Lecture 4
Approximation of functions

1 Best approximation

Given a function f that is continuous on a given interval $[a,b]$, consider approximating it by some polynomial P. To measure the error in P as an approximation, introduce

$$E(P) = \max_{a \leqslant x \leqslant b} |f(x) - P(x)|.$$

This is called the **maximum error** or **uniform error** of approximation of f by P on $[a,b]$. Also this expression is called the **infinity norm** of the error $f(x) - P(x)$:

$$E(P) = \|f(x) - P(x)\|_\infty = \max_{a \leqslant x \leqslant b} |f(x) - P(x)|.$$

One possibility to approximate a function f with a polynomial is to consider Taylor polynomials of degree n, P_n. Other possibility is to consider the interpolating polynomial on a set of data points from interval $[a,b]$.

With an eye towards efficiency, we want to find the "best"possible approximation of a given degree n. With this in mind, introduce the following:

$$\rho_n(f) = \min_{\deg(P) \leqslant n} E(P) = \min_{\deg(P) \leqslant n} \left[\max_{a \leqslant x \leqslant b} |f(x) - P(x)| \right].$$

The number $\rho_n(f)$ will be the smallest possible uniform error, or **minimax error**, when approximating function f by polynomials of degree at most n. If there is a polynomial giving this smallest error, we call it **minimax approximation** or **best approximation** and denote this polynomial by $M_n(x)$. Thus $E(M_n) = \rho_n(f)$.

Example 1 *Consider function $f(x) = e^x$ on $[-1,1]$. In the following table, we give the values of maximum error $E(P_n)$, where $P_n(x)$ is the Taylor polynomial of degree n for e^x about $x = 0$, and $E(M_n)$.*

n	$E(P_n)$	$E(M_n)$
1	$7.18e-01$	$2.79e-01$
2	$2.18e-01$	$4.50e-02$
3	$5.16e-02$	$5.53e-03$
4	$9.95e-03$	$5.47e-04$
5	$1.62e-03$	$4.52e-05$
6	$2.26e-04$	$3.21e-06$
7	$2.79e-05$	$2.00e-07$
8	$3.06e-06$	$1.11e-08$
9	$3.01e-07$	$5.52e-10$

Consider graphically how we can improve on the Taylor polynomial

$$P_1(x) = 1 + x$$

as a uniform approximation to e^x on the interval $[-1,1]$.

The linear minimax approximation is

$$M_1(x) = 1.2643 + 1.1752x.$$

Observe that the error in minimax approximation is much smaller than the error in Taylor approximation. For example, when using cubic polynomials to approximate e^x the minimax error is smaller by a factor of 10. Also, note that in the case of minimax approximation, the error is more distributed through interval $[-1,1]$, whether in the case of Taylor approximation it is of the same sign and it gets bigger when approaching endpoints.

We can derive an estimate for minimax error.

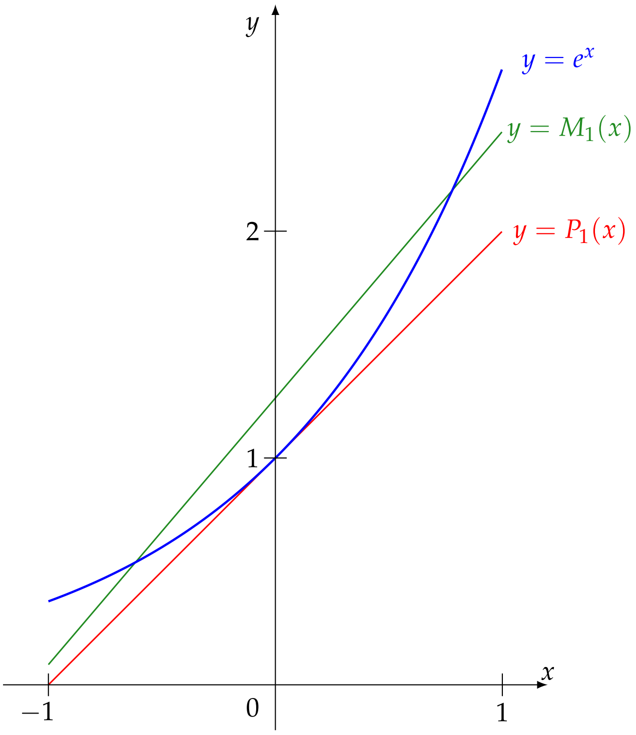

Figure 1: Function e^x (blue) and approximating polynomials P_1 (red) and M_1 (green)

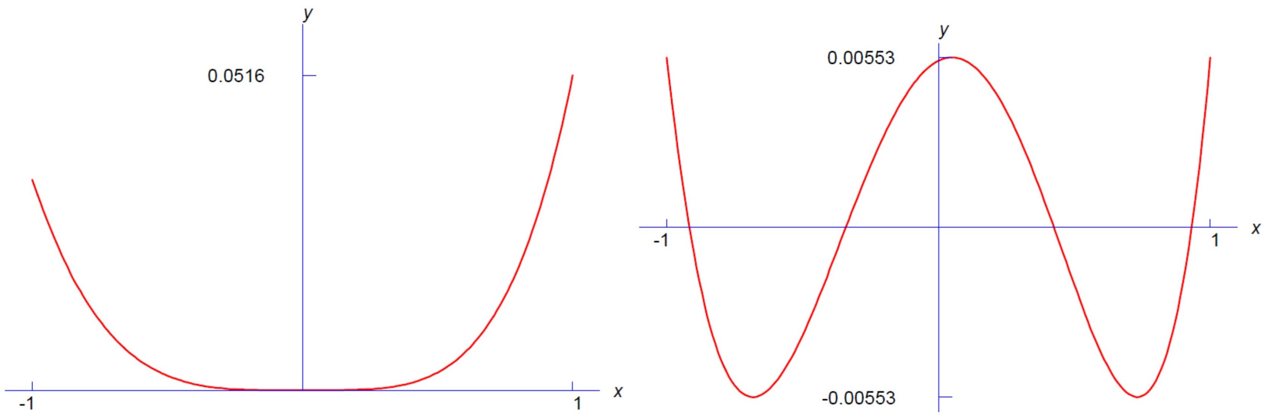

Figure 2: Error in approximation e^x by Taylor polynomial P_3 (left) and by minimax polynomial M_3 (right)

Theorem 1 *Let f be a function defined on interval $[a, b]$ that is at least $n+1$ times continuously differentiable. Then the minimax error satisfies the following estimate*

$$\rho_n(f) \leqslant \frac{\left(\frac{b-a}{2}\right)^{n+1}}{(n+1)! \cdot 2^n} \cdot \max_{a \leqslant x \leqslant b} \left|f^{(n+1)}(x)\right|.$$

This error bound does not always become smaller with increasing n, but it will give a fairly accurate bound for many common functions f.

Example 2 *Let $f(x) = e^x$ on interval $[-1, 1]$. Then $\frac{b-a}{2} = 1$ and*

$$\max_{-1 \leqslant x \leqslant 1} \left|f^{(n+1)}(x)\right| = \max_{-1 \leqslant x \leqslant 1} \left|e^x\right| = e.$$

Therefore, the error bound in this case becomes

$$\rho_n(e^x) \leqslant \frac{e}{(n+1)! \cdot 2^n}. \tag{E*}$$

In the table below there are presented the values of $\rho_n(e^x)$ and the values provided by estimate $(E*)$. Note that the bound $(E*)$ can serve as an estimate for the minimax error.

n	$Bound(E*)$	$\rho_n(e^x)$
1	$6.80e-01$	$2.79e-01$
2	$1.13e-01$	$4.50e-02$
3	$1.42e-02$	$5.53e-03$
4	$1.42e-03$	$5.47e-04$
5	$1.18e-04$	$4.52e-05$
6	$8.43e-06$	$3.21e-06$
7	$5.27e-07$	$2.00e-07$

Table 1: Comparison between bound $(E*)$ and $\rho_n(e^x)$

The main difficulty is that the process of constructing the best approximation is not straightforward and complex. We will look other alternatives.

2 Chebyshev polynomials

Chebyshev polynomials are used in many parts of numerical analysis, and more generally, in applications of mathematics.

Definition 1 *For an integer $n \geqslant 0$, the function defined by*

$$T_n(x) = \cos\left(n\cos^{-1}(x)\right), \qquad -1 \leqslant x \leqslant 1$$

is called the Chebyshev polynomial of degree n.

This may not appear to be a polynomial, but we will show it is a polynomial of degree n. To simplify, introduce notation

$$\theta = \cos^{-1}(x) \quad \text{or} \quad x = \cos(\theta), \qquad 0 \leqslant \theta \leqslant \pi.$$

Then, obviously,

$$T_n(x) = \cos(n\theta).$$

It can be checked immediately that

$$\begin{aligned} T_0(x) &= cos(0 \cdot \theta) = 1, \\ T_1(x) &= cos(1 \cdot \theta) = \cos(\theta) = x, \\ T_2(x) &= cos(2 \cdot \theta) = 2\cos^2(\theta) - 1 = 2x^2 - 1. \end{aligned}$$

Clearly, T_0, T_1 and T_2 are polynomials of degree 0, 1, and 2, respectively.

Recall the trigonometric addition formulas,

$$\begin{aligned} \cos(\alpha+\beta) &= \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta), \\ \cos(\alpha-\beta) &= \cos(\alpha)\cos(\beta) + \sin(\alpha)\sin(\beta). \end{aligned}$$

Let $n \geqslant 1$, and apply these identities to get

$$\begin{aligned} T_{n+1}(x) &= \cos\left((n+1)\theta\right) = \cos\left(n\theta + \theta\right) = \cos(n\theta)\cos(\theta) - \sin(n\theta)\sin(\theta), \\ T_{n-1}(x) &= \cos\left((n-1)\theta\right) = \cos(n\theta - \theta) = \cos(n\theta)\cos(\theta) + \sin(n\theta)\sin(\theta). \end{aligned}$$

Add these two equations, to obtain

$$T_{n+1}(x) + T_{n-1} = 2\cos(n\theta)\cos(\theta) = 2xT_n(x),$$

where we used that $\cos(\theta) = x$. Rewriting the above relation we get

$$\boxed{T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n \geqslant 1.}$$

This is called **the triple recursion relation** for the Chebyshev polynomials. It is often used in evaluating them, rather than using the explicit formula from definition.

Recall that

$$T_0(x) = 1 \quad \text{and} \quad T_1(x) = x.$$

Then from triple recursion relation we have

$$\begin{aligned}
T_2(x) &= 2xT_1(x) - T_0(x) = 2x \cdot x - 1 = 2x^2 - 1, \\
T_3(x) &= 2xT_2(x) - T_1(x) = 2x(2x^2 - 1) - x = 4x^3 - 3x, \\
T_4(x) &= 2xT_3(x) - T_2(x) = 2x(4x^3 - 3x) - (2x^2 - 1) = 8x^4 - 8x^2 + 1, \\
T_5(x) &= 2xT_4(x) - T_3(x) = 2x(8x^4 - 8x^2 + 1) - (4x^3 - 3x) = 16x^5 - 20x^3 - x.
\end{aligned}$$

We can see that T_n for $n \leqslant 5$ are polynomials of degree n. For the general proof use strong induction and triple recursion relation. Suppose that T_n and T_{n-1} are polynomials of degree n and $n-1$, respectively. Then, obviously $2xT_n$ will be a polynomial of degree $n+1$, and from triple recursion relation it follows that T_{n+1} is a polynomial of degree $n+1$.

The graphs of first 5 Chebyshev polynomials on $[-1, 1]$ are shown below in figure 3. Note that these graphs on $[-1, 1]$ are contained in the unit square. This is a general property for all Chebyshev polynomials.

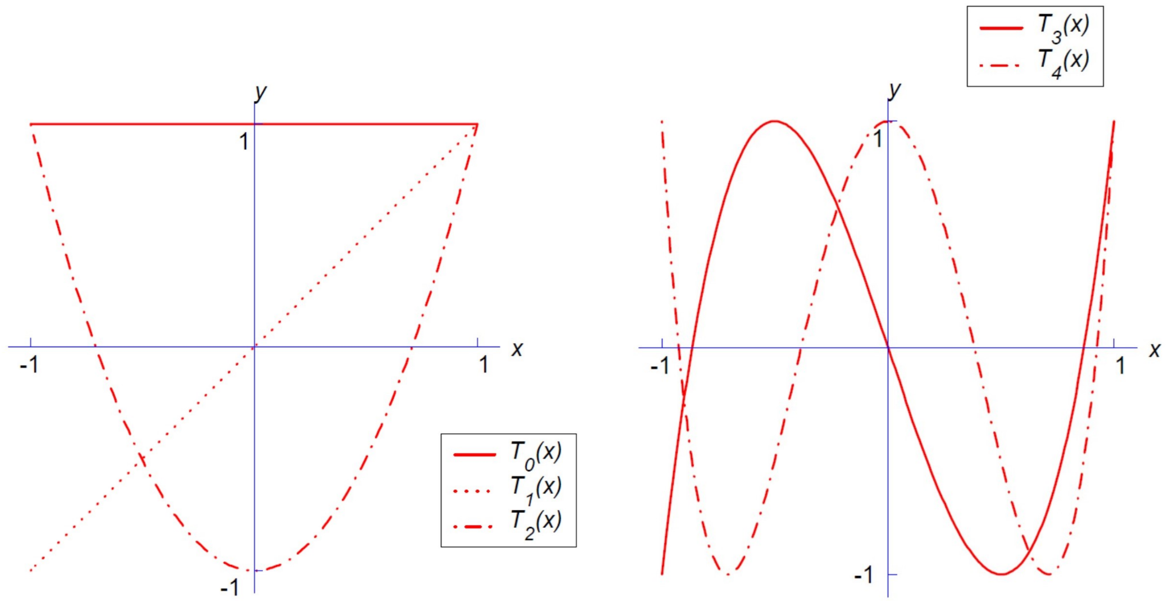

Figure 3: Graphs of Chebyshev polynomials T_n, $n = 0, 1, \ldots, 4$

The minimum size property. Note that for all $n \geqslant 0$ we have

$$\left|T_n(x)\right| \leqslant 1, \quad \text{for all} \quad -1 \leqslant x \leqslant 1.$$

Also note that

$$T_n(x) = 2^{n-1}x^n + \text{ lower degree terms}, \quad n \geqslant 1.$$

These properties can be proven using triple recursion formula and mathematical induction.

Introduce a modified version of $T_n(x)$ (so-called modified Chebyshev polynomial):

$$\widetilde{T}_n(x) = \frac{1}{2^{n-1}} T_n(x) = x^n + \text{ lower degree terms}.$$

From minimum size property we get immediately that

$$\left|\widetilde{T}_n(x)\right| \leqslant \frac{1}{2^{n-1}} \quad \text{for all} \quad -1 \leqslant x \leqslant 1.$$

A polynomial whose highest degree term has a coefficient of 1 is called a **monic polynomial**. Formula for modified Chebyshev polynomial says that it is a monic polynomial and has size $\frac{1}{2^{n-1}}$ on $[-1,1]$, and this becomes smaller as the degree n increases. In comparison,

$$\max_{-1\leqslant x\leqslant 1} \left|x^n\right| = 1.$$

Thus x^n is a monic polynomial whose size does not change with increasing n.

Theorem 2 *Let $n \geqslant 1$ be an integer, and consider all possible monic polynomials of degree n. Then the degree n monic polynomial with the smallest maximum on $[-1,1]$ is the modified Chebyshev polynomial $\widetilde{T}_n(x)$, and its maximum value on $[-1,1]$ is $\frac{1}{2^{n-1}}$.*

This result is used in devising applications of Chebyshev polynomials. We apply it to obtain an improved interpolation scheme.

3 Near-minimax (Chebyshev) approximation method

Let f be a continuous function on interval $[a,b] = [-1,1]$. Consider approximating f by an interpolation polynomial of degree at most $n = 3$. Let x_0, x_1, x_2, x_3 be interpolation node points in $[-1,1]$. Let P_3 be interpolating polynomial of degree $\leqslant 3$ of function f at $\{x_0, x_1, x_2, x_3\}$. The interpolation error is

$$f(x) - P_3(x) = \frac{\Psi_3(x)}{4!} f^{(4)}(\xi), \quad -1 \leqslant x \leqslant 1,$$

where

$$\Psi_3(x) = (x-x_0)(x-x_1)(x-x_2)(x-x_3)$$

and ξ some number (generally unknown) in $[-1,1]$.

The general idea is that we want to choose the nodes $\{x_0, x_1, x_2, x_3\}$ so as to minimize the maximum value of $|f(x) - P_3(x)|$ on interval $[-1,1]$.

From the interpolation error formula we can see that the only general quantity, independent of f is Ψ_3 (which is defined from the nodes). Therefore, we choose interpolation nodes $\{x_0, x_1, x_2, x_3\}$ to minimize

$$\max_{-1\leqslant x\leqslant 1} \left|\Psi_3(x)\right|$$

Expand Ψ_3 to get

$$\Psi_3(x) = x^4 + \text{ lower degree terms.}$$

This is a monic polynomial of degree 4. From the theorem 2 it follows that the smallest possible maximum for monic polynomials on $[-1,1]$ is obtained for modified Chebyshev polynomial. Thus, if we want the smallest possible maximum for Ψ it should be $\widetilde{T}_4$

$$\Psi_3(x) = \widetilde{T}_4(x) = \frac{T_4(x)}{2^3} = \frac{1}{8}(8x^4 - 8x^2 + 1)$$

and the smallest value is $\frac{1}{8}$.

Since $\Psi_3 = (x-x_0)(x-x_1)(x-x_2)(x-x_3)$ it follows immediately that nodes $\{x_i\}_{i=0}^3$ must be the roots of Chebyshev polynomial T_4. It means that we need to solve equation $T_4(x) = 0$,

$$\begin{aligned} T_4(x) = \cos(4\theta) = 0, &\qquad x = \cos(\theta). \\ 4\theta = \pm\tfrac{\pi}{2}, \pm\tfrac{3\pi}{2}, \pm\tfrac{5\pi}{2}, \pm\tfrac{7\pi}{2}, \dots & \\ \theta = \pm\tfrac{\pi}{8}, \pm\tfrac{3\pi}{8}, \pm\tfrac{5\pi}{8}, \pm\tfrac{7\pi}{8}, \dots &\qquad x = \cos(\theta) \\ x = \cos\left(\tfrac{\pi}{8}\right), \cos\left(\tfrac{3\pi}{8}\right), \cos\left(\tfrac{5\pi}{8}\right), \cos\left(\tfrac{7\pi}{8}\right), \dots & \end{aligned}$$

We know that the first four values are distinct and the following ones are repetitive. For example

$$\cos\left(\frac{9\pi}{8}\right) = \cos\left(\frac{7\pi}{8}\right).$$

The first four values are $\{\pm 0.382683, \pm 0.923880\}$, therefore

$$\{x_0 = -0.923880, \; x_1 = -0.382683, \; x_2 = 0.382683, \; x_3 = 0.923880\}.$$

Example 3 *Let $f(x) = e^x$ on interval $[-1, 1]$. Use these four nodes from $[-1, 1]$ to construct the interpolating polynomial C_3 of degree $\leqslant 3$. Polynomial C_3 is called the near-minimax or Chebyshev approximation of function e^x on $[-1, 1]$. We have the following data table*

i	x_i	$f(x_i)$	$D^i(f)$
0	0.923880	2.5190442	2.5190442
1	0.382683	1.4662138	1.9453769
2	−0.382683	0.6820288	0.7047420
3	−0.923880	0.3969760	0.1751757

Table 2: Interpolation data and divided differences for example 3

From the interpolation error formula and the bound of $\frac{1}{8}$ for $|\Psi_3(x)|$ on $[-1, 1]$, we have

$$\|e^x - C_3(x)\|_\infty = \max_{-1\leqslant x\leqslant 1} \left|e^x - C_3(x)\right| \leqslant \frac{1/8}{4!} \max_{-1\leqslant x\leqslant 1} \left|e^\xi\right| \leqslant \frac{e}{192} \approx 0.014158.$$

By direct calculation

$$\|e^x - C_3(x)\|_\infty = \max_{-1\leqslant x\leqslant 1} \left|e^x - C_3(x)\right| \approx 0.00666$$

For comparison, the errors for Taylor polynomial P_3 and minimax approximation M_3 are

$$\|e^x - P_3(x)\|_\infty = \max_{-1\leqslant x\leqslant 1} \left|e^x - P_3(x)\right| \approx 0.0142,$$

$$\rho_3(e^x) \approx 0.00553.$$

We can see that near-minimax (Chebyshev) approximation has error values close to the minimax approximation, best possible approximation by cubic polynomials.

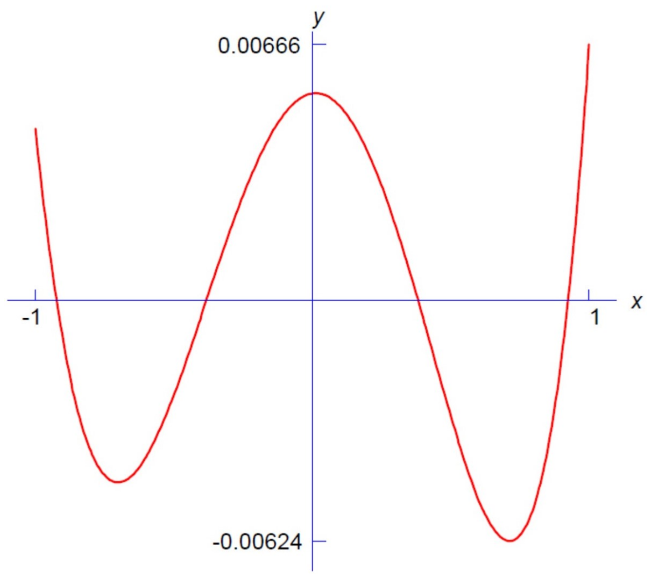

Figure 4: The error $e^x - C_3(x)$

3.1 General case

Consider interpolating function f on $[-1,1]$ by a polynomial of degree $\leqslant n$, with the interpolation nodes $\{x_0, x_1, \dots, x_n\}$ in $[-1,1]$. Denote the interpolation polynomial by C_n. The interpolation error on $[-1,1]$ is given by

$$f(x) - C_n(x) = \frac{\Psi_n(x)}{(n+1)!} f^{(n+1)}(\xi)$$

with

$$\Psi_n(x) = (x - x_0)(x - x_1) \cdots (x - x_n)$$

and ξ some number in $[-1,1]$. In order to minimize the interpolation error, we seek to minimize

$$\max_{-1 \leqslant x \leqslant 1} |\Psi_n(x)|.$$

The polynomial Ψ_n is a monic polynomial of degree $n+1$,

$$\Psi_n(x) = x^{n+1} + \text{ lower degree terms.}$$

From the theorem of the preceding section, this minimum is attained by the monic polynomial

$$\widetilde{T}_{n+1}(x) = \frac{1}{2^n} T_{n+1}(x).$$

Thus, the interpolation nodes must be the roots of Chebyshev polynomial T_{n+1}, and by procedure in the last section, these roots are given by

$$x_j = \cos\left(\tfrac{2j+1}{2n+2}\pi\right), \qquad j = 0, 1, 2, \dots, n.$$

The near-minimax (Chebyshev) approximation of degree n (denoted by C_n) is obtained by interpolating $f(x)$ at these $n+1$ nodes on $[-1,1]$.

Example 4 *Consider again $f(x) = e^x$. The following table contains the maximum errors in C_n on $[-1,1]$ for varying n. For comparison, we also include the corresponding minimax errors. These figures illustrate that for practical purposes, C_n is a satisfactory replacement for the minimax approximation M_n.*

n	$\|e^x - C_n(x)\|_\infty$	$\rho_n(e^x)$
1	$3.72e-01$	$2.79e-01$
2	$5.65e-02$	$4.50e-02$
3	$6.66e-03$	$5.53e-03$
4	$6.40e-04$	$5.47e-04$
5	$5.18e-05$	$4.52e-05$
6	$3.80e-06$	$3.21e-06$

Table 3: Comparison between $\|e^x - C_n(x)\|_\infty$ and $\rho_n(e^x)$

3.2 Other intervals

Near-minimax (Chebyshev) approximations were considered for functions defined on interval $[-1,1]$. Now let's consider the general case of approximating a function f defined on some finite interval $[a,b]$.

Let f be a function defined on $[a,b]$. Introduce the change of variables

$$x = \tfrac{1}{2}\Big((1-t)a + (1+t)b\Big),$$
$$t = \tfrac{2}{b-a}\Big(x - \tfrac{b+a}{2}\Big).$$

Clearly if t varies between -1 and 1, then x takes values from a to b, and vice versa. Introduce the function

$$F(t) = f\left(\frac{1}{2}\Big((1-t)a + (1+t)b\Big)\right), \qquad -1 \leqslant t \leqslant 1$$

Function F defined on $[-1,1]$ is equivalent to function f on $[a,b]$. Now, if we need to approximate function f on $[a,b]$, then we approximate F on $[-1,1]$.

Example 5 *Consider approximating function $f(x) = \cos(x)$ on interval $[0, \frac{\pi}{2}]$. Approximating this function is equivalent to approximating function*

$$F(t) = \cos\left((1+t)\frac{\pi}{4}\right), \qquad -1 \leqslant t \leqslant 1$$

Once near-minimax approximation for $F(t)$ on $[-1,1]$ is computed, substitute $t = \frac{4}{\pi}x - 1$, to get $C_n(x)$ that approximates $f(x)$.

3.3 Lebesgue constants

Recall the Runge's example form Lecture Notes 3, which was caused by evenly spaced interpolation nodes.

Example 6 *Let $f(x) = \frac{1}{1+25x^2}$ on $[-1,1]$ and instead of approximating it with interpolation polynomial on evenly spaced nodes, consider the near-minimax approximation (nodes are the roots of Chebyshev polynomial).*

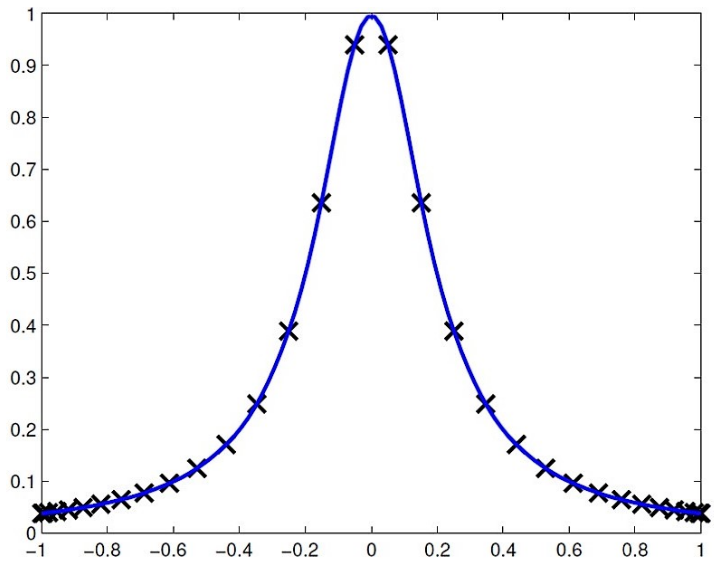

Figure 5: Chebyshev approximation of the function $f(x) = \frac{1}{1+25x^2}$

Notice that, using Chebyshev points for interpolation , we overcame the pathology of Runge's Example. In other words, interpolation points chosen for approximation have big effect on how accurately the interpolation polynomial approximates $f(x)$.

Chebyshev interpolation points were chosen in order to minimize the interpolation error formula for $|f(x) - P_n(x)|$. This formula depends on function $f(x)$ and interpolation partition. On the other hand, we would prefer to have a measure of interpolation accuracy that is independent of f. Something that would measure the quality of interpolation points. This is provided by so-called **Lebesgue constant**.

Definition 2 *Let $\mathcal{P}$ denote a set of interpolation points:*

$$\mathcal{P} = \{x_0, x_1, \ldots, x_n\} \subset [a,b].$$

The Lebesgue constant of partition $\mathcal{P}$ is defined as,

$$\Lambda_n(\mathcal{P}) = \max_{x \in [a,b]} \sum_{k=0}^{n} \big|L_k(x)\big|,$$

where $L_k(x)$ are the Lagrange basis functions associated with interpolation set $\mathcal{P}$.

Small Lebesgue constant means that our interpolation can't be much worse that the best polynomial approximation! In figure 6, there are presented the plots of $\sum_{k=0}^{10} |L_k(x)|$ for uniform partition $\mathcal{P}_{unif}$ and Chebyshev partition $\mathcal{P}_{cheb}$ with 11 data points in $[-1,1]$, in figure 7 with 21 data points and in figure 8, with 31 data points in $[-1,1]$.

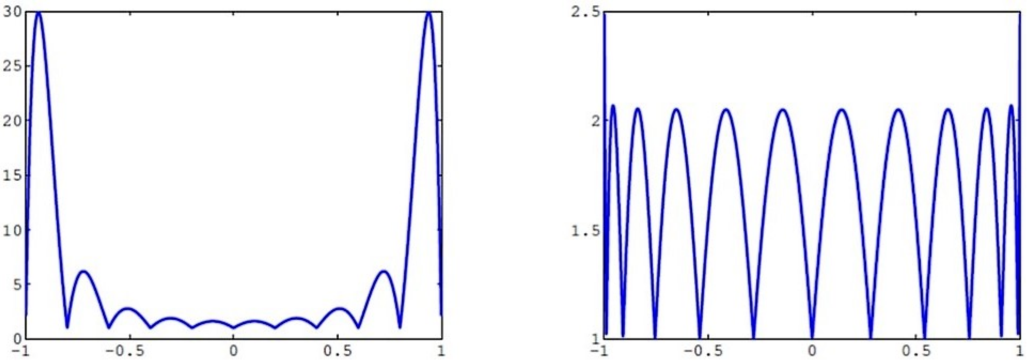

Figure 6: $\Lambda_{10}(\mathcal{P}_{unif}) \approx 29.9$ $\Lambda_{10}(\mathcal{P}_{cheb}) \approx 2.49$

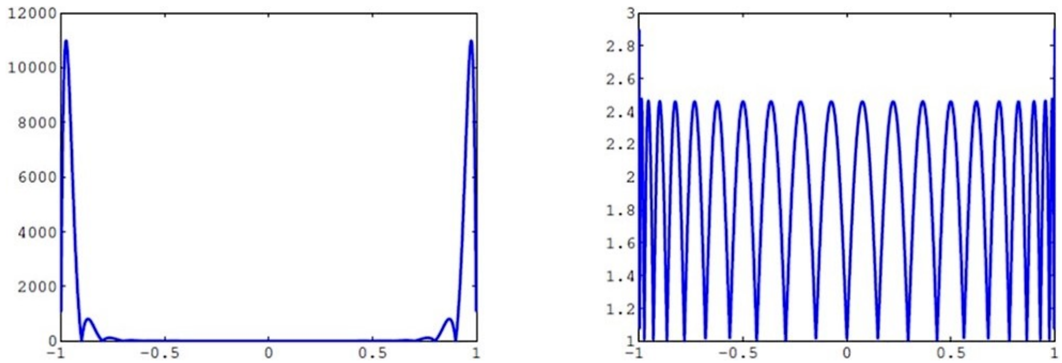

Figure 7: $\Lambda_{20}(\mathcal{P}_{unif}) \approx 10987$ $\Lambda_{20}(\mathcal{P}_{cheb}) \approx 2.9$

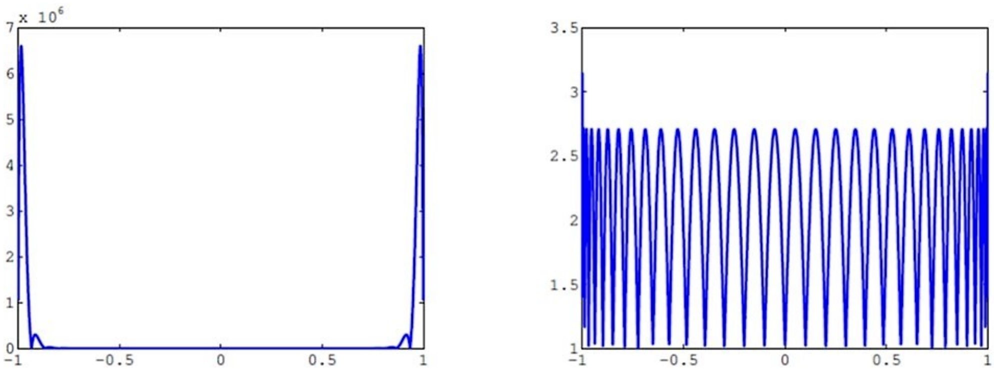

Figure 8: $\Lambda_{30}(\mathcal{P}_{unif}) \approx 6600000$ $\Lambda_{30}(\mathcal{P}_{cheb}) \approx 3.15$

The explosive growth of $\Lambda_n(\mathcal{P}_{unif})$ is an explanation for Runge's example pathology. Also, it has been shown that as $n \to \infty$,

$$\Lambda_n(\mathcal{P}_{unif}) \approx \frac{2^n}{e \cdot n \log n},$$

whereas

$$\Lambda_n(\mathcal{P}_{cheb}) < \frac{2}{\pi} \log(n+1) + 1.$$

In other words, as $n \to \infty$ Lebesgue constants associated with uniform partitions $\Lambda_n(\mathcal{P}_{unif})$ will increase fast to infinity, but the Lebesgue constants associated to Chebyshev partitions will be bounded by $\log(n)$.