SOLUTION 5 – MATH-250 Numerical Analysis

The exercise sheet is divided into two sections: quiz and exercises. The quiz will be discussed in the beginning of the lecture on Thursday, March 27. The exercises marked with ($\star$) are graded homework. The exercises marked with **(Python)** are implementation based and can be solved in the Jupyter notebooks which are available on Moodle/Noto. **The deadline for submitting your solutions to the homework is Friday, March 28 at 10h15.**

Problem 4.

We consider the integral

$$\int_{-1}^{1} f(x)w(x)\,\mathrm{d}x, \quad w(x) = \frac{1}{\sqrt{1-x^2}}.$$

Additionally, we define the following inner product

$$\langle u, v\rangle_w = \int_{-1}^{1} u(x)v(x)w(x)\,\mathrm{d}x. \tag{1}$$

(a) Consider

$$I_n = \int_{-1}^{1} x^n w(x)\,\mathrm{d}x.$$

For n odd, meaning that $n \equiv 1 \mod 2$, explain why $I_n = 0$. Compute the value of I_0. For even n, meaning $n \equiv 0 \mod 2$, derive the recurrence relation

$$I_n = \frac{n-1}{n} I_{n-2}.$$

(b) Apply the Gram-Schmidt algorithm to orthogonalize the monomials $1, x, x^2, x^3$ with respect to the inner product $\langle\cdot,\cdot\rangle_w$ as defined in (1). Use the result from (a) to perform these calculations. Normalize the resulting orthogonal polynomials p_0, p_1, p_2, and p_3 such that $p_i(1) = 1$ for $i = 0, 1, 2, 3$.

It turns out that the resulting polynomials, orthogonal with respect to the scalar product defined in (1), are the so called *Chebyshev polynomials*, and it can be shown that the roots of p_{n+1} are the *Chebyshev nodes*

$$x_i = \cos\left(\pi \frac{2i+1}{2(n+1)}\right), \quad i = 0, 1 \ldots, n.$$

Write a Python script that verifies this equality. For the handling of polynomials, NumPy provides the class `numpy.polynomial.polynomial.Polynomial`. Use this class and its associated function `roots` to compute the roots.

Hint: The Jupyter notebook provided on Moodle imports the `Polynomial` class under the alias `poly`. Hence, you can directly call `poly` to create your polynomials.

(c) Now let $p_{n+1} \in \mathbb{P}_{n+1}$ be the polynomial that is orthogonal to $\mathbb{P}_n$ with respect to (1) and satisfies $p_{n+1}(1) = 1$. The polynomial p_{n+1} has $n+1$ distinct roots $x_0, \ldots, x_n \in (-1, 1)$, defined above. Consider the quadrature rule defined by

$$Q_n[f] = \sum_{i=0}^{n} \alpha_i f(x_i), \quad \alpha_i = \int_{-1}^{1} \ell_i(x) w(x) \, \mathrm{d}x,$$

where $\ell_0, \ell_1, \ldots, \ell_n$ are the usual Lagrange polynomials associated with $x_0, x_1, \ldots, x_n$.

Show that the quadrature rule Q_n has order $2n + 2$ for the weighted integral, that is,

$$Q_n[p] = \int_{-1}^{1} p(x) w(x) \, \mathrm{d}x \quad \forall p \in \mathbb{P}_{2n+1}.$$

Hint: Adapt the arguments made in the beginning of Section 2.5 in the lecture notes.

(d) Write a Python function `cheb_quad(f, num)` implementing the quadrature rule $Q_n[f]$ from (c) using the weights

$$\alpha_i = \frac{\pi}{n+1}, \quad i = 1, 2, \ldots, n.$$

Apply $Q_n[f]$ to $f_1(x) = \frac{|x|^{1/5}}{|x+2|+|x-2|}$ and $f_2(x) = \frac{\exp(-x^2)}{\cos x|x|}$ for $n = 1, 2, \ldots, 1000$. Display the approximation errors of f_1 on a loglog plot, and the approximation errors of f_2 on a semilogy plot, with the x-axis showing the number of nodes n. For the computation of the reference integral you can use SciPy's integration module with the function `scipy.integrate.quad(f, -1, 1, epsabs=1e-16)`. Make sure you use the correct function f!

Solution.

All Python code is available in the Jupyter notebook `homework05-sol.ipynb` on Moodle.

(a) Let n be odd, $n \equiv 1 \mod 2$. Observe that $f_n(x) = \frac{x^n}{\sqrt{1-x^2}}$ satisfies $f_n(x) = -f_n(-x)$. This implies that we can split the interval $[-1, 1]$ into $[-1, 0]$ and $[0, 1]$, and that

$$\int_{-1}^{0} f_n(x) \, \mathrm{d}x = -\int_{0}^{1} f_n(x) \, \mathrm{d}x$$

holds true. Hence, the integral I_n equals 0 for odd values of n.

For $n = 0$ we note that the primitive of $f_0(x) = \frac{1}{1-x^2}$ is $F_0(x) = \arcsin(x) + C$, and therefore the integral has the value $I_0 = \pi$.

Now, let $n > 1$. Applying integration by parts twice results in the following equation

$$\begin{aligned} I_n &= \int_{-1}^{1} x^{n-1} \frac{x}{1-x^2} \,\mathrm{d}x = (n-1) \int_{-1}^{1} x^{n-2} \sqrt{1-x^2} \,\mathrm{d}x \\ &= (n-1) \int_{-1}^{1} x^{n-2} \frac{1-x^2}{\sqrt{1-x^2}} \,\mathrm{d}x \\ &= (n-1) \int_{-1}^{1} \frac{x^{n-2}}{\sqrt{(1-x^2)}} \,\mathrm{d}x - (n-1) \int_{-1}^{1} \frac{x^n}{\sqrt{1-x^2}} \,\mathrm{d}x \\ &= (n-1) I_{n-2} - (n-1) I_n. \end{aligned}$$

Rearranging these terms gives us $I_n = \frac{n-1}{n} I_{n-2}$.

(b) We start with $p_0(x) = 1$.

- $\langle x, p_0 \rangle_w = I_1 = 0$, and hence $p_1(x) = x$.
- $\langle x^2, p_0 \rangle_w = I_2 = \frac{\pi}{2}, \langle p_0, p_0 \rangle_w = I_0 = \pi$, and $\langle x^2, p_1 \rangle_w = I_3 = 0$; hence we get $p_2(x) = x^2 - \frac{1}{2}$.
- $\langle x^3, p_0 \rangle_w = I_3 = 0, \langle x^3, p_1 \rangle_w = I_4 = \frac{3\pi}{8}, \langle p_1, p_1 \rangle_w = I_2 = \frac{\pi}{2}$, and $\langle x^3, p_2 \rangle_w = I_5 - \frac{\pi}{2} I_3 = 0$; hence we get $p_3(x) = x^3 - \frac{3}{4} x$.

Normalization of the above polynomials gives us

$$\hat{p_0}(x) = 1, \quad \hat{p_1}(x) = x, \quad \hat{p_2}(x) = 2x^2 - 1, \quad \text{and} \quad \hat{p_3}(x) = 4x^3 - 3x.$$

Please find the Python code for this task in the Jupyter notebook on Moodle.

(c) Applying the Lagrange interpolation ansatz we see that

$$Q_n[f] = \sum_{i=0}^{n} \int_{-1}^{1} f(x_i) \ell_i(x) w(x) \,\mathrm{d}x.$$

Here, we have used n interpolation points, whence $Q_n[f]$ must be exact for polynomials of degree up to n, meaning that the order of Q_n is at least $n + 1$.

Now, let $f(x) \in \mathbb{P}_{2n+1}$ and define $g(x) = \prod_{i=0}^{n} (x - x_i)$. Polynomial division yields

$$f(x) = g(x) q(x) + r(x) \tag{2}$$

with two polynomials $q, r \in \mathbb{P}_n$. Integrating (2), we observe

$$\int_{-1}^{1} f(x) w(x) \,\mathrm{d}x - \int_{-1}^{1} g(x) q(x) w(x) \,\mathrm{d}x = \int_{-1}^{1} r(x) w(x) \,\mathrm{d}x = \sum_{i=0}^{n} \alpha_i r(x_i) = Q_n[r].$$

Now we can choose the x_i to be the zeros of a polynomial $h \in \mathbb{P}_{n+1}$ such that h is orthogonal to $\mathbb{P}_n$ w.r.t. $\langle \cdot, \cdot \rangle_w$. This means that

$$\int_{-1}^{1} g(x) q(x) w(x) \,\mathrm{d}x = 0,$$

and further, and finally, we get from the original polynomial division (2) that

$$\sum_{i=0}^{n} \alpha_i r(x_i) = \sum_{i=0}^{n} \alpha_i f(x_i).$$

This concludingly proves that the quadrature rule Q_n has the order $2n + 2$.

(d) Please find the Python code for this task in the Jupyter notebook on Moodle.