CHEBYSHEV INTERPOLATION

NICHOLAS F. MARSHALL

1. Introduction

1.1. **Summary.** In this note we state some key results about polynomial interpolation. In particular, we state the general remainder formula for polynomial interpolation, and consider the example of Chebyshev nodes of the first kind.

1.2. **Motivation.** Before discussing interpolation, we recall the Weierstrass approximation theorem. Let f be a real-valued function defined on a compact interval $[a, b]$ of $\mathbb{R}$. Then, for any $\varepsilon > 0$, there exists a polynomial $p(x)$ such that

$$|f(x) - p(x)| \leq \varepsilon,$$

for all $x \in [a, b]$. This result motivates the study of numerical methods to approximate functions by polynomials; interpolation is one such method.

1.3. **Basic result.** Let $f : [a, b] \to \mathbb{R}$, and $x_1 < \cdots < x_n \in [a, b]$ be given. Then, there exists a unique polynomial of degree at most $n - 1$ that interpolates f at $x_1, \ldots, x_n$ in the sense that

$$p(x_j) = f(x_j),$$

for $j = 1, \ldots, n$. We call p the interpolating polynomial of f at $x_1, \ldots, x_n$.

Proof. The interpolating polynomial p can be expressed explicitly in Lagrange form by

$$p(x) = \sum_{j=1}^{n} f(x_j) q_j(x),$$

where

$$q_j(x) = \prod_{k=1, k\neq j}^{n} \frac{x - x_k}{x_j - x_k}.$$

If p and q are two polynomials of degree at most $n - 1$ that interpolate f at $x_1, \ldots, x_n$, then $w = p - q$, is a polynomial of degree at most $n - 1$ that vanishes at n points $x_1, \ldots, x_n$. It follows that w is identically zero, which implies $p = q$. □

2. Interpolation error and implementation

2.1. **Remainder term.** Let f be an m times continuously differentiable function on a compact interval $[a, b]$. Suppose that p is the $m - 1$ degree polynomial that interpolates f at $x_1, \ldots, x_m \in [a, b]$. Then,

$$R(x) := f(x) - p(x) = \frac{f^{(m)}(\xi)}{m!}(x - x_1) \cdots (x - x_m),$$

for some $\xi \in [a,b]$. It follows that

$$|f(x) - p(x)| \leq \frac{\phi(x)}{m!} \max_{y\in[a,b]} \left|f^{(m)}(y)\right|, \tag{1}$$

where $\phi(x) = (x - x_1)\cdots(x - x_m)$ is the monic polynomial of degree m with roots at $x_1, \ldots, x_m$.

2.2. **Chebyshev nodes.** The points

$$x_k = \cos\left(\frac{2k-1}{m}\frac{\pi}{2}\right),$$

for $k = 1, \ldots, m$ are called Chebyshev nodes (of the first kind). They are roots of the degree m Chebyshev polynomial (of the first kind) defined by

$$T_m(x) = \cos(m \arccos x),$$

for $x \in [-1, 1]$. The Chebyshev polynomials satisfy the recursion formula $T_0(x) = 1$, $T_1(x) = x$,

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x),$$

for $k \geq 1$, and thus the leading coefficient of T_k is 2^{k-1}. Moreover, observe that

$$|T_m(x)| = |\cos(m \arccos(x)| \leq 1$$

for $x \in [-1, 1]$. Thus, if $x_1, \ldots, x_m$ are Chebyshev nodes (of the first kind), then

$$|(x - x_1)\cdots(x - x_m)| = \left|\frac{1}{2^{m-1}}T_m(x)\right| \leq \frac{1}{2^{m-1}},$$

for $x \in [-1, 1]$. If we define the map $l : [-1, 1] \to [a, b]$ by

$$l(x) = \frac{b-a}{2}x + \frac{b+a}{2},$$

then it follows that

$$|(x - l(x_1))\cdots(x - l(x_m))| \leq \frac{1}{2^{m-1}}\left(\frac{b-a}{2}\right)^m, \tag{2}$$

for $x \in [a, b]$. Combining (1) and (2) gives the following result.

Lemma 1. *Suppose that f is an m times continuously differentiable function on the compact interval $[a, b]$, and let p be the $(m-1)$-degree polynomial that interpolates f at the points*

$$x_k = \left(\frac{b-a}{2}\right)\cos\left(\frac{2k-1}{m}\frac{\pi}{2}\right) + \frac{b+a}{2},$$

for $k = 1, \ldots, m$. Then,

$$|f(x) - p(x)| \leq \frac{2C_m}{m!}\left(\frac{b-a}{4}\right)^m,$$

for $x \in [a, b]$ where $C_m := \max_{y\in[a,b]} |f^{(m)}(y)|$.

2.3. **Implementation.** See https://people.maths.ox.ac.uk/trefethen/barycentric.pdf.

```
% Example
f = @(x) exp(x); % function
m = 30; % number of interpolation points
ts = -1+1/m:2/m:1-1/m; xs = cos(pi*(ts+1)/2);
ys = f(xs);
x = linspace(-1,1,5000);

% Barycentric interpolation
c = (-1).^(0:m-1).*sin(pi*(ts+1)/2);
numer = zeros(size(x));
denom = zeros(size(x));
exact = zeros(size(x));
for j = 1:m
    xdiff = x-xs(j);
    temp = c(j)./xdiff;
    numer = numer + temp*ys(j);
    denom = denom + temp;
    exact(xdiff==0) = j;
end
y = numer./denom; jj = find(exact); y(jj) = ys(exact(jj));

% Check error
max(abs(y - f(x)))
```