Numerical Analysis: Midterm

(**40 marks**)

Urbain Vaes

November 15, 2024

Question 1 (Floating point arithmetic, **10 marks**). True or false?

1. Let $(\bullet)_2$ denote binary representation. It holds that $(0.1011)_2 + (0.0101)_2 = 1$.
2. Let $(\bullet)_3$ denote base 3 representation. It holds that $(1000)_3 \times (0.002)_3 = 2$.
3. A natural number with binary representation $(b_4 b_3 b_2 b_1 b_0)_2$ is even if and only if $b_0 = 0$.
4. In Julia, `Float64(.4) == Float32(.4)` evaluates to `true`.
5. Machine addition $\widehat{+}$ is a commutative operation. More precisely, given any two double-precision floating point numbers $x \in \mathbf{F}_{64}$ and $y \in \mathbf{F}_{64}$, it holds that $x \mathbin{\widehat{+}} y = y \mathbin{\widehat{+}} x$.
6. Let $\mathbf{F}_{32}$ and $\mathbf{F}_{64}$ denote respectively the sets of single and double precision floating point numbers. It holds that $\mathbf{F}_{32} \subset \mathbf{F}_{64}$.
7. In Julia, `eps(Float16)` returns the smallest strictly positive number that can be represented exactly in the `Float16` format.
8. Let $\mathbf{F}_{64}$ be the set of double precision floating point numbers. For any $x \in \mathbf{R}$ such that $x \in \mathbf{F}_{64}$, it holds that $x + 1 \in \mathbf{F}_{64}$.
9. Let $x \in \mathbf{R}$ and $y \in \mathbf{R}$ be two numbers that are exactly representable in the `Float64` format. Then $x \mathbin{\widehat{+}} y = x + y$: machine addition is exact in this case.
10. It holds that $(0.\overline{2200})_3 = (0.9)_{10}$.

⚙ **Question 2** (Interpolation and approximation, **10 marks**). Are the following assertions true or false? Throughout this exercise, we use the notation $x_i^n = i/n$. The notation $\mathbf{P}(n)$ denotes the set of polynomials of degree less than or equal to n. We proved in class that, for any function $u\colon \mathbf{R} \to \mathbf{R}$ and for all $n \in \mathbf{N}_{>0}$, there exists a unique polynomial $p_n \in \mathbf{P}(n)$ such that

$$\forall i \in \{0, \dots, n\}, \qquad p_n(x_i^n) = u(x_i^n). \tag{1}$$

1. If u is not the zero function, then the degree of p_n is exactly n.

2. If $u(x) = 2x + 1$, then $p_n = u$ for all $n \in \{1, 2, 3, \dots\}$.

3. Fix $u(x) = 1 + \sin(57\pi x)$. Then $p_3(x) = 1$.

4. Fix $u(x) = (2x - 1)^3$. Then $p_2(x) = 2x - 1$.

5. Fix $n \in \mathbf{N}_{>0}$ and suppose that $u\colon \mathbf{R} \to \mathbf{R}$ is a smooth function. There exists a constant $K > 0$ independent of x such that

$$\forall x \in \mathbf{R}, \qquad u(x) - p_n(x) = K \prod_{i=0}^{n} (x - x_i^n)\,.$$

6. It holds that

$$\forall x \in [0, 1], \qquad \Bigl|(x - x_0^n) \dots (x - x_n^n)\Bigr| \leqslant n! \left(\frac{1}{n}\right)^{n+1}.$$

7. In Julia, assuming that n and the function u have already been defined, the following code enables to calculate the interpolating polynomial p_n of u:

```julia
using Polynomials
# Assume `n=5` and `u` have already been defined
x = LinRange(0, 1, n + 1)
p = fit(x, u)
```

8. Let Δ denote the finite difference operator: for a function $f\colon \mathbf{R} \to \mathbf{R}$, the function $\Delta f\colon \mathbf{R} \to \mathbf{R}$ is defined as

$$\Delta f(x) = f(x + 1) - f(x).$$

Then $f \in \mathbf{P}(n)$ if and only if $\Delta^{n+1} f = 0$. Here Δ^{n+1} denotes the composition of $n + 1$ applications of the operator Δ.

9. In Julia, the following code enables to fit the data `x` and `y` by a straight line.

```julia
using Polynomials
x = [1, 2, 3, 4]
y = [16, 9, 4, 1]
p = fit(x, y, 1)
```

10. Using Chebyshev instead of equidistant points can improve on the quality of the interpolation.

⚙ **Question 3** (Numerical integration, **10 marks**). The Gauss–Legendre quadrature formula with n nodes is an approximate integration formula of the form

$$I(u) := \int_{-1}^{1} u(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} w_i\, u(x_i) =: \widehat{I}_n(u), \tag{2}$$

which is exact when u is a polynomial of degree less than or equal to $2n-1$. (Note that the nodes are here numbered starting from 1.)

1. (**5 marks**) Find the nodes and weights of the Gauss–Legendre rule with $n = 3$ nodes, without using any formula (unless you prove it beforehand).

   Hint: Recall that a necessary and sufficient condition in order for (2) to be satisfied for any polynomial $p \in \mathbf{P}(5)$ is that

   $$\int_{-1}^{1} x^d\,\mathrm{d}x = \sum_{i=1}^{n} w_i x_i^d, \qquad \text{for all } d \in \{0,1,2,3,4,5\}.$$

   Furthermore, given the symmetry of the problem, it is reasonable to look for a solution of the following form, which enables to reduce the number of unknowns.

   $$(x_1, x_2, x_3, w_1, w_2, w_3) = (-x, 0, x, w_1, w_2, w_1).$$

2. (**5 marks**) Are the following assertions true of false :

   - The degree of precision of the composite trapezium rule is 2.
   - The composite Simpson rule can be used to integrate exactly a quadratic polynomial.
   - The degree of precision of the following rule is 1:

     ```
     function my_integrate(f, a, b)
         x = LinRange(a, b, 100)
         h = x[2] - x[1]
         return h * sum(f, x[1:end-1])
     end
     ```

   - The degree of precision of the following integration rule is 2:

     $$\int_{-1}^{1} f(x)\,\mathrm{d}x \approx 2f(0) + \frac{1}{3}f''(0).$$

   - Suppose that $u\colon \mathbf{R} \to \mathbf{R}$ is a smooth function, and let $\widehat{I}_n(u)$ denote an approximation of the integral $I(u) := \int_{-1}^{1} u(x)\,\mathrm{d}x$ by the composite trapezium approximation with n points. Let

     $$\widehat{J}_n(u) = 2\widehat{I}_{2n}(u) - \widehat{I}_n(u).$$

     It holds that

     $$\lim_{n\to\infty} n^2 \left| I(u) - \widehat{J}_n(u) \right| = 0.$$

💻 **Computer exercise 1** (Interpolation, **10 marks**). Consider the following data:

Time (hours)	**Temperature (°C)**
6	10.5
9	15.0
12	20.2
15	25.1
18	22.8
21	17.4

Table 1: Recorded temperatures at different times of the day.

We wish to approximate the temperature as a smooth function of time. To this end, calculate the interpolating polynomial p_{int}, as well as the best quadratic polynomial approximation p_{app} (in the sense that the sum of square errors is minimized). You may use the `Polynomials` library. Plot on the same graph:

- The data points using `scatter`;
- The polynomial p_{int} interpolating the data points;
- The quadratic polynomial p_{app} that best approximates the data, in the sense of least squares.

💻 **Computer exercise 2** (Numerical integration, **10 marks**). Boole's integration rule reads

$$\int_{-1}^{1} u(x)\,\mathrm{d}x \approx \frac{7}{45}u(-1) + \frac{32}{45}u\left(-\frac{1}{2}\right) + \frac{12}{45}u\left(0\right) + \frac{32}{45}u\left(\frac{1}{2}\right) + \frac{7}{45}u(1).$$

- Write a function `comp_boole(u, a, b, N)`, which returns an approximation of the integral

  $$I(u) = \int_a^b u(x)\,\mathrm{d}x$$

  obtained by partitioning the integration interval $[a, b]$ into N cells, and applying Boole's rule within each cell.

- Take $u(x) = \cos(x)$, $a = -1$ and $b = 1$. Plot the evolution of the error for $N \in \{1, \ldots, 200\}$.

- Estimate the order of convergence with respect to N, i.e. find α such that

  $$|\widehat{I}_N - I| \propto CN^{-\alpha},$$

  where I denotes the exact value of the integral and $\widehat{I}_N$ denotes its approximation by the composite Boole's rule. Use the method you prefer in order to find α. You can, for example, use the function `fit` from the `Polynomials` package to find a linear approximation of the form

  $$\log|\widehat{I}_N - I| \approx \log(C) - \alpha \log(N).$$