Numerical Approximation of PDEs

Spring Semester 2025

Lecturer: Prof. Annalisa Buffa   Assistant: Mohamed Ben Abdelouahab

Session 3: March 13, 2025

Exercise 1. [The reference element] Let K be a positively-oriented triangle. We consider a reference transformation $F_K : \hat{K} \to K$ with B_K as defined in the lecture.

(1a) Show that $\det B_K$ relates to the surface area $|K|$ of the triangle K as follows:

$$\det B_K = 2|K|.$$

(Note that there is a mistake in the lecture notes, erroneously claiming that $\det B_K = \frac{1}{2}|K|$*).*

(1b) Proof that the following two estimates hold

$$\|B_K\| \leq \frac{h_K}{\hat{\rho}}, \quad \|B_K^{-1}\| \leq \frac{\hat{h}}{\rho_K}, \tag{1}$$

where $\hat{\rho}$ and $\hat{h}$ are the inner and outer diameters of the reference triangle $\hat{K}$ while ρ_K and h_K denote the inner and outer diameters of the triangle K.

Deduce that there exists $C_0, C_1 > 0$ independent of K such that :

$$C_0 \rho_K \leq \|B_K\| \leq C_1 h_K, \quad \text{and} \quad C_2 \frac{\rho_K}{h_k} \leq \|B_K\| \|B_K^{-1}\| \leq C_3 \frac{h_K}{\rho_K}. \tag{2}$$

(1c) The quantity h_K/ρ_K is often called aspect ratio in the literature.[1] Someone proposes instead to measure the quality of a triangle by the ratio of the longest edge and the shortest edge. Is that a good idea?

Solution:
(1a) *The matrix* B_K *has columns* $\mathbf{v}_0 := \mathbf{b} - \mathbf{a}$ *and* $\mathbf{v}_1 := \mathbf{c} - \mathbf{a}$*, where* $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\} \subset \mathbb{R}^2$ *denote the vertices of* K *in counter-clockwise ordering.*
A basic result from geometry is that the surface area of the parallelogram P *with edges represented by the vectors* $\mathbf{v}_0$ *and* $\mathbf{v}_1$ *is given by*

$$|P| = \|\mathbf{v}_0\| \|\mathbf{v}_1\| |\sin(\phi)|,$$

with ϕ *the angle between* $\mathbf{v}_0$ *and* $\mathbf{v}_1$*.*
We also know that the cross product

$$\mathbf{c} := \begin{bmatrix} \mathbf{v}_0 \\ 0 \end{bmatrix} \times \begin{bmatrix} \mathbf{v}_1 \\ 0 \end{bmatrix},$$

[1] Different authors use different definitions.

i.e., taking the $\mathbf{v}_i$ *as vectors in* $\mathbb{R}^3$*, only has a non-vanishing* z*-component. Therefore*

$$\left\| \begin{bmatrix} \mathbf{v}_0 \\ 0 \end{bmatrix} \times \begin{bmatrix} \mathbf{v}_1 \\ 0 \end{bmatrix} \right\| = \|\mathbf{v}_0\| \|\mathbf{v}_1\| |\sin(\phi)| = c_z,$$

where c_z *is the* z*-component of the cross product. A straightforward computation shows that* $c_z = \det B_K$*. Since the surface area of* K *satisfies* $|P| = 2|K|$*, we have*

$$\det B_K = 2|K|,$$

which is what had to be shown.
(1b) *For any* $r > 0$*, we may write*

$$\|B_K\| = \sup_{\boldsymbol{\xi} \in \mathbb{R}^2, \boldsymbol{\xi} \neq 0} \frac{\|B_K \boldsymbol{\xi}\|}{\|\boldsymbol{\xi}\|} = \frac{1}{r} \sup_{\boldsymbol{\xi} \in \mathbb{R}^2, \|\boldsymbol{\xi}\| = r} \|B_K \boldsymbol{\xi}\|,$$

in particular for $r = \hat{\rho}$*, i.e.,*

$$\|B_K\| = \frac{1}{\hat{\rho}} \sup_{\boldsymbol{\xi} \in \mathbb{R}^2, \|\boldsymbol{\xi}\| = \hat{\rho}} \|B_K \boldsymbol{\xi}\|. \tag{3}$$

Now, for each $\boldsymbol{\xi}$ *with* $\|\boldsymbol{\xi}\| = \hat{\rho}$ *we can find a pair of points* $\hat{\mathbf{x}}, \hat{\mathbf{y}} \in \hat{K}$ *such that* $\boldsymbol{\xi} = \hat{\mathbf{y}} - \hat{\mathbf{x}}$*. Let* $F_K(\boldsymbol{\xi}) := \mathbf{a} + B_K \boldsymbol{\xi}$ *be the map to the triangle* K *with root vertex* $\mathbf{a} \in \mathbb{R}^2$*. The map satisfies* $F_K(\boldsymbol{\xi}) = F_K(\hat{\mathbf{y}}) - F_K(\hat{\mathbf{x}})$*. Since the distance* $\|F_K(\hat{\mathbf{y}}) - F_K(\hat{\mathbf{x}})\| = \|B_K(\hat{\mathbf{x}} - \hat{\mathbf{y}})\| = \|\mathbf{y} - \mathbf{x}\|$ *is bounded by* h_K*, where* $\mathbf{x} = F_K(\hat{\mathbf{x}})$ *and* $\mathbf{y} = F_K(\hat{\mathbf{y}})$*, we find* $\|B_K \boldsymbol{\xi}\| \leq h_K$*, which proves the first inequality. The second inequality is derived by exchanging the roles of* $\hat{K}$ *and* K*.*
Now by definition of the spectral norm, we have

$$\|B_k B_K^{-1}\| \leq \|B_k\| \|B_K^{-1}\|,$$

and using the two previous inequalities we get:

$$\frac{\rho_K}{\hat{h}} \leq \|B_k\| \leq \frac{h_K}{\hat{\rho}}, \quad \frac{\hat{\rho}}{h_K} \leq \|B_k^{-1}\| \leq \frac{\hat{h}}{\rho_K}, \tag{4}$$

Finally, the result holds for $C_0 = 1/\hat{h}$*,* $C_1 = 1/\hat{\rho}$*.* $C_2 = \hat{\rho}/\hat{h}$ *and* $C_3 = \hat{h}/\hat{\rho}$*.*

(1c) *No, it is not. We can easily construct a triangle whose longest edge has length* 1 *and whose other two edges have length* $1/2 + \epsilon$ *for any* $\epsilon > 0$*. As* ϵ *goes to zero, the triangle gets flatter and flatter, but the ratio of longest and shortest edge converges to* 2*.*

Exercise 2. [Finite Element Method in 1D] Consider Poisson's equation with a non-constant diffusion coefficient $k(x) \in C^0([a,b])$, where $k(x) > 0$ for all $x \in (a,b)$:

$$\begin{cases} -(k(x)u'(x))' = f(x), & x \in (a,b), \\ u(a) = g_a, \quad u(b) = g_b. \end{cases} \tag{5}$$

(2a) Use the midpoint quadrature formula to derive the stiffness matrix A with entries

$$A_{i,j} = \int_a^b k(x) \phi_i'(x) \phi_j'(x) dx$$

and right hand side for a piecewise linear finite element approximation of the solution of system (5) with $g_a = g_b = 0$, when using a uniform grid, and compare it to the stiffness matrix that arises when using a second-order accurate finite difference approximation.

(2b) Implement the finite element approximation derived in **(2a)** for $a = 0, b = 1, g_a = 0$ and $g_b = 0$. Here, the diffusion coefficient and the right hand side are defined as

$$k(x) = \begin{cases} 0.5 + x, & x \leq 1/2 \\ 1.5 - x, & x > 1/2 \end{cases}$$

$$f(x) = \begin{cases} 0.5\pi \sin(\pi x) + \pi x \sin(\pi x) - \cos(\pi x), & x \leq 1/2 \\ 1.5\pi \sin(\pi x) - \pi x \sin(\pi x) + \cos(\pi x), & x > 1/2. \end{cases}$$

Note that the exact solution is given by $u(x) = \frac{1}{\pi} \sin(\pi x)$.

(2c) Derive a piecewise linear finite element approximation of problem (5) with $k(x) = k =$ constant and non-homogeneous Dirichlet boundary conditions $u(a) = g_a$ and $u(b) = g_b$.
Hint: Write the solution as a linear combination of the basis functions $(\phi_0, \ldots, \phi_N)$ and split

$$u_h = u_h^0 + g_h, \text{ where } u_h^0(x) \text{ is zero on the boundary, while } g_h = g_a\phi_0 + g_b\phi_N. \tag{6}$$

Note that this essentially eliminates two unknowns from the problem!

Solution:
(2a) *Multiplying the given differential equation by a test function v, integrating by part, and using the homogeneous Dirichlet boundary conditions (BCs) we obtain*

$$\int_a^b k(x)u'(x)v'(x)\, dx = \int_a^b f(x)v(x)\, dx. \tag{7}$$

Let

$$h = \frac{b-a}{N}, \quad x_i = a + ih, \quad i = 0, \ldots, N, K_i = [x_{i-1}, x_i], \quad i = 1, \ldots, N. \tag{8}$$

Moreover, we define the basis functions $\{\varphi_j\}_{j=1}^{N-1}$ as

$$\varphi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{h} & x \in K_i \\ \dfrac{x_{i+1} - x}{h} & x \in K_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

We then approximate the solution u by

$$u_h(x) = \sum_{j=1}^{N-1} u_j \varphi_j(x). \tag{9}$$

Note that due to the homogeneous Dirichlet BCs, we have $u_0 = u_N = 0$. Now we plug this into the weak form (7), set $v = \varphi_i$, and see that

$$\sum_{j=1}^{N-1} u_j \underbrace{\int_a^b k(x)\varphi_j'(x)\varphi_i'(x)\, dx}_{A_{i,j}} = \int_a^b f(x)\varphi_i(x)\, dx, \quad i = 1, \ldots, N-1. \tag{10}$$

This results in a $(N-1)\times(N-1)$ system of equations to solve. Now, we focus on $A_{i,j}$. For the derivatives of the basis functions, we immediately see that

$$\varphi_i'(x) = \begin{cases} \dfrac{1}{h} & x \in K_i \\ \dfrac{-1}{h} & x \in K_{i+1} \\ 0 & \textit{otherwise} \end{cases}$$

Clearly, when $|i-j|>1$, it follows that $A_{i,j}=0$ due to the compact support of the basis functions. Moreover,

$$A_{i,i-1} = \int_a^b k(x)\varphi_{i-1}'(x)\varphi_i'(x)\,dx = \int_{K_i} k(x)\varphi_{i-1}'(x)\varphi_i'(x)\,dx \approx hk(x_{i-1/2})\frac{-1}{h^2} = \frac{-k(x_{i-1/2})}{h}.$$

Similarly, for $A_{i,i+1}$, we have

$$A_{i,i+1} = \int_a^b k(x)\varphi_i'(x)\varphi_{i+1}'(x)\,dx = \int_{K_{i+1}} k(x)\varphi_i'(x)\varphi_{i+1}'(x)\,dx \approx hk(x_{i+1/2})\frac{-1}{h^2} = \frac{-k(x_{i+1/2})}{h}.$$

The entry $A_{i,i}$ is also given by

$$\begin{aligned} A_{i,i} &= \int_a^b k(x)\varphi_i'(x)\varphi_i'(x)\,dx = \int_{K_i} k(x)\varphi_i'(x)\varphi_i'(x)\,dx + \int_{K_{i+1}} k(x)\varphi_i'(x)\varphi_i'(x)\,dx \\ &\approx h\left(k(x_{i-1/2})+k(x_{i+1/2})\right)\frac{1}{h^2} = \frac{k(x_{i-1/2})+k(x_{i+1/2})}{h} \end{aligned}$$

The right hand side of the weak form (10) *may also be approximated by the midpoint rule or the right rectangular rule as follows*

$$\int_a^b f(x)\varphi_i(x)\,dx = \int_{K_i} f(x)\varphi_i(x)\,dx + \int_{K_{i+1}} f(x)\varphi_i(x)\,dx \approx hf(x_i).$$

This leads to the following system of equations

$$-\frac{1}{h}\left(k(x_{i-1/2})u_{i-1} - \left(k(x_{i-1/2})+k(x_{i+1/2})\right)u_i + k(x_{i+1/2})u_{i+1}\right) = hf(x_i), \quad i=1,\dots N-1,$$

together with the boundary conditions $u_0=u_N=0$. On the other hand, a centred finite difference approximation of the original equation gives

$$\begin{aligned} -\left(k(x)u'(x)\right)'(x_i) &= -\left(\frac{k_{i+1/2}u'(x_{i+1/2}) - k_{i-1/2}u'(x_{i-1/2})}{h}\right) + O(h^2) \\ &\approx -\left(k_{i+1/2}\frac{u_{i+1}-u_i}{h^2} - k_{i-1/2}\frac{u_i-u_{i-1}}{h^2}\right) \\ &= -\frac{1}{h^2}\left(k_{i+1/2}u_{i+1} - \left(k_{i+1/2}+k_{i-1/2}\right)u_i + k_{i-1/2}u_{i-1}\right) = f(x_i), \end{aligned}$$

which is precisely the same linear system as the finite element discretization rescaled by $1/h$.

(2b) *The Python code for this Exercise can be found in Listing 1.*
(2c) *Let* $K_j := [x_{j-1}, x_j], \quad h_j := x_j - x_{j-1}, \quad j = 1, \ldots, N$, *with* $a = x_0 < x_1 < \ldots < x_N = b$. *We define the basis functions* $\{\varphi_j\}_{j=0}^N$ *as*

$$\varphi_0(x) = \begin{cases} \dfrac{x_1 - x}{h_1}, & x \in K_1 \\ 0, & \textit{otherwise} \end{cases},$$

$$\varphi_N(x) = \begin{cases} \dfrac{x - x_{N-1}}{h_N}, & x \in K_N \\ 0, & \textit{otherwise} \end{cases},$$

and for $j = 1, \ldots, N-1$

$$\varphi_j(x) = \begin{cases} \dfrac{x - x_{j-1}}{h_j}, & x \in K_j \\ \dfrac{x_{j+1} - x}{h_{j+1}}, & x \in K_{j+1} \\ 0, & \textit{otherwise} \end{cases},$$

To derive a finite element approximation, we plug $u_h^0(x)$ *into the differential equation, multiply the equation by a test function* φ_j, *where* $j \in \{1, 2, \ldots, N-1\}$ *is a fixed integer, and integrate by parts to see that*

$$\sum_{i=1}^{N-1} \underbrace{\int_a^b k\varphi_i'(x)\varphi_j'(x)\,dx}_{A_{ij}}\, u_i^0 = \underbrace{\int_a^b f\varphi_j(x)\,dx}_{F_j} \underbrace{- \int_a^b k g_h'(x)\varphi_j'(x)\,dx}_{G_j}. \tag{11}$$

Note that the boundary terms vanish since $\varphi_j(a) = \varphi_j(b) = 0$, *for all* $j = 1, \ldots, N-1$. *The derivatives of the basis functions are given by*

$$\varphi_j'(x) = \begin{cases} \dfrac{1}{h_j} & x \in K_j \\ \dfrac{-1}{h_{j+1}} & x \in K_{j+1} \\ 0 & \textit{otherwise} \end{cases}$$

Clearly, $A_{ij} = 0$ *if* $|i - j| > 1$. *Moreover,*

$$A_{j,j-1} = \int_a^b k\varphi_{j-1}'(x)\varphi_j'(x)\,dx = \int_{K_j} k\varphi_{j-1}'(x)\varphi_j'(x)\,dx \approx h_j k \frac{-1}{h_j^2} = \frac{-k}{h_j}.$$

Similarly, for $A_{j,j+1}$, *we have*

$$A_{j,j+1} = \int_a^b k\varphi_j'(x)\varphi_{j+1}'(x)\,dx = \int_{K_{j+1}} k\varphi_j'(x)\varphi_{j+1}'(x)\,dx \approx h_{j+1} k \frac{-1}{h_{j+1}^2} = \frac{-k}{h_{j+1}}.$$

The entry $A_{j,j}$ *is also given by*

$$A_{j,j} = \int_a^b k\varphi_j'(x)\varphi_j'(x)\,dx = \int_{K_j} k\varphi_j'(x)\varphi_j'(x)\,dx + \int_{K_{j+1}} k\varphi_j'(x)\varphi_j'(x)\,dx$$

$$\approx h_j k \frac{1}{h_j^2} + h_{j+1} k \frac{1}{h_{j+1}^2} = k \left(\frac{1}{h_j} + \frac{1}{h_{j+1}} \right)$$

The term F_j in the right hand side of the weak form (11) *may be approximated as*

$$\int_a^b f(x)\varphi_j(x)\,dx = \int_{K_j} f(x)\varphi_j(x)\,dx + \int_{K_{j+1}} f(x)\varphi_j(x)\,dx \approx h_j f(x_j).$$

Finally,

$$\begin{aligned}
G_j &= -\left(\int_a^b k g_h'(x)\varphi_j'(x)\,dx \right) \\
&= -\left(\int_a^b k\left(g_a\varphi_0(x) + g_b\varphi_N(x)\right)' \varphi_j'(x)\,dx \right) \\
&= -k\left(\int_a^b g_a\varphi_0'(x)\varphi_j'(x)\,dx + \int_a^b g_b\varphi_N'(x)\varphi_j'(x)\,dx \right) \\
&= -k\left(g_a h_1 \delta_{1,j} \frac{-1}{h_1^2} + g_b h_N \delta_{j,N-1} \frac{-1}{h_N^2} \right) \\
&= k\left(\delta_{1,j} \frac{g_a}{h_1} + \delta_{j,N-1} \frac{g_b}{h_N} \right),
\end{aligned}$$

hence G_j is nonzero only for $j = 1$ and $j = N - 1$. In particular, in the case of a uniform grid $h_j = h, \forall j$, we obtain the system

$$\frac{k}{h} \begin{bmatrix} 2 & -1 & \\ -1 & \ddots & \ddots \\ & \ddots & 2 \end{bmatrix} \begin{bmatrix} u_1^0 \\ \vdots \\ \vdots \\ u_{N-1}^0 \end{bmatrix} = \begin{bmatrix} hf(x_1) + kg_a/h \\ hf(x_2) \\ \vdots \\ hf(x_{N-2}) \\ hf(x_{N-1}) + kg_b/h \end{bmatrix}$$

which, once again, coincides with a centred second order finite difference approximation.

Listing 1: Python code

```
#!/usr/bin/python3

"""
@author: Jochen Hinz
"""

import numpy as np
from matplotlib import pyplot as plt
from scipy import sparse
from scipy.sparse import linalg as splinalg

def main(N=21):
  """ N: number of grid points. """

  # step size
  h = 1 / (N - 1)
```

```
    # grid point coordinates
    x = np.arange(0, 1+h, h)

    # the x_{i + 1/2}
    x_12 = x[:-1] + h/2

    # we utilise the np.piecewise function to define the diffusivity and the rhs
    k12 = np.piecewise(x_12, [x_12 <= .5, x_12 > .5], [lambda x: .5 + x, lambda
        x: 1.5 - x])

    # create the rhs function in the relevant points by using, again, the
        np.piecewise function
    rhs = h * np.piecewise(x, [x <= .5, x > .5],
                            [lambda x:  .5 * np.pi * np.sin(np.pi * x) +
                                np.pi * x * np.sin(np.pi * x) - np.cos(np.pi
                                * x),
                             lambda x: 1.5 * np.pi * np.sin(np.pi * x) -
                                np.pi * x * np.sin(np.pi * x) + np.cos(np.pi
                                * x)] )

    # tridiagonal sparse matrix with off-diagonal entries -k12[1:-1]
    # and diagonal entries k12[:-1] + k12[1:]
    # (scaled by 1/h)
    A = 1 / h * sparse.diags( diagonals=[-k12[1:-1], k12[:-1] + k12[1:],
        -k12[1:-1]],
                            offsets=[-1, 0, 1],
                            format='csr'
                                                                    )

    # sol = [0, inner gridpoint solution, 0]
    sol = np.array([0, *splinalg.spsolve(A, rhs[1:-1]), 0])

    # plot the solution
    fig, ax = plt.subplots()

    # plot the approximate solution
    ax.plot(x, sol, '-o', c='b', label='approximate solution')

    # plot the exact solution for comparison
    xi = np.linspace(0, 1, 1001)
    ax.plot(xi, 1 / np.pi * np.sin(np.pi * xi), c='r', label='exact solution')

    ax.legend()
    ax.grid(True, which='both', axis='both', color='gray', linestyle='--',
        linewidth=0.5)

    ax.set_title(r"Numerical solution of $-(k(x) u'(x))' = f(x)$")
    ax.set_xlabel("x")

    plt.show()

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