Fachbereich Mathematik und Informatik/ Institut für Mathematik
Prof. Dr. Volker John, john@wias-berlin.de
Balázs Kossovics, b.kossovics@fu-berlin.de

Berlin, 03.06.2024

Numerik I

English translation of Übungsserie 07

Attention: Only solutions which provide a comprehensible reasoning will be graded. Every statement has to be argued. You can use results from the lecture. Statments without reasoning won't get any points.

1. *Quadrature rule with derivative of the function.* Let $f \in C^1([0,h])$. Determine the weights λ_i, $i \in \{0,1,2,3\}$, such that

$$\int_0^h f(x)\ dx = h\left(\lambda_0 f(0) + \lambda_1 f(h)\right) + h^2\left(\lambda_2 f'(0) + \lambda_3 f'(h)\right)$$

   for every polynomial $p = f$ of degree less or equal to 3.

   Let $f \in C^1([a,b])$. Subsequently derive a quadrature rule $\tilde{I}_h(f)$ for $\int_a^b f(x)\ dx$ based on the grid points

$$z_k = a + ih, \quad h = \frac{b-a}{m}, \quad 0 \le i \le m,$$

   and show that $\tilde{I}_h(f)$ is of the following form

$$\tilde{I}_h(f) = T_h(f) - \frac{h^2}{12}\left(f'(b) - f'(a)\right).$$

   Hereby the summed up trapezoidal rule is denoted by $T_h(f)$ when applied to f. **3 points**

2. *Higher order of convergence for Newton-Cotes rules for even n.* Consider the n-th Newton-Cotes rule for the numerical quadrature of a real valued function on $[a,b]$, where the grid points are distributed equidistantly and $x_0 = a$, $x_n = b$. Prove that for every even n even polynomials of degree $n+1$ are integrated exactly.
   Hint: Consider the error of the according interpolation polynomial of degree n, for an arbitrary polynomial of degree $n+1$. Subsequently prove the function in this error to be odd with respect to the midpoint of the interval. **4 points**

3. *Bound on the order of quadrature rules.* Prove the following statement. There is no quadrature rule

$$\tilde{I}(f) = \sum_{i=0}^{n} \lambda_i f(x_i), \quad x_i \in [a,b],\ x_i \neq x_j \text{ for } i \neq j,$$

   integrating all polynomials of degree $2n+2$ on $[a,b]$ exactly.
   Hint: Find a polynomial, such that for every choice of the weights, the polynomial is not integrated exactly. **2 points**

4. *Programming exercise, Spline interpolation* Implement a program for interpolation using cubic splines. Hereby one can use the formulas from the lecture. The program should compute the cubic spline for the nodes and nodal values given by the following table.

x_k	-7	-4	-3	-2	-1	0	2	4	7	10
y_k	4	-3	-7	0	2	8	6	-4	5	0

Furthermore it should depict the spline graphically. The two additional necessary conditions are given by

$$s''(-7) = s''(10) = 0.$$

Hint: If you use the formulas you have to consider the indexing scheme of the programming language of your choice (i.e. if it starts to count from zero or one). To draw the spline one computes approximately ten equidistant nodes of the spline on each subinterval and use these values as for the figure. **4 points**

The exercises should be solved in groups of two students. They have to be submitted until Sie **Monday, 10.06.2024, 10:00**, either in the box of the tutor or electronically via whiteboard.