universität wien

MASTERARBEIT

Titel der Masterarbeit

"Gaussian quadrature rules and the evaluation of a family of integrals"

Verfasser

Helene Ranetbauer, BSc

angestrebter akademischer Grad

Master of Science (MSc)

Wien, im März 2014

Studienkennzahl lt. Studienblatt:	A 066 821
Studienrichtung lt. Studienblatt:	Masterstudium Mathematik
Betreuer:	ao. Univ.-Prof. Dipl.-Ing. Dr. Hermann Schichl

Abstract

In this thesis, we are concerned with Gaussian quadrature formulas and the evaluation of a special family of integrals.
In the first part of the thesis, the theory of quadrature formulas is discussed. Mainly, we focus on the Gaussian quadrature rules. We proof basic properties and introduce methods which transform the moments for integration into the coefficients needed for applying the Gaussian quadrature rules.
In the second part, we want to find Gaussian quadrature formulas for this special family of integrals. This project falls into several steps. First we focus on the moments for integration for the family of integrals. After finding asymptotic formulas for several limit cases by means of Gaussian quadrature rules of first and second order, we construct natural blending functions which have the asymptotic behaviour. Afterwards, we analyse how good these approximations are.
Applying the theoretical part, the approximations for the moments for integration lead to the intended approximate Gaussian quadratures for our family of integrals.
In the end, we present some examples.

Zusammenfassung

Diese Arbeit befasst sich mit Gaußschen Quadraturformeln und der Auswertung einer speziellen Familie von Integralen.
Der erste Teil der Arbeit beinhaltet die Theorie der Quadraturformeln, wobei das Hauptaugenmerk auf den Gaußschen Quadraturformeln liegt. Es werden grundlegende Eigenschaften der Gaußschen Quadraturformeln bewiesen und Algorithmen vorgestellt, aus denen man die Koeffizienten gewinnt, die man für die Auswertung der Gaußschen Quadraturformeln benötigt.
Der zweite Teil der Arbeit behandelt die Konstruktion Gaußscher Quadraturformeln für die spezielle Familie von Integralen. Dieses Projekt zerfällt in mehrere Schritte. Zuerst werden asymptotische Formeln für die Momente (Integration) in einigen Grenzfällen mit Hilfe Gaußscher Quadraturformeln erster und zweiter Ordnung ermittelt. Darauf folgend werden Funktionen, die in den Grenzfällen das korrekte asymptotische Verhalten besitzen, konstruiert. Anschließend wird die Güte dieser Approximationen der Momente untersucht.
Danach wird die Theorie aus dem ersten Teil auf die spezielle Familie von Integralen angewendet, woraus die gewünschten näherungsweisen Gaußschen Quadraturformeln aus den Approximationen der Momente resultieren.
Am Ende werden noch einige Beispiele präsentiert.

Acknowledgement

I want to express my gratitude to my advisor Prof. Hermann Schichl and to Prof. Arnold Neumaier for their helpful hints and comments. Without their constant support this work would not have been possible.

Contents

1 Introduction

In this thesis, we are concerned with quadrature rules for approximating integrals. Our main focus of attention is placed on Gaussian quadrature rules and on the project of finding a reasonably simple closed form for a specific family of integrals, namely

$$I_{d,\omega,t}(f) := \int_0^\infty e^{-x^d+\omega x} x^{t-1} f(x)\, dx$$

with real $d > 1$, $t > 0$, ω, where f is a function of at most polynomial growth depending analytically on $\log x$.
Applications in the physics of phase transitions make it desirable to have quadrature rules calculating this family of integrals. See BUGAEV [1] and FISHER AND FELDERHOF [2].

The structure of the thesis is the following: In Section 2 the concept of quadrature rules is introduced. Sections 3 and 4 deal with geometrically constructed and interpolatory quadrature rules.
In Section 5 we prove the basic properties of Gaussian quadratures and treat two methods for determining the recursion coefficients of orthogonal polynomials needed for the Gaussian quadrature formulas.
Section 6 finds asymptotic formulas for the family of integrals

$$\Gamma_d(\omega,t) := \int_0^\infty e^{-x^d+\omega x} x^{t-1}\, dx$$

(real $d > 1$, $t > 0$) when $\omega \to 0, -\infty, +\infty$ using Gaussian quadrature rules of first and second order.
Section 7 constructs functions which have the same asymptotic behaviour as the one calculated in Section 6. This results in approximations of $\Gamma_d(\omega,t)$.
In Section 8 the quality of the approximations is tested, and Section 9 visualizes the parameters for which the approximations are sufficiently good and for which they are not.
In Section 10, some examples are presented in which the theory from Section 5 is applied to our family of integrals $I_d(\omega,t)(f)$. In particular, the approximations of $\Gamma_d(\omega,t)$ are used to determine approximate Gaussian quadrature rules for $I_d(\omega,t)(f)$.

2 Quadrature rules

2.1 Motivation

As a motivation for defining quadrature formulas, we give an example of an integral that cannot be evaluated analytically.
For more details in this section see ENGELS [3].

Example

Given the problem of computing the arc length of the half ellipse $\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1, y \geq 0$, $b > a$, we are led to solving an elliptic integral of the first kind

$$A = b \int_{-\pi}^{0} (1 - k^2 \sin^2 \phi)^{\frac{1}{2}} d\phi,$$

where $k = \sqrt{1 - \frac{a^2}{b^2}}$.
The antiderivative of the integrand cannot be expressed in terms of elementary functions.
So, it would be advantageous to introduce methods which approximate the solution. In this thesis, we will focus on methods which only use a discrete set of values of the integrand.

We want to find approximations to

$$I(f) := \int_a^b w(x) f(x) dx,$$

where $w(x)$ denotes a so called weight function which can make computations of several integrals easier. We assume $w(x) \geq 0$ for $x \in [a, b]$.

Definition 2.1.1. *A **quadrature formula** is a numerical rule whereby the value of a definite integral is approximated by the use of information about the integrand only at a set of discrete points (where the integrand is defined). Any formula doing this is a quadrature formula.*
*A function $w(x)$ is called a **weight function** if it is a factor of the integrand and if the*

quadrature rule for the approximative calculation of the integral uses only values of $f(x)$ and none of $w(x)$.
We consider quadrature formulas of the form

$$I(f) \approx \sum_{i=1}^{n} A_i f(x_i) =: Q_n(f).$$

*The numbers A_i in a quadrature formula $Q_n(f)$ are called **weights** or **coefficients** and the discrete points x_i are called **nodes** or **points** of the quadrature formula.*
*The **quadrature error** is defined as $E_n(f) := I(f) - Q_n(f)$.*

3 Geometrically constructed quadrature formulas

Suppose we have given a function $f : [a,b] \to \mathbb{R}$ and a subdivision of the interval $[a,b]$ defined through a set of distinct and equidistantly spaced points $a = x_1, \ldots, x_n = b$. Furthermore, we assume that $\omega(w) \equiv 1$.
Then some simple geometric observations lead to the following quadrature formulas.
See ENGELS [3].

3.1 Midpoint rule

As the name already indicates, we approximate the integral by rectangles whose heights are defined as the function value of the midpoint of the particular interval. Thus, we have the following quadrature formula:

$$M_n(f) := h \sum_{i=0}^{n-1} f\left(\frac{x_i + x_{i+1}}{2}\right) = h \sum_{i=0}^{n-1} f\left(a + ih + \frac{h}{2}\right), \text{ with } h = \frac{b-a}{n}.$$

3.2 Trapezoidal rule

This rule is similar to the one above. The only difference is that instead of approximating the subinterval by a rectangle, we use a trapezoid. This leads to the following equation:

$$T_n(f) := \frac{h}{2}\sum_{i=0}^{n-1}(f(x_i)+f(x_{i+1})) = h\left(\frac{f(a)+f(b)}{2}+\sum_{k=1}^{n-1}f(a+kh)\right), \text{ with } h=\frac{b-a}{n}.$$

Observing these two quadrature formulas, one can easily verify that $T_{2n} = \frac{1}{2}(M_n + T_n)$.

Proposition 1. *For the trapezoidal rule, the quadrature error is*

$$E_n(f) = \int_a^b f(x)\,dx - T_n(f) = -\frac{(b-a)f''(\eta)}{12}h^2$$

for some $\eta \in [a,b]$.

Proof. See SCHICHL [4]. □

4 Interpolatory quadratures

Next, we want to introduce a famous quadrature formula using approximations to the integrand $f(x)$ which are analytically integrable as well as interpolants to $f(x)$ in order to integrate the approximations rather than the original $f(x)$. For more details see STOER [5] and ENGELS [3].

Definition 4.0.1. *An interpolant to a function* $f : \mathbb{R} \to \mathbb{R}$ *is a new function* $\tilde{f} : \mathbb{R} \to \mathbb{R}$ *constructed based on a distinct and discrete set of known data points* $(x_0, f(x_0)), \ldots, (x_n, f(x_n))$ *such that* $\tilde{f}(x_i) = f(x_i)$.

4.1 Lagrange interpolation

The interpolatory quadrature formula which we will explain is based on a polynomial interpolation method called the Lagrange interpolation.

Let Π_n be the set of all real polynomials P with $deg(P) \leq n$.

Theorem 1. *Let* $(x_0, f(x_0)), \ldots, (x_n, f(x_n))$ *be* $n+1$ *data points with* $x_i \neq x_j$ *for* $i \neq j$. *Then there exists a unique polynomial* $P \in \Pi_n$ *with* $P(x_i) = f(x_i)$ *for* $i = 0, \ldots, n$.

Proof. Uniqueness: Assume that there are two polynomials $P_1, P_2 \in \Pi_n$ with $P_1(x_i) = P_2(x_i) = f(x_i)$ for $i = 0, 1, \ldots, n$.
Then the polynomial $P := P_1 - P_2 \in \Pi_n$ and has $n+1$ different roots which implies that P has to be the zero polynomial. Hence, $P_1 \equiv P_2$.
Existence: For constructing the interpolating polynomial, let $L_i \in \Pi_n$ for $i = 0, \ldots, n$ satisfying

$$L_i(x_k) = \delta_{i,k} = \begin{cases} 1 & \text{for } i = k \\ 0 & \text{for } i \neq k. \end{cases}$$

Obviously,

$$L_i(x) := \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \qquad i = 0, \ldots, n$$

satisfy these properties.
With these polynomials L_i, $i = 0, \ldots, n$ we can now give the solution of the interpolation problem:

$$P(x) \equiv \sum_{i=0}^{n} f(x_i) L_i(x) = \sum_{i=0}^{n} f(x_i) \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}.$$

□

This way of interpolating f is called **Lagrange interpolation**.

Going back to our problem concerning the solution of $\int_a^b w(x) f(x) dx$, we assume to have distinct points $x_0, \ldots, x_n$ in the interval $[a, b]$.
If we now replace f by a suitable interpolating polynomial $P_n \in \Pi_n$ with $P_n(x_i) = f(x_i)$ for $i = 0, 1, \ldots, n$ which is exactly the Lagrange interpolating polynomial due to Theorem 1, we get a corresponding quadrature formula

$$Q_n(f) = \int_a^b w(x) P_n(x)\, dx = \int_a^b w(x) \sum_{i=0}^{n} f(x_i) L_i(x)\, dx = \sum_{i=0}^{n} f(x_i) \underbrace{\int_a^b w(x) L_i(x)\, dx}_{=: A_i^n} = \sum_{i=0}^{n} A_i^n f(x_i)$$

provided that the integrals $\int_a^b w(x) L_i(x)\, dx$ for $i = 0, \ldots, n$ exist.

If $w(x) \equiv 1$ and the points $x_0, \ldots, x_n$ are spaced equidistantly in the interval $[a, b]$, i.e.

$x_i = a + ih$ for $i = 0, \ldots, n$ and $h := (b-a)/n$, we obtain the so called **Newton-Côtes formulas**:

$$NC_n(f) := \int_a^b P_n(x)dx = \int_a^b \sum_{i=0}^{n} f(x_i)L_i(x)dx = \sum_{i=0}^{n} f(x_i) \underbrace{\int_a^b L_i(x)dx}_{=:A_i^n} = \sum_{i=0}^{n} A_i^n f(x_i).$$

One can show that the approximation error $E_n(f)$ is of the form

$$E_n(f) = \int_a^b f(x)dx - NC_n(f) = -h^{p+1} \cdot K \cdot f^{(p)}(\xi), \quad \xi \in (a,b)$$

assuming that f is sufficiently often differentiable. Of course, p and K depend on n but not on f. See SCHICHL [4].

Remark:

We say that a method is of *order* p, if p is the biggest integer for which the method integrates all polynomials with degree less than p exactly.

For $n = 1, \ldots, 6$, we get:

n	error	name
1	$-h^3 \frac{1}{12} f^{(2)}(\xi)$	trapezoidal rule
2	$-h^5 \frac{1}{90} f^{(4)}(\xi)$	Simpson rule
3	$-h^5 \frac{3}{80} f^{(4)}(\xi)$	3/8 - rule
4	$-h^7 \frac{8}{945} f^{(6)}(\xi)$	Milne rule
5	$-h^7 \frac{275}{12096} f^{(6)}(\xi)$	—
6	$-h^9 \frac{9}{1400} f^{(8)}(\xi)$	Weddle rule

Unfortunately, for $n \geq 7$, additional errors are caused by numerical cancellation because unlike in the cases summarized above ($n \leq 6$), some of the A_i^n are negative. Hence, they are numerically useless.
See SCHICHL [4].

5 Gaussian quadratures

Now we introduce a concept constructing quadrature formulas which we will use and focus on throughout this thesis. See STOER [5].
Let us again consider integrals of the form $I(f) = \int_a^b w(x)f(x)dx$ where we also allow integrals over intervals like $[0,\infty]$ or $[-\infty,\infty]$. The weight function should satisfy the following conditions $(*)$:

- $w(x) \geq 0$ and measurable on $[a,b]$.
- All moments $\mu_k := \int_a^b x^k w(x)dx$, $k = 0,1,\ldots$ exist and are finite.
- For every polynomial $P(x)$ with $\int_a^b w(x)P(x)dx = 0$ and $P(x) \geq 0$ for $x \in [a,b]$ it follows that $P(x) \equiv 0$.

These assumptions are for example satisfied if $w(x)$ is positive and continuous and $[a,b]$ is bounded.

So far, most methods that we have considered concerning quadrature formulas $Q_n = \sum_{i=1}^n A_i f(x_i)$ had the property that the x_i were spaced equidistantly.
Now, we want to get rid of that fact in order to increase the order of the quadrature formula. In particular, we want that the quadrature error E_n vanishes for polynomials with highest possible degree.
We will see that by choosing the A_i and the x_i $(i = 1,\ldots,n)$ properly, the best possible result are the so called Gaussian quadrature formulas which are of order $2n$.

For proving that, we need several elementary properties concerning orthogonal polynomials.

First of all, define the set

$$\overline{\Pi}_j := \{p \in \Pi_j \,|\, p(x) = x^j + a_1 x^{j-1} + \ldots + a_j\}$$

of all **normed real polynomials** of degree j.

Furthermore, we define a scalar product

$$\langle f,g\rangle := \int_a^b w(x)f(x)g(x)dx$$

on the space $L^2_\omega[a,b]$ of all real functions f for which the integral

$$\langle f,f\rangle = \int_a^b w(x)f(x)^2dx$$

exists and is finite.
The functions $f,g \in L^2_\omega[a,b]$ are called **orthogonal** if $\langle f,g\rangle = 0$.
The following theorem shows the existence of orthogonal polynomials with respect to the weight function $w(x)$.

Theorem 2. *For every $j = 0,1,\ldots$ there exist unique polynomials $p_j \in \overline{\Pi}_j$ with*

$$\langle p_i,p_k\rangle = 0 \quad for \quad i,k \in \{0,1,\ldots\}\,,\ i \neq k.$$

These polynomials can be expressed recursively by

a) $p_0(x) \equiv 1$,

b) $p_{i+1}(x) = (x-\delta_{i+1})p_i(x) - \gamma^2_{i+1}p_{i-1}(x) \quad for\ i \geq 0$,

where $p_{-1} :\equiv 0$ and

c) $\delta_{i+1} := \frac{\langle xp_i,p_i\rangle}{\langle p_i,p_i\rangle} \quad for\ i \geq 0$,

d) $\gamma^2_{i+1} := \begin{cases} 0 & for\ i = 0\,, \\ \frac{\langle p_i,p_i\rangle}{\langle p_{i-1},p_{i-1}\rangle} & for\ i \geq 1. \end{cases}$

Proof. With the Gram-Schmidt process, the polynomials can be constructed recursively. Obviously, $p_0(x) \equiv 1$. We assume inductively that there already exist unique polynomials $p_j \in \overline{\Pi}_j$ for all $j \leq i$ with the desired properties.
We have to show that there exists a unique polynomial $p_{i+1} \in \overline{\Pi}_{i+1}$ satisfying

$$\langle p_{i+1},p_j\rangle = 0 \quad \text{for} \quad j \leq i \tag{1}$$

and *b)*. Due to the fact that all the polynomials in $\overline{\Pi}_{i+1}$ are normed, each $p_{i+1} \in \overline{\Pi}_{i+1}$ is of the form

$$p_{i+1}(x) = (x - \delta_{i+1})p_i(x) + c_{i-1}p_{i-1}(x) + c_{i-2}p_{i-2}(x) + \ldots + c_0 p_0(x)$$

with unique coefficients δ_{i+1} and c_k, $k \leq i-1$.
Because of our induction hypothesis, (1) is satisfied if and only if

$$\langle p_{i+1}, p_i \rangle = \langle x p_i, p_i \rangle - \delta_{i+1} \langle p_i, p_i \rangle = 0, \tag{2}$$

$$\langle p_{i+1}, p_{j-1} \rangle = \langle x p_{j-1}, p_i \rangle + c_{j-1} \langle p_{j-1}, p_{j-1} \rangle = 0 \text{ for } j \leq i \tag{3}$$

are satisfied.
These equations are solvable for δ_{i+1} and c_{j-1} because $p_i \not\equiv 0$ and $p_{j-1} \not\equiv 0$ imply $\langle p_i, p_i \rangle > 0$ and $\langle p_{j-1}, p_{j-1} \rangle > 0$ (cf. assumptions on $w(x)$).
From (2) we immediately get c) and if we solve our induction hypothesis

$$p_j(x) = (x - \delta_j)p_{j-1}(x) - \gamma_j^2 p_{j-2}(x)$$

for $x p_{j-1}(x)$, we conclude that

$$\langle x p_{j-1}, p_i \rangle = \langle p_j, p_i \rangle \text{ for } j \leq i.$$

In combination with (3), this gives

$$c_{j-1} = -\frac{\langle p_j, p_i \rangle}{\langle p_{j-1}, p_{j-1} \rangle} = \begin{cases} -\gamma_{i+1}^2 & \text{for } j = i, \\ 0 & \text{for } j < i, \end{cases}$$

which concludes our proof. □

Theorem 3. *Let p_n be the polynomials according to Theorem 2. Then the zeros x_i, $i = 1, \ldots, n$ of p_n are real, simple and lie in the interval (a,b).*

Proof. Let $a < x_1 < \cdots < x_l < b$ be the zeros of p_n in (a,b) at which p_n changes its sign. That means we consider zeros with odd multiplicity. We want to show that $l = n$. Assume that $l < n$. Then the polynomial $q(x) := \prod_{j=1}^{l}(x - x_j) \in \overline{\Pi}_l$ satisfies $\langle p_n, q \rangle = 0$. This is true because every polynomial in $\overline{\Pi}_k$ can be written as a linear combination of

the polynomials p_i, $i \leq k$.
Having a look at the definition of $q(x)$, we see that $p_n(x)q(x)$ does not change its sign on (a,b) which implies that $\langle p_n, q\rangle = \int_a^b w(x)p_n(x)q(x)dx \neq 0$ which is a contradiction to $\langle p_n, q\rangle = 0$.
Hence, $l = n$ and all the zeros are real, simple and are in the interval (a,b). □

Note

In this proof occurs an important observation which we will emphasize explicitly:

$$\langle p, p_n\rangle = 0 \text{ for all } p \in \Pi_{n-1} \tag{4}$$

Before we get to our main result, we need one more theorem.

Theorem 4. *For arbitrary $t_1 < t_2 < \ldots < t_n$, the $n \times n$-matrix*

$$A := \begin{pmatrix} p_0(t_1) & \ldots & p_0(t_n) \\ \vdots & & \vdots \\ p_{n-1}(t_1) & \ldots & p_{n-1}(t_n) \end{pmatrix}$$

is nonsingular.

Proof. If A were singular, there would exist a vector $c^T = (c_0, \ldots, c_{n-1})$, $c \neq 0$ with $c^T A = 0$. In particular, the polynomial

$$q(x) := \sum_{i=0}^{n-1} c_i p_i(x)$$

with $\deg(q) \leq n-1$ would have n different zeros, namely $t_1, \ldots, t_n$. Hence, q(x) would be the zero polynomial. But then the property that the highest coefficient of each p_i is 1 implies $c = 0$ which is a contradiction. □

Remark:

The last theorem is equivalent to the unique solvability of the interpolation problem of finding a function p of the form

$$p(x) = \sum_{i=0}^{n-1} c_i p_i(x)$$

with $p(t_i) = f(x_i)$, $i = 1, \ldots, n$.

Now we have finished all required preparations for our main theorem of this section.

Theorem 5. ***Gaussian quadrature formula of order n***

(a) Let $x_1, \ldots, x_n$ be the zeros of p_n and $\alpha_1, \ldots, \alpha_n$ the solution to the linear system of equations

$$\sum_{i=1}^{n} p_k(x_i)\alpha_i = \begin{cases} \langle p_0, p_0 \rangle & \text{if } k = 0, \\ 0 & \text{if } k = 1, \ldots, n-1. \end{cases} \tag{5}$$

Then $\alpha_i > 0$ for $i = 1, \ldots, n$ and

$$\int_a^b w(x)p(x)dx = \sum_{i=1}^{n} \alpha_i p(x_i) \tag{6}$$

for all polynomials $p \in \Pi_{2n-1}$.

(b) Furthermore, if there exist some real numbers α_i, x_i, $i = 1, \ldots, n$ such that (6) holds for all $p \in \Pi_{2n-1}$, then the x_i are the zeros of p_n and the α_i satisfy (5).

(c) Moreover, there are no real numbers α_i, x_i, $i = 1, \ldots, n$ such that (6) is true for all $p \in \Pi_{2n}$.

Proof. According to Theorem 3, we know that the zeros x_i, $i = 1, \ldots, n$ of p_n are real numbers in (a, b) with $x_i \neq x_j$ for $i, j \in \{1, \ldots, n\}, i \neq j$.

Thus, after Theorem 4 the matrix

$$A := \begin{pmatrix} p_0(x_1) & \dots & p_0(x_n) \\ \vdots & & \vdots \\ p_{n-1}(x_1) & \dots & p_{n-1}(x_n) \end{pmatrix} \tag{7}$$

is nonsingular und therefore the linear system of equations (5) has a unique solution. Now let $p \in \Pi_{2n-1}$ be arbitrary. Then p is of the form

$$p(x) = p_n(x)q(x) + r(x) \tag{8}$$

where $p, r \in \Pi_{n-1}$ and are therefore of the form

$$q(x) = \sum_{k=0}^{n-1} \alpha_k p_k(x), \qquad r(x) = \sum_{k=0}^{n-1} \beta_k p_k(x).$$

Due to (4), it follows that

$$\int_a^b w(x)p(x)dx = \int_a^b w(x)p_n(x)q(x) + \int_a^b w(x)r(x)\cdot 1 = \langle p_n, q\rangle + \langle r, p_0\rangle = \beta_0\langle p_0, p_0\rangle.$$

On the other hand, considering (8), $p_n(x_i) = 0$ and (5), it holds that

$$\sum_{i=1}^{n} \alpha_i p(x_i) = \sum_{i=1}^{n} \alpha_i r(x_i) = \sum_{k=0}^{n-1} \beta_k \left(\sum_{i=1}^{n} \alpha_i p_k(x_i) \right) = \beta_0\langle p_0, p_0\rangle,$$

which means that (6) is valid.

If we now insert the polynomials $\overline{p}_i(x) := \prod_{\substack{j=1 \\ j\neq i}}^{n} (x - x_j)^2 \in \Pi_{2n-2}, \quad i = 1, \dots, n$ in (6), it immediately follows that $\alpha_i > 0$ for $i = 1, \dots, n$. Hence, *(a)* is proven.

Next, we show *(c)*. The assumption that there exist numbers α_i, x_i, $i = 1, \dots, n$ such that (6) holds for all $p \in \Pi_{2n}$ leads with $\overline{p}(x) := \prod_{j=1}^{n} (x - x_j)^2 \in \Pi_{2n}$ to a contradiction because

$$0 < \int_a^b w(x)\overline{p}(x)dx = \sum_{i=1}^{n} \alpha_i \overline{p}(x_i) = 0.$$

For proving *(b)* we insert $p = p_k$, $k = 0,1,\ldots,n-1$ in (6) and get

$$\sum_{i=1}^{n} \alpha_i p_k(x_i) = \int_a^b w(x)p_k(x)dx = \langle p_k, p_0\rangle = \begin{cases} \langle p_o, p_o\rangle & \text{if } k = 0, \\ 0 & \text{if } k = 1,\ldots,n-1, \end{cases}$$

which means that the α_i, $i = 1,\ldots,n$ must satisfy (5).
Now, let us insert $p(x) := p_k(x)p_n(x)$, $k = 0,1,\ldots,n-1$ in (6), then

$$0 = \langle p_k, p_n\rangle = \sum_{i=1}^{n} \alpha_i p_n(x_i)p_k(x_i), \quad k = 0,1,...,n-1.$$

Thus, $Ac = 0$ where $c := (\alpha_1 p_n(x_1),\ldots,\alpha_n p_n(x_n))^T$ and A is the matrix in (7). Due to *(c)*, we get that $x_1 < \cdots < x_n$ which implies that A is nonsingular. Hence, $c = 0$, i.e.

$$\alpha_i p_n(x_i) = 0 \quad \text{for} \quad i = 1,\ldots,n.$$

In the same way as before, we get that $\alpha_i > 0$ which eventually implies that $p_n(x_i) = 0$, $i = 1,\ldots,n$. So, we have also proven *(b)*. □

For some special weight functions $w(x)$, we get the following famous polynomials:

$[a,b]$	$w(x)$	name of the orthogonal polynomials
$[-1,1]$	1	Legendre polynomials
$[-1,1]$	$(1-x^2)^{-1/2}$	$T_n(x)$, Tschebyscheff polynomials
$[0,\infty]$	e^{-x}	$L_n(x)$, Laguerre polynomials
$[-\infty,\infty]$	e^{-x^2}	$H_n(x)$, Hermite polynomials

The following question remains: How to determine the α_i, x_i for $i = 1,\ldots,n$ in general? Let us first consider the problem under the condition of knowing the coefficients δ_i,γ_i, $i = 1,\ldots,n$ appearing in the recursion formula in Theorem 2, *b)*. The determination of these coefficients is in general a numerically difficult problem. We will treat that in Sections 5.2 and 5.3.
The theory of orthogonal polynomials is strongly connected to real tridiagonal matrices. Let us consider the matrix J_n with its principal submatrices J_i:

$$J_n := \begin{pmatrix} \delta_1 & \gamma_2 & & & \\ \gamma_2 & \delta_2 & \cdot & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \gamma_n \\ & & & \gamma_n & \delta_n \end{pmatrix}, \qquad J_i := \begin{pmatrix} \delta_1 & \gamma_2 & & & \\ \gamma_2 & \delta_2 & \cdot & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \gamma_i \\ & & & \gamma_i & \delta_i \end{pmatrix}.$$

We denote the characteristic polynomial of J_i by $p_i(x) := \det(J_i - xI)$, where I is the suitable identity matrix, i.e.

$$p_i(x) := \left| \begin{pmatrix} \delta_1 - x & \gamma_2 & & & \\ \gamma_2 & \delta_2 - x & \cdot & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \gamma_i \\ & & & \gamma_i & \delta_i - x \end{pmatrix} \right|.$$

Then one can immediately verify using Laplace's formula that the following recursion formula holds:

$$\begin{aligned} p_0(x) &:\equiv 1, \\ p_1(x) &= (\delta_1 - x) \cdot 1, \\ p_i(x) &= (\delta_i - x) p_{i-1}(x) - \gamma_i^2 p_{i-2}(x), \quad i = 2, \ldots, n. \end{aligned}$$

From that observation we get the following theorem:

Theorem 6. *The zeros x_i, $i = 1, \ldots, n$ of the nth orthogonal polynomial p_n are the eigenvalues of the tridiagonal matrix J_n.*

These eigenvalues can be calculated with the QR-algorithm. Furthermore, we claim that we can also get the weights α_i from the QR-algorithm. This will confirm the following theorem together with the fact that the QR-algorithm can easily provide the first component of an eigenvector.

Theorem 7. *Let $v^{(k)} = (v_1^{(k)}, \dots, v_n^{(k)})$ be the eigenvector to the eigenvalue x_k of J_n with the scaling that $(v^{(k)})^T v^{(k)} = \langle p_0, p_0 \rangle = \int_a^b w(x)dx$.*
Then it holds that

$$\alpha_k = (v_1^{(k)})^2, \quad k = 1, \dots, n.$$

Proof. Because of the recursion formulas in 2, *b)*, one can prove that the vector of polynomials

$$\tilde{v}(x) := (\rho_0 p_0(x), \rho_1 p_1(x), \dots, \rho_{n-1} p_{n-1}(x))^T$$

with

$$\rho_j := \begin{cases} 1 & \text{for} \quad j = 0 \\ \frac{1}{\gamma_2 \gamma_3 \cdots \gamma_{j+1}} & \text{for} \quad j = 1, \dots, n-1, \end{cases}$$

satisfies

$$(J_n - xI)\tilde{v}(x) = -\rho_{n-1} p_n(x) e_n, \quad e_n := (0, \dots, 0, 1)^T \in \mathbb{R}^n.$$

If we choose for x the n zeros x_i of $p_n(x)$, it follows that the vectors

$$\tilde{v}^{(k)} := \tilde{v}(x_k), \quad k = 1, \dots, n$$

do not vanish because $\rho_0 = 1$ and $p_0 \equiv 1$ imply that $\tilde{v}_1(x) \equiv 1$. Moreover, they are eigenvectors of J_n to the eigenvalues x_k, i.e. $J_n \tilde{v}^{(k)} = x_k \tilde{v}^{(k)}$.
As $\rho_0 \equiv 1$ and $\rho_j \neq 0$, $j = 1, \dots, n-1$, the system of equations (5) from Theorem (5) is equivalent to

$$(\tilde{v}^{(1)}, \dots, \tilde{v}^{(n)})\alpha = \langle p_0, p_0 \rangle \cdot e_1$$

where $\alpha = (\alpha_1, \dots, \alpha_n)^T$ and $e_1 := (1, 0, \dots, 0)^T$.
Eigenvectors of symmetric matrices corresponding to different eigenvalues are orthogonal. Due to $x_i \neq x_k$ for $i \neq k$, a multiplication from the left with $\tilde{v}^{(k)T}$ leads to

$$(\tilde{v}^{(k)T} \tilde{v}^{(k)})\alpha_k = \langle p_0, p_0 \rangle \tilde{v}^{(k)T} e_1 = \langle p_0, p_0 \rangle, \tag{9}$$

because $\tilde{v}^{(k)T} e_1 = \tilde{v}_1^{(k)} = 1$.
Moreover, $\tilde{v}_1^{(k)} = 1$ implies $v_1^{(k)} \tilde{v}^{(k)} = v^{(k)}$ for the normed eigenvector $v^{(k)}$. If we now multiply (9) with $(v_1^{(k)})^2$, we get

$$v^{(k)T} v^{(k)} \alpha_k = (v_1^{(k)})^2 \langle p_0, p_0 \rangle.$$

So, $\alpha_k = (v_1^{(k)})^2$ because $v^{(k)T} v^{(k)} = \langle p_0, p_0 \rangle$.

□

Eventually, we want to give an error estimation of the Gaussian quadrature formulas. The proof can be found in STOER [5].

Theorem 8. *For $f \in C^{2n}[a,b]$, the quadrature error is of the form*

$$\int_a^b w(x) f(x) dx - \sum_{i=1}^{n} \alpha_i f(x_i) = \frac{f^{(2n)}(\xi)}{(2n)!} \langle p_n, p_n \rangle$$

where $\xi \in (a,b)$.

If we compare the Gaussian formulas to other quadrature formulas as for example to the Newton-Côtes formulas, the Gaussian ones give much more accurate results because they have highest order. If we know how to choose n given a desired accuracy ε, then the method is definitely superior to the others. Unfortunately, it is very difficult to find a useful estimate for $f^{(2n)}(\xi)$, $a \leq \xi \leq b$.
So, one could use the Gaussian quadrature formulas for growing n until the approximation is good enough. But there is a big disadvantage. In contrast to other formulas, one cannot reuse the calculated function values from n also for $n+1$ which soon weakens the positive theoretic results.
However, there exist techniques, see for example KRONROD (1965), which widely avoid these disadvantages of the Gaussian method.

5.1 Moments for integration

Coming back to our project of finding quadrature rules calculating

$$I_{d,\omega,t}(f) := \int_0^\infty e^{-x^d + \omega x} x^{t-1} f(x) dx$$

with real $d > 1, t > 0, \omega$, we first consider the so called moments for integration with the weight function $\omega_{a,d,t}(x) := e^{-ax^d} x^{t-1}$ with real $a > 0$, $d > 0$, $t > 0$. They can be found explicitly in terms of the Gamma function and in Section 6 they will turn out to

be very useful.

Definition 5.1.1. HEUSER *[7]*
Let t be a positive real number. Then the Gamma function is defined as the integral

$$\Gamma(t) := \int_0^\infty x^{t-1} e^{-x} dx,$$

which converges absolutely.

Properties:

- $\Gamma(x+1) = x\Gamma(x)$ for all $x > 0$.

- $\Gamma(n+1) = n!$ for $n = 0, 1, 2, ...$

Because of the second property, one says that the Gamma function interpolates the factorials.

The kth moment for integration with the weight function $\omega_{a,d,t}(x)$ is the integral

$$\int_0^\infty e^{-ax^d} x^{t-1} x^k dx.$$

As we have already mentioned, an explicit expression involving the Gamma function can be found.
After substituting $ax^d = y$ which implies that $x = \left(\frac{y}{a}\right)^{\frac{1}{d}}$, we get

$$\begin{aligned}
\int_0^\infty e^{-ax^d} x^{t-1} x^k dx &= \frac{1}{ad} \int_0^\infty e^{-y} x^{t-1+k} \frac{1}{x^{d-1}} dy \\
&= \frac{1}{ad} \int_0^\infty e^{-y} x^{t-d+k} dy \\
&= \frac{1}{ad} \int_0^\infty e^{-y} \left(\frac{y}{a}\right)^{\frac{t-d+k}{d}} dy \\
&= \frac{1}{da} \int_0^\infty e^{-y} \left(\frac{y}{a}\right)^{\frac{t+k}{d}-1} dy \\
&= \frac{1}{da^{\frac{t+k}{d}}} \int_0^\infty e^{-y} y^{\frac{t+k}{d}-1} dy \\
&= \frac{1}{da^{\frac{t+k}{d}}} \Gamma\left(\frac{t+k}{d}\right).
\end{aligned}$$

5.2 Gaussian quadratures from moments for integration

In general, moments for integration have a useful property. They allow one to determine corresponding Gaussian quadrature formulas. We know from Theorem 6 that we need the coefficients δ_i for $i = 1, \ldots, n$ and γ_i for $i = 2, \ldots, n$ in order to determine the zeros x_i, $i = 1, \ldots, n$ of the nth orthogonal polynomial p_n and to get the weights α_i, $i = 1, \ldots, n$.

Thus, we want to transform the moments to the recursion coefficients. This can be done in several ways. We will discuss two methods:

• **Algorithm based on Gautschi.** See GAUTSCHI [8] and GOLUB UND WELSCH [10]. Let $M = (m_{ij})_{1 \le i,j \le n+1}$ be the $(n+1) \times (n+1)$ - Gram matrix for the system $\{x^i \,|\, 0 \le i \le n\}$, i.e.

$$M = \left(\langle x^{i-1}, x^{j-1} \rangle\right)_{1 \le i,j \le n+1} = \left(\int_a^b w(x) x^{i-1} x^{j-1} dx\right)_{1 \le i,j \le n+1}.$$

As M is a positive definite matrix, we can calculate the Cholesky decomposition $M = R^T R$ with

$$r_{ii} = \left(m_{ii} - \sum_{k=1}^{i-1} r_{ki}^2\right)^{\frac{1}{2}} \qquad \text{and} \qquad r_{ij} = \frac{\left(m_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj}\right)}{r_{ii}} \qquad i < j$$

for i and j between 1 and $n+1$. We define the matrix S as

$$S = R^{-1} = \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1,n+1} \\ 0 & s_{22} & \cdots & s_{2,n+1} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & s_{n+1,n+1} \end{pmatrix}.$$

An observation of Mysovskih [11] leads to the fact that the polynomials

$$F_i(x) = s_{1i} + s_{2i}x + \cdots + s_{ii}x^{i-1} \quad i = 1, \ldots, n+1$$

form an orthonormal system. This implies that $\dfrac{F_i(x)}{s_{ii}} = p_{i-1}(x)$ for $i = 1, \ldots, n+1$.
If we now substitute this into the recurrence relation

$$p_{i+1}(x) = (x - \delta_{i+1})p_i(x) - \gamma_{i+1}^2 p_{i-1}(x)$$

and compare the coefficients of x_i and x_{i-1}, we get explicit formulas for δ_i, $i = 1, \ldots, n$ and γ_i, $i = 2, \ldots, n$.

• **Chebyshev algorithm.** See GAUTSCHI [12] and WHEELER [13].
Based on an observation of Chebyshev, the coefficients δ_i, $i = 1, \ldots, n$ and γ_i, $i = 2, \ldots, n$ can be directly retrieved from the following quantities

$$\sigma_{kl} := \int_a^b w(x)p_k(x)x^l dx \quad k = -1, 0, 1, \ldots, n \text{ and } l = 0, \ldots, \min\{2n-k-1, 2n-1\},$$
$$\sigma_{-1,-1} := 1,$$
$$\sigma_{k,-1} := 0 \text{ for } k = 0, \ldots, n.$$

Since the $\{p_i\}_{i=1}^n$ form an orthogonal system with $\deg(p_i) = i$ for $i = 1, \ldots, n$, it follows that $\sigma_{kl} = 0$ for $k > l$. Furthermore, $\sigma_{-1l} = 0$ for $l = 0, \ldots, 2n-1$ because p_{-1} was defined to be the zero polynomial and $\sigma_{0l} = \mu_l$ for $l = 0, \ldots, 2n-1$ where μ_l denotes the lth power moment. From our recurrence relation $p_{k+1}(x) = (x - \delta_{k+1})p_k(x) - \gamma_{k+1}^2 p_{k-1}(x)$, we get that

$$\begin{aligned}\sigma_{k+1,l} = \langle p_{k+1}(x), x^l \rangle &= \langle x p_k(x), x^l \rangle - \delta_{k+1} \langle p_k(x), x^l \rangle - \gamma_{k+1}^2 \langle p_{k-1}(x), x^l \rangle \\ &= \langle p_k(x), x^{l+1} \rangle - \delta_{k+1} \langle p_k(x), x^l \rangle - \gamma_{k+1}^2 \langle p_{k-1}(x), x^l \rangle \\ &= \sigma_{k,l+1} - \delta_{k+1} \sigma_{kl} - \gamma_{k+1}^2 \sigma_{k-1,l}\end{aligned}$$

for $k = 0, \ldots, n-1$ and $l = k+1, \ldots, \min\{2n-k-2, 2n-2\}$.
From that equation we can determine the coefficients δ_k and γ_k for $k = 1, \ldots, n$:

$$\begin{aligned}\sigma_{k+1,k-1} = 0 \text{ for } k = 0, \ldots, n-1 \Rightarrow \gamma_{k+1}^2 &= \frac{\sigma_{kk}}{\sigma_{k-1,k-1}}, \\ \sigma_{k+1,k} = 0 \text{ for } k = 0, \ldots, n-1 \Rightarrow \delta_{k+1} &= \frac{\sigma_{k,k+1}}{\sigma_{kk}} - \frac{\sigma_{k-1,k}}{\sigma_{k-1,k-1}}.\end{aligned}$$

Note:

Even though in this algorithm $\gamma_1^2 = \mu_0$ which may be a contradiction to Theorem 2, this does not make any difference because γ_1^2 is just multiplied by 0.

Explanation:

If we know for example all σ_{ij} for $i = -1, \ldots, k$ and $j = -1, \ldots, \min\{2n-k-1, 2n-1\}$, we can determine γ_{k+1} and δ_{k+1}. Having done that, we can use the equation $\sigma_{k+1,l} = \sigma_{k,l+1} - \delta_{k+1}\sigma_{kl} - \gamma_{k+1}^2 \sigma_{k-1,l}$ to calculate $\sigma_{k+1,l}$ for all $l = 0, \ldots, \min\{2n-k-1, 2n-1\}$.
Then, we can determine γ_{k+2} and δ_{k+2} and so on and so forth.

Thus, if we are given the first $2n$ moments $\mu_0, \ldots, \mu_{2n-1}$, we can determine the first n coefficients $\delta_1, \ldots, \delta_n$ and $\gamma_1, \ldots, \gamma_n$.
The corresponding algorithm described above is called the **Chebyshev algorithm**.

In fact, these two methods are related to each other. While the first method performs a Cholesky factorization of the matrix $M = \left(\langle x^{i-1}, x^{j-1} \rangle\right)_{1 \le i,j \le n+1}$ into the product of

a lower left triangular matrix and its transpose, the second method carries out an LU decomposition into a lower left triangular matrix with ones in the diagonal and an upper right triangular matrix U with

$$\begin{pmatrix} \sigma_{0,0} & \cdots & \sigma_{0,n} \\ \vdots & & \vdots \\ \sigma_{n,0} & \cdots & \sigma_{n,n} \end{pmatrix} = U.$$

Unfortunately, the transformation from power moments to the polynomial recursion coefficients in general is exponentially ill-conditioned. So, the coefficients often contain an increasing error which is then carried over to the abscissas and weights.
The reason for this bad condition is that all functions x^n look very similar and they sample $w(x)$ only near $x = 1$ for large n.
It would be desirable to have polynomials which sample the entire interval over which is integrated.
Sack and Donovan [9] and Gautschi [8] have discovered that polynomials $\tilde{p}_n(x)$ which are orthogonal with respect to a weight function $\tilde{w}(x)$, where $\tilde{w}(x)$ is positive on the same interval as $w(x)$ and is in some sense similar to it, seems to form a good choice.
For doing that, we will introduce the so called modified moments.

5.3 Gaussian quadratures from modified moments

See GAUTSCHI [8] and WHEELER [13].

Definition 5.3.1. *Let $w(x)$ be a weight function satisfying our usual assumptions listed at the beginning of Section 5. Given a set $\{\tilde{p}_k(x)\}_{k=0}^{\infty}$ of polynomials with* $\deg(\tilde{p}_k) = k$, *then*

$$\nu_k = \int_a^b \tilde{p}_k(x) w(x) dx, \qquad k = 0, 1, ...$$

*are called the **modified moments** of $w(x)$.*

For the transformation from modified moments to the recursion coefficients δ_i and γ_i for $i = 1, \ldots, n$, we first need the modified moments with respect to the polynomials $\tilde{p}_k(x)$ for $k = 0, \ldots, 2n$. Without loss of generality we assume that these polynomials are

normed.
As we consider orthogonal polynomials, they satisfy a recurrence relation of the form

$$\tilde{p}_{k+1}(x) = (x - \tilde{\delta}_{k+1})\tilde{p}_k(x) - \tilde{\gamma}_{k+1}^2 \tilde{p}_{k-1}(x) \quad \text{for } k \geq 0$$

and $\tilde{p}_{-1}(x) \equiv 0$.
This relation has to be known for the following methods.

If the modified moments are not explicitly known, the transformation from the power moments to the modified moments is again very ill-conditioned.
Wheeler [13] explained a method in which the exact modified moments can be computed if enough precision is available to represent the power moments exactly.
This can be done recursively. Given n power moments $\mu_0, \ldots, \mu_{n-1}$, we build a $n \times n$-matrix Y with entries $Y_{kl} := \langle \tilde{p}_k, x^l \rangle$ for $0 \leq k, l \leq n-1$. So the first row of Y contains the power moments. Using the recurrence relation $\tilde{p}_{k+1}(x) = (x - \tilde{\delta}_{k+1})\tilde{p}_k(x) - \tilde{\gamma}_{k+1}^2 \tilde{p}_{k-1}(x)$ we get that

$$\begin{aligned} Y_{k+1,l} &= \langle \tilde{p}_{k+1}(x), x^l \rangle = \langle x\tilde{p}_k(x), x^l \rangle - \tilde{\delta}_{k+1} \langle \tilde{p}_k(x), x^l \rangle - \tilde{\gamma}_{k+1}^2 \langle \tilde{p}_{k-1}, x^l \rangle \\ &= \langle \tilde{p}_k(x), x^{l+1} \rangle - \tilde{\delta}_{k+1} \langle \tilde{p}_k(x), x^l \rangle - \tilde{\gamma}_{k+1}^2 \langle \tilde{p}_{k-1}, x^l \rangle \\ &= Y_{k,l+1} - \tilde{\delta}_{k+1} Y_{kl} - \tilde{\gamma}_{k+1}^2 Y_{k-1,l} \end{aligned}$$

for $0 \leq k \leq n-2$ and $0 \leq l \leq n-k-2$ where $Y_{-1,l} = 0$ and $Y_{0,l} = \mu_l$.
So a triangular part of Y can be built containing the desired modified moments in the first column.

Having the modified moments, the transformation can again be done in several ways.

• Algorithm based on Gautschi.
This algorithm is similar to the one described above. Let now $M = (m_{ij})_{1 \leq i,j \leq n+1}$ be the $(n+1) \times (n+1)$ - Gram matrix for the system $\{\tilde{p}_i(x) \,|\, 0 \leq i \leq n\}$, i.e.

$$M = \left(\langle \tilde{p}_i(x), \tilde{p}_j(x) \rangle \right)_{0 \leq i,j \leq n} = \left(\int_a^b w(x) \tilde{p}_i(x) \tilde{p}_j(x) dx \right)_{0 \leq i,j \leq n}.$$

As M is again a positive definite matrix, we can just proceed like we did above and get the Matrix

$$S = R^{-1} = \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1,n+1} \\ 0 & s_{22} & \cdots & s_{2,n+1} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & s_{n+1,n+1} \end{pmatrix}.$$

Another observation of Mysovskih [11] leads to the fact that the polynomials

$$F_i(x) = s_{1i} + s_{2i}\tilde{p}_1(x) + \cdots + s_{ii}\tilde{p}_i(x) \quad i = 1, \ldots, n+1$$

also form an orthonormal system which again implies that $\frac{F_i(x)}{s_{ii}} = p_{i-1}(x)$ for $i = 1, \ldots, n$.

As above, we substitute this into the recurrence relation $p_{i+1}(x) = (x - \delta_{i+1})p_i(x) - \gamma_{i+1}^2 p_{i-1}(x)$ and make use of the second relation $\tilde{p}_{i+1}(x) = (x - \tilde{\delta}_{i+1})\tilde{p}_k(x) - \tilde{\gamma}_{i+1}^2 \tilde{p}_{i-1}(x)$. The explicit formulas for δ_i, $i = 1, \ldots, n$ and γ_i, $i = 2, \ldots, n$ can then again be found by a comparison of coefficients.

•The modified Chebyshev algorithm

Similar to Section 5.2, the coefficients δ_i, $i = 1, \ldots, n$ and γ_i, $i = 2, \ldots, n$ can be directly retrieved from the following quantities

$\sigma_{kl} := \int_a^b w(x)p_k(x)\tilde{p}_l(x)dx \quad k = -1, 0, 1, \ldots, n$ and $l = 0, \ldots, \min\{2n-k-1, 2n-1\}$,

$\sigma_{-1,-1} :\equiv 1$,

$\sigma_{k,-1} := 0$ for $k = 0, \ldots, n$.

As before, $\sigma_{kl} = 0$ for $k > l$, $\sigma_{-1l} = 0$ for $l = 0, \ldots, 2n-1$ and $\sigma_{0l} = \nu_l$ for $l = 0, \ldots, 2n-1$ where ν_l denotes the lth modified power moment. From our recurrence relations $p_{k+1}(x) = (x - \delta_{k+1})p_k(x) - \gamma_{k+1}^2 p_{k-1}(x)$ and $\tilde{p}_{k+1}(x) = (x - \tilde{\delta}_{k+1})\tilde{p}_k(x) - \tilde{\gamma}_{k+1}^2 \tilde{p}_{k-1}(x)$, we get that

$$\begin{aligned}\sigma_{k+1,l} &= \langle p_{k+1}(x), \tilde{p}_l(x)\rangle = \langle x p_k(x), \tilde{p}_l(x)\rangle - \delta_{k+1}\langle p_k(x), \tilde{p}_l\rangle - \gamma_{k+1}^2\langle p_{k-1}(x), \tilde{p}_l\rangle \\ &= \langle p_k(x), x\tilde{p}_l(x) - \tilde{\delta}_{l+1}\tilde{p}_l(x) - \tilde{\gamma}_{l+1}^2\tilde{p}_{l-1}(x)\rangle + \tilde{\delta}_{l+1}\langle p_k(x), \tilde{p}_l(x)\rangle \\ &\quad + \tilde{\gamma}_{l+1}^2\langle p_k(x), \tilde{p}_{l-1}\rangle - \delta_{k+1}\langle p_k(x), \tilde{p}_l(x)\rangle - \gamma_{k+1}^2\langle p_{k-1}(x), \tilde{p}_l(x)\rangle \\ &= \sigma_{k,l+1} - (\delta_{k+1} - \tilde{\delta}_{l+1})\sigma_{kl} - \gamma_{k+1}^2\sigma_{k-1,l} + \tilde{\gamma}_{l+1}^2\sigma_{k,l-1}\end{aligned}$$

for $k = 0, \ldots, n-1$ and $l = k+1, \ldots, \min\{2n-k-2, 2n-2\}$.
From that equation we can determine the coefficients δ_k and γ_k for $k = 1, \ldots, n$:

$$\begin{aligned}\sigma_{k+1,k-1} = 0 \text{ for } k = 0, \ldots, n-1 &\Rightarrow \gamma_{k+1}^2 = \frac{\sigma_{kk}}{\sigma_{k-1,k-1}} \\ \sigma_{k+1,k} = 0 \text{ for } k = 0, \ldots, n-1 &\Rightarrow \delta_{k+1} = \tilde{\delta}_{k+1} + \frac{\sigma_{k,k+1}}{\sigma_{kk}} - \frac{\sigma_{k-1,k}}{\sigma_{k-1,k-1}}.\end{aligned}$$

Explanation:

If we know for example all σ_{ij} for $i = -1, \ldots, k$ and $j = -1, \ldots, \min\{2n-k-1, 2n-1\}$, we can determine γ_{k+1} and δ_{k+1}. Having done that, we can use the equation $\sigma_{k+1,l} = \sigma_{k,l+1} - (\delta_{k+1} - \tilde{\delta}_{l+1})\sigma_{kl} - \gamma_{k+1}^2\sigma_{k-1,l} + \tilde{\gamma}_{l+1}^2\sigma_{k,l-1}$ to calculate $\sigma_{k+1,l}$ for all $l = 0, \ldots, \min\{2n-k-1, 2n-1\}$.
Then, we can determine γ_{k+2} and δ_{k+2} and so on and so forth.

Thus, if we know the first $2n$ modified moments $\nu_0, \ldots, \nu_{2n-1}$, we get the first n coefficients $\delta_1, \ldots, \delta_n$ and $\gamma_1, \ldots, \gamma_n$ using this so called **modified Chebyshev algorithm**.

Concerning modified moments for the weight function $\omega_{a,d,t}(x) := e^{-ax^d}x^{t-1}$ with real $a > 0$, $d > 0$, $t > 0$ from the beginning of Section 5.1, the Laguerre polynomials seem to form a suitable choice for the orthogonal polynomials which we use to calculate modified moments. One reason for that is that the Laguerre polynomials $L_n(x)$, $n \geq 0$ are orthogonal with respect to the inner product

$$\int_0^\infty e^{-x}L_n(x)L_m(x)dx.$$

Another reason is that the integral boundaries are the same as in our example and so

the weight function $w(x) = e^{-x}$ is positive on the same interval as the weight function $\omega_{a,d,t}(x)$.
Another possible choice would be the generalized Laguerre polynomials $L_n^{(\alpha)}(x)$, $n \geq 0$ for arbitrary real α which are orthogonal with respect to the inner product

$$\int_0^\infty x^\alpha e^{-x} L_n^{(\alpha)}(x) L_m^{(\alpha)}(x).$$

However, we do not aim to calculate Gaussian quadrature formulas for integrals with the weight function $\omega_{a,d,t}(x)$ here, but we will come back to the concept of determining the coefficients $\delta_1, \ldots, \delta_n$ and $\gamma_1, \ldots, \gamma_n$ from moments in Section 10.

6 Asymptotic formulas

In this section, we want to focus on the integral

$$\Gamma_d(\omega, t) := \int_0^\infty e^{-x^d + \omega x} x^{t-1}\, dx$$

(real $d > 1$, $t > 0$) when $\omega \to 0, -\infty, +\infty$. Getting one step closer to our goal, we want to find asymptotic formulas for this integral in the three limit cases of ω by applying Gaussian quadratures.
We use the weight functions

$$w(x) = \begin{cases} e^{-x^d} x^{t-1} & \text{for } \omega \to 0 \\ e^{\omega x} x^{t-1} & \text{for } \omega \to -\infty \\ e^{-z^2} & \text{for } \omega \to \infty. \end{cases}$$

The z in the last case results from the substitution $z = x^{\frac{d}{2}} - x_0^{\frac{d}{2}}$, where x_0 is the maximizer of the exponent $-x^d + \omega x$.

In each of the three limit cases, we will use Gaussian quadrature formulas of first and second order to see how the integral behaves asymptotically for $\omega \to 0, -\infty, \infty$.
In each case, we reduce $\Gamma_d(\omega, t)$ to some integral $\int_0^\infty e^{-\tilde{a} x^{\tilde{d}}} x^{\tilde{t}-1}\, dx$ with real $\tilde{a} > 0$, $\tilde{d} > 0$, $\tilde{t} > 0$ which we have already solved explicitly in Section 5.2.

Case 1: $\omega \to 0$

In this case, our weight function is $w(x) = e^{-x^d} x^{t-1}$ and the function, to which the Gaussian quadrature formulas are applied, is $f(x) = e^{\omega x}$.

• $n = 1$ (First order Gaussian formula)

$p_0(x) \equiv 1$

$p_1(x) = (x - \delta_1)p_0$ where

$$\delta_1 = \frac{\langle xp_0, p_0\rangle}{\langle p_0, p_0\rangle} = \frac{\int_o^\infty e^{-x^d} x^t\, dx}{\int_0^\infty e^{-x^d} x^{t-1}\, dx} = \frac{\frac{1}{d}\Gamma(\frac{t+1}{d})}{\frac{1}{d}\Gamma(\frac{t}{d})} = \frac{\Gamma(\frac{t+1}{d})}{\Gamma(\frac{t}{d})}.$$

So, $p_1(x) = x - \delta_1$ with the one zero $x_1 = \delta_1$.

Since the corresponding weights $\alpha_1, \ldots, \alpha_n$ for a Gaussian formula of order n can be calculated by solving the linear system of equations

$$\sum_{i=1}^{n} p_k(x_i)\alpha_i = \begin{cases} \langle p_0, p_0\rangle & \text{if } k = 0, \\ 0 & \text{if } k = 1, \ldots, n-1, \end{cases} \tag{10}$$

we get that

$$\alpha_1 = \langle p_0, p_0\rangle = \frac{1}{d}\Gamma(\frac{t}{d}).$$

Because we know that the first moment for integration is $m_0 = \langle p_0, p_0\rangle$, we get that $\alpha_1 = m_0$.

So the first order Gaussian formula is

$$G_1(f) := \alpha_1 f(x_1) = \alpha_1 e^{\omega\delta_1}$$

which converges to the first moment $\alpha_1 = m_0 = \frac{1}{d}\Gamma(\frac{t}{d})$ as $\omega \to 0$.

• $n = 2$ (Second order Gaussian formula)

$p_0(x) \equiv 1$

$p_1(x) = x - \frac{\Gamma(\frac{t+1}{d})}{\Gamma(\frac{t}{d})} \qquad (\delta_1 = \frac{\Gamma(\frac{t+1}{d})}{\Gamma(\frac{t}{d})})$

$p_2(x) = (x - \delta_2)p_1 - \gamma_2^2 p_0$ where

$$\delta_2 = \frac{\langle xp_1, p_1\rangle}{\langle p_1, p_1\rangle} = \frac{\Gamma\left(\frac{t+3}{d}\right) - 2\delta_1\Gamma\left(\frac{t+2}{d}\right) + \delta_1^2\Gamma\left(\frac{t+1}{d}\right)}{\Gamma\left(\frac{t+2}{d}\right) - 2\delta_1\Gamma\left(\frac{t+1}{d}\right) + \delta_1^2\Gamma\left(\frac{t}{d}\right)} = \frac{\Gamma\left(\frac{t+3}{d}\right) - 2\delta_1\Gamma\left(\frac{t+2}{d}\right) + \delta_1^2\Gamma\left(\frac{t+1}{d}\right)}{\Gamma\left(\frac{t+2}{d}\right) - \delta_1\Gamma\left(\frac{t+1}{d}\right)}$$

and

$$\gamma_2^2 = \frac{\langle p_1, p_1 \rangle}{\langle p_0, p_0 \rangle} = \frac{\Gamma\left(\frac{t+2}{d}\right) - 2\delta_1\Gamma\left(\frac{t+1}{d}\right) + \delta_1^2\Gamma\left(\frac{t}{d}\right)}{\Gamma\left(\frac{t}{d}\right)} = \frac{\Gamma\left(\frac{t+2}{d}\right)}{\Gamma\left(\frac{t}{d}\right)} - \delta_1^2.$$

So, $p_2(x) = (x-\delta_2)(x-\delta_1) - \gamma_2^2$ with zeros

$$x_1 = \frac{1}{2}\left(\delta_1 + \delta_2 + \sqrt{(\delta_1 - \delta_2)^2 + 4\gamma_2^2}\right)$$

and

$$x_2 = \frac{\delta_1\delta_2 - \gamma_2^2}{x_1}.$$

If we define

$$q := \sqrt{(\delta_1 - \delta_2)^2 + 4\gamma_2^2},$$

we get that

$$x_1 - x_2 = q.$$

According to equation (10), α_1 and α_2 solve the following system of equations:

$$\begin{aligned} p_0(x_1)\alpha_1 + p_0(x_2)\alpha_2 &= \tfrac{1}{d}\Gamma\left(\tfrac{t}{d}\right) \\ p_1(x_1)\alpha_1 + p_1(x_2)\alpha_2 &= 0. \end{aligned}$$

Defining $r := \frac{\delta_1 - \delta_2}{q}$, we get that

$$\alpha_1 = \frac{\Gamma(\frac{t}{d})}{2d}(1+r)$$

and

$$\alpha_2 = \frac{\Gamma(\frac{t}{d})}{2d}(1-r).$$

Now we can look at the second order Gaussian formula

$$G_2(f) = \alpha_1 f(x_1) + \alpha_2 f(x_2) = \frac{\Gamma(\frac{t}{d})}{2d}(1+r)\,e^{\omega x_1} + \frac{\Gamma(\frac{t}{d})}{2d}(1-r)\,e^{\omega x_2}$$

which converges for $\omega \to 0$ again to

$$\frac{\Gamma(\frac{t}{d})}{2d}(1+r)e^0 + \frac{\Gamma(\frac{t}{d})}{2d}(1-r)e^0 = \frac{1}{d}\Gamma\left(\frac{t}{d}\right) = m_0.$$

Case 2: $\omega \to -\infty$
In Case 2, we use the weight function $\omega(x) = e^{\omega x}x^{t-1}$ and apply the Gaussian formulas to the function $f(x) = e^{-x^d}$.

• $n = 1$ (First order Gaussian formula)
$p_0(x) \equiv 1$
$p_1(x) = (x - \delta_1)p_0$ where

$$\delta_1 = \frac{\langle xp_0, p_0\rangle}{\langle p_0, p_0\rangle} = \frac{\int_o^\infty e^{\omega x}x^t\,dx}{\int_0^\infty e^{\omega x}x^{t-1}\,dx} = \frac{\frac{1}{(-\omega)^{t+1}}\Gamma(t+1)}{\frac{1}{(-\omega)^t}\Gamma(t)} = \frac{\Gamma(t+1)}{-\omega\Gamma(t)} = \frac{t}{-\omega}.$$

Note that $\frac{t}{-\omega} \to 0$ for $\omega \to -\infty$.
As one can easily check solving the system of equations (10) that $\alpha_1 = \frac{1}{(-\omega)^t}\Gamma(t)$, we get the first order Gaussian formula

$$G_1(f) := \alpha_1 f(x_1) = \frac{\Gamma(t)\,e^{-\left(\frac{t}{-\omega}\right)^d}}{(-\omega)^t}$$

which converges to 0 as $\omega \to -\infty$.

• $n = 2$ (Second order Gaussian formula)
$p_0(x) \equiv 1$
$p_1(x) = x - \frac{t}{-\omega} \qquad (\delta_1 = \frac{t}{-\omega})$
$p_2(x) = (x - \delta_2)p_1 - \gamma_2^2 p_0$ where

$$\delta_2 = \frac{t'}{-\omega}$$

with $t' = \frac{\Gamma(t+3) - 2t\Gamma(t+2) + t^2\Gamma(t+1)}{\Gamma(t+2) - 2t\Gamma(t+1) + t^2\Gamma(t)}$ and

$$\gamma_2^2 = \frac{t''}{(-\omega)^2}$$

with $t'' = \frac{\Gamma(t+2)-2t\Gamma(t+1)+t^2\Gamma(t)}{\Gamma(t)}$.
We note that $\delta_1 \to 0$, $\delta_2 \to 0$ and $\gamma_2^2 \to 0$ for $\omega \to -\infty$.
The zeros of the polynomial $p_2(x) = (x-\delta_2)(x-\delta_1) - \gamma_2^2$ are as before

$$x_1 = \frac{1}{2}\left(\delta_1 + \delta_2 + \sqrt{(\delta_1-\delta_2)^2 + 4\gamma_2^2}\right) = \frac{y_1}{-\omega}$$

with $y_1 = \frac{1}{2}\left(t + t' + \sqrt{(t-t')^2 + 4t''}\right)$ and

$$x_2 = \frac{\delta_1\delta_2 - \gamma_2^2}{x_1} = \frac{y_2}{-\omega}$$

with $y_2 = \frac{tt'-t''}{y_1}$.
Solving the system of equation

$$\begin{aligned} p_0(x_1)\alpha_1 + p_0(x_2)\alpha_2 &= \tfrac{1}{(-\omega)^t}\Gamma(t) \\ p_1(x_1)\alpha_1 + p_1(x_2)\alpha_2 &= 0. \end{aligned}$$

gives

$$\alpha_1 = \frac{\Gamma(t+1) + \omega x_2\Gamma(t)}{(-\omega)^{t+1}(x_1-x_2)} = \frac{\mu_1}{(-\omega)^t}$$

with $\mu_1 = \frac{\Gamma(t+1)-y_2\Gamma(t)}{y_1-y_2}$ and

$$\alpha_2 = \frac{(-\omega)x_1\Gamma(t) - \Gamma(t+1)}{(-\omega)^{t+1}(x_1-x_2)} = \frac{\mu_2}{(-\omega)^t}$$

with $\mu_2 = \frac{y_1\Gamma(t)-\Gamma(t+1)}{y_1-y_2}$. Both α_1 and α_2 converge to 0 as $\omega \to \infty$.
So we have that

$$G_2(f) = \alpha_1 f(x_1) + \alpha_2 f(x_2) = \frac{\mu_1}{(-\omega)^t} e^{-\left(\frac{y_1}{-\omega}\right)^d} + \frac{\mu_2}{(-\omega)^t} e^{-\left(\frac{y_2}{-\omega}\right)^d}$$

with the obvious property that $G_2(f) \to 0$ for $\omega \to -\infty$.

Case 3: $\omega \to +\infty$

In the last case, we use the weight function $w(x) = e^{-z^2}$ after performing the substitution $z = x^{\frac{d}{2}} - x_0^{\frac{d}{2}}$, where x_0 is the maximizer of the exponent $x^d + \omega x$ as explained

above. Then we apply the Gaussian formulas to the function $f(z)$ which results from the substitution.

The maximizer of the exponent $x^d + \omega x$ can easily be found by differentiating with respect to x and setting the resulting term to zero. Thus, $x_0 = (\frac{\omega}{d})^{\frac{1}{d-1}}$ which has the property that $x_0 \to \infty$ as $\omega \to \infty$.
After the change of variables outlined above, we should have that

$$\int_{-c}^{\infty} e^{-z^2} f(z)\, dz = \int_0^{\infty} e^{-x^d+\omega x} x^{t-1}\, dx,$$

where $c := x_0^{\frac{d}{2}} = \left(\frac{\omega}{d}\right)^{\frac{d}{2d-2}}$.
In order to find our function $f(z)$, we use the fact that $dz = z'(x)dx$ and get

$$f(z) = \frac{e^{z^2-x^d+\omega x} x^{t-1}}{z'(x)} = \frac{e^{z^2-x^d+\omega x}}{\frac{d}{2} x^{\frac{d}{2}-1}}.$$

If we now use that $z = x^{\frac{d}{2}} - x_0^{\frac{d}{2}}$, we get that

$$f(z) = \frac{2}{d}(z+c)^{\frac{2t}{d}-1} e^{-2cz-c^2+\omega(z+c)^{\frac{2}{d}}},$$

where $f(z) \to \infty$ for $\omega \to \infty$.

Now, we want to get rid of the lower integral boundary $-c$ because, as we have already mentioned, we want the integral to be of the form $\int_0^{\infty} e^{-\tilde{a}x^{\tilde{d}}} x^{\tilde{t}-1}\, dx$ with real $\tilde{a} > 0$, $\tilde{d} > 0$, $\tilde{t} > 0$.
For achieving this, we start by simply setting the lower integral boundary to $-\infty$. This is a reasonable step because as ω gets bigger, the error tends to zero.
In particular, $c \to \infty$ if $\omega \to \infty$.

If we have a closer look at our integral

$$\int_{-\infty}^{\infty} e^{-z^2} f(z)\, dz,$$

it is not very difficult to see that

$$\int_{-\infty}^{\infty} e^{-z^2} f(z)\,dz = \int_{-\infty}^{\infty} e^{-z^2} \frac{f(z)+f(-z)}{2}\,dz.$$

We only have to make the substitution $z \to -z$ which results in

$$\int_{-\infty}^{\infty} e^{-z^2} f(z)\,dz = \int_{\infty}^{-\infty} -e^{-z^2} f(-z)\,dz = \int_{-\infty}^{\infty} e^{-z^2} f(-z)\,dz.$$

But now we are almost finished because not only e^{-z^2} but also $\frac{f(z)+f(-z)}{2}$ is an even analytic function. So, we get that

$$\int_{-\infty}^{\infty} e^{-z^2} \frac{f(z)+f(-z)}{2}\,dz = \int_{0}^{\infty} e^{-z^2} \left(f(z)+f(-z)\right)\,dz = \int_{0}^{\infty} e^{-z^2} g(z)\,dz,$$

where $g(z) := f(z)+f(-z)$.

If we now approximate $g(z)$ by a second order Taylor expansion $\tilde{g}(z)$ around $z = 0$, we get that

$$g(z) = \tilde{g}(z) + O(z^4) = g_0 e^{r_1}(1+q_1 z^2) + O(z^4)$$

with

$$g_0 = \frac{4}{d}\left(\frac{\omega}{d}\right)^{\frac{d-2t}{2-2d}},$$

$$r_1 = (d-1)\left(\frac{\omega}{d}\right)^{\frac{d}{d-1}},$$

and

$$q_1 = \frac{1}{d^2}\left(2d - d^2 + (d-2t)(d-t)\left(\frac{\omega}{d}\right)^{\frac{d}{1-d}}\right).$$

We will use $\tilde{g}(z)$ in the Gaussian quadrature formulas.
Remark: $g(z) \to \infty$ and $\tilde{g}(z) \to \infty$ for $\omega \to \infty$.

Now, we can apply the Gaussian quadrature formulas.

$p_0(z) \equiv 1$

$p_1(z) = (z - \delta_1)p_0$ where

$$\delta_1 = \frac{\langle zp_0, p_0\rangle}{\langle p_0, p_0\rangle} = \frac{\int_0^\infty e^{-z^2} z\,dz}{\int_0^\infty e^{-z^2}\,dz} = \frac{\frac{1}{2}\Gamma(1)}{\frac{1}{2}\Gamma(\frac{1}{2})} = \frac{1}{\sqrt{\pi}}.$$

As $z_1 = \frac{1}{\sqrt{\pi}}$ and $\alpha_1 = \langle p_0, p_0\rangle = \frac{1}{2}\Gamma(\frac{1}{2}) = \frac{\sqrt{\pi}}{2}$, we can apply the first order Gaussian formula

$$G_1(g) \approx \alpha_1 \tilde{g}(z_1) = \frac{\sqrt{\pi}}{2}\tilde{g}\left(\frac{1}{\sqrt{\pi}}\right) = \frac{\pi + q_1}{2\sqrt{\pi}} g_0 e^{r_1}.$$

For the second order Gaussian formula, we also need
$p_2(z) = (z - \delta_2)p_1 - \gamma_2^2 p_0$
where

$$\delta_2 = \frac{\langle zp_1, p_1\rangle}{\langle p_1, p_1\rangle} = \frac{2}{(\pi - 2)\sqrt{\pi}} \approx 0.9884$$

and

$$\gamma_2^2 = \frac{\langle p_1, p_1\rangle}{\langle p_0, p_0\rangle} = \frac{\pi - 2}{2\pi} \approx 0.1817.$$

So, we have $p_2(z) = z^2 - \frac{2\sqrt{\pi}}{2\pi - 4}z + \frac{4-\pi}{2\pi-4}$ with zeros

$$z_1 = \frac{\sqrt{\pi} - s}{2\pi - 4} \approx 0.3002$$

and

$$z_2 = \frac{\sqrt{\pi} + s}{2\pi - 4} \approx 1.2524,$$

where $s = \sqrt{16 - 11\pi + 2\pi^2}$.
Solving

$$\begin{aligned} p_0(z_1)\alpha_1 + p_0(z_2)\alpha_2 &= \tfrac{1}{2}\Gamma(\tfrac{1}{2}) \\ p_1(z_1)\alpha_1 + p_1(z_2)\alpha_2 &= 0 \end{aligned}$$

leads to

$$\alpha_1 = \alpha - \frac{\beta}{s} \approx 0.6405$$

and

$$\alpha_2 = \alpha + \frac{\beta}{s} \approx 0.2457,$$

where

$$\alpha = \frac{\sqrt{\pi}}{4}$$

and

$$\beta = \frac{\pi}{4} - 1.$$

With these results we get the second order Gaussian formula

$$\begin{aligned} G_2(g) &\approx \alpha_1 \tilde{g}(z_1) + \alpha_2 \tilde{g}(z_2) \\ &= \left(\alpha - \frac{\beta}{s}\right) g_0 e^{r_1} \left(1 + q_1 \left(\frac{\sqrt{\pi} - s}{2\pi - 4}\right)^2\right) + \left(\alpha + \frac{\beta}{s}\right) g_0 e^{r_1} \left(1 + q_1 \left(\frac{\sqrt{\pi} + s}{2\pi - 4}\right)^2\right) \\ &= \alpha g_0 e^{r_1} (2 + q_1). \end{aligned}$$

The fact that $g(z) \to \infty$ for $\omega \to \infty$ naturally implies that $G_1(g) \to \infty$ and $G_2(g) \to \infty$ for $\omega \to \infty$.

7 Blending functions

In this step, we want to construct natural blending functions that have the correct limits as determined in Section 6. This results in approximations $\tilde{\Gamma}_d(\omega, t)$ for $\Gamma_d(\omega, t)$.

Remark: We use here several abbreviations for terms which were defined in Section 6.

7.1 1st order

First, we want to find a function $\tilde{\Gamma}_d(\omega, t)$ which has the asymptotic behaviour determined by the first order Gaussian formula.
For this, we make the ansatz

$$\tilde{\Gamma}_d(\omega,t) = r(\omega)e^{s(\omega)}.$$

In the previous step, the first order Gaussian formula led to the limits

$$\begin{aligned}
\Gamma_d(\omega,t) &\approx \frac{1}{d}\Gamma(\frac{t}{d})e^{\omega\frac{\Gamma(\frac{t+1}{d})}{\Gamma(\frac{t}{d})}} \quad \text{for } \omega \to 0,\\
\Gamma_d(\omega,t) &\approx \frac{\Gamma(t)e^{-(\frac{t}{-\omega})^d}}{(-\omega)^t} \quad \text{for } \omega \to -\infty,\\
\Gamma_d(\omega,t) &\approx \frac{\pi+q_1}{2\sqrt{\pi}}g_0 e^{r_1} \quad \text{for } \omega \to \infty.
\end{aligned}$$

So, the functions $r(\omega)$ and $s(\omega)$ should satisfy

$$r(\omega) \approx \begin{cases} \frac{\Gamma(t)}{(-\omega)^t} & \text{for } \omega \to -\infty \\ \frac{1}{d}\Gamma(\frac{t}{d}) & \text{for } \omega \to 0 \\ \frac{\pi+q_1}{2\sqrt{\pi}}g_0 & \text{for } \omega \to \infty, \end{cases}$$

$$s(\omega) \approx \begin{cases} -(\frac{t}{-\omega})^d & \text{for } \omega \to -\infty \\ \omega\frac{\Gamma(\frac{t+1}{d})}{\Gamma(\frac{t}{d})} & \text{for } \omega \to 0 \\ r_1 & \text{for } \omega \to \infty. \end{cases}$$

We set

$$r(\omega) = \begin{cases} \frac{1}{\frac{d}{\Gamma(\frac{t}{d})}+\frac{(-\omega)^t}{\Gamma(t)}} & \text{if } \omega \le 0 \\ \frac{\Gamma(\frac{t}{d})+\lambda\frac{\pi+q_1}{2\sqrt{\pi}}g_0\omega}{d+\lambda\omega} & \text{if } \omega \ge 0 \end{cases}$$

and

$$s(\omega) = \begin{cases} \frac{\omega\Gamma(\frac{t+1}{d})-\lambda t^d\omega^2}{\Gamma(\frac{t}{d})+\lambda(-\omega)^{d+2}} & \text{if } \omega \le 0 \\ \frac{\omega\Gamma(\frac{t+1}{d})+\lambda r_1\omega^2}{\Gamma(\frac{t}{d})+\lambda\omega^2} & \text{if } \omega \ge 0, \end{cases}$$

where $\lambda > 0$ can be chosen arbitrary as a function of t and d.

7.2 2nd order

Our next step is to find a function $\widetilde{\Gamma}_d(\omega,t)$ which has the limits determined by the second order Gaussian formula.
For this, we make the ansatz

$$\widetilde{\Gamma}_d(\omega,t) = \widetilde{\Gamma}_{d,1}(\omega,t) + \widetilde{\Gamma}_{d,2}(\omega,t)$$

where $\Gamma_{d,1}$ and $\Gamma_{d,2}$ are of the same form as above:

$$\Gamma_{d,1}(\omega,t) = r_1(\omega)e^{s_1(\omega)}$$
$$\Gamma_{d,2}(\omega,t) = r_2(\omega)e^{s_2(\omega)}$$

From the second order Gaussian formula, we derived the following asymptotic behaviour.

$$\begin{aligned}
\Gamma_d(\omega,t) &\approx \frac{\Gamma(\frac{t}{d})}{2d}(1+r)e^{\omega x_1} + \frac{\Gamma(\frac{t}{d})}{2d}(1-r)e^{\omega x_2} \quad \text{for } \omega \to 0,\\
\Gamma_d(\omega,t) &\approx \frac{\mu_1}{(-\omega)^t}e^{-(\frac{y_1}{-\omega})^d} + \frac{\mu_2}{(-\omega)^t}e^{-(\frac{y_2}{-\omega})^d} \quad \text{for } \omega \to -\infty,\\
\Gamma_d(\omega,t) &\approx \alpha g_0 e^{r_1}(2+q_1) \quad \text{for } \omega \to \infty.
\end{aligned}$$

From these limits we can conclude that $r_i(\omega)$ and $s_i(\omega)$ for $i = 1,2$ should satisfy

$$r_1(\omega) \approx \begin{cases} \frac{\mu_1}{(-\omega)^t} & \text{for } \omega \to -\infty \\ \frac{\Gamma(\frac{t}{d})}{2d}(1+r) & \text{for } \omega \to 0 \\ 2\alpha g_0 & \text{for } \omega \to \infty, \end{cases}$$

$$r_2(\omega) \approx \begin{cases} \frac{\mu_2}{(-\omega)^t} & \text{for } \omega \to -\infty \\ \frac{\Gamma(\frac{t}{d})}{2d}(1-r) & \text{for } \omega \to 0 \\ \alpha q_1 g_0 & \text{for } \omega \to \infty, \end{cases}$$

$$s_1(\omega) \approx \begin{cases} -(\frac{y_1}{-\omega})^d & \text{for } \omega \to -\infty \\ \omega x_1 & \text{for } \omega \to 0 \\ r_1 & \text{for } \omega \to \infty, \end{cases}$$

$$s_2(\omega) \approx \begin{cases} -(\frac{y_2}{-\omega})^d & \text{for } \omega \to -\infty \\ \omega x_2 & \text{for } \omega \to 0 \\ r_1 & \text{for } \omega \to \infty. \end{cases}$$

So, we set

$$r_1(\omega) = \begin{cases} \frac{1}{\frac{2d}{\Gamma(\frac{t}{d})(1+r)} + \frac{(-\omega)^t}{\mu_1}} & \text{if } \omega \leq 0 \\ \frac{\Gamma(\frac{t}{d})(1+r)+2\lambda\alpha g_0\omega}{2d+\lambda\omega} & \text{if } \omega \geq 0, \end{cases}$$

$$s_1(\omega) = \begin{cases} \frac{\omega x_1 - \lambda y_1^d \omega^2}{1+\lambda(-\omega)^{d+2}} & \text{if } \omega \leq 0 \\ \frac{\omega x_1 + \lambda r_1 \omega^2}{1+\lambda\omega^2} & \text{if } \omega \geq 0 \end{cases}$$

and

$$r_2(\omega) = \begin{cases} \frac{1}{\frac{2d}{\Gamma(\frac{t}{d})(1-r)} + \frac{(-\omega)^t}{\mu_2}} & \text{if } \omega \leq 0 \\ \frac{\Gamma(\frac{t}{d})(1-r)+\lambda\alpha q_1 g_0\omega}{2d+\lambda\omega} & \text{if } \omega \geq 0, \end{cases}$$

$$s_2(\omega) = \begin{cases} \frac{\omega x_2 - \lambda y_2^d \omega^2}{1+\lambda(-\omega)^{d+2}} & \text{if } \omega \leq 0 \\ \frac{\omega x_2 + \lambda r_1 \omega^2}{1+\lambda\omega^2} & \text{if } \omega \geq 0, \end{cases}$$

where $\lambda > 0$ can again be chosen arbitrary as a function of t and d.

8 Testing

Since we have constructed approximations $\widetilde{\Gamma}_d(\omega,t)$ which have the correct asymptotic behaviour for $\omega \to 0, \pm\infty$, we want to test how good these functions really approximate

the integral

$$\Gamma_d(\omega,t) = \int_0^\infty e^{-x^d+\omega x} x^{t-1}\, dx$$

with real $d > 1, t > 0, \omega$.

8.1 Polynomial approximation

Our first try is based on a polynomial approximation.
We know from Section 6 that $\Gamma_d(\omega,t) = \int_0^\infty e^{-x^d+\omega x} x^{t-1}\, dx$ is unknown but can be written as

$$\Gamma_d(\omega,t) = \int_0^\infty w(x) f(x)\, dx$$

for different weight functions $w(x)$ and functions $f(x)$ depending on the size of ω.

Example

If $\omega < 0$ and $|\omega|$ is 'large', then $w(x) = e^{\omega x} x^{t-1}$ and $f(x) = e^{-x^d}$.

Now, we want to approximate the function $f(x)$ by a polynomial $p(x)$ which can be integrated exactly by the Gaussian quadrature formulas determined in Section 6. This can be done in the following way.

8.1.1 Polynomial approximation algorithm

Given a function $f(x)$, we want to fit polynomials $p(x)$ and $e(x)$ to points x_k and function values $f(x_k)$ such that

$$|p(x_k) - f(x_k)| \leq e(x_k) \text{ for all } k.$$

We use as x_k $(k = 1,\ldots,2N+1)$ the nodes x_{2k} $(k = 1,\ldots,N)$ of an N-point Gaussian quadrature formula, the geometric means $x_{2k+1} = \sqrt{x_{2k} x_{2k+2}}$ $(k = 1,\ldots,N-1)$, and the points $x_1 = \dfrac{x_2^2}{x_3}$ and $x_{2N+1} = \dfrac{x_{2N}^2}{x_{2N-1}}$.
If $N = 1$, we take $x_1 = \dfrac{x_2^2}{2}$ and $x_3 = 2x_2^2$.

These additional points are chosen in this way to include the points between the Gaussian quadrature nodes where the largest errors typically occur.

If the two polynomials should be of degree d, we make the ansatz

$$p(x) = \sum_{n=0}^{d} p_n P_n(x) \text{ and } e(x) = \sum_{n=0}^{d} e_n P_n(x),$$

where P_n are some polynomials of degree n for $n = 0, \ldots, d$ and p_n, e_n for $n = 0, \ldots, d$ are the coefficients that we want to compute.
Our goal is to minimize the error, in particular we want to solve the problem

$$\begin{aligned} &\text{minimize} && \sum_{n=0}^{2N+1} e(x_k) \\ &\text{subject to} && |p(x_k) - f(x_k)| \leq e(x_k), \quad k = 1, \ldots, 2N+1. \end{aligned}$$

Since we want to formulate this optimization problem as a linear program, note that the constraints can also be expressed as

$$\begin{aligned} -p(x_k) + e(x_k) &\geq -f(x_k), \quad k = 1, \ldots, 2N+1 \\ p(x_k) + e(x_k) &\geq f(x_k), \quad k = 1, \ldots, 2N+1. \end{aligned}$$

If we now define

$$\widetilde{M} := \begin{pmatrix} P_0(x_1) & \cdots & P_d(x_1) \\ \vdots & & \vdots \\ P_0(x_{2N+1}) & \cdots & P_d(x_{2N+1}) \end{pmatrix} \quad \text{and} \quad u = \begin{pmatrix} u_p \\ u_e \end{pmatrix} := \begin{pmatrix} p_0 \\ \vdots \\ p_d \\ e_0 \\ \vdots \\ e_d \end{pmatrix},$$

we know that

$$\widetilde{M}u_p = \begin{pmatrix} p(x_1) \\ \vdots \\ p(x_{2N+1}) \end{pmatrix} \quad \text{and} \quad \widetilde{M}u_e = \begin{pmatrix} e(x_1) \\ \vdots \\ e(x_{2N+1}) \end{pmatrix}.$$

So, our optimization problem is equivalent to the linear program

$$\begin{aligned} &\text{minimize} && c^T u \\ &\text{subject to} && Mu \geq f, \end{aligned}$$

where

$$c = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \sum_{i=1}^{2N+1} P_0(x_i) \\ \vdots \\ \sum_{i=1}^{2N+1} P_d(x_i) \end{pmatrix}, \quad M = \begin{pmatrix} -\widetilde{M} & \widetilde{M} \\ \widetilde{M} & \widetilde{M} \end{pmatrix} \text{ and } f = \begin{pmatrix} -f(x_1) \\ \vdots \\ -f(x_{2N+1}) \\ f(x_1) \\ \vdots \\ f(x_{2N+1}) \end{pmatrix}.$$

Let us now apply this method 8.1.1 to the functions $f(x)$.
Because we want to choose the degree of the polynomials $p(x)$ and $e(x)$ in a way that they are integrated exactly by the Gaussian quadrature formula of Nth order, we have:

- If $N = 1 \Rightarrow d = 1$.
- If $N = 2 \Rightarrow d = 3$.

If we now define

$$\widehat{\Gamma}_d(\omega,t) := \int_0^\infty w(x)p(x)\,dx,$$

which is computable for each $d > 1, \omega$ and $t > 0$, we have that

$$\begin{aligned}\left|\Gamma_d(\omega,t) - \widehat{\Gamma}_d(\omega,t)\right| &= \left|\int_0^\infty w(x) f(x)\,dx - \int_0^\infty w(x) p(x)\,dx\right| \\ &\leq \int_0^\infty w(x)\,|f(x) - p(x)|\,dx \\ (*) &\leq \int_0^\infty w(x) e(x)\,dx,\end{aligned}$$

where

$$\delta_d(\omega,t) := \int_0^\infty w(x) e(x)\,dx$$

is computable for each $d > 1, \omega$ and $t > 0$.

Remark:

Even if the inequality $(*)$ is not true in general, we still hope that due to the choice of the x_k $(k = 1,\ldots,2N+1)$, it is approximately true.
Then,

$$\begin{aligned}|\Gamma_d(\omega,t) - \widetilde{\Gamma}_d(\omega,t)| &= |\Gamma_d(\omega,t) - \widehat{\Gamma}_d(\omega,t) + \widehat{\Gamma}_d(\omega,t) - \widetilde{\Gamma}_d(\omega,t)| \\ &\leq |\Gamma_d(\omega,t) - \widehat{\Gamma}_d(\omega,t)| + |\widehat{\Gamma}_d(\omega,t) - \widetilde{\Gamma}_d(\omega,t)| \\ &\leq \delta_d(\omega,t) + |\widehat{\Gamma}_d(\omega,t) - \widetilde{\Gamma}_d(\omega,t)|,\end{aligned}$$

where

$$\varepsilon_d(\omega,t) := \delta_d(\omega,t) + |\widehat{\Gamma}_d(\omega,t) - \widetilde{\Gamma}_d(\omega,t)|$$

is again computable for each $d > 1, \omega$ and $t > 0$.
Thus, if the polynomial approximation works, we get an upper bound for the absolute errors.

Case 1: $\omega \to -\infty$
First we start with the polynomial approximation method based on $\omega \to -\infty$. Therefore,

$w(x) = e^{\omega x}x^{t-1}$ and $f(x) = e^{-x^d}$. We want to find out how the error $\varepsilon_d(\omega,d)$ behaves and if the polynomial approximation is a useful method for estimating the error. As already mentioned, we use the Gaussian formulas of first and second order in the polynomial approximation. To emphasize which one is used, we write either a 1 or a 2 in square brackets.

Example 1. *If we take $d = 2$, $t = 3$, this gives us:*

	$\Gamma_d(\omega,t)$	$\widehat{\Gamma}_d(\omega,t)[1]$	$\widetilde{\Gamma}_d(\omega,t)[1]$	$\delta_d(\omega,t)[1]$	$\widehat{\Gamma}_d(\omega,t)[2]$	$\widetilde{\Gamma}_d(\omega,t)[2]$	$\delta_d(\omega,t)[2]$
$\omega = -50$	1.5924e−5	1.5928e−5	1.5942e−5	1.4174e−8	1.5928e−5	1.5925e−5	2.6130e−9
$\omega = -20$	2.4277e−4	2.4317e−4	2.4427e−4	1.2725e−6	2.4252e−4	2.4341e−4	1.1301e−6
$\omega = -10$	0.0018	0.0018	0.0018	2.9538e−5	0.0018	0.0018	7.1196e−5
$\omega = -9$	0.0024	0.0024	0.0024	4.4869e−5	0.0023	0.0024	1.1449e−4
$\omega = -8$	0.0033	0.0033	0.0034	6.8850e−5	0.0032	0.0034	1.8490e−4
$\omega = -7$	0.0047	0.0047	0.0048	1.0372e−4	0.0046	0.0048	2.9472e−4
$\omega = -6$	0.0070	0.0071	0.0070	1.3842e−4	0.0069	0.0071	4.3971e−4
$\omega = -5$	0.0111	0.0111	0.0107	7.5304e−5	0.0110	0.0108	4.9166e−4
$\omega = -4$	0.0185	0.0185	0.0164	6.9625e−4	0.0190	0.0166	3.6601e−4
$\omega = -3$	0.0337	0.0337	0.0228	0.0058	0.0366	0.0231	0.0068
$\omega = -2$	0.0684	0.0607	0.0168	0.0343	0.0799	0.0176	0.0474
$\omega = -1$	0.1592	0.0704	0.0018	0.0701	0.2407	0.0028	0.2401

Remark:

From now on, the values $\Gamma_d(\omega,t)$ are always calculated by the built-in function *quadgk* of Matlab.

Observations:

- The values of $\widetilde{\Gamma}_d(\omega,t)$ appear to be really good approximations to $\Gamma_d(\omega,t)$.
- The polynomial approximation method seems to perform reasonably well in the cases where $\omega \in [-10,-2]$ and the error can be estimated reliably by

$$\varepsilon_d(\omega,t) = \delta_d(\omega,t) + |\widehat{\Gamma}_d(\omega,t) - \widetilde{\Gamma}_d(\omega,t)|.$$

- In the case $\omega = -1$ neither $\widetilde{\Gamma}_d(\omega,t)$ nor $\widehat{\Gamma}_d(\omega,t)$ are good approximations. Thus, we try to get more information about this behaviour.

Example 2. *If we enlarge the parameter t, in particular if we take* $d = 2$, $t = 10$ *and* $\omega = [-3 : -1]$, *this gives us:*

	$\Gamma_d(\omega,t)$	$\widehat{\Gamma}_d(\omega,t)[1]$	$\widetilde{\Gamma}_d(\omega,t)[1]$	$\delta_d(\omega,t)[1]$	$\widehat{\Gamma}_d(\omega,t)[2]$	$\widetilde{\Gamma}_d(\omega,t)[2]$	$\delta_d(\omega,t)[2]$
$\omega = -3$	0.0456	0.1274	1.7256e−4	0.1273	1.4393	5.7383e−5	1.4392
$\omega = -2$	0.2399	0.2280	3.8474e−5	0.2280	43.531	4.088e−10	43.531
$\omega = -1$	1.5234	1.6799e−6	0.0271	1.6799e−6	1366.9	8.8679e−22	1366.9

Observations:

- We discover that in these cases the polynomial approximation fails. The values of $\widehat{\Gamma}_d(\omega,t)[2]$ and $\delta_d(\omega,t)[2]$ are far to large and $\widehat{\Gamma}_d(\omega,t)$ and $\delta_d(\omega,t)$ are almost the same. In the case $\omega = -1$, the estimation

  $$\varepsilon_d(\omega,t)[1] = \delta_d(\omega,t)[1] + |\widehat{\Gamma}_d(\omega,t)[1] - \widetilde{\Gamma}_d(\omega,t)[1]|$$

  even results in an underestimation.
  A reason for that failure may be that the points x_k $(k = 1,\ldots,2N+1)$ which are used in the polynomial algorithm are not appropriate for large t. Moreover, $|\omega|$ is small. So maybe things get better if we take the polynomial approximation method based on the case $\omega \to 0$. But we will come to that later.

- Even if we can still estimate our error $\varepsilon_d(\omega,t)[2]$ by the polynomial approximation in cases of failure, it results in a huge overestimation.
  $\varepsilon_d(\omega,t)[1]$ can only be estimated in the usual way for $\omega = -3,-2$.

- Compared to the actual values $\Gamma_d(\omega,t)$, we realize that the blending functions $\widetilde{\Gamma}_d(\omega,t)[2]$ perform a bad approximation.
  In fact a further increase of the parameter t makes $\Gamma_d(\omega,t)$ larger and larger while $\widetilde{\Gamma}_d(\omega,t)[2]$ tends to zero.
  The blending functions $\widetilde{\Gamma}_d(\omega,t)[1]$ are at least a bit better in such cases even though they are not growing fast enough.

Case 2: $\omega \to 0$ with $\omega < 0$

Now, we try the polynomial approximation method based on $\omega \to 0$. Thus, $w(x) =$

$e^{-x^d}x^{t-1}$ and $f(x) = e^{\omega x}$.
As a basis for the polynomial approximation we take again the Gaussian formulas of order 1 and 2.

Let us have a look at the example where the polynomial approximation failed in the case $\omega \to -\infty$.

Example 3. *Let $d = 2$, $t = 10$ and $\omega = [-3 : -1]$.*

	$\Gamma_d(\omega,t)$	$\widehat{\Gamma}_d(\omega,t)[1]$	$\widetilde{\Gamma}_d(\omega,t)[1]$	$\delta_d(\omega,t)[1]$	$\widehat{\Gamma}_d(\omega,t)[2]$	$\widetilde{\Gamma}_d(\omega,t),[2]$	$\delta_d(\omega,t)[2]$
$\omega = -3$	0.0456	0.1605	1.7256e−4	0.1432	0.0417	5.7383e−05	0.0043
$\omega = -2$	0.2399	0.5286	3.8474e−5	0.3755	0.2347	4.0886e−10	0.0069
$\omega = -1$	1.5234	2.0473	0.0271	0.6921	1.5210	8.8679e−22	0.0038

Observations:

- We can observe that using the polynomial approximation method based on the Gaussian quadrature formula of second order in the case of $\omega \to 0$ solves our problem from above. So in this case we get a useful estimate of our error $\varepsilon_d(\omega,t)[2]$ by
$$\varepsilon_d(\omega,t)[2] = \delta_d(\omega,t)[2] + |\widehat{\Gamma}_d(\omega,t)[2] - \widetilde{\Gamma}_d(\omega,t)[2]|.$$
- Even though the values $\varepsilon_d(\omega,t)[1]$ result in an overestimation, they can still be used.

Remark:

From now on, we restrict ourselves to the case based on the Gaussian quadrature formulas of second order although the other case may also have its advantages. However, the following methods can easily be adapted to the first order case.

Notation:

$PA_{-\infty}$ will denote the polynomial approximation method based on $\omega \to -\infty$ and PA_0 the polynomial approximation method based on $\omega \to 0$.

The simplest recipe for deciding which way of estimating our error to choose is the following.
One can always choose the method $PA_{-\infty}$ first and if the polynomial approximation fails (if $\widehat{\Gamma}_d(\omega,t)$ and $\delta_d(\omega,t)$ are far too large and almost equal), then one should try the method PA_0. If this one does not work either (there are cases where $\delta_d(\omega,t) < 0$ which may give an underestimated absolute error), we take the overestimated error from method $PA_{-\infty}$.
Even if there exist some cases where the method $PA_{-\infty}$ does not fail but the method PA_0 would give slightly better results, the estimated error derived from method $PA_{-\infty}$ is satisfying enough.

Of course, it is also possible to write a program which always chooses the best method. But in terms of our project, this is not necessary.

8.2 Analysis of the absolute errors

Knowing how to give a reasonable upper bound on the error $|\Gamma_d(\omega,t) - \widetilde{\Gamma}_d(\omega,t)|$ for fixed $d > 1, t > 0$ and $\omega < 0$, our next goal is to find a simple closed expression $e_d(\omega,t)$ where $e_d(\omega,t)$ is the solution to the following linear program:

We have the input vectors

$$d = \begin{pmatrix} d(1) \\ \vdots \\ d(l_1) \end{pmatrix}, \omega = \begin{pmatrix} \omega(1) \\ \vdots \\ \omega(l_2) \end{pmatrix} \text{ and } t = \begin{pmatrix} t(1) \\ \vdots \\ t(l_3) \end{pmatrix}.$$

Without loss of generality, we assume that $l_1 = l_2 = l_3 = l$.
We make the ansatz

$$e_d(\omega,t) = (x_1 + x_2 t)\omega + (x_3 + x_4 d)t + x_5 + x_6 d + x_7 d t^2 + \frac{x_8}{\omega^2} + x_9 t^2 + \frac{x_{10} t}{w^2} + \frac{x_{11}\log(t)}{w^2}$$

and consider the optimization program

$$\begin{aligned} &\text{minimize} && \sum_{i,j,k=1}^{l} e_{d(i)}(\omega(j),t(k)) \\ &\text{subject to} && e_{d(i)}(\omega(j),t(k)) \geq |\varepsilon_{d(i)}(\omega(j),t(k))|, \quad i,j,k \in \{1,\ldots,l\}. \end{aligned}$$

This problem is equivalent to the linear program

$$\begin{aligned} &\text{minimize} && c^T x \\ &\text{subject to} && Ax \geq b, \end{aligned}$$

with

$$c = \left(l^2 \sum \omega, l \sum d\omega, l^2 \sum t, l \sum dt, l^3, l^2 \sum d, l \sum dt^2, l^2 \sum \frac{1}{\omega^2}, l^2 \sum t^2, l \sum \frac{t}{\omega^2}, l \sum \frac{\log t}{\omega^2} \right)$$

and

$$A = \begin{pmatrix} \omega(1) & d(1)\omega(1) & t(1) & d(1)t(1) & 1 & d(1) & d(1)t(1)^2 & \frac{1}{\omega(1)^2} & t(1)^2 & \frac{t(1)}{\omega(1)^2} & \frac{\log t(1)}{\omega(1)^2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \omega(l) & d(l)\omega(l) & t(l) & d(l)t(l) & 1 & d(l) & d(l)t(l)^2 & \frac{1}{\omega(l)^2} & t(l)^2 & \frac{t(l)}{\omega(l)^2} & \frac{\log t(l)}{\omega(l)^2} \end{pmatrix}.$$

Thus, A is a $l^3 \times 8$ matrix because every row corresponds to one combination.

The vector b consists of the values $|\varepsilon_{d(i)}(\omega(j),t(k))|,\ i,j,k \in \{1,\ldots,l\}$ in the proper order which implies that b is a l^3-dimensional vector.

Remark:

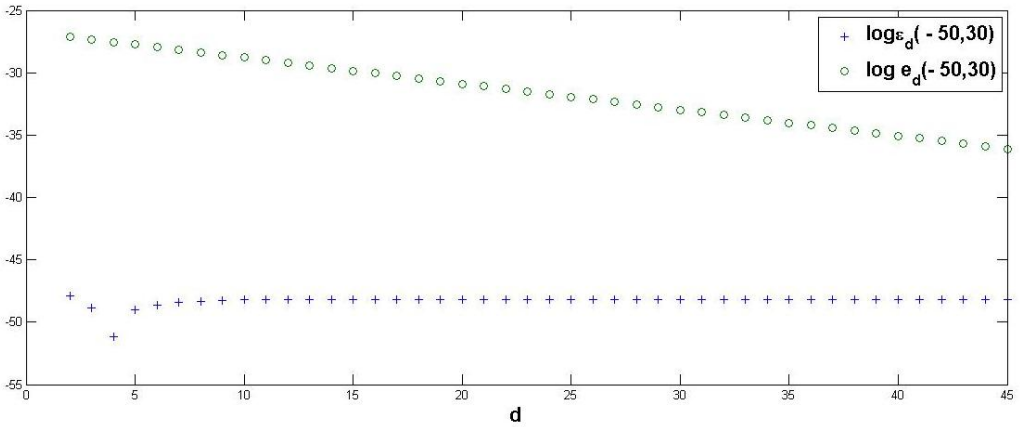
Since the absolute errors are all very small, we solve the linear program

$$\begin{aligned} &\text{minimize} && c^T x \\ &\text{subject to} && Ax \geq \log b. \end{aligned}$$

Let $\omega = [-101 : 10 : -1]$, $d = [2 : 4 : 42]$, and $t = [1 : 4 : 41]$. Then the solution to the linear program is the vector

$$\begin{aligned} x \approx \big(&0.0262 \quad 0.0212 \quad -0.6114 \quad -0.0051 \quad -3.2905 \quad 0.0054 \\ &-0.0001 \quad 3.2082 \quad 0.0312 \quad 0.2784 \quad 1.3888\big)^T. \end{aligned}$$

Example 4. *To get an impression of the quality of this approximation, we fix* $\omega = -50$*,* $t = 30$ *and let* $d = [2 : 45]$.



We can see clearly that the values $e_d(\omega,t)$ are much too high.
The function $e_d(\omega,t)$ can be improved if we slightly change our linear program.
Some errors $\varepsilon_d(\omega,t)$ are relatively considered very high, which means that the approximation $\widetilde{\Gamma}_d(\omega,t)$ is quite bad. As these cases are not really interesting anyway in terms of the error description, we classify them as *bad values* and only accept *good values*

with the property that

$$\frac{|\widetilde{\Gamma}_d(\omega,t)-\Gamma_d(\omega,t)|}{\Gamma_d(\omega,t)} \leq 0.5.$$

So, we will only analyse the regions containing *good values*. Concerning the *bad values*, further considerations would be necessary.

This restriction on the relative errors will improve our results as the Example 5 illustrates.

Remark:

In Section 9 we will discuss in detail the classification between the *good* and the *bad values*. So, we will find out for which values (d,ω,t) the property

$$\frac{|\widetilde{\Gamma}_d(\omega,t)-\Gamma_d(\omega,t)|}{\Gamma_d(\omega,t)} \leq 0.5$$

is satisfied and for which it is not.

Remark:

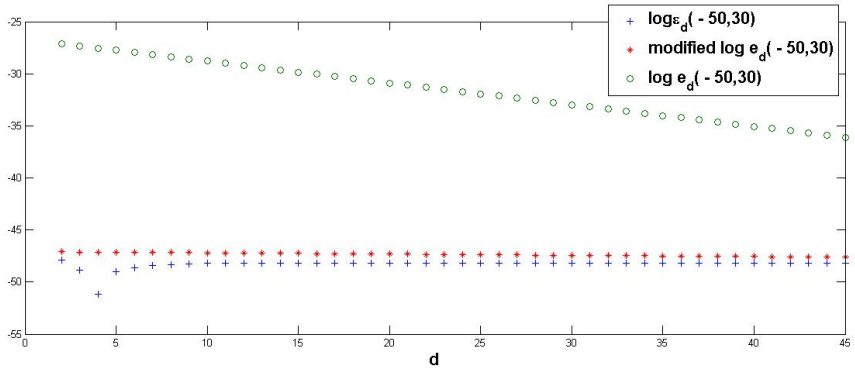
There is one disadvantage in that modification. In comparison to calculating an upper bound for the absolute errors $|\widetilde{\Gamma}_d(\omega,t)-\Gamma_d(\omega,t)|$ where we do not use any built-in function of Matlab for calculating one of the arising integrals, we use the built-in function *quadgk* for calculating the values $\Gamma_d(\omega,t)$ appearing in the relative error. Nevertheless, we will see that the results are significantly better.

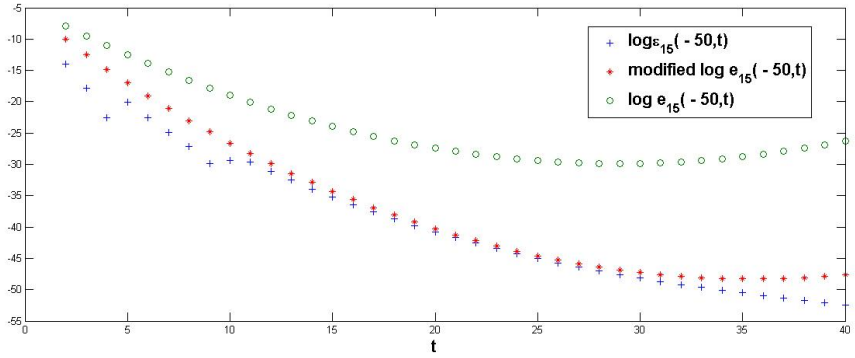
Example 5. *Modified version.*

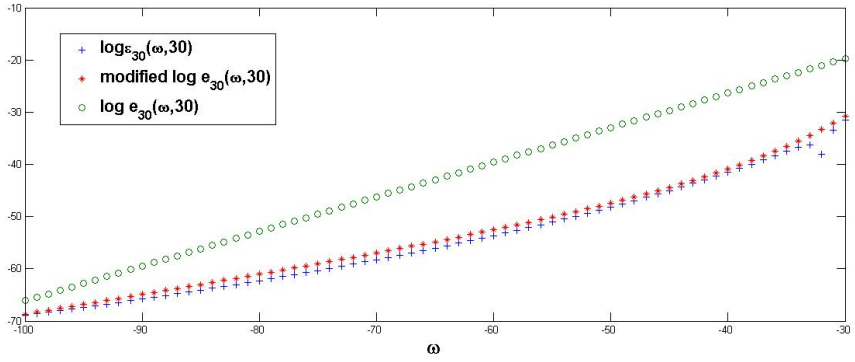
The following figures show that the function $e_d(\omega,t)$ which is defined through the solution vector

$$x \approx \big(0.0078 \quad 0.0111 \quad -2.0105 \quad 0.0011 \quad -4.9052 \quad -9.59\text{e}-4 \quad -4.72\text{e}-5 \quad 718.9253 \quad 0.0330 \quad 692.0794 \quad -2.24\text{e}+3\big)^T$$

is much better if we only accept *good values*.







Altogether, it seems that the blending functions are nice approximations in many cases and the polynomial approximation method for estimating the absolute error works well. In Subsection 8.1 we have seen that some difficulties may arise for example if $|\omega|$ gets

small.

We have constructed a function $e_d(\omega,t)$ which approximates the absolute error of the *good values* in the intervals which we have chosen for the linear program.

Case 3: $\omega \to +\infty$
Considering now the positive values of ω, we first analyse the polynomial approximation method based on $\omega \to +\infty$. As before, we restrict ourselves without loss of generality to the Gaussian formula of order 2 as the basis for this method.

To gain some insight in the complexity of this case, we start with an example.

Example 6. *We take* $d = 2$, $t = 1$ *and* $\omega = [5 : 10]$.

	$\Gamma_d(\omega,t)$	$\widehat{\Gamma}_d(\omega,t)$	$\widetilde{\Gamma}_d(\omega,t)$	$\delta_d(\omega,t)$
$\omega = 5$	917.9670	459.0769	755.2794	4.3422e−14
$\omega = 6$	1.4362e+4	7.1812e+3	1.1815e+4	2.5827e−12
$\omega = 7$	3.7041e+5	1.8520e+5	3.0523e+5	5.8897e−11
$\omega = 8$	1.5750e+7	7.8751e+6	1.3010e+7	5.5416e−9
$\omega = 9$	1.1042e+9	5.5209e+8	9.1445e+8	1.3317e−7
$\omega = 10$	1.2763e+11	6.3813e+10	1.0598e+11	7.0528e−5

Observations:

- We immediately see that this case is much more complex. Here, the polynomial approximation method is not an acceptable choice because in estimating our error we assume that the inequality

$$\left|\Gamma_d(\omega,t) - \widehat{\Gamma}_d(\omega,t)\right| \leq \int_0^\infty w(x)e(x)\,dx = \delta_d(\omega,t),$$

  is at least approximately true. Comparing the values in the tabular, this is not true at all. In Example 8 the polynomial approximation method based on $\omega \to 0$ is used for $\omega > 0$.

- This example let us assume that the values $\widetilde{\Gamma}_d(\omega,t)$ grow too slowly in general for $\omega > 0$. In fact, this is not true as the Example 7 shows.

Example 7.

	$\Gamma_d(\omega,t)$	$\widetilde{\Gamma}_d(\omega,t)$
$d=50,\ \omega=67,\ t=4$	2.2509e+27	8.0717e+27
$d=5,\ \omega=67,\ t=40$	2.3177e+55	2.4281e+55
$d=20,\ \omega=6,\ t=4$	38.3104	52.4592

Case 4: $\omega \to 0$ with $\omega > 0$
Using the polynomial approximation method based on $\omega \to 0$, leads to the following results.

Example 8. *We take* $d=2$, $t=1$ *and* $\omega=[5:10]$.

	$\Gamma_d(\omega,t)$	$\widehat{\Gamma}_d(\omega,t)$	$\widetilde{\Gamma}_d(\omega,t)$	$\delta_d(\omega,t)$
$\omega=5$	917.9670	971.8989	755.2794	840.2008
$\omega=6$	1.4362e+4	1.1458e+4	1.1815e+4	1.1003e+4
$\omega=7$	3.7041e+5	1.4424e+5	3.0523e+5	1.4266e+5
$\omega=8$	1.5750e+7	1.8496e+6	1.3010e+7	1.8441e+6
$\omega=9$	1.1042e+9	2.3838e+7	9.1445e+8	2.3818e+7
$\omega=10$	1.2763e+11	3.0763e+8	1.0598e+11	3.0757e+8

Looking at this example, the estimation of the error would not be wrong in the sense of violating our assumptions, but it results in a large overestimation which is not useful either.
Thus, we leave the idea of the polynomial approximation method for estimating the absolute errors behind and use a completely different approach.

8.3 Analysis of the relative errors

Since the size of the absolute errors differs enormously, we now focus on the relative errors

$$\eta_d(\omega,t) := \frac{\widetilde{\Gamma}_d(\omega,t)-\Gamma_d(\omega,t)}{\Gamma_d(\omega,t)}$$

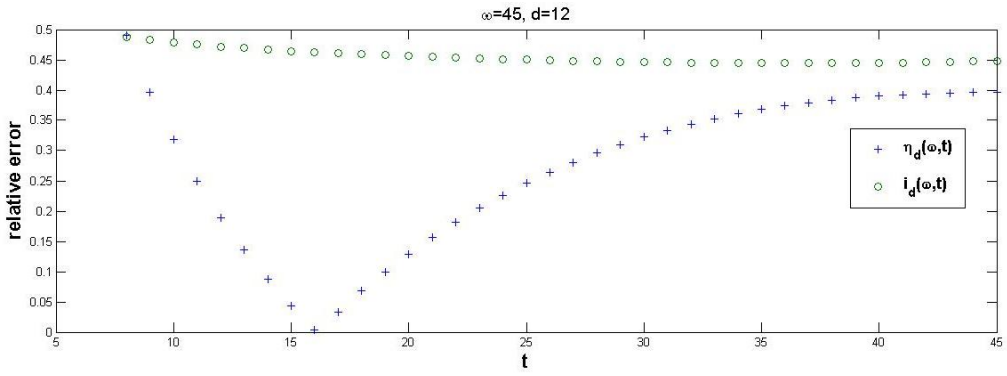
for real $d > 1$, $t > 0$ and ω, where the $\Gamma_d(\omega,t)$ is again calculated with the built-in function *quadgk* in Matlab.

Analogously to the case of the absolute errors, our goal is to find a simple closed expression $i_d(\omega,t)$, where $i_d(\omega,t)$ is the solution to the linear program from Section 8.2. Naturally, we have to take $\eta_d(\omega,t)$ instead of the values $\varepsilon_d(\omega,t)$.
As before, only values with relative error smaller or equal 0.5 are taken into account for the calculation of $i_d(\omega,t)$.

For the case $\omega < 0$, we simply take the same ansatz as in Section 8.2.
But for the case $\omega > 0$, it turns out that it is really challenging to find a useful solution. Despite trying various approaches, the best of them is still not very satisfying. The most useful ansatz is

$$i_d(\omega,t) = (x_1 + x_2 t)\omega + x_3 t^4 + x_4 dt + x_5 + x_6 d + x_7 \omega t + \frac{x_8}{\omega^2} + x_9 \frac{d}{t} + x_{10} \frac{d}{\omega^2} + x_{11}\sqrt{d} + x_{12} \frac{d\omega}{t} + x_{13} \frac{t}{\omega^2} + x_{14} d\sqrt{\omega}.$$

The main reason for that difficulty is that the errors vary too much. If we for instance perform only a little change in t, $\eta_d(\omega,t)$ may increase or decrease a lot. To visualize that matter of fact, let us take $\omega = 45$, $d = 12$ and $t = [8:45]$. A comparison of the resulting function $i_d(\omega,t)$ based on $\omega = [1:4:89]$, $d = [2:2:46]$, $t = [1:2:45]$ and the actual relative errors $\eta_d(\omega,t)$ looks like this:



We can observe see that between $t = 8$ to $t = 16$, the relative error decreases from just under 0.5 to almost 0. Afterwards, the error increases again up to over 0.4.

However, in our case the function $i_d(\omega,t)$ is acceptable as it restricts the relative error. Unfortunately, the blending functions for $\omega > 0$ are simply not that satisfactory as we had intended. To get better approximations, one has to go back to Section 7 and try another ansatz for the blending functions.

The goal of this whole error estimation is to know how large the error might be in calculating the integral

$$I_{d,\omega,t}(f) := \int_0^\infty e^{-x^d+\omega x} x^{t-1} f(x) dx$$

with real $d > 1, t > 0$, ω, where f is a function of at most polynomial growth depending analytically on $\log x$.
Naturally, we want to apply Gaussian quadrature formulas to approximate this integral. In particular, we want to use Gaussian quadrature formulas from moments for integration because the kth moment for integration with the weight function $w(x) = e^{-x^d+\omega x} x^{t-1}$ is exactly the integral

$$\Gamma_d(\omega,t+k) := \int_0^\infty e^{-x^d+\omega x} x^{t+k-1}\, dx,$$

which we have treated in detail. We have found asymptotic formulas for $\Gamma_d(\omega,t)$ for $\omega \to 0, \pm\infty$, have constructed natural blending functions that have the correct limits and last but not least analysed the error.
It may happen that the polynomial coefficients needed in the Gaussian quadrature rule contain an increasing error due to the mentioned ill-condition of the transformation from power moments to the polynomial coefficients.

In any case, we know that the error functions are only valid for the so called *good values* and it still remains to classify them before we get to the actual application of the Gaussian quadrature formulas from moments for integration.

Remark:

As we need the values $\Gamma_d(\omega,t+k)\,(k=0,\ldots,2n-1)$ for applying the Gaussian quadrature formulas of order n, one can fix ω and d in order to find a more accurate function $\tilde{i}_d(\omega,t)$ under the assumption that we know for which ω and d we want to solve the integral $I_{d,\omega,t}(f)$, .

9 Good and bad values

In Section 8.2, we defined values with the property that

$$\frac{|\widetilde{\Gamma}_d(\omega,t)-\Gamma_d(\omega,t)|}{\Gamma_d(\omega,t)} \leq 0.5$$

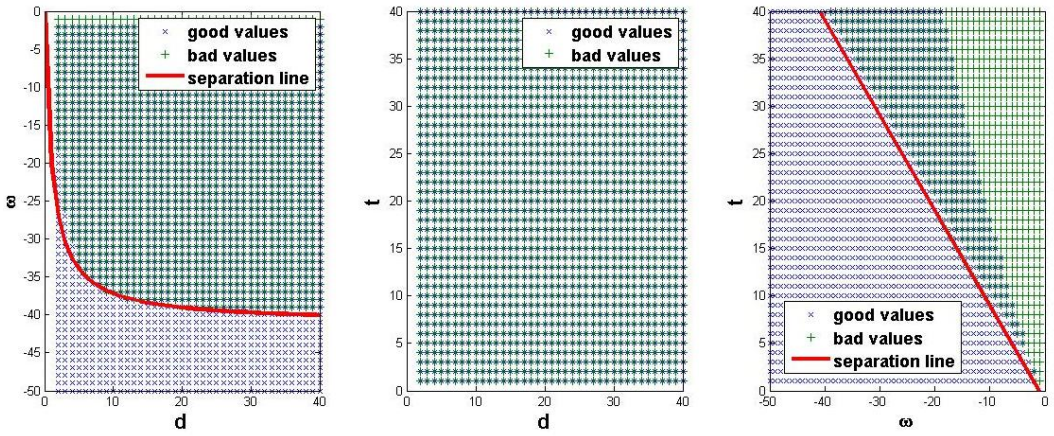
as good values and values that do not have that property as bad values.
As we need to know which values are the good ones, we do the following categorization.
For a wide-ranging set of values (d,ω,t), we test each value on the good value property and save the resulting good values in a matrix M_{good} and the bad values in a matrix M_{bad}. These matrices consist of three columns corresponding to the parameters d, ω an t.

Remark:

The following results are based on the blending functions of second order. The results of the first order case are similar and the categorization works analogously.

9.1 Case 1

First, we consider the case $\omega < 0$.
Creating a two-dimensional plot of each column pair of the matrices M_{good} and M_{bad} already reveals much information.



Remark:

The separation line in the left figure is approximately given by the function

$$s_1 = \frac{-41.07x}{x+1.04}$$

and the one in the right figure by

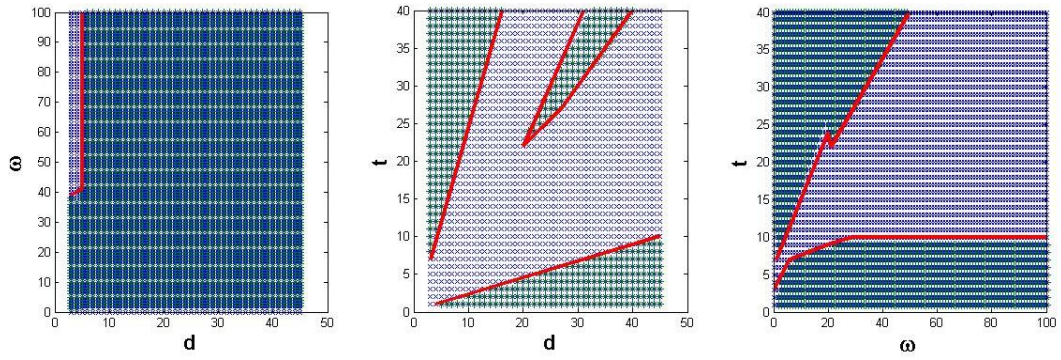
$$s_2 = -x-2.$$

Explanation: The good values are denoted by a cross and the bad values by a plus. If a plus and a cross overlap, a star is produced.
In the third figure we can observe that for all $d > 1$, the values (d, ω, t) for which (ω, t) lie in the upper right region are bad values while the region below the separation line consists of good values only.
There is also a region in the middle where there exists at least one $d > 1$ with $\eta_d(\omega, t) \leq 0.5$ as well as one $d > 1$ with $\eta_d(\omega, t) > 0.5$.
The first figure can be interpreted similarly while the second figure just demonstrates that for all $(d, t) \in (1, 40] \times (0, 40]$ there exists at least one $\omega_g < 0$ for which (d, ω_g, t) is good and one $\omega_b < 0$ for which (d, ω_b, t) is bad.
Altogether, if we could figure out what happens in the middle region of the third figure, all values with $\omega < 0$ would be classified.
Experimenting with the size of the value d shows that for increasing d, the middle region

reduces and turns into a bad area from the right. In particular, the last point in our figure which is converted from a star into a plus is $(\omega,t) = (-42,40)$ which happens at $d = 51$.

9.2 Case 2

Having fully treated the case $\omega < 0$, we want to focus on the case $\omega > 0$.
As before, we create a two-dimensional plot of each column pair of the matrices M_{good} and M_{bad}.



Remark:

The separation line in the left figure is approximately the linear interpolation of the points

- $(3,39)$, $(5,41)$ and $(5,100)$.

The three separation lines in the middle figure are the linear interpolations of the points

- $(16,40)$, $(3,7)$
- $(4,1)$, $(45,10)$
- $(31,40)$, $(20,22)$, $(27,27)$, $(40,40)$

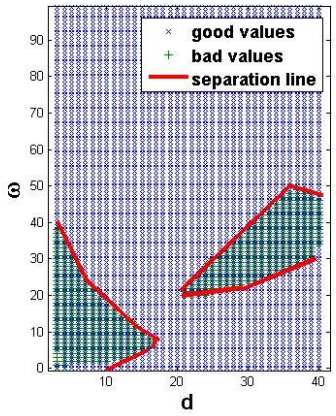
and the two separation lines in the right figure are given by the linear interpolations of the points

- $(1,7)$, $(20,24)$, $(21,22)$, $(50,40)$

- $(0,3)$, $(6,7)$, $(20,9)$, $(29,10)$, $(100,10)$.

In the middle as well as in the right figure, we can observe that there exists a quite large region of good values in the interior of the separation lines. Concerning the left figure, for every (d,ω) except for those in the small region marked-off through the separation line, there exists a $t_g > 0$ such that (d,ω,t_g) is a good point as well as a t_b such that (d,ω,t_b) is a bad point.

If we choose $t = [10:40]$ instead of $t = [1:40]$ and compare this figure to the left one above, we can deduce that mostly the small values of t are responsible for the bad values.



10 Application

Finally, we want to find out how good the integral

$$I_{d,\omega,t}(f) := \int_0^\infty e^{-x^d+\omega x} x^{t-1} f(x)dx$$

can be approximated based on the blending functions from Section 7.
Using the Chebyshev algorithm from Section 5.2, we obtain the coefficients $\delta_1,\ldots,\delta_n$ and $\gamma_2,\ldots,\gamma_n$ that we need for the n-dimensional Gaussian quadrature formula.

Remarks:

- We restrict ourselves in the following examples to the blending functions of second order, since the procedure works analogously for the first order.
- The calculation of the estimated relative errors is based on Subsection 8.3. For $\omega < 0$, the linear program is solved by means of the values $\omega = [-101:10:-1]$,

$d = [2:4:42]$ and $t = [1:4:41]$ while for $\omega > 0$ the values $\omega = [1:4:89]$, $d = [2:2:46]$ and $t = [1:2:45]$ form the basis of the linear program.

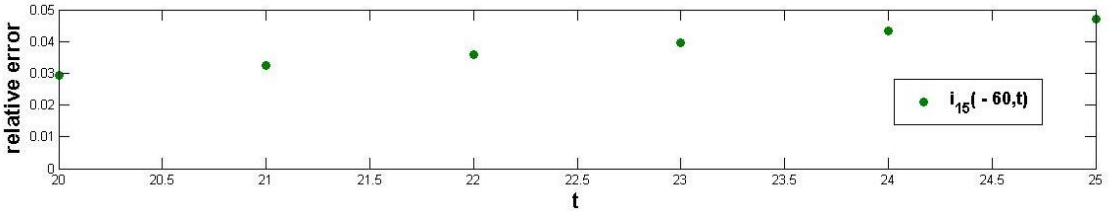
10.1 Example

Let

$$f(x) = (\log(x))^k \quad \text{for} \quad k = 1,2,3.$$

10.1.1 $f(x) = \log(x)$

We take $d = 15$, $\omega = -60$, $t = 20$ and want to apply a Gaussian quadrature formula of order 3. We know from Section 5.2 that we need 6 moments for integration. In particular, we need $\Gamma_{15}(-60,20+k)$ for $k = 0,\ldots,5$. We do not know the exact values of the moments for integration, but we have approximations $\widetilde{\Gamma}_{15}(-60,20+k)$ for $k = 0,\ldots,5$.
The corresponding estimated relative errors are visualized in the following figure.



The result of our approximate integral derived from the Gaussian quadrature formula based on our blending functions gives

$$I_{d,\omega,t}(f) \approx -3.7384\text{e -}19.$$

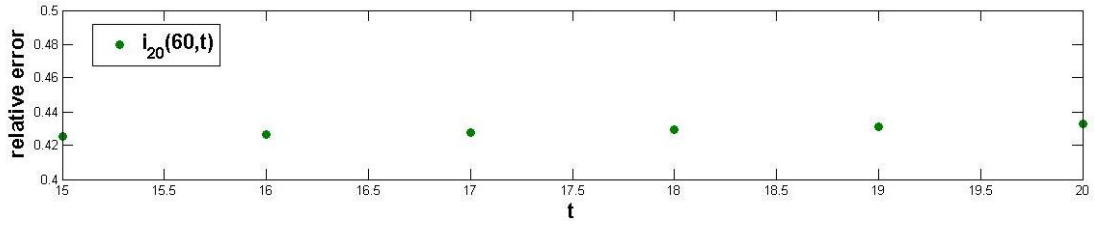
On the other hand, if we calculate $I_{d,\omega,t}(f)$ with Mathematica, we get

$$I_{d,\omega,t}(f) = -3.7391\text{e -}19.$$

10.1.2 $f(x) = (\log(x))^2$

Let us take $d = 20$, $\omega = 60$, $t = 15$ and perform a Gaussian quadrature formula of order 3.

Then the estimated relative errors in the required values $\widetilde{\Gamma}_{20}(60,15+k)$ for $k=0,\ldots,5$ are much larger than in the example 10.1.1.



The result of the Gaussian quadrature formula based on our blending functions is

$$I_{d,\omega,t}(f) \approx 1.2916\text{e+}23,$$

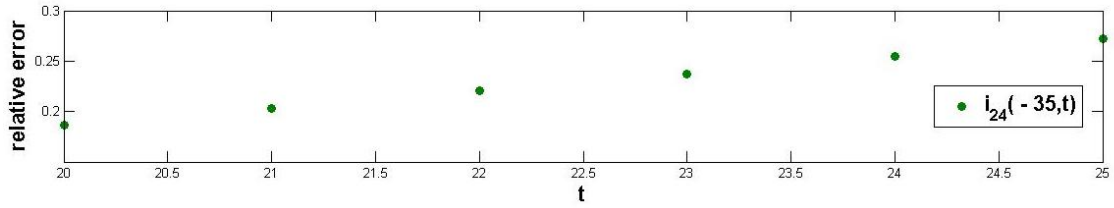
while

$$I_{d,\omega,t}(f) = 1.3206\text{e+}23$$

according to Mathematica.

10.1.3 $f(x) = (\log(x))^3$

We choose $d = 24, \omega = -35$ and $t = 20$.



The result derived from the approximated Gaussian quadrature formula of order 3 is

$$I_{d,\omega,t}(f) \approx -4.6415\text{e -}15,$$

and the result according to Mathematica is

$$I_{d,\omega,t}(f) = -4.6774\text{e -}15.$$

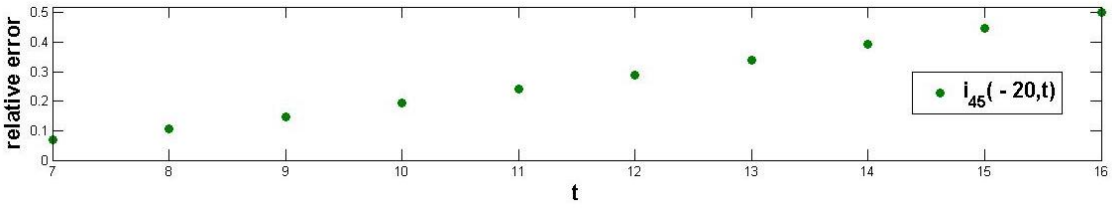
10.2 Example

Let

$$f(x) = \frac{1}{a+x} \quad \text{for} \quad a > 0.$$

10.2.1 $f(x) = \frac{1}{5+x}$

Let $d = 45, \omega = -20$ and $t = 7$. To approximate the Gaussian rule of order 5, we need 10 moments for integration. So, we need to evaluate $\widetilde{\Gamma}_{45}(-20, 7+k)$ for $k = 0, \ldots, 9$. The estimated relative errors are visualized below.



The result derived from the approximated Gaussian quadrature formula is

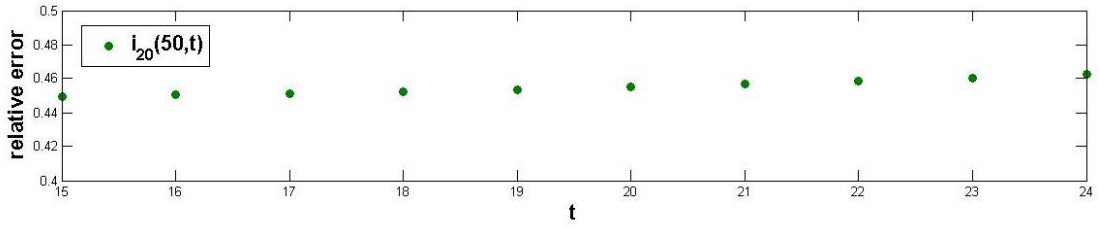
$$I_{d,\omega,t}(f) \approx 1.0512\text{e -7},$$

and the result according to Mathematica is

$$I_{d,\omega,t}(f) = 1.0517\text{e -7}.$$

10.2.2 $f(x) = \frac{1}{35+x}$

Let $d = 20, \omega = 50$ and $t = 15$. To approximate the Gaussian rule of order 5, we need 10 moments for integration. So, we need to evaluate $\widetilde{\Gamma}_{20}(50, 15+k)$ for $k = 0, \ldots, 9$. The estimated relative errors are visualized below.



The result derived from the approximated Gaussian quadrature formula is

$$I_{d,\omega,t}(f) \approx 1.2574\text{e+}19,$$

and the result according to Mathematica is

$$I_{d,\omega,t}(f) = 2.0098\text{e+}19.$$

Remarks:

- The larger the estimated relative errors, the less accurate the results may get.
- As we only consider approximate values for $\Gamma_d(\omega,t)$, it may also happen that some of the calculated weights or nodes needed for the Gaussian quadrature formula are negative which in theory contradicts Theorems 3 and 5, respectively.
- There are even cases in which the calculated weights and nodes are complex numbers which should not be the case either.
- Since the relative error may accumulate, the application of this approximate Gaussian quadrature formulas is only of limited suitability. Especially the relative errors according to values (d,ω,t) with $\omega > 0$ are mostly too large to achieve enough accuracy.
  One possibility would be to modify the classification of good and bad values in a way that the upper limit of the relative error is decreased.

11 Conclusion

In this thesis, we mainly focused on Gaussian quadrature rules. We proved basic properties and treated the (modified) Chebyshev algorithm which transforms the (modified) moments for integration to the coefficients needed for applying the Gaussian quadrature formulas.
After the theoretic part, we used first and second order Gaussian quadrature rules to find the asymptotic formulas needed for the construction of natural blending functions which are supposed to act as approximations $\widetilde{\Gamma}_d(\omega,t)$ for for the family of integrals

$$\Gamma_d(\omega,t) = \int_0^\infty e^{-x^d+\omega x} x^{t-1}\, dx$$

with real $d > 1$, $t > 0$ and ω.
This family of integrals includes all moments for integration with the weight function $w(x) = e^{-x^d+\omega x} x^{t-1}$ with real $d > 1$, $t > 0$ and ω.
With the use of the approximate moments for integration, we finally obtained approximate Gaussian quadrature formulas for the family of integrals introduced at the beginning of the thesis:

$$I_{d,\omega,t}(f) := \int_0^\infty e^{-x^d+\omega x} x^{t-1} f(x)\, dx$$

with real $d > 1$, $t > 0$ and ω.

Analysing the approximation quality of the blending functions constructed in Section 7, it turned out that there is a large range of d, ω and t for which the approximations $\widetilde{\Gamma}_d(\omega,t)$ are really good. In order to visualize this range of d, ω and t for which the approximations are good, a classification has been made.

Further research

Even though the approximations are satisfactory in large regions, there are some further issues which could be considered in order to improve the results.
As we have already mentioned in Section 10, one possibility would be to modify the classification in a way that the upper limit of the relative error is decreased. Most likely, this would result in better approximations of the Gaussian quadrature formulas. In re-

turn, the region of good values would reduce.
In future work, the asymptotic formulas from Section 6 could be based on Gaussian quadrature formulas of higher order and especially for values (d, ω, t) with $\omega > 0$ or $|\omega|$ small, we could modify the ansatz of the blending functions made in Section 7.

References

[1] K.A. Bugaev – Exactly solvable methods: The road towards a rigorous treatment of phase transitions in finite systems, Physics of Particles and Nuclei 28, 2007, 447-468

[2] M.E. Fisher and B.U. Felderhof – Phase transitions in one-dimensional cluster-interaction fluids IA. Thermodynamics, Ann. Physics 58, 1970, 176-216

[3] H. Engels – Numerical Quadrature and Cubature, Academic Press 1980

[4] Hermann Schichl – Numerik 1, Skriptum zur Vorlesung SS 2008/09

[5] Josef Stoer – Numerische Mathematik 1, 7.Auflage – Springer-Verlag Berlin Heidelberg New York, 1994

[6] Josef Stoer, Roland Bulirsch – Numerische Mathematik 2, 3.Auflage – Springer-Verlag Berlin Heidelberg New York, 1990

[7] Harro Heuser – Lehrbuch der Analysis Teil 2, 14.Auflage – Vieweg + Teubner Verlag, 2008

[8] Walter Gautschi – On the construction of Gaussian quadrature rules from modified moments, Mathematics of Computation, Volume 24, 1970

[9] R. A. Sack and A. F. Donovan – An algorithm for Gaussian quadrature formulas given modified moments, Numerische Mathematik, Volume 18, 1972

[10] Gene H. Golub and John H. Welsch – Calculation of Gauss Quadrature Rules, Mathematics of Computation, Volume 23, 1969

[11] I. P. Mysovskih – On the construction of cubature formulas with the smallest number of nodes, Doklady Akademii Nauk SSSR, Volume 9, 1968

[12] Walter Gautschi – On generating orthogonal polynomials, SIAM Journal on Scientific and Statistical Computing, Volume 3, 1974

[13] John C. Wheeler – Modified Moments and Gaussian quadratures, Rocky Mountain Journal of Mathematics, Volume 4, 1974

Helene Ranetbauer

Lebenslauf

Ausbildung

Juni 2008	**Reifeprüfung mit ausgezeichnetem Erfolg**, *Stiftergymnasium Linz*.
2008–2011	**Bachelorstudium Mathematik**, *Universität Wien*. Abschluss mit ausgezeichnetem Erfolg
2011-Jetzt	**Masterstudium Mathematik**, *Universität Wien*. Angewandte Mathematik und Scientific Computing

Berufserfahrung

Sommer 2007	Ferialpraktikum im Landesdienstleistungszentrum, Linz
Sommer 2009	Ferialpraktikum im Krankenhaus der Elisabethinen, Linz
Sommer 2011/2012	Mathematiknachhilfelehrerin in der Schülerhilfe, Linz
Sommer 2013	Ferialpraktikum bei der UNIQA, Wien
Zwischendurch	Nachhilfe in Mathematik

Sprachen

Deutsch	Muttersprache
Englisch	fließend in Wort und Schrift
Französisch	gute Kenntnisse