LINMA2171 (2025–2026) — Numerical Analysis: Approximation, Interpolation, Integration

Homework 2

Deadline: Tuesday 4 November 2025 at 10:30am

Polynomial interpolation: Newton and Neville We would like to compute the polynomial $p(x)$ of degree $n-1$ which interpolates a function f at n equidistant nodes in the interval $[a,b]$, then evaluate $p(x)$ at m distinct points.

1. Implement the following interpolation algorithms:
   (a) Newton's interpolation using divided differences.
   (b) Neville's algorithm for the recursive evaluation of the interpolant at a given point.
2. For each method, analyze the computational complexity:
   (a) Coefficients computation (if applicable),
   (b) One-point evaluation,
   (c) Storage requirements. *Hint: it should depend on m (the number of evaluation points).*
3. Apply both methods to the following test functions on the interval $[-1,1]$:

$$f_1(x) = \cos(x), \qquad f_2(x) = \frac{1}{1+25x^2}.$$

   (a) Plot the resulting interpolant for f_1 using Neville's method with $n \in \{5, 10, 15\}$, and for f_2 using Newton's method with the same values $n \in \{5, 10, 15\}$. Use a fine grid of m points (e.g. $m = 200$) to plot the original function.
   (b) Compute and plot the interpolation error for both functions, for each n, using the ℓ^2-norm on a fine grid. Comment on your results. Would the results be the same if you swapped the methods (Newton on f_1, Neville on f_2)? Explain why.

Rational interpolation: Floater–Hormann We now consider an alternative interpolation scheme: the Floater–Hormann family of barycentric rational interpolants [1]. Given $n+1$ distinct nodes $x_0, \ldots, x_n$ and a function f, choose a blending degree $d \in \{0, \ldots, n\}$. For each $i \in \{0, \ldots, n-d\}$, let p_i be the polynomial of degree at most d interpolating f at the $d+1$ nodes $\{x_i, \ldots, x_{i+d}\}$, i.e., $p_i(x_{i+j}) = f(x_{i+j})$ for $j = 0, \ldots, d$. We define

$$r(x) := \frac{\displaystyle\sum_{i=0}^{n-d} \lambda_i(x)\, p_i(x)}{\displaystyle\sum_{i=0}^{n-d} \lambda_i(x)},$$

where

$$\lambda_i(x) := \frac{(-1)^i}{\displaystyle\prod_{j=0}^{d}(x - x_{i+j})}.$$

It can be shown that r interpolates the $n+1$ data points $(x_i, f(x_i))$ for $i = 0, \ldots, n$. We shall evaluate $r(x)$ at m distinct points.

1. Prove that this interpolant is a rational function. What are the degrees of the numerator and the denominator? What does the interpolant become when the target function is a polynomial of degree at most d?

2. Prove that the Floater–Hormann interpolant can be written in barycentric form:

$$r(x) = \frac{\displaystyle\sum_{i=0}^{n} \frac{w_i}{x - x_i} f(x_i)}{\displaystyle\sum_{i=0}^{n} \frac{w_i}{x - x_i}},$$

with weights $w_0, \ldots, w_n$ given by

$$w_i = \sum_{j=\max(0,\, i-d)}^{\min(i,\, n-d)} (-1)^j \prod_{\substack{k=j \\ k \neq i}}^{j+d} \frac{1}{x_i - x_k}, \qquad i = 0, \ldots, n.$$

3. Implement the Floater–Hormann method to interpolate a function at n equidistant nodes in the interval $[a, b]$.

4. Apply the method to the following test functions on $[-1, 1]$:

$$f_1(x) = \cos(x), \qquad f_2(x) = \frac{1}{1 + 25x^2}.$$

   (a) Plot the Floater–Hormann interpolants for $n \in \{5, 10, 15\}$ and several values of d (e.g. $d = 0, 3, 5, 8$). Use a fine grid of m points (e.g. $m = 200$) to plot the original function.

   (b) Compute the interpolation error using the ℓ^2-norm on a fine grid for different pairs (n, d). Can you identify an "optimal" d (possibly depending on n)?

   (c) Compare your results with Newton's and Neville's polynomial interpolations and discuss.

Practical information.

Meetings with the TA: by appointment.
Writing : You must do all the writing (report and code) ***individually***. Never share your production, but you are allowed, and even encouraged, to exchange ideas on how to address the homework. If you use an LLM, mention it in the report and explain for which purpose it was used. However, we believe that this assignment will be more useful to you—and the report more pleasant to read for us—if you use LLMs sparingly, if at all.
Questions: Feel free to contact me at antoine.springael@uclouvain.be to ask questions or to set up a meeting.
Submission: Using Moodle. Keep in mind that the deadline is automatically enforced, using the clock of Moodle!
Python/Octave/Matlab/Julia codes: Please don't copy your code to your report and add the `.py`/`.ipynb`/`.m`/`.jl` files as attachments. Add a script named `run` that reproduces all your results.
Language: Reports in French are accepted without penalty. However, English is strongly encouraged. The quality of the English will not impact the grade, provided that the text is intelligible.

Deadline: Tuesday 4 November 2025 at 10:30am

References

[1] Michael S. Floater and Kai Hormann. "Barycentric rational interpolation with no poles and high rates of approximation". In: *Numerische Mathematik* 107.2 (2007), pp. 315–331.