Numerical Approximation of PDEs

Spring Semester 2025

Lecturer: Prof. Annalisa Buffa Assistant: Mohamed Ben Abdelouahab

Session 5: March 27, 2025

Exercise 1. Consider a regular N-simplex K with vertices $\{v_i\}_{1\leq i\leq N+1}$.

1. Show that the quadrature formulas

$$\int_K p(x)\,dx \approx |K| p(v_0),$$

where $|K|$ denotes the volume of K and

$$v_0 = \frac{1}{N+1}\sum_{i=1}^{N+1} v_i,$$

is the barycenter of K, and

$$\int_K p(x)\,dx \approx \frac{|K|}{N+1}\sum_{i=1}^{N+1} p(v_i)$$

are exact for all linear polynomials $p \in \mathbb{P}_1(K)$.

2. Consider now $\{v_i\}_{1\leq i\leq I}$ a set points in the N-simplex K and $\{\omega_i\}_{1\leq i\leq I}$ real weights. Suppose we have a quadrature formula

$$\int_K \varphi(x)\,dx \approx |K|\sum_{i=1}^{I} \omega_i \varphi(v_i)$$

which is exact for $p \in \mathbb{P}_k(K)$ (polynomials of maximum degree k). Show that for a sufficiently smooth function φ, we have

$$\frac{1}{|K|}\int_K \varphi(x)\,dx = \sum_{i=1}^{I} \omega_i \varphi(v_i) + \mathcal{O}(h^{k+1}),$$

where h is the diameter of K.

Hint: For *1*, express $p(x)$ in terms of the barycentric coordinates $\{\lambda_i(x)\}_{1\leq i\leq N+1}$ on K and use their integral properties over the simplex. For *2*, use Taylor expansion for φ and the Taylor-Lagrange inequality.

Solution:

1. *Let $p(x)$ be a polynomial of degree 1. Since the vertices v_i define the simplex, we can express $p(x)$ in terms of barycentric coordinates as*

$$p(x) = \sum_{i=1}^{N+1} \lambda_i(x) p(v_i).$$

We integrate both sides over K:

$$\int_K p(x)\, dx = \sum_{i=1}^{N+1} p(a_i) \int_K \lambda_i(x)\, dx.$$

Using the known integral property of barycentric coordinates,

$$\int_K \lambda_i(x)\, dx = \frac{|K|}{N+1},$$

we obtain

$$\int_K p(x)\, dx = \frac{|K|}{N+1} \sum_{i=1}^{N+1} p(v_i).$$

This proves the exactness of the second quadrature formula for linear functions.

For the second quadrature formula, recall that v_0 satisfies

$$v_0 = \frac{1}{N+1} \sum_{i=1}^{N+1} v_i.$$

Since $p(x)$ is affine, evaluating at v_0 gives

$$p(v_0) = \frac{1}{N+1} \sum_{i=1}^{N+1} p(v_i).$$

Multiplying by $|K|$ on both sides, we conclude

$$\int_K p(x)\, dx = |K| p(v_0),$$

which proves the exactness of the first quadrature formula.

2. *Let φ be a function of class C^{k+1}. By performing a Taylor expansion, there exists a constant C such that for any point x_0, there exists a polynomial T_{x_0} depending on φ, of degree at most k, such that*

$$|\varphi(x) - T_{x_0}(x - x_0)| \leq C|x - x_0|^{k+1} \quad \forall x \in \mathbb{R}^N.$$

We integrate the previous inequality over $x \in K$, x_0 is chosen such that $x_0 - x \in K$ (in particular, $|x - x_0| < h$), we obtain

$$\left| \int_K \varphi \, dx - \int_K T_{x_0}(x - x_0) \, dx \right| \leq C|K|h^{k+1}.$$

Since the quadrature formula is exact for polynomials of degree at most k, we deduce

$$\left| \int_K \varphi \, dx - |K| \sum_i \omega_i T_{x_0}(v_i - x_0) \right| \leq C|K|h^{k+1}.$$

Now, for any v_i, we use the Taylor expansion of φ at x_0,

$$|\varphi(v_i) - T_{x_0}(v_i - x_0)| \leq Ch^{k+1}$$

and we conclude using the triangle inequality

$$\begin{aligned} \left| \int_K \varphi \, dx - |K| \sum_i \omega_i \varphi(v_i) \right| &\leq \left| \int_K \varphi \, dx - |K| \sum_i \omega_i T_{x_0}(v_i - x_0) \right| + |K| \sum_i \omega_i \, |T_{x_0}(v_i - x_0) - \varphi(v_i)| \\ &\leq 2C|K|h^{k+1}. \end{aligned}$$

which completes the proof.

Exercise 2. [First Strang Lemma] The use of quadrature instead of exact integration introduces an error in the bilinear and linear forms $a(\,\cdot\,,\,\cdot\,)$ and $F(\,\cdot\,)$.
We may denote the approximated forms by $a_h(\,\cdot\,,\,\cdot\,)$ and $F_h(\,\cdot\,)$, respectively. Assume that the quadrature is accurate enough to retain uniform coercivity of $a_h(\,\cdot\,,\,\cdot\,)$ over the discrete space $\mathcal{V}_h \times \mathcal{V}_h$ for all $h > 0$. Then there exists a constant α^* such that

$$a_h(v_h, v_h) \geq \alpha^* \|v_h\|^2 \quad \forall v_h \in \mathcal{V}_h \tag{1}$$

and the unique solution u_h of $a_h(u_h, v_h) = F(v_h)$, $\forall v_h \in \mathcal{V}_h$ satisfies

$$\|u_h\| \leq \frac{1}{\alpha^*} \sup_{v_h \in \mathcal{V}_h, \|v_h\| \neq 0} \frac{F_h(v_h)}{\|v_h\|}. \tag{2}$$

Let u be the solution to $a(u, v) = F(v)$, $\forall v \in \mathcal{V}$ (the continuous problem). Show that

$$\begin{aligned} \|u - u_h\| \leq \inf_{w_h \in \mathcal{V}_h} [\Big(1 + \frac{\gamma}{\alpha^*}\Big) \|u - w_h\| + \frac{1}{\alpha^*} \sup_{v_h \in \mathcal{V}_h, \|v_h\| \neq 0} \frac{|a(w_h, v_h) - a_h(w_h, v_h)|}{\|v_h\|}] \\ + \frac{1}{\alpha^*} \sup_{v_h \in \mathcal{V}_h, \|v_h\| \neq 0} \frac{|F(v_h) - F_h(v_h))|}{\|v_h\|}, \end{aligned} \tag{3}$$

where $\gamma > 0$ is the continuity constant that satisfies $|a(v, w)| \leq \gamma \|v\| \|w\|$, $\forall v, w \in \mathcal{V}$.

Solution:
Let $w_h \in \mathcal{V}_h$ be arbitrary. Set $\sigma_h = u_h - w_h$. Using (1), we get

$$\alpha^* \|\sigma_h\|^2 \leq a_h(\sigma_h, \sigma_h) = a(u - w_h, \sigma_h) + a(w_h, \sigma_h) - a_h(w_h, \sigma_h) + F_h(\sigma_h) - F(\sigma_h).$$

We assume $\sigma_h \neq 0$ and divide both side by $\sigma_h \neq 0$ and use the definition of γ to find

$$\begin{aligned} \alpha^* \|\sigma_h\| \leq \quad & \gamma \|u - w_h\| + \frac{|a(w_h, \sigma_h) - a_h(w_h, \sigma_h)|}{\|\sigma_h\|} + \frac{|F(\sigma_h) - F_h(\sigma_h)|}{\|\sigma_h\|} \\ \leq \quad & \gamma \|u - w_h\| + \sup_{v_h \in \mathcal{V}_h, \|v_h\| \neq 0} \frac{|a(w_h, v_h) - a_h(w_h, v_h)|}{\|v_h\|} \\ & + \sup_{v_h \in \mathcal{V}_h, \|v_h\| \neq 0} \frac{|F(v_h) - F_h(v_h)|}{\|v_h\|}. \end{aligned} \tag{4}$$

Note that the above also holds for $\sigma_h = 0$. Combining the above with the triangle inequality $\|u - u_h\| \leq \|u - w_h\| + \|\sigma_h\|$ and taking the infimum with respect to $w_h \in \mathcal{V}_h$, the result follows.

Exercise 3. Last week we learned how to assemble the mass and stiffness matrices along with the load vector which enabled us to solve basic elliptic PDEs subject to pure Dirichlet data. This week, we will consider a problem with mixed Dirichlet / Neumann boundary conditions

$$\begin{aligned} -\Delta u + u = 1 \quad & \text{in } \Omega \\ u = 0 \quad & \text{on } \partial\Omega_D \\ \partial_n u = g \quad & \text{on } \partial\Omega_N, \end{aligned} \tag{5}$$

where $\Omega = (0,1)^2$ while $\partial\Omega_N = [0,1]$, i.e., the bottom part of $\partial\Omega$ and $\partial\Omega_D = \partial\Omega \setminus \partial\Omega_n$ and $g \in L^2(\partial\Omega_N)$.

1. Derive the weak form of (4) and derive an expression for the additional right hand side vector that arises as a result of the Neumann condition.

2. Utilize the provided template script `code05_3_template.py` to write a FEM code for the problem with $g = 1$. The template contains a function `assemble_neumann_rhs` which takes as input the mesh, the neuman data g (here: $g = 1$) and a boolean mask of shape `(mesh.lines.shape[0],)` indicating whether the '`i`'-th boundary line is part of the Neumann boundary. As such, the function `reaction_diffusion` requires you to additionally find this boolean mask. Note that the local Neumann load vector only has length two.

The template contains the same functions as last week's template and requires you to fill in exactly one additional line of code!

Solution:
Find the solution script on Moodle.

Exercise 4. [Non-constant coefficients]
Consider the following problem in $\Omega = (0,1)^2$

$$\left\{ \begin{aligned} -\nabla \cdot (a(x,y) \nabla u(x,y)) + r(x,y)\, u(x,y) &= f(x,y) \quad \text{in } \Omega, \\ \partial_n u(x,y) &= 0 \quad \text{on } \partial\Omega \end{aligned} \right. \tag{6}$$

with coefficients and right-hand side

$$a(x,y) = 1 + x, \quad r(x,y) = 4\pi^2(1+x)$$
$$f(x,y) = 2\pi\cos(2\pi y)[\sin(2\pi x) + 6\pi(1+x)\cos(2\pi x)].$$

In this exercise, we compute the entries of the stiffness matrix, the mass matrix, and the right-hand side. As we have seen in lecture, we need to compute integrals over single elements. While we have computed the integrals exactly last week, this exercise lets you explore the use of quadrature formulas.

You are given auxiliary files and template codes in Python: `mesh.py` contains the code for mesh management. `quad.py` contains a Python class for quadrature formulas. `util.py` provides utility functions. The file `integrate_template.py` contains a function `solve_problem_3` that you can complete.

1. Discuss why or under what circumstances you would use a quadrature formula in the first place instead of trying to compute the integrals exactly.
2. Implement a function `shape2D_LFE` which takes in an N_p-by-2 array of points in the reference triangle and returns an array N_p-by-3 array which contains in column i the evaluation of the reference Lagrange basis function $\hat{\varphi}_i$ at points `x`, $i = 1, 2, 3$.
3. Implement a function `grad_shape2D_LFE` which takes in an N_p-by-2 array of points in the reference triangle and returns an N_p-by-3-by-2 array which contains in entry `[i, j, k]` the k-th entry of the gradient of $\hat{\varphi}_j$ evaluated in the i-th quadrature point.
4. Implement a function `mass_with_reaction_iter` which takes as input the triangulation, a quadrature formula, and (optionally) a reaction term coefficient function, and which computes the mass matrix with a given reaction term.
5. Implement a function `stiffness_with_diffusivity_iter` which takes as input the triangulation, a quadrature formula, and (optionally) a diffusion term coefficient function, and which computes the stiffness matrix with a given diffusion term.
6. Implement the function `poisson_rhs_iter` which takes as input the triangulation, a quadrature formula, and a callable function, and which computes the right-hand side in the linear system of equations.
7. Solve problem (5) using a piecewise linear finite element method. Note that the true solution is given by $u(x,y) = \cos(2\pi x)\cos(2\pi y)$. You can use the seven-point Gaussian quadrature rule that is already provided.

CHALLENGE YOURSELF BY USING NUMPY VECTORISATION

Solution:
The solution files are provided on Moodle.