MA5233 Computational Mathematics

Homework Sheet 5

Simon Etter, 2019/2020

Final deadline: 11 October 2019, 7pm

1 Barycentric interpolation formulae

Lecture 14 introduced Lagrange's interpolation formula

$$p(x) = \sum_{j=0}^{n} f(x_j)\, \ell_j(x) \qquad \text{with} \qquad \ell_j(x) = \prod_{i \neq j} \frac{x - x_i}{x_j - x_i} \tag{1}$$

and mentioned that one drawback of this formula is that it requires $\mathcal{O}(n^2)$ floating-point operations for every evaluation of $p(x)$. An easy solution to this problem is to rewrite (1) in the form

$$p(x) = \ell(x) \sum_{j=0}^{n} \frac{w_j\, f(x_j)}{x - x_j} \tag{2}$$

where

$$\ell(x) = \prod_{i=0}^{n} (x - x_i) \qquad \text{and} \qquad w_j = \prod_{i \neq j} \frac{1}{x_j - x_i}.$$

Equation (2) is known as the *first barycentric interpolation formula*, and compared to Lagrange's formula (1) it has the advantage that it can be evaluated with $\mathcal{O}(n)$ floating-point operations once the barycentric weights w_j are available. Computing these weights still requires $\mathcal{O}(n^2)$ operations, but since the w_j are independent of x these computations have to be done only once rather than for every single evaluation of $p(x)$. Furthermore, for many important sets of interpolation points the w_j are given by a simple formula. For example, we have for the Chebyshev extreme points (cf. lecture slides) that

$$\Big(x_i = \cos\big(\pi \tfrac{i}{n}\big)\Big)_{i=0}^{n} \qquad \Longrightarrow \qquad w_j = \begin{cases} (-1)^j\, 2^{n-2}/n & \text{if } j = 0 \text{ or } j = n, \\ (-1)^j\, 2^{n-1}/n & \text{otherwise.} \end{cases} \tag{3}$$

Tasks

1. Complete the functions `baryweights(xx)` and `barycentric(xx,w,f,x)` in the file `sheet5.jl` which comes along with this exercise sheet. Make sure your implementation passes the provided tests, and please do not modify the tests.

   Hint. You may find that your first implementation of `barycentric()` fails the `advanced` test set. Investigate why this is the case, and modify your code such that it also passes the `advanced` test set.

2. We conclude from (3) that at least for some interpolation points x_i, the barycentric weights w_j grow exponentially in the number of points $n+1$. This can be a problem for large n since on a computer we store the w_j in the form

$$w_j = s \times f \times 2^e \quad \text{where} \quad s \in \{-1, 1\},\ f \in [1, 2),\ e \in [-1022, 1023] \cap \mathbb{Z}, \tag{4}$$

and w_j gets "rounded" to `Inf` if it exceeds the range of numbers of the above form (recall Lecture 1).

Assuming $w_j = 2^{n-1}$, determine the largest number $n_{\max}$ such that w_j can still be represented in the form (4). Demonstrate numerically that the weights computed by `baryweights(xx)` indeed round to `Inf` if `xx = @.(cos(pi*(0:n)/n)` and `n` is larger than $n_{\max}$.

Remark. The above derivation of $n_{\max}$ reflects the actual computations only approximately for two reasons:

- We use $w_j = 2^{n-1}$ rather than the exact value of w_j given in (3).
- Computing the w_j involves computing intermediate numbers which are somewhat larger than w_j.

You will find that these two effects approximately cancel such that n_{max} agrees with the behaviour observed in the experiments.

Solution. It follows from $e \leq 1023$ and $w_j = 2^{n-1}$ that $n_{\max} = 1023 + 1$. Indeed, we observe

```
julia> max_w = n -> maximum(baryweights(@.(cos(pi*(0:n)/n))));

julia> max_w(1024)
8.777798510079906e304

julia> max_w(1025)
Inf
```

3. The issue described in Task 2 can be avoided by switching to the *second barycentric formula*

$$p(x) = \sum_{j=0}^{n} \frac{w_j\, f(x_j)}{x - x_j} \Bigg/ \sum_{j=0}^{n} \frac{w_j}{x - x_j}\ . \tag{5}$$

This formula can be obtained in two steps.

- Observe that

$$1 = \ell(x) \sum_{j=0}^{n} \frac{w_j}{x - x_j} \tag{6}$$

  since the right-hand side of (6) describes the polynomial interpolant $p(x)$ through the data points $f(x_i) = 1$ for which it must hold $p(x) = 1$.
- Divide (2) by (6) to arrive at (5).

The second barycentric formula has the advantage that we can replace the weights w_j by $\hat{w}_j := \alpha\, w_j$ in (5) without affecting the correctness of the formula since the newly introduced factor $\alpha \in \mathbb{R}$ cancels. In the case of the Chebyshev points from (3), this allows us to rewrite (5) in the form

$$p(x) = \sum_{j=0}^{n} \frac{\hat{w}_j\, f(x_j)}{x - x_j} \Bigg/ \sum_{j=0}^{n} \frac{\hat{w}_j}{x - x_j} \quad \text{where} \quad \hat{w}_j = \begin{cases} (-1)^j/2 & \text{if } j = 0 \text{ or } j = n, \\ (-1)^j & \text{otherwise.} \end{cases} \tag{7}$$

Implement (7) in the function `chebyshev(f,x)` from `sheet5.jl`. Make sure your implementation passes the provided tests.