Mathematical Tripos Part IB: Lent 2020
Numerical Analysis – Lecture 1[1]

What is numerical analysis?

Numerical analysis is the study of algorithms for problems in continuous mathematics.[2] The key word here is *algorithms*. We are looking for algorithms that *run fast* and that are *stable* against various sources of errors and "noise". Some examples of problems from continuous mathematics include:

- Algebraic equations: Solve $f(x) = 0$ where $f : \mathbb{R}^n \to \mathbb{R}$ is a given function.
- Differential equations: Solve $\frac{dx}{dt} = f(x)$ where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a given function.
- Optimization: Find $\min\{f(x) : x \in \mathbb{R}^n\}$ where $f : \mathbb{R}^n \to \mathbb{R}$ is a given function.

Needless to say that such problems arise in many different application areas!

A note about computational complexity We measure the complexity of an algorithm by the number of *elementary operations* ($+,-,\times,\div$) it needs. We use the Big-Oh notation, e.g., $\mathcal{O}(n)$ or $\mathcal{O}(n^2)$ where n is the size of the input. More precisely an algorithm has complexity $\mathcal{O}(f(n))$ if the number of operations it needs is at most $C \cdot f(n)$ where $C > 0$ is a constant.

1 Polynomial interpolation

We denote by $\mathbb{P}_n[x]$ the *linear space* of all real polynomials of degree at most n.

1.1 The interpolation problem

Given $n+1$ distinct real points $x_0, x_1, \ldots, x_n$ and real numbers $f_0, f_1, \ldots, f_n$, we seek a polynomial $p \in \mathbb{P}_n[x]$ such that $p(x_i) = f_i$, $i = 0, 1, \ldots, n$. Such a polynomial is called an *interpolant*. Note that a polynomial $p \in \mathbb{P}_n[x]$ has $n+1$ degrees of freedom, while interpolation at $x_0, x_1, \ldots, x_n$ constitutes $n+1$ conditions. This, intuitively, justifies seeking an interpolant from $\mathbb{P}_n[x]$.

1.2 The Lagrange formula

Although, in principle, we may solve a linear problem with $n+1$ unknowns to determine a polynomial interpolant, this can be accomplished more easily by using the explicit *Lagrange formula*. We claim that

$$p(x) = \sum_{k=0}^{n} f_k \prod_{\substack{\ell=0 \\ \ell \neq k}}^{n} \frac{x - x_\ell}{x_k - x_\ell}, \qquad x \in \mathbb{R}.$$

Note that $p \in \mathbb{P}_n[x]$, as required. We wish to show that it interpolates the data. Define

$$L_k(x) := \prod_{\substack{\ell=0 \\ \ell \neq k}}^{n} \frac{x - x_\ell}{x_k - x_\ell}, \qquad k = 0, 1, \ldots, n$$

(*Lagrange cardinal polynomials*). It is trivial to verify that $L_j(x_j) = 1$ and $L_j(x_k) = 0$ for $k \neq j$, hence

$$p(x_j) = \sum_{k=0}^{n} f_k L_k(x_j) = f_j, \qquad j = 0, 1, \ldots, n,$$

[1]Corrections and suggestions to these notes should be emailed to `h.fawzi@damtp.cam.ac.uk`.

[2]This definition is taken from the essay *The definition of numerical analysis* by L. N. Trefethen.

and p is an interpolant,

Uniqueness Suppose that both $p \in \mathbb{P}_n[x]$ and $q \in \mathbb{P}_n[x]$ interpolate to the same $n+1$ data. Then the nth degree polynomial $p-q$ vanishes at $n+1$ distinct points. But the only nth-degree polynomial with $\geq n+1$ zeros is the zero polynomial. Therefore $p - q \equiv 0$ and the interpolating polynomial is unique.

Complexity For each $k = 0, \ldots, n$, evaluating $L_k(x)$ takes $\mathcal{O}(n)$ operations, and thus the total complexity of evaluating $p(x)$ using the Lagrange formula is $\mathcal{O}(n^2)$.

1.3 The error of polynomial interpolation

Let $[a,b]$ be a closed interval of $\mathbb{R}$. We denote by $C[a,b]$ the space of all continuous functions from $[a,b]$ to $\mathbb{R}$ and let $C^s[a,b]$, where s is a positive integer, stand for the linear space of all functions in $C[a,b]$ that possess s continuous derivatives.

Theorem Given $f \in C^{n+1}[a,b]$, let $p \in \mathbb{P}_n[x]$ interpolate the values $f(x_i)$, $i = 0, 1, \ldots, n$, where $x_0, \ldots, x_n \in [a,b]$ are pairwise distinct. Then for every $x \in [a,b]$ there exists $\xi \in [a,b]$ such that

$$f(x) - p(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi) \prod_{i=0}^{n} (x - x_i). \tag{1.1}$$

Proof. The formula (1.1) is true when $x = x_j$ for $j \in \{0, 1, \ldots, n\}$, since both sides of the equation vanish. Let $x \in [a,b]$ be any other point and define

$$\phi(t) := f(t) - \left(p(t) + (f(x) - p(x)) \frac{\prod_{i=0}^{n}(t - x_i)}{\prod_{i=0}^{n}(x - x_i)} \right), \qquad t \in [a,b].$$

[Note: The variable in ϕ is t, whereas x is a fixed parameter.] Note that $\phi(x_j) = 0$, $j = 0, 1, \ldots, n$, and $\phi(x) = 0$. Hence, ϕ has at least $n+2$ distinct zeros in $[a,b]$. Moreover, $\phi \in C^{n+1}[a,b]$.
We now apply the *Rolle theorem:* if the function $g \in C^1[a,b]$ vanishes at two distinct points in $[a,b]$ then its derivative vanishes at an intermediate point. We deduce that ϕ' vanishes at (at least) $n+1$ distinct points in $[a,b]$. Next, applying Rolle to ϕ', we conclude that ϕ'' vanishes at n points in $[a,b]$. In general, we prove by induction that $\phi^{(s)}$ vanishes at $n+2-s$ distinct points of $[a,b]$ for $s = 0, 1, \ldots, n+1$. Letting $s = n+1$, we have $\phi^{(n+1)}(\xi) = 0$ for some $\xi \in [a,b]$. Hence

$$0 = \phi^{(n+1)}(\xi) = f^{(n+1)}(\xi) - \left(p^{(n+1)}(\xi) + (f(x) - p(x)) \frac{(n+1)!}{\prod_{i=0}^{n}(x - x_i)} \right)$$

where we used the fact that $\frac{\mathrm{d}^{n+1}}{\mathrm{d}t^{n+1}} \prod_{i=0}^{n}(t - x_i) = (n+1)!$. Using the fact that $p^{(n+1)} \equiv 0$ (since p is a polynomial of degree n) we finally get (1.1). □

Runge's example We interpolate $f(x) = 1/(1+x^2)$, $x \in [-5,5]$, at the equally-spaced points $x_j = -5 + 10\frac{j}{n}$, $j = 0, 1, \ldots, n$. Some of the errors are displayed below

x	$f(x) - p(x)$	$\prod_{i=0}^{n}(x - x_i)$
0.75	3.2×10^{-3}	-2.5×10^{6}
1.75	7.7×10^{-3}	-6.6×10^{6}
2.75	3.6×10^{-2}	-4.1×10^{7}
3.75	5.1×10^{-1}	-7.6×10^{8}
4.75	$4.0 \times 10^{+2}$	-7.3×10^{10}

Table: Errors for $n = 20$

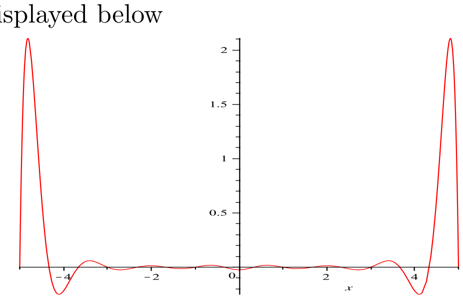

Figure: Errors for $n = 15$

The growth in the error is explained by the product term in (1.1) (the rightmost column of the table). Adding more interpolation points makes the largest error even worse. A remedy to this state of affairs is to cluster points toward the end of the range. A considerably smaller error is attained for $x_j = 5 \cos \frac{(n-j)\pi}{n}$, $j = 0, 1, \ldots, n$ (so-called *Chebyshev points*). It is possible to prove that this choice of points minimizes the magnitude of $\max_{x \in [-5,5]} |\prod_{i=0}^{n}(x - x_i)|$.