NYU | PARIS

MATH-UA 9252 · Numerical Analysis

Midterm exam — Solutions

Fall $(11111101010)_2$ · Urbain Vaes

Name: _____

Pledge of academic integrity

I pledge that I will complete this exam on my own, without giving or receiving help from anyone, and without using any notes, devices or other materials not explicitly allowed.

Signature: _____

Exercise	Topic	Marks	Score
Question 1	Floating point arithmetic	15	
Implementation exercise 1	Julia basics	10	
Question 2	Interpolation	10	
Question 3	Interpolation (open-ended)	5	
Implementation exercise 2	Approximation	5	
Question 4	Numerical integration	10	
Question 5	Numerical integration (open-ended)	5	
Implementation exercise 3	Numerical integration	10	
Question 6 (bonus)	Numerical integration (open-ended)	+5	
	Total	**70**	

Question 1 Floating point arithmetic — 15 marks

True or false? Answer in the grid below.

Answer grid	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	B1	B2
True	■	■	□	■	□	■	■	■	□	■	■	□	□	■	■	■	□
False	□	□	■	□	■	□	□	□	■	□	□	■	■	□	□	□	■

1. Let $(\bullet)_2$ denote binary representation. Then $(0.0111)_2 + (0.1001)_2 = (1.0)_2$.
2. Let ε_{32} denote the machine epsilon for the `Float32` format, i.e. `eps(Float32)`. Then the number 4 is representable exactly in this format, and the next representable number is $4 + 4\varepsilon_{32}$.
3. It holds that $(100)_2 \times (0.0110)_2 = (11)_2$.
4. In Julia, `Float64(.75) == Float32(.75)` evaluates to `true`.
5. In Julia, the expression `0.1 + 0.2 == 0.3` evaluates to `true`.
6. It holds that $\left(0.\overline{011}\right)_2 = \frac{3}{7}$.
7. Machine multiplication $\hat{*}$ is a *commutative* operation: for any two numbers $x, y \in \mathbf{F}_{64}$, it holds that $x \,\hat{*}\, y = y \,\hat{*}\, x$.
8. There are exactly as many `Float64` numbers in the interval $[1, 2)$ as in the interval $[2, 4)$.
9. Let ε denote the machine epsilon of a binary floating point format. Then the spacing between successive floating point numbers in the interval $\left[\frac{1}{2}, 1\right)$ is equal to 2ε.
10. In Julia, the expression `1 + eps()/4 == 1` evaluates to `true`.
11. It holds that $(101.101)_2 = 5.625$.
12. A real number has a finite representation in base 2 if and only if it is a rational number.
13. The machine epsilon of a floating point format is the smallest strictly positive number that can be represented exactly in the format.
14. If `x` is a `Float16` number and `y` is a `Float64` number, then the result of `x + y` is a `Float64` number.
15. In Julia, the expression `eps()/3 + eps()/3 + 1 > 1` evaluates to `true`.

B1. **Bonus.** In Julia, the expression `sqrt(1 + 4eps()) == 1 + 2eps()` evaluates to `true`.

> **Solution.** **True.** With $\varepsilon = 2^{-52}$, the number $1 + 4\varepsilon$ is representable, and $\sqrt{1 + 4\varepsilon} = 1 + 2\varepsilon - 2\varepsilon^2 + \mathcal{O}(\varepsilon^3)$. This is much closer to the representable number $1 + 2\varepsilon$ than half the spacing $\varepsilon/2$, and `sqrt` is correctly rounded, so the result is exactly $1 + 2\varepsilon$.

B2. **Bonus.** In Julia, the expression `(1 + 2eps()/3) - 1 == 2eps()/3` evaluates to `true`.

> **Solution.** **False.** The number $1 + 2\varepsilon/3$ is closer to $1 + \varepsilon$ than to 1, so the addition is rounded to $1 + \varepsilon$. The subtraction is then exact and gives $\varepsilon \neq 2\varepsilon/3$.

Solution. Below, $\varepsilon = 2^{-52}$ denotes the machine epsilon of `Float64`.

1. **True:** $(0.0111)_2 + (0.1001)_2 = \frac{7}{16} + \frac{9}{16} = 1$.

2. **True:** $4 = 2^2 \times (1.0)_2$ is representable, and the spacing between successive `Float32` numbers in $[4, 8)$ is $2^2\varepsilon_{32}$.
3. **False:** $(100)_2 \times (0.0110)_2 = 4 \times \frac{3}{8} = \frac{3}{2} = (1.1)_2$.
4. **True:** $0.75 = (0.11)_2$ is exactly representable in both formats.
5. **False:** 0.1, 0.2 and 0.3 are not exactly representable, and `0.1 + 0.2` evaluates to `0.30000000000000004`.
6. **True:** if $x = \left(0.\overline{011}\right)_2$, then $2^3 x = 3 + x$, so $x = \frac{3}{7}$.
7. **True:** $x \mathbin{\hat{*}} y$ is the rounding of $xy = yx$.
8. **True:** each interval $\left[2^e, 2^{e+1}\right)$ contains 2^{52} numbers $2^e(1 + k\varepsilon)$, $k \in \{0, \ldots, 2^{52} - 1\}$.
9. **False:** the spacing in $[1, 2)$ is ε, so the spacing in $\left[\frac{1}{2}, 1\right)$ is $\varepsilon/2$.
10. **True:** $1 + \varepsilon/4$ is closer to 1 than to $1 + \varepsilon$, so it is rounded to 1.
11. **True:** $(101.101)_2 = 4 + 1 + \frac{1}{2} + \frac{1}{8} = 5.625$.
12. **False:** the expansion is finite if and only if the number is of the form $m/2^k$; for example, $\frac{1}{3} = \left(0.\overline{01}\right)_2$.
13. **False:** the machine epsilon is the distance between 1 and the next representable number; much smaller positive numbers, such as 2^{-1074} in `Float64`, are representable.
14. **True:** `x` is promoted to `Float64` before the addition.
15. **True:** the sum is evaluated from left to right; `eps()/3 + eps()/3` is approximately $2\varepsilon/3$, which is larger than $\varepsilon/2$, so adding 1 rounds to $1 + \varepsilon > 1$.

Implementation exercise 1 Julia basics — 10 marks

Let $f(x) = \mathrm{e}^{-x}\cos(\pi x)$. Write Julia code performing the following tasks, *without using any loop* (comprehensions are allowed). You may assume that the `Plots` library is loaded.

▶ **Functions.** Define a Julia function `f` that returns $f(x)$ for a number `x`.

Solution.

```
f(x) = exp(-x) * cos(pi * x)
```

▶ **Broadcasting.** Let `x = LinRange(0, 2, 101)`. Using the function `f`, construct the vector `y` containing the values of f at the points of `x`.

Solution.

```
y = f.(x)
```

▶ **Slicing.** Using `x` and `y`, construct the vector of the 100 finite difference approximations $\frac{y_{i+1}-y_i}{x_{i+1}-x_i}$ of the derivative f'.

Solution.

```
d = (y[2:end] - y[1:end-1]) ./ (x[2:end] - x[1:end-1])
```

Alternatively, `d = diff(y) ./ diff(x)`.

▶ **Matrices.** Let `z = [0, 0.5, 1.5, 2]`. Construct the matrix $\mathsf{V} \in \mathbf{R}^{4\times 4}$ whose entry on the i-th row and j-th column is $V_{ij} = z_i^{j-1}$.

Solution.

```
V = z .^ (0:3)'   # or [zi^(j-1) for zi in z, j in 1:4]
```

▶ **Plotting.** Plot on the same graph, over the interval $[0, 2]$, the function f together with the function $g(x) = \mathrm{e}^{-x}$, ideally with a legend indicating which curve is which.

Solution.

```
plot(f, 0, 2, label="f")
plot!(x -> exp(-x), 0, 2, label="g")
```

Question 2 Interpolation — 10 marks

Are the following statements true or false? Justify your answers. Recall that $\mathcal{P}_d$ denotes the set of polynomials of degree at most d.

1. Assume that $x_0 < x_1 < x_2$ and y_0, y_1, y_2 are given real numbers. Then there exists a *unique* polynomial $p \in \mathcal{P}_3$ such that $p(x_i) = y_i$ for all $i \in \{0, 1, 2\}$. Prove or disprove from first principles, i.e. without merely invoking a theorem from class.

Solution. **False.** *Existence:* the polynomial

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

belongs to $\mathcal{P}_2 \subset \mathcal{P}_3$, and $p(x_i) = y_i$ because each fraction equals 1 at one node and 0 at the other two. *Non-uniqueness:* for any $c \in \mathbf{R}$, the polynomial $p(x) + c(x-x_0)(x-x_1)(x-x_2) \in \mathcal{P}_3$ takes the same values at the nodes. Uniqueness only holds in $\mathcal{P}_2$.

2. Assume that $x_0 < x_1 < \cdots < x_n$, and let $\ell_0, \ldots, \ell_n \in \mathcal{P}_n$ denote the associated Lagrange basis polynomials, so that $\ell_i(x_j) = 1$ if $i = j$ and $\ell_i(x_j) = 0$ otherwise. Then it holds that

$$\forall x \in \mathbf{R}, \quad \sum_{i=0}^{n} \ell_i(x) = 1.$$

Solution. **True.** The polynomial $q = \sum_{i=0}^n \ell_i - 1$ belongs to $\mathcal{P}_n$ and satisfies $q(x_j) = \sum_{i=0}^n \ell_i(x_j) - 1 = 1 - 1 = 0$ for all $j \in \{0, \ldots, n\}$. A nonzero polynomial of degree at most n has at most n roots, so q, which has $n+1$ distinct roots, is the zero polynomial.

3. Let $p \in \mathcal{P}_d$ be a polynomial of degree $d \geq 2$, and let $q : \mathbf{R} \to \mathbf{R}$ be given by

$$q(x) = p(x+2) - 2p(x+1) + p(x).$$

Then it holds that $q \in \mathcal{P}_{d-2}$.

Hint: it may be useful to express q using the difference operator $\Delta f(x) := f(x+1) - f(x)$.

Solution. **True.** By definition, $\Delta(\Delta p)(x) = \Delta p(x+1) - \Delta p(x) = p(x+2) - 2p(x+1) + p(x)$, so $q = \Delta^2 p$. If $p(x) = cx^k + r(x)$ with $c \neq 0$, $k \geq 1$ and $r \in \mathcal{P}_{k-1}$, then

$$\Delta p(x) = c\big((x+1)^k - x^k\big) + \Delta r(x) = ckx^{k-1} + (\text{terms of degree} \leq k-2),$$

so Δ lowers the degree by exactly one (and maps constants to 0). Applying Δ twice gives $q = \Delta^2 p \in \mathcal{P}_{d-2}$, and in fact q has degree exactly $d-2$.

4. Let $f \in C^\infty[a,b]$, and let f_n denote the *piecewise linear* interpolant of f on a uniform mesh of $[a,b]$ with n cells, of size $h_n = \frac{b-a}{n}$. Then it holds that

$$\lim_{n\to\infty} n\left(\max_{x\in[a,b]} |f(x) - f_n(x)|\right) = 0.$$

Solution. **True.** Let $x_i = a + ih_n$ denote the mesh points, and fix $x \in [x_i, x_{i+1}]$. By the interpolation error formula, there exists $\xi \in [x_i, x_{i+1}]$ such that

$$f(x) - f_n(x) = \frac{f''(\xi)}{2}(x - x_i)(x - x_{i+1}).$$

Since $|(x - x_i)(x - x_{i+1})| \le h_n^2/4$, it follows that

$$\max_{x\in[a,b]} |f(x) - f_n(x)| \le \frac{h_n^2}{8} \max_{x\in[a,b]} |f''(x)| = \frac{(b-a)^2}{8n^2} \max_{x\in[a,b]} |f''(x)|,$$

so $n \max_{x\in[a,b]} |f(x) - f_n(x)| \le C/n \to 0$. (The statement would be false with n^2 instead of n: for $f(x) = x^2$, the error is exactly $h_n^2/4$.)

Question 3 Interpolation (open-ended) — 5 marks

A continuous function $f : [a, b] \to \mathbf{R}$ can be approximated either by a single polynomial of degree n interpolating f at $n + 1$ nodes, or by a piecewise polynomial of fixed low degree (for example, piecewise linear) interpolating f on a mesh of $[a, b]$ with n cells. Compare these two approaches. For example, how does the interpolation error behave as n increases in each case, and on which properties of f and of the nodes does this behavior depend? In which situations would you prefer one approach over the other?
Illustrate your answer with examples of your choice, but go beyond merely stating formulas.

Solution. Many answers are possible; a good answer contains elements such as the following.

- *Global interpolation.* The error is $f(x) - p_n(x) = f^{(n+1)} \frac{\xi}{(n+1)!} \prod_{i=0}^{n} (x - x_i)$, so it depends both on the growth of the derivatives of f and on the nodes. For equidistant nodes, the interpolant may diverge as $n \to \infty$ even for smooth f (Runge phenomenon, e.g. $f(x) = 1/(1 + 25x^2)$ on $[-1, 1]$). With Chebyshev nodes, which minimize $\max|\prod_i (x - x_i)|$, convergence is very fast (exponential) for analytic functions.
- *Piecewise interpolation.* The error is local: for piecewise linear interpolation and $f \in C^2$, it is bounded by $\frac{h^2}{8} \max|f''|$ with $h = (b - a)/n$, so the interpolant converges for any choice of n, at a rate set by the local degree. It also converges (more slowly) for merely continuous f, and an irregular behavior of f in one region only affects nearby cells.
- *Comparison.* Global interpolation can be very accurate with few nodes for smooth (analytic) functions, provided the nodes are chosen well, and gives a single smooth formula. Piecewise interpolation is more robust: it works with equidistant data, for functions of limited smoothness, and for large n, but converges only algebraically and its derivatives are discontinuous.

Implementation exercise 2 Approximation — 5 marks

Write Julia code that computes and plots the polynomial $p \in \mathcal{P}_2$ that best approximates the following data in the least squares sense, i.e. that minimizes

$$\sum_i |p(x_i) - y_i|^2,$$

for the following data:

x_i	0	1	2	3	4	5
y_i	1.1	1.9	5.2	9.8	17.1	26.0

The plot should display both the data points and the graph of the polynomial over an appropriate range. Do not use any other library than the ones already imported.

Solution. The coefficients α of $p(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2$ solve the normal equations $\mathsf{A}^T\mathsf{A}\alpha = \mathsf{A}^T y$:

```julia
using LinearAlgebra
using Plots
x = [0, 1, 2, 3, 4, 5]
y = [1.1, 1.9, 5.2, 9.8, 17.1, 26.0]
A = [xi^j for xi in x, j in 0:2]
α = (A' * A) \ (A' * y)   # or A \ y
p(t) = α[1] + α[2] * t + α[3] * t^2
scatter(x, y, label="Data")
plot!(p, -0.5, 5.5, label="Least squares approximation")
```

This gives $p(x) \approx 1.068 - 0.053x + 1.009x^2$.

Question 4 Numerical integration — 10 marks

Are the following statements true or false? Justify your answers.

1. The degree of precision of the following quadrature rule is 2:

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

Solution. False. The rule is exact for 1 (it gives 2) and x^2 (it gives $\frac{2}{3}$), and by symmetry for x and x^3, for which both sides vanish. For x^4, it gives $\frac{2}{3} \neq \frac{2}{5}$. The degree of precision is therefore 3, not 2.

2. The degree of precision of the following quadrature rule is 5:

$$\int_{0}^{1} u(x)\,\mathrm{d}x \approx u\left(\frac{1}{2}\right) + \frac{1}{24}u''\left(\frac{1}{2}\right).$$

Solution. False. Write u in terms of $t = x - \frac{1}{2}$, so that the integral is over $t \in \left[-\frac{1}{2}, \frac{1}{2}\right]$ and the nodes are at $t = 0$. The rule is exact for 1, for t and t^3 (both sides vanish), and for t^2: $\int_{-\frac{1}{2}}^{\frac{1}{2}} t^2\,\mathrm{d}t = \frac{1}{12} = \frac{1}{24} \times 2$. For t^4, the rule gives 0 whereas $\int_{-\frac{1}{2}}^{\frac{1}{2}} t^4\,\mathrm{d}t = \frac{1}{80}$. The degree of precision is therefore 3, not 5.

3. Let $x_1 < \cdots < x_n$ be distinct nodes in $[-1, 1]$. If the quadrature rule

$$\int_{-1}^{1} u(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} w_i u(x_i)$$

has a degree of precision at least $n - 1$, then its weights are given by

$$w_i = \int_{-1}^{1} \ell_i(x)\,\mathrm{d}x, \qquad \ell_i(x) = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}, \qquad i = 1, \ldots, n.$$

Solution. True. Each ℓ_i belongs to $\mathcal{P}_{n-1}$, so the rule integrates it exactly. Since $\ell_i(x_j) = 1$ if $i = j$ and 0 otherwise,

$$\int_{-1}^{1} \ell_i(x)\,\mathrm{d}x = \sum_{j=1}^{n} w_j \ell_i(x_j) = w_i.$$

4. Let $x_1, ..., x_n$ and $w_1, ..., w_n$ denote the nodes and weights of the n-point Gauss–Legendre rule on $[-1, 1]$. Then the n-point Gauss–Legendre rule on a general interval $[a, b]$ is given by

$$\int_a^b u(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} w_i u\left(a + \frac{b-a}{2}(x_i + 1)\right).$$

Solution. False. The nodes are mapped correctly, but the change of variables $x = a + \frac{b-a}{2}(t + 1)$ has Jacobian $\frac{b-a}{2}$, by which the weights must be multiplied. For example, with $u = 1$ the formula gives $\sum_i w_i = 2$ instead of $b - a$.

5. There exist nodes $x_1, x_2 \in [-1, 1]$ and weights $w_1, w_2 \in \mathbf{R}$ such that the following quadrature rule has a degree of precision equal to 4:

$$\int_{-1}^{1} u(x)\,\mathrm{d}x \approx w_1 u(x_1) + w_2 u(x_2).$$

Solution. False. For any nodes x_1, x_2, the polynomial $u(x) = (x - x_1)^2(x - x_2)^2$ belongs to $\mathcal{P}_4$, is nonnegative and not identically zero, so $\int_{-1}^{1} u(x)\,\mathrm{d}x > 0$, whereas the rule gives $w_1 u(x_1) + w_2 u(x_2) = 0$. With two nodes, the highest possible degree of precision is 3 (two-point Gauss–Legendre rule, with nodes $\pm 1/\sqrt{3}$ and weights 1).

Question 5 Numerical integration (open-ended) — 5 marks

Composite quadrature rules are obtained by partitioning the integration interval into N cells of size h and applying a simple quadrature rule within each cell. Discuss how the error of a composite rule behaves as $h \to 0$, and how this behavior is related to the degree of precision of the underlying simple rule.
Illustrate your answer with Simpson's rule, considering both the simple rule and its composite version, but go beyond merely stating formulas.

Solution. A good answer contains the following elements.

- If the simple rule has degree of precision d, then on a cell of size h it integrates exactly the Taylor polynomial of degree d of u at the center of the cell. The error on the cell is therefore bounded by the integral of the Taylor remainder, which is of order h^{d+1} times the length h of the cell, i.e. $\mathcal{O}\big(h^{d+2}\big)$, for $u \in C^{d+1}$.
- Summing over the $N = (b-a)/h$ cells, the error of the composite rule is $\mathcal{O}\big(h^{d+1}\big)$.
- Simpson's rule is built on three points, so it is exact for $\mathcal{P}_2$ by construction, and also for x^3 by symmetry, but not for x^4: its degree of precision is $d = 3$. The error is $\mathcal{O}(h^5)$ per cell and $\mathcal{O}(h^4)$ for the composite rule, provided $u \in C^4$: doubling N divides the error by about 16.
- For less regular functions, the convergence is slower than predicted.

Implementation exercise 3 Numerical integration — 10 marks

Consider the following quadrature rule:

$$\int_{-1}^{1} u(x)\,\mathrm{d}x \approx \frac{1}{12}\left(11u\left(-\frac{3}{5}\right) + u\left(-\frac{1}{5}\right) + u\left(\frac{1}{5}\right) + 11u\left(\frac{3}{5}\right)\right).$$

▶ (**2 marks**) Write a function `simple_rule(u)` that returns the approximation of $\int_{-1}^{1} u(x)\,\mathrm{d}x$ given by this quadrature rule.

Solution.

```
function simple_rule(u)
    return (11u(-3/5) + u(-1/5) + u(1/5) + 11u(3/5)) / 12
end
```

▶ (**3 marks**) Use the function `simple_rule` to determine numerically the degree of precision of the rule, by applying it to the monomials $u(x) = x^d$ for $d \in \{0, 1, 2, \ldots\}$. What is the degree of precision?

Solution.

```
for d in 0:5
    exact = iseven(d) ? 2 / (d + 1) : 0.0
    println("d = $d: error = ", simple_rule(x -> x^d) - exact)
end
```

The error is zero for $d \leq 3$ but not for $d = 4$ (it is about -0.162), so the degree of precision is 3.

▶ (**5 marks**) Write a function `composite_rule(u, a, b, N)` that returns an approximation of the integral

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

obtained by partitioning the integration interval $[a, b]$ into N cells, and applying the quadrature rule within each cell. Your function should call the function `simple_rule` you wrote in the previous item.

Solution. On the cell with center m and size h, the change of variables $x = m + \frac{h}{2}t$ gives $\int_{m-h/2}^{m+h/2} u(x)\,\mathrm{d}x = \frac{h}{2}\int_{-1}^{1} u\big(m + \frac{h}{2}t\big)\,\mathrm{d}t$:

```
function composite_rule(u, a, b, N)
    h = (b - a) / N
    result = 0.0
    for i in 0:N-1
        m = a + (i + 1/2) * h
        result += h/2 * simple_rule(t -> u(m + h/2 * t))
    end
    return result
end
```

Question 6 (bonus) Numerical integration (open-ended) — 5 bonus marks

Let $\hat{I}(h)$ denote the approximation of an integral I given by a composite quadrature rule with step size h, and assume that

$$\hat{I}(h) = I + Ch^p + \mathcal{O}(h^q),$$

for some constant $C \neq 0$ independent of h and some exponents $q > p > 0$. Explain how Richardson extrapolation can be used to obtain a more accurate approximation from the two approximations $\hat{I}(h)$ and $\hat{I}(2h)$.
Illustrate your answer with the composite trapezoidal rule, but go beyond merely stating formulas.

Solution. By assumption, $\hat{I}(h) = I + Ch^p + \mathcal{O}(h^q)$ and $\hat{I}(2h) = I + 2^p Ch^p + \mathcal{O}(h^q)$. Eliminating Ch^p between these two equations gives

$$\frac{2^p \hat{I}(h) - \hat{I}(2h)}{2^p - 1} = I + \mathcal{O}(h^q),$$

which is more accurate than $\hat{I}(h)$ since $q > p$. The same idea gives an estimate of the error of $\hat{I}(h)$ that does not require I: $I - \hat{I}(h) \approx \frac{\hat{I}(h) - \hat{I}(2h)}{2^p - 1}$.

For the composite trapezoidal rule $T(h)$ and a smooth integrand, $p = 2$ and $q = 4$, so

$$\frac{4T(h) - T(2h)}{3} = I + \mathcal{O}(h^4).$$

This combination coincides with the composite Simpson rule. Applying the idea repeatedly to eliminate higher-order terms leads to Romberg's method. The method relies on the error expansion, which requires the integrand to be smooth enough; otherwise there is no gain.