UCLouvain

LINMA2171 - Numerical Analysis: Approximation, Interpolation, Integration

Homework 2

Student
Lucas Ahou (35942200)

Teacher
P-A. Absil

October 30, 2025

Polynomial interpolation: Newton & Neville

2. Computational complexities

Newton's divided differences

a) The coefficients for the Newton's algorithm are the divided differences :

$$a_j := [y_0, ..., y_j]$$

where :

$$[y_k] := y_k$$
$$[y_k, ..., y_{k+j}] := \frac{[y_{k+1}, ..., y_{k+j}] - [y_k, ..., y_{k+j-1}]}{x_{k+j} - x_k}$$

with y_k the points we want to interpolate.
For each $[y_k, ..., y_{k+j}]$, we perform 2 substractions and 1 division. The number of elements to compute to get all the coefficients is $\frac{n(n-1)}{2}$. The total number of flops to compute all the coefficients is :

$$3 \cdot \frac{n(n-1)}{2} \sim \mathcal{O}(n^2)$$

b) Assuming all the coefficients are already computed, we simply need to compute:

$$p(x) = \sum_{j=0}^{n-1} a_j n_j(x)$$
$$n_j(x) = \begin{cases} \prod_{i=0}^{j-1} (x - x_i) & j > 0 \\ 1 & j = 0 \end{cases}$$

Every $n_j(x)$ can be computed based on $n_{j-1}(x)$ by multiplying by $(x - x_j)$. Each term of the sum thus requires 1 addition, 1 substraction and 1 multiplication. The total number of operations is $n - 1$ multiplications and $2(n - 1)$ additions/substractions. The complexity is thus $\mathcal{O}(n)$.

c) First, let's examine the storage requirements to compute the coefficients. We know that the divided differences table contains $\frac{n(n-1)}{2}$ entries. However, the evaluated points y_k are only necessary to compute the k-th coefficient. We can thus use the given array that contains the y_k to compute the coefficients in place. The total storage requirement for the coefficient computation is $\mathcal{O}(n)$ (n for the nodes x_i and n for the evaluated points y_i).

Second, to evaluate the interpolation at m distinct points, we simply need a single variable that gets updated for each point. The storage requirement is thus $\mathcal{O}(m)$ for the evaluation part.

The total storage requirement is $\mathcal{O}(n + m)$.

Neville's algorithm

a) Neville's algorithm is designed for direct evaluation and does not precompute coefficients. So, there is no complexity for the coefficient computation.

b) The triangular table contains $\frac{n(n+1)}{2}$ and each entry $P_{i,j}(x)$ requires 2 substractions, 1 multiplication and 1 division. Since the first column of the table is given, the total number of operation is:

$$4 \cdot \left(\frac{n(n+1)}{2} - n \right) = 2n(n-1) \sim \mathcal{O}(n^2)$$

c) For a single point, we need at each step at most n slots. In fact, after computing $P_{0,1}$, we can use the slot where y_0 was stored to place it. The same goes for the other entries. The space requirement for a single point is thus $\mathcal{O}(n)$.

For m evaluation, we simply need m addition slots. The total complexity is $\mathcal{O}(n+m)$.

3. Test functions

References

[1] "Newton polynomial." [Online]. Available: *https://en.wikipedia.org/wiki/Newton_polynomial*

[2] "Neville's algorithm." [Online]. Available: *https://en.wikipedia.org/wiki/Neville%27s_algorithm*