Exercise set 4 – MATH-250 Advanced Numerical Analysis I

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 20. The exercises marked with ($\star$) are graded homework. **The deadline for submitting your solutions to the homework is Friday, March 21 at 10h15.**

Quiz

(a) Given a function $f \in C^\infty([a,b])$, we consider the composite trapezoidal rule $Q_h^{(1)}[f]$ and the composite Simpson rule $Q_h^{(2)}[f]$ on the interval $[a,b]$. Which of the following statements are correct?

(i) $\lim_{h\to 0} Q_h^{(1)}[f] = \lim_{h\to 0} Q_h^{(2)}[f] = \int_a^b f(x)\,\mathrm{d}x$.

☐ True ☐ False

(ii) $|Q_h^{(1)}[f] - \int_a^b f(x)\,\mathrm{d}x| \le |Q_H^{(1)}[f] - \int_a^b f(x)\,\mathrm{d}x|$ if $h \le H$.

☐ True ☐ False

(iii) $|Q_h^{(2)}[f] - \int_a^b f(x)\,\mathrm{d}x| \le |Q_h^{(1)}[f] - \int_a^b f(x)\,\mathrm{d}x|$ for all sufficiently small values $h > 0$.

☐ True ☐ False

(iv) If $f(x) \ge 0$ for all $x \in [a,b]$ then

$$0 \le Q_h^{(1)}[f] \le \int_a^b f(x)\,\mathrm{d}x.$$

☐ True ☐ False

(v) If f is convex on $[a,b]$ then

$$Q_h^{(1)}[f] \ge \int_a^b f(x)\,\mathrm{d}x.$$

☐ True ☐ False

Exercises (Exercises marked ($\star$) will be graded.)

Problem 1.

(a) Compute the approximation errors for the trapezoidal rule and Simpson rule applied to the integrals:

$$\int_0^1 x^4\,\mathrm{d}x \quad \text{and} \quad \int_0^1 x^5\,\mathrm{d}x$$

(b) Find C such that the trapezoidal rule gives the exact result for the integral

$$\int_0^1 x^5 - Cx^4 \,\mathrm{d}x.$$

(c) Show that the trapezoidal rule gives a better approximation than the Simpson rule in the case that $\frac{15}{14} < C < \frac{85}{74}$.

Problem 2.

Let $\{b_i\}_{i=1}^N \subset \mathbb{R}$ and $\{c_i\}_{i=1}^N \subset [0,1]$ define a quadrature rule $Q[f] = \sum_{i=1}^N b_i f(c_i)$ with $N \in \mathbb{N}$ nodes for approximating $\int_0^1 f(x)\,\mathrm{d}x$. The quadrature rule Q is called *symmetric* if $c_i = 1 - c_{N+1-i}$ and $b_i = b_{N+1-i}$, for all $i = 1, 2, \ldots, N$.

Show that any symmetric quadrature rule has an even order, that is if Q is exact for polynomials of degree $\leq 2m-2$ for some $m \in \mathbb{N}$, then it is automatically exact for polynomials of degree $2m-1$.

Problem 3.

We wish to approximate

$$I_{[a,b]}[f] = \int_a^b f(x)\,\mathrm{d}x$$

using the midpoint rule, where $f \in C^2([a,b])$.

Let $Q_{[a,b]}[f]$ denote the midpoint rule. We know from Theorem 2.5 that

$$|I_{[a,b]}[f] - Q_{[a,b]}[f]| \leq \frac{(b-a)^3}{24}\|f''\|_\infty$$

where $\|f''\|_\infty = \sup_{x\in[a,b]} |f''(x)|$. Let $x_i = a + ih$ for $i = 0, 1, \ldots, N$ and $h = \frac{b-a}{N}$. Let $Q_h(f)$ denote the composite midpoint rule, where we apply the midpoint rule to each of the N subintervals of length h.

(a) Derive an upper bound for the approximation error of the composite midpoint rule:

$$|Q_h[f] - I_{[a,b]}[f]| \tag{1}$$

(b) Let $[a,b] = [0,1]$ and $f(x) = \exp(-x^2)$ and fix $\varepsilon > 0$. Using your result from (a), find the value for N such that (1) is guaranteed to be smaller than ε.

(c) Implement a Python function `midpoint_rule(f, a, b, N)` for the composite midpoint rule. With the functions $f_1(x) = \sqrt{x}$ and $f_2(x) = \frac{1}{\sqrt{x}}$ in $[a,b] = [0,1]$ display $|Q_h[f] - I_{[a,b]}[f]|$ with respect to N or h on a doubly logarithmic plot (use the `matplotlib` function `matplotlib.pyplot.loglog/plt.loglog`). Using these plots, find suitable values of p that describe the asymptotic behavior $\mathcal{O}(N^{-p})$ of the error for f_1 and f_2, respectively.

Problem 4. ($\star$)

Given a function $g\colon [-1,1] \to \mathbb{R}$ we approximate the integral

$$\int_{-1}^{1} g(z)\,\mathrm{d}z$$

with the "simple" Gauss-Lobatto quadrature rule with 4 quadrature points

$$Q_{[-1,1]}[g] = \frac{1}{6}(g(1) + g(-1)) + \frac{5}{6}(g(z_1) + g(z_2)), \tag{2}$$

where $z_{1,2} = \frac{\pm 1}{\sqrt{5}}$.

(a) Determine the order of $Q_{[-1,1]}$ experimentally by evaluating it for the monomials $1, x, x^2, \ldots$

(b) Given a function $f\colon [a,b] \to \mathbb{R}$ for $a < b$ we can rewrite

$$\int_a^b f(x)\,\mathrm{d}x = \frac{b-a}{2} \int_{-1}^{1} f\left(\frac{b-a}{2}z + \frac{b+a}{2}\right) \mathrm{d}z$$

and therefore apply (2) to get the "general" Gauss-Lobatto quadrature rule $Q_{[a,b]}[f]$ with 4 quadrature points on the interval $[a,b]$.

Implement the Python function `general_gauss_lobatto(f, a, b)` for the "general" Gauss-Lobatto quadrature rule $Q_{[a,b]}[f]$.

Hint: You can use the implementation of `simple_gauss_lobatto(f)` we provide in the Jupyter notebook for the "simple" Gauss-Lobatto quadrature rule $Q_{[-1,1]}[f]$ as a guide.

(c) Consider the interval $I = [a,b]$ and split it into N equally sized subintervals $[x_i, x_{i+1}] = [a + i \cdot h, a + (i+1) \cdot h]$ for $i = 0, 1, \ldots, N-1$ with $h = \frac{b-a}{N}$. We define the composite quadrature rule $Q_h[f]$ for the function $f\colon [a,b] \to \mathbb{R}$

$$Q_h[f] = \sum_{i=0}^{N-1} Q_{[x_i, x_{i+1}]}[f]$$

where $Q_{[x_i,x_{i+1}]}$ refers to the "general" Gauss-Lobatto quadrature rule from (b) with $a = x_i$ and $b = x_{i+1}$.

Implement the Python function `composite_gauss_lobatto(f, a, b, N)` for the "general" Gauss-Lobatto quadrature rule $Q_h[f]$.

(d) Consider the functions

$$f_1(x) = \exp(x)\cos(x) + 1 \quad \text{and} \quad f_2(x) = \sqrt{|x|^5}$$

with their respective exact integrals

$$\int_0^3 f_1(x)\,\mathrm{d}x = \frac{\exp(3)(\sin(3) + \cos(3)) - 1}{2} + 3 \quad \text{and}$$
$$\int_{-2}^{2} f_2(x)\,\mathrm{d}x = \frac{32\sqrt{2}}{7}.$$

Compute the approximated integrals with the composite Gauss-Lobatto rule and the composite trapezoidal rule for both f_1 and f_2. For f_1 and f_2. You can use the implementation of `composite_trapezoidal(f, a, b, N)` that we provided you or your own.

Plot the errors of both quadrature rules with respect to the exact integrals for decreasing $h = \frac{b-a}{N}$ in a doubly logarithmic plot (`matplotlib.pyplot.loglog`/`plt.loglog`).

Hint: When choosing the number of subintervals N for the error computation you can define them as `np.logspace(1, 4, num=15, dtype=int)`.

(e) Consider the plots from (d). The errors for the Gauss-Lobatto rules are of order $\mathcal{O}(h^{s_{\mathrm{GL}}})$ and the errors for the trapezoidal rule are of order $\mathcal{O}(h^{s_{\mathrm{T}}})$.

Write down s_{GL} and s_{T} for f_1 and f_2, respectively. Be sure to clearly mark which of the four values correspond to which function.

(f) (**This is for your understanding and will not be graded.**) Assume that our implementation of the composite Gauss-Lobato quadrature calls the Python function `general_gauss_lobatto(f, a, b)` on each of the $N > 0$ subintervals. This way of implementing the composite rule is inefficient because we evaluate f more often than it is necessary.

Given a fixed number of subintervals $N > 0$, state how many evaluations of f can be avoided. Explain how you avoided those additional evaluations of f. If you want, implement your more efficient quadrature rule as a Python function.

Remember to upload the completed Jupyter notebook `homework04.ipynb` corresponding to the homework to the submission panel on Moodle until Friday, March 21 at 10h15. To download your notebook from Noto, use `File > Download` Only your submissions to Moodle will be considered for grading.