Numerical Approximation of PDEs

Spring Semester 2025

Lecturer: Prof. Annalisa Buffa Assistant: Mohamed Ben Abdelouahab

Session 2: March 6, 2025

Exercise 1. Let $I = [0, 1]$. Discuss whether the function

$$f : I \to \mathbb{R}, \quad x \mapsto x^{5/4}$$

is a member of $L^2(I)$ or $H^1(I)$. Is f' a member of $H^1(I)$?

Solution:
We compute the derivatives:

$$f'(x) = \frac{5}{4}x^{1/4}, \quad f''(x) = \frac{5}{16}x^{-3/4}.$$

We square them, which gives

$$f(x)^2 = x^{5/2}, \quad f'(x)^2 = \frac{25}{16}x^{1/2}, \quad f''(x)^2 = \frac{25}{256}x^{-3/2}.$$

Now we integrate and calculate

$$\int_I f(x)^2 dx = \frac{2}{7},$$
$$\int_I f'(x)^2 dx = \frac{25}{24}$$
$$\int_I f''(x)^2 dx = \frac{25}{128}x^{-1/2}\big|_0^1 = \infty.$$

We conclude that $f \in L^2(I)$ and $f' \in L^2(I)$, so that $f \in H^1(I)$. However, $f'' \notin L^2(I)$ and so $f' \notin H^1(I)$.

Exercise 2. Consider the open half unit ball in $\mathbb{R}^2$, denoted by $B = \{x \in \mathbb{R}^2 : |x| < 1\} \cap \mathbb{R}^+$. Show that the function $u(x) = |\log(|x|/2)|^\lambda$ belongs to the space $H^1(B)$ for $0 < \lambda < \frac{1}{2}$, but is not bounded in any neighborhood of the origin.
Hint: Use the polar coordinate transformation to compute the integrals, and note that $\nabla|x| = \frac{x}{|x|}$ for any $x \neq 0$.

NB. *This is to see that in higher dimensions $d \geq 2$, a member of $H^1(\Omega)$ is not necessarily continuous. As a result, we can only refer to the value of a function $u \in H^1(\Omega)$ "almost everywhere" in Ω rather than in the usual pointwise sense. Because of this, it's not immediately clear how to define the "value at the boundary", or restriction of u on $\partial\Omega$, since $\partial\Omega$ is negligible set with respect to the d-dimensional Lebesgue measure. However, there is a way to define the trace $u|_{\partial\Omega}$ of an $H^1(\Omega)$ function. This result is formally established by the trace theorem.*

Solution:
It's clear that u is not bounded in the neighborhood of the origin. However $u \in L^2(B)$ since:

$$\|u\|^2_{L^2(B)} = \int_B |u|^2 \, dx = \pi \int_0^1 |\log(r/2)|^{2\lambda} \, r dr < +\infty, \qquad \text{for} \quad 0 < 2\lambda < 1.$$

Using the identity $\nabla|x| = \frac{x}{|x|}$, we compute the classical derivative of u in $B\backslash\{0\}$:

$$\nabla u = \frac{\lambda x}{|x|^2} \left|\log\left(|x|/2\right)\right|^{\lambda-2} \log\left(|x|/2\right) = w.$$

We now prove that $w \in L^2(B)^2$:

$$\begin{aligned}
\int_B |w|^2 \, dx &= \lambda^2 \int_B \frac{1}{|x|^2} \left|\log\left(|x|/2\right)\right|^{2(\lambda-1)} \, dx = \lambda^2 \pi \int_0^1 \frac{\left|\log\left(r/2\right)\right|^{2(\lambda-1)}}{r} dr, \\
&= -\frac{\lambda^2 \pi}{2\lambda - 1} \left|\log\left(r/2\right)\right|^{2\lambda-1} \Big|_0^1 < \infty
\end{aligned}$$

Next, we prove that the weak derivative of u corresponds to w in B. For simplicity, we show that w is the weak derivative of u in the unit ball $B_0 = \{x \in \mathbb{R}^2 : |x| < 1\}$ and assume that $u \in H^1(B_0)$ and $w \in L^2(B_0)^2$ (can be justified by same computations above.).

Let $\phi \in C_c^\infty(B_0)$. For any $\varepsilon > 0$, using integration by parts, we write:

$$\begin{aligned}
\int_B u \nabla\phi \, dx &= \int_{\{\varepsilon<|x|<1\}} u\nabla\phi \, dx + \int_{\{|x|<\varepsilon\}} u\nabla\phi \, dx \\
&= -\int_{\{\varepsilon<|x|<1\}} w\phi + \int_{\{|x|=\varepsilon\}} u\phi \, ds + \int_{\{|x|<\varepsilon\}} u\nabla\phi \, dx, \\
&= -\int_{\{\varepsilon<|x|<1\}} w\phi + |\log(\varepsilon/2)|^\lambda \int_{\{|x|=\varepsilon\}} \phi \, ds + \int_{\{|x|<\varepsilon\}} u\nabla\phi \, dx
\end{aligned}$$

Now, we estimate the boundary term:

$$\begin{aligned}
|\log(\varepsilon/2)|^\lambda \int_{\{|x|=\varepsilon\}} \phi \, ds &\le |\log(\varepsilon/2)|^\lambda \int_0^{2\pi} \varepsilon \sup_{\{|x|=\varepsilon\}} |\phi(x)| \, d\theta \\
&= 2\pi\varepsilon \left|\log(\varepsilon/2)\right|^\lambda \sup_{\{|x|=\varepsilon\}} |\phi(x)| \longrightarrow_{\varepsilon\to 0} 0.
\end{aligned}$$

For the last integral, we use Cauchy-Schwarz and note tha $\nabla\phi$ is bounded,

$$\left|\int_{\{|x|<\varepsilon\}} u\nabla\phi \, dx\right| \le \|u\|^2_{L^2(B_0)} \left|\int_{\{|x|<\varepsilon\}} |\nabla\phi|^2 \, dx\right| \longrightarrow_{\varepsilon\to 0} 0.$$

Thus, we obtain by taking the limit:

$$\int_{B_0} u\nabla\phi \, dx = -\int_{B_0} w\phi \, dx, \qquad \forall \phi \in C_c^\infty(B_0).$$

In particular since $u \in H^1(B)$:

$$\int_B u\nabla\phi \, dx = -\int_B w\phi \, dx, \qquad \forall \phi \in C_c^\infty(B).$$

Exercise 3. [Poisson equation with mixed boundary conditions]
Let $\Omega \subset \mathbb{R}^d$ ($d = 2, 3$) be a bounded Lipschitz domain whose boundary $\partial\Omega$ can be split into two (essentially) disjoint parts: $\partial\Omega = \Gamma_D \cup \Gamma_N$.
Let $f \in L^2(\Omega)$ and $g_N \in L^2(\Gamma_N)$. Let $g_D \in H^{1/2}(\Gamma_D)$, so that there exists $G \in H^1(\Omega)$ satisfying $\gamma|_{\Gamma_D}(G) = g_D$. In other words, the trace of G on Γ_D is g_D.
We consider the Poisson equation

$$\begin{aligned} -\Delta u(x) &= f(x) && \text{in } \Omega, \\ u(x) &= g_D(x) && \text{on } \Gamma_D, \\ \partial_n u(x) &= g_N(x) && \text{on } \Gamma_N, \end{aligned}$$

Define the sets

$$V_{g_D} = \left\{ v \in H^1(\Omega) : \gamma|_{\Gamma_D}(v) = g_D \right\},$$
$$V_0 = H^1_{0,\Gamma_D}(\Omega) = \left\{ v \in H^1(\Omega) : \gamma|_{\Gamma_D}(v) = 0 \right\}$$

1. Suppose that u solves the Poisson problem. Show that $u_0 = u - G$ belongs to V_0 and satisfies an equation of the form

$$a(u_0, v) = F(v), \quad \forall v \in V_0. \tag{1}$$

   Give the explicit expressions of a and F.

   Hint: Multiply by a test function in V_0 and perform integration by parts.

2. Show that the conditions of the Lax-Milgram lemma are satisfied and use it to show that Problem (1) is well-posed in V_0, i.e., there exists a unique $u_0 \in V_0$ such that

$$a(u_0, v) = F(v), \quad \forall v \in V_0.$$

   Hint: The Poincaré inequality holds also in V_0 : $\forall v \in V_0, \|v\|_{L^2(\Omega)} \leq C_F \|\nabla v\|_{L^2(\Omega)}$.

3. Show there exists a unique $u \in V_{g_D}$ that is the solution of the original weak formulation.

4. Explain why we cannot apply the Lax-Milgram lemma directly to the set V_{g_D}.

Solution:
We first derive a weak formulation from the strong formulation. We multiply by $v \in V_0$, integrate over Ω, and performing integration by parts. Thus

$$\begin{aligned} \int_\Omega fv = -\int_\Omega (\Delta u)v &= -\int_\Omega (\mathrm{div}\nabla u)v \\ &= -\int_\Omega \mathrm{div}(\nabla u \cdot v) + \int_\Omega \nabla u \cdot \nabla v \\ &= -\int_{\partial\Omega} v \cdot (\nabla u \cdot \vec{n}) + \int_\Omega \nabla u \cdot \nabla v \\ &= -\oint_{\partial\Omega} v \cdot \partial_n u + \int_\Omega \nabla u \cdot \nabla v = -\oint_{\Gamma_N} v g_N + \int_\Omega \nabla u \cdot \nabla v. \end{aligned}$$

In other words,

$$\int_\Omega \nabla u \cdot \nabla v = \int_\Omega fv + \int_{\Gamma_N} g_N v, \quad \forall v \in V_0. \tag{2}$$

We have that $u_0 \in V_0$ since by construction

$$\gamma_{\Gamma_D} u_0 = \gamma_{\Gamma_D} u - \gamma_{\Gamma_D} G = g_D - g_D = 0.$$

Now, if we plug $u = u_0 + G$ into Problem (2), we obtain

$$\int_\Omega \nabla u \cdot \nabla v = \int_\Omega \nabla u_0 \cdot \nabla v + \int_\Omega \nabla G \cdot \nabla v$$

$$\int_\Omega \nabla u_0 \cdot \nabla v = \int_\Omega f v + \int_{\Gamma_N} g_N v - \int_\Omega \nabla G \cdot \nabla v, \quad \forall v \in V_0.$$

Then we can define the bilinear form $a : V_0 \times V_0 \to \mathbb{R}$ and the linear form $F : V_0 \to \mathbb{R}$ by

$$a(w, v) = \int_\Omega \nabla w \cdot \nabla v, \quad F(v) = \int_\Omega f v + \int_{\Gamma_N} g_N v - \int_\Omega \nabla G \cdot \nabla v.$$

Next, we verify the assumptions of Lax-Milgram lemma:

1. *The space V_0 equipped with the semi-norm $\|\nabla \cdot\|_{L^2(\Omega)}$ is an Hilbert space since V_0 is a subspace of $H^1(\Omega)$ and the norm $\|\nabla \cdot\|_{L^2(\Omega)}$ is equivalent to the H^1 norm thanks to the Friedrichs' inequality.*

2. *a is clearly bilinear, continuous and coercive (see lecture).*

3. *F is linear. Continuity can be proven in the following way :*

$$\begin{aligned} |F(v)| &\leq \left| \int_\Omega f v \right| + \left| \int_{\Gamma_N} g_N v \right| + \left| \int_\Omega \nabla G \cdot \nabla v \right| \\ &\leq \|f\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} + \|g_N\|_{L^2(\Gamma_N)} \|v\|_{L^2(\Gamma_N)} + \|\nabla G\|_{L^2(\Omega)} \|\nabla v\|_{L^2(\Omega)} \\ &\leq \|f\|_{L^2(\Omega)} \|v\|_{H^1(\Omega)} + C_\gamma \|g_N\|_{L^2(\Gamma_N)} \|v\|_{H^1(\Gamma_N)} + \|\nabla G\|_{L^2(\Omega)} \|v\|_{H^1(\Omega)}, \end{aligned}$$

   where we use Cauchy-Schwarz inequality, the trace inequality, and the properties of G. We give more details for the second term. We have

$$\|g_N\|_{L^2(\Gamma_N)} \|v\|_{L^2(\Gamma_N)} \leq \|g_N\|_{L^2(\Gamma_N)} \|v\|_{L^2(\partial\Omega)}.$$

   By the trace inequality applied to $\|v\|_{L^2(\partial\Omega)}$, we have

$$\|g_N\|_{L^2(\Gamma_N)} \|v\|_{L^2(\Gamma_N)} \leq \|g_N\|_{L^2(\Gamma_N)} C_T \|v\|_{H^1(\Omega)}. \tag{3}$$

We set $u = u_0 + G$. By construction, $u \in V_{g_D}$ and satisfies (2). The uniqueness is a consequence of the uniqueness of $u_0 \in V_0$ that is solution to (1). Indeed if u_1, u_2 are two distinct solutions to (2), then $u_1 - G \in V_0$ and $u_2 - G \in V_0$ are two distinct solutions to (1), that is a contradiction.
We cannot directly apply the Lax-Milgram lemma to the set V_{g_D} because, for example, it is generally not a linear subspace. Unless $g_D = 0$, the sum of two members of V_{g_D} is not in V_{g_D} again.

Exercise 4. [Equivalence of hat functions with the space of piecewise linears]
Denote the discretization parameter by $h = \frac{1}{N}$, where $N \in \mathbb{N}^*$ and consider a uniform subdivision $\mathcal{E}_h$ of $[a, b]$: $\mathcal{E}_h = \{x_0 = a, x_1, x_2, \ldots, x_{N-1}, x_N = b\}$ such that $h = x_{i+1} - x_i$.

Consider the space

$$V_h := \left\{ v \in C^0(\Omega) : \quad v(a) = v(b) = 0 \quad \text{and} \quad v|_{I_i} \in \mathbb{P}_1 \right\},$$

where $I_i = [x_i, x_{i+1}]$ are the subintervals forming the partition of $[a, b]$ with N elements and $\mathbb{P}_1 = \{p \,|\, p(x) = ax + b,\ (a, b) \in \mathbb{R}\}$ is the space of linear polynomials.
Next, consider the space

$$W_h = \operatorname{span} \{\lambda_1, \ldots, \lambda_{N-1}\},$$

where the λ_i are the hat functions (see figure 1) defined by:

$$\forall i \in [0 \ldots N]\ \forall x_j \in \mathcal{E}_h \quad \lambda_i \in V_h \quad \text{and} \quad \lambda_i(x_j) = \delta_{ij}.$$

Prove that $V_h = W_h$.

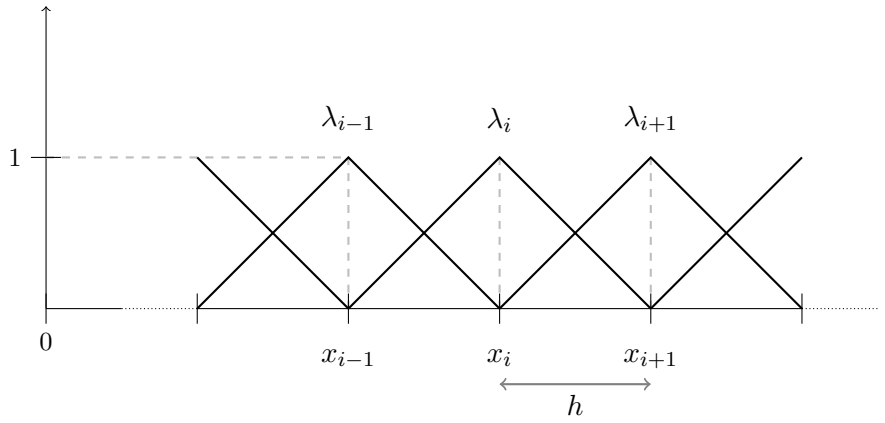

Figure 1: Basis hat functions λ_i.

Solution:
Suppose $v_h \in V_h$ with $v_h(x_i) = c_i$. Take the function $w_h = \sum_i c_i \lambda_i$. Clearly, $w_h \in W_h$. We have

$$(v_h - w_h)(x_i) = 0, \quad \forall x_i.$$

However, since both V_h and W_h are piecewise linear spaces, this is only possible if $v_h - w_h = 0$, $\implies$ $w_h = v_h$ and therefore $v_h \in W_h$.
Conversely, let $w_h = \sum_i c_i \lambda_i$. Then $w_h \in V_h$ trivially because each $\lambda_i \in V_h$.

Exercise 5. [Assembly of the mass matrix with nonconstant reaction term in $1D$]
Let $\Omega = (0, 1)$, we are considering the so-called mass matrix with a nonconstant reactivity $C^\infty(\Omega) \ni c(x) > 0$ which has the following entries:

$$M_{i,j} = \int_\Omega c(x) \lambda_i(x) \lambda_j(x) \mathrm{d}x. \tag{4}$$

The assembly iterates over all elements and then looks up which functions are nonzero on the element. The integral from (4) is split into its contributions from each element and then added to a sparse matrix at the right position.
Since $c(x)$ can be anything, the only way of computing the entries numerically is using a quadrature formula over the element (x_i, x_{i+1}). Use the provided script as a starting point to implement the assembly of this matrix for general $c(x)$ and then assemble M for $c(x) = 1 + \frac{1}{2}\sin(\pi x)$. The script provides a function for acquiring various Gauss quadrature formulas and for the midpoint rule.
YOU DO NOT HAVE TO USE NUMPY VECTORISATION YET

Listing 1: Python code

Solution:

```
#!/usr/bin/python3

"""
@author: Jochen Hinz
"""

import numpy as np
from scipy import sparse
from numbers import Number
from typing import Callable, Tuple

from functools import partial

def gauss_quadrature(a: Number, b: Number, order: int = 3) ->
    Tuple[np.ndarray, np.ndarray]:
  """ Given the element boundaries `(a, b)`, return the weights and
      evaluation points
      corresponding to a gaussian quadrature scheme of order `order`.

      Parameters
      ----------

      a : `float`
        the left boundary of the element
      b : `float`
        the right boundary of the element
      order : `int`
        the order of the Gaussian quadrature scheme

      Returns
      -------

      weights : `np.ndarray`
        the weights of the quadrature scheme
      points : `np.ndarray`
        the points (abscissae) over (a, b)
  """
  assert b > a
  points, weights = np.polynomial.legendre.leggauss(order)
  points = (points + 1) / 2
  return (b - a) / 2 * weights, a + points * (b - a)
```

```
gauss1 = partial(gauss_quadrature, order=1)
gauss2 = partial(gauss_quadrature, order=2)
gauss3 = partial(gauss_quadrature, order=3)
gauss4 = partial(gauss_quadrature, order=4)

# and so on ...

def midpoint_rule(a: Number, b: Number) -> Tuple[np.ndarray, np.ndarray]:
  """ Same as `gauss_quadrature` but without the `order` argument. """
  assert b > a
  return np.array([(b - a)]), np.array([a + b]) / 2

def assemble(nelems: int, c: Callable = None, quadrule: Callable = gauss3) ->
    sparse.csr_matrix:
  """
    Given the number of elements, the reactivity c and the quadrule, assemble
    M_{i,j} with M_{i,j} = \int_\Omega c(x)\lambda_i(x)\lambda_j(x)dx.

    >>> c = lambda x: 1 + x ** 2
    >>> quadrule = gauss3
    >>> M = assemble(10, c=c, quadrule=gauss3)
  """

  # if the reaction term is not passed, take it constant one
  if c is None:
    c = lambda x: np.ones_like(x)

  # the mesh's nodes are uniformly distributed over [0, 1]
  nodes = np.linspace(0, 1, nelems + 1)

  # the left and right boundaries of the i-th element are given by
     elem_boundaries[i]
  elem_boundaries = np.stack([nodes[:-1], nodes[1:]], axis=1)

  # the number of hat functions equals len(nodes)
  ndofs = len(nodes)

  # make an empty sparse matrix of shape (ndofs, ndofs) in lil-format
  # the lil-format can be directly assigned to

  M = sparse.lil_matrix((ndofs, ndofs))

  # iterate in parallel over the index of the element, the element boundaries
     as well as the weights and the points on each element
  for ielem, ((a, b), (weights, points)) in enumerate(zip(elem_boundaries,
     map(quadrule, *elem_boundaries.T))):

    # the active dofs on the i-th element are simply given by (i, i+1)
    dofs = np.array([ielem, ielem+1])

    # create an empty matrix of shape (2, 2)
    # where m_loc[i, j] contains the integral of c lambda_{dofs[i]}
       lambda_{dofs[j]} over the i-th element
    m_loc = np.empty((2, 2), dtype=float)

    # fill the local matrix of shape (2, 2)
    lambda0, lambda1 = 1 - (points - a) / (b - a), (points - a) / (b - a)
```

```
      for i, lam0 in enumerate([lambda0, lambda1]):
        for j, lam1 in enumerate([lambda0, lambda1]):
          m_loc[i, j] = (weights * c(points) * lam0 * lam1).sum()

      # add the contribution to the correct position in M
      # np.ix_(dofs, dofs) returns the entries we need.
      M[np.ix_(dofs, dofs)] += m_loc

    # return the M matrix in csr format
    return M.tocsr()

def main():
    nelems = 10

    c = lambda x: 1 + .5 * np.sin(np.pi * x)

    # generate three matrices with different quadratule rules
    M3 = assemble(nelems, c=c, quadrule=gauss3)
    M4 = assemble(nelems, c=c, quadrule=gauss4)
    Mmidpoint = assemble(nelems, c=c, quadrule=midpoint_rule)

    # print the outcomes:

    for i, mat in enumerate((M3, M4, Mmidpoint), 1):
      print(f"Matrix number {i}: \n", mat.todense(), '\n')

if __name__ == '__main__':
    main()
```