

Mathematical Tripos Part IB: Lent Term 2024

Numerical Analysis – Lecture 4

4 Approximation of linear functionals

Given a vector space $\mathbb{X}$ (e.g., $\mathbb{R}^n$, or $C^m[a, b]$), a *linear functional* is a linear mapping

$$\lambda : \mathbb{X} \rightarrow \mathbb{R} \quad \left(\text{so } \lambda(\alpha f + \beta g) = \alpha \lambda(f) + \beta \lambda(g), \quad \forall f, g \in \mathbb{X} \text{ and } \forall \alpha, \beta \in \mathbb{R} \right).$$

We will treat the space $\mathbb{X} = C^{n+1}[a, b]$, and the functionals

$$1) \quad \lambda(f) = f^{(k)}(\bar{x}), \quad (\bar{x}, k \text{ being fixed}), \quad 2) \quad \lambda(f) = \int_a^b f(x) w(x) dx. \quad (4.1)$$

Our goal is to find, with some N , an approximation of the form

$$\lambda(f) \approx \sum_{i=0}^N a_i f(x_i), \quad f \in C^{n+1}[a, b]. \quad (4.2)$$

For the functionals 1)–2) in (4.1), this is called *numerical differentiation* and *numerical integration*, respectively.

Method 4.1 A suggestive approximation method is to interpolate f by $p \in \mathcal{P}_n$ and take $\lambda(f) \approx \lambda(p)$. This is called the *interpolating formula* (of degree n), in this case $N = n$. We have already seen an interpolating formula, namely that of Lagrange for the functional $\lambda(f) = f(\bar{x})$:

$$f(\bar{x}) \approx p(\bar{x}) = \sum_{i=0}^n \ell_i(\bar{x}) f(x_i).$$

By linearity of λ , we have $\lambda(p) = \sum_{i=0}^n \lambda(\ell_i) p(x_i)$, thus the interpolating formula has the form

$$\lambda(f) \approx \sum_{i=0}^n \lambda(\ell_i) f(x_i). \quad (4.3)$$

Here $(\ell_i)_{i=0}^n$ are the fundamental Lagrange polynomials of degree n , $\ell_i(x_j) = \delta_{ij}$.

Method 4.2 Another method is to require from the formula (4.2) to be *exact on $\mathcal{P}_n$* , that is to become equality for any $f \in \mathcal{P}_n$. In this case the number N of terms (almost) need not to be restricted, and if $N > n$, then it is certainly not a polynomial of degree n that substitutes the function. If $N < n$, then the formula is of *high accuracy*. However, if $N = n$, then both methods are the same.

Lemma 4.3 *The formula $\lambda(f) \approx \sum_{i=0}^n a_i f(x_i)$ is interpolating $\Leftrightarrow$ it is exact on $\mathcal{P}_n$.*

Proof. The interpolating formula (4.3) is exact on $\mathcal{P}_n$ by definition. Conversely, if the formula in the lemma is exact on $\mathcal{P}_n$, take $f(x) = \ell_j(x)$ to obtain $\lambda(\ell_j) = \sum_{i=0}^n a_i \ell_j(x_i) = a_j$, i.e., (4.3). $\square$

4.1 Numerical integration

A formula of numerical integration (with some integral weight $w(x) \geq 0$)

$$\lambda(f) := I(f) := \int_a^b f(x) w(x) dx \approx \sum_{i=0}^n a_i f(x_i), \quad f \in C[a, b],$$

is called a *quadrature formula* (or *rule*), with *nodes* (x_i) and *weights* (a_i) . By Lemma 4.3, for any $n+1$ fixed (x_i) , the interpolating formula $a_i = I(\ell_i)$ is exact on $\mathcal{P}_n$. Can one find a *quadrature of higher accuracy* which is exact on $\mathcal{P}_m$ with some $m > n$? Since we are free in choosing $n+1$ nodes (x_i) , we may hope to increase the degree of accuracy from degree n up to $2n+1$. More is impossible.

Claim 4.4 No quadrature formula with $n + 1$ nodes is exact for all $p \in \mathcal{P}_m$ if $m \geq 2n + 2$.

Proof. Take $p(x) = \prod_{i=0}^n (x - x_i)^2 \geq 0$. Then $p \in \mathcal{P}_{2n+2}$, and $I(p) > 0$, but $\sum_{i=0}^n a_i p(x_i) = 0$ for any a_i s. Hence the integral and the quadrature do not match. $\square$

Our next goal is to show that $m = 2n + 1$ can be attained. For this we need

Lemma 4.5 Let Q_{n+1} be orthogonal to all $p_n \in \mathcal{P}_n$ on $[a, b]$. Then all zeros of Q_{n+1} are real, distinct and lie in the interval (a, b) .

Proof. Denote by k the number of the sign changes of Q_{n+1} in (a, b) and assume that $k \leq n$. If $k = 0$ set $p_k \equiv 1$, and if $1 \leq k \leq n$ set $p_k(x) := \prod_{i=1}^k (x - t_i)$ where t_i s are the points where a sign change of Q_{n+1} occurs. Then $p_k \in \mathcal{P}_k$, $k \leq n$, hence $(Q_{n+1}, p_k) = 0$. On the other hand, by construction, $Q_{n+1}(x)p_k(x)$ does not change sign throughout $[a, b]$ and vanishes at a finite number of points, hence $|(Q_{n+1}, p_k)| := \left| \int_a^b Q_{n+1}(x)p_k(x)w(x)dx \right| = \int_a^b |Q_{n+1}(x)p_k(x)|w(x)dx > 0$, a contradiction. Thus, $k \geq n + 1$ (hence $k = n + 1$) and the statement follows. $\square$

Theorem 4.6 Let a quadrature with $n + 1$ nodes $(x_i)_{i=0}^n$ be exact on $\mathcal{P}_n$ (i.e. interpolating). Then it is exact on $\mathcal{P}_{2n+1}$ if and only if its nodes $(x_i)_{i=0}^n$ are the zeros of the $(n + 1)$ -st orthogonal polynomial Q_{n+1} .

Proof. 1) If a quadrature with $n + 1$ nodes (x_i) is exact for all $p \in \mathcal{P}_{2n+1}$, then setting $Q_{n+1}(x) := \prod_{i=0}^n (x - x_i) \in \mathcal{P}_{n+1}$ and taking any $q_n \in \mathcal{P}_n$ we find

$$\int_a^b Q_{n+1}(x)q_n(x)w(x)dx = \sum_{i=0}^n a_i [Q_{n+1}(x_i)q_n(x_i)] = 0,$$

i.e., Q_{n+1} is orthogonal to all $q_n \in \mathcal{P}_n$.

2) Conversely, let $(x_i)_{i=0}^n$ be the zeros of orthogonal Q_{n+1} (residing in $[a, b]$ by Lemma 4.5). Given any $p_{2n+1} \in \mathcal{P}_{2n+1}$, we can represent it uniquely as

$$p_{2n+1}(x) = Q_{n+1}(x)r_n(x) + s_n(x), \quad \text{with some } r_n, s_n \in \mathcal{P}_n$$

(note that s_n interpolates p_{2n+1} at the points (x_i)). Since Q_{n+1} is orthogonal to r_n , we have

$$I(p_{2n+1}) = \int_a^b p_{2n+1}(x)w(x)dx = \int_a^b s_n(x)w(x)dx = I(s_n).$$

On the other hand, because $Q_{n+1}(x_i) = 0$,

$$\sum_{i=0}^n a_i p_{2n+1}(x_i) = \sum_{i=0}^n a_i s_n(x_i).$$

But $s_n \in \mathcal{P}_n$, while the quadrature is exact on $\mathcal{P}_n$ by assumption, hence $I(s_n) = \sum a_i s_n(x_i)$, i.e., the right-hand sides of two previous equalities coincide, therefore the left-hand sides coincide too, i.e., $I(p_{2n+1}) = \sum a_i p_{2n+1}(x_i)$. $\square$

Definition 4.7 A quadrature with $n + 1$ nodes that is exact on $\mathcal{P}_{2n+1}$ is called *Gaussian quadrature*.

Example 4.8 For $[a, b] = [-1, 1]$ and $w(x) \equiv 1$, the underlying orthogonal polynomials are the *Legendre polynomials* (see Example 3.5). The corresponding weights and nodes of the Gaussian quadratures for $n = 0, 1, 2$ are as follows.

$$\begin{array}{llll} n = 0 & P_1(x) = x & x_0 = 0 & a_0 = 2 \\ n = 1 & P_2(x) = \frac{3}{2}(x^2 - \frac{1}{3}) & x_0 = -\frac{1}{\sqrt{3}}, x_1 = \frac{1}{\sqrt{3}} & a_0 = 1, a_1 = 1 \\ n = 2 & P_3(x) = \frac{5}{2}(x^3 - \frac{3}{5}x) & x_0 = -\sqrt{\frac{3}{5}}, x_1 = 0, x_2 = \sqrt{\frac{3}{5}} & a_0 = \frac{5}{9}, a_1 = \frac{8}{9}, a_2 = \frac{5}{9} \end{array}$$

Example 4.9 For $[a, b] = [-1, 1]$ and $w(x) = (1 - x^2)^{-1/2}$, the orthogonal polynomials are the *Chebyshev polynomials* and the quadrature rule is particularly simple:

$$T_{n+1}(x) = \cos(n + 1) \arccos x, \quad x_i = \cos \frac{2i+1}{2n+2} \pi, \quad a_i = \frac{\pi}{n+1}, \quad i = 0, \dots, n.$$

Example 4.10 (Simplest quadrature formulae ($w \equiv 1$)).

the rectangle rule:	$I(f) \approx (b-a)f(a)$	1-point, non-Gaussian exact on constants
the midpoint rule:	$I(f) \approx (b-a)f(\frac{a+b}{2})$	1-point, Gaussian, exact on linear fns
the trapezoidal rule:	$I(f) \approx (b-a)[\frac{1}{2}f(a) + \frac{1}{2}f(b)]$	2-point, non-Gaussian, exact on linear fns
the Simpson rule:	$I(f) \approx (b-a)[\frac{1}{6}f(a) + \frac{2}{3}f(\frac{a+b}{2}) + \frac{1}{6}f(b)]$	3-point, non-Gaussian, but of higher accuracy (exact on cubics)

Method 4.11 (Composite quadrature formulas) Suppose that, for some $f \in C[0, 1]$, we want to approximate the integral $I(f) = \int_0^1 f(x) dx$ with some given accuracy $\epsilon \ll 1$. Clearly, the simplest quadratures as above with two, three or even 10 nodes in $[0, 1]$ will not do the job. Using the Gaussian quadrature with 100 or 1000 nodes could be too costly (finding zeros of the Legendre polynomial of degree 100 is an enormous task).

A standard solution is to partition the interval $[0, 1]$ into N equal intervals of length $h = 1/N$, and to apply one and the same simple quadrature (say, from the list above) at each smaller subinterval. This is called a *composite quadrature formula*. For example, for the *trapezoidal rule*, we obtain

$$\int_0^1 f(x) dx = I(f) \approx I_N(f) = \frac{1}{N} \sum_{i=0}^{N-1} \left[\frac{1}{2}f\left(\frac{i}{N}\right) + \frac{1}{2}f\left(\frac{i+1}{N}\right) \right] = \frac{1}{N} \left[\frac{1}{2}f(0) + \sum_{i=1}^{N-1} f\left(\frac{i}{N}\right) + \frac{1}{2}f(1) \right].$$

This formula is exact on the polynomials of degree $n = 1$ and, in fact, on piecewise linear functions (broken lines) with the breakpoints $(\frac{i}{N})$. This is an example of the formula (4.2) with $N > n$.

One can prove that if a quadrature formula is exact on $\mathcal{P}_n$, then for the functions $f \in C^{n+1}[0, 1]$ the *composite quadrature formula* with the step size h will have an error $\mathcal{O}(h^{n+1})$.

4.2 Numerical differentiation

Consider the interpolating formulae for numerical differentiation

$$\lambda(f) = f^{(k)}(\bar{x}) \approx \sum_{i=0}^n a_i f(x_i), \quad a_i = \lambda(\ell_i) = \ell_i^{(k)}(\bar{x}),$$

which are exact on the polynomials of degree n . The simplest ones are with $n = k$, i.e., $f^{(k)}(\bar{x}) \approx p^{(k)}(\bar{x})$ where p is the interpolating polynomial of degree k . But $p^{(k)}(\bar{x})$ is $k!$ times the leading coefficient of p , i.e., $k!f[x_0, \dots, x_k]$, and we obtain the simplest rules

$$f^{(k)}(\bar{x}) \approx k!f[x_0, \dots, x_k].$$

Example 4.12

the forward difference:	$f'(x) \approx f[x, x+h] = \frac{f(x+h) - f(x)}{h}$	2-point, exact on linear fns
the central difference:	$f'(x) \approx f[x-h, x+h] = \frac{f(x+h) - f(x-h)}{2h}$	2-point, of higher accuracy, exact on quadratics
the 2-nd central difference:	$f''(x) \approx 2f[x-h, x, x+h] = \frac{f(x-h) - 2f(x) + f(x+h)}{h^2}$	3-point, of higher accuracy, exact on cubics

Example 4.13 $n = 2, k = 1, [a, b] = [0, 2]$.

$$f'(0) \approx -\frac{3}{2}f(0) + 2f(1) - \frac{1}{2}f(2). \quad (4.4)$$

Of course, one can transform any formula to any interval, and formulas of numerical differentiation (and integration) are usually used on the intervals of a small length, say length $\mathcal{O}(h)$ as in the examples above.

In (4.4), and in any other formula $\lambda(f) \approx \sum_{i=0}^n a_i f(x_i)$ known to be exact on $\mathcal{P}_n$, given the nodes $(x_i)_{i=0}^n$, in our case $\{0, 1, 2\}$, we can find the corresponding coefficients (a_i) in two ways.

1) We determine the fundamental Lagrange polynomials ℓ_i (such that $\ell_i(x_j) = \delta_{ij}$),

$$\ell_0(x) = \frac{1}{2}(x-1)(x-2), \quad \ell_1(x) = -x(x-2), \quad \ell_2(x) = \frac{1}{2}x(x-1),$$

and set $a_i = \lambda(\ell_i)$:

$$a_0 = \ell'_0(0) = -\frac{3}{2}, \quad a_1 = \ell'_1(0) = 2, \quad a_2 = \ell'_2(0) = -\frac{1}{2}.$$

2) Sometimes it is easier to solve the system of linear equations which arises if we require the formula to be exact on monomials (x^m) . In our case, we get

$$\begin{cases} a_0 + a_1 + a_2 = 0 & (f(x) = 1) \\ a_1 + 2a_2 = 1 & (f(x) = x) \\ a_1 + 4a_2 = 0 & (f(x) = x^2) \end{cases} \Rightarrow a_2 = -\frac{1}{2}, \quad a_1 = 2, \quad a_0 = -\frac{3}{2}.$$

3) On the interval $[x, x+2h]$, the same formula (4.4) will take the form

$$f'(x) \approx \frac{1}{h} \left[-\frac{3}{2}f(x) + 2f(x+h) - \frac{1}{2}f(x+2h) \right].$$