ETH Lecture 401-0663-00L Numerical Methods for PDEs

Mid-term Exam

Spring Term 2024

April 9, 2024, 16:15, HG G 19.1

Don't panic!

		%
Family Name		
First Name		
Department		
ETH Id Nr.		
Date	April 9, 2024	

Points:

	1a	1b	1c	1d	1e	1f	1g	Total
max	2	3	3	2	3	5	4	22
achvd								

(100% = 22 pts. , passed = 8 pts.)

General Instructions

- Upon entering the exam room take a seat at a desk on which you find an exam paper with this cover page!
- This is a **closed-book exam**, no aids are allowed.
- Keep only writing paraphernalia and your ETH ID card on the table.
- Turn off mobile phones, tablets, smartwatches, etc. and stow them away in your bag.
- When told to do so fill out the cover sheet first. Do not turn pages yet!
- Turn the cover sheet only when instructed to do so.
- You will be given **10 minutes of advance reading** time to familiarize yourself with the topic areas of the problems. When told, drop any pen! Then you may turn the pages and start reading the problems. You *must not write* anything during those 10 minutes.
- Start writing only when the start of the exam time proper is announced.
- Do not forget to write your name and ETH ID number on *every* page.
- **Write your answers in the appropriate (green) solution boxes on these problem sheets.**
- Wrong ticks in multiple-choice boxes can lead to points being subtracted. Hence, mere guessing is really dangerous! If you have no clue, leave all tickboxes empty.
- If you change your mind about an answer to a (multiple-choice) question, write a clear NO next to the old answer, draw fresh solution boxes/tickboxes and fill them.
- **Anything written outside the answer boxes will not be taken into account.**

- Do not write with red/green color or with pencil.
- Two blank pages are handed out with the exam: space for personal notes, not graded!
- **Duration: 30 minutes + 10 minutes advance reading time.**
- When the exam is over, drop your pens, tidily arrange your exam paper. Make sure that no page is missing.
- The exam proctors will collect the filled exam papers. Remain seated until they have finished.

Throughout the exam use the notations introduced in class:

- $(\mathbf{A})_{i,j}$ to refer to the entry of the matrix $\mathbf{A} \in \mathbb{K}^{m,n}$ at position (i,j).
- $(\mathbf{A})_{:,i}$ to designate the ith-column of the matrix $\mathbf{A}$,
- $(\mathbf{A})_{i,:}$ to denote the i-th row of the matrix $\mathbf{A}$,
- $(\mathbf{A})_{i:j,k:\ell}$ to single out the sub-matrix $\left[(\mathbf{A})_{r,s}\right]_{\substack{i\le r\le j\\ k\le s\le \ell}}$ of the matrix $\mathbf{A}$,
- $\vec{\mu}$, $\vec{\varphi}$, etc., for coefficient vectors,
- $(\vec{\mu})_k$ to reference the k-th entry of the vector $\vec{\mu}$,
- $\mathbf{e}_j \in \mathbb{R}^n$ to write the j-th Cartesian coordinate vector,
- $\mathbf{I}$ to denote the identity matrix,
- $\mathbf{O}$ to write a zero matrix,
- $\mathcal{P}_n$ for the space of (univariate polynomials of degree $\le n$),
- $\mathcal{S}^0_p(\mathcal{M})$ for degree-p Lagrange finite-element spaces on a mesh $\mathcal{M}$,
- $L^2(\Omega)$, $H^m(\Omega)$, for Sobolev spaces of functions on a domain Ω,
- $\|\cdot\|_X$ for the norm on a function space X,
- and superscript indices in brackets to denote iterates: $\mathbf{x}^{(k)}$, etc.

By default, vectors are regarded as column vectors.

Problem 0-1: Finite Element Galerkin Discretization of the Convection Bilnear Form

The problem examines the bilinear form occurring in the weak formulation of stationary convection boundary value problem, which will be treated in depth in [Lecture → Section 10.2]. Here, we study a straighforward finite element Galerkin discretization with a focus on element matrices and Galerkin matrices.

This problem is based on [Lecture → Section 2.7.4] and [Lecture → Section 3.8].
▷ problem name/problem code folder: ConvectionBilinearForm

Let $\Omega \subset \mathbb{R}^2$ a bounded, polygonal, computational domain. The convection bilinear form is

$$\mathsf{a}(u,v) = \int_\Omega \mathbf{a} \cdot \mathbf{grad}\, u(\boldsymbol{x})\, v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\;, \quad u \in H^1(\Omega),\ v \in L^2(\Omega)\;, \tag{0.1.1}$$

where $\mathbf{a} = [a_1, a_2]^\top \in \mathbb{R}^2$ is a given vector.

(0-1.a) (2 pts.) Which of the following estimates hold for the bilinear form a from (0.1.1) and with "generic constants" $C > 0$?

$\lvert\mathsf{a}(u,v)\rvert \le C\lVert u\rVert_{H^1(\Omega)}\,\lVert v\rVert_{H^1(\Omega)}\quad \forall u,v \in H^1(\Omega)\,,$	TRUE	○	FALSE	○	(0.1.2a)
$\lvert\mathsf{a}(u,v)\rvert \le C\lVert u\rVert_{L^2(\Omega)}\,\lVert v\rVert_{H^1(\Omega)}\quad \forall u,v \in H^1(\Omega)\,,$	TRUE	○	FALSE	○	(0.1.2b)
$\lvert\mathsf{a}(u,v)\rvert \le C\lVert u\rVert_{H^1(\Omega)}\,\lVert v\rVert_{L^2(\Omega)}\quad \forall u,v \in H^1(\Omega)\,,$	TRUE	○	FALSE	○	(0.1.2c)
$\lvert\mathsf{a}(u,v)\rvert \le C\lVert u\rVert_{L^2(\Omega)}\,\lVert v\rVert_{L^2(\Omega)}\quad \forall u,v \in H^1(\Omega)\,.$	TRUE	○	FALSE	○	(0.1.2d)

SOLUTION of (0-1.a):

❶ We show

$$\exists C > 0:\quad \lvert\mathsf{a}(u,v)\rvert \le C\lVert u\rVert_{H^1(\Omega)}\lVert v\rVert_{L^2(\Omega)}\quad \forall u \in H^1(\Omega),\ v \in L^2(\Omega)\,. \tag{0.1.3}$$

We rely on the Cauchy-Schwarz inequality (CSI) in $L^2(\Omega)$,

$$\left|\int_\Omega w(\boldsymbol{x})v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\right| \le \left(\int_\Omega |w(\boldsymbol{x})|^2\,\mathrm{d}\boldsymbol{x}\right)^{1/2}\left(\int_\Omega |v(\boldsymbol{x})|^2\,\mathrm{d}\boldsymbol{x}\right)^{1/2} = \lVert w\rVert_{L^2(\Omega)}\lVert v\rVert_{L^2(\Omega)}$$

[Lecture → Eq. (1.3.4.15)]

for all $w, v \in L^2(\Omega)$:

$$\begin{aligned}\lvert\mathsf{a}(u,v)\rvert &\le \int_\Omega |\mathbf{a}\cdot\mathbf{grad}\,u(\boldsymbol{x})|\,|v(\boldsymbol{x})|\,\mathrm{d}\boldsymbol{x} \le \int_\Omega \lVert\mathbf{a}\rVert\,\lVert\mathbf{grad}\,u(\boldsymbol{x})\rVert\,|v(\boldsymbol{x})|\,\mathrm{d}\boldsymbol{x}\\ &\overset{\text{(CSI)}}{\le} \lVert\mathbf{a}\rVert \int_\Omega \lVert\mathbf{grad}\,u(\boldsymbol{x})\rVert^2\,\mathrm{d}\boldsymbol{x}\cdot\int_\Omega |v(\boldsymbol{x})|^2\,\mathrm{d}\boldsymbol{x}\\ &= |\mathbf{grad}\,u|_{H^1(\Omega)}\,\lVert v\rVert_{L^2(\Omega)}\quad \forall u \in H^1(\Omega),\ v \in L^2(\Omega)\,.\end{aligned} \tag{0.1.4}$$

This is (0.1.3) with $C = 1$.

❷ Since $\lVert v\rVert_{L^2(\Omega)} \le \lVert v\rVert_{H^1(\Omega)}$ for all $v \in H^1(\Omega)$, it is clear that (0.1.3) also implies $\lvert\mathsf{a}(u,v)\rvert \le C\lVert u\rVert_{H^1(\Omega)}\,\lVert v\rVert_{H^1(\Omega)}$.

❸ Yet, we cannot bound $|\mathsf{a}(u,v)|$ in terms of the $L^2(\Omega)$-norm of u: Take $v(\boldsymbol{x}) = u(\boldsymbol{x}) = \sin(k\,\mathbf{a}\cdot\boldsymbol{x})$, $k \in \mathbb{N}$, which means

$$\int_\Omega \mathbf{a}\cdot\mathbf{grad}\,u(\boldsymbol{x})\,v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = k\,\|\mathbf{a}\|^2 \int_\Omega \sin^2(k\,\mathbf{a}\cdot\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \to \infty \quad \text{for} \quad k\to\infty\,,$$

whereas

$$\|\{\boldsymbol{x}\to\sin(k\,\mathbf{a}\cdot\boldsymbol{x})\}\|_{L^2(\Omega)} \leq \sqrt{|\Omega|} \quad \forall k\in\mathbb{N}\,.$$

Thus the right answers are

$\lvert\mathsf{a}(u,v)\rvert \leq C\|u\|_{H^1(\Omega)}\,\|v\|_{H^1(\Omega)} \quad \forall u,v\in H^1(\Omega)\,,$	**TRUE**	FALSE
$\lvert\mathsf{a}(u,v)\rvert \leq C\|u\|_{L^2(\Omega)}\,\|v\|_{H^1(\Omega)} \quad \forall u,v\in H^1(\Omega)\,,$	TRUE	**FALSE**
$\lvert\mathsf{a}(u,v)\rvert \leq C\|u\|_{H^1(\Omega)}\,\|v\|_{L^2(\Omega)} \quad \forall u,v\in H^1(\Omega)\,,$	**TRUE**	FALSE
$\lvert\mathsf{a}(u,v)\rvert \leq C\|u\|_{L^2(\Omega)}\,\|v\|_{L^2(\Omega)} \quad \forall u,v\in H^1(\Omega)\,.$	TRUE	**FALSE**

▲

(0-1.b) ⚃ (3 pts.) Assuming sufficient smoothness of u and v and applying Green's first formula

Theorem [Lecture → Thm. 1.5.2.7]. Green's first formula

For all vector fields $\mathbf{j}\in(C^1_{\mathrm{pw}}(\overline{\Omega}))^d$ *and functions* $v\in C^1_{\mathrm{pw}}(\overline{\Omega})$ *holds*

$$\int_\Omega \mathbf{j}\cdot\mathbf{grad}\,v\,\mathrm{d}\boldsymbol{x} = -\int_\Omega \operatorname{div}\mathbf{j}\,v\,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega}\mathbf{j}\cdot\mathbf{n}\,v\,\mathrm{d}S\,, \tag{0.1.5}$$

where $\boldsymbol{n}:\partial\omega\to\mathbb{R}^d$ *is the exterior unit normal vector field on* $\partial\Omega$.

we obtain an identity for a from (0.1.1):

$$\mathsf{a}(u,v) = -\int_\Omega \boxed{}\operatorname{div}\left(\boxed{}\right)(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega}\left(\boxed{}\right)u(\boldsymbol{x})v(\boldsymbol{x})\,\mathrm{d}S(\boldsymbol{x})\,.$$

Fill in the missing terms.

SOLUTION of (0-1.b):

We use (0.1.5) with $\mathbf{j}(\boldsymbol{x}) \leftarrow \mathbf{a}\,v(\boldsymbol{x})$ and $v\leftarrow u$. Making these substitutions immediately yields

$$\mathsf{a}(u,v) = -\int_\Omega u(\boldsymbol{x})\,\operatorname{div}(\mathbf{a}\,v)(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega}(\mathbf{a}\cdot\boldsymbol{n}(\boldsymbol{x}))u(\boldsymbol{x})v(\boldsymbol{x})\,\mathrm{d}S(\boldsymbol{x}) \tag{0.1.6}$$

Omitting the "$(\boldsymbol{x})$" is acceptable.

▲

We perform a Galerkin finite element discretization of a from (0.1.1) using a triangular mesh $\mathcal{M}$ of Ω and the following finite element spaces

- The first argument u is approximated in the lowest-order Lagrangian finite element space $\mathcal{S}^0_1(\mathcal{M})$ (piecewise linear C^0 finite-element functions).

- The second argument v we choose from the space $\mathcal{S}_0^{-1}(\mathcal{M})$ of *$\mathcal{M}$-piecewise constant* functions.

The following bases (sets of global shape functions) will be used:

- For $\mathcal{S}_1^0(\mathcal{M})$ we employ the usual nodal basis comprising tent functions associated with the nodes of the mesh.
- For $\mathcal{S}_0^{-1}(\mathcal{M})$ the basis functions are the **characteristic functions** of the mesh cells, that is, we use the basis $\{\chi_K\}_{K\in\mathcal{M}}$, with

$$\chi_K(\boldsymbol{x}) := \begin{cases} 1 & \text{, if } \quad x \in K\,, \\ 0 & \text{, if } \quad x \notin K\,, \end{cases} \qquad \boldsymbol{x} \in \Omega\,.$$

(0-1.c) (3 pts.) [depends on Sub-problem (0-1.a)]

Compute the entries of the $\mathcal{S}_1^0(\mathcal{M}) \times \mathcal{S}_0^{-1}(\mathcal{M})$ element matrix $\mathbf{A}_{\widehat{K}}$ for the bilinear form a and the "unit triangle" $\widehat{K} := \operatorname{convex}\left\{\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}\right\}$. Use this numbering of the vertices.

$$\mathbf{A}_{\widehat{K}} = \begin{bmatrix} \boxed{} & \boxed{} & \boxed{} \end{bmatrix} \in \mathbb{R}^{1,3}\,.$$

HINT 1 for (0-1.c): The barycentric coordinate functions for $\widehat{K}$ with vertices sorted as $\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}$ are

$$\lambda_1(\boldsymbol{x}) = 1 - x_1 - x_2 \quad , \quad \lambda_2(\boldsymbol{x}) = x_1 \quad , \quad \lambda_3(\boldsymbol{x}) = x_2\,, \quad \boldsymbol{x} = [x_1, x_2]^\top\,.$$

SOLUTION of (0-1.c):

The entries of the element matrix can be computed by plugging local shape functions into the localized bilinear form. For $\mathcal{S}_1^0(\mathcal{M})$ on a triangular mesh the local shape functions are the barycentric coordinate functions. Thus we arrive at

$$\mathbf{A}_{\widehat{K}} = \left[\int_{\widehat{K}} \mathbf{a} \cdot \mathbf{grad}\,\lambda_j(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\right]_{j=1,2,3} = \begin{bmatrix} -\frac{1}{2}(a_1 + a_2) & \frac{1}{2}a_1 & \frac{1}{2}a_2 \end{bmatrix}\,. \tag{0.1.7}$$

▲

(0-1.d) (2 pts.) In order to assemble the $\mathcal{S}_1^0(\mathcal{M}) \times \mathcal{S}_0^{-1}(\mathcal{M})$ Galerkin matrix for a in LEHRFEM++ on a 2D triangular mesh we need suitable **lf::assemble::UniformFEDofHandler** objects. Fill in the missing parts of the following code snippet in order to create those (`mesh_p` is a pointer to the underlying mesh):

1. For $\mathcal{S}_1^0(\mathcal{M})$:

```
lf::assemble::UniformFEDofHandler trial_dof_handler(
mesh_p, {{lf::base::RefEl::kPoint(), [      ] },
  {lf::base::RefEl::kSegment(), [      ] },
  {lf::base::RefEl::kTria(), [      ] },
  {lf::base::RefEl::kQuad(), [      ] }});
```

❷ For $\mathcal{S}_0^{-1}(\mathcal{M})$:

```
lf::assemble::UniformFEDofHandler test_dof_handler(
mesh_p, {{lf::base::RefEl::kPoint(), [    ] },
  {lf::base::RefEl::kSegment(), [    ] },
  {lf::base::RefEl::kTria(), [    ] },
  {lf::base::RefEl::kQuad(), [    ] }});
```

SOLUTION of (0-1.d):

❶ As we learned in [Lecture → § 2.7.4.16] the right **lf::assemble::UniformFEDofHandler** for $\mathcal{S}_1^0(\mathcal{M})$ is initialized with

```
lf::assemble::UniformFEDofHandler dof_handler(
mesh_p, {{lf::base::RefEl::kPoint(), 1},
  {lf::base::RefEl::kSegment(), 0},
  {lf::base::RefEl::kTria(), 0},
  {lf::base::RefEl::kQuad(), 0}});
```

❷ For $\mathcal{S}_0^{-1}(\mathcal{M})$ the global shape functions χ_K are associated with the cells of the mesh.

```
lf::assemble::UniformFEDofHandler dof_handler(
mesh_p, {{lf::base::RefEl::kPoint(), 0},
  {lf::base::RefEl::kSegment(), 0 },
  {lf::base::RefEl::kTria(), 1},
  {lf::base::RefEl::kQuad(), 1}});
```

Here you may also assign the number 0 as number of global shape functions on quadrilateral cells, because only triangular meshes are admitted.

▲

(0-1.e) ⚂ (3 pts.) We write $\mathbf{A}(\mathbf{a})$ for the $\mathcal{S}_1^0(\mathcal{M}) \times \mathcal{S}_0^{-1}(\mathcal{M})$ Galerkin matrix for a from (0.1.1) to indicate that it depends on the vector parameter $\mathbf{a} \in \mathbb{R}^2$.

Give a sharp upper bound of the number of non-zero elements of $\mathbf{A}(\mathbf{a})$ in terms of the numbers $\sharp\mathcal{M}$, $\sharp\mathcal{E}(\mathcal{M})$, and $\sharp\mathcal{V}(\mathcal{M})$ of cells, edges, and vertices of the triangular mesh $\mathcal{M}$.

$$\max_{\mathbf{a}\in\mathbb{R}^2} \mathrm{nnz}(\mathbf{A}(\mathbf{a})) = \boxed{}\,.$$

SOLUTION of (0-1.e):

Since the trial shape functions are the characteristic functions χ_K of mesh cells, there is one row of $\mathbf{A}(\mathbf{a})$ for every cell of the mesh. The support of χ_K overlaps with the supports of exactly three tent functions (global shape functions for $\mathcal{S}_1^0(\mathcal{M})$), which introduces at most three non-zero entries in each row of

$\mathbf{A}(\mathbf{a})$. These entries will be equal to the entries of the element matrix $\mathbf{A}_K$ for the cell corresponding to the row. For a "generic" vector $\mathbf{a}$ none of these entries will vanish, see (0.1.7).

$$\max_{\mathbf{a}\in\mathbb{R}^2} \operatorname{nnz}(\mathbf{A}(\mathbf{a})) = \boxed{3 \cdot \sharp\mathcal{M}} \; .$$

(0-1.f) (5 pts.) This class implements a **ENTITY_MATTRIX_PROVIDER** data type for the convection bilinear form a from (0.1.1), trial space $\mathcal{S}_1^0(\mathcal{M})$ and test space $\mathcal{S}_0^{-1}(\mathcal{M})$, $\mathcal{M}$ a triangular mesh.

C++ code 0.1.8: Element matrix provider type for convection bilinear form

```cpp
class ConvectionBilinearFormEMP final {
 public:
  using ElemMat = Eigen::Matrix<double, 1, 3>;
  explicit ConvectionBilinearFormEMP(Eigen::Vector2d a) : a_(std::move(a)) {}
  ConvectionBilinearFormEMP(const ConvectionBilinearFormEMP &) = delete;
  ConvectionBilinearFormEMP(ConvectionBilinearFormEMP &&) noexcept = default;
  ConvectionBilinearFormEMP &operator=(const ConvectionBilinearFormEMP &) =
      delete;
  ConvectionBilinearFormEMP &operator=(ConvectionBilinearFormEMP &&) = delete;
  ~ConvectionBilinearFormEMP() = default;
  [[nodiscard]] static bool isActive(const lf::mesh::Entity & /*cell*/) {
    return true;
  }
  [[nodiscard]] ElemMat Eval(const lf::mesh::Entity &cell) const;

 private:
  Eigen::Vector2d a_;  // transport direction
};
```

Fill in the blanks of the following listing with valid C++ code so that you complete a proper implementation of the `Eval()` member function of **ConvectionBilinearFormEMP**.

C++ code 0.1.9: `Eval()` method of element matrix provider type for convection bilinear form

```cpp
ConvectionBilinearFormEMP::ElemMat ConvectionBilinearFormEMP::Eval(
const lf::mesh::Entity &cell) const {
  const lf::geometry::Geometry *geo_p = cell.Geometry();
  // Reference coordinates of barycenter
  const Eigen::Vector2d ref_baryc{0.3, 0.3};
  const double det(std::abs((geo_p->IntegrationElement(ref_baryc))[0]));
  const Eigen::Matrix<double,2,2> JinvT(
  geo_p->JacobianInverseGramian(ref_baryc).block(0, 0, 2, 2));
  // Gradients of barycentric coordinate functions on reference
      element
  const Eigen::Matrix<double, 2, 3> refgrad{
    (Eigen::Matrix<double, 2, 3>() << -1.0, 1.0, 0.0, -1.0, 0.0, 1.0)
    .finished()};
  return                                  ;
}
```

HINT 1 for (0-1.f): Remember the transformation formula for gradients:

Lemma [Lecture → Lemma 2.8.3.10]. Transformation formula for gradients

For differentiable $u : K \mapsto \mathbb{R}$ *and any diffeomorphism* $\mathbf{\Phi} : \widehat{K} \mapsto K$ *we have*

$$(\mathbf{grad}_{\widehat{x}}(\mathbf{\Phi}^* u))(\widehat{x}) = (\mathrm{D}\mathbf{\Phi}(\widehat{x}))^{\top} \underbrace{(\mathbf{grad}_x\, u)(\mathbf{\Phi}(\widehat{x}))}_{=\mathbf{\Phi}^*(\mathbf{grad}\, u)(\widehat{x})} \quad \forall \widehat{x} \in \widehat{K}\,. \qquad \text{[Lecture} \to \text{Eq. (2.8.3.11)]}$$

SOLUTION of (0-1.f):

We recall the transformation to a reference element elaborated in [Lecture → Section 2.8.3]. Pikcing $K \in \mathcal{M}$ and writing b_K^j, $j = 1,2,3$, for the local shape functions of $\mathcal{S}_1^0(\mathcal{M})$ on K, the barycentric coordinate functions, we have

$$(\mathbf{A}_K)_{1,j} = \int_K \mathbf{a} \cdot \mathbf{grad}\, b_K^j(x)\,\mathrm{d}x\,, \quad j = 1,2,3\,,$$

because the local shape function for $\mathcal{S}_0^{-1}(\mathcal{M})$ on K attains the value $= 1$.

Let $\mathbf{\Phi}_K : \widehat{K} \to K$ stand for the unique affine mapping [Lecture → Lemma 2.7.5.14] taking the "unit triangle" $\widehat{K}$ to K. First we transform the integral,

$$\begin{aligned}\int_K \mathbf{a} \cdot \mathbf{grad}\, b_K^j(x)\,\mathrm{d}x &= \int_{\widehat{K}} \mathbf{a} \cdot \mathbf{grad}\, b_K^j(\mathbf{\Phi}_K(\widehat{x}))\,|\det \mathrm{D}\mathbf{\Phi}_K(\widehat{x})|\,\mathrm{d}\widehat{x} \\ &= \int_{\widehat{K}} \mathbf{a} \cdot \mathrm{D}\mathbf{\Phi}_K(\widehat{x})^{-\top}\,\mathbf{grad}\,\widehat{b}^j(\widehat{x})\,|\det \mathrm{D}\mathbf{\Phi}_K(\widehat{x})|\,\mathrm{d}x\,,\end{aligned}$$

where we used the transformation of gradients

Lemma [Lecture → Lemma 2.8.3.10]. Transformation formula for gradients

For differentiable $u : K \mapsto \mathbb{R}$ *and any diffeomorphism* $\mathbf{\Phi} : \widehat{K} \mapsto K$ *we have*

$$(\mathbf{grad}_{\widehat{x}}(\mathbf{\Phi}^* u))(\widehat{x}) = (\mathrm{D}\mathbf{\Phi}(\widehat{x}))^{\top} \underbrace{(\mathbf{grad}_x\, u)(\mathbf{\Phi}(\widehat{x}))}_{=\mathbf{\Phi}^*(\mathbf{grad}\, u)(\widehat{x})} \quad \forall \widehat{x} \in \widehat{K}\,. \qquad \text{[Lecture} \to \text{Eq. (2.8.3.11)]}$$

and the fact that $\mathcal{S}_1^0(\mathcal{M})$ is a parametric finite element space, that is,

$$\widehat{b}^j = \mathbf{\Phi}_K^* b_K^j\,, \quad \widehat{b}^j \;\hat{=}\; j\text{-th barycentric coordinate function on } \widehat{K}.$$

Note that $\widehat{x} \mapsto \mathrm{D}\mathbf{\Phi}_K(\widehat{x})$ is constant, and so is $\mathbf{grad}\,\widehat{b}^j$, which implies that

$$(\mathbf{A}_K)_{1,j} = \tfrac{1}{2}\mathbf{a} \cdot \mathrm{D}\mathbf{\Phi}_K(\widehat{c})^{-\top}\,\mathbf{grad}\,\widehat{b}^j(\widehat{c})\,|\det \mathrm{D}\mathbf{\Phi}_K(\widehat{c})|\,,$$

where $\widehat{c}$ is any point $\in \widehat{K}$, for instance the barycenter $\widehat{c} = \begin{bmatrix}\frac{1}{3}\\ \frac{1}{3}\end{bmatrix}$. In particular, we have

$$\mathbf{G} := \begin{bmatrix}\mathbf{grad}\,\widehat{b}^1 & \mathbf{grad}\,\widehat{b}^2 & \mathbf{grad}\,\widehat{b}^3\end{bmatrix} = \begin{bmatrix}-1 & 1 & 0\\ -1 & 0 & 1\end{bmatrix}.$$

Then final formula is

$$\mathbf{A}_K = \tfrac{1}{2}\mathbf{a}^{\top}\mathrm{D}\mathbf{\Phi}_K(\widehat{c})^{-\top}\mathbf{G}\,|\det \mathrm{D}\mathbf{\Phi}_K(\widehat{c})|\,. \tag{0.1.10}$$

In LEHRFEM++ we can obtain

- $\mathsf{D}\mathbf{\Phi}_K(\widehat{c})^{-\top}$ by calling the `JacobianInverseGramian()` method of the **Geometry** object associated with K [Lecture → § 2.8.3.14], and
- $\det \mathsf{D}\mathbf{\Phi}_K(\widehat{c})$ as the result of `Geometry::IntegrationElement()`.

C++ code 0.1.11: `Eval()` method of element matrix provider type for convection bilinear form

```cpp
ConvectionBilinearFormEMP::ElemMat ConvectionBilinearFormEMP::Eval(
    const lf::mesh::Entity &cell) const {
  LF_ASSERT_MSG_CONSTEXPR(cell.RefEl() == lf::base::RefEl::kTria(),
                          "Only triangles supported ");
  // Obtain associated geometry object
  const lf::geometry::Geometry *geo_p = cell.Geometry();
  // Reference coordinates of barycenter
  const Eigen::Vector2d ref_baryc{0.3, 0.3};
  // Obtain metric factor (Jacobian determinant)
  const double det(std::abs((geo_p->IntegrationElement(ref_baryc))[0]));
  // Fetch gradient transformation matrix
  const Eigen::Matrix<double, 2, 2> JinvT(
      geo_p->JacobianInverseGramian(ref_baryc).block(0, 0, 2, 2));
  // Gradients of barycentric coordinate functions on reference
    element
  const Eigen::Matrix<double, 2, 3> refgrad{
      (Eigen::Matrix<double, 2, 3>() << -1.0, 1.0, 0.0, -1.0, 0.0, 1.0)
          .finished()};
  // Transformed gradients
  const Eigen::Matrix<double, 2, 3> trf_grad = JinvT * refgrad;
  // Direction vector * trasnformed gradients
  const Eigen::Matrix<double, 1, 3> a_grad = a_.transpose() * trf_grad;
  // "Inegrate" constant over reference triangle, area = 0.5
  return 0.5 * a_grad * det;
}
```

▲

(0-1.g) ⚁ (4 pts.) Given a sequence $\mathcal{M}_\ell$, $\ell \in \mathbb{N}_0$, of triangular meshes of Ω generated by iterated uniform regular refinement we let

- $\mathsf{I}_\ell : C^0(\overline{\Omega}) \to \mathcal{S}_1^0(\mathcal{M})$ be the (nodal) linear interpolation operator according to

  Definition [Lecture → Def. 3.3.2.1]. Linear interpolation in 2D

  The linear interpolation operator $\mathsf{I}_1 : C^0(\bar{\Omega}) \mapsto \mathcal{S}_1^0(\mathcal{M})$ is defined by

  $$\mathsf{I}_1 u \in \mathcal{S}_1^0(\mathcal{M}) \quad , \quad \mathsf{I}_1 u(\boldsymbol{p}) = u(\boldsymbol{p}) \quad \forall \boldsymbol{p} \in \mathcal{V}(\mathcal{M}) \,.$$

- $\mathsf{Q}_\ell : L^2(\Omega) \to \mathcal{S}_0^{-1}(\mathcal{M})$ be the $L^2(\Omega)$-orthogonal projection onto $\mathcal{S}_0^{-1}(\mathcal{M}_\ell)$ defined as

  $$\mathsf{Q}_h w \in \mathcal{S}_0^{-1}(\mathcal{M}_\ell): \quad \int_\Omega (\mathsf{Q}_\ell w)(\boldsymbol{x}) v_h(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} = \int_\Omega w(\boldsymbol{x}) v_h(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} \tag{0.1.12}$$

  for all $v_h \in \mathcal{S}_0^{-1}(\mathcal{M}_\ell)$, $w \in L^2(\Omega)$.

Appealing to [Lecture → Cor. 3.3.3.4] and other results from finite-element theory, the following estimates hold for these operators, with generic constants $C > 0$ independent of ℓ,

$$\|u - \mathsf{I}_\ell u\|_{L^2(\Omega)} \leq C h_\ell^2 |u|_{H^2(\Omega)} \quad \forall u \in H^2(\Omega) \,, \tag{0.1.13}$$

$$|u - \mathsf{I}_\ell u|_{H^1(\Omega)} \leq C h_\ell |u|_{H^2(\Omega)} \quad \forall u \in H^2(\Omega)\,, \tag{0.1.14}$$

$$\|u - \mathsf{Q}_\ell u\|_{L^2(\Omega)} \leq C h_\ell \|u\|_{H^1(\Omega)} \quad \forall u \in H^1(\Omega)\,, \tag{0.1.15}$$

where h_ℓ designates the meshwidth of $\mathcal{M}_\ell$.

For given fixed $u, v \in C^\infty(\overline{\Omega})$ predict the asymptotic algebraic convergence of $|\mathsf{a}(\mathsf{I}_\ell u, \mathsf{Q}_\ell v) - \mathsf{a}(u, v)|$,

$$|\mathsf{a}(\mathsf{I}_\ell u, \mathsf{Q}_\ell v) - \mathsf{a}(u, v)| = O\Big(\boxed{}\Big) \quad \text{for} \quad h_\ell \to 0\,.$$

SOLUTION of (0-1.g):

We use the continuity properties of a found in Sub-problem (0-1.a), (0.1.3),

$$\exists C > 0\colon \quad |\mathsf{a}(u, v)| \leq C\|u\|_{H^1(\Omega)} \|v\|_{L^2(\Omega)} \quad \forall u \in H^1(\Omega),\ v \in L^2(\Omega)\,.$$

Exploiting this and the bilinearity of a we obtain

$$\begin{aligned}
|\mathsf{a}(\mathsf{I}_\ell u, \mathsf{Q}_\ell v) - \mathsf{a}(u, v)| &\leq |\mathsf{a}(\mathsf{I}_\ell u - u, \mathsf{Q}_\ell v) + \mathsf{a}(u, \mathsf{Q}_\ell v - v)| \\
&\leq |\mathsf{a}(\mathsf{I}_\ell u - u, \mathsf{Q}_\ell v)| + |\mathsf{a}(u, \mathsf{Q}_\ell v - v)| \\
&\leq C\Big(\|u - \mathsf{I}_\ell u\|_{H^1(\Omega)} \|\mathsf{Q}_\ell v\|_{L^2(\Omega)} + \|u\|_{H^1(\Omega)} \|v - \mathsf{Q}_\ell v\|_{L^2(\Omega)}\Big) \\
&\leq C\Big(h_\ell |u|_{H^2(\Omega)} \|v\|_{L^2(\Omega)} + \|u\|_{H^1(\Omega)} h_\ell \|v\|_{H^1(\Omega)}\Big) = O(h_\ell)
\end{aligned}$$

for $h_\ell \to 0$. The smoothness of both u and v permits us to estimate the projection errors by (0.1.14) and (0.1.15). In addition, we used that by the inverse $\triangle$-inequality

$$|\mathsf{I}_\ell u|_{H^1(\Omega)} \leq |u|_{H^1(\Omega)} + C h_\ell |u|_{H^2(\Omega)} \leq C\|u\|_{H^2(\Omega)}\,,$$

and that by the very definition of the $L^2(\Omega)$-projection Q_ℓ (set $v_h := \mathsf{Q}_\ell w$ in (0.1.12))

$$\|\mathsf{Q}_\ell w\|_{L^2(\Omega)} \leq \|w\|_{L^2(\Omega)} \quad \forall w \in L^2(\Omega)\,. \tag{0.1.16}$$

▲

End Problem 0-1 , 22 pts.