

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 7-15: Convolution Quadrature

Convolution quadrature is a special numerical technique for the evaluation of the convolution of two causal functions, when one is given in the form of its Laplace transform. This problem studies formulas for the weights of convolution quadrature and their efficient numerical computation.

This problem requires knowledge of the discrete Fourier transform (DFT) [Lecture → Def. 4.2.1.18], its implementation in EIGEN, some complex analysis, and numerical quadrature for periodic analytic functions [Lecture → § 7.5.0.18].

The **convolution** of two causal integrable functions $f, g : \mathbb{R}^+ := [0, \infty[\rightarrow \mathbb{R}$, written as $f * g$, is a function $\mathbb{R}^+ \rightarrow \mathbb{R}$ defined as

$$(f * g)(t) := \int_0^t f(t - \tau)g(\tau) d\tau, \quad t \geq 0. \quad (7.15.1)$$

Given a fixed “timestep size” $\tau > 0$, we approximate $f * g$ at equidistant times $t_n := \tau n$, $n \in \mathbb{N}_0$, by

$$(f * g)(t_n) \approx \sum_{\ell=0}^n w_{n-\ell}^\tau g_\ell, \quad g_\ell := g(t_\ell), \quad n \in \mathbb{N}_0. \quad (7.15.2)$$

with suitable convolution quadrature weights $w_j^\tau \in \mathbb{C}$, $j \in \mathbb{N}_0$.

In the sequel let the function f be fixed and denote by $F : D \subset \mathbb{C} \rightarrow \mathbb{C}$ its **Laplace transform**, which is supposed to be known.

Assumption 7.15.3. Holomorphy of the Laplace transform

The Laplace transform $F : D \subset \mathbb{C} \rightarrow \mathbb{C}$ of f is **analytic** (holomorphic) on an open set $D \subset \mathbb{C}$ with

$$\mathbb{C}^+ := \{z \in \mathbb{C} : \operatorname{Re} z > 0\} \subset D,$$

that is, F is analytic in the right half-plane of $\mathbb{C}$.

As explained in [HAS16] the convolution quadrature weights w_j^τ in (7.15.2) are given by the following complex contour integrals: for any $0 < r < 1$

$$w_j^\tau = \frac{1}{2\pi i} (-1)^j \int_{|z|=r} \frac{1}{z^{j+1}} F\left(\frac{1-z}{\tau}\right) dz, \quad j \in \mathbb{Z}. \quad (7.15.4)$$

With the parameterization $z(\varphi) := re^{2\pi i \varphi}$, $\varphi \in [0, 1[$, we convert (7.15.4) into

$$w_j^\tau = r^{-j} \int_0^1 \psi_j(\varphi) d\varphi, \quad \psi_j(\varphi) := e^{-2\pi i j \varphi} \cdot F\left(\frac{1 - re^{2\pi i \varphi}}{\tau}\right), \quad j \in \mathbb{Z}. \quad (7.15.5)$$

(7-15.a) 🕒 (10 min.)

We identify $\psi_j : \mathbb{R} \rightarrow \mathbb{C}$ from (7.15.5) with its extension to a neighborhood of the real axis in the complex plane. Merely taking for granted Ass. 7.15.3, what is the **maximal** $\rho > 0$ such that ψ_j is analytic in the strip

$$S_\rho := \{z \in \mathbb{C} : |\operatorname{Im} z| < \rho\} ?$$

Give that maximal ρ as a function of r :

$$\rho(r) = \boxed{}.$$

SOLUTION of (7-15.a):

Obviously ψ_j is a 1-periodic function. To emphasize that we now consider ψ_j as a function of a complex argument we write

$$\psi_j(w) := e^{-2\pi i j w} F\left(\frac{1 - re^{2\pi i w}}{\tau}\right), \quad w \in \mathbb{C}.$$

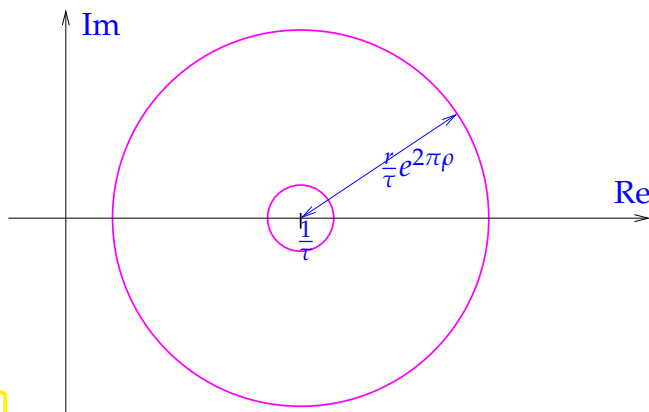
For every $j \in \mathbb{Z}$ the function $w \mapsto e^{2\pi i j w}$ is an entire function (analytic in all of $\mathbb{C}$), so that it does not affect the domain of analyticity of ψ_j . We can safely ignore this forefactor. In particular, the domain of analyticity of ψ_j does not depend on j .

Owing to Ass. 7.15.3, ψ_j is analytic in the pre-image of $\mathbb{C}^+$ under the map $w \mapsto \frac{1 - e^{2\pi i w}}{\tau}$. To begin with, recall that

$$w \mapsto e^{2\pi i w} = e^{-2\pi \operatorname{Im} w} (\cos(2\pi \operatorname{Re} w) + i \sin(2\pi \operatorname{Re} w))$$

maps the strip S_ρ to an annulus:

$$\{w \mapsto e^{2\pi i w}\}(S_\rho) = T_\rho := \{z \in \mathbb{C} : e^{-2\pi\rho} < |z| < e^{2\pi\rho}\}. \quad (7.15.6)$$



Hence, $w \mapsto \frac{1 - e^{2\pi i w}}{\tau}$ maps S_ρ to an annulus shifted by $\frac{1}{\tau}$ and scaled by $\frac{r}{\tau}$.

$\triangleleft$ Domain $\frac{1 - rT_\rho}{\tau}$: image of strip.

Fig. 83

This means that

$$\frac{1}{\tau} - \frac{r}{\tau} T_\rho \subset \mathbb{C}^+ \Rightarrow \psi_j \text{ is analytic in } S_\rho.$$

Geometric consideration reveal, see Fig. 83,

$$\frac{1}{\tau} - \frac{r}{\tau} T_\rho \subset \mathbb{C}^+ \Leftrightarrow 1 - re^{2\pi\rho} > 0 \Leftrightarrow \rho < -\frac{1}{2\pi} \log r.$$

The largest possible width of the strip is $\frac{1}{2\pi} \log 1/r$.

▲

(7-15.b) ☐ (4 min.)

The integral in (7.15.5) is approximated by numerical quadrature using the equidistant $(N+1)$ -point, $N \in \mathbb{N}_0$, **trapezoidal rule**. Write down the formula for the resulting approximate convolution quadrature weights $w_j^{\tau, N}$, $j \in \mathbb{Z}$.

$$w_j^\tau \approx w_j^{\tau, N} =$$

SOLUTION of (7-15.b):

Starting from

$$w_j^\tau = r^{-j} \int_0^1 \psi_j(\varphi) \, d\varphi, \quad \psi_j(\varphi) := e^{-2\pi i j \varphi} F\left(\frac{1 - re^{2\pi i \varphi}}{\tau}\right).$$

and the definition of the equidistant trapezoidal quadrature rule

$$\int_a^b f(t) \, dt \approx T_m(f) := h \left(\frac{1}{2}f(a) + \sum_{k=1}^{m-1} f(kh) + \frac{1}{2}f(b) \right), \quad h := \frac{b-a}{m},$$

[Lecture → Eq. (7.5.0.17)]

applied on $[0, 1]$ with $m = N + 1$ and taking into account the 1-periodicity of the integrand we arrive at

$$w_j^\tau \approx w_j^{\tau, N} = \frac{1}{(N+1)r^j} \sum_{k=0}^N \psi_j\left(\frac{k}{N+1}\right) = \frac{1}{(N+1)r^j} \sum_{k=0}^N e^{-2\pi i j \frac{k}{N+1}} F\left(\frac{1 - re^{2\pi i \frac{k}{N+1}}}{\tau}\right).$$

▲

(7-15.c) 🕒 (25 min.) [depends on Sub-problem (7-15.b)]

In the file `convolutionquadrature.hpp` implement an *efficient* C++ function

template <typename **FFUNCTOR**>

Eigen::VectorXcd **compute_cq_weights**(
FFUNCTOR&& F, **unsigned int** N, **double** tau, **double** r = 0.0);

that computes the approximate convolution quadrature weights $w_j^{\tau, N}$ by means of the $(N+1)$ -point equidistant trapezoidal rule and for $j = 0, \dots, N$, and returns them as the components of a complex column vector. The functor object F passes the Laplace transform F . The parameter r specifies the radius r of the integration contour. If $r=0.0$, then a default value is chosen, see the code template.

“Efficient” means that the function should enjoy an asymptotic computational cost of $O(N \log N)$ for $N \rightarrow \infty$.

SOLUTION of (7-15.c):

In Sub-problem (7-15.b) we derived the formula

$$w_j^{\tau, N} = \frac{1}{(N+1)r^j} \sum_{k=0}^N e^{-2\pi i j \frac{k}{N+1}} F\left(\frac{1 - re^{2\pi i \frac{k}{N+1}}}{\tau}\right).$$

Let us introduce the vector

$$\mathbf{v} = \left[F\left(\frac{1 - re^{2\pi i \frac{k}{N+1}}}{\tau}\right) \right]_{k=0}^N. \quad (7.15.7)$$

Its components can be obtained by sampling F on a circle in $\mathbb{C}^+$ ($r < 1$). Next recall the discrete Fourier transform from [Lecture → Def. 4.2.1.18]:

Definition [Lecture → Def. 4.2.1.18]. Discrete Fourier transform

The linear map $\text{DFT}_n : \mathbb{C}^n \mapsto \mathbb{C}^n$, $\text{DFT}_n(\mathbf{y}) := \mathbf{F}_n \mathbf{y}$, $\mathbf{y} \in \mathbb{C}^n$, is called **discrete Fourier transform** (DFT), i.e. for $[c_0, \dots, c_{n-1}] := \text{DFT}_n(\mathbf{y})$

$$c_k = \sum_{j=0}^{n-1} y_j \omega_n^{kj} = \sum_{j=0}^{n-1} y_j \exp(-2\pi i \frac{kj}{n}) \quad , \quad k = 0, \dots, n-1. \quad [\text{Lecture} \rightarrow \text{Eq. (4.2.1.19)}]$$

Matching (7.15.7) and [Lecture → Eq. (4.2.1.19)] we find

$$\left[w_j^{\tau, N} \right]_{j=0}^N = \mathbf{D} \cdot \text{DFT}_{N+1}(\mathbf{v}), \quad \mathbf{D} = \frac{1}{N+1} \text{diag}(1, r^{-1}, r^{-2}, \dots, r^{-N}). \quad (7.15.8)$$

For the implementation one has to take into account how DFT is done by means of EIGEN's built-in helper class **Eigen::FFT**, see [Lecture → Code 4.2.1.22].

C++ code 7.15.9: Computation of convolution quadrature weights

```

2  template <typename FFUNCTOR>
3  Eigen::VectorXcd compute_cq_weights(FFUNCTOR&& F, unsigned int N, double tau,
4                                     double r = 0.0) {
5      using Comp = std::complex<double>;
6      assert(N >= 1 && "At least two quadrature points have to be used!");
7      // Default choice for circle radius for integration contour
8      if (r == 0.0) {
9          r = std::pow(std::numeric_limits<double>::epsilon(), 0.5 / (N + 1));
10     }
11     Eigen::VectorXcd w(N + 1);
12     const Comp iu = 2.0 * Comp{0.0, 1.0} * M_PI; // 2*pi*i
13     // START student code
14     Eigen::VectorXcd v(N + 1);
15     // Sampling of F on a circle in the complex plane, see (7.15.7)
16     const Comp expfac = iu / ((double)N + 1);
17     for (unsigned int k = 0; k <= N; ++k) {
18         const Comp z = (Comp(1.0, 0.0) - r * std::exp(expfac * (double)k)) / tau;
19         v[k] = F(z);
20     }
21     // Compute DFT of the vector v
22     Eigen::FFT<double> fft;
23     w = fft.fwd(v); // DFT in EIGEN
24     // Finally perform the scaling of the components
25     double fac = N + 1;
26     for (unsigned int k = 0; k <= N; ++k) {
27         w[k] /= fac;
28         fac *= r;
29     }
30     // END Student code
31     return w;
32 }

```

Get it on  GitLab (convolutionquadrature.hpp).

(7-15.d) 🕒 (5 min.) [depends on Sub-problem (7-15.a), Sub-problem (7-15.b)]

Assume computations in exact arithmetic. How will the error introduced into the computation of the convolution quadrature weights by numerical quadrature by means of the $(N+1)$ -point equidistant trapezoidal rule, that is $|w_j^\tau - w_j^{\tau,N}|$, behave as a function of N asymptotically for $N \rightarrow \infty$? Explain your answer.

SOLUTION of (7-15.d):

In Sub-problem (7-15.a) we have seen that the integrand in

$$w_j^\tau = r^{-j} \int_0^1 \psi_j(\varphi) d\varphi$$

is 1-periodic and possesses an analytic extension to a strip in the complex plane parallel to the real axis. Therefore the quadrature error will **converge exponentially** to zero as $N \rightarrow \infty$:

$$\exists q \in [0, 1[: \quad |w_j^\tau - w_j^{\tau,N}| = O(q^N) \quad \text{for } N \rightarrow \infty. \quad (7.15.10)$$

The reason for this behavior is that the $(N+1)$ -point equidistant trapezoidal rule integrates trigonometric polynomials of degree $2N$ exactly, see [Lecture → Lemma 7.5.0.22]. Then the approximation results of [Lecture → Section 6.5.3] can be applied.

Remark. The following lemma from the course

Lemma [Lecture → Lemma 7.5.0.22]. Exact quadrature by equidistant trapezoidal rule

The n -point equidistant trapezoidal quadrature rule

$$\int_a^b f(t) dt \approx T_n(f) := h \left(\frac{1}{2}f(a) + \sum_{k=1}^{n-1} f(a+kh) + \frac{1}{2}f(b) \right), \quad h := \frac{b-a}{n}.$$

[Lecture → Eq. (7.5.0.17)]

is exact for trigonometric polynomials (→ [Lecture → Section 6.5.1]) of degree $\leq 2n-2$.

together with

Theorem [Lecture → Thm. 6.5.3.14]. Exponential convergence of trigonometric interpolation for analytic interpolands

If $f: \mathbb{R} \rightarrow \mathbb{C}$ is 1-periodic and possesses an **analytic extension** to the strip

$$\bar{S} := \{z \in \mathbb{C}: -\eta \leq \operatorname{Im} z \leq \eta\}, \quad \text{for some } \eta > 0,$$

then there is $C_\eta > 0$ depending only on η such that

$$\|f - T_n f\|_* \leq C_\eta e^{-\pi\eta n} \|f\|_{L^\infty(\bar{S})}, \quad n \in \mathbb{N}, \quad (* = L^2([0,1]), L^\infty([0,1])).$$

[Lecture → Eq. (6.5.3.15)]

tells us how the width ρ of the strip of analyticity determines the number q in (7.15.10):

$$q = \exp(-\pi\rho).$$



End Problem 7-15 , 44 min.