Numerical Analysis
MS-C1650
Tölle/Nyman
Exercises, Week 18, 2022

DATE[1]

On assignments: Submit homework to your assistant electronically via the course pages. MATLAB-assignments are submitted via MyCources.

1 Univariate Barycentric Formulation

We have seen that Newton's interpolation polynomials appear to have an advantage in that they can be updated more efficiently than for instance Lagrange's interpolation polynomial. However, the Lagrange form can be written more efficiently in the so-called barycentric form, where the evaluation is faster.
Let us introduce the following quantities

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

EXERCISE 1

(a) Show that the value at x of the polynomial p can be written as

$$p(x) = \varphi(x) \sum_{i=0}^{n} \frac{w_i}{x - x_i} y_i.$$

(b) Derive the barycentric formula

$$p(x) = \left(\sum_{i=0}^{n} \frac{w_i}{x - x_i} y_i \right) \Big/ \left(\sum_{i=0}^{n} \frac{w_i}{x - x_i} \right).$$

[1] Published on 2021-04-30 10:08:10+03:00.

(c) Show that the updated weights w_k can be computed in $O(n)$ arithmetic operations after each added point x_{n+1}.

EXERCISE 2

(a) Write a function lagweights.m that computes the weights w_k for the nodes x_k.

(b) Write a function specialsum.m that computes the quantity $\sum_{i=0}^{n} \frac{z_i}{t-x_i}$, when x and z are arrays of size n and t is an array of size s. The output has to be an array of size s. That is, t has the values where the interpolation polynomial is evaluated.

(c) Write a program lagpolint.m that computes the barycentric form of p at points t.

(d) Test lagpolint.m by sampling from the function $y = \sqrt{|t|}$ on $[-1, 1]$. Try first 9 uniform points and then 101 Chebyshev points

$$x_j = -\cos(j\pi/n), \quad j = 0, 1, \ldots, 100 := n.$$

Plot the polynomials.

2 Application to Interpolating Surfaces

Let us next consider a regular grid of points (x_i, y_j) and the surface values $z_{ij} = f(x_i, y_j)$. Let $\mathbf{x} = (x_0, \ldots, x_m) \in \mathbb{R}^{m+1}$ and $\mathbf{y} = (y_0, \ldots, y_n) \in \mathbb{R}^{n+1}$. In this setting the surface can be interpolated using a product (tensor product) of univariate interpolation polynomials. In the sequel we denote the y-dependent quantities with a bar, for instance, $\bar{l}_q(t)$ for the corresponding Lagrange basis polynomial in the y-direction.

EXERCISE 3

(a) Show that $P(s,t) = \sum_{p=0}^{m} \sum_{q=0}^{n} z_{pq} l_p(s) \bar{l}_q(t)$ is an interpolation polynomial.

(b) Show that

$$P(s,t) = \left(\sum_{p=0}^{m} \sum_{q=0}^{n} \frac{w_p \bar{w}_q}{(s-x_p)(t-y_q)} z_{pq} \right) \Big/ \left(\sum_{p=0}^{m} \frac{w_p}{s-x_p} \sum_{q=0}^{n} \frac{\bar{w}_q}{t-y_q} \right).$$

EXERCISE 4 Write a program interpolsurf.m such that given the grid points as x and y, and the sampled values $\mathbf{z} = (f(x_i, y_j)) \in \mathbb{R}^{(m+1)\times(n+1)}$, computes the values of the polynomial $P(s,t)$.
Test with $f(s,t) = \sin(s+t)$, $m = 3$, x a uniform partition of $[0,\pi]$, and $n = 7$, y a uniform partition of $[0, 2\pi]$. Plot $P(s,t)$, $f(s,t)$, and the difference $f - P$.