N, \nu_1 b_N^1 + \dots + \nu_N b_N^N) = f(\nu_1 b_N^1 + \dots + \nu_N b_N^N) \quad \forall \nu_1, \dots, \nu_N \in \mathbb{R} \,,$$

$$\sum_{k=1}^{N}\sum_{j=1}^{N} \mu_k \nu_j a(b_N^k, b_N^j) = \sum_{j=1}^{N} \nu_j f(b_N^j) \quad \forall \nu_1, \dots, \nu_N \in \mathbb{R} \,,$$

$$\sum_{j=1}^{N} \nu_j \left(\sum_{k=1}^{N} \mu_k a(b_N^k, b_N^j) - f(b_N^j) \right) = 0 \quad \forall \nu_1, \dots, \nu_N \in \mathbb{R} \,,$$

$$\sum_{k=1}^{N} \mu_k a(b_N^k, b_N^j) = f(b_N^j) \quad \text{for } j = 1, \dots, N \,.$$

$$\boxed{\mathbf{A}\vec{\boldsymbol{\mu}} = \vec{\boldsymbol{\varphi}}} \,, \qquad \mathbf{A} = \left(a(b_N^k, b_N^j) \right)_{j,k=1}^{N} \in \mathbb{R}^{N,N} \,, \; \vec{\boldsymbol{\varphi}} = \left(f(b_N^j) \right)_{j=1}^{N} \,, \quad \vec{\boldsymbol{\mu}} = (\mu_1, \dots, \mu_N)^T \in \mathbb{R}^N$$

Discrete variational problem
$u_N \in V_N$: $a(u_N, v_N) = f(v_N) \quad \forall v_N \in V_N$

$\xrightarrow{\text{Choosing basis } \mathfrak{B}_N}$

Linear system of equations
$\mathbf{A}\vec{\boldsymbol{\mu}} = \vec{\boldsymbol{\varphi}}$

Stiffness matrix: $\mathbf{A} = \left(a(b_N^k, b_N^j) \right)_{j,k=1}^{N} \in \mathbb{R}^{N,N}$,

Load vector: $\vec{\boldsymbol{\varphi}} = \left(f(b_N^j) \right)_{j=1}^{N} \in \mathbb{R}^N$,

Coefficient vector: $\vec{\boldsymbol{\mu}} = (\mu_1, \dots, \mu_N)^T \in \mathbb{R}^N$,

Recovery of solution: $u_N = \sum_{k=1}^{N} \mu_k b_N^k$.

Corollary 3.1.5. (3.1.1) *has unique solution* $\Leftrightarrow$ $\mathbf{A}$ *regular*

Impact of choice of basis ?

Choice of $\mathfrak{B}_N$ does not affect u_N $\Rightarrow$ No impact on discretization error !

Properties of matrix $\mathbf{A}$ crucially depend on basis $\mathfrak{B}_N$!

Lemma 3.1.6. *Consider* (3.1.1) *and two bases of* V_N,

$$\mathfrak{B}_N := \{b_N^1, \dots, b_N^N\} \quad , \quad \underline{\mathfrak{B}}_N := \{\underline{b}_N^1, \dots, \underline{b}_N^N\} \,,$$

related by

$$\underline{b}_N^j = \sum_{k=1}^{N} s_{jk} b_N^k \quad \text{with } \mathbf{S} = (s_{jk})_{j,k=1}^{N} \in \mathbb{R}^{N,N} \text{ regular.}$$

▶ *Stiffness matrices* $\mathbf{A}, \underline{\mathbf{A}} \in \mathbb{R}^{N,N}$, *load vectors* $\vec{\boldsymbol{\varphi}}, \underline{\vec{\boldsymbol{\varphi}}} \in \mathbb{R}^N$, *and coefficient vectors* $\vec{\boldsymbol{\mu}}, \underline{\vec{\boldsymbol{\mu}}} \in \mathbb{R}^N$, *respectively, satisfy*

$$\underline{\mathbf{A}} = \mathbf{S}\mathbf{A}\mathbf{S}^T \quad , \quad \underline{\vec{\boldsymbol{\varphi}}} = \mathbf{S}\vec{\boldsymbol{\varphi}} \quad , \quad \underline{\vec{\boldsymbol{\mu}}} = \mathbf{S}^{-T}\vec{\boldsymbol{\mu}} \,. \tag{3.1.6}$$

Proof.

$$\underline{\mathbf{A}}_{lm} = a(\underline{b}_N^m, \underline{b}_N^l) = \sum_{k=1}^{N}\sum_{j=1}^{N} s_{mk} a(b_N^k, b_N^j) s_{lj} = \sum_{k=1}^{N} \Big(\underbrace{\sum_{j=1}^{N} s_{lj} \mathbf{A}_{jk}}_{(\mathbf{SA})_{lk}} \Big) s_{mk} = (\mathbf{SAS}^T)_{lm} \,,$$

Reminder of linear algebra:

Definition 3.1.7 (Congruent matrices). *Two matrices* $\mathbf{A} \in \mathbb{R}^{N,N}$, $\mathbf{B} \in \mathbb{R}^{N,N}$, $N \in \mathbb{N}$, *are called congruent, if there is a regular matrix* $\mathbf{S} \in \mathbb{R}^{N,N}$ *such that* $\mathbf{B} = \mathbf{SAS}^T$.

▶ Equivalence relation on square matrices

Lemma 3.1.8. *Matrix property invariant under congruence* $\Leftrightarrow$ *Property of stiffness matrix invariant under change of basis* $\mathfrak{B}_N$

Matrix properties invariant under congruence =
- regularity
- symmetry
- positive definiteness

Reminder:

Definition 3.1.9 (Positive definite matrix). *Matrix* $\mathbf{B} \in \mathbb{R}^{N,N}$, $N \in \mathbb{N}$, *is positive definite* $\Leftrightarrow$ $\vec{\boldsymbol{\xi}}^T \mathbf{B} \vec{\boldsymbol{\xi}} > 0$ *for all* $\vec{\boldsymbol{\xi}} \in \mathbb{R}^N \setminus \{0\}$.

3.1.3 Principles of FEM

1D case $\longrightarrow$ Sect. 1.2.3.1, now higher dimensional, "complicated" domain Ω:

$\Omega \subset \mathbb{R}^d$, $d = 2, 3$, bounded computational domain: assumed polygonal $d = 2$, polyhedral $d = 3$

First main ingredient: triangulation/mesh of Ω, *cf.* Sect. 1.2.1

Definition 3.1.10. *A mesh (or triangulation) of $\Omega \subset \mathbb{R}^d$ is a finite collection $\{K_i\}_{i=1}^M$, $M \in \mathbb{N}$, of open non-degenerate polygons ($d = 2$)/polyhedra ($d = 3$) such that*

(A) $\overline{\Omega} = \bigcup\{\overline{K}_i, i = 1, \dots, M\}$,

(B) $K_i \cap K_j = \emptyset \quad \Leftrightarrow \quad i \neq j$,

(C) for all $i, j \in \{1, \dots, M\}$, $i \neq j$, the intersection $\overline{K}_i \cap \overline{K}_j$ is a vertex, edge, or face of both K_i and K_j.

▶ "vertex", "edge", "face" of polygon/polyhedron: $\longrightarrow$ geometric intuition

Terminology: Given mesh $\mathcal{M} := \{K_i\}_{i=1}^M$: K_i called cell or element.
Vertices of a mesh $\longrightarrow$ nodes (set $\mathcal{N}(\mathcal{M})$)

Types of meshes:

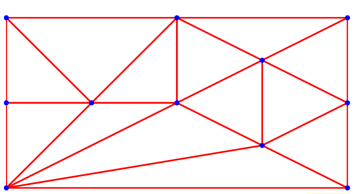


Fig. 29

Triangular mesh in 2D

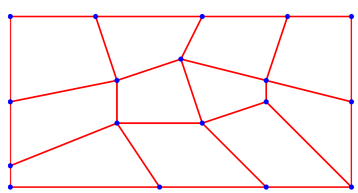


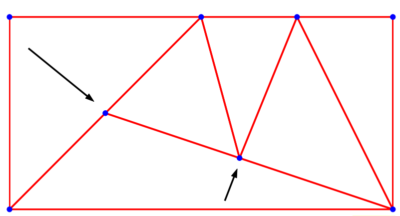
Fig. 30

Quadrilateral mesh in 2D

If (C) does not hold

▶ Triangular non-conforming mesh (with hanging nodes)

$\overline{K}_i \cap \overline{K}_j$ is only part of an edge/face for at most one of the adjacent cells.



(However, conforming if degenerate quadrilaterals admitted)

Simplicial mesh = triangular mesh in 2D
tetrahedral mesh in 3D

Second main ingredient: space of piecewise polynomial functions

$$V_N := \{v \in V \colon v_{|K} \in \mathcal{P}_p(K)\ \forall K \in \mathcal{M}\}\ ,$$

$\mathcal{P}_p(K)$ = polynomials of degree $\leq p$ on cell K .

Note: $v \in V \longrightarrow$ conformity conditions at interelement boundaries

Lemma 3.1.11 (Conformity condition for H^1)**.** *Let $\mathcal{M} := \{K_i\}_{i=1}^M$ be a triangulation ($\longrightarrow$ Def. 3.1.10) of $\Omega \subset \mathbb{R}^d$ and assume that $v : \Omega \mapsto \mathbb{R}$ satisfies that $v_{|K}$ can be extended to a function in $C^\infty(\overline{K})$ for any $K \in \mathcal{M}$. Then*

$$v \in H^1(\Omega) \quad \Leftrightarrow \quad v \in C^0(\overline{\Omega})\ .$$

▶ Conformity condition for H^1 = global continuity (C^0, **not** C^1 ! $\longrightarrow$ Ex. 28)
(recall physical constraints on temperature distributions!)

Thanks to notion of weak derivative, Sect. 2.4 **!**

Definition 3.1.12 (Conformity)**.** *Let V be a function space. A $\mathcal{M}$-piecewise polynomial space V_N is called V-conforming, if $V_N \subset V$.*

Third main ingredient: Locally supported basis functions

Basis functions $b_N^1, \dots, b_N^N$ for a finite element trial/test space V_N built on a mesh $\mathcal{M}$ satisfy:

- each b_N^i associated with a single cell/edge/face/vertex of $\mathcal{M}$,
- $\operatorname{supp}(b_N^i) = \bigcup\{\overline{K} \colon K \in \mathcal{M}, \mathbf{p} \subset \overline{K}\}$, if b_N^i associated with cell/edge/face/vertex $\mathbf{p}$.

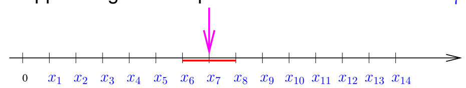
Finite element terminology: b_N^i = global shape/basis functions

Example 33 (Supports of global shape functions in 1D). $\rightarrow$ Sect. 1.2.3.1

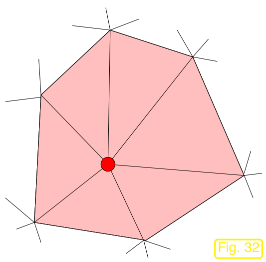
- $\Omega =]a,b[\,\hat{=}$ interval
- Equidistant mesh

$$\mathcal{M} := \{]x_{j-1}, x_j[,\ j = 1,\dots,N\}\ ,$$
$$x_j := a + hj,\ h := (b-a)/N,\ N \in \mathbb{N}.$$

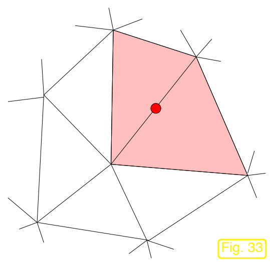
Support of global shape function associated with x_7



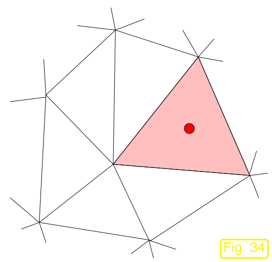
Example 34 (Supports of global shape functions on triangular mesh).



Support of node-associated basis function



Support of edge-associated basis function



Support of cell-associated basis function

Rationale for small supports **?**

Recall bilinear form $\leftrightarrow -\Delta$: $\quad a(u,v) := \int_\Omega \mathbf{grad}\, u \cdot \mathbf{grad}\, v \,\mathrm{d}\mathbf{x}$

Use triangular mesh $\mathcal{M}$, test/trial space $V_N \subset H^1(\Omega)$ with basis $\mathfrak{B}_N := \{b_N^1, \dots, b_N^N\}$ ($\rightarrow$ Sect. 3.1.2)

▶ Stiffness matrix $\mathbf{A} \in \mathbb{R}^{N,N}$ with $\quad a_{ij} := a(b_N^j, b_N^i),\ i,j = 1,\dots,N$

▼

b_N^i, b_N^j associated with nodes not linked by an edge ▶ $a_{ij} = 0$ (because $\mathrm{vol}(\mathrm{supp}(b_N^i) \cap \mathrm{supp}(b_N^j)) = 0$)

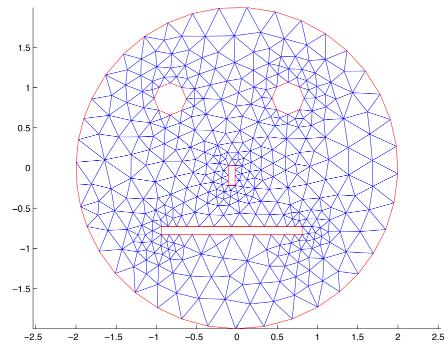
▼

Finite element stiffness matrices are sparse

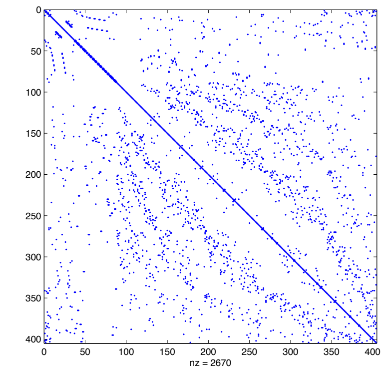
Definition 3.1.13 (Sparse matrix). *A matrix* $\mathbf{A} \in \mathbb{R}^{N,N}$ *is called sparse, if* $\mathrm{nnz}(\mathbf{A}) := \sharp\{(i,j)\colon a_{ij} \neq 0\} \ll N^2$.

Example 35 (Sparse stiffness matrices).

V_N: one basis function associated with each vertex



Finite element mesh $\mathcal{M}$



Resulting sparsity pattern of stiffness matrix

Visualization of sparsity pattern: MATLAB-`spy()`-Funktion

Remark 36 (Storing sparse matrices).

- Special (efficient) storage formats for sparse matrices, e.g., CRS-format
- Special MATLAB commands: `sparse, spones, speye, spdiags`
  ($\rightarrow$ use **mandatory** !)

3.1.4 Linear H^1-conforming finite elements

$\mathcal{M}$ = simplicial mesh of polygonal/polyhedral computational domain $\Omega \subset \mathbb{R}^d$, $d = 2,3$

Linear H^1-conforming finite elements

= Simplest $H^1(\Omega)$-conforming finite element space

= Simplest finite element scheme for scalar second order elliptic BVP on Ω

$$\mathcal{S}_1^0(\mathcal{M}) := \{v \in C^0(\overline{\Omega}) \colon v_{|K} \in \mathcal{P}_1(K)\ \forall K \in \mathcal{M}\} \subset H^1(\Omega)$$

Representation: $\mathcal{P}_1(K) := \{\mathbf{x} \mapsto \alpha + \boldsymbol{\beta}\cdot\mathbf{x},\ \mathbf{x} \in K,\quad \alpha \in \mathbb{R}, \boldsymbol{\beta} \in \mathbb{R}^d\}$.

(space of d-variate polynomials of total degree ≤ 1)

► $\dim \mathcal{P}_1(K) = d + 1$

Notation: $\mathcal{S}_1^0(\mathcal{M})$ — continuous functions, *cf.* $C^0(\Omega)$; locally 1st degree polynomials, *cf.* $\mathcal{P}_1$

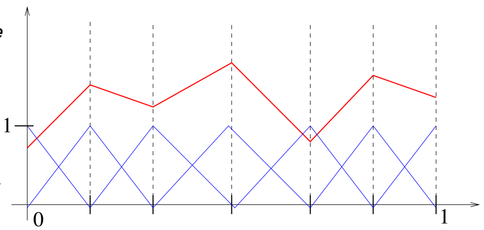
Example 37. *($H^1(\Omega)$-conforming linear finite element space in 1D)*

$d = 1$, $\Omega =]0,1[$,

mesh $\mathcal{M}$ = partion of $]0,1[$ into intervals

red: function $\in \mathcal{S}_1^0(\mathcal{M})$

blue: hat function basis of $\mathcal{S}_1^0(\mathcal{M})$ $\longrightarrow$



Locally supported basis functions in 2D **?**

On a triangle T with vertices $\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3$: $q \in \mathcal{P}_1(T)$ uniquely determined by values $q(\mathbf{a}^i)$.

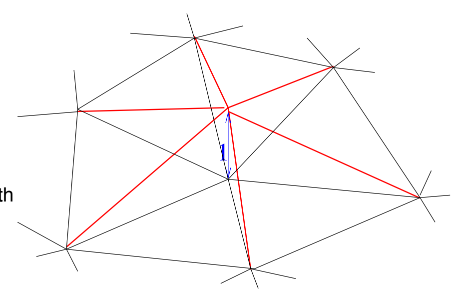
► $v_N \in \mathcal{S}_1^0(\mathcal{M})$ uniquely determined by $\{v_N(\mathbf{x}),\ \mathbf{x}$ node of $\mathcal{M}\}$!

► $\dim \mathcal{S}_1^0(\mathcal{M}) = \sharp\mathcal{N}(\mathcal{M})$ ($\mathcal{N}(\mathcal{M})$ = set of vertices of $\mathcal{M}$)

If $\mathcal{N}(\mathcal{M}) = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}$, nodal basis $\mathfrak{B}_N := \{b_N^1, \dots, b_N^N\}$ of $\mathcal{S}_1^0(\mathcal{M})$ defined by $b_N^i(\mathbf{x}^j) = \delta_{ij}$.

Piecewise linear nodal basis function
("hat function")
(= global shape function for $\mathcal{S}_1^0(\mathcal{M})$)

► coefficient μ_j = "nodal value" of u_N at j-th node of $\mathcal{M}$



$$\text{Global shape functions} \xrightarrow{\text{Restriction to element}} \text{local shape functions} \qquad (3.1.7)$$

Example 38 (Local shape functions for $\mathcal{S}_1^0(\mathcal{M})$).

Triangle K with vertices $\mathbf{a}^1 = \binom{0}{0}, \mathbf{a}^2 = \binom{1}{0}, \mathbf{a}^3 = \binom{0}{1}$:

Local shape functions: $b_K^1(\mathbf{x}) = 1 - x_1 - x_2$, $b_K^2(\mathbf{x}) = x_1$, $b_K^3(\mathbf{x}) = x_2$.

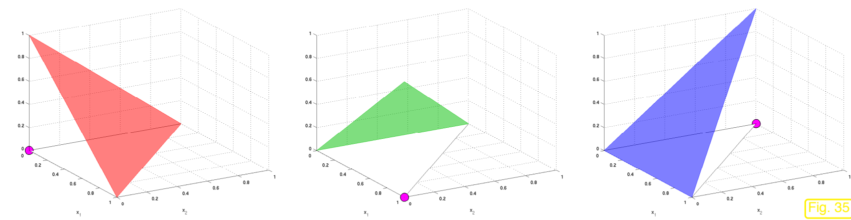


Fig. 35

► Local shape functions for $\mathcal{S}_1^0(\mathcal{M})$ on triangle/tetrahedron = barycentric coordinate functions

Definition 3.1.14 (Barycentric coordinates). *Given $d+1$ points $\mathbf{a}^1, \dots, \mathbf{a}^{d+1} \in \mathbb{R}^d$ that do not lie in a hyperplane the barycentric coordinates $\lambda_1 = \lambda_1(\mathbf{x}), \dots, \lambda_{d+1} = \lambda_{d+1}(\mathbf{x}) \in \mathbb{R}$ of $\mathbf{x} \in \mathbb{R}^d$ are uniquely defined by*

$$\lambda_1(\mathbf{x}) + \dots + \lambda_{d+1}(\mathbf{x}) = 1 \quad , \quad \lambda_1(\mathbf{x})\,\mathbf{a}^1 + \dots + \lambda_{d+1}(\mathbf{x})\mathbf{a}^{d+1} = \mathbf{x} \quad \forall \mathbf{x} \in \mathbb{R}^d .$$

Barycentric coordinates obtained by solving

$$\begin{pmatrix} a_1^1 & \cdots & a_1^{d+1} \\ \vdots & & \vdots \\ a_d^1 & \cdots & a_d^{d+1} \\ 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} \lambda_1(\mathbf{x}) \\ \vdots \\ \lambda_d(\mathbf{x}) \\ \lambda_{d+1}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} x_1 \\ \vdots \\ x_d \\ 1 \end{pmatrix} . \qquad (3.1.8)$$

Corollary 3.1.15. *Given $d+1$ points $\mathbf{a}^1, \dots, \mathbf{a}^{d+1} \in \mathbb{R}^d$ as in Def. 3.1.14, the barycentric coordinates are affine linear functions on $\mathbb{R}^d$, which satisfy*

$$\lambda_j(\mathbf{a}^i) = \delta_{ij} := \begin{cases} 1\,, & \text{if } i = j\,, \\ 0\,, & \text{else}\,, \end{cases} \quad 1 \leq i,j \leq d+1 .$$

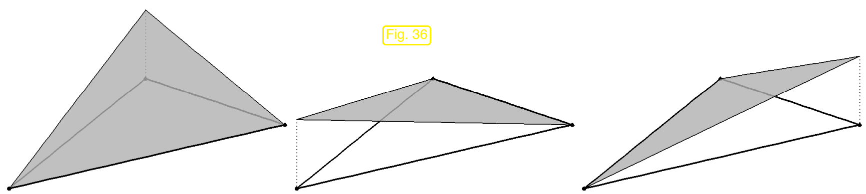


Fig. 36

Barycentric coordinate functions (= Local shape functions for $\mathcal{S}_1^0(\mathcal{M})$) on a triangle

How to get $H_0^1(\Omega)$-conforming finite element space $\mathcal{S}_{1,0}^0(\mathcal{M}) := \mathcal{S}_1^0(\mathcal{M}) \cap H_0^1(\Omega)$ **?**

► Discard nodal basis functions associated with vertices on $\partial\Omega$!

Remark 39. Piecewise linear finite element subspace of $H_*^1(\Omega)$ **?**

► There exist no locally supported piecewise linear basis functions.

3.1.5 Simplicial Lagrangian finite elements

$\mathcal{M}$ = simplicial mesh of polygonal/polyhedral computational domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$

Idea: Use higher degree polynomials $\longrightarrow$ "better accuracy" (*cf.* interpolation)

Higher degree polynomials $\mathcal{P}_p(\mathbb{R}^d) := \{\mathbf{x} \in \mathbb{R}^d \mapsto \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^d,\, |\boldsymbol{\alpha}|\leq p} \kappa_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}\,,\ \kappa_{\boldsymbol{\alpha}} \in \mathbb{R}\}$.

Notation: $\boldsymbol{\alpha}$ = "multiindex" $(\alpha_1, \dots, \alpha_d)$, $|\boldsymbol{\alpha}| = \alpha_1 + \dots + \alpha_d$, $\mathbf{x}^{\boldsymbol{\alpha}} := x_1^{\alpha_1} \cdot \dots \cdot x_d^{\alpha_d}$.

Example: $\mathcal{P}_2(\mathbb{R}^2) = \mathrm{Span}\left\{1, x_1, x_2, x_1^2, x_2^2, x_1x_2\right\}$

Lemma 3.1.16. $\dim \mathcal{P}_p(\mathbb{R}^d) = \binom{d+p}{p}$ *for all* $p \in \mathbb{N}$, $d \in \mathbb{N}$

Definition 3.1.17 (Higher order Lagrangian finite element spaces)**.** *Space of p-th degree Lagrangian finite element functions on mesh* $\mathcal{M}$

$$\mathcal{S}_p^0(\mathcal{M}) := \{v \in C^0(\overline{\Omega}) \colon v_{|K} \in \mathcal{P}_p(K) \quad \forall K \in \mathcal{M}\} .$$

Notation: $\mathcal{S}_p^0(\mathcal{M})$ — continuous functions, *cf.* $C^0(\Omega)$; locally polynomials of degree p , *cf.* $\mathcal{P}_p(\mathbb{R}^d)$

Construction: $\boxed{\text{Local shape functions}}$ $\xrightarrow{\text{"Glueing"}}$ $\boxed{\text{Global FE space}}$

(Glueing must ensure global continuity $\leftrightarrow$ $H^1(\Omega)$-conformity)

! Design of local shape functions must make glueing possible

Example 40 (Quadratic Lagrangian finite elements).

Local shape functions for $\mathcal{P}_2(K)$, K triangle:

$$\begin{aligned} b_1^K &= -\lambda_1(1-2\lambda_1)\,, & b_{12}^K &= 4\lambda_1\lambda_2\,,\\ b_2^K &= -\lambda_2(1-2\lambda_2)\,, & b_{13}^K &= 4\lambda_1\lambda_3\,,\\ b_3^K &= -\lambda_3(1-2\lambda_3)\,, & b_{23}^K &= 4\lambda_2\lambda_3\,. \end{aligned}$$

λ_i = barycentric coordinate function ($\rightarrow$ Def. 3.1.14) for vertex $\mathbf{a}_i$

$$b_j^K(\mathbf{a}_i) = \delta_{ij}\,, \quad i,j \in \{1,2,3,(12),(23),(13)\}\,.$$

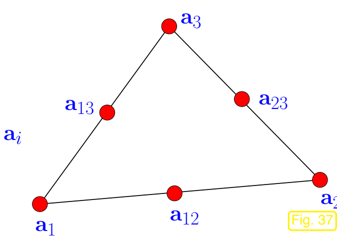


Fig. 37

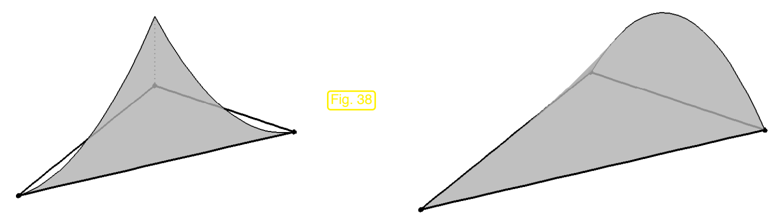


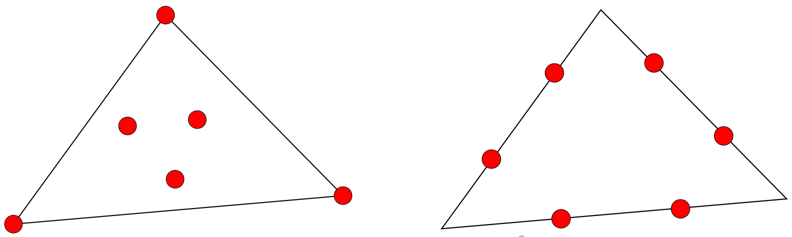
Fig. 38

► Local shape functions = Lagrangian (interpolatory) polynomials for local nodes in K

► Specifying local interpolation nodes $\Leftrightarrow$ specifying local shape functions

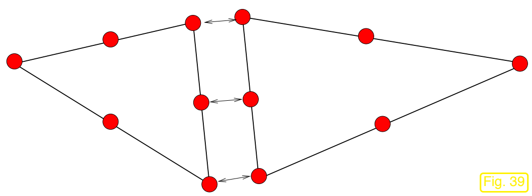
When are local nodes $\mathbf{q}_i$, $i = 1, \dots, Q$, (for $\mathcal{P}_p(K)$) suitable for "glueing" **?**

- Unisolvence: $v \in \mathcal{P}_p(K)\colon\ v(\mathbf{q}_i) = 0\ \forall i \quad \Leftrightarrow \quad v \equiv 0$
- Fixing traces: locally unisolvent interpolation on each vertex/edge/face

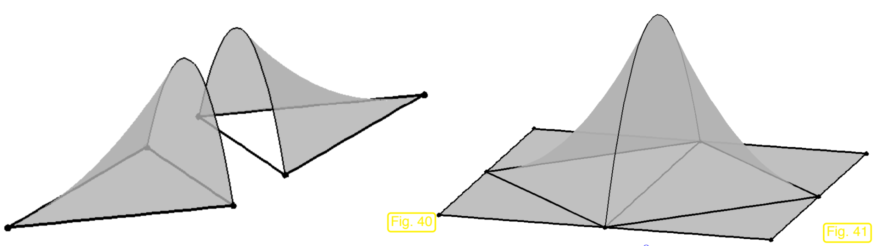


Invalid choice of local nodes for $\mathcal{S}_2^0(\mathcal{M})$

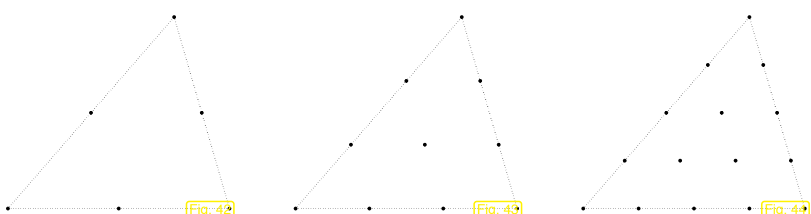
- Interelement matching: corresponding nodes on joint edges/faces



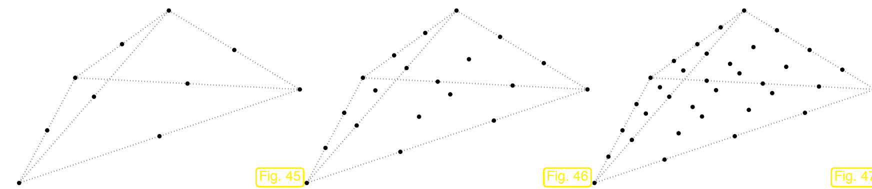
Matching nodes for quadratic Lagrangian finite elements



"Glueing": edge-associated local and resulting global shape function for $\mathcal{S}_2^0(\mathcal{M})$, $\mathcal{M}$ triangular



Location of local (interpolation) nodes for triangular Lagrangian finite elements of degree 2 (left), degree 3 (middle), and degree 4 (right)



Local nodes for tetrahedral Lagrangian finite elements (left: $p=2$, middle: $p=3$, right: $p=4$)

Can we find other locally supported bases for $\mathcal{S}_2^0(\mathcal{M})$ **?** **YES!**

▶ Alternative p-hierarchical local shape functions:

$$b_1^K = \lambda_1 \,, \quad b_{12}^K = 4\lambda_1\lambda_2 \,,$$
$$b_2^K = \lambda_2 \,, \quad b_{13}^K = 4\lambda_1\lambda_3 \,,$$
$$b_3^K = \lambda_3 \,, \quad b_{23}^K = 4\lambda_2\lambda_3 \,.$$

▶ Set comprises local shape functions for $p=1$.

▼ Glueing can easily be accomplished

No "canonical" local shape functions/global basis functions for higher order Lagrangian finite elements.

▶ Selection of $\mathfrak{B}_N$ to get "good" matrix properties.

3.1.6 Parametric finite elements

Definition 3.1.18 (Affine transformation)**.** *Mapping* $\mathbf{\Phi}:\mathbb{R}^d \mapsto \mathbb{R}^d$ *affine, if* $\mathbf{\Phi}(\mathbf{x}) = \mathbf{F}\mathbf{x} + \boldsymbol{\tau}$ *with* $\mathbf{F}\in\mathbb{R}^{d,d}$, $\boldsymbol{\tau}\in\mathbb{R}^d$.

Usually:

All elements of a mesh = affine images of reference element(s) $\widehat{K}$

$\exists\widehat{K}\quad \forall K\in\mathcal{M}$: $\exists\mathbf{F}_K\in\mathbb{R}^{d,d}$ regular, $\boldsymbol{\tau}_K\in\mathbb{R}^d$: $K = \mathbf{\Phi}_K(\widehat{K})$ with $\mathbf{\Phi}_K(\widehat{\mathbf{x}}) := \mathbf{F}_K\widehat{\mathbf{x}} + \boldsymbol{\tau}_K$.

“Unit triangle”: $\widehat{K} = \left\langle \binom{0}{0}, \binom{1}{0}, \binom{0}{1} \right\rangle$

For $K = \text{convex}\,\{\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3\}$:

$$\mathbf{F}_K = \begin{pmatrix} a_1^2 - a_1^1 & a_1^3 - a_1^1 \\ a_2^2 - a_2^1 & a_2^3 - a_2^1 \end{pmatrix} \;, \quad \boldsymbol{\tau}_K = \mathbf{a}^1 .$$

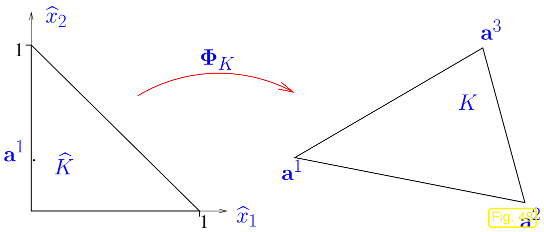


Fig. 48

Transformations of elements ⇒ transformation of functions:

Definition 3.1.19 (Pullback)**.** *Given domains $\Omega, \widehat{\Omega}$ and a bijective mapping $\mathbf{\Phi} : \widehat{\Omega} \mapsto \Omega$, the pullback $\mathbf{\Phi}^* u : \widehat{\Omega} \mapsto \mathbb{R}$ of a function $u : \Omega \mapsto \mathbb{R}$ is defined by*

$$(\mathbf{\Phi}^* u)(\widehat{\mathbf{x}}) := u(\mathbf{\Phi}(\widehat{\mathbf{x}})) \;, \quad \widehat{\mathbf{x}} \in \widehat{\Omega} .$$

Notation: If unambiguous: $\widehat{u} := \mathbf{\Phi}^* u$

Note: Consider $\mathcal{S}_1^0(\mathcal{M})$, triangle $K \in \mathcal{M}$, unit triangle $\widehat{K}$, affine mapping $\mathbf{\Phi}_K : \widehat{K} \mapsto K$

- b_1^K, b_2^K, b_3^K (standard) local shape functions on K
- $\widehat{b}_1, \widehat{b}_2, \widehat{b}_3$ (standard) local shape functions on $\widehat{K}$

Observation: $\widehat{b}_i = \mathbf{\Phi}_K^* b_i^K$

Terminology: affine equivalent finite elements

▶ All families of Lagrangian finite elements (Sect. 3.1.5) (equipped with “natural” local shape functions) are affine equivalent.

STEP 1: define local shape functions on reference element $\widehat{K}$

STEP 2: local shape functions on $K \in \mathcal{M}$ via pullback $(\mathbf{\Phi}^{-1})^*$ (→ Def. 3.1.19)

Parametric finite elements

Generalization: curvilinear meshes with curved edges/faces (→ Sect. 3.2.8)

Now: $\mathbf{\Phi}_K$ diffeomorphism !
(i.e. $\mathbf{\Phi}_K$ and $\mathbf{\Phi}_K^{-1}$ are differentiable)

$$b_i^K := (\mathbf{\Phi}_K^{-1})^* \widehat{b}_i \,.$$

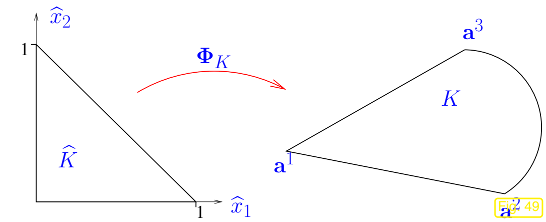


Fig. 49

▶ Application: approximation of curved interfaces/boundaries (→ Sect 3.2.8)

3.1.7 Lagrangian finite elements on quadrilaterals/hexahedra

Parametric construction: start from reference element $\widehat{K} =]0,1[^d$ (unit cube)

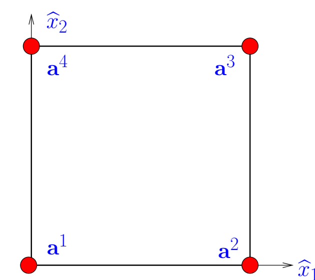
Lowest polynomial degree $p = 1$, 2D: piecewise bilinear finite elements

Local shape functions on $\widehat{K} =]0,1[^2$

$$\widehat{b}_1 = (1-\widehat{x}_1)(1-\widehat{x}_2) \;, \; \widehat{b}_2 = \widehat{x}_1(1-\widehat{x}_2) \;,$$
$$\widehat{b}_3 = \widehat{x}_1\widehat{x}_2 \;, \qquad \widehat{b}_4 = (1-\widehat{x}_1)\widehat{x}_2 \,.$$

Bilinear Lagrangian interpolation polynomials w.r.t. corner points $\mathbf{a}_i$ of $\widehat{K}$

Note: $\widehat{b}_i$ linear on edges



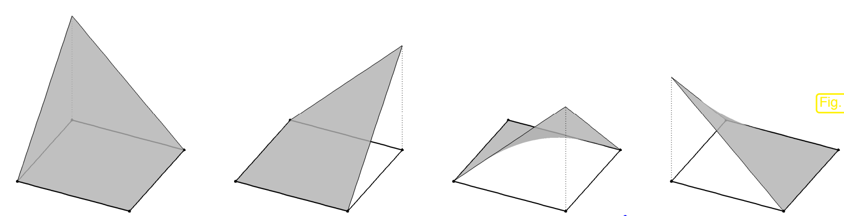
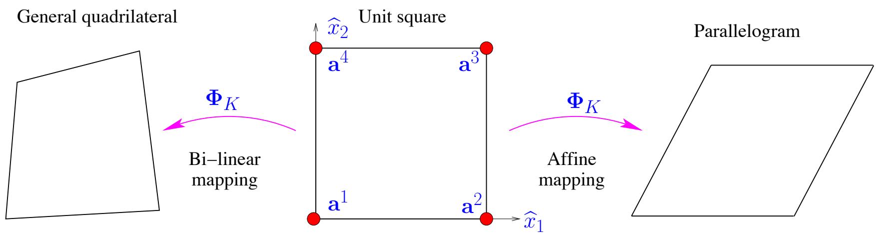


Fig. 50

Bilinear local shape functions on unit square $\widehat{K}$



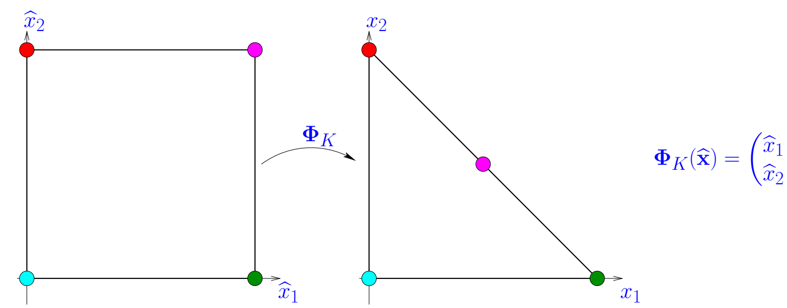
Affine mapping = linear transformation + translation

Bilinear mapping to general quadrilateral:

$$\mathbf{\Phi}_K(\widehat{\mathbf{x}}) = \begin{pmatrix} \alpha_1 + \beta_1\widehat{x}_1 + \gamma_1\widehat{x}_2 + \delta_1\widehat{x}_1\widehat{x}_2 \\ \alpha_2 + \beta_2\widehat{x}_2 + \gamma_2\widehat{x}_2 + \delta_2\widehat{x}_2\widehat{x}_2 \end{pmatrix} \;, \quad \alpha_i, \beta_i, \gamma_i, \delta_i \in \mathbb{R} \;.$$

$\widehat{b}_i$, $i = 1,2,3,4$, bilinear local shape functions on $]0,1[^2$, $\mathbf{\Phi}_K : \widehat{K} \mapsto K$ bilinear $\Rightarrow$ $b_i^K = (\mathbf{\Phi}_K^{-1})^*\widehat{b}_i$ linear on edges of K

Example 41 (Triangle as degenerate quadrilateral).



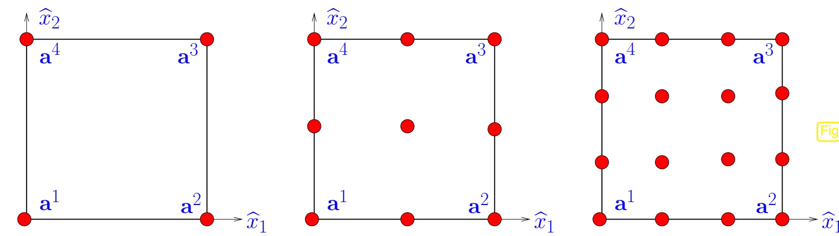
$$\mathbf{\Phi}_K(\widehat{\mathbf{x}}) = \begin{pmatrix} \widehat{x}_1 - \frac{1}{2}\widehat{x}_1\widehat{x}_2 \\ \widehat{x}_2 - \frac{1}{2}\widehat{x}_1\widehat{x}_2 \end{pmatrix}$$

Higher order quadrilateral Lagrangian finite elements:

Definition 3.1.20 (Tensor product polynomials). *Space of tensor product polynomials of degree $p \in \mathbb{N}$ in each coordinate direction*

$$\mathcal{Q}_p(\mathbb{R}^d) := \{\mathbf{x} \mapsto p_1(x_1) \cdot \dots \cdot p_d(x_d), \, p_i \in \mathcal{P}_p(\mathbb{R}), i = 1, \dots, d\} \;.$$

▶ Local trial space $\mathcal{Q}_p(\widehat{K})$ on $\widehat{K} =]0,1[^2$ + parametric construction



Nodal points $p = 1$ Nodal points $p = 2$ Nodal points $p = 3$

Fig. 51

3.1.8 Degrees of freedom*

Recall: Lagrangian local shape functions b_i^K fixed by $b_i^K(\mathbf{q}_j) = \delta_{ij}$ for nodes $\mathbf{q}_i$, $i,j = 1,\dots,Q$, $Q \in \mathbb{N}$ (→ Sects. 3.1.5, 3.1.7).

Definition 3.1.21 (Dual basis). *Given a vector space V with basis $\mathfrak{B} := \{b_1, \dots, b_Q\}$, $Q \in \mathbb{N}$, the corresponding dual basis is a set $l_1, \dots, l_Q$ of linear forms on V such that*

$$l_j(b_i) = \delta_{ij} \;, \quad i,j \in \{1, \dots, Q\} \;.$$

For Lagrangian finite elements $\mathcal{S}_p^0(\mathcal{M})$, on element K:

Nodal evaluation functionals $v \mapsto v(\mathbf{q}_j)$, $j = 1,\dots,Q$, form dual basis w.r.t. basis $\{b_1^K, \dots, b_Q^K\}$ of local trial space $\mathcal{S}_p(K)$.

Definition 3.1.22 (Local degrees of freedom). *A dual basis of the local trial space corresponding to the local shape functions provides local degrees of freedom (d.o.f.).*

Role reversal: degrees of freedom $\Rightarrow$ local shape functions

Example 42 (Cubic Hermitian Finite Elements on triangular mesh).

- Local trial space $V_K := \mathcal{P}_3(K)$ for each $K \in \mathcal{M}$,

- Local degrees of freedom, K triangle with vertices $\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3 \in \mathbb{R}^2$

$$
\begin{aligned}
&l_1(v) = v(\mathbf{a}^1)\,, && l_2(v) = v(\mathbf{a}^2)\,, && l_3(v) = v(\mathbf{a}^3)\,,\\
&l_4(v) = \mathbf{grad}\, v(\mathbf{a}^1)\cdot(\mathbf{a}^2-\mathbf{a}^1)\,, && l_5(v) = \mathbf{grad}\, v(\mathbf{a}^2)\cdot(\mathbf{a}^3-\mathbf{a}^2)\,, && l_6(v) = \mathbf{grad}\, v(\mathbf{a}^3)\cdot(\mathbf{a}^1-\mathbf{a}^3\\
&l_7(v) = \mathbf{grad}\, v(\mathbf{a}^1)\cdot(\mathbf{a}^3-\mathbf{a}^1)\,, && l_8(v) = \mathbf{grad}\, v(\mathbf{a}^2)\cdot(\mathbf{a}^1-\mathbf{a}^2)\,, && l_9(v) = \mathbf{grad}\, v(\mathbf{a}^3)\cdot(\mathbf{a}^2-\mathbf{a}^3\\
&l_{10}(v) = v(\tfrac{1}{3}(\mathbf{a}^1+\mathbf{a}^2+\mathbf{a}^3))\,.
\end{aligned}
$$

Dual basis (→ Def. 3.1.21) **?**

- $\sharp$ functionals $= \dim V_K = \dim \mathcal{P}_3(\mathbb{R}^2) = \binom{3+2}{2} = 10$,

- If $l_j(v) = 0$ for all $j = 1, \dots, 10$, then $\Rightarrow v(\mathbf{a}^i) = 0$ and $\mathbf{grad}\, v(\mathbf{a}^i) = 0$, $i = 1, 2, 3$,
  $\Rightarrow v \in \mathcal{P}_3(K) \equiv 0$ on any edge,
  $\Rightarrow$ ($= 0$ at center of gravity) $v \equiv 0$ for any $v \in V_K$.

▶ Unisolvence of local d.o.f.

Suitable for glueing **?**

YES, because $v_{|\text{edge}}$ uniquely determined by d.o.f. associated with the edge.

> A local degree of freedom l is regarded as associated with an edge E, if $l(v)$ only depends on $v_{|E}$, $\mathbf{grad}\, v_{|E}$, ….

Symbolic notation for local d.o.f. for cubic Hermitian elements:
(filled circle = nodal values, circle = first derivatives, arrows = directional derivatives) ▶

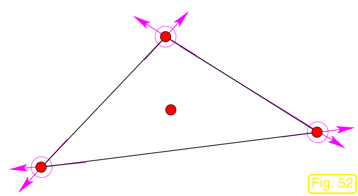


Fig. 52

HOWEVER, alternative choice of local degrees of freedom possible
(on triangle K with vertices $\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3 \in \mathbb{R}^2$)

$$
\begin{aligned}
&l_1(v) = v(\mathbf{a}^1)\,, && l_2(v) = v(\mathbf{a}^2)\,, && l_3(v) = v(\mathbf{a}^3)\,,\\
&l_4(v) = \frac{\partial v}{\partial x_1}(\mathbf{a}^1)\,, && l_5(v) = \frac{\partial v}{\partial x_1}(\mathbf{a}^2)\,, && l_6(v) = \frac{\partial v}{\partial x_1}(\mathbf{a}^3)\,,\\
&l_7(v) = \frac{\partial v}{\partial x_2}(\mathbf{a}^1)\,, && l_8(v) = \frac{\partial v}{\partial x_2}(\mathbf{a}^2)\,, && l_9(v) = \frac{\partial v}{\partial x_2}(\mathbf{a}^3)\,,\\
&l_{10}(v) = v(\mathbf{a}^{123})\,.
\end{aligned}
$$

▼

Three d.o.f. associated with each vertex

Fewer global shape functions compared to previous choice!

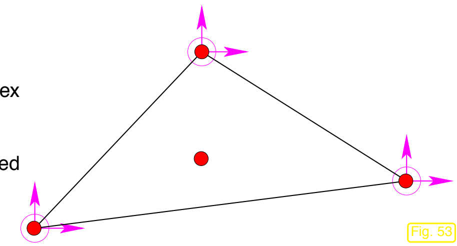


Fig. 53

3.2 Implementation

3.2.1 Mesh file format

Data flow in (most) finite element software packages:

Parameters

▼

Mesh generator ▶ Finite element solver (computational kernel) ▶ Post-processor (e.g. visualization)

Example 43 (Mesh file format (triangular mesh of polygonal domain)).

$$
\begin{array}{lll}
\multicolumn{3}{l}{\texttt{\# Two-dimensional simplicial mesh}} \\
1 \;\; \xi_1 \;\; \eta_1 & & \texttt{\# Coordinates of first node} \\
2 \;\; \xi_2 \;\; \eta_2 & & \texttt{\# Coordinates of second node} \\
\vdots & & \\
N \;\; \xi_N \;\; \eta_N & & \texttt{\# Coordinates of } N\texttt{-th node} \\
1 \;\; n_1^1 \;\; n_2^1 \;\; n_3^1 \;\; X_1 & & \texttt{\# Indices of nodes of first triangle} \\
2 \;\; n_1^2 \;\; n_2^2 \;\; n_3^2 \;\; X_2 & & \texttt{\# Indices of nodes of second triangle} \\
\vdots & & \\
M \;\; n_1^M \;\; n_2^M \;\; n_3^M \;\; X_M & & \texttt{\# Indices of nodes of } M\texttt{-th triangle}
\end{array}
\tag{3.2.1}
$$

$X_i, i = 1, \dots, M \longrightarrow$ extra information (e.g. material properties in triangle $\#i$).

Optional: additional information about edges (on $\partial\Omega$):

$$
\begin{array}{ll}
K \in \mathbb{N} & \texttt{\# Number of edges on } \partial\Omega \\
n_1^1 \, n_2^1 \;\; Y_1 & \texttt{\# Indices of endpoints of first edge} \\
n_1^2 \, n_2^2 \;\; Y_2 & \texttt{\# Indices of endpoints of second edge} \\
\vdots & \\
n_1^K \, n_2^K \;\; Y_K & \texttt{\# Indices of endpoints of } K\texttt{-th edge}
\end{array}
\tag{3.2.2}
$$

$Y_k, k = 1, \dots, K \longrightarrow$ extra information

Example 44 (Mesh file format for MATLAB code).

Vertex coordinate file:

```
% List of vertices
1 +0.000000e+00  -1.000000e+00
2 +1.000000e+00  +0.000000e+00
3 +0.000000e+00  +1.000000e+00
4 -1.000000e+00  +0.000000e+00
5 +0.000000e+00  +0.000000e+00
```

Cell information file:

```
% List of elements
1     1     2     5
2     2     3     5
3     3     4     5
4     4     1     5
```


Loading a mesh

```
m = load_Mesh('Coord_Circ.dat',...
              'Elem_Circ.dat');
plot_Mesh(m,'apts');
```

Option flags:

`'a'`: with axes

`'p'`: vertex labels on

`'t'`: cell labels on

`'s'`: caption/title on

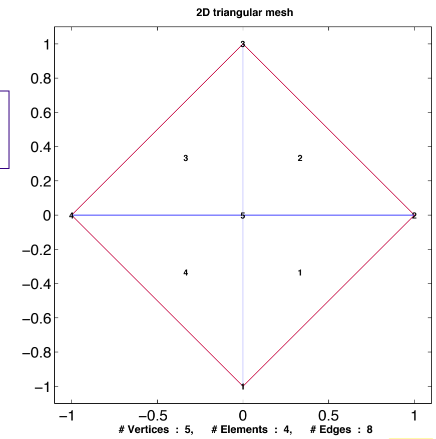


Fig. 54

How to create a mesh **?**

$\longrightarrow$ Mesh generation (beyond scope of this course)

$\longrightarrow$ `http://www.andrew.cmu.edu/user/sowen/mesh.html`

Free software:

- NETGEN (`http://www.hpfem.jku.at/netgen/`)
- Triangle (`http://www.cs.cmu.edu/~quake/triangle.html`)
- TETGEN (`http://tetgen.berlios.de`)

Example 45 (Mesh generation in MATLAB code).

Algorithm & details $\longrightarrow$ [?]

MATLAB-CODE: mesh generation for circular domain

```
BBOX = [-1 -1; 1 1];
H0 = 0.1;
DHD = @(x) sqrt(x(:,1).^2+x(:,2).^2)-1;
HHANDLE = @(x) ones(size(x,1),1);
Mesh = init_Mesh(BBOX,H0,DHD,...
                 HHANDLE,[],1);
save_Mesh(Mesh,'Coordinates.dat',...
          'Elements.dat');
```

Bounding box

Largest reasonable edge length

Signed distance function $\varphi(\mathbf{x})$: (distance from $\partial\Omega$, $\varphi(\mathbf{x}) < 0 \Leftrightarrow \mathbf{x} \in \Omega$)

Element size function (determines local edge length)

$\Diamond$

3.2.2 Assembly

$\longrightarrow$ term used for computing entries of stiffness matrix/load vector.

Discrete variational problem (V_N = FE space, $\dim V_N = N \in \mathbb{N}$, see Sect. 3.1.1)

$$u_N \in V_N\colon \quad a(u_N, v_N) = f(v_N) \quad \forall v_N \in V_N \,. \tag{3.1.1}$$

To be computed:

- Global stiffness matrix: $\mathbf{A} = \left(a(b_N^j, b_N^i)\right)_{i,j=1}^N \in \mathbb{R}^{N,N}$
- Global load vector: $\vec{\varphi} := \left(f(b_N^i)\right)_{i=1}^N \in \mathbb{R}^N$

both can be written in terms of local cell contributions:

$$a(u,v) = \sum_{K\in\mathcal{M}} a_K(u_{|K}, v_{|K}) \quad , \quad f(v) = \sum_{K\in\mathcal{M}} f_K(v_{|K}) \,. \tag{3.2.3}$$

Example: bilinear form/linear form arising from 2nd-order elliptic BVPs ($\to$ Sect. 2.5)

$$a(u,v) := \int_\Omega D\,\mathbf{grad}\,u \cdot \mathbf{grad}\,v \,\mathrm{d}\mathbf{x} = \sum_{K\in\mathcal{M}} \underbrace{\int_K D\,\mathbf{grad}\,u \cdot \mathbf{grad}\,v \,\mathrm{d}\mathbf{x}}_{=:a_K(u_{|K}, v_{|K})} \,,$$

$$f(v) := \int_\Omega f v \,\mathrm{d}\mathbf{x} = \sum_{K\in\mathcal{M}} \underbrace{\int_K f v \,\mathrm{d}\mathbf{x}}_{=:f_K(v_{|K})} \,.$$

Recall (3.1.7): Restrictions of global shape functions to cells = local shape functions

Definition 3.2.1. *Given local shape functions $\{b_1^K, \dots, b_Q^K\}$, we call*

element stiffness matrix $\mathbf{A}_K := \left(a_K(b_j^K, b_i^K)\right)_{i,j=1}^Q \in \mathbb{R}^{Q,Q}$,

element load vector $\vec{\varphi}_K := \left(f_K(b_i^K)\right)_{i=1}^Q \in \mathbb{R}^Q$.

Theorem 3.2.2. *The stiffness matrix and load vector can be obtained from their cell counterparts by*

$$\mathbf{A} = \sum_K \mathbf{T}_K^T \mathbf{A}_K \mathbf{T}_K \quad , \quad \vec{\varphi} = \sum_K \mathbf{T}_K^T \vec{\varphi}_K \,, \tag{3.2.4}$$

with the index mapping matrices ("T-matrices") $\mathbf{T}_K \in \mathbb{R}^{Q,N}$, defined by

$$(\mathbf{T}_K)_{ij} := \begin{cases} 1 & \text{, if } \ (b_N^j)_{|K} = b_i^K \,, \\ 0 & \text{, otherwise.} \end{cases} \qquad 1 \le i \le Q, 1 \le j \le N \,.$$

Proof.

$$(\mathbf{A})_{ij} = a(b_N^j, b_N^i) = \sum_{K\in\mathcal{M}} a_K(b^j_{N|K}, b^i_{N|K}) = \sum_{\substack{K\in\mathcal{M},\,\mathrm{supp}(b_N^j)\cap K\neq\emptyset,\\ \mathrm{supp}(b_N^i)\cap K\neq\emptyset}} a_K(b^K_{l(j)}, b^K_{l(i)}) = \sum_{\substack{K\in\mathcal{M},\,\mathrm{supp}(b_N^j)\cap K\neq\emptyset,\\ \mathrm{supp}(b_N^i)\cap K\neq\emptyset}} (\mathbf{A}_K)_{l(i),l(j)}$$

$l(i) \in \{1, \dots, k_K\}$, $1 \le i \le N \mathrel{\hat=}$ index of the local shape function corresponding to the global shape function b_N^i on K.

$$\Rightarrow \quad (\mathbf{A})_{ij} = \sum_{\substack{K\in\mathcal{M},\,\mathrm{supp}(b_N^j)\cap K\neq\emptyset,\\ \mathrm{supp}(b_N^i)\cap K\neq\emptyset}} \sum_{l=1}^Q \sum_{n=1}^Q (\mathbf{T}_K)_{li} (\mathbf{A}_K)_{ln} (\mathbf{T}_K)_{nj} \,. \qquad \square$$

Example 46 (Assembly for linear Lagrangian finite elements on triangular mesh).

Using the local/global numbering indicated beside

$$\to \mathbf{T}_{K^*} = \begin{pmatrix} 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0 \end{pmatrix}$$

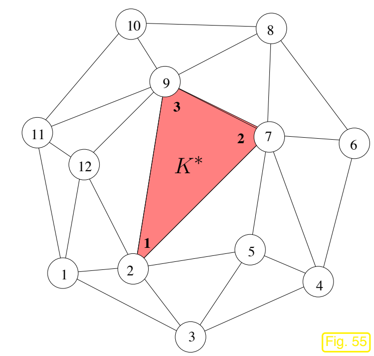


Fig. 55

Definition 3.2.3 (Abstract assembly operator)**.** *We denote the assembly operator in (3.2.4) symbolically by*

$$\mathbf{A} = \mathop{\mathcal{A}}_{K\in\mathcal{M}} \mathbf{A}_K, \quad \vec{\varphi} = \mathop{\mathcal{A}}_{K\in\mathcal{M}} \vec{\varphi}_K.$$

3.2.3 Mesh data structures

Issue: internal representation of mesh (→ Def. 3.1.10) in computer code

mesh data structure must provide:

1. offer unique identification of cells/(faces)/(edges)/vertices
2. represent mesh topology (= incidence relationships of cells/faces/edges/vertices)
3. describe mesh geometry (= location/shape of cells/faces/edges/vertices)
4. allow sequential access to edges/faces of a cell
   (→ traversal of local shape functions/degrees of freedom)
5. make possible traversal of cells of the mesh (→ global numbering)

Focus: **array oriented data layout** (→ MATLAB, FORTRAN)

Notation:
$\mathcal{M}$ = mesh (set of elements), $\mathcal{N}(\mathcal{M})$ = set of nodes (vertices) in $\mathcal{M}$, $\mathcal{E}(\mathcal{M})$ = set of edges in $\mathcal{M}$

Case: d-dimensional simplicial triangulation $\mathcal{M}$, *minimal* data structure (*cf.* Sect. 3.2.1)

→ Coordinates of vertices $\mathcal{N}(\mathcal{M})$: $\sharp\mathcal{N}(\mathcal{M}) \times d$-array `Coordinates` of `reals`

→ Vertex indices for cells: $\sharp\mathcal{M} \times (d+1)$-array `Elements` of `integers`.

➤ Already offers complete description of the mesh topology and geometry !

Optional extra information:

→ Edge connecting vertices: $\sharp\mathcal{N}(\mathcal{M}) \times \sharp\mathcal{N}(\mathcal{M})$ symmetric sparse integer matrix $I_{\mathcal{E}}$

$$(\mathbf{I}_{\mathcal{E}})_{ij} := \begin{cases} 0 & \text{, if vertex } \sharp i \text{ not linked to } \sharp j \\ e_{ij} & \text{, if edge connecting } \sharp i \text{ and } \sharp j \end{cases}$$

here e_{ij} is the unique edge number $\in \{1, 2, \ldots, \sharp\mathcal{E}(\mathcal{M})\}$

→ End points of the edges: $\sharp\mathcal{E}(\mathcal{M}) \times 2$ array of integer (= vertex indices of end points).

→ Cell adjacent to edges: $\sharp\mathcal{E}(\mathcal{M}) \times 2$ array of integers (=cell indices)
(one cell index =0 if edge is on $\partial\Omega$)

Note: Global shape functions associated with edges/faces ➤ extra information required !

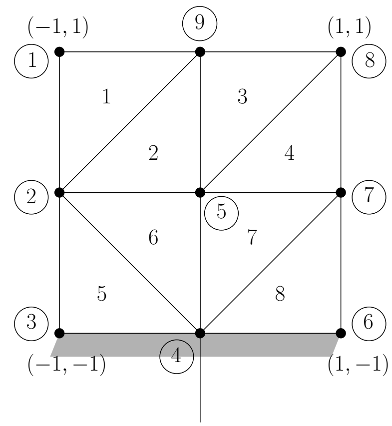


Fig. 56

Example 47 (Arrays storing 2D triangular mesh).

i	Coordinates	
1	-1	1
2	-1	0
3	-1	-1
4	0	-1
5	0	0
6	1	-1
7	1	0
8	1	1
9	0	1

Array `Coordinates`

K_j	Vertex indices		
1	1	2	9
2	2	5	9
3	5	8	9
4	5	7	8
5	3	4	2
6	4	5	2
7	4	7	5
8	4	6	7

Array `Elements`

Global shape functions associated with edges/faces ➤ extra information required !

Example 48 (Extended MATLAB mesh data structure).

```
mesh = add_Edge2Elem(add_Edges(init_Mesh(BBOX,H0,DHD,HHANDLE,[],1)))
```

(`init_Mesh` → Ex. 45)

mesh =
- Coordinates: [5x2 double] — vertex coordinates, see Ex. 44
- Elements: [4x3 double] — vertex indices of triangles, see Ex. 44
- Edges: [8x2 double] — indices of endpoints in `Coordinates` array
- Vert2Edge: [5x5 double] — $\sharp\mathcal{N}(\mathcal{M}) \times \sharp\mathcal{N}(\mathcal{M})$ *sparse* integer matrix: entry (i,j) = edge index, if $\neq 0$
- Edge2Elem: [8x2 double] — $\sharp\mathcal{E}(\mathcal{M}) \times 2$ integer array: indices of adjacent cells in `Elements` array
- EdgeLoc: [8x2 double] — $\sharp\mathcal{E}(\mathcal{M}) \times 2$ integer array: local indices of edges w.r.t. adjacent cells

Notation: $\mathcal{E}(\mathcal{M}) \hat{=}$ edges of 2D mesh

◇

How to number ↔ order local shape functions / global shape functions **?**

`Elements`, `Edges` arrays ➤ ordering of vertices of cells/endpoints of edges

Arrays (of vertices,cells,edges) ➤ array indices ➤ numbering of global shape functions

3.2.4 Algorithms

$$\boxed{\text{Cell oriented assembly} \quad \leftrightarrow \quad (3.2.4) \leftrightarrow \mathbf{A} = \mathcal{A}_{K\in\mathcal{M}}\mathbf{A}_K}$$

$$\Updownarrow$$

$$\mathbf{A} = \mathcal{A}_{K\in\mathcal{M}}\mathbf{A}_K := \left\{ \begin{array}{l} \textbf{foreach } K \in \mathcal{M} \textbf{ do} \\ \quad \text{local operations on } K \ (\rightarrow \mathbf{A}_K) \text{ and } \mathbf{A} = \mathbf{A} + \mathbf{T}_K^T \mathbf{A}_K \mathbf{T}_K \\ \textbf{enddo} \end{array} \right\}$$

Notion: local operations $\hat{=}$
- required only data from fixed "neighbourhood" of K
- computational effort "$O(1)$": independent of $\sharp\mathcal{M}$

Cell oriented assembly in MATLAB

```
function A = assemble(Mesh)

for k = Mesh.Elements'
  idx = ❶
  Aloc = ❷
  A(idx,idx) = A(idx,idx)+Aloc;
end
```

❶ row vector of index numbers of global shape functions $b_{i_1},\dots,b_{i_Q} \in V_N$ corresponding to local shape functions $b_1^K,\dots,b_Q^K$:

▶ $\texttt{idx} = (i_1,\dots,i_Q)$
(encodes index mapping matrix $\mathbf{T}_K$)

❷ $Q \times Q$ element stiffness matrix

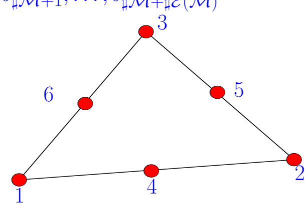
For Lagrangian FEM ($\rightarrow$ Sect. 3.1.5):

the total computational effort is of the order $O(\sharp\mathcal{M}) = O(N)$, $N := \dim V_N$.

Example 49 (Assembly for quadratic Lagrangian FE in MATLAB code).

Setting: FE space $\mathcal{S}_2^0(\mathcal{M})$ on triangular mesh $\mathcal{M}$ of polygon $\Omega \subset \mathbb{R}^2$
Recall: 6 local shape functions: 3 vertex-associated, 3 edge-associated $\rightarrow$ Ex. 40, Sect. 3.1.5
Convention: vertex-associated global shape functions $\rightarrow$ $b_1,\dots,b_{\sharp\mathcal{M}}$
edge-associated global shape functions $\rightarrow$ $b_{\sharp\mathcal{M}+1},\dots,b_{\sharp\mathcal{M}+\sharp\mathcal{E}(\mathcal{M})}$

Local numbering $\rightarrow$



MATLAB-CODE: assembly for quadratic Lagrangian FE

```
function A = assemMat_QFE(Mesh,EHandle,varargin)   ①

nV = size(Mesh.Coordinates,1);
nE = size(Mesh.Elements,1)

I ② = zeros(36*nE,1); J = I; a = I; offset = 0;
for k =1:nE
  vidx = Mesh.Elements(k,:)
  idx ③ = [vidx,...
          Mesh.Vert2Edge(vidx(1),vidx(2))+nV,...
          Mesh.Vert2Edge(vidx(2),vidx(3))+nV,...
          Mesh.Vert2Edge(vidx(3),vidx(1))+nV];
  Aloc ④ = transpose(EHandle(Mesh.Coordinates(vidx,:),...
               Mesh.⑤ElemFlag(k),varargin{:}));

  Qsq = prod(size(Aloc)); range = offset + 1:Qsq;
  t = idx(ones(length(idx),1),:)'; I(range) = t(:);
  t = idx(ones(1,length(idx)),:); J(range) = t(:);
  a(range) = Aloc(:);
  offset = offset + Qsq;
end
A ⑥ = sparse(I,J,a);
```


①: `EHandle` (function handle) $\rightarrow$ provides element stiffness matrix $\mathbf{A}_K \in \mathbb{R}^{6,6}$

②: `I,J,a` $\hat{=}$ linear arrays storing (i, j, a_{ij}) for stiffness matrix $\mathbf{A}$.
Initialized with 0 for the sake of efficiency $\rightarrow$ Ex. 50

③: `idx` $\hat{=}$ index mapping vector, see ❶ above

④: `Aloc` = $\mathbf{A}_K \in \mathbb{R}^{6,6}$ (element stiffness matrix)

⑤: `Mesh.ElemFlag(k)` marks groups of elements (e.g. to select local heat conductivity D in (2.5.4))

⑥: Build *sparse* MATLAB-matrix ($\rightarrow$ Def. 3.1.13) from index-entry arrays

◇

Example 50 (Efficient implementation of assembly).

`tic-toc`-timing (min of 4v runs), MATLAB V7, Intel Pentium 4 Mobile CPU 1.80GHz, Linux
Computation of element stiffness matrices skipped !

• *Sparse assembly*:

```
A(idx,idx) = A(idx,idx) + Aloc;
```

• *Array assembly I*: "growing arrays"

```
I = []; J = []; a = [];
...
t = idx(:,ones(length(idx),1))';
I = [I;t(:)];
t = idx(:,ones(1,length(idx)));
J = [J;t(:)];
a = [a; Aloc(:)];
```

• *Array assembly III*

$\longrightarrow$ see code fragment above

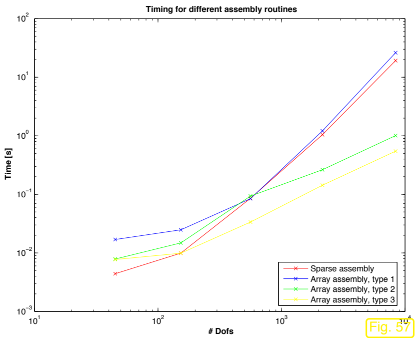


Fig. 57

$\diamond$

3.2.5 Local computations

First option: analytic evaluations

We discuss bilinear form related to $-\Delta$, triangular Lagrangian finite elements of degree p:

K triangle: $a_K(u,v) := \int_K \mathbf{grad}\, u \cdot \mathbf{grad}\, v \,\mathrm{d}\mathbf{x}$ ▶ element stiffness matrix .

Use barycentric coordinate representations of local shape functions

$$b_i^K = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^3, |\boldsymbol{\alpha}|\leq p} \kappa_{\boldsymbol{\alpha}}\, \lambda_1^{\alpha_1}\lambda_2^{\alpha_2}\lambda_3^{\alpha_3}\,, \quad \kappa_{\boldsymbol{\alpha}} \in \mathbb{R}\,, \tag{3.2.5}$$

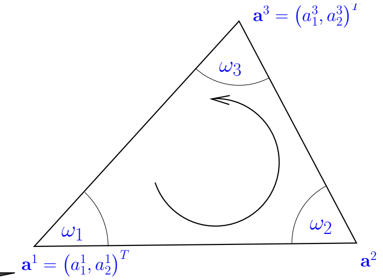
$$\Rightarrow \quad \mathbf{grad}\, b_i^K = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^3, |\boldsymbol{\alpha}|\leq p} \kappa_{\boldsymbol{\alpha}} \Big(\alpha_1\lambda_1^{\alpha_1-1}\lambda_2^{\alpha_2}\lambda_3^{\alpha_3}\,\mathbf{grad}\,\lambda_1 + \alpha_2\lambda_1^{\alpha_1}\lambda_2^{\alpha_2-1}\lambda_3^{\alpha_3}\,\mathbf{grad}\,\lambda_2 + \alpha_3\lambda_1^{\alpha_1}\lambda_2^{\alpha_2}\lambda_3^{\alpha_3-1}\,\mathbf{grad}\,\lambda_3 \Big)\,. \tag{3.2.6}$$

▼

to evaluate $\displaystyle\int_K \lambda_1^{\beta_1}\lambda_2^{\beta_2}\lambda_3^{\beta_3}\mathbf{grad}\,\lambda_i \cdot \mathbf{grad}\,\lambda_j \,\mathrm{d}\mathbf{x}\,, \quad i,j\in\{1,2,3\},\ \beta_k\in\mathbb{N}\,.$ (3.2.7)

If $\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3$ vertices of K (counterclockwise ordering):

$$\lambda_1(\mathbf{x}) = \frac{1}{2|K|}\left(\mathbf{x} - \begin{pmatrix} a_1^2 \\ a_2^2 \end{pmatrix}\right) \cdot \begin{pmatrix} a_2^2 - a_2^3 \\ a_1^3 - a_1^2 \end{pmatrix}\,,$$
$$\lambda_2(\mathbf{x}) = \frac{1}{2|K|}\left(\mathbf{x} - \begin{pmatrix} a_1^3 \\ a_2^3 \end{pmatrix}\right) \cdot \begin{pmatrix} a_2^3 - a_2^1 \\ a_1^1 - a_1^3 \end{pmatrix}\,,$$
$$\lambda_3(\mathbf{x}) = \frac{1}{2|K|}\left(\mathbf{x} - \begin{pmatrix} a_1^1 \\ a_2^1 \end{pmatrix}\right) \cdot \begin{pmatrix} a_2^1 - a_2^2 \\ a_1^2 - a_1^1 \end{pmatrix}\,.$$



▼

$$\mathbf{grad}\,\lambda_1 = \frac{1}{2|K|}\begin{pmatrix} a_2^2 - a_2^3 \\ a_1^3 - a_1^2 \end{pmatrix}\,,\quad \mathbf{grad}\,\lambda_2 = \frac{1}{2|K|}\begin{pmatrix} a_2^3 - a_2^1 \\ a_1^1 - a_1^3 \end{pmatrix}\,,\quad \mathbf{grad}\,\lambda_3 = \frac{1}{2|K|}\begin{pmatrix} a_2^1 - a_2^2 \\ a_1^2 - a_1^1 \end{pmatrix}\,. \tag{3.2.8}$$

▼

$$\left(\int_K \mathbf{grad}\,\lambda_i \cdot \mathbf{grad}\,\lambda_j \,\mathrm{d}\mathbf{x}\right)_{i,j=1}^3 = \frac{1}{2}\begin{pmatrix} \cot\omega_3 + \cot\omega_2 & -\cot\omega_3 & -\cot\omega_2 \\ -\cot\omega_3 & \cot\omega_3 + \cot\omega_1 & -\cot\omega_1 \\ -\cot\omega_2 & -\cot\omega_1 & \cot\omega_2 + \cot\omega_1 \end{pmatrix}\,. \tag{3.2.9}$$

$\rightarrow$ Exercise.

Lemma 3.2.4 (Integration of powers of barycentric coordinate functions). *For any non-degenerate d-simplex K and $\alpha_j \in \mathbb{N}$, $j = 1, \dots, d+1$,*

$$\int_K \lambda_1^{\alpha_1} \cdot \dots \cdot \lambda_{d+1}^{\alpha_{d+1}} \,\mathrm{d}\mathbf{x} = d!|K| \frac{\alpha_1!\alpha_2! \cdot \dots \cdot \alpha_{d+1}!}{(\alpha_1 + \alpha_2 + \dots + \alpha_{d+1} + d)!} \quad \forall \boldsymbol{\alpha} \in \mathbb{N}_0^{d+1}\,. \tag{3.2.10}$$

Remark. Alternative: symbolic computing (MAPLE, Mathematica) for local computations

3.2.6 Numerical quadrature

Second option (for local evaluations): Numerical quadrature

$$\int_\Omega f(\mathbf{x})\,\mathrm{d}\mathbf{x} \approx \sum_{K\in\mathcal{M}} |K| \sum_{l=1}^{P_K} \omega_l^K f(\boldsymbol{\pi}_l^K)\,, \quad \boldsymbol{\pi}_l^K \in K\,,\ \omega_l^K \in \mathbb{R}\,. \tag{3.2.11}$$

Terminology:

$\omega_l^K \rightarrow$ weights , $\boldsymbol{\pi}_l^K \rightarrow$ quadrature nodes

(3.2.11) = local quadrature rule

Mandatory
- for computation of load vector (f complicated/only available in procedural form)
- for computation of stiffness matrix, if $D = D(\mathbf{x})$ does not permit analytic integration.

Guideline: only quadrature rules with positive weights are numerically stable.

For affine equivalent finite elements (→ Sect. 3.1.6):

➤ Parametric definition of local quadrature rules on reference cell $\widehat{K}$:

$$\int_{\widehat{K}} f(\widehat{\mathbf{x}})\,\mathrm{d}\widehat{\mathbf{x}} \approx |\widehat{K}| \sum_{l=1}^{P} \widehat{\omega}_l f(\widehat{\boldsymbol{\pi}}_l) \quad \blacktriangleright \quad \int_\Omega f(\mathbf{x})\,\mathrm{d}\mathbf{x} \approx \sum_{K\in\mathcal{M}} |K| \sum_{l=1}^{P} \omega_l^K f(\boldsymbol{\pi}_l^K)$$

with $\omega_l^K = \widehat{\omega}_l,\ \boldsymbol{\pi}_l^K = \boldsymbol{\Phi}_K(\widehat{\boldsymbol{\pi}}_l)$.

How to gauge the quality of parametric local quadrature rules **?**

Quality of a parametric local quadrature rule on K $\sim$ largest space of polynomials on $\widehat{K}$ integrated exactly by the corresponding quadrature rule on $\widehat{K}$.

Parlance: Quadrature rule exact for $\mathcal{P}_p(\widehat{K})$ $\Rightarrow$ quadrature rule of order $p+1$
degree of exactness p

Example 51 (Local quadrature rules on triangles).

If K triangle $\Rightarrow$ $\widehat{K} := \operatorname{convex}\left\{\binom{0}{0}, \binom{1}{0}, \binom{0}{1}\right\}$.

Quadrature rules described by pairs $(\widehat{\omega}_1, \widehat{\boldsymbol{\pi}}_1), \ldots, (\widehat{\omega}_P, \widehat{\boldsymbol{\pi}}_P)$, $P \in \mathbb{N}$.

- Quadrature rule of order 2 (exact for $\mathcal{P}_1(\widehat{K})$)

$$\left\{\left(\tfrac{1}{3}, \begin{pmatrix}0\\0\end{pmatrix}\right), \left(\tfrac{1}{3}, \begin{pmatrix}0\\1\end{pmatrix}\right), \left(\tfrac{1}{3}, \begin{pmatrix}1\\0\end{pmatrix}\right)\right\}\,. \tag{3.2.12}$$

- Quadrature rule of order 3 (exact for $\mathcal{P}_2(\widehat{K})$)

$$\left\{\left(\tfrac{1}{3}, \begin{pmatrix}1/2\\0\end{pmatrix}\right), \left(\tfrac{1}{3}, \begin{pmatrix}0\\1/2\end{pmatrix}\right), \left(\tfrac{1}{3}, \begin{pmatrix}1/2\\1/2\end{pmatrix}\right)\right\}\,. \tag{3.2.13}$$

- One-point quadrature rule of order 2 (exact for $\mathcal{P}_1(\widehat{K})$)

$$\left\{\left(1, \begin{pmatrix}1/3\\1/3\end{pmatrix}\right)\right\}\,. \tag{3.2.14}$$

- Quadrature rule of order 6 (exact for $\mathcal{P}_5(\widehat{K})$)

$$\begin{aligned}\Bigg\{&\left(\tfrac{9}{40}, \begin{pmatrix}1/3\\1/3\end{pmatrix}\right), \left(\frac{155+\sqrt{15}}{1200}, \begin{pmatrix}6+\sqrt{15}/21\\6+\sqrt{15}/21\end{pmatrix}\right), \left(\frac{155+\sqrt{15}}{1200}, \begin{pmatrix}9-2\sqrt{15}/21\\6+\sqrt{15}/21\end{pmatrix}\right),\\ &\left(\frac{155+\sqrt{15}}{1200}, \begin{pmatrix}6+\sqrt{15}/21\\9-2\sqrt{15}/21\end{pmatrix}\right), \left(\frac{155-\sqrt{15}}{1200}, \begin{pmatrix}6-\sqrt{15}/21\\9+2\sqrt{15}/21\end{pmatrix}\right),\\ &\left(\frac{155-\sqrt{15}}{1200}, \begin{pmatrix}9+2\sqrt{15}/21\\6-\sqrt{15}/21\end{pmatrix}\right), \left(\frac{155-\sqrt{15}}{1200}, \begin{pmatrix}6-\sqrt{15}/21\\6-\sqrt{15}/21\end{pmatrix}\right)\Bigg\}\end{aligned} \tag{3.2.15}$$

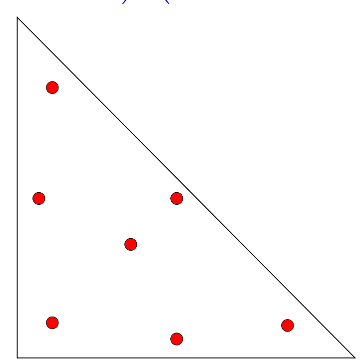


Fig. 58

In [**?**]: quadrature rules up to order $p = 21$ with $P \leq 1/6p(p+1)+5$

Example 52 (Local quadrature rules on quadrilaterals).

If K quadrilateral $\Rightarrow$ $\widehat{K} := \operatorname{convex}\left\{\binom{0}{0},\binom{1}{0},\binom{0}{1},\binom{1}{1}\right\}$ (unit square).

On $\widehat{K}$: tensor product construction:

If $\{(\omega_1,\pi_1),\dots,(\omega_P,\pi_P)\}$, $P \in \mathbb{N}$, quadrature rule on the interval $]0,1[$, exact for $\mathcal{P}_p]0,1[$, then

$$\left\{\begin{matrix} (\omega_1^2,\binom{\pi_1}{\pi_1}) & \cdots & (\omega_1\omega_P,\binom{\pi_1}{\pi_P}) \\ \vdots & & \vdots \\ (\omega_1\omega_P,\binom{\pi_P}{\pi_1}) & \cdots & (\omega_P^2,\binom{\pi_P}{\pi_P}) \end{matrix}\right\}$$

quadrature rule on $\widehat{K}$, exact for $\mathcal{Q}_p(\widehat{K})$.

Quadrature rules on $]0,1[$ ($\rightarrow$ basic numerics):

- classical Newton-Cotes formulas (equidistant quadrature nodes).
- Gauss-Legendre quadrature rules, exact for $\mathcal{P}_{2P}(]0,1[)$ using only P nodes.
- Gauss-Lobatto quadrature rules: P nodes including $\{0,1\}$, exact for $\mathcal{P}_{2P-1}(]0,1[)$.

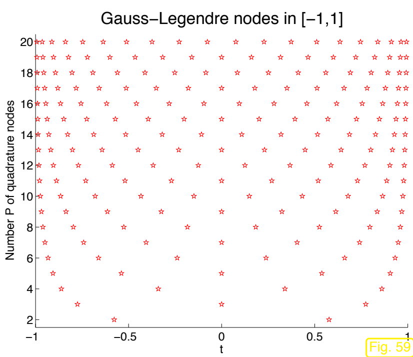


Fig. 59

3.2.7 Treatment of essential boundary conditions

Remember Sect. 2.7: extension $g \rightarrow \tilde{g}$ of Dirichlet data into Ω yielded linear variational problem.

Adaptation to finite element setting:

V_N = finite element space without constraints on $\partial\Omega$.

FIRST STEP: Interpolation/projection of boundary data

FE-space V_N $\Rightarrow$ $W_N := V_{N|\partial\Omega}$ (FE trace space)

Example: if $V_N = \mathcal{S}_1^0(\mathcal{M})$, then W_N = set of piecewise linear, continuous functions on boundary mesh $\mathcal{M}_{|\partial\Omega}$.

BUT, not necessarily $g \in W_N$!

▶ Replace g by (interpolant, least squares fit, etc.) $g_N \in W_N$

Example: if $V_N = \mathcal{S}_1^0(\mathcal{M})$ and $g \in C^0(\partial\Omega)$, then choose g_N as p.w. linear interpolant.

SECOND STEP: Trivial extension of $g_N \rightarrow \widetilde{g}_N \in V_N$

▶ Only nodal basis functions associated with node/edge/face $\subset \partial\Omega$ contribute to $\widetilde{g}_N$!

Example: if $V_N = \mathcal{S}_1^0(\mathcal{M})$, g_N p.w. linear continuous on $\mathcal{M}_{|\partial\Omega}$

▶ $\widetilde{g}_N = \sum_{\mathbf{p}\in\mathcal{N}(\mathcal{M}_{|\partial\Omega})} g_N(\mathbf{p})\, b_N^{\mathbf{p}}$, where $b_N^{\mathbf{p}} =$ "hat function" for node $\mathbf{p}$.

$$u_N \in V_{N,0}:\quad a(u_N + \widetilde{g}_N, v_N) = f(v_N) \quad \forall v_N \in V_{N,0}\ . \tag{3.2.16}$$

$V_{N,0} := \{v_N \in V_N\colon v_N = 0 \text{ on } \partial\Omega\}$ = span of "interior" basis functions.

$$\text{Elimination} \ \blacktriangleright \quad \mathbf{A}_{\Omega\Omega}\vec{\boldsymbol{\mu}}_\Omega = \vec{\boldsymbol{\varphi}}_\Omega - \mathbf{A}_{\Gamma\Omega}\vec{\boldsymbol{\mu}}_\Gamma\ . \quad \text{(cf. Sect. 2.7)} \tag{3.2.17}$$

Remark 53. Alternative: elimination on element level $\Rightarrow$ modified $\vec{\boldsymbol{\varphi}}_K$

$$\mathbf{A}_K = \begin{pmatrix}\mathbf{A}_{ii} & \mathbf{A}_{bi}\\ \mathbf{A}_{ib} & \mathbf{A}_{bb}\end{pmatrix}\ ,\quad \vec{\boldsymbol{\varphi}}_K = \begin{pmatrix}\vec{\boldsymbol{\varphi}}_i\\ \vec{\boldsymbol{\varphi}}_b\end{pmatrix} \ \blacktriangleright\ \widetilde{\mathbf{A}}_K = \mathbf{A}_{ii}\ ,\quad \widetilde{\vec{\boldsymbol{\varphi}}}_K = \vec{\boldsymbol{\varphi}}_i - \mathbf{A}_{bi}\vec{\boldsymbol{\mu}}_{\Gamma,K}\ .$$

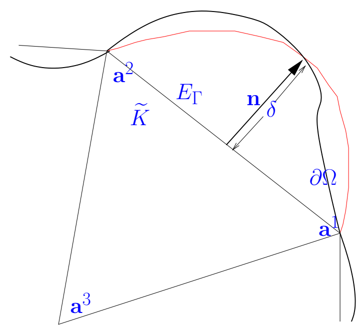
Then do assembly based on $\widetilde{\mathbf{A}}_K$ and $\widetilde{\vec{\boldsymbol{\varphi}}}_K$.

3.2.8 Boundary approximation

Sect 3.1.6 $\longrightarrow$ approximate treatment of curved $\partial\Omega$ by parametric FE:

Idea: Piecewise polynomial approximation of boundary (boundary fitting)
($\partial\Omega$ locally considered as function over straight edge of an element)

Example: Piecewise quadratic boundary approximation
(Part of $\partial\Omega$ between $\mathbf{a}^1$ and $\mathbf{a}^2$ approximated by parabola)



Mapping $\widetilde{K} \longrightarrow$ "curved element" K:

$$\mathbf{\Phi}(\widetilde{\mathbf{x}}) := \widetilde{\mathbf{x}} + 4\delta\,\lambda_1(\widetilde{\mathbf{x}})\lambda_2(\widetilde{\mathbf{x}})\,\mathbf{n}\ .$$

(λ_i barycentric coordinate functions on $\widetilde{K}$, $\mathbf{n}$ normal to E_Γ)

Note: Essential: $\mathbf{\Phi}$ diffeomorphism $\leftrightarrow$ δ sufficiently small

Transformation formula for gradients: for $u : K \mapsto \mathbb{R}$, diffeomorphism $\mathbf{\Phi} : \widetilde{K} \mapsto K$

$$(\mathbf{grad}_{\widetilde{\mathbf{x}}}(\mathbf{\Phi}^*u))(\widetilde{\mathbf{x}}) = (D\mathbf{\Phi}(\widetilde{\mathbf{x}}))^T\,(\mathbf{grad}_{\mathbf{x}}\,u)(\mathbf{\Phi}(\widetilde{\mathbf{x}}))\quad \forall \widetilde{\mathbf{x}} \in \widetilde{K}\ . \tag{3.2.18}$$

Proof: **chain rule**:

$$\frac{\partial u}{\partial x_i}(\mathbf{x}) = \frac{\partial}{\partial x_i}\mathbf{\Phi}^*u(\mathbf{\Phi}^{-1}(\mathbf{x})) = \sum_{j=1}^{d}\frac{\partial \mathbf{\Phi}^*u}{\partial \widetilde{x}_j}(\mathbf{\Phi}^{-1}(\mathbf{x}))\,\frac{\partial \mathbf{\Phi}_j^{-1}}{\partial x_i}(\mathbf{x})\ .$$

$$\blacktriangleright\quad \mathbf{grad}\,u(\mathbf{x}) = \left(D\mathbf{\Phi}^{-1}(\mathbf{x})\right)^T\mathbf{grad}_{\widetilde{\mathbf{x}}}(\mathbf{\Phi}^*u)(\mathbf{\Phi}^{-1}(\mathbf{x})) = D\mathbf{\Phi}(\mathbf{\Phi}^{-1}(\mathbf{x}))^{-T}(\mathbf{grad}\,\mathbf{\Phi}^*u)(\mathbf{\Phi}^{-1}(\mathbf{x}))\ .$$

Parametric construction:

$$\underset{\substack{\uparrow\\ \text{Local shape functions on } \widetilde{K}}}{b_i^{\widetilde{K}}} = \mathbf{\Phi}^*\underset{\substack{\uparrow\\ \text{Local shape functions on } K}}{b_i^{K}}\ ,\quad i = 1,\dots,Q$$

Local computations use (3.2.18) & transformation formula (for multidimensional integrals):

$$\int_K f(\mathbf{\Phi}(\mathbf{x}))\,\mathrm{d}\mathbf{x} = \int_{\widetilde{K}} f(\widetilde{\mathbf{x}})|\det D\mathbf{\Phi}(\widetilde{\mathbf{x}})|\,\mathrm{d}\widetilde{\mathbf{x}}\quad \text{for } f : K \mapsto \mathbb{R}\ ,$$

$$\int_K \mathbf{grad}\,u\cdot\mathbf{grad}\,v\,\mathrm{d}\mathbf{x} = \int_{\widetilde{K}}(\mathbf{grad}\,u)(\mathbf{\Phi}(\widetilde{\mathbf{x}}))\cdot(\mathbf{grad}\,v)(\mathbf{\Phi}(\widetilde{\mathbf{x}}))\,|\det D\mathbf{\Phi}(\widetilde{\mathbf{x}})|\,\mathrm{d}\widetilde{\mathbf{x}} = \int_{\widetilde{K}} D\mathbf{\Phi}^{-T}(\widetilde{\mathbf{x}})\,\mathbf{grad}_{\widetilde{\mathbf{x}}}(\mathbf{\Phi}^*u)\cdot D\mathbf{\Phi}^{-T}(\widetilde{\mathbf{x}})\,\mathbf{grad}_{\widetilde{\mathbf{x}}}(\mathbf{\Phi}^*v)\,|\det D\mathbf{\Phi}(\widetilde{\mathbf{x}})|\,\mathrm{d}\widetilde{\mathbf{x}}\ .$$

$$\int_K \mathbf{grad}\,b_i^K\cdot\mathbf{grad}\,b_j^K\,\mathrm{d}\mathbf{x} = \int_{\widetilde{K}}\left\{D\mathbf{\Phi}(\widetilde{\mathbf{x}})^T D\mathbf{\Phi}(\widetilde{\mathbf{x}})\right\}^{-1}\mathbf{grad}\,b_i^{\widetilde{K}}\cdot\mathbf{grad}\,b_j^{\widetilde{K}}\,|\det D\mathbf{\Phi}(\widetilde{\mathbf{x}})|\,\mathrm{d}\widetilde{\mathbf{x}}\ .$$

Note: local shape functions $b_i^{\widetilde{K}}$ simple polynomials !

For parabolic boundary fitting:

$$D\mathbf{\Phi} = Id + 4\delta\,\mathbf{n}\cdot\mathbf{grad}(\lambda_1\lambda_2)^T \in \mathbb{R}^{2,2}\quad,\quad \det(D\mathbf{\Phi}) = 1 + 4\delta\,\mathbf{n}\cdot\mathbf{grad}(\lambda_1\lambda_2)\ .$$

Next: numerical quadrature ($\longrightarrow$ Sect. 3.2.6) on $\widetilde{K}$

3.2.9 Static condensation

interior basis functions = global shape functions supported inside a cell

(occur for $\mathcal{S}_3^0(\mathcal{M})$ on triangular mesh $\mathcal{M}$ in 2D)

Sorting of global basis functions: coefficients for interior basis functions last

▶ Block structure of resulting linear system $\mathbf{A}\vec{\boldsymbol{\mu}} = \vec{\boldsymbol{\varphi}}$

$$\mathbf{A}\vec{\boldsymbol{\mu}} = \begin{pmatrix}\mathbf{A}_{oo} & \mathbf{A}_{oi}\\ \mathbf{A}_{io} & \mathbf{A}_{ii}\end{pmatrix}\begin{pmatrix}\vec{\boldsymbol{\mu}}_o\\ \vec{\boldsymbol{\mu}}_i\end{pmatrix} = \begin{pmatrix}\vec{\boldsymbol{\varphi}}_o\\ \vec{\boldsymbol{\varphi}}_i\end{pmatrix} = \vec{\boldsymbol{\varphi}}\ . \tag{3.2.19}$$

$\mathbf{A}_{ii} \longleftarrow$ coupling among interior basis functions
$\mathbf{A}_{oi} \longleftarrow$ coupling between interior b.f. & basis functions on nodes/edges

Note: $\mathbf{A}_{ii}$ is block-diagonal with small blocks ▶ "easy to invert"

[Elimination of $\vec{\boldsymbol{\mu}}_i$ (Static condensation)]

Schur complement system: $\left(\mathbf{A}_{oo} - \mathbf{A}_{oi}\mathbf{A}_{ii}^{-1}\mathbf{A}_{io}\right)\vec{\boldsymbol{\mu}}_o = \vec{\boldsymbol{\varphi}}_o - \mathbf{A}_{oi}\mathbf{A}_{ii}^{-1}\vec{\boldsymbol{\varphi}}_i\ .$