Numerical Analysis: Final Exam

(**50 marks**, only the 4 best questions count)

Urbain Vaes

December 2024

Academic integrity pledge

☐ I certify that I will not give or receive any unauthorized help on this exam, and that all work will be my own. (Tick ✓ or copy the sentence on your answer sheet).

Question 1 (Floating point arithmetic, **10 marks**).

1. **(T/F)** Let $(\bullet)_2$ denote base 2 representation. It holds that

$$3 \times (0.0101)_2 = (0.1111)_2.$$

2. **(T/F)** Does the following equality hold? Explain your reasoning.

$$(0.\overline{011})_2 = \frac{3}{4}.$$

3. **(T/F)** In Julia, `Float64(x) == Float32(x)` is `true` if `x` is a rational number.

4. **(T/F)** The value of the machine epsilon for the double precision format is the same in Julia and Python.

5. **(T/F)** The spacing (in absolute value) between successive double-precision (`Float64`) floating point numbers is equal to the machine epsilon.

6. **(T/F)** All the natural numbers can be represented exactly in the double precision floating point format `Float64`.

7. **(T/F)** Machine addition in the `Float64` format is associative but not commutative.

8. **(T/F)** In Julia, let `f(x) = (x == x/100.0) ? x : f(x/100.0)`. Then `f(a)` returns `0.0` for all finite number `a` representable in the `Float64` format.

9. (**1 mark**) In Julia `exp(eps()) == 1 + eps()` evaluates to `true`. Explain briefly why.

10. (**1 mark**) In Julia `sqrt(1 + eps()) == 1 + eps()` evaluates to `false`. Explain briefly why.

Question 2 (Interpolation, **10 marks**).

1. **(T/F)** The only polynomial p of degree at most 3 such that $p(-1) = p(0) = p(1) = 1$ is the constant polynomial $p(x) = 1$.

2. **(T/F)** In polynomial interpolation, using Chebyshev nodes can help reduce the interpolation error compared to using equidistant nodes.

3. **(2 marks)** Let $S(n) = \sum_{i=1}^{n} i$. Given that $S(n)$ is a quadratic polynomial, calculate the expression of $S(n)$ by interpolation. Include the details of your calculation.

4. **(T/F)** Given $x_0 < x_1 < x_2$ and $y_0, y_1, y_2 \in \mathbb{R}$, the unique polynomial passing through these data points is given by

$$p(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} y_0 + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} y_1 + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} y_2.$$

5. **(T/F)** Let $f(x) = \exp(5x)$, and for any $n \in \mathbb{N}$, let $f_n \in \mathcal{P}_n$ denote the polynomial interpolating f at $n + 1$ equidistant points $-1 = x_0 < x_1 < \ldots < x_n = 1$. Then

$$\lim_{n \to \infty} \left(\max_{-1 \leqslant x \leqslant 1} \left| f(x) - f_n(x) \right| \right) = 0.$$

6. **(2 marks)** Given $x_0 < \ldots < x_n$ and $y_0, \ldots, y_n \in \mathbb{R}$, prove that the constant polynomial p that minimizes the expression $\sum_{i=0}^{n} |y_i - p(x_i)|^2$ is given by

$$p(x) = \frac{1}{n+1} \sum_{i=0}^{n} y_i.$$

7. **(2 marks)** We wish to find a, b, c such that the function $f(x) := a\cos(x) + b\sin(x) + c$ interpolates the data points $(0, 0), (1, 1), (2, 2)$. Complete on paper the following code for calculating a, b, c.

```
x = [0.0, 1.0, 2.0]
y = [0.0, 1.0, 2.0]
# Your code below
```

Question 3 (Integration, **10 marks**).

1. **(T/F)** The degree of precision of the following quadrature rule is 1:

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

2. **(T/F)** The closed Newton–Cotes rule with n points is exact for all linear polynomials.

3. **(T/F)** The degree of precision of the following quadrature rule is 5:

$$\int_{-1}^{1} f(x)\,\mathrm{d}x \approx 2f(0) + \frac{2}{3}f''(0) + \frac{2}{5}f^{(4)}(0).$$

4. **(T/F)** Suppose that $f \in C^\infty[a,b]$ and let $I_n[f]$ denote the approximate integral of f using the composite trapezium rule with n integration points. Then it holds that

$$\lim_{n\to\infty} \Big| I[f] - I_n[f] \Big| = 0, \qquad I[f] := \int_a^b f(x)\,\mathrm{d}x.$$

5. **(T/F)** Suppose that $f \in C^\infty[a,b]$ and let $I_n[f]$ denote the approximate integral of f using the composite trapezium rule with n integration points. Then it holds that

$$\lim_{n\to\infty} n^2 \Big| I[f] - I_n[f] \Big| < \infty.$$

6. **(2 marks)** Calculate weights w_1, w_2 so that the degree of precision of the following rule is 1:

$$\int_{-1}^{1} f(x)\,\mathrm{d}x = w_1 f(0) + w_2 f(1).$$

7. **(2 marks)** Implement the composite trapezium rule with n points:

```
function I_approx(f, a, b, n)
    x = LinRange(a, b, n)
```

8. **(1 mark)** The following code implements the midpoint rule with n points, but there is an error. Spot and correct the error.

```
function I_approx(f, a, b, n)
    x = LinRange(a, b, n)
    h = (b - a) / n
    return h * sum(f, x .+ h/2)
end
```

Question 4 (Iterative method for linear systems, **10 marks**). Assume that $\mathsf{A} \in \mathbf{R}^{n \times n}$ is an *invertible* matrix and that $\boldsymbol{b} \in \mathbf{R}^n$. We wish to solve the linear system

$$\mathsf{A}\boldsymbol{x} = \boldsymbol{b}. \tag{1}$$

1. (**3 marks**) We first consider a basic iterative method where each iteration is of the form

$$\mathsf{M}\boldsymbol{x}_{k+1} = \mathsf{N}\boldsymbol{x}_k + \boldsymbol{b}. \tag{2}$$

Here $\mathsf{A} = \mathsf{M} - \mathsf{N}$ is a splitting of A such that M is nonsingular, and $\boldsymbol{x}_k \in \mathbf{R}^n$ denotes the k-th iterate of the numerical scheme. Let $\boldsymbol{e}_k := \boldsymbol{x}_k - \boldsymbol{x}_*$, where $\boldsymbol{x}_*$ is the exact solution to (1). Prove that the error satisfies

$$\forall k \in \mathbf{N}, \qquad \boldsymbol{e}_{k+1} = \mathsf{M}^{-1}\mathsf{N}\boldsymbol{e}_k.$$

2. **(T/F)** If $\|\mathsf{M}^{-1}\mathsf{N}\|_2 < 1$, then the iterative method (2) is convergent.
3. **(T/F)** The Gauss–Seidel iterative method is a particular case of (2).
4. (**3 marks**) Write down on paper a few iterations of the Jacobi method when

$$\mathsf{A} = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}, \qquad \boldsymbol{b} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \qquad \boldsymbol{x}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Is the method convergent?

5. (**2 marks**) Suppose that A is lower triangular. Implement the forward substitution algorithm for solving (1) in this case.

```
A = [1.0 0.0 0.0; 1.0 2.0 0.0; 1.0 2.0 3.0]
b = [1.0; 3.0; 6.0]
```

Question 5 (Nonlinear equations, **10 marks**). We consider the following iterative method for calculating $\sqrt[3]{2}$:

$$x_{k+1} = F(x_k), \qquad F(x) := \omega x + (1-\omega)\frac{2}{x^2}, \tag{3}$$

with $\omega \in [0,1]$ a fixed parameter.

1. (**2 marks**) Show that $x_* := \sqrt[3]{2}$ is a fixed point of the iteration (3).

2. (**4 marks**) Write down a Julia program based on the iteration (3) for calculating $\sqrt[3]{2}$. Use an appropriate stopping criterion that does not require to know the value of $\sqrt[3]{2}$.

3. **(T/F)** The iterative method (3) converges to $\sqrt[3]{2}$ for any $\omega \in [0,1]$.

4. **(T/F)** The secant method is usually faster than the Newton–Raphson method.

5. **(T/F)** The following iteration converges to $\sqrt[3]{2}$ for all initialization x_0:

$$x_{k+1} = x_k - \frac{x_k^3 - 2}{10}.$$

6. (**1 mark**) Illustrate a few iterations of the Newton–Raphson iteration for finding $\sqrt[3]{2}$ on the following figure (see next page), where the function $f(x) = x^3 - 2$ is plotted.

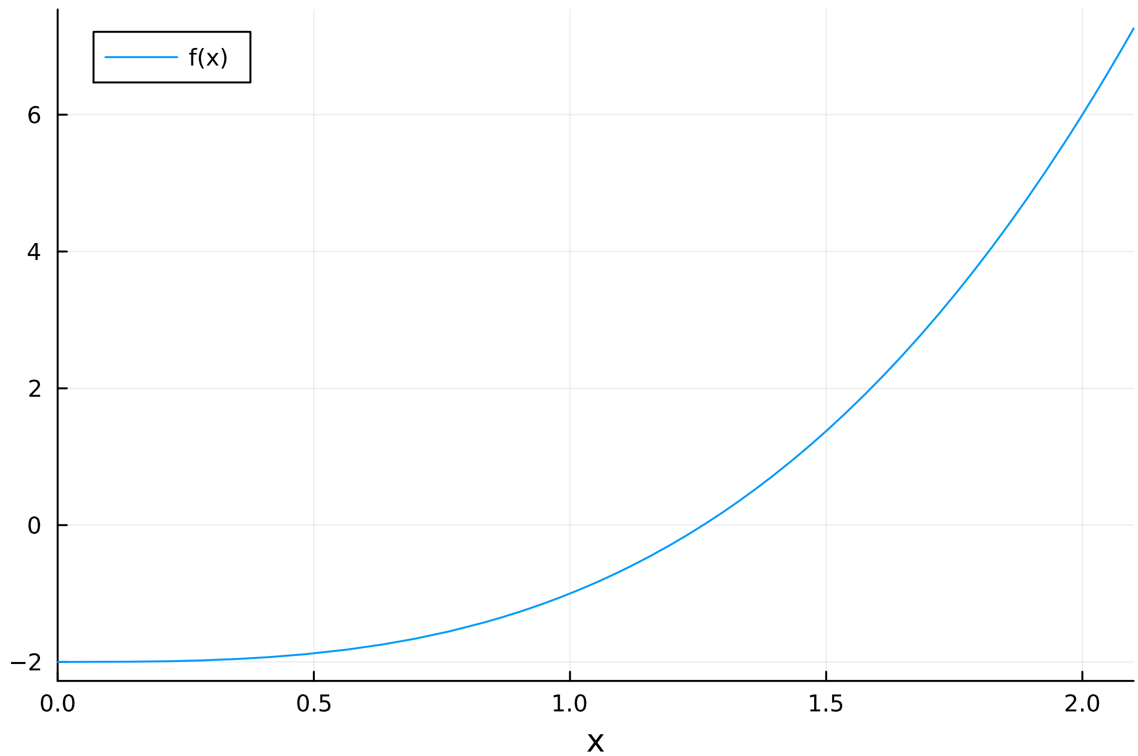

Figure 1: You can use this figure to illustrate the Newton–Raphson method.