

ETH Lectures 401-0663-00L Numerical Methods for Computer Science
401-2673-00L Numerical Methods for CSE

Homework Problems and Projects

Prof. R. Hiptmair, SAM, ETH Zurich

Autumn Term 2021, Version of October 2, 2022
(C) Seminar für Angewandte Mathematik, ETH Zürich

[Link](#) to the current version of this homework collection

0.1 General Information

0.1.1 Weekly Homework Assignments

All problems will be published in this single “.pdf” file. Every week, we publish a list of problem numbers on the lecture Moodle page. These are the current assignments for that week. Details are given on the lecture Moodle page.

0.1.2 Importance of Homework

Homework assignments are **not** mandatory. However, it is **very important** that you constantly exercise with the material you learn. “Solving” the homework assignments one week before the main exam by looking at the solutions will likely result in failure.

We provide hints and solutions from the beginning. It is your responsibility to look at the solutions only if you are stuck with a difficult problem and you tried to find a solution for a sufficient amount of time.

Most homework problems come with an estimate for the time required for its solution. These times are what we expect a bright and well-prepared student to require for that problem during an exam.

0.1.3 Corrections and Grading of Assignments

Homework assignments will not be graded.

However, you may submit your handwritten solutions by handing them to your tutor directly or uploading them via the Moodle upload interface, see the lecture Moodle page for instructions. The assistants will

Problem 6-4: Chebyshev polynomials and their properties

Chebyshev polynomials as defined in [Lecture → Def. 6.2.3.3] are a sequence of orthogonal polynomials, which can be defined recursively, see [Lecture → Thm. 6.2.3.4]. In this problem we will examine these polynomials and a few of their many properties.

Involves a little implementation in C++

Let $T_n \in \mathcal{P}_n$ be the n -th Chebyshev polynomial, as defined in [Lecture → Def. 6.2.3.3] and $\xi_0^{(n)}, \dots, \xi_{n-1}^{(n)}$ be the n zeros of T_n , the **Chebyshev nodes**. According to [Lecture → Eq. (6.2.3.10)], these are given by:

$$\xi_j^{(n)} = \cos\left(\frac{2j+1}{2n} \pi\right), \quad j = 0, \dots, n-1. \quad (6.4.1)$$

Recall the notion of a “almost” inner product.

Definition [Lecture → Def. 6.3.1.1]. (Semi-)inner product [NIS02]

Let V be a vector space over the field $\mathbb{K}$. A mapping $b : V \times V \rightarrow \mathbb{K}$ is called an **inner product** on V , if it satisfies

- (i) b is **linear** in the first argument: $b(\alpha v + \beta w, u) = \alpha b(v, u) + \beta b(w, u)$ for all $\alpha, \beta \in \mathbb{K}$, $u, v, w \in V$,
- (ii) b is (anti-)symmetric: $b(v, w) = \overline{b(w, v)}$ ($\overline{} \triangleq$ complex conjugation),
- (iii) b is **positive definite**: $v \neq 0 \Leftrightarrow b(v, v) > 0$.

b is a **semi-inner product**, if it still complies with (i) and (ii), but is only positive semi-definite: $b(v, v) \geq 0$ for all $v \in V$.

We define the family of **discrete L^2 semi-inner products** [Lecture → Eq. (6.3.2.4)]:

$$(f, g)_n := \sum_{j=0}^{n-1} f(\xi_j^{(n)}) g(\xi_j^{(n)}), \quad f, g \in C^0([-1, 1]). \quad (6.4.2)$$

We also define the special **weighted L^2 inner product**:

$$(f, g)_w := \int_{-1}^1 \frac{1}{\sqrt{1-t^2}} f(t) g(t) dt \quad f, g \in C^0([-1, 1]). \quad (6.4.3)$$

(6-4.a) ☹️ (10 min.) Show that the weight function in (6.4.3) $w(t) := (1-t^2)^{-\frac{1}{2}}$ satisfies $\int_{-1}^1 |w(t)| dt < \infty$.

HINT 1 for (6-4.a): Perform integration by substitution: $t = \sin s$. ┘

SOLUTION of (6-4.a):

Obviously the weight function is positive so that we can drop the $|\cdot|$. Then a direct computation of the integral by means of the substitution

$$t = \sin s \Rightarrow dt = \cos s ds, \quad -\frac{\pi}{2} \leq s \leq \frac{\pi}{2},$$

yields

$$\int_{-1}^1 \frac{1}{\sqrt{1-t^2}} dt = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{\cos s}{|\cos s|} ds = \pi.$$



(6-4.b) 🕒 (15 min.) [depends on Sub-problem (6-4.a)]

Show that the Chebyshev polynomials are an **orthogonal family of polynomials** with respect to the inner product defined in (6.4.3) in the sense of [Lecture → Def. 6.3.2.7]:

$$(T_k, T_\ell)_w = 0 \quad \text{for all } k, \ell \in \mathbb{N}_0, \quad k \neq \ell. \quad (6.4.4)$$

HINT 1 for (6-4.b): Recall the trigonometric identity $2 \cos(x) \cos(y) = \cos(x+y) + \cos(x-y)$. ┘

SOLUTION of (6-4.b):

For $k, l = 0, \dots, n$ with $k \neq l$, by using the substitution

$$t = \cos s \Leftrightarrow s = \arccos t \Rightarrow ds = -\frac{1}{\sqrt{1-t^2}} dt,$$

and simple trigonometric identities we readily compute:

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

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



Recall the concept of an orthonormal basis:

Definition [Lecture → Def. 6.3.1.14]. Orthonormal basis

A subset $\{b_1, \dots, b_N\}$ of an N -dimensional vector space V with inner product ($\rightarrow ??$) $(\cdot, \cdot)_V$ is an **orthonormal basis (ONB)**, if $(b_k, b_j)_V = \delta_{kj}$.

A basis of $\{b_1, \dots, b_N\}$ of V is called **orthogonal**, if $(b_k, b_j)_V = 0$ for $k \neq j$.

The following is a surprising result.

Theorem 6.4.5. Discrete orthogonality of Chebychev polynomials

The family of polynomials $\{T_0, \dots, T_n\}$ is an orthogonal basis [Lecture → Def. 6.3.1.14] of $\mathcal{P}_n$ with respect to the inner product $(\cdot, \cdot)_{n+1}$ defined in (6.4.2).

(6-4.c) 🕒 (10 min.) Show that for every $n \in \mathbb{N}_0$ $(\cdot, \cdot)_{n+1}$ as defined in (6.4.2) is an inner product on the (real) vector space $\mathcal{P}_n$ of polynomials of degree $\leq n$.

SOLUTION of (6-4.c):

Bi-linearity, symmetry, and non-negativity should be evident for $(\cdot, \cdot)_{n+1}$. Positive definiteness we conclude from

$$(p, p)_{n+1} = \sum_{j=0}^n |p(\xi_j^{(n+1)})|^2 = 0 \Rightarrow p(\xi_j^{(n+1)}) = 0 \quad \forall j = 0, \dots, n \Rightarrow p \equiv 0,$$

because by [Lecture → Thm. 5.2.2.7] a polynomial of degree $\leq n$ that vanishes in $n+1$ distinct points must be zero.



(6-4.d) Write a C++ function

bool checkDiscreteOrthogonality(unsigned int n);

meant to test the assertion of Thm. 6.4.5 numerically.

HINT 1 for (6-4.d): [Lecture → Code 6.2.3.6] demonstrates the efficient evaluation of Chebyshev polynomials based on their 3-term recurrence formula from [Lecture → Thm. 6.2.3.4].

C++11-code 6.4.6: Code [Lecture → Code 6.2.3.6]

```
2 vector<double> chebpolmult(const int &n, const double &x)
3 {
4     vector<double> V={1, x};
5     for (int k=1; k<n; k++)
6         V.push_back(2*x*V[k]-V[k-1]);
7     return V;
8 }
```



HINT 2 for (6-4.d): Revisit and heed [Lecture → Rem. 1.5.3.15] to devise a “numerically safe” test.

SOLUTION of (6-4.d):

As a consequence of [Lecture → Thm. 6.2.3.4], we already know that $\{T_0, \dots, T_n\}$ is a basis for $\mathcal{P}_n$. We check the orthogonality with the C++ code given in (??).

C++ code 6.4.7: Solution of (6-4.d)

```
2 // Check the orthogonality of Chebyshev polynomials
3 n=10;
4 vector<double> V;
5 Eigen::MatrixXd scal(n+1, n+1);
6 for (int j=0; j<n+1; j++) {
7     V=chebpolmult(n, cos(M_PI*(2*j+1)/(2*(n+1))));
8     for (int k=0; k<n+1; k++) scal(j, k)=V[k];
9 }
10 cout<<"Scalar products: "<<endl;
11 for (int k=0; k<n+1; k++)
12     for (int l=k+1; l<n+1; l++)
13         cout<<scal.col(k).dot(scal.col(l))<<endl;
```

Get it on  GitLab ([chebpolyproperties.cpp](#)).



(6-4.e)  Prove Thm. 6.4.5.

HINT 1 for (6-4.e): Use the relationship of trigonometric functions and the complex exponential together with the summation formula for geometric sums. 

SOLUTION of (6-4.e):

It remains to check the orthogonality condition, namely that $(T_k, T_l)_n = 0$ for $k \neq l$. For $k, l = 0, \dots, n+1$ with $k \neq l$, by (6.4.1) we have:

$$\begin{aligned} (T_k, T_l)_{n+1} &= \sum_{j=0}^n T_k(\zeta_j^{(n)}) T_l(\zeta_j^{(n)}) \\ &= \sum_{j=0}^n \cos\left(k \frac{2j+1}{2(n+1)} \pi\right) \cos\left(l \frac{2j+1}{2(n+1)} \pi\right) \\ &= \frac{1}{2} \sum_{j=0}^n \left[\cos\left((k+l) \frac{2j+1}{2(n+1)} \pi\right) + \cos\left((k-l) \frac{2j+1}{2(n+1)} \pi\right) \right] \end{aligned}$$

It is now enough to show that:

$$\sum_{j=0}^n \cos\left(m \frac{2j+1}{2(n+1)} \pi\right) = 0 \quad \forall m \in \mathbb{Z} \setminus \{0\}. \quad (6.4.8)$$

In order to verify this, observe that:

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

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

$$\begin{aligned} e^{im \frac{1}{2(n+1)} \pi} \sum_{j=0}^n e^{im \frac{j}{n+1} \pi} &= e^{im \frac{1}{2(n+1)} \pi} \frac{1 - e^{im \frac{n+1}{n+1} \pi}}{1 - e^{im \frac{1}{n+1} \pi}} \\ &= \frac{1 - e^{im \pi}}{e^{-im \frac{1}{2(n+1)} \pi} - e^{im \frac{1}{2(n+1)} \pi}} = -\frac{1 - e^{im \pi}}{2} \frac{1}{i \sin\left(m \frac{1}{2(n+1)} \pi\right)} \in i \cdot \mathbb{R}. \end{aligned}$$

This implies that

$$\operatorname{Re}(e^{im \frac{1}{2(n+1)} \pi} \sum_{j=0}^n e^{im \frac{j}{n+1} \pi}) = 0,$$

as desired.



(6-4.f) ☐ (20 min.) [depends on Sub-problem (6-4.c) and Thm. 6.4.5]

Given $n, m \in \mathbb{N}$, $m \leq n$, and a function $f \in C^0([-1, 1])$, find an expression for the best approximant $q_m \in \mathcal{P}_m$ of f in the discrete L^2 -norm:

$$q_m = \operatorname{argmin}_{p \in \mathcal{P}_m} \|f - p\|_{n+1}, \quad (6.4.9)$$

where $\|\cdot\|_{n+1}$ is the norm induced by the scalar product $(\cdot, \cdot)_{n+1}$ from (6.4.2). You should represent q_m through a **Chebyshev expansion** of the form:

$$q_m = \sum_{j=0}^m \alpha_j T_j, \quad \alpha_j \in \mathbb{R}, \quad (6.4.10)$$

see also [Lecture → Section 6.2.3.3].

HINT 1 for (6-4.f): The task boils down to determining the coefficients α_j . Use Thm. 6.4.5 and a slight extension of [Lecture → Cor. 6.3.1.15].

Corollary [Lecture → Cor. 6.3.1.15]. ONB representation of best approximant

If $\{b_1, \dots, b_N\}$ is an orthonormal basis [Lecture → Def. 6.3.1.14] of $V \subset X$, then the best approximant $q := \operatorname{argmin}_{p \in V} \|f - p\|_X$ of $f \in X$ has the representation

$$q = \sum_{j=1}^N (f, b_j)_X b_j. \quad [\text{Lecture} \rightarrow \text{Eq. (6.3.1.16)}]$$

SOLUTION of (6-4.f):

In view of the theorem, the family $\{T_0, \dots, T_m\}$ is an orthogonal basis of $\mathcal{P}_m$ with respect to the inner product (→ Sub-problem (6-4.c)) $(\cdot, \cdot)_{n+1}$. By ((6-4.e)) and (6.4.8) we have for $k \in \{0, \dots, n\}$

$$\lambda_k^2 := \|T_k\|_{n+1}^2 = (T_k, T_k)_{n+1} = \begin{cases} \frac{1}{2} \sum_{j=0}^n (\cos(0) + \cos(0)) = n+1 & \text{if } k=0, \\ \frac{1}{2} \sum_{j=0}^n \cos(0) = (n+1)/2 & \text{otherwise.} \end{cases} \quad (6.4.11)$$

Thus the family $\{T_k/\lambda_k : k=0, \dots, m\}$ is an **orthonormal basis** of $\mathcal{P}_m$ with respect to the inner product $(\cdot, \cdot)_{n+1}$. Hence, we apply [Lecture → Cor. 6.3.1.15] with

$$(\cdot, \cdot)_X := (\cdot, \cdot)_{n+1} \quad \text{and} \quad \{b_1, \dots, b_N\} := \left\{ \frac{T_0}{\lambda_0}, \dots, \frac{T_m}{\lambda_m} \right\} \text{ bcom}$$

and get

$$q_m = \sum_{j=0}^m (f, T_j/\lambda_j)_{n+1} \frac{T_j}{\lambda_j} = \sum_{j=0}^m \alpha_j T_j,$$

$$\blacktriangleright \quad \alpha_j := \frac{1}{n+1} \begin{cases} (f, T_0)_{n+1} & \text{if } j=0, \\ 2(f, T_j)_{n+1} & \text{for } j=1, \dots, m. \end{cases} \quad (6.4.12)$$

(6-4.g) (10 min.) [depends on Sub-problem (6-4.f)]

Give an interpretation of q_m as defined through (6.4.9) in the case $m = n$.

SOLUTION of (6-4.g):

If $m = n$ we can achieve $\|f - q_n\|_{n+1} = 0$ by choosing q_n as the **Chebyshev interpolant** of f , that is,

$$q_n \in \mathcal{P}_n: \quad q_n(\xi_j^{(n+1)}) = f(\xi_j^{(n+1)}) \quad \forall j = 0, \dots, n.$$

**(6-4.h)** (20 min.) [depends on Sub-problem (6-4.f)]

Write a C++ function

```
template <typename Function>
Eigen::VectorXd
    bestpolchebnodes(const Function &f,
                    unsigned int n, unsigned int m);
```

that returns the vector of coefficients α_j , $j = 0, \dots, m$, in (6.4.10) given the function f . The degree m of the polynomial and the number $n+1$ of the Chebyshev nodes entering the inner product are passed through the arguments m and n . The input f is a lambda-function of signature `std::function<double(double)>`, for instance,

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

SOLUTION of (6-4.h):

Of course, we rely on (6.4.12), which can be implemented directly.

C++11-code 6.4.13: Solution of (6-4.h)

```
2  template <typename Function>
3  void bestpolchebnodes(const Function &f, Eigen::VectorXd &alpha) {
4      int n=alpha.size()-1;
5      Eigen::VectorXd fn(n+1);
6      for (int k=0; k<n+1; k++) {
7          double temp=cos(M_PI*(2*k+1)/(n+1));
8          fn(k)=f(temp);
9      }
10
11     vector<double> V;
12     Eigen::MatrixXd scal(n+1,n+1);
13     for (int j=0; j<n+1; j++) {
14         V=chebpolmult(n,cos(M_PI*(2*j+1)/(n+1)));
15         for (int k=0; k<n+1; k++) scal(j,k)=V[k];
16     }
17
18     for (int k=0; k<n+1; k++) {
19         alpha(k)=0;
20         for (int j=0; j<n+1; j++) {
21             alpha(k)+=2*fn(j)*scal(j,k)/(n+1);
22         }
23     }
```

```

23     }
24     alpha(0)=alpha(0)/2;
25 }

```

Get it on  [GitLab \(chebpolyproperties.cpp\)](#).



(6-4.i)  (20 min.) [depends on Sub-problem (6-4.h)]

We test the implementation of `bestpolchebnodes()` with the function $f(x) = \frac{1}{(5x)^2+1}$. To that end write a C++ function

```
std::vector<double> testBestPolyChebNodes (unsigned int n);
```

that takes the integer n as argument and both tabulates and returns the sequence of the approximation error norms $\|f - q_m\|_{L^\infty([-1,1])}$, $m = 0, \dots, n$, q_m as in (6.4.9).

Approximate the supremum norm of the approximation error by sampling on an equidistant grid with 10^6 points.

HINT 1 for (6-4.i): Again, [Lecture → Code 6.2.3.6] is useful for evaluating Chebychev polynomials.

└

SOLUTION of (6-4.i):

C++11-code 6.4.14: Solution of (6-4.i)

```

2  // Test the implementation
3  auto f = [] (double & x) { return 1/(pow(5*x,2)+1); };
4  n=20;
5  Eigen::VectorXd alpha(n+1);
6  bestpolchebnodes(f, alpha);
7
8  // Compute the error
9  Eigen::VectorXd X = Eigen::VectorXd::LinSpaced(1e6,-1,1);
10 auto qn = [&alpha,&n] (double & x) {
11     double temp=0;
12     vector<double> V=chebpolmult(n,x);
13     for (int k=0; k<n+1; k++) temp+=alpha(k)*V[k];
14     return temp;
15 };
16 double err_max=abs(f(X(0))-qn(X(0)));
17 for (int i=1; i<1e6; i++) err_max=std::max(err_max,abs(f(X(i))-qn(X(i))));
18 cout<<"Error: "<< err_max <<endl;

```

Get it on  [GitLab \(chebpolyproperties.cpp\)](#).

The output (plotted in MATLAB) is shown in Fig. 57.

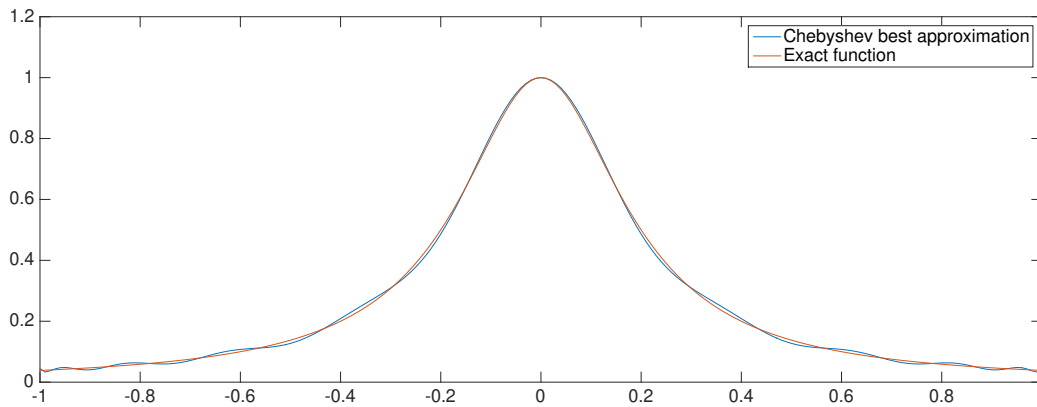


Fig. 57

(6-4.j) (20 min.) [depends on Sub-problem (6-4.f)]

Let $L_j \in \mathcal{P}_n$, $j = 0, \dots, n$, be the **Lagrange polynomials** [Lecture $\rightarrow$ § 5.2.2.3] associated with the Chebyshev nodes $t_j := \xi_j^{(n+1)}$ in $[-1, 1]$. Show that:

$$L_j(t) = \frac{1}{n+1} + \frac{2}{n+1} \sum_{l=1}^n T_l(\xi_j^{(n+1)}) T_l(t) \quad , \quad j = 0, \dots, n. \quad (6.4.15)$$

HINT 1 for (6-4.j): Note that (6.4.15) is a **Chebyshev expansion**. Then use the result of Sub-problem (6-4.f) to express the coefficients of the Chebyshev expansions of L_j .

SOLUTION of (6-4.j):

We have already seen that $\{T_l/\lambda_l : l = 0, \dots, n\}$ is an $(\cdot, \cdot)_{n+1}$ -ONB of $\mathcal{P}_n$. Exploiting the defining property of Lagrange polynomials

$$L_j(\xi_\ell^{(n+1)}) = \delta_{k,\ell} \quad , \quad k, \ell \in \{0, \dots, n\} \quad ,$$

we can thus 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:

$$L_j = \sum_{l=0}^n T_l(\xi_j^{(n+1)}) \frac{T_l}{\lambda_l^2}$$

Finally, the conclusion immediately follows from (6.4.11).

End Problem 6-4 , 125 min.