Prof. R. Hiptmair
G. Alberti,
F. Leonardi

AS 2015

Numerical Methods for CSE

ETH Zürich
D-MATH

Problem Sheet 9

Problem 1. Chebychev interpolation of analytic functions (core problem)

This problem concerns Chebychev interpolation (cf. [1, Section 4.1.3]). Using techniques from complex analysis, notably the residue theorem [1, Thm. 4.1.56], in class we derived an expression for the interpolation error [1, Eq. (4.1.62)] and from it an error bound [1, Eq. (4.1.63)], as much sharper alternative to [1, Thm. 4.1.37] and [1, Lemma 4.1.41] for *analytic* interpolands. The bound tells us that for all $t \in [a,b]$

$$|f(t) - \mathsf{L}_{\mathcal{T}} f(t)| \leq \left| \frac{w(x)}{2\pi i} \int_\gamma \frac{f(z)}{(z-t)w(z)} dz \right| \leq \frac{|\gamma|}{2\pi} \frac{\max_{a \leq \tau \leq b} |w(\tau)|}{\min_{z \in \gamma} |w(z)|} \frac{\max_{z \in \gamma} |f(z)|}{d([a,b],\gamma)} ,$$

where $d([a,b],\gamma)$ is the geometric distance of the integration contour $\gamma \subset \mathbb{C}$ from the interval $[a,b] \subset \mathbb{C}$ in the complex plane. The contour γ must be contractible in the domain D of analyticity of f and must wind around $[a,b]$ exactly once, see [1, Fig. 124].

Now we consider the interval $[-1,1]$. Following [1, Rem. 4.1.87], our task is to find an upper bound for this expression, in the case where f possesses an analytical extension to a complex neighbourhood of $[-1,1]$.

For the analysis of the Chebychev interpolation of analytic functions we used the elliptical contours, see [1, Fig. 138],

$$\gamma_\rho(\theta) := \cos(\theta - i \log(\rho)) , \quad \forall 0 \leq \theta \leq 2\pi , \quad \rho > 1 . \tag{39}$$

(1a) ⚁ Find an upper bound for the length $|\gamma_\rho|$ of the contour γ_ρ.

HINT: You may use the arc-length formula for a curve $\gamma : I \to \mathbb{R}^2$:

$$|\gamma| = \int_I \|\dot{\gamma}(\tau)\| d\tau, \tag{40}$$

where $\dot{\gamma}$ is the derivative of γ w.r.t the parameter τ. Recall that the "length" of a complex number z viewed as a vector in $\mathbb{R}^2$ is just its modulus.

Solution: $\frac{\partial \gamma_\rho(\theta)}{\partial \theta} := -\sin(\theta - i\log(\rho))$, therefore:

$$
\begin{aligned}
|\gamma_\rho| &= \int_{[0,2\pi]} |\sin(\tau - i\log(\rho))| d\tau && (41)\\
&= \frac{1}{2\rho}\int_{[0,2\pi]} \sqrt{\sin^2(\tau)(-1+\rho^2)^2 + \cos^2(\tau)(1+\rho^2)^2}\, d\tau && (42)\\
&\leq \frac{1}{2\rho}\int_{[0,2\pi]} \sqrt{2+2\rho^4}\, d\tau \qquad \leq \frac{1}{\rho}\pi\sqrt{2(1+\rho^4)} && (43)
\end{aligned}
$$

Now consider the S-curve function (the logistic function):

$$
f(t) := \frac{1}{1+e^{-3t}}\ , \quad t \in \mathbb{R}\ .
$$

(1b) Determine the maximal domain of analyticity of the extension of f to the complex plane $\mathbb{C}$.

HINT: Consult [1, Rem. 4.1.64].

Solution: f is analytic in $D := \mathbb{C} \setminus \{\frac{2}{3}\pi i c - \frac{1}{3}\pi i \mid c \in \mathbb{Z}\}$. In fact, $g(t) := 1 + exp(-3t)$ is an entire function, whereas $h(x) := \frac{1}{x}$ is analytic in $\mathbb{C}\setminus\{0\}$. Therefore, using [1, Thm. 4.1.65], f is analytic in $\mathbb{C} \setminus \{z \in \mathbb{C} \mid g(z) = 0\} =: \mathbb{C} \setminus S$. Let $z := a + ib$, $a, b \in \mathbb{R}$. Since:

$$
\begin{aligned}
-1 = \exp(z) = \exp(a+ib) = \exp(a)(\cos(b) + i\sin(b)) &\Leftrightarrow a = 0, b \in 2\pi\mathbb{Z} + \pi && (44)\\
\exp(z) = -1 &\Leftrightarrow z \in i(2\pi\mathbb{Z} + \pi) && (45)\\
\exp(-3z) = -1 &\Leftrightarrow z \in \frac{i(2\pi\mathbb{Z} - \pi)}{3} && (46)
\end{aligned}
$$

Therefore $S = \frac{2}{3}\pi i\mathbb{Z} - \frac{1}{3}\pi i$.

(1c) Write a MATLAB function that computes an approximation M of:

$$
\min_{\rho > 1} \frac{\max_{z\in\gamma_\rho} |f(z)|}{d([-1,1], \gamma_\rho)}, \tag{47}
$$

by sampling, where the distance of $[a, b]$ from γ_ρ is formally defined as

$$
d([a,b], \gamma) := \inf\{|z - t| \mid z \in \gamma, t \in [a,b]\}. \tag{48}
$$

HINT: The result of (1b), together with the knowledge that γ_ρ describes an ellipsis, tells you the maximal range $(1, \rho_{max})$ of ρ. Sample this interval with 1000 equidistant steps.

HINT: Apply geometric reasoning to establish that the distance of γ_ρ and $[-1,1]$ is $\frac{1}{2}(\rho + \rho^{-1}) - 1$.

HINT: If you cannot find ρ_{max} use $\rho_{max} = 2.4$.

HINT: You can exploit the properties of $\cos$ and the hyperbolic trigonometric functions $\cosh$ and $\sinh$.

Solution: The ellipse must be restricted such that the minor axis has length $\leq 2\pi/3$ (2 times the smallest point, in absolute value, where f is not-analytic). Since this corresponds to the imaginary part of $\gamma_\delta(\theta)$, when $\theta = \pi/2$, we find:

$$\cos(\pi/2 - i\log(\rho_{max})) = 1/3\pi i \Leftrightarrow \sinh(\log(\rho_{max})) = \pi/3 \Leftrightarrow \rho_{max} = \exp(\sinh^{-1}(\pi/3)).$$

See `cheby_approx.m` and `cheby_analytic.m` for the MATLAB code.

(1d) Based on the result of (1c), and [1, Eq. (4.1.89)], give an "optimal" bound for

$$\|f - L_n f\|_{L^\infty([-1,1])},$$

where L_n is the operator of Chebychev interpolation on $[-1,1]$ into the space of polynomials of degree $\leq n$.

Solution: Let M be the approximation of (1c). Then

$$\|f - L_n f\|_{L^\infty([-1,1])} \lesssim \frac{M\sqrt{2(\rho^{-2} + \rho^2)}}{\rho^{n+1} - 1}.$$

(1e) Graphically compare your result from (1d) with the measured supremum norm of the approximation error of Chebychev interpolation of f on $[-1,1]$ for polynomial degree $n = 1, \ldots, 20$. To that end, write a MATLAB-code and rely on the provided function `intpolyval` (cf. [1, Code 4.4.12]).

HINT: Use semi-logarithmic scale for your plot `semilogy`.

Solution: See `cheby_analytic.m`.

(1f) Rely on pullback to $[-1,1]$ to discuss how the error bounds in [1, Eq. (4.1.89)] will change when we consider Chebychev interpolation on $[-a, a], a > 0$, instead of $[-1,1]$, whilst keeping the function f fixed.

Solution: The rescaled function $\Phi^* f$ will have a different domain of analyticity and a different growth behavior in the complex plane. The larger a, the closer the pole of $\Phi^* f$ will move to $[-1,1]$, the more the choice of the ellipses is restricted (i.e. ρ_{max} becomes smaller). This will result in a larger bound.

Using [1, Eq. (4.1.99)], if follows immediately that the asymptotic behaviour of the interpolation does not change after rescaling of the interval. In fact, if Φ^* is the affine pullback from $[-a,a]$ to $[-1,1]$, then:

$$\|f - (\hat{L_n})f\|_{L^\infty([-a,a])} = \|\Phi^* f - L_n \Phi^* f\|_{L^\infty([-1,1])},$$

where $(\hat{L_n})$ is the interpolation on $[-a,a]$.

Problem 2. Piecewise linear approximation on graded meshes (core problem)

One of the messages given by [1, Section 4.1.3] is that the quality of an interpolant depends heavily on the choice of the interpolation nodes. If the function to be interpolated has a "bad behavior" in a small part of the domain, for instance it has very large derivatives of high order, more interpolation points are required in that area of the domain. Commonly used tools to cope with this task, are *graded meshes*, which will be the topic of this problem.

Given a mesh $\mathcal{T} = \{0 \leq t_0 < t_1 < \cdots < t_n \leq 1\}$ on the unit interval $I = [0,1]$, $n \in \mathbb{N}$, we define the *piecewise linear* interpolant

$$\mathsf{I}_{\mathcal{T}} : C^0(I) \to \mathcal{P}_{1,\mathcal{T}} = \{s \in C^0(I),\ s_{|[t_{j-1},t_j]} \in \mathcal{P}_1\ \forall\ j\}, \quad \text{s.t.} \quad \big(\mathsf{I}_{\mathcal{T}} f\big)(t_j) = f(t_j), \quad j = 0,\dots,n;$$

(see also [1, Section 3.3.2]).

(2a) If we choose the uniform mesh $\mathcal{T} = \{t_j\}_{j=0}^n$ with $t_j = j/n$, given a function $f \in C^2(I)$, what is the asymptotic behavior of the error

$$\|f - \mathsf{I}_{\mathcal{T}} f\|_{L^\infty(I)},$$

when $n \to \infty$?

HINT: Look for a suitable estimate in [1, Section 4.5.1].

Solution: Equation [1, (4.5.12)] says

$$\|f - \mathsf{I}_{\mathcal{T}} f\|_{L^\infty(I)} \leq \frac{1}{2n^2} \left\| f^{(2)} \right\|_{L^\infty(I)},$$

because the meshwidth is $h = 1/n$. So, the convergence is quadratic, i.e., algebraic with order 2.

(2b) What is the regularity of the function

$$f : I \to \mathbb{R}, \qquad f(t) = t^\alpha, \qquad 0 < \alpha < 2\,?$$

In other words, for which $k \in \mathbb{N}$ do we have $f \in C^k(I)$?

HINT: Notice that I is a closed interval and check the continuity of the derivatives in the endpoints of I.

Solution: If $\alpha = 1$, $f(t) = t$ clearly belongs to $C^\infty(I)$. If $0 < \alpha < 1$, $f'(t) = \alpha t^{\alpha-1}$ blows up to infinity for t going to 0, therefore $f \in C^0(I) \setminus C^1(I)$. If $1 < \alpha < 2$, f' is continuous but $f''(t) = \alpha(\alpha-1)t^{\alpha-2}$ blows up to infinity for t going to 0, therefore $f \in C^1(I) \setminus C^2(I)$.

More generally, for $\alpha \in \mathbb{N}$ we have $f(t) = t^\alpha \in C^\infty(I)$; on the other hand if $\alpha > 0$ is not an integer, $f \in C^{\lfloor \alpha \rfloor}(I)$, where $\lfloor \alpha \rfloor = \mathrm{floor}(\alpha)$ is the largest integer not larger than α.

(2c) Study numerically the h-convergence of the piecewise linear approximation of $f(t) = t^\alpha$ ($0 < \alpha < 2$) on uniform meshes; determine the order of convergence using linear regression based on MATLAB's `polyfit`, see [1, Section 4.5.1].

HINT: Linear regression and `polyfit` have not been introduced yet. Please give a quick look at the examples in http://ch.mathworks.com/help/matlab/ref/polyfit.html#examples to see `polyfit` in action. For instance, the code to determine the slope of a line approximating a sequence of points $(x_i, y_i)_i$ in doubly logarithmic scale is

```
P = polyfit(log(x),log(y),1);
slope = P(1);
```

Solution: The interpolant is implemented in Listing 41, the convergence for our choice of f is studied in file `PWlineConv.m` and the results are plotted in Figure 17. The convergence is clearly algebraic, the rate is equal to α if it is smaller than 2, and equal to 2 otherwise. In brief, we can say that the order is $\min\{\alpha, 2\}$.

Be careful with the case $\alpha = 1$: here the interpolant gets exactly the solution, with every mesh.

Listing 41: Piecewise linear interpolation.

```
function  y_ev = PWlineIntp (t,y,t_ev)
```

```
% compute and evaluate piecewise linear interpolant
% t and y      data vector of the same size
% t_ev         vector with evaluation points
% --> y_ev     column vector with evaluations in t_ev

t=t(:);  y = y(:);  t_ev = t_ev(:);
n = size(t,1)-1;     % # intervals
if n≠size(y,1)-1; error('t, y must have the same size'); end;

[t,I] = sort(t);     % sort t and y if not sorted
y = y(I);

y_ev = zeros(size(t_ev));
for k=1:n
    t_left  = t(k);
    t_right = t(k+1);
    ind = find ( t_ev ≥ t_left & t_ev < t_right );
    y_ev(ind) = y(k) + (y(k+1)-y(k))/(t_right-t_left)*(t_ev(ind)-t_left);
end
% important! take care of last node:
y_ev(find(t_ev == t(end))) = y(end);
```

(2d) ⚃ In which mesh interval do you expect $|f - \mathsf{I}_{\mathcal{T}} f|$ to attain its maximum?

HINT: You may use the code from the previous subtask to get an idea.

HINT: What is the meaning of [1, Thm. 4.1.37] in the piecewise linear setting?

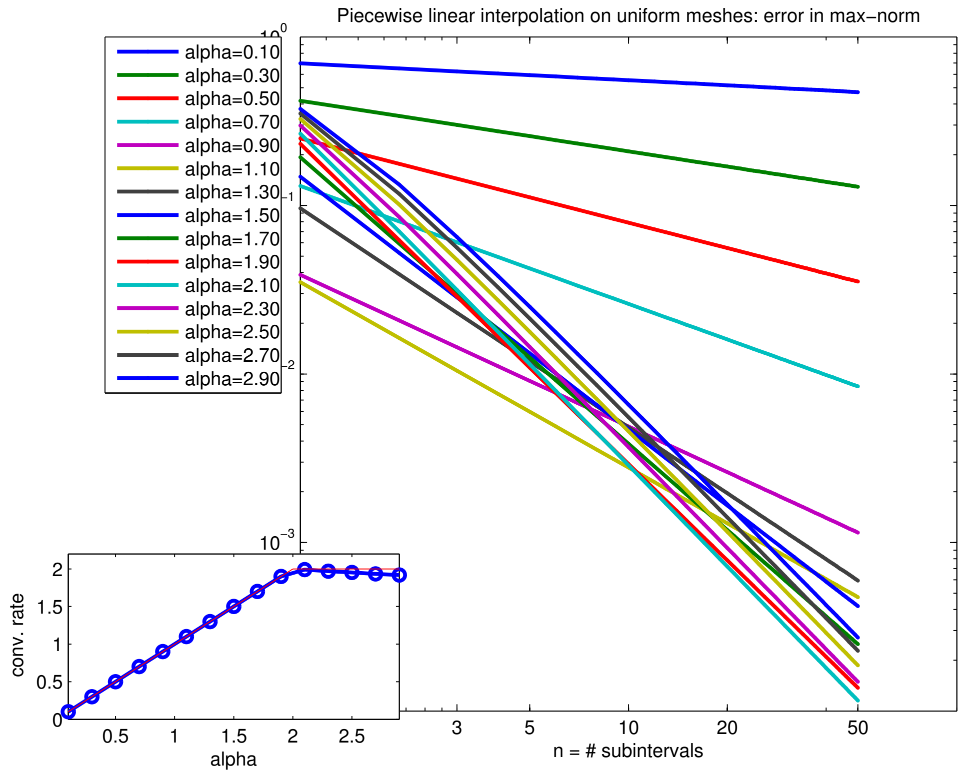
Solution: The error representation [1, (4.1.38)] in the linear case ($n = 1$) reads

$$\forall\, t \in (t_j, t_{j+1}) \qquad \left|f(t) - \left(\mathsf{I}_{\mathcal{T}} f\right)(t)\right| = \frac{1}{2}|f''(\tau_t)\ (t - t_j)(t - t_{j+1})| \le \frac{1}{8}|f''(\tau_t)|\ (t_{j+1} - t_j)^2 = \frac{1}{8n^2}|f''(\tau_t)|,$$

for some $\tau_t \in (t_j, t_{j+1})$. Therefore the error can be large only in the subintervals where the second derivative of f is large. But

$$|f''(t)| = \left|\alpha(\alpha - 1)t^{\alpha-2}\right|$$

Figure 17: h-convergence of piecewise linear interpolation for $f(t) = t^\alpha$, $\alpha = 0.1, 0.3, \dots, 2.9$. The convergence rates are shown in the small plot.



is monotonically decreasing for $0 < \alpha < 2$, therefore we can expect a large error in the first subinterval, the one that is closer to 0.

In line 23 of the code in `PWlineConv.m`, we check our guess: the maximal error is found in the first interval for every $\alpha \in (0, 2)$ $(\alpha \neq 1)$ and in the last one for $\alpha > 2$.

(2e) ⚁ Compute by hand the exact value of $\|f - \mathsf{I}_{\mathcal{T}} f\|_{L^\infty(I)}$.

Compare the order of convergence obtained with the one observed numerically in (2b).

HINT: Use the result of (2d) to simplify the problem.

Solution: From (2d) we expect that the maximum is taken in the first subinterval. For every $t \in (0, 1/n)$ and $0 < \alpha < 2$ ($\alpha \neq 1$) we compute the minimum of the error function φ

$$\begin{aligned}
\varphi(t) &= f(t) - \big(\mathsf{I}_{\mathcal{T}} f\big)(t) = t^\alpha - t\,\frac{1}{n^{\alpha-1}}, \qquad\qquad (\varphi(0) = \varphi(1/n) = 0),\\
\varphi'(t) &= \alpha t^{\alpha-1} - \frac{1}{n^{\alpha-1}},\\
\varphi'(t^*) &= 0 \quad \text{if} \quad t^* = \frac{1}{n}\alpha^{1/(1-\alpha)} \leq \frac{1}{2n},\\
\max_{t\in(0,1/n)} |\varphi(t)| &= |\varphi(t^*)| = \left|\frac{\alpha^{\alpha/(1-\alpha)}}{n^\alpha} - \frac{\alpha^{1/(1-\alpha)}}{n^\alpha}\right| = \frac{1}{n^\alpha}\left|\alpha^{\alpha/(1-\alpha)} - \alpha^{1/(1-\alpha)}\right| = \mathcal{O}(n^{-\alpha}) = \mathcal{O}(h^\alpha).
\end{aligned}$$

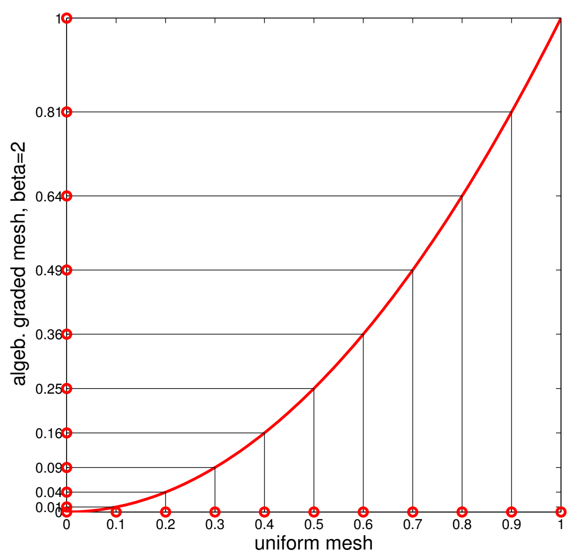
The order of convergence in $h = 1/n$ is equal to the parameter α, as observed in Figure 17.

(2f) ⚃ Since the interpolation error is concentrated in the left part of the domain, it seems reasonable to use a finer mesh only in this part. A common choice is an **algebraically graded mesh**, defined as

$$\mathcal{G} = \left\{ t_j = \left(\frac{j}{n}\right)^\beta, \quad j = 0, \ldots, n \right\},$$

for a parameter $\beta > 1$. An example is depicted in Figure 18 for $\beta = 2$.

Figure 18: Graded mesh $x_j = (j/n)^2$, $j = 0, \ldots, 10$.



For a fixed parameter α in the definition of f, numerically determine the rate of convergence of the piecewise linear interpolant $\mathsf{I}_{\mathcal{G}}$ on the graded mesh $\mathcal{G}$ as a function of the parameter β. Try for instance $\alpha = 1/2$, $\alpha = 3/4$ or $\alpha = 4/3$.

How do you have to choose β in order to recover the optimal rate $\mathcal{O}(n^{-2})$ (if possible)?

Solution: The code in file `PWlineGraded.m` studies the dependence of the convergence rates on β and α. The result for $\alpha = 0.5$ is plotted in Figure 19.

The comparison of this plot with the analogous ones for different values of α suggests that the choice of $\beta = 2/\alpha$ guarantees quadratic convergence, run the code to observe it.

Proceeding as in (2e), we can see that the maximal error in the first subinterval $(0, t_1) = (0, 1/n^{\beta})$ is equal to $1/n^{\alpha\beta}\,(\alpha^{\alpha/(1-\alpha)} - \alpha^{1/(1-\alpha)}) = \mathcal{O}(n^{-\alpha\beta})$. This implies that a necessary condition to have quadratic convergence is $\beta \geq 2/\alpha$. In order to prove un upper bound on the optimal β, we should control the error committed in every subinterval, here the exact computation of $\varphi(t^*)$ becomes quite long and complicate.

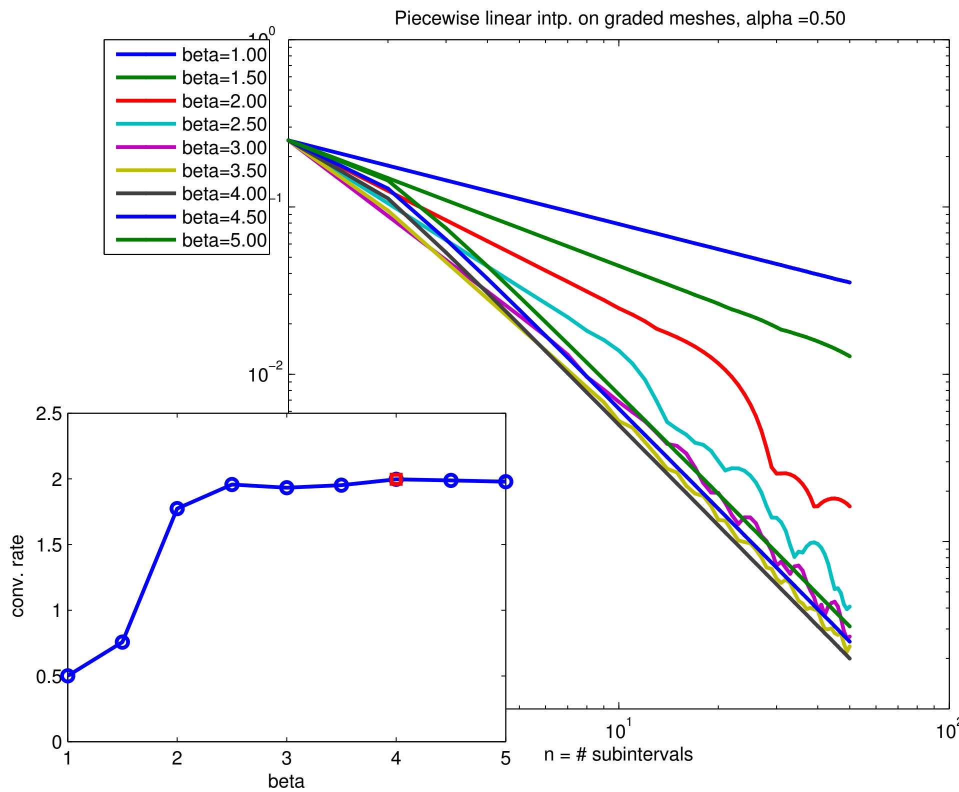
For larger values of the grading parameter, the error in last few subintervals begins to increase. The variable `LocErr` contains the index of the interval where the maximal error is attained (take a look at its values). It confirms that the largest error appears in the first subinterval if $\alpha\beta \ll 2$ and in the last one if $\alpha\beta \gg 2$, the intermediate cases are not completely clear.

Figure 18 has been created with the code in Listing 42.

Listing 42: Plot of algebraically graded mesh.

```
n = 10;                  b = 2;
t_eq = (0:n)/n;          t_gr =t_eq.^b;
close all; plot([0:0.01:1],(0:0.01:1).^b,'r','linewidth',2); hold on;
for j=1:n+1;    plot([t_eq(j),t_eq(j),0], [0,t_gr(j),t_gr(j)],'k');
    plot([t_eq(j),0], [0,t_gr(j)],'ro','linewidth',2); end;
axis square;  xlabel('uniform mesh','fontsize',14);
ylabel('algeb. graded mesh, beta=2','fontsize',14);
set(gca,'xtick',t_eq,'ytick',t_gr);print -depsc2 'GradedMesh.eps';
```

Figure 19: h-convergence of piecewise linear interpolation for $f(t) = t^{\alpha}$, $\alpha = 0.5$, on algebraically graded meshes with parameters $\beta \in [1, 5]$. The convergence rates in dependence on β are shown in the small plot.



Problem 3. Chebyshev polynomials and their properties

Let $T_n \in \mathcal{P}_n$ be the n-th Chebyshev polynomial, as defined in [1, Def. 4.1.67] and $\xi_0^{(n)}, \ldots, \xi_{n-1}^{(n)}$ be the n zeros of T_n. According to [1, Eq. (4.1.75)], these are given by

$$\xi_j^{(n)} = \cos\left(\frac{2j+1}{2n}\pi\right), \quad j = 0, \ldots, n-1. \tag{49}$$

We define the family of discrete L^2 semi inner products, *cf.* [1, Eq. (4.2.21)],

$$(f, g)_n := \sum_{j=0}^{n-1} f(\xi_j^{(n)}) g(\xi_j^{(n)}), \quad f, g \in C^0([-1, 1]) \tag{50}$$

and the special weighted L^2 inner product

$$(f,g)_w := \int_{-1}^{1} \frac{1}{\sqrt{1-t^2}} f(t)g(t)\,dt \quad f,g \in C^0([-1,1]) \tag{51}$$

(3a) Show that the Chebyshev polynomials are an orthogonal family of polynomials with respect to the inner product defined in (51) according to [1, Def. 4.2.24], namely $(T_k, T_l)_w = 0$ for every $k \neq l$.

HINT: Recall the trigonometric identity $2\cos(x)\cos(y) = \cos(x+y) + \cos(x-y)$.

Solution: For $k, l = 0, \ldots, n$ with $k \neq l$, by using the substitution $s = \arccos t$ ($ds = -\frac{1}{\sqrt{1-t^2}}\,dt$) and simple trigonometric identities we readily compute

$$\begin{aligned}
(T_k, T_l)_w &= \int_{-1}^{1} \frac{1}{\sqrt{1-t^2}} \cos(k \arccos t)\cos(l \arccos t)\,dt \\
&= \int_0^{\pi} \cos(ks)\cos(ls)\,ds \\
&= \frac{1}{2}\int_0^{\pi} \cos((k+l)s) + \cos((k-l)s)\,ds \\
&= \frac{1}{2}\left([\sin((k+l)s)]_0^{\pi} + [\sin((k-l)s)]_0^{\pi}\right) \\
&= 0,
\end{aligned}$$

since $k + l \neq 0$ and $k - l \neq 0$.

Consider the following statement.

> **Theorem**. The family of polynomials $\{T_0, \ldots, T_n\}$ is an orthogonal basis ($\rightarrow$ [1, Def. 4.2.13]) of $\mathcal{P}_n$ with respect to the inner product $(\ ,\ )_{n+1}$ defined in (50).

(3b) Write a C++ code to test the assertion of the theorem.

HINT: [1, Code 4.1.70] demonstrates the efficient evaluation of Chebychev polynomials based on their 3-term recurrence formula from [1, Thm. 4.1.68].

Solution: As a consequence of [1, Thm. 4.1.68], we already know that $\{T_0, \ldots, T_n\}$ is a basis for $\mathcal{P}_n$. We check the orthogonality in the C++ code given in Listing 43.

Listing 43: Orthogonality of Chebyshev polynomials

```
#include <iostream>
```

```
#include <math.h>
#include <vector>
#include <Eigen/Dense>

using namespace std;

//Evaluate the Chebyshev polynomials up to order n in x.
vector<double> chebpolmult(const int &n,const double &x)
{
    vector<double> V={1,x};
    for (int k=1; k<n; k++)
        V.push_back(2*x*V[k]-V[k-1]);
    return V;
}

//Check orthogonality of Chebyshev polynomials.
int main(){
    int n=10;
    vector<double> V;
    Eigen::MatrixXd scal(n+1,n+1);
    for (int j=0; j<n+1; j++) {
        V=chebpolmult(n,cos(M_PI*(2*j+1)/2/(n+1)));
        for (int k=0; k<n+1; k++) scal(j,k)=V[k];
    }
    cout<<"Scalar products: "<<endl;
    for (int k=0; k<n+1; k++)
        for (int l=k+1; l<n+1; l++)
            cout<<scal.col(k).dot(scal.col(l))<<endl;
}
```

(3c) ⚃ Prove the theorem.

HINT: Use the relationship of trigonometric functions and the complex exponential together with the summation formula for geometric sums.

Solution: It remains to check the orthogonality condition, namely that $(T_k, T_l)_n = 0$ for

$k \neq l$. For $k, l = 0, \dots, n$ with $k \neq l$ by (49) we have

$$
\begin{aligned}
(T_k, T_l)_n &= \sum_{j=0}^{n} T_k(\xi_j^{(n)}) T_l(\xi_j^{(n)}) \\
&= \sum_{j=0}^{n} \cos\left(k \frac{2j+1}{2(n+1)}\pi\right) \cos\left(l \frac{2j+1}{2(n+1)}\pi\right) \\
&= \frac{1}{2} \sum_{j=0}^{n} (\cos\left((k+l)\frac{2j+1}{2(n+1)}\pi\right) + \cos\left((k-l)\frac{2j+1}{2(n+1)}\pi\right)).
\end{aligned}
\tag{52}
$$

It is now enough to show that

$$
\sum_{j=0}^{n} \cos\left(m \frac{2j+1}{2(n+1)}\pi\right) = 0, \qquad m \in \mathbb{Z}^*. \tag{53}
$$

In order to verify this, observe that

$$
\sum_{j=0}^{n} \cos\left(m \frac{2j+1}{2(n+1)}\pi\right) = \mathrm{Re}(\sum_{j=0}^{n} e^{im\frac{2j+1}{2(n+1)}\pi}) = \mathrm{Re}(e^{im\frac{1}{2(n+1)}\pi} \sum_{j=0}^{n} e^{im\frac{j}{n+1}\pi}).
$$

Finally, by the standard formula for the geometric sum we have

$$
\sum_{j=0}^{n} e^{im\frac{j}{n+1}\pi} = \frac{1 - e^{im\frac{n+1}{n+1}\pi}}{1 - e^{im\frac{1}{n+1}\pi}} = \frac{1-1}{1 - e^{im\frac{1}{n+1}\pi}} = 0,
$$

as desired.

(3d) Given a function $f \in C^0([-1,1]$, find an expression for the best approximant $q_n \in \mathcal{P}_n$ of f in the discrete L^2-norm:

$$
q_n = \operatorname*{argmin}_{p \in \mathcal{P}_n} \|f - p\|_{n+1},
$$

where $\| \ \|_{n+1}$ is the norm induced by the scalar product $(\ ,\ )_{n+1}$. You should express q_n through an expansion in Chebychev polynomials of the form

$$
q_n = \sum_{j=0}^{n} \alpha_j T_j \tag{54}
$$

for suitable coefficients $\alpha_j \in \mathbb{R}$.

HINT: The task boils down to determining the coefficients α_j. Use the theorem you have just proven and a slight extension of [1, Cor. 4.2.14].

Solution: In view of the theorem, the family $\{T_0, \dots, T_n\}$ is an orthogonal basis of $\mathcal{P}_n$ with respect to the inner product $(\ ,\ )_{n+1}$. By (52) and (53) we have

$$\lambda_k^2 := \|T_k\|_{n+1}^2 = (T_k, T_k)_{n+1} = \begin{cases} \frac{1}{2}\sum_{j=0}^{n}(\cos(0) + \cos(0)) = n+1 & \text{if } k = 0, \\ \frac{1}{2}\sum_{j=0}^{n}\cos(0) = (n+1)/2 & \text{otherwise.} \end{cases} \tag{55}$$

The family $\{T_k/\lambda_k : k = 0, \dots, n\}$ is an ONB of $\mathcal{P}_n$ with respect to the inner product $(\ ,\ )_{n+1}$. Hence, by [1, Cor. 4.2.14] we have that

$$q_n = \sum_{j=0}^{n} (f, T_j/\lambda_j)_{n+1} \frac{T_j}{\lambda_j} = \sum_{j=0}^{n} \alpha_j T_j, \qquad \alpha_j = \frac{1}{n+1} \begin{cases} (f, T_j)_{n+1} & \text{if } k = 0, \\ 2(f, T_j)_{n+1} & \text{otherwise.} \end{cases}$$

(3e) ⚁ Write a C++ function

```
template <typename Function>
void bestpolchebnodes(const Function &f, Eigen::VectorXd
    &alpha)
```

that returns the vector of coefficients $(\alpha_j)_j$ in (54) given a function f. Note that the degree of the polynomial is indirectly passed with the length of the output `alpha`. The input `f` is a lambda-function, e.g.

```
auto f = [] (double & x) {return 1/(pow(5*x,2)+1);};
```

Solution: See file `ChebBest.cpp`.

(3f) ⚁ Test `bestpolchebnodes` with the function $f(x) = \frac{1}{(5x)^2+1}$ and $n = 20$. Approximate the supremum norm of the approximation error by sampling on an equidistant grid with 10^6 points.

HINT: Again, [1, Code 4.1.70] is useful for evaluating Chebychev polynomials.

Solution: See file `ChebBest.cpp`. The output (plotted with Matlab) is shown in Figure 20.

(3g) ⚁ Let L_j, $j = 0, \dots, n$, be the Lagrange polynomials associated with the nodes $t_j = \xi_j^{(n+1)}$ of Chebyshev interpolation with $n+1$ nodes on $[-1, 1]$, see [1, Eq. (4.1.75)].

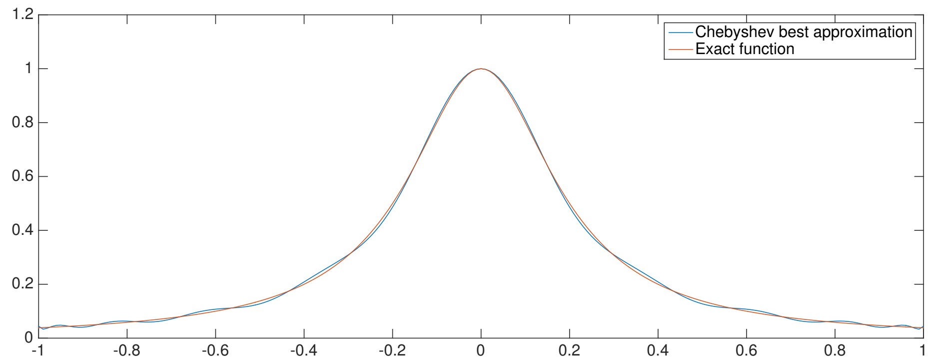


Figure 20: The result of code `ChebBest.m`.

Show that
$$L_j = \frac{1}{n+1} + \frac{2}{n+1}\sum_{l=1}^{n} T_l(\xi_j^{(n+1)})T_l.$$

HINT: Again use the above theorem to express the coefficients of a Chebychev expansion of L_j.

Solution: We have already seen that $\{T_l/\lambda_l : l = 0,\dots,n\}$ is an ONB of $\mathcal{P}_n$. Thus we can write
$$L_j = \sum_{l=0}^{n} \frac{(L_j, T_l)_{n+1}}{\lambda_l^2} T_l = \sum_{l=0}^{n}\sum_{k=0}^{n} L_j(\xi_k^{(n+1)})T_l(\xi_k^{(n+1)})\frac{T_l}{\lambda_l^2}$$
By definition of Lagrange polynomials we have $L_j(\xi_k^{(n+1)}) = \delta_{jk}$, whence
$$L_j = \sum_{l=0}^{n} T_l(\xi_l^{(n+1)})\frac{T_l}{\lambda_l^2}.$$
Finally, the conclusion immediately follows from (55).

Problem 4. Piecewise cubic Hermite interpolation

Piecewise cubic Hermite interpolation with exact slopes on a mesh
$$\mathcal{M} := \{a = x_0 < x_1 < \cdots < x_n = b\}$$
was defined in [1, Section 3.4]. For $f \in C^4([a,b])$ it enjoys h-convergence with rate 4 as we have seen in [1, Exp. 4.5.15].

Now we consider cases, where perturbed or reconstructed slopes are used. For instance, this was done in the context of monotonicity preserving piecewise cubic Hermite interpolation as discussed in [1, Section 3.4.2].

(4a) ⚁ Assume that piecewise cubic Hermite interpolation is based on perturbed slopes, that is, the piecewise cubic function s on $\mathcal{M}$ satisfies:

$$s(x_j) = f(x_j) \quad , \quad s'(x_j) = f'(x_j) + \delta_j,$$

where the δ_j may depends on $\mathcal{M}$, too.

Which rate of asymptotic h-convergence of the $\sup$-norm of the approximation error can be expected, if we know that for all j

$$|\delta_j| = O(h^\beta) \ , \quad \beta \in \mathbb{N}_0 \ ,$$

for mesh-width $h \to 0$.

HINT: Use a local generalized cardinal basis functions, cf. [1, § 3.4.3].

Solution: Let s be the piecewise cubic polynomial interpolant of f. We can rewrite s using the local representation with cardinal basis:

$$\begin{aligned} s(t) &= y_{i-1}H_1(t) + y_i H_2(t) + c_{i-1}H_3(t) + c_i H_4(t) \\ &= y_{i-1}H_1(t) + y_i H_2(t) + (f(t_{i-1}) + \delta_{i-1})H_3(t) + (f(t_i) + \delta_i)H_4(t) \\ &= y_{i-1}H_1(t) + y_i H_2(t) + f(t_{i-1})H_3(t) + f(t_i)H_4(t) + \delta_{i-1}H_3(t) + \delta_i H_4(t) \end{aligned}$$

Hence, if we denote by $\tilde{s}$ the Hermite interpolant with exact slopes:

$$\begin{aligned} \|f - s\|_{L^\infty([a,b])} \le \|f - \tilde{s} + \tilde{s} - s\|_{L^\infty([a,b])} &\le \|f - \tilde{s}\|_{L^\infty([a,b])} + \|\tilde{s} - s\|_{L^\infty([a,b])} \\ &\le O(h^4) + \max_i \|\delta_{i-1}H_3(t) + \delta_i H_4(t)\|_{L^\infty([t_{i-1},t_i])} \\ &= O(h^4) + O(h^{b+1}) = O(\min(4, \beta + 1)) \end{aligned}$$

since $\|H_3(t)\|_{L^\infty([t_{i-1},t_i])} = \|H_4(t)\|_{L^\infty([t_{i-1},t_i])} = O(h)$ (attain maximum at $t = \frac{1}{3h}(t_i - t)$ resp. minimum at $t = \frac{2}{3h}(t_i - t)$, with value $h(\left(\frac{2}{3}\right)^3 - \left(\frac{2}{3}\right)^2)$).

(4b) ⚂ Implement a strange piecewise cubic interpolation scheme in C++ that satisfies:

$$s(x_j) = f(x_j) \quad , \quad s'(x_j) = 0$$

and empirically determine its convergence on a sequence of equidistant meshes of $[-5,5]$ with mesh-widths $h = 2^{-l}, l = 0,\dots,8$ and for the interpoland $f(t) := \frac{1}{1+t^2}$.

As a possibly useful guideline, you can use the provided C++ template, see the file `piecewise_hermite_interpolation_template.cpp`.

Compare with the insight gained in (4a).

Solution: According to the previous subproblem, since $s'(x_j) = f'(x_j) - f'(x_j)$, i.e. $|\delta_j| = O(1)$, $\beta = 0$, the convergence order is limited to $O(h)$.

For the C++ solution, cf. `piecewise_hermite_interpolation.cpp`.

(4c) ⊡ Assume equidistant meshes and reconstruction of slopes by a particular averaging. More precisely, the $\mathcal{M}$-piecewise cubic function s is to satisfy the generalized interpolation conditions

$$s(x_j) = f(x_j),$$
$$s'(x_j) = \begin{cases} \frac{-f(x_2)+4f(x_1)-3f(x_0)}{2h} & \text{for } j = 0 \,, \\ \frac{f(x_{j+1})-f(x_{j-1})}{2h} & \text{for } j = 1,\dots,n-1 \,, \\ \frac{3f(x_n)-4f(x_{n-1})+f(x_{n-2})}{2h} & \text{for } j = n \,. \end{cases}$$

What will be the rate of h-convergence of this scheme (in $\sup$-norm)?

(You can solve this exercise either theoretically or determine an empiric convergence rate in a numerical experiment.)

HINT: If you opt for the theoretical approach, you can use what you have found in sub-subsection (4a). To find perturbation bounds, rely on the Taylor expansion formula with remainder, see [1, Ex. 1.5.58].

Solution: First, we show that the approximation $s'(x_j) = f'(x_j) + O(h^2)$. This follows from Taylor expansion:

$$f(x) = f(x_j) + f'(x_j)(x-x_j) + f''(x_j)(x-x_j)^2/2 + O(h^3)$$

Using $x = x_{j-1}$ and $x = x_{j+1}$ (and $h = x_{j+1} - x_j$):

$$\frac{f(x_{j+1}) - f(x_j)}{h} = f'(x_j) + f''(x_j)h/2 + O(h^2)$$
$$\frac{f(x_{j-1}) - f(x_j)}{h} = -f'(x_j) + f''(x_j)h/2 + O(h^2)$$

Subtracting the second equation to the first equation:

$$\frac{f(x_{j+1}) - f(x_{j-1})}{h} = 2f'(x_j) + O(h^2)$$

For the one-sided difference we expand at $x = x_{j+2}$ and $x = x_{j+1}$:

$$\frac{f(x_{j+1}) - f(x_j)}{h} = f'(x_j) + f''(x_j)h/2 + O(h^2)$$
$$\frac{f(x_{j+2}) - f(x_j)}{2h} = f'(x_j) + f''(x_j)h + O(h^2)$$

Subtracting the first equation to half of the second equation:

$$\frac{f(x_{j+2}) - f(x_j)}{4h} - \frac{f(x_{j+1}) - f(x_j)}{h} = f'(x_j)(1/2 - 1) + O(h^2)$$
$$\frac{f(x_{j+1}) - f(x_j) - 4f(x_{j+1}) + 4f(x_j)}{4h} = f'(x_j)(1/2 - 1) + O(h^2)$$
$$\frac{-f(x_{j+2}) + 4f(x_{j+1}) - 3f(x_j)}{2h} = f'(x_j) + O(h^2)$$

The other side is analogous.

According to the previous subproblem, since $s'(x_j) = f'(x_j) + O(h^2)$ and $\beta = 2$, the convergence order is limited to $O(h^3)$.

For the C++ solution, cf. `piecewise_hermite_interpolation.cpp`.

Issue date: 12.11.2015
Hand-in: 19.11.2015 (in the boxes in front of HG G 53/54).

Version compiled on: November 23, 2015 (v. 1.0).