JÜLICH
Forschungszentrum

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Aachen University of Applied Sciences
Campus Jülich

Faculty: Medical Engineering and Technomathematics
Master's Program: Applied Mathematics and Computer Science

Witsenhausen's Counterexample

A Refined Approach using Variational Analysis

Master's Thesis

by

René Noffke
Matriculation Number: 3277303

Jülich, 21st August 2025

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

This master's thesis was created at the Jülich Supercomputing Centre (JSC).

Division:	Mathematics and Education
Algorithms, Tools and Methods Laboratory:	Numerical and Statistical Methods

under the supervision of Prof. Dr. Andreas Kleefeld.

This thesis was supervised by:

1st examiner:	Prof. Dr. Andreas Kleefeld
2nd examiner:	Prof. Dr. Daniel Gaigall

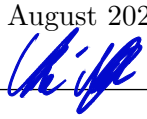
Declaration of Originality

I hereby declare that I have written the thesis with the title

Witsenhausen's Counterexample
A Refined Approach using Variational Analysis

on my own. No sources or resources other than those listed have been used.

Name: René Noffke
Jülich, 21st August 2025

Signature: 

Abstract

Witsenhausen's counterexample is a well known problem from control theory illustrating, linear controllers are not always the best choice. Studies on theoretical and numerical results have been conducted for now more than 50 years and mathematicians are still searching for new attempts gaining better controllers for the problem. The performance of these controllers is compared on a benchmark based on the problem's underlying cost functional.

In this thesis first a new method to evaluate the named cost functional was developed. Hereby the method was built as it works adaptively, requiring only as much computing capacity as is necessary. Moreover, the method includes a discontinuity detection to handle step functions which are often used for Witsenhausen's counterexample.
Next, it was shown that Witsenhausen's counterexample is a problem from variational analysis and a necessary criterion for optimality, based on the Euler-Lagrange, equation was derived. Based on this result, a basis function fulfilling the gained criterion was computed.
In the first performed optimization step, the described basis functions were combined to gain an approximation for an optimal controller.
The next optimization step was created based on the insights from previous papers indicating that adding a curve to each step improves the results.

The result on the one hand was an evaluation method computing the cost for an analytically known result in less than a second for a precision of 10^{-8}. Moreover, this method was able to determine the value up to a precision of 10^{-14}. On the other hand, the optimization yielded the fourth best value known up to now, with an absolute difference of $3.159 \cdot 10^{-5}$ to the best known.

Contents

1. Introduction, Relevance and Methodology

Control problems occur in our daily life, often without us knowing they are *Control problems*. The most obvious problem is the communication of a transmitter and a receiver using a disturbed communication medium. This happens to us every day when we use our smartphones communicating via WiFi, mobile communications, etc.

The question is: How to deal with a medium's disturbance?

To get an idea on how this question might be answered, we have a look on Figure 1.1. Here we want a mobile device to send a document M to a receiver. As already mentioned, the used communication medium is disturbed, e.g. by bad weather or other obstacles to mobile communication. To reduce the information loss during the communication, two things are done:

1. A controller C_1 is added to the transmitting device, modifying M to M_1 which should make it more resistant against influences.
2. A controller C_2 is added to the receiving device trying to obtain the original M from the noised $\hat{M}_1$.

Adding those two controllers makes it possible to increase the liability of the communication.

Figure 1.1.: Illustration of mobile communication as controller problem

Therefore, the new question is: How to choose the controllers C_1 and C_2 best for the least communication cost and least information loss?

This question belongs to a set of problems with non classical information pattern and is addressed in Witsenhausen's counterexample.

1.1. Relevance

Witsenhausen himself pointed out three areas, where the problem he stated is important. One of them is the already seen communication problem. Moreover he names controller with a very limited memory, wherefore he explains the idea of a *zero memory controller*. He also points out, his problem fits into the category of non classical control pattern, which makes the theory accessible to a large number of problems. [29, p.146]

Besides the areas Witsenhausen pointed out, newer publications from different fields show that the counterexample also fits into their topic. For example, Li and Marden in [20] presented the problem as a problem from game theory. This enabled them to achieve new results and show that the problem can be applied in even more areas. Even for more practical topics use cases were found as e.g. in [8] the problem is compared to problems from the area of multimedia security.

1.2. Methodology

The aim of this thesis is to obtain a new method, approximating the optimal controller functions for Witsenhausen's counterexample. To reach this, two major points have to be considered:

1. Obtain an efficient method to evaluate the cost functional.
2. Obtain a method optimizing functions that minimize the cost functional.

Therefore the thesis is separated into two parts.

In part I, we first formalize in Chapter 2 the illustrative problem introduced in this Chapter and gain the problem considered by Witsenhausen. Moreover, we have a look on prior theoretical results, we will use in further chapters, to realize the evaluation and minimization of the cost functional. After introducing the problem, in Chapter 3 we look on the main result Witsenhausen had to the problem stated and have a look on the first attempt, outperforming the linear solution. The next step, described in Chapter 4, is to introduce major prior attempts to the problem stated by Witsenhausen and have a closer look on some of them. After introducing known theory and results, in Chapter 5 we introduce a new method to evaluate the cost functional, which was developed as a part of the research to this thesis.

As the used optimization method is based on results from variational analysis, we start Part II with an introduction to the most important theory of variational analysis in Chapter 6. There, the theory as well as the numerical usage is demonstrated using the example of the Brachistochrone. After introducing theory, in Chapter 7 we focus on the question, why we might use variational analysis to search for an optimal solution to Witsenhausen's counterexample and determine a numerical criterion we use further. Then in Chapter 8 we develop an optimization method, based on these results, and discuss the results obtained.

Last, in Chapter 9 we have a conclusion and an outlook.

Part I.

Introduction to the Counterexample and Computation of the Cost Value

2. Introduction and theoretical Results

2.1. Introduction and Mathematization of the Problem

In this section the main ideas of the article "A counterexample in stochastic optimum control" by H. S. Witsenhausen will be summarized. Therefore, the underlying problem will be pointed out and possible simplifications will be explained. The discussion is based on the original paper [29].

The problem is based on two controllers C_1 and C_2. If the problem would fit the *classical information pattern*, these controllers would know the input value of each other. The structure of the problem, Witsenhausen pointed out, is different. As to see in Figure 2.1 controller C_1 gets the input value x_0. Each controller has a given function that will be evaluated for its input value. For C_1 the function u_1 is evaluated for x_0 and added to the input value. The output value, also named *state function*, is called x_1. The second

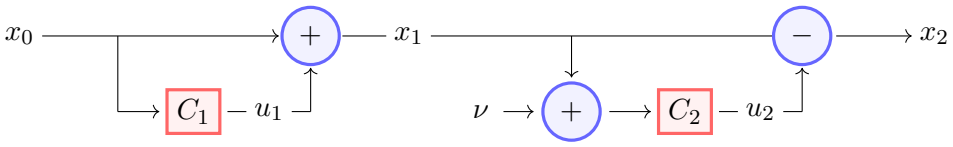


Figure 2.1.: Structure of the problem

controller C_2 does not get the value x_1. Instead, its input is disturbed by a random variable ν that can be imagined as noise added during the transport of x_1. C_2 evaluates the function u_2 for the given input and subtracts it from the input. The output or *state function* is named x_2.

As we already see in this figurative explanation, the controllers do not know the input of the other controller. Therefore, the problem cannot be described with the *classical information pattern*.

We want to formulate the problem more general. Therefore, the state functions of the controllers can be expressed with

$$x_1 = x_0 + u_1(y_0)$$
$$x_2 = x_1 - u_2(y_1).$$

Hereby, u_1 and u_2 are Borel measurable functions. Furthermore, we denote x_0 as x and the input values of the controllers with y_0 and y_1 whereby $y_0 = x$ and $y_1 = x_1 + \nu$. Hereby, x_0 and ν are two independent random variables.

The aim of the problem is to find a pair of functions (u_1, u_2) out of a set Γ that minimizes the expected squared value of the function u_1 added in C_1 and the residuum $(x+u_1-u_2)^2$ in C_2. Therefore, we get the cost function

$$k^2 u_1^2(y_0) + (x + u_1 - u_2)^2 = k^2 u_1^2(y_0) + x_2^2,$$

whereby $k \in \mathbb{R}^+ \setminus \{0\}$ is a parameter to influence the properties of the obtained result. To simplify the expected value, we define the functions

$$f(x) = x + u_1(x)$$
$$g(x) = u_2(x)$$

and can write the expected value of the cost functional as

$$J(f, g) = E\left[k^2(x - f(x))^2 + (f(x) - g(f(x) + \nu))^2\right].$$

Therefore, we get J as the term to minimize. In this case we have to optimize above f and g instead of u_1 and u_2. (cf. [29, pp. 131–132])

Remark 2.1.1

If we have a look on the function J, we know that

$$J(f, g) = \int\int (k^2 u_1^2 + x_2^2) f_x(x)\,\mathrm{d}x f_\nu(\nu)\,\mathrm{d}\nu,$$

whereby f_x is the probability density function of x and f_ν the probability density function of ν. As shifting the moments of x and ν just influences f_x and f_ν the best possible solution for f and g is just influenced in such a way that they will be shifted in the x- and y-axis and they will be rescaled. The generality is not lost. [29]

We get another simplification by assuming that $E[x] = E[\nu] = 0$ and $E[\nu^2] = 1$. As we see in Remark 2.1.1 this may be assumed without the loss of generality. (cf. [29, p. 132])

As the general problem was presented, finally, the original problem treated by Witsenhausen can be introduced in the following definition.

Definition 2.1.1

Let f, g and J be defined as before and

$$x \sim \mathcal{N}\left(0, \sigma^2\right), \quad \nu \sim \mathcal{N}(0, 1).$$

Then the problem of minimizing the functional J for f and g is denoted by $\pi(k^2, \sigma^2)$. (cf. [29, p. 132])

2.2. Theoretical Results

Before the main result, the "counterexample" is induced, a few theoretical details will be shown. These results are needed to understand the example and are used in many papers that followed on the original from Witsenhausen.

2.2.1. Existence of an optimal Solution

For the existence of an optimal solution Witsenhausen gives the following lemma.

Lemma 2.2.1

Let the variables be defined as in the previous section, then

a) the optimal result J^* is defined as $J^* = \inf\{J(f,g)|(f,g) \in \Gamma\}$ and it is valid that $0 \leq J^* \leq \min(1, k^2\sigma^2)$

b) $\forall (f,g) \in \Gamma\ \exists (f^*, g^*) : J(f^*, g^*) \leq J(f,g)$ whereby $E[f^*(x)] = 0$, $E[(x - f^*(x))] \leq \sigma^2$ and $E[(f^*(x))^2] \leq 4\sigma^2$

Proof:

a) First, we show $J^* \geq 0$. As we can write J as

$$J = E\left[a(x)^2 + b(x,\nu)^2\right]$$

and the expected value of a nonnegative function cannot be negative, the assertion is valid. Secondly, we show $J^* \leq \min(1, k^2\sigma^2)$. We assume

$$f_1(x) = g_1(x) = 0\,,$$

therefore, we get

$$\begin{aligned} J(f_1, g_1) = E\left[k^2x^2\right] &= k^2\left(E[x^2] - E[x]^2 + E[x]^2\right) \\ &= k^2(\mathrm{Var}(x) + E[x]^2) = k^2\sigma^2. \end{aligned}$$

By choosing the functions with $f_2(x) = g_2(x) = x$, we obtain another value with

$$J(f_2, g_2) = E[\nu^2] = 1.$$

As we have two different values for J we can get the upper bound by $J^* \leq \min(1, k^2\sigma^2)$.

b) The second part of the proof will be done using a case distinction. First, we choose $(f,g) \in \Gamma$ in a way that $E\left[(x - f(x))^2\right] > \sigma^2$. Since $J(f,g) \geq k^2 E\left[(x - f(x))^2\right]$ we get $J(f,g) > k^2\sigma^2$. Assuming this, after a), we can choose the functions with

$$f^*(x) = g^*(x) = 0$$

and thereby improve the solution found.

On the other hand, we have to consider $E\left[(x - f(x))^2\right] \leq \sigma^2$. In this case, we see

$$\begin{aligned} E\left[(x - f(x))^2\right] &= E\left[x^2\right] - 2E\left[xf(x)\right] + E\left[f^2(x)\right] \\ &= \sigma^2 - 2E\left[xf(x)\right] + E\left[f^2(x)\right] \leq \sigma^2\,, \end{aligned}$$

which results in

$$E\left[f^2(x)\right] \leq 2E\left[xf(x)\right].$$

If $E[f^2(x)] \leq 4\sigma^2$ should be valid it must be true that $E\left[xf(x)\right] \leq 2\sigma^2$. Therefore it must be valid that $f(x) \leq 2x$. Assuming the opposite, $f(x) > 2x$, we get

$$\begin{aligned} E\left[(x - f(x))^2\right] &> E\left[(x - 2x)^2\right] \\ &= E\left[x^2\right] = \sigma^2. \end{aligned}$$

This is a contradiction and the assumption that $E[f^2(x)] \leq 4\sigma^2$ must be true.

As the second moment $E[f^2(x)]$ exists also the first moment $E[f(x)]$ must exist. We define $m = E[f(x)]$, $f_1(x) = f(x) - m$ and $g_1(x) = g(x + m) - m$. By using the linearity of expected values we get $E[f_1(x)] = 0$. As $E[x] = 0$ we get

$$\begin{aligned} E[(x - f_1(x))^2] &= E[x^2] - 2E[xf_1(x)] + E[f_1^2(x)] \\ &= E[x^2 - 2xf(x) + f^2(x)] + 2E\left[x\right]E\left[f(x)\right] \\ &\quad - 2E\left[f(x)\right]E\left[f(x)\right] + E\left[f(x)\right]^2 \\ &= E[(x - f(x))^2] - E\left[f(x)\right]^2 \\ &= E[(x - f(x))^2] - m^2 \end{aligned}$$

Since $E[(x - f(x))^2] \leq \sigma^2$ and m^2 is a nonnegative value, also $E[(x - f_1(x))^2] \leq \sigma^2$. Determining the difference of $J(f_1, g_1)$ and $J(f, g)$, we get

$$\begin{aligned} J(f_1, g_1) - J(f, g) &= E\left[2k^2 xE\left[f\left(x\right)\right] - 2k^2 f\left(x\right)E\left[f\left(x\right)\right] + k^2 E\left[f\left(x\right)\right]^2\right] \\ &= 2k^2 E\left[f\left(x\right)\right]E\left[x\right] - k^2 E\left[f\left(x\right)\right]^2 \\ &= -k^2 m^2 \end{aligned}$$

and therefore

$$J(f_1, g_1) = J(f, g) - k^2 m^2.$$

This shows that it is possible to get a better result in both considered cases. Therefore a optimal solution must exist. [29] ■

As (f^*, g^*) just differs from (f, g) if $E\left[f(x)\right] \neq 0$ the optimal solution for f must have $E\left[f^*(x)\right] = 0$. Since $E\left[(f^*(x))^2\right] \leq 4\sigma^2$, all functions not fulfilling this equation can be ignored for the search of the minimum of J. (cf. [29, pp. 132–133])

2.2.2. Determining the optimal g^* for fixed f

Witsenhausen showed there is a way to determine the optimal g^* if the optimal function f^* is known. Knowing this, the search for a global minimum can be reduced to a problem in just one function.

The result of this idea is given in the following lemma.

Lemma 2.2.2

Let f be fixed. Then the g^* minimizing the expected value of the cost function is given by

$$g^*(y) = \frac{\int f(x) f_{x,y}(x,y)\,\mathrm{d}x}{f_y(y)}, \tag{2.1}$$

whereby f_y is the density function of $y = f(x) + \nu$ and $f_{x,y}$ the joint density function of x and y.

Proof: Assuming we know a way to determine g^* depending on f, we get an expression for J just depending on f, with

$$\begin{aligned} J(f) &= E\left[k^2(x - f(x))^2 + (f(x) - g^*(f(x)+\nu))^2\right] \\ &= E\left[k^2(x - f(x))^2\right] + E\left[(f(x) - g^*(f(x)+\nu))^2\right]. \end{aligned}$$

As the first summand is independent of g, we may disregard it when optimizing for g. Therefore, we can express the optimal g^* as

$$g^*(y) = \underset{g^*}{\operatorname{argmin}}\ E\left[(f(x) - g^*(y))^2 \mid f(x) + \nu = y\right]. \tag{2.2}$$

It may be seen, the term to minimize corresponds to the mean squared error of two random variables. The minimizer for such a term is well known from literature and given by

$$g^*(y) = E\left[f(x) \mid f(x) + \nu = y\right] \tag{2.3}$$

(cf. [25]). As the conditional expected value of two random variables X and Y is defined as

$$E\left[X|Y = y\right] = \int x f_{X|Y}(x)\,\mathrm{d}x$$

whereby the conditional density function is given with

$$f_{X|Y}(x) = \frac{f_{X,Y}(x,y)}{f_Y(y)}, \quad f_Y(y) > 0\,.$$

Hereby, $f_{X,Y}$ means the joint density function of X and Y and f_Y the marginal density function of Y. Applying this to Equation (2.3), we get

$$g^*(y) = \frac{\int f(x) f_{x,y}(x,y)\,\mathrm{d}x}{f_y(y)}.$$

(cf. [29, pp. 133–134]) ■

2.2.3. Computation Rule of the Second Part of the Cost Functional

If we know the g is derived that way and therefore is optimal for the given f, the following Lemma provides information about the expected value of the cost function.

Lemma 2.2.3

Let g_f^* be chosen as in Lemma 2.2.2. Then

$$J(f, g_f^*) - k^2 E\left[(x - f(x))^2\right] = E\left[(f(x) - g_f^*(y))^2\right] = E[f^2(x)] - E[g_f^{*2}(y)].$$

Proof: We know the optimal g depending on a given f with

$$g_f^* = E\left[f(x) \mid y = f(x) + \nu\right].$$

This results in

$$\begin{aligned} E\left[(f(x) - g_f^*(y))^2\right] &= E\left[\ (f(x) - E\left[f(x) \mid y = f(x) + \nu\right])^2\ \right] \\ &= E\left[\ \mathrm{Var}(f(x) \mid y = f(x) + \nu)\ \right]. \end{aligned}$$

We know that

$$\mathrm{Var}(X|Y = y) = E\left[X^2|Y = y\right] - E\left[X|Y = y\right]^2$$

[25, Section 5.1.5]. Therefore we get

$$\begin{aligned} E\left[(f(x) - g_f^*(y))^2\right] &= E\left[\ \mathrm{Var}(f(x) \mid y = f(x) + \nu)\ \right] \\ &= E\left[E\left[f^2(x) \mid y = f(x) + \nu\right] - E\left[f(x) \mid y = f(x) + \nu\right]^2\right]. \end{aligned}$$

As $E[X] = E\left[E[X|Y]\right]$ [25, Section 5.1.5] and $g_f^* = E\left[f(x) \mid y = f(x) + \nu\right]$ this can be simplified to the expected result

$$\begin{aligned} E\left[(f(x) - g_f^*(y))^2\right] &= E\left[E\left[f^2(x) \mid y = f(x) + \nu\right] - g_f^{*2}(y))\right] \\ &= E\left[f^2(x)\right] - E\left[g_f^{*2}(y)\right]. \end{aligned}$$

(cf. [29]) ■

Remark 2.2.1

As we know the distribution of x and ν, we can write down the marginal density function of y as well as the joint density function of x and $y = f(x) + \nu$. We get

$$\begin{aligned} f_y(y) &= \int f_\nu(y - f(x)) f_x(x) \mathrm{d}x, \\ f_{x,y}(x, y) &= f_x(x) f_\nu(y - f(x)), \end{aligned}$$

whereby f_x is the density function of x and f_ν is the density function of ν.

Knowing those facts, we may determine another simpler formulation for the second part of the cost functional.

Lemma 2.2.4

It is valid that

$$\begin{aligned} J_2(f) - k^2 E[(x - f(x))^2] &= E[f^2(x)] - E[g_f^{*2}(y)] \\ &= 1 - I(D_f) \end{aligned}$$

whereby $I(D_f)$ means the Fisher information of y that is given by

$$I(D_f) = 4 \int \left(\frac{\mathrm{d}}{\mathrm{d}y} \sqrt{D_f(y)} \right)^2 \mathrm{d}y$$

with

$$D_f(y) = \int f_\nu(y - f(x)) f_x(x) \, \mathrm{d}x \, .$$

[29]

Proof: We know from Lemma 2.2.3

$$E[f^2(x)] - E[g_f^{*2}(y)] = E[(f(X) - E[f(x)|y])^2] = \mathrm{MMSE}(f(x)|y) \, .$$

As explained in [7, p. 3] for a random variable $Y = \sqrt{\mathrm{snr}} X + Z$, whereby X is an arbitrary random variable, Z has standard normal distribution, Y has pdf f_Y and snr is the signal to noise ratio, we know

$$I(f_Y) = 1 - \mathrm{snr} \cdot \mathrm{MMSE}(X|Y) \, .$$

As we may choose an arbitrary random variable X, we choose a scaling factor as snr becomes 1. This allows us to choose

$$Y = f(x) + \nu \, ,$$

which leads to

$$I(f_y) = 1 - \mathrm{MMSE}(f(x)|y) \, .$$

As D_f is the density of y [29], we have

$$\mathrm{MMSE}(f(x)|y) = 1 - I(D_f) \, .$$

The explicit computation of $I(D_f)$ is left out. (cf. [29]) ■

2.2.4. Monotonicity of the optimal Function

An important result is given in the monotony of the optimal function. Therefore, we will discuss two major results from history. The first was already discussed in the original paper [29] from Witsenhausen, the second more than 30 years later in the paper [31] from Vu and Verdú. As a full presentation of the corresponding proofs would require lengthy derivations, we omit them here and focus on the main insights.

To introduce the Lemma pointed out by Witsenhausen, we first have to introduce a concept, Witsenhausen also used in his publication [29, p. 135].

Definition 2.2.1

Let P be the distribution of a real valued random variable. Then, let $\alpha(P)$ be the smallest convex set wherefore

$$P(\alpha(P)) = 1 \, .$$

Knowing this, we may introduce the result Witsenhausen gained in [29, p. 137].

Lemma 2.2.5

Let F be the probability distribution of the random variable x and $\alpha(F)$ as defined before. Then for $E\left[(f_0(x)-x)^2\right] \leq \sigma^2$ and g_f^* the optimal function g for f there is a function f^* that is monotonically non-decreasing on $\alpha(F)$ with

$$J(f^*, g_f^*) = \min\{J(f,g) \mid f \text{ Borel }\}.$$

This already gives insights on the monotony of the optimal f function. Years later, the question about the monotony was asked again and yields the Theorem from [31, p. 5735], where Vu and Verdú formulated the problem stated by Witsenhausen as problem from optimal transport theory.

Theorem 2.2.1

For a probability measure with real analytic strictly positive density, any optimal controller f is a strictly increasing unbounded piece wise real analytic function with a real analytic left inverse. (cf. [31])

A few more details to the theorem are added in the following Remark.

Remark 2.2.2

Real analytic left inverse combines two properties of a function.

- *real analytic: For $I \subset \mathbb{R}$ an open set, a function $f : I \to \mathbb{R}$ is called real analytic if for all $x_0 \in I$ there exists J such as there exists a series wherefore*

  $$f(x) = \sum_{n\in\mathbb{N}} a_n (x - x_0)^n \quad \forall x \in J$$

  is true. [13]
- *left inverse: For functions h and f we call for*

  $$h \circ f = \mathrm{id}$$

  h the left inverse of f. [31]

3. Witsenhausen's Counterexample

For problems based on the classical information pattern, it can be assumed that the optimal solution can be found in the set of affine functions. Witsenhausen showed, this does not apply for non classical information pattern (cf. [29, p. 131]). In this section first the optimal affine solution will be derived, then an example for a not affine solution will be given, that improves the value of J compared to the affine solution.

3.1. Deriving the best affine Solution

Solving the optimization problem $\pi(k^2, \sigma^2)$ for f, g affine, Witsenhausen came to the following results.

Lemma 3.1.1

Solving the problem $\pi(k^2, \sigma^2)$ over the affine class, thus searching

$$J_a^* = \inf \{J(f, g) | f, g \text{ affine}\},$$

results in

a) the optimal affine controller function f_a^*, with

$$f_a^*(x) = \lambda x, \quad \lambda \in \mathbb{R},$$

b) the optimal affine controller function g_a^* depending on f_a^*, with

$$g_a^*(y) = g_{f_a}^*(y) = \frac{\sigma^2 \lambda^2}{1 + \sigma^2 \lambda^2} y,$$

c) the expexted cost function value

$$J(f_a^*, g_a^*) = k^2 \sigma^2 (1 - \lambda)^2 + \frac{\lambda^2 \sigma^2}{1 + \lambda^2 \sigma^2}.$$

Proof:

a) Since f must be affine, we know it is of the form

$$f(x) = c_1 + c_2 x.$$

As known from chapter 2.2, it must be valid that $E[f(x)] = 0$. Therefore, $c_1 = 0$. Denoting c_2 as λ, we get the statement.

b) We already derived that for f given, the optimal g is given by

$$g_f^*(y) = \frac{\int f(x) f_{x,y}(x,y)\,\mathrm{d}x}{f_y(y)}.$$

Solving the integral and setting $g_a^*(x) = g_f^*(x)$ leads to the statement.

c) J is given by

$$J(f,g) = E\left[k^2(x - f(x))^2 + (f(x) - g_a^*(f(x) + \nu))^2\right].$$

Inserting f_a^* and g_a^*, we get

$$\begin{aligned} J(f_a^*, g_a^*) &= k^2 E\left[x^2(1-\lambda)^2\right] + \frac{\lambda^2}{(\lambda^2\sigma^2+1)^2} E\left[(\lambda\nu\sigma^2 - x)^2\right] \\ &= k^2(1-\lambda)^2 E\left[x^2\right] + \frac{\lambda^2}{(\lambda^2\sigma^2+1)^2}\left(\lambda^2\sigma^4 E\left[\nu^2\right] - 2\lambda\sigma^2 E\left[x\nu\right] + E\left[x^2\right]\right). \end{aligned}$$

Since x and ν are independent, it is valid that $E[x\nu] = E[x]E[\nu]$. Moreover, $E[x] = E[\nu] = 0$, $E[x^2] = \sigma^2$ and $E[\nu^2] = 1$. Therefore, the expression simplifies to

$$\begin{aligned} J(f_a^*, g_a^*) &= k^2\sigma^2(1-\lambda)^2 + \frac{\lambda^2}{(\lambda^2\sigma^2+1)^2}\left(\lambda^2\sigma^4 + \sigma^2\right) \\ &= k^2\sigma^2(1-\lambda)^2 + \frac{\lambda^2\sigma^2\left(\lambda^2\sigma^2+1\right)}{(\lambda^2\sigma^2+1)^2} \\ &= k^2\sigma^2(1-\lambda)^2 + \frac{\lambda^2\sigma^2}{1+\lambda^2\sigma^2}. \end{aligned}$$

■

(cf. [29, pp. 140–141])

As the form of the optimal equations is known and g_a^* can be determined for a given f, the optimal f_a^* has to be found. f_a just depends on $\lambda \in \mathbb{R}$. Therefore, we get an optimization problem in λ. For the optimal λ, Witsenhausen obtained the following results.

Lemma 3.1.2

If $t = \sigma\lambda$, then t must be a real root value of

$$(t - \sigma)(1 + t^2)^2 + \frac{1}{k^2}t = 0. \tag{3.1}$$

Proof: As f_a can just be varied by λ, J behaves like a function with $J : \mathbb{R} \to \mathbb{R}$. Denoting $J(f_a, g_a)$ as $J(\lambda)$, for a minimum

$$\frac{\mathrm{d}}{\mathrm{d}\lambda} J(\lambda) = 0$$

must be valid. This leads to

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\lambda}J(\lambda) &= -2k^2\sigma^2\left(1-\lambda\right) + \frac{2\lambda\sigma^2}{1+\lambda^2\sigma^2} - \frac{2\lambda^3\sigma^4}{\left(1+\lambda^2\sigma^2\right)^2} \\ &= -2k^2\sigma(\sigma-t) + \frac{2t\sigma}{1+t^2} - \frac{2t^3\sigma}{(1+t^2)^2} \\ &= 2k^2\sigma(t-\sigma) + 2\sigma\left(\frac{t}{1+t^2} - \frac{t^3}{(1+t^2)^2}\right).\end{aligned}$$

As $\mathrm{d}J/\mathrm{d}\lambda \stackrel{!}{=} 0$, we get

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\lambda}J(\lambda) \stackrel{!}{=} 0 \quad &\Leftrightarrow \quad 0 = 2k^2\sigma(t-\sigma) + 2\sigma\left(\frac{t}{1+t^2} - \frac{t^3}{\left(1+t^2\right)^2}\right) \\ &\Leftrightarrow \quad 0 = k^2(t-\sigma) + \left(\frac{t\left(1+t^2\right) - t^3}{\left(1+t^2\right)^2}\right) \\ &\Leftrightarrow \quad 0 = k^2(t-\sigma)\left(1+t^2\right)^2 + t \\ &\Leftrightarrow \quad 0 = (t-\sigma)\left(1+t^2\right)^2 + \frac{t}{k^2},\end{aligned}$$

which shows that the statement is true. (cf. [29, p. 141]) ■

Remark 3.1.1

Equation (3.1) can be interpreted more intuitively. Instead of solving

$$0 = (t-\sigma)\left(1+t^2\right)^2 + \frac{t}{k^2},$$

we can also search the intersection points of the curve k and the line l, with

$$k(t) = \frac{t}{\left(1+t^2\right)^2}, \quad l(t) = k^2\left(\sigma - t\right),$$

as

$$\begin{aligned}k(t) = l(t) \quad &\Leftrightarrow \quad \frac{t}{\left(1+t^2\right)^2} = k^2\left(\sigma - t\right) \\ &\Leftrightarrow \quad (t-\sigma)\left(1+t^2\right)^2 + \frac{t}{k^2} = 0.\end{aligned}$$

Using this insight, the following results can be gained easily. (cf. [29, p. 141])

The last question to answer is whether solving (3.1) yields a unique value for λ, and thus a unique minimum for J. Therefore, we state the following lemma without proof, as it follows from standard results in real analysis.

Lemma 3.1.3

Let h be a sufficiently smooth function with

$$m_h = \max\left\{\left|\frac{\mathrm{d}}{\mathrm{d}x}h(x)\right| \,\middle|\, x \in \mathbb{R}\right\},$$

and exactly one extremum, k a line with

$$m_k = \frac{\mathrm{d}}{\mathrm{d}x}k(x)$$

and

$$|m_k| > m_h.$$

Then there is exactly one intersection point between h and k. [29]

As σ and k^2 are always positive, solutions of (3.1) lead to positive t. Looking at $t > 0$ we get the following lemma.

Lemma 3.1.4

Solving $\pi(k^2, \sigma^2)$ over the class of affine functions we get
a) a unique solution for $k^2 \geq \frac{1}{4}$,
b) two equivalent solutions for $k^2 < \frac{1}{4}$ and $\sigma = \sigma_c = k^{-1}$.

Proof:

a) Using the derivative

$$\frac{\mathrm{d}}{\mathrm{d}t}k(t) = -\frac{3t^2 - 1}{(t^2 + 1)^3},$$

we get that, for positive t, k has its maximum at $\sqrt{3}/3$ and from thereon decays to zero. As the maximum absolute slope can be found at $t = 1$ with $1/4$ and the deviation of k is not constant, the statement follows according to Lemma 3.1.3.

b) Assuming $k^2\sigma = 1$ and therefore $1/k^2 = \sigma^2$, we can write the condition $(t-\sigma)(1+t^2)^2+t/k^2$ as

$$(t^2 - \sigma t + 1)(t^3 + t - \sigma) = 0.$$

The first factor leads to the roots

$$t_0, t_1 = \frac{1}{2}\sigma \pm \sqrt{\frac{\sigma^2}{4} - 1}.$$

And the second yields the real root

$$t_2 = \frac{\left(108\sigma + 12\sqrt{81\sigma^2 + 12}\right)^{\frac{2}{3}} - 12}{6\left(108\sigma + 12\sqrt{81\sigma^2 + 12}\right)^{\frac{1}{3}}}.$$

As $k^2 > 1/4$ and $k^2\sigma^2 = 1$ it must be valid that $|\sigma| > 2$. Solving $\mathrm{d}^2/\mathrm{d}t^2 J(t_i) = 0$ for σ, we get ± 2 as real roots. Testing for $|\sigma| > 2$, we get

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}J(t_0) > 0, \quad \frac{\mathrm{d}^2}{\mathrm{d}t^2}J(t_1) > 0, \quad \frac{\mathrm{d}^2}{\mathrm{d}t^2}J(t_2) < 0.$$

As $J(t_0) = J(t_1)$ for $k^2\sigma^2 = 1$, the optimal solutions are found with t_0 and t_1, wherefore we know $\sigma_c = k^{-1}$ is the critical value.

(cf. [29]) ■

3.2. Two-point distributed Variables

As an intermediate step to come to the final result of Witsenhausen's paper, we assume x would have an easier distribution than before.

Definition 3.2.1

In this subsection the variable x is defined as a two-point distributed random variable, with

$$f_x(\sigma) = f_x(-\sigma) = \frac{1}{2}.$$

Simplifying the distribution leads to an easier way to determine the functions f and g and the following Lemma.

Lemma 3.2.1

Let $f(\sigma) = a$, $a \in \mathbb{R}$, then
a) $f(x) = a\,\mathrm{sgn}(x)$
b) the optimal g for f is given with

$$g_f^*(y) = a\tanh(ay)\,.$$

Proof:

a) As shown in Lemma 2.2.1 it must be valid that $E[f(x)] = 0$. Therefore, $f(\sigma) = f(-\sigma)$.

b) As already discussed in Lemma 2.2.2 the optimal g, depending on f is given by

$$g_f^*(y) = E\left[f(x) \mid f(x) + \nu = y\right].$$

As we are in the discrete case for x and $(\mathrm{e}^x + \mathrm{e}^{-x})/2 = \cosh(x)$ [23, p. 107], we get

$$\begin{aligned}
f_y(y) &= \frac{1}{2}\left(f_\nu\left(y-a\right) + f_\nu\left(y+a\right)\right) \\
&= \frac{1}{2\sqrt{2\pi}}\left(\exp\left(\frac{-y^2-a^2}{2}\right)\exp\left(ay\right) + \exp\left(\frac{-y^2-a^2}{2}\right)\exp\left(-ay\right)\right) \\
&= \frac{1}{\sqrt{2\pi}}\exp\left(\frac{-y^2-a^2}{2}\right)\frac{\exp\left(ay\right)+\exp\left(-ay\right)}{2} \\
&= 2\pi\frac{1}{\sqrt{2\pi}}\exp\left(\frac{-y^2}{2}\right)\frac{1}{\sqrt{2\pi}}\exp\left(\frac{-a^2}{2}\right)\cosh(ay) \\
&= 2\pi f_\nu(y) f_\nu(a)\cosh(ay)
\end{aligned}$$

as the denominator. The nominator of g_f^* can be obtained with

$$\begin{aligned} N_g(y) &= \frac{a}{2} f_\nu(y-a) - \frac{a}{2} f_\nu(y+a) \\ &= \frac{a}{2\sqrt{2\pi}} \left(\exp\left(\frac{-y^2-a^2}{2}\right) \exp(ay) - \exp(ay) + \exp\left(\frac{-y^2-a^2}{2}\right) \exp(-ay) \right) \\ &= 2a\pi \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-y^2}{2}\right) \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-a^2}{2}\right) \left(\frac{\exp(ay) - \exp(-ay)}{2}\right) \\ &= 2a\pi f_\nu(y) f_\nu(a) \sinh(ay). \end{aligned}$$

Therefore, with $\tanh(x) = \sinh(x)/\cosh(x)$ [23, p. 108], we obtain the optimal g, with

$$g_f^*(y) = \frac{N_g^*(y)}{f_y(y)} = a \tanh(ay).$$

(cf. [29]) ■

The expected value of the cost functional can easily be derived as shown in the next Lemma.

Lemma 3.2.2

Let f and g be chosen as described in Lemma 3.2.1, then J is given with

$$J(f) = k^2(a-\sigma)^2 + h(a),$$

whereby

$$h(a) = \sqrt{2\pi}\ a^2 f_\nu(a) \int \frac{f_\nu(y)}{\cosh(ay)}\, \mathrm{d}y.$$

Proof: Figuring out the expected value of the cost function, we get

$$J(f,g) = E\left[k^2(x - a\ \mathrm{sgn}(x))^2\right] + E\left[\left(a\ \mathrm{sgn}(x) - a\tanh\left(a(a\ \mathrm{sgn}(x) + \nu)\right)\right)^2\right]$$

The first summand is easy to determine as the distribution of x is just a two-point distribution. This restricts the values of x to the set $\{-\sigma, \sigma\}$. Therefore, we obtain

$$\begin{aligned} E\left[k^2(x - a\ \mathrm{sgn}(x))^2\right] &= \frac{1}{2}\left(k^2\left(\sigma - a\ \mathrm{sgn}(\sigma)\right)^2 + k^2\left(-\sigma - a\ \mathrm{sgn}(-\sigma)\right)^2\right) \\ &= k^2\left(a-\sigma\right)^2. \end{aligned}$$

The second part is more complicated as there is not just a two-point distribution but also a normal distribution for ν. We know

$$\begin{aligned} f^2(x) &= a^2, \\ g_f^{*2}(x) &= a^2\ \tanh^2(ay) \\ &= a^2 - a^2\ \mathrm{sech}^2(ay). \end{aligned}$$

Moreover, we can determine the probability density function of y with

$$\begin{aligned} f_y(y) &= \frac{1}{2}\left(f_\nu(y-a) + f_\nu(y+a)\right) \\ &= \frac{1}{2\sqrt{2\pi}} \exp\left(-\frac{a^2}{2}\right) \exp\left(-\frac{y^2}{2}\right) \left(\exp(ay) + \exp(-ay)\right) \\ &= \sqrt{2\pi}\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{a^2}{2}\right) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right) \frac{\exp(ay) + \exp(-ay)}{2} \\ &= \sqrt{2\pi} f_\nu(a) f_\nu(y) \cosh(ay). \end{aligned}$$

Determining the expected values of the squared functions we get for f

$$E\left[f^2(x)\right] = a^2$$

and for g_f^*

$$\begin{aligned} E\left[g_f^{*2}\right] &= E\left[a^2 - a^2 \operatorname{sech}^2(ay)\right] \\ &= a^2 - \int f_y(y) a^2 \operatorname{sech}^2(ay)\, \mathrm{d}y. \end{aligned}$$

As $\operatorname{sech}(x) = 1/\cosh(x)$, this results in

$$E\left[g_f^{*2}\right] = a^2 - \sqrt{2\pi}\, a^2 f_\nu(a) \int \frac{f_\nu(y)}{\cosh(ay)}\, \mathrm{d}y$$

and will be written as

$$E\left[g_f^{*2}\right] = a^2 - h(a),$$

whereby

$$h(a) = \sqrt{2\pi}\, a^2 f_\nu(a) \int \frac{f_\nu(y)}{\cosh(ay)}\, \mathrm{d}y.$$

As we know, g is chosen optimal for the given f, we get

$$J(f,g) = k^2(a-\sigma)^2 + E\left[f^2(x)\right] - E\left[g^2(x)\right]$$

from Lemma 2.2.3 and therefore

$$J(f,g) = k^2(a-\sigma)^2 + h(a).$$

(cf. [29]) ■

Lemma 3.2.3

Let h be as before, then there is an upper bound given with

$$h(a) \leq \sqrt{2\pi} a^2 f_\nu(a) = a^2 \exp\left(-\frac{a^2}{2}\right).$$

Proof: h is given by

$$h(a) = \sqrt{2\pi}\ a^2 f_\nu(a) \int \frac{f_\nu(y)}{\cosh(ay)}\,\mathrm{d}y.$$

As

$$\int \frac{f_\nu(y)}{\cosh(ay)}\,\mathrm{d}y \leq \int f_\nu(y)\,\mathrm{d}y = 1\,,$$

the statement is valid. (cf. [29]) ■

This design idea will be considered to obtain the final result in the next section.

3.3. Witsenhausen's Counterexample

Leaving the simplification behind and assuming that x is normal distributed, we come to the *Witsenhausen Counterexample*, which is the main result in [29].

Theorem 3.3.1

There are parameters σ and k wherefore the optimal solution J^* of the problem $\pi(k^2, \sigma^2)$ is less than the optimal affine solution J_a^*.

Proof: Considering the design idea from Lemma 3.2.1, we choose

$$f(x) = a\ \mathrm{sgn}(x), \quad g(y) = a\tanh(ay)$$

and set $a = \sigma$. Since f is two-point distributed again, determining J for this choice of functions, we get

$$\begin{aligned} J(f,g) &= E\left[k^2(x - f(x)^2\right] + E\left[\left(f(x) - g(f(x)+\nu)\right)^2\right] \\ &= k^2 E\left[(x - \sigma\ \mathrm{sgn}(x))^2\right] + h(\sigma). \end{aligned}$$

Evaluating the first part we get

$$\begin{aligned} k^2 E\left[(x - \sigma\ \mathrm{sgn}(x))^2\right] &= k^2 E\left[x^2 - 2x\sigma\mathrm{sgn}(x) + \sigma^2\right] \\ &= k^2\left(\sigma^2 - 2E\left[\sigma|x|\ \right] + \sigma^2\right) \\ &= 2k^2\sigma^2\left(1 - E\left[\left|\frac{x}{\sigma}\right|\right]\right)\,. \end{aligned}$$

As the limit of the integral in the expexted value is known as $\sqrt{2/\pi}$, the expression can be determined. For $k^2\sigma^2 = 1$ together with Lemma 3.2.3, we get

$$J(f,g) \leq 2\left(1 - \sqrt{\frac{2}{\pi}}\right) + \frac{1}{k^2} f_\nu\left(\frac{1}{k}\right)\,.$$

For $k \to 0$ this tends to $2\left(1 - \sqrt{2/\pi}\right) \approx 0.4042308783943$. Using Lemma 3.1.1, we get for the affine solution

$$J_a^*(f,g) = k^2\sigma^2(1-\lambda)^2 + \frac{\lambda^2\sigma^2}{1 + \lambda^2\sigma^2}.$$

As $k^2\sigma^2 = 1$, we can use Lemma 3.1.4 to determine the optimal λ with

$$\lambda = \frac{1}{2} \pm \sqrt{\frac{1}{4} - k^2}.$$

As both λ lead to the same result, we may choose one arbitrary and get

$$J_a^* = 1 - k^2.$$

For $k \to 0$ this results in 1. Therefore, $J^* < J_a^*$ which shows the statement is valid. (cf. [29]) ■

4. Historical Results

In the last decades, various people worked on the problem, stated by Witsenhausen. They have published theoretical results on the properties of the optimal control function, as well as numerical attempts, gaining such an optimal function. In this Chapter, we will focus on the second case and focus on a selection out of the numerical attempts. As we cannot consider all results, we try to focus on a couple of main results.

Before we start to select results, we have a look on the most relevant results from the past five decades. This results are listed in Table 4.1. The table is gained as a combination of the results summarized in [20, p. 2] and [28, p. 5017]. In this chapter, we just focus on results for the benchmark configuration $\sigma = 5$, $k = 0.2$.

Year	Idea	Author	J
1968	Affine solution	Witsenhausen [29]	0.961852
1968	1-step function	Witsenhausen [29]	0.404253
1987	1-step function	Bansal & Basar [5]	0.365015
1999	2-step function	Deng & Ho [1]	0.19
2000	25-step function	Ho & Lee [15]	0.1717
2001	2.5-step function	Baglietto et al. [4]	0.1701
2001	3.5-step function	Lee et al. [19]	0.1673132
2009	3.5-step function	Li et al. [20]	0.1670790
2011	Sloped 4-step function	Karlsson et al. [17]	0.16692462
2014	Sloped 5-step function	Mehmetoglu et al. [21]	0.16692291
2017	Curved step function	Tseng & Tang [28]	0.166897

Table 4.1.: Major results regarding found f for Witsenhausen's Counterexample

As to see in Table 4.1 there are three major attempts for the function type chosen for f:

1. Until 2001, the functions are chosen as n-step functions and the step positions and heights were optimized.
2. From 2001 to 2009, the functions are chosen as $n.5$-step function, which means the first step is not positioned in $x = 0$. Such a function is shown in Figure 4.1.
3. From 2011 to 2014, sloped step functions were used to find an optimal f for the given problem. Which adds a slope to each step of a step function.

4. The function determined in 2017, which is the best currently known, is given as a point representation, approximating a curved step function.

Now, as we found four different classes, we can group the optimized functions in, we want to focus on the methods used, to gain them. Since we cannot introduce all methods used and results gained, we will focus on one approach from each function class. Therefore, we start with the approach pointed out by Deng and Ho in 1999, continue with the idea of Baglietto et al., then introduce the method of Karlsson et al. and in the end present the results of Tseng and Tang.

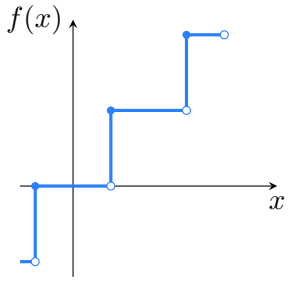


Figure 4.1.: 2.5-step function

4.1. Gaining a 2-step Function: Results from Deng and Ho [1]

In their publication, "An ordinal optimization approach to optimal control problems," Deng and Ho 1999 introduced a new method based on ordinal optimization to obtain a new controller function f for the problem stated by Witsenhausen. For the common benchmark $k = 0.2, \sigma = 5$ they reached a cost value which is more then 50% better than the best value known before. [1]

The strategy Deng and Ho use, is build on the main idea of ordinal optimization, which is a strategy to speed up stochastic optimizations by considering two main ideas:

1. It is easier to obtain an order than a value.
2. It is easier to find *good enough with high probability* than *best for sure*.

These main ideas should be used for problems where the design space Θ for an optimization problem is very large. In such cases, the number of combinations to be considered becomes too huge to compute all of them. For this reason, they want to search for subspaces $\Theta_1, \Theta_2, \cdots \subset \Theta$ and determine up to a high probability, which one includes the top-k combinations. [1]

As indicator for such a decision, they introduce the *Performance Density Function* and the *Performance Distribution Function* (PDF). The *Performance Density Function* for a design space Θ and the cost function J_Θ is the histogram gained by evaluating

$$J(\theta), \quad \forall \theta \in \Theta\,.$$

The PDF is the integral of the *Performance Density Function*. For a chosen number of samples N, they found rules to guarantee the approximated PDF is near the true PDF.[1]

For two subsets, $\Theta_1, \Theta_2 \subset \Theta$ choosing one is now done by just comparing the approximated PDFs. Concerning the case where the cost functional J_Θ might not be evaluated without an inaccuracy, Deng and Ho showed under a few extra conditions that the probability choosing the better subspace based on the approximated PDF is > 0.5. [1]

Applying the method explained on the problem stated by Witsenhausen, they first assume the optimal function f is odd what they justify with the property $E[f(x)] = 0$ for the optimal f. Then they use the method by choosing key parameters to split the design space into subspaces. First, they choose the number of intervals to obtain function values for, which is equivalent to choosing the number of steps. The PDFs gained for $n = 1, 2, 5, 10$ intervals is shown in Figure 4.2. This lead to the decision for a 2-step function, as two intervals seem to have combinations reaching lower cost values than any other combination.

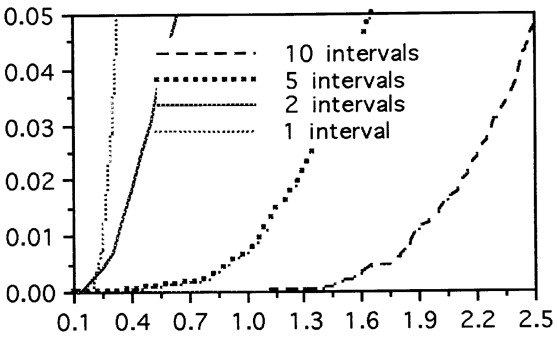


Figure 4.2.: PDFs reached for n intervals. Taken from [1]

The same method is applied to find the optimal jump point. Which leads to the optimal function

$$f(x) = \begin{cases} 3.1686, & 0 \leq x < 6.41 \\ 9.0479, & x \geq 6.41 \\ -f(-x), & x < 0 \end{cases}$$

which is also shown in Figure 4.3. This function reaches the cost value 0.19 for $\sigma = 5, k = 0.2$. As the best value known before was 0.365015, this is an reduction of more than 50%. Moreover, they were able to beat all previous known values for benchmarks different to $\sigma = 5, k = 0.2$. [1]

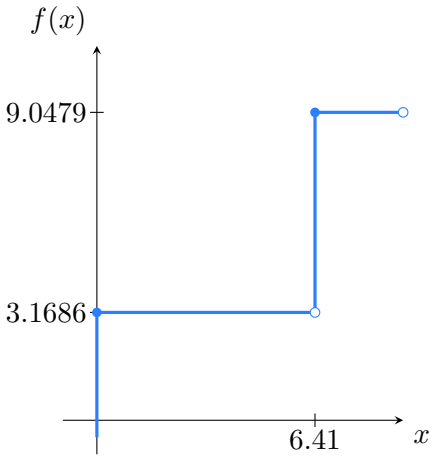


Figure 4.3.: Optimal function from Deng and Ho for $x \geq 0$

4.2. Gaining a 3.5-step Function: Results from Li et al. [20]

In their paper, "Learning Approaches to the Witsenhausen Counterexample From a View of Potential Games," Li, Marden and Shamma consider the problem stated by Witsenhausen as a problem from game theory. They develop a method based on ideas from

game theory which in the end lead to a 3.5 step function. Doing this, they outperformed any cost value previously known. [20]

To apply ideas from game theory on the problem stated by Witsenhausen, the problem first has to be converted into a *game*. Therefore, it should be clear what goal is pursued in game theory. Usually in game theory the aim is to obtain a so called *Nash Equilibrium*. The definition of a *Nash Equilibrium* in simple words is given by:

> "Nash equilibrium is a concept in game theory where the game reaches a state that gives individual players no incentive to deviate from their initial strategy. The players know their opponents' strategy and can't deviate from their chosen strategy because it remains optimal." [16]

Before we might search such a state, Witsenhausen's problem must be converted to a problem from game theory. The problem will be converted to a *potential game*. Which means there are n players $\mathcal{N} = \{1, 2, \dots, n\}$ where all players have the same utility function

$$U_i(a) = U_g(a), \quad U_g : \mathcal{A} \to \mathbb{R}, \quad i = 1, \dots, n\,,$$

whereby $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_n$ is the action set of all players and $a = (a_1, \dots, a_n) \in \mathcal{A}$. Furthermore, the notation

$$a_{-i} = (a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n)$$

is used. Knowing this, the definition of a *pure Nash equilibrium* is simple. An $a^* \in \mathcal{A}$ is called a *pure Nash equilibrium* if

$$U_i(a_i^*, a_{-i}^*) = \max_{a_i \in \mathcal{A}_i} U_i(a_i, a_{-i}^*) \quad \forall i = 1, 2, \dots, n\,.$$

This means the aim is to find an action state a^* as the utility function $U_g(a)$ is maximized for all players. [20]

When converting the problem stated by Witsenhausen into a *potential game*, Li et al. first assume the optimal f is an odd function. Then, the interval $[0, \infty)$, on which the function is approximated, is divided into $n + 1$ subintervals

$$[b_i, b_{i+1}), \quad b_0 = 0, b_{n+1} = \infty, \quad i = 1, \dots, n\,.$$

Each subinterval is seen as one player in the game we define. The values $a_i\,, i = 1, \dots, n$ chosen on those intervals are seen as the actions of the players. This leads to the following representation of the function in the game

$$f(x) = \begin{cases} a_1, & 0 \le x < b1 \\ a_2, & b_1 \le x < b_2 \\ \vdots & \\ a_n, & b_{n-1} \le x < b_n \\ a_{n+1}, & b_n \le x < \infty \\ -f(-x), & x < 0 \end{cases}.$$

The utility function is then chosen as the negative cost function J from Witsenhausen's counterexample, which leads to

$$U = -J$$

as the utility function. [20]

This way, the problem from Witsenhausen's counterexample is formulated as a potential game. On this game the learning algorithm *fading memory joint strategy fictitious play (JSFP) with inertia* is applied to gain pure Nash equilibriums. A detailed description of the algorithm is omitted, since it would go beyond the scope of this discussion. [20]

Choosing $n = 600$ players they reached the function represented in Table 4.2. The plot

$f(x)$	Interval	$f(x)$	Interval
0.00	$0.000 \leq x < 0.467$	13.233	$10.667 \leq x < 11.667$
0.033	$0.467 \leq x < 1.400$	13.267	$11.667 \leq x < 12.633$
0.067	$1.400 \leq x < 2.333$	13.300	$12.633 \leq x < 13.633$
0.100	$2.333 \leq x < 3.333$	13.333	$13.633 \leq x < 14.600$
6.467	$3.333 \leq x < 4.133$	13.367	$14.600 \leq x < 15.567$
6.500	$4.133 \leq x < 5.100$	13.400	$15.567 \leq x < 16.533$
6.533	$5.100 \leq x < 6.033$	13.433	$16.533 \leq x < 16.867$
6.567	$6.033 \leq x < 7.000$	20.267	$16.867 \leq x < 17.531$
6.600	$7.000 \leq x < 7.933$	20.300	$17.531 \leq x < 18.567$
6.633	$7.933 \leq x < 8.867$	20.333	$18.567 \leq x < 19.600$
6.667	$8.867 \leq x < 9.833$	20.367	$19.600 \leq x < 20$
6.700	$9.833 \leq x < 9.967$	20.400	$x \geq 20$
13.200	$9.967 \leq x < 10.667$	$-f(-x)$	$x < 0$

Table 4.2.: Function determined by Li et al. for $n = 600$ [20]

of the function is shown in Figure 4.4. The function reaches for $\sigma = 5, k = 0.2$ the cost value 0.1670790 which outperforms any cost value gained before.

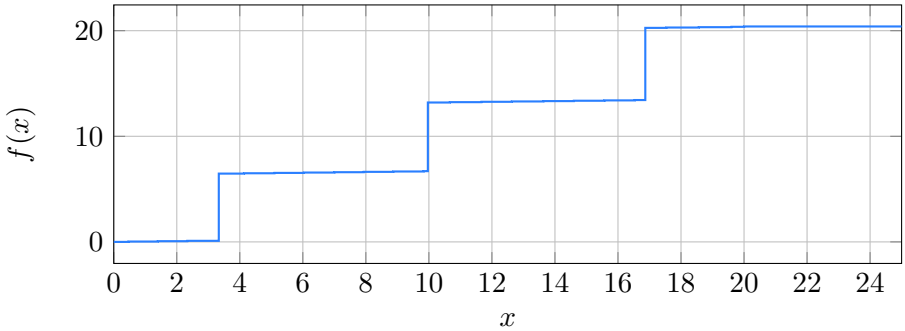


Figure 4.4.: Plot of function gained by Li et al. for $x \geq 0$

4.3. Gaining a Sloped 4-step Function: Results from Karlsson et al. [17]

In their paper "Iterative Source-Channel Coding Approach to Witsenhausen's Counterexample" Karlsson, Gattami, Oechtering and Skoglund introduce a new optimization approach, which alternately optimizes f and g while keeping the other function fixed. [17]

The approach published is based on the idea of the Lloyd-algorithm. The main idea of the Lloyd-algorithm is just optimizing one variable part regarding a cost function. The *variable rest* is then assumed to be fixed and just one part is optimized. The algorithm will then converge against a local minimum. This idea should now be generalized, to gain a *generalization of the Lloyd-algorithm.* Therefore, four main points must be considered:

1. Derive necessary conditions for whether optimizing f or g.
2. Discretize inputs for f and g to finite set.
3. Optimize f and g alternately to fulfill their necessary conditions.
4. Use *noise channel relaxation* to make solutions less dependent on the initialization.

[17] Before focusing on the necessary conditions, we have a look on the based cost functional to minimize. This is given by

$$J(f,g) = E\left[k^2(f(x)-x)^2 + (f(x)-g(f(x)+\nu))^2\right] .$$

When g is fixed and we name

$$F(x,x_1,g(y')) = \left(k^2(x_1-x)^2 + (x_1-g(y'))^2\right)$$

then the necessary condition for the optimal f is given as

$$f(x) = \arg\min_{x_i\in\mathbb{R}} \left(f_{y|x}(y')F(x,x_1,y')\,\mathrm{d}y\right) ,$$

whereby $f_{y|x}(y')$ means the probability density function of $y = f(x)+\nu$. For fixed f the first part of the expected value does not have to be considered. This means, optimizing J for fixed f is equivalent to minimizing

$$\min_g E[(f(x)-g(f(x)+\nu))^2] .$$

As we know, this yields the MMSE as optimal function g. Therefore, the necessary condition for optimal g is given as

$$g(y') = E[f(x)|y=y'] ,$$

which is the optimal g we already know. [17]

As the inputs x, y to the necessary conditions both come from infinite sets, a discretization must be chosen for the input variables. Therefore, Karlsson et al. choose

$$S_L = \left\{ -\Delta\frac{L-1}{2}, -\Delta\frac{L-3}{2} \ldots, \Delta\frac{L-3}{2}, \Delta\frac{L-1}{2} \right\}$$

whereby $L \in \mathbb{N}$ and $\Delta \in \mathbb{R}_+$ determine the number of points and the spacing between them. Then, the inputs are chosen as

$$\begin{aligned} x &\in S_L \\ y' &\in Q_{S_L}(y')\,, \end{aligned}$$

whereby Q_{S_L} for an input y' returns the nearest value to y' included in S_L. [17]

The iterative optimization then is performed including a *Noise Channel Relaxation* (NCR). The NCR therefore starts optimizing for some changed parameters that obtain a simpler solution, to a maybe different scenario. Then, the obtained variables are chosen as an input for the next iteration which is closer to the original scenario. [17]

Using this idea and the NCR, Karlsson et al. were able to reach the cost value 0.16692462 for $\sigma = 5, k = 0.2$ with a sloped 4-step function. The function gained is shown in Figure 4.5. [17]

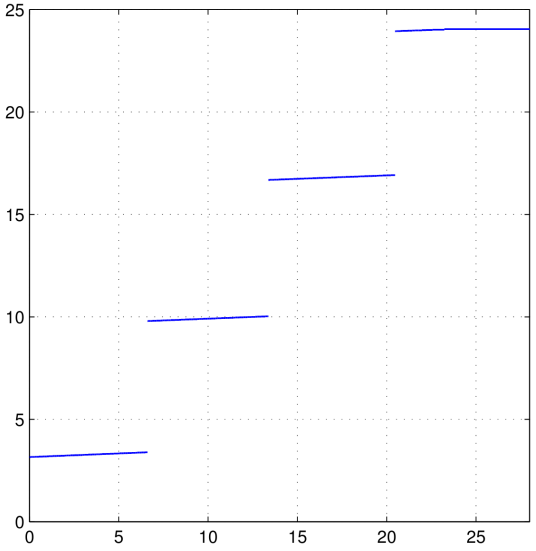


Figure 4.5.: Function gained by Karlsson et al. Taken from [17]

4.4. Gaining a Curved Step Function: Results from Tseng and Tang [28]

In their paper, "A Local Search Algorithm for the Witsenhausen's Counterexample," Tseng and Tang use variational analysis to gain necessary conditions for the optimal controller for Witsenhausen's counterexample. They do not search within a specific type of functions and gain the currently best known solution. [28]

As the conditions that might be gained from variational analysis only fit for local minima, they first introduce so called *local Nash minimizers*. This means that, for the known cost functional J, (f, g) is a *local Nash minimizer* if it fulfills

$$\begin{aligned} J(f + \delta f, g) &\geq J(f, g) \\ J(f, g + \delta g) &\geq J(f, g) \end{aligned}$$

for arbitrary functions δf and δg. They then focus on finding *good* local minima instead of the global minimum. [28]

By using variational analysis they gain a first and second order condition for a *local Nash minimizer*. The first order condition is given as

$$\int \frac{\delta J(f,g)(x_0)}{\delta f}\delta f(x_0)\,\mathrm{d}x_0 = 0$$
$$\int \frac{\delta J(f,g)(y)}{\delta g}\delta g(y)\,\mathrm{d}y = 0\,.$$

This implies, as δf and δg are arbitrary,

$$\frac{\delta J(f,g)(x_0)}{\delta f} = 0, \qquad \frac{\delta J(f,g)(y)}{\delta g} = 0\,.$$

They gained the second order condition as

$$\int \frac{\partial}{\partial f}\frac{\delta J(f,g)(x_0)}{\delta f}\delta f^2(x_0)\,\mathrm{d}x_0 \geq 0$$
$$\int \frac{\partial}{\partial g}\frac{\delta J(f,g)(y)}{\delta g}\delta g^2(y)\,\mathrm{d}y \geq 0\,.$$

Using those results Tseng and Tang were able to determine a rule to update a function given by a point representation. For $\frac{\partial}{\partial x_1}\frac{\delta J(f,g)(x_0)}{\delta f} = 0$ the rule is given as

$$f(x_0) \leftarrow f(x_0) - \tau\frac{\delta J}{\delta g}(f,g)(x_0) \tag{4.1}$$

otherwise by

$$f(x_0) \leftarrow f(x_0) - \frac{\frac{\delta J}{\delta g}(f,g)(x_0)}{\left|\frac{\partial}{\partial f}\frac{\delta J}{\delta f}(f,g)(x_0)\right|}\,. \tag{4.2}$$

Applying this rule leads to the function plots as for example shown in Figure 4.6. As to see, those plots include noise points that seem to do not converge against the true solution. [28]

To address this problem, Tseng and Tang added a denoising step, wherein they take the noised values as input and then optimize the function locally for a fixed g. This leads to the cost function

$$C_X(a,x_0) = k^2(a-x_0)^2 + \int (a-g(y))^2 f_x(x_0) f_\nu(y-a)\,\mathrm{d}y$$

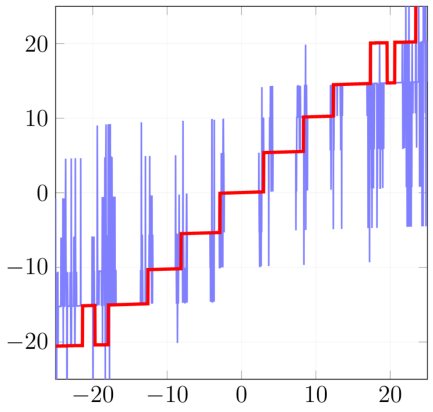
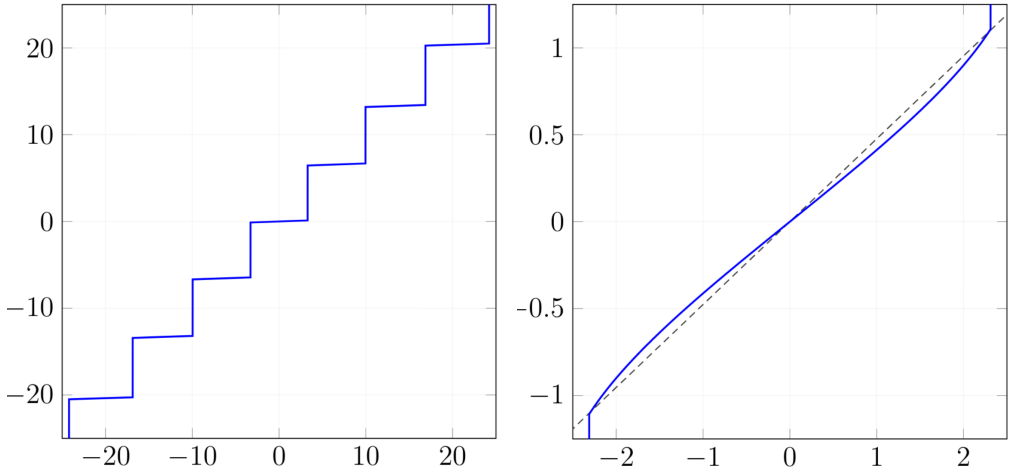


Figure 4.6.: Noise occurance in reached function. Taken from [28]



(a) Best function for $\sigma = 5, k = 0.2$. Taken from [28]

(b) Step plateau for $k = 0.733$. Taken from [28]

Figure 4.7.: Functions gained by Tseng and Tang

and therefore to the denoising rule

$$f(x_0) \leftarrow \underset{x' \in B_r(x_0)}{\operatorname{argmin}} C_X(f(x'), x_0)\,.$$

Applying the search based on the necessary criteria and the denoising step, leads to cost value of 0.1668797 which is the best known up to now. The function determined for $\sigma = 5, k = 0.2$ is shown in Figure 4.7 (a). Another interesting results Tseng and Tang gained, is shown in Figure 4.7 (b). There, it might be seen that for $k = 0.733$ the plateaus of the steps are not affine. It might be seen that curved plateaus occure, which is a quite new observation. [28]

5. Implementation of the Cost Functional

Gaining an efficient as well as reliable method to approximate the cost functional is one of the main goals in this thesis. In this Chapter, we will develop a method fulfilling the requirements and report about the implementation.

5.1. Implementation of the Cost Function

As Witsenhausen showed, it is valid that

$$J(f) - k^2 E[(x - f(x))^2] = 1 - 4 \int \left(\frac{\mathrm{d}}{\mathrm{d}y} \sqrt{D_f(y)} \right)^2 \mathrm{d}y, \tag{5.1}$$

whereby

$$D_f(y) = \int f_\nu(y - f(x)) f_x(x) \mathrm{d}x \tag{5.2}$$

[29]. We want to approximate the integral as well as the derivative in one step. Moreover, we need an estimator for the error that follows by the approximation we use. If the error becomes too large, we have to decrease the step size. We assume we know the intervals $[x_\ell, x_u]$ and $[y_\ell, y_u]$ that have to be considered when integrating with respect to x respectively y.

As the right side of Equation (5.1) consists out of an integral and a derivative with respect to y, we want to approximate both in once. This will be done by using spline interpolation.

5.2. Principle of Approximating the Integrand

As to see in Equation (5.2) the integrand $\left(\frac{\mathrm{d}}{\mathrm{d}y} \sqrt{D_f(y)} \right)^2$ in Equation (5.1) is defined as an improper integral and we first need a way to handle it. We choose to approximate $\sqrt{D_f}$ with a function that can be handled easily and for which the derivative, the square and the integral can be obtained analytically. Moreover, we do not want to restrict our method on a special class of functions for f as for example Lee et al. did in [19] for the class of step functions. By doing this, we cannot use an analytical expression for the integrand which results in higher costs for determining it. As the approximation for $\sqrt{D_f}$ is chosen in a way the integration can be handled analytically, the evaluation

of the second integral is much cheaper than in a way where just the integrand can be expressed analytically.

The approximation of $\sqrt{D_f}$ is done by spline interpolation. Therefore, easy to handle functions are gained as well as there is a better stability for small distances between the points interpolated. We use cubic splines which results in a representation

$$s(x) = \begin{cases} C_1(x), & x_0 \le x \le x_1 \\ \quad \dots & \\ C_i(x), & x_{i-1} < x \le x_i \\ \quad \dots & \\ C_n(x), & x_{n-1} < x \le x_n \end{cases} \tag{5.3}$$

$$= C_1(x)\mathbb{1}_{x_0 \le x \le x_1}(x) + \sum_{i=2}^{n} C_i(x)\mathbb{1}_{x_{i-1} < x \le x_i}(x) \tag{5.4}$$

(cf. [9]) for a approximation of $\sqrt{D_f}$, whereby $C_i(x) = a_i + b_i x + c_i x^2 + d_i x^3$. For each cubic polynomial C_i the conditions

1. $\forall i = 1, \dots n: \quad C_i(x_{i-1}) = \sqrt{D_f(x_{i-1})},\ C_i(x_i) = \sqrt{D_f(x_i)}$
2. $\forall i = 1, \dots, n-1: \quad C_i'(x_i) = C_{i+1}'(x_i)$
3. $\forall i = 1, \dots, n-1: \quad C_i''(x_i) = C_{i+1}''(x_i)$

must be fulfilled. Moreover, the boundary conditions

4. $C_1'''(x_1) = C_2'''(x_1)$
5. $C_{n-1}'''(x_{n-1}) = C_{n-1}'''(x_{n-1})$

are chosen as there is no knowledge about the derivative or the bendings (cf. [27]).

The next section will give information about how to choose the grid points x_i. For this moment, we assume there are $n \in \mathbb{N}$ grid points known and the spline interpolation s was figured out. Then, we can determine the integrand of the integral in Equation (5.1) for each $C_i,\ i = 1, \dots, n$ analytically. We obtain

$$\left(\frac{\mathrm{d}}{\mathrm{d}x}\sqrt{D_f(x)}\right)^2 = \left(\frac{\mathrm{d}}{\mathrm{d}x}s(x)\right)^2 + \varepsilon, \quad \varepsilon \in \mathbb{R}\,. \tag{5.5}$$

As we know the shape of s, we get for $x_{i-1} < x < x_i$

$$\begin{aligned}\left(\frac{\mathrm{d}}{\mathrm{d}x}s(x)\right)^2 &= \left(\frac{\mathrm{d}}{\mathrm{d}x}\left(a_i + b_i x + c_i x^2 + d_i x^3\right)\right)^2 \\ &= \left(b_i + 2c_i x + 3d_i x^2\right)^2.\end{aligned}$$

This representation of the integrand is easy to handle and results in a simple way to approximate the integral.

5.3. Adaptive Choice of Grid Points

In the previous section the grid points $x_i,\ i = 1, \ldots, n$ were *god given*. Of course, this is not the case and a way to choose them properly must be found. For this reason we remember ε from Equation (5.5). This variable characterizes the local error made by the approximation s in a position x. As the quality of the approximation of the integral in Equation (5.1) will depend on the quality of s, we will try to control ε by the way we choose the grid points.

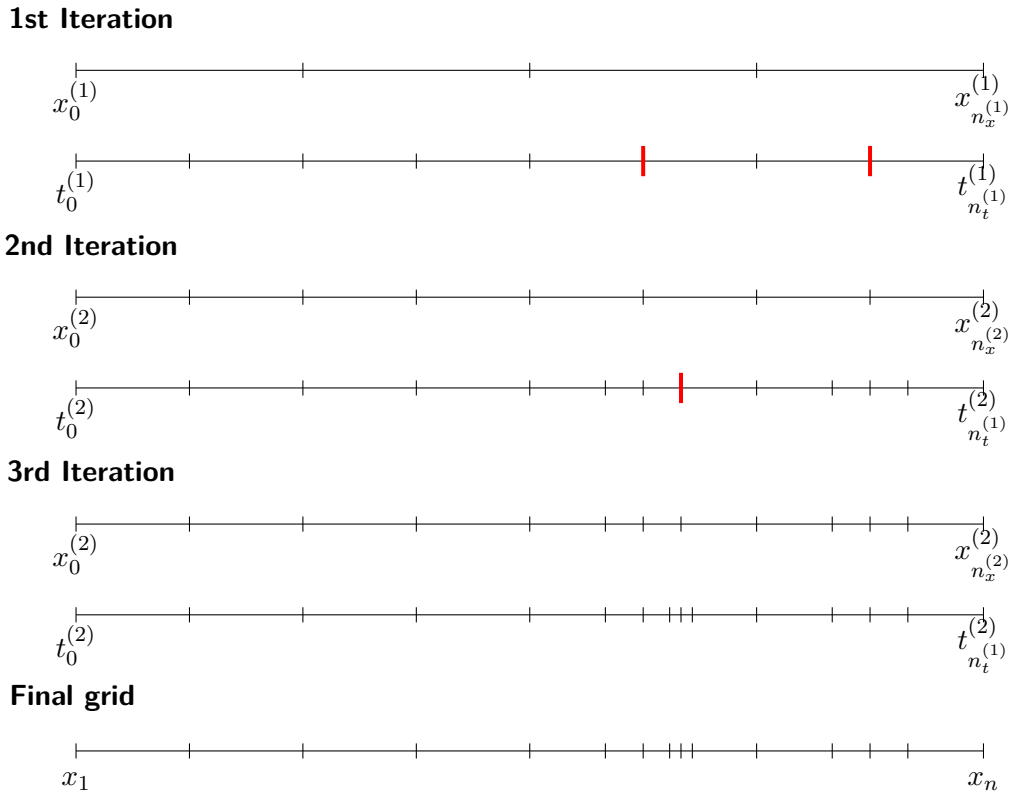


Figure 5.1.: Grid finding algorithm, problematic areas are colored red.

The concept chosen for this is really easy and depicted in Figure 5.1. We start with $n_x^{(1)}$ initial grid points $x_i^{(1)},\ i = 1, \ldots, n_x^{(1)}$. Moverover, we use $n_t^{(1)}$ test points to determine an estimator for ε. In the first iteration, $n_x^{(1)} = n_t^{(1)}/2$ and the grid points $t_i^{(1)},\ i = 1, \ldots, n_t^{(1)}$ are uniform sampled on the interval to consider. Therefore, every second t_i also is covered by a x_j. We call

$$\left\{ \left(x_i^{(j)}, \sqrt{D_f(x_i^{(j)})} \right) \;\middle|\; i = 1, \ldots, n_x^{(j)} \right\}$$

the j-th interpolating set and

$$\left\{ \left(t_i^{(j)}, \sqrt{D_f(t_i^{(j)})} \right) \;\middle|\; i = 1, \ldots, n_t^{(j)} \right\}$$

the j-th test set.

In each iteration, we determine a cubic spline s_j, as described in the previous section, using the j-th interpolation set. Then, we evaluate the spline for each position given in the j-th test set and compare it to the value gained by evaluating the original function. We get the local error by

$$e_i^{(j)} = \left| s_j(t_i^{(j)}) - \sqrt{D_f(t_i^{(j)})} \right|, \quad i = 1, \ldots n_t^{(j)}.$$

As we want to control the local error, we have to set a threshold $\varepsilon_{\max}$ where $e_i^{(j)} > \varepsilon_{\max}$ leads to a sharpening of the grid at this position.

If a sharpening at position $t_k^{(j)}$ is needed, the tuples

$$\left(t_k^{(j)} - h_l, \sqrt{D_f(t_k^{(j)} - h)} \right), \quad \left(t_k^{(j)} + h_r, \sqrt{D_f(t_k^{(j)} + h_r)} \right),$$

whereby

$$h_l = t_k^j - (t_k^j - t_{k-1}^j)/3, \quad h_r = t_k^j + (t_{k+1}^j - t_k^j)/3,$$

are added to the $(j+1)$-th test set. This test set also contains all tuples the j-th test aready included. Then, the j-th test set becomes the $(j+1)$-th interpolation set.

This procedure is repeated until there is no more sharpening needed and all positions have a deviation less than $\varepsilon_{\max}$. Finally, the cubic spline that will be used is gained by interpolating the test set gained last. The positions in the final grid are named $x_i,\ i = 1, \ldots, n$.

5.4. Determining the Integral needed

As we now have a method to choose the grid points and also know how to gain a spline interpolating of these points, we have to think about how to determine the right side of Equation (5.1).

Therefore, we look at the representation of the spline we found in Equation (5.4). We add all polynomials C_i and therefore can cover the whole area $[x_0, x_n]$. We need to differentiate, square and integrate this expression and therefore get

$$\int s(x)\,\mathrm{d}x = \int C_1(x)\mathbb{1}_{x_0 \le x \le x_1}(x) + \sum_{i=2}^{n} C_i(x)\mathbb{1}_{x_{i-1} < x \le x_i}(x)\,\mathrm{d}x.$$

We assume, we can reduce the improper integral on an area (ℓ, u), whereby $[x_0, x_n] \subset [\ell, u]$, and use that the Lebesgue measure of a point is zero. This yields to

$$\begin{aligned}
\int_\ell^u \left(\frac{\mathrm{d}}{\mathrm{d}x}s(x)\right)^2 \mathrm{d}x &= \int_\ell^u \left(\frac{\mathrm{d}}{\mathrm{d}x}\left[C_1(x)\mathbb{1}_{x_0\le x\le x_1}(x) + \sum_{i=2}^{n} C_i(x)\mathbb{1}_{x_{i-1}<x\le x_i}(x)\right]\right)^2 \mathrm{d}x \\
&= \int_\ell^u \left(\frac{\mathrm{d}}{\mathrm{d}x}C_1(x)\mathbb{1}_{x_0\le x\le x_1}(x)\right)^2 \mathrm{d}x \\
&+ \sum_{i=2}^{n}\int_\ell^u \left(\frac{\mathrm{d}}{\mathrm{d}x}C_i(x)\mathbb{1}_{x_{i-1}<x\le x_i}(x)\right)^2 \mathrm{d}x \\
&= \int_{x_0}^{x_1} \left(\frac{\mathrm{d}}{\mathrm{d}x}C_1(x)\right)^2 \mathrm{d}x + \sum_{i=2}^{n}\int_{x_{i-1}}^{x_i} \left(\frac{\mathrm{d}}{\mathrm{d}x}C_i(x)\right)^2 \mathrm{d}x \\
&= \sum_{i=1}^{n}\int_{x_{i-1}}^{x_i} \left(\frac{\mathrm{d}}{\mathrm{d}x}C_i(x)\right)^2 \mathrm{d}x.
\end{aligned}$$

Using that $C_i(x) = a_i + b_i x + c_i x^2 + d_i x^3$, we get

$$\begin{aligned}
\sum_{i=1}^{n}\int_{x_{i-1}}^{x_i} \left(\frac{\mathrm{d}}{\mathrm{d}x}C_i(x)\right)^2 \mathrm{d}x &= \sum_{i=1}^{n}\int_{x_{i-1}}^{x_i} \left(b_i + 2c_i x + 3d_i x^2\right)^2 \mathrm{d}x \\
&= \sum_{i=1}^{n}\int_{x_{i-1}}^{x_i} 9d_i^2x^4 + 12c_i d_i\, x^3 + 6b_i d_i x^2 + 4c_i^2 x^2 + 4b_i c_i x + b_i^2 \,\mathrm{d}x \\
&= \sum_{i=1}^{n}\left[\frac{9{d_i}^2x^5}{5} + 3c_i d_i x^4 + \frac{\left(6b_i d_i + 4{c_i}^2\right)x^3}{3} + 2b_i c_i x^2 + {b_i}^2 x\right]_{x_{i-1}}^{x_i}.
\end{aligned}$$

This expression can be evaluated really cheap and will be used for determining the integral in the used implementation.

5.5. Implementation Details

In the second part of this thesis, we will focus on how to determine an approximation for the optimal controller function. For this optimization, the amount of evaluations of the cost functional will easily reach a few thousand evaluations per optimization. For this reason, it is mandatory that the evaluations might be done in the least possible amount of time. A few details on the implementation of the cost functional will be introduced in this section.

5.5.1. Approximating Integrals for possibly non-smooth Functions

The function that Witsenhausen used in his counterexample, as well as nearly all functions that have been published as the "best function for Witsenhausen's counterexample

for the moment", had discontinuities. As most usual integration methods may run into troubles for those functions, this issue is addressed in this subsection.

Performing the integration on a possibly non-smooth function, involves two steps:

1. Detect discontinuities
2. Use common integration method considering discontinuities

Detecting the discontinuities is done by applying the definition of right-continuity. There we call a position $x_d \in \mathbb{R}$ a discontinuity of a function f if

$$\lim_{x \to x_d^+} f(x) \neq f(x_d)\,.$$

In our case we express discontinuity by a threshold with

$$|f(x_d) - f(x_d + \delta)| > \varepsilon, \quad \delta, \varepsilon > 0\,.$$

Numerical experiments indicate that $\delta = 10^{-6}$ and $\varepsilon = 0.5$ are appropriate values. Choosing a relatively large value for ε is reasonable, given that the functions, currently known for being the best controllers for Witsenhausen's counterexample, have step heights bigger than 0.5. This method is implemented as a grid search which results in run times less than 0.001 seconds to detect the step positions.

For the integration two cases are handled differently. The first case considers the *uncritical* part of the cost functional, which means

$$J_1(f) = \int k^2 f_x(x) \cdot (f(x) - x)^2 \,\mathrm{d}x\,.$$

As this expression just has to be evaluated once to determine J, we just choose a modified `scipy` wrapper of the `QUADPACK` package. There we pass the determined positions of discontinuity as breakpoints to ensure the right grid points are chosen. This leads to the needed accuracy.

The more critical integral expression is

$$D_f(y) = \int f_\nu(y - f(x)) f_x(x) \,\mathrm{d}x\,,$$

which is also part of the integral expressing J_2, the second part of the cost functional. For this reason, the expression is evaluated hundreds of times just to determine the cost value once. Therefore, the evaluation of this integral is done on the GPU. This is done by using the Python package `PyTorch`. `PyTorch` is primarily known in the Machine Learning and Deep Learning community, where it is widely used and for this reason, offers extensive tools for developing, training and evaluating such networks. Moreover, it provides a flexible and efficient way to access the GPU enabling us fast numerical computations in a familiar environment. [26]

Since no package offers an integration method that allows us to specify the breakpoints for PyTorch, we created an own implementation. This is also done by using a quadrature rule. In our case, we use the Gauß-Legendre Quadrature as basis method. Therefore, the weights are used from the `scipy` library and then transferred to the GPU using the `PyTorch` implementation. Then, the interval, the integration is performed on, is divided into subintervals with length 1. Moreover, those intervals are divided at the step positions determined before. For those intervals the weights and nodes are obtained. Afterwards, the multidimensional node array is flattened to be able to evaluate the function in one dimension at once on the GPU. Doing this, we minimize the GPU overhead occurring because of the data transfer between CPU and GPU. After evaluating the function, the results are reshaped back to the old shape. Then, we are able to perform the needed matrix multiplications on GPU and gain the value of the integral.

5.5.2. Gaining Borders for the Integrals

In the previous sections, we assumed how to get areas $[u, \ell]$ that can be considered instead of solving improper integrals. In practice, this assumption works, and there are ways to justify it.

If we have a look on the cost function J, we get

$$J(f, g) = E\left[k^2(x - f(x))^2\right] + E\left[(f(x) - g(f(x) + \nu))^2\right].$$

As the random variables x and ν have Gaussian distribution, we know there are probability densities f_x and f_ν. Since $E[x] = E[\nu] = 0$, these densities are strictly monotonously decreasing on $(0, \infty)$ respectively strictly monotonously increasing on $(-\infty, 0)$. Therefore, we can find $[\ell_x, u_x]$ and $[\ell_\nu, u_\nu]$ such that

$$\begin{aligned} f_x(\ell_x) < \varepsilon_x, \quad f_x(u_x) < \varepsilon_x, \\ f_\nu(\ell_\nu) < \varepsilon_\nu, \quad f_\nu(u_\nu) < \varepsilon_\nu. \end{aligned}$$

In Section 5.3, we chose an upper boundary $\varepsilon_{\max}$ for the local error. It seems logical that we have to integrate at least above an area $[u, \ell]$ such that $f_\nu(u) < \varepsilon_{\max}$ and $f_\nu(\ell) < \varepsilon_{\max}$. On the other hand, we have to consider that $f_\nu(u) >$ `machine epsilon`, $f_\nu(\ell) >$ `machine epsilon` as otherwise the results can become arbitrary. Using this, we can find intervals wherein u as well as ℓ should be found. In this work, the described procedure is applied in every case where improper integrals representing estimation values based on a Gaussian distribution arise, in order to determine appropriate integration boundaries. An exception is made when the resulting boundaries do not cover the interval $[-50{,}50][-50{,}50]$; in such cases, this interval is used instead.

5.5.3. Results in Performance and Precision

To have a look on the performance as well as the precision gained by using this method, two examples discussed in the original article by Witsenhausen will be discussed. Therefore, we define the default benchmark, whereby $\sigma = 5$ and $k = 0.2$.

First of all, we want to look on the optimal affine solution he chose in his article. For an affine function

$$f_a(x) = \lambda x, \quad \lambda > 0$$

he derived that the cost function can be expressed for the optimal chosen g_a^* by

$$J(f_a, g_a^*) = k^2\sigma^2(1-\lambda)^2 + \frac{\lambda^2\sigma^2}{1+\lambda^2\sigma^2}$$

[29]. Choosing λ arbitrary with 0.158 and estimating the expression for the given benchmark in `Maple` with up to 16 digits, we get 1.0932383673419124. Using the algorithm described before using $\varepsilon_{\max} = 10^{-15}$, we get 1.0932383673419135 which yields a precision of 14 digits.

Moreover, we want to consider the function, Witsenhausen used to proof there is a better solution than the affine one. This function is given by

$$f(x) = \sigma \operatorname{sgn}(x)$$

[29]. As there is just an upper bound for the cost functional J for this f, we have to consider the results from another paper to compare our results. We take the results from the paper of Lee et al. [19]. They gained the value 0.404253198895. As this value just contains 13 digits, we will choose $\varepsilon_{\max} = 10^{-14}$. This yields 0.4042531988953495, which is exactly the same up to the 13-th digit.

In Figure 5.2 the ratio of the needed compute time and the reached absolute error is shown. As to see, the algorithm always reaches the needed accuracy in a time less than 10 seconds. For accuracies below 10^{-8} the compute time is less than a second. All tests were performed on an office computer with an Intel® Core™ i7-14700 CPU and a NVIDIA GeForce GTX 1650 GPU.

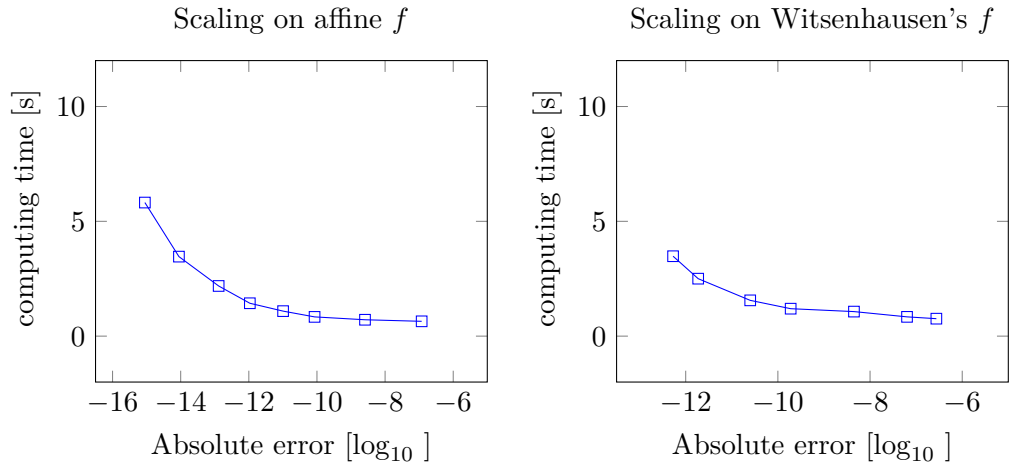


Figure 5.2.: Performance of the algorithm obtained

Part II.

Optimization using Variational Analysis

6. Introduction to Variational Analysis

We want to build an optimization method based on variational analysis. Therefore, in this Chapter, the needed theory is introduced. The theory is illustrated at the well known example of the Brachistochrone.

6.1. The Quarrel of two Brothers or The Problem of the Brachistochrone

The title of this section may sound a little sensational. Well, on the one hand, there is this quarrel between the brothers Johann and Jakob Bernoulli, but on the other hand, this quarrel gave rise to one of the first problems of the calculus of variations.

The family history of the two brothers did not start out so combative. Both brothers shared a talent for and love of mathematics, as well as the fate of being forbidden by their father to study mathematics. Jakob, who studied philosophy and theology but secretly attended mathematics lectures, taught his little brother Johann and set him mathematical challenges. Even if they shared the same fate and together they found a way to deal with it, the brothers became rivals. To show the world his mathematical brilliance, Johann Bernoulli in 1696 published the *Brachistochrone Problem* knowing, he may solve it and prompts the *most brilliant mathematicians in the world* to solve it. The problem asks the question which way a ball has to choose in a vertical plane to get from a position A to a position B just by using the force of gravity. [12]

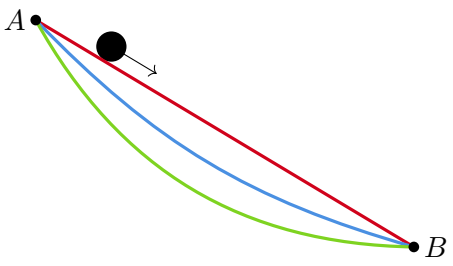


Figure 6.1.: Searching the way the ball needs the lowest amount of time

After some time, several mathematicians have responded to the publication and found the optimal solution using various ways. We want to focus on the way Johann Bernoulli published. As the concept of calculus of variations was invented in the 18th century, Bernoulli chose a different way based on Fermat's principle.

6.1.1. Mathematization of the Problem

Before we may solve the problem, we have to find a way to express the given question in a mathematical way. We know from the Pythagorean theorem that

$$\begin{aligned}\Delta s &= \sqrt{\Delta x^2 + \Delta y^2} \\ &= \Delta x \sqrt{1 + \left(\frac{\Delta y}{\Delta x}\right)^2}, \quad \Delta x \neq 0\,, \end{aligned} \tag{6.1}$$

whereby Δs is the change in the distance, Δx the change in the x-coordinate and Δy the change in the y-coordinate. To keep things easy, the friction is ignored and the starting point is chosen as $A = (0, 0)$. Then, the law of conservation of energy tells us

$$m \cdot g \cdot y + \frac{1}{2} m \cdot v^2 = \text{const}\,,$$

whereby m is the mass of the ball, g the gravitational force, $y < 0$ the y-coordinate and v the current speed of the ball. As we chose the ball to start in $A = (0, 0)$, we get

$$m \cdot g \cdot y + \frac{1}{2} m \cdot v^2 = 0$$

in this position. Using this and the definition of speed, we get

$$\begin{aligned} v =& \sqrt{2g}\sqrt{-y} = \frac{\Delta s}{\Delta t} \\ \Leftrightarrow& \Delta t = \frac{\Delta s}{\sqrt{2g}\sqrt{-y}}\,. \end{aligned} \tag{6.2}$$

Inserting the results from equation (6.1) into equation (6.2), we obtain

$$\Delta t = \frac{1}{\sqrt{2g}} \Delta x \, \frac{\sqrt{1 + \left(\frac{\Delta y}{\Delta x}\right)^2}}{\sqrt{-y}}\,.$$

For $\Delta t \to 0$, we get

$$\frac{\Delta y}{\Delta x} \to y'\,.$$

Doing many little steps $i \in I$ with the step widths Δx_i, Δy_i, we get in the limit

$$T(y) = \lim_{\Delta t \to 0} \frac{1}{\sqrt{2g}} \sum_{i \in I} \frac{\sqrt{1 + \left(\frac{\Delta y_i}{\Delta x_i}\right)^2}}{\sqrt{-y_i}} \Delta x_i = \frac{1}{\sqrt{2g}} \int_0^{x_B} \frac{\sqrt{1 + y'(x)^2}}{\sqrt{-y(x)}}\, \mathrm{d}x\,,$$

for the time the ball needs to get from $A = (0, 0)$ to $B = (x_B, y_B)$. As we want to minimize this time, we want to find a $y^* : \mathbb{R} \to \mathbb{R}$ such as

$$T(y^*) = \min\{T(y) \mid y : \mathbb{R} \to \mathbb{R}\}$$

is fulfilled. (cf. [24])

6.1.2. Johann Bernoullis' Solution - Not Knowing the Calculus of Variations

As mentioned, Bernoullis' idea is based on Fermats' principle. The principle says that light does not take the shortest way from one position to another but the way it takes the light least time. Today, such a method would be called discretization of the solution.

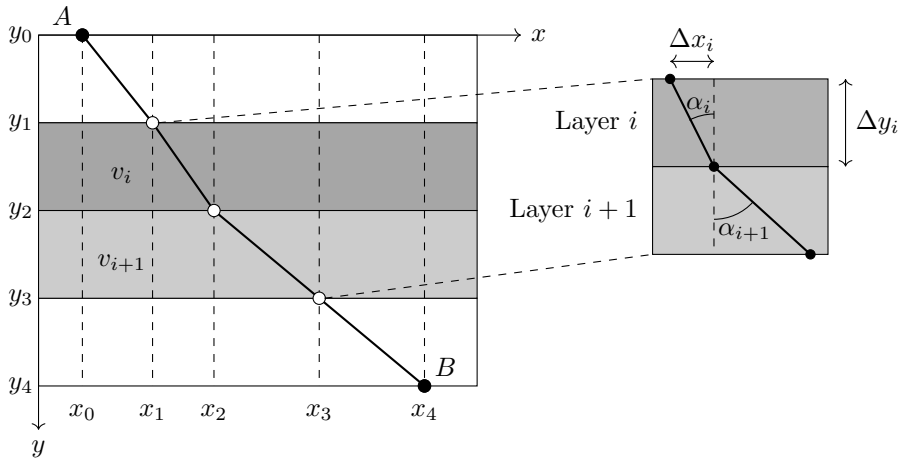


Figure 6.2.: Fermat's principle used by Johann Bernoulli, like in [24]

As to see in Figure 6.2, there are several layers y_i, $i = 0, \ldots, N$ where the light is refracted and reaches another transmission medium v_{i+1} with another speed of light c_{i+1}, $i = 1, \ldots, N$. The light now takes the refraction angle that minimizes the time needed.

To keep the notation simple, we have a look on the light going from a point $\tilde{A}$ to a point $\tilde{B}$ through a point X that has to be chosen as it minimizes the time to reach $\tilde{B}$. Above X the speed of the light is given by c_1, below by c_2. If the distances are given like in Figure 6.3, we get for the covered distance in the upper medium

$$s_1 = \sqrt{x^2 + a^2}$$

and for the covered distance in the lower medium

$$s_2 = \sqrt{(d-x)^2 + b^2}\,.$$

Knowing the speed of the light in each medium, we get the time to get from $\tilde{A}$ to $\tilde{B}$ by evaluating

$$f(x) = \frac{\sqrt{x^2 + a^2}}{c_1} + \frac{\sqrt{(d-x)^2 + b^2}}{c_2}\,.$$

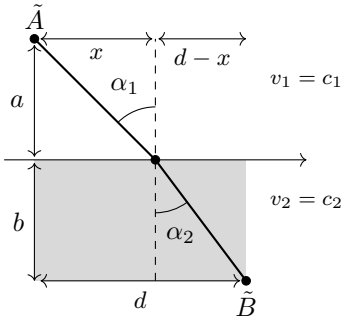


Figure 6.3.: Situation to choose X, like in [24]

To get the optimal point X^*, we have to minimize f, which leads us to a common

optimization problem, we may solve. We get

$$\begin{aligned} 0 = f'(x) &= \frac{1}{c_1} \cdot \frac{2x}{2\sqrt{x^2+a^2}} - \frac{1}{c_2} \cdot \frac{2(d-x)}{2\sqrt{(d-x)^2+b^2}} \\ &= \frac{1}{c_1} \cdot \frac{x}{s_1} - \frac{1}{c_2} \cdot \frac{d-x}{s_2} \\ &= \frac{\sin(\alpha_1)}{c_1} - \frac{\sin(\alpha_2)}{c_2}. \end{aligned}$$

This relation may be written as

$$\frac{\sin(\alpha_1)}{\sin(\alpha_2)} = \frac{c_1}{c_2}\,. \tag{6.3}$$

As $f''(x) > 0$ is valid for all x, solving this, leads to a global minimum.

We may apply this concept to the layers $y_0, \ldots, y_N$. As we started in the point $A = (0,0)$, the speed in the i-th medium is given by

$$c_i = \sqrt{2g}\sqrt{-y_i}\,.$$

Inserting this into Equation (6.3), we get

$$\frac{\sin(\alpha_i)}{\sin(\alpha_{i+1)}} = \frac{\sqrt{-y_i}}{\sqrt{-y_{i+1}}}\,.$$

For $\sin(\alpha_i)$ we may also write

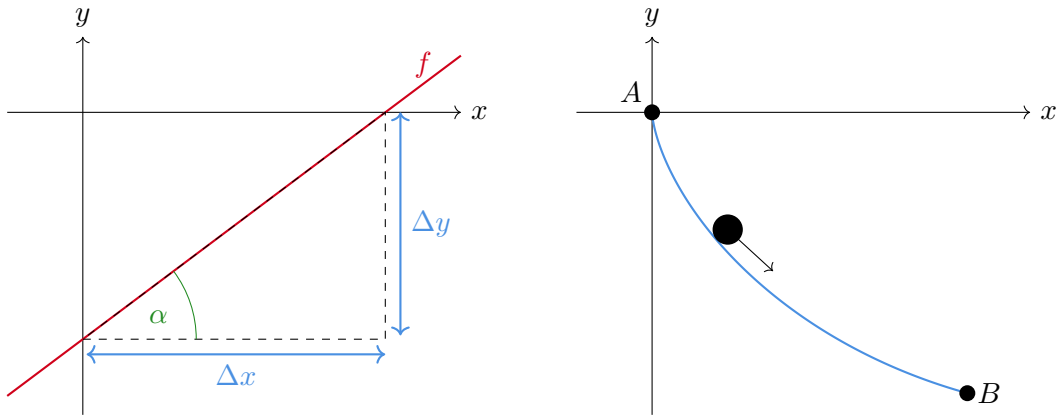
$$\sin(\alpha_i) = \frac{\Delta x_i}{\sqrt{\Delta x_i^2 + \Delta y_i^2}} = \frac{1}{\sqrt{1 + \left(\frac{\Delta y_i}{\Delta x_i}\right)^2}}$$

which leads us to

$$\begin{aligned} \frac{\sqrt{-y_i}}{\sqrt{-y_{i+1}}} &= \frac{\sqrt{1 + \left(\frac{\Delta y_{i+1}}{\Delta x_{i+1}}\right)^2}}{\sqrt{1 + \left(\frac{\Delta y_i}{\Delta x_i}\right)^2}} \\ \Leftrightarrow \sqrt{-y_{i+1}}\sqrt{1 + \left(\frac{\Delta y_{i+1}}{\Delta x_{i+1}}\right)^2} &= \sqrt{-y_i}\sqrt{1 + \left(\frac{\Delta y_i}{\Delta x_i}\right)^2}. \end{aligned}$$

As the right side just depends on variables in $i+1$ and the left side just on variables in i, we know that

$$\sqrt{-y_i}\sqrt{1 + \left(\frac{\Delta y_i}{\Delta x_i}\right)^2} = \text{const} > 0, \quad \forall i = 1, \ldots, N\,.$$



(a) Relation slope – tangens (b) Optimal Brachistochrone

Figure 6.4.: Illustration of the described theory

For $N \to \infty$, respectively $\Delta x_i \to 0$, $\Delta y_i \to 0$, we get

$$\begin{aligned} \text{const} =& \sqrt{-y(x)}\sqrt{1+y'(x)^2} := C > 0, \quad \forall x > 0 \\ \Leftrightarrow & y(x)\left[1+y'(x)^2\right] = -C^2 =: -2r\,, \end{aligned}$$

which is an ordinary differential equation. (cf. [24, pp. 26 – 28])

The solution of this differential equation may be gained in its parametric shape. First, for an easier notation, we write $K = -2r$ and get

$$y\left[1+y'^2\right] = -2r = K\,.$$

As it is known and also illustrated again in Figure 6.4 (a), the slope may be written as

$$y' = \tan(\alpha)\,.$$

As $\tan(\alpha) = \cot(90-\alpha)$ with $\varphi = 90 - \alpha$, we may also write

$$y' = \cot(\varphi)\,.$$

Moreover, we know

$$1 + \cot^2(\varphi) = \frac{1}{\sin^2(\varphi)}\,,$$

which leads us to

$$\begin{aligned} y\left[1+y'^2\right] = K \Leftrightarrow & y\left[1+\cot^2(\varphi)\right] = K \\ \Leftrightarrow & y\left[\frac{1}{\sin^2(\varphi)}\right] = K \\ \Leftrightarrow & y = K\sin^2(\varphi)\,. \end{aligned}$$

By differentiating this expression, we get

$$\begin{aligned} y' &= 2K\sin(\varphi)\cos(\varphi)\varphi' \\ \Leftrightarrow \cot(\varphi) &= 2K\sin(\varphi)\cos(\varphi)\varphi' \,. \end{aligned}$$

As $\sin(x)\cos(x) = \cot(x)\sin^2(x)$, this is equivalent to

$$\begin{aligned} \cot(\varphi) &= 2K\cot(\varphi)\sin^2(\varphi)\varphi' \\ \Leftrightarrow 1 &= 2K\sin^2(\varphi)\frac{\mathrm{d}\varphi}{\mathrm{d}x} \,. \end{aligned}$$

Moreover, we know that $\sin^2(x) = \frac{1}{2} - \frac{1}{2}\cos(2x)$, which brings us to

$$\begin{aligned} \mathrm{d}x &= 2K\left[\frac{1}{2} - \frac{1}{2}\cos(2\varphi)\right]\mathrm{d}\varphi \\ &= K\left[1 - \cos(2\varphi)\right]\mathrm{d}\varphi \,. \end{aligned}$$

We may integrate both sides and get

$$\begin{aligned} & \int 1\mathrm{d}x = \int K\left[1 - \cos(2\varphi)\right]\mathrm{d}\varphi \\ \Leftrightarrow \quad & x + C_1 = K\left[\varphi - \frac{1}{2}\sin(2\varphi)\right] + C_2 \\ \Leftrightarrow \quad & x = K\left[\varphi - \frac{1}{2}\sin(2\varphi)\right] + C \end{aligned}$$

and therefore

$$x(\varphi) = K\left[\varphi - \frac{1}{2}\sin(2\varphi)\right] + C \,.$$

As we start in $A = (0, 0)$, we know that $x(0) = 0$ and therefore $C = 0$. Now, using the definition of y and knowing that $\sin^2(x) = \frac{1}{2} - \frac{1}{2}\cos(x)$, we get

$$\begin{aligned} y &= K\sin^2(\varphi) \\ &= K(\frac{1}{2} - \frac{1}{2}\cos(2\varphi)) = \frac{K}{2}(1 - \cos(2\varphi)) \,. \end{aligned}$$

Defining $t = 2\varphi$ and using $K = -2r$, we have

$$\begin{aligned} x &= K(\varphi - \frac{1}{2}\sin(2\varphi)) \\ &= \frac{K}{2}(2\varphi - \sin(2\varphi) = \frac{K}{2}(t - \sin(t)) \\ &= -r(t - \sin(t)) \end{aligned}$$

and

$$\begin{aligned} y &= \frac{K}{2}(1 - \cos(2\varphi)) \\ &= \frac{K}{2}(1 - \cos(t)) = -r(1 - \cos(t)) \,. \end{aligned}$$

Therefore, the optimal solution in its parametric shape is given by

$$x(t) = -r(t - \sin(t)), \quad y(t) = -r(1 - \cos(t))\,.$$

Letting the ball run into the positive x direction, leads to the more intuitive way

$$x(t) = r(t - \sin(t)), \quad y(t) = -r(1 - \cos(t))\,,$$

which is shown in Figure 6.4 (b). For $t \in [0, T]$ the $T > 0$ and the $r > 0$ must be chosen as as $(x(T), y(T)) = B$. Hereby, r represents the radius of the cycloid. [30, pp. 385 –387]

The solution chosen by Johann Bernoulli seems very elegant. But it requires many steps, and the method cannot be applied to many problems. This is the reason why a general way to handle the optimization of functionals is needed and was invented a few years later. We will get to know this general way and then come back to the brachistochrone problem.

6.2. Theory on using Variational Analysis for Optimization

In this section, the theory needed to understand how to optimize a functional using variational analysis is introduced. Unless otherwise stated, the definitions, theorems and proofs follow those in the book of Hansjörg Kielhöfer's [18]. To keep things short, explicit citations will be provided only for deviations or additional references not found in this source.

6.2.1. Basic Theory on Variational Analysis

We want to focus on functionals of the shape

$$J(f) = \int_a^b L(x, f, f')\,\mathrm{d}x\,, \tag{6.4}$$

whereby

$$L : [a, b] \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$$

is the Lagrangian function and assumed to be continuous on $[a, b] \times \mathbb{R} \times \mathbb{R}$. The shape of the functional J may be used to cover the cost function, appearing in Witsenhausen's counterexample, later.

We have to assume a few things. Let $J : D \subset X \to \mathbb{R}$ be a functional and X a normed vector space. Moreover, we assume that for $f \in D$ and for $\eta \in X$ fixed also $y + h\eta \in D$, whereby $h \in (-\varepsilon, \varepsilon) \subset \mathbb{R}$. Last, we define $g : (\varepsilon, \varepsilon) \to \mathbb{R}$ as $g(h) = J(f + h\eta)$.

Definition 6.2.1 (*Gâteaux Differentiability*)

A functional J is Gâteaux differentiable in $y \in D$ in direction $\eta \in X$ if the expression

$$g'(0) = \frac{\mathrm{d}}{\mathrm{d}h} J(f + h\eta)\Big|_{h=0} = \lim_{h \to 0} \frac{J(f + h\eta) - J(f)}{h}$$

exists in $\mathbb{R}$. *The derivative* $g'(0)$ *is called* $\mathrm{d}J(f,\eta)$.

Knowing Gâteaux differentiability, we are now able to define the first variation. A separate definition is necessary as the linearity of $\mathrm{d}J$ is not mandatory.

Definition 6.2.2 (*First Variation*)

If $\mathrm{d}J(f,\eta)$ *exists in* $\mathbb{R}$ *for* $f \in D \subset X$, $\eta \in X$ *and* $\mathrm{d}J(f,\eta)$ *is linear in* η, $\mathrm{d}J(f,\eta)$ *is called first variation. In this case, we write*

$$\mathrm{d}J(f,\eta) = \delta J(f)\eta.$$

If this is valid for all $\eta \in X_0 \subset X$, *whereby* X_0 *is a subvector space of* X, *the mapping*

$$\delta J(f) : X_0 \to \mathbb{R}$$

is a linear functional.

As we know the first variation, we are now able to define the functional derivative.

Definition 6.2.3 (*Functional Derivative*)

If a functional $\frac{\delta J}{\delta f}$ *fits the condition*

$$\mathrm{d}J(f,\eta) = \int_a^b \frac{\delta J(f)}{\delta f(x)}(x)\eta(x)\,\mathrm{d}x\,,$$

it is called the functional derivative of J *in* f. *[11]*

We will now obtain an important theorem that will help us to explicitly determine the first variation of a functional. But before, we need to define function sets and norms to work on them.

Definition 6.2.4

In the following, we use the terms

$$\begin{aligned} C^{pw}[a,b] &= \{f \mid f : [a,b] \to \mathbb{R},\, f \in C[x_{i-1},x_i],\, i = 1,\dots,m\},\\ C^{1,pw}[a,b] &= \{f \,|\, f \in C^1[x_{i-1},x_i],\, i = 1,\dots,m\}, \quad a = x_0 < x_1 < \cdots < x_m = b,\\ C_0^{1,pw}[a,b] &= \{f \in C^{1,pw}[a,b] \,|\, f(a) = 0,\, f(b) = 0\}\,, \end{aligned}$$

with $a = x_0 < x_1 < \cdots < x_m = b$.

Remark 6.2.1

Attentive readers may have recognized, that the definition of the function space $C^{pw}[a,b]$ *is mathematically not completely precise. If a mapping* $f \in C^{pw}[a,b]$ *has a jump at one of the positions* x_i, $i = 1,\dots,m$, f *does not define a function. To keep things simple, we ignore this fact, as it does not affect the further work.*

Definition 6.2.5

From now on, for the norms we use the terms

$$\begin{aligned}\|f\|_0 = \|f\|_{0,[a,b]} &= \max_{x\in[a,b]} |f(x)|, \quad f \in C^0[a,b] \\ \|f\|_1 = \|f\|_{1,[a,b]} &= \|f\|_{0,[a,b]} + \|f'\|_{0,[a,b]}, \quad f \in C^1[a,b] \\ \|f\|_{1,pw} = f\|_{1,pw,[a,b]} &= \|f\|_{0,[a,b]} + \max_{i\in\{1,\dots,m\}} \{\|f'\|_{0,[x_{i-1},x_i]}\}, \quad f \in C^{1,pw}[a,b]\,.\end{aligned}$$

Theorem 6.2.1

Let the functional

$$J(f) = \int_a^b L(x,f,f')\,\mathrm{d}x$$

be defined on $D \subset C^{1,pw}[a,b]$. Moreover, we assume that the Lagrangian function L is continuous on $[a,b]\times\mathbb{R}\times\mathbb{R}$ and partially differentiable in f and f'. For all $\eta \in C_0^{1,pw}[a,b]$ and $y \in D$ we assume that $f + h\eta \in D$ for $h \in (-\varepsilon,\varepsilon) \subset \mathbb{R}$ which may depend on η. Then, for all $f \in D$ and all $\eta \in C_0^{1,pw}[a,b]$ the Gâteaux derivative exists and is given by

$$\delta J(f)\eta = \int_a^b L_f(x,f,f')\eta + L_{f'}(x,f,f')\eta'\,\mathrm{d}x\,,$$

where L_f, $L_{f'}$ name the partial derivatives.

Proof: To proof the statement, we first have to assume that

$$\lim_{h\to 0} \frac{L(x,f+h\eta,f'+h\eta') - L(x,f,f')}{h}$$

converges uniformly against the derivative. We will proof that assumption later. First, we want to determine against what the expression converges. To do this, we may write it as

$$m(h) = (L\circ g)(h)$$

with $L : \mathbb{R}^3 \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}^3$, $g(h) = (x, f(x) + h\eta(x), f'(x) + h\eta'(x))$. This enables us to write the limit as

$$\frac{\mathrm{d}}{\mathrm{d}h} m(h)\big|_{h=0} = m'(0)\,.$$

The multidimensional chain rule tells us that the derivative is given by the matrix product

$$m'(h) = J_{L\circ g}(h) = J_L(L(g(h))) \cdot J_g(h)$$

[10, p. 304]. With ∂_i as the partial derivative with respect to the i-th component and g_i as the i-th component of the function g, we may determine the Jacobians as

$$\begin{aligned}J_{L\circ g}(h) &= (\partial_1 L(g(h)), \partial_2 L(g(h)), \partial_3 L(g(h))) \\ J_g(h) &= (g_1'(h), g_2'(h), g_3'(h))^\top = (0, \eta(x), \eta'(x))^\top\,.\end{aligned}$$

This leads us to

$$\begin{aligned} m'(0) &= \partial_1 L(g(h)) \cdot 0 + \partial_2 L(g(h)) \cdot \eta(x) + \partial_3 L(g(h)) \cdot \eta'(x) \\ &= L_f(x, f(x), f'(x))\eta(x) + L_{f'}(x, f(x), f'(x))\eta'(x). \end{aligned}$$

Considering our assumption that this limit converges uniformly, we are allowed to swap the order of integration and determining the limit [10, p. 67] and obtain

$$\begin{aligned} \lim_{h\to 0} \frac{J(f+h\eta) - J(f)}{h} &= \lim_{h\to 0} \sum_{i=1}^{m} \int_{x_{i-1}}^{x_i} \frac{1}{h}\big(L(x, f(x) + h\eta(x), f'(x) + h\eta'(x)) \\ &\quad - L(x, f(x), f'(x))\big)\,\mathrm{d}x \\ &= \sum_{i=1}^{m} \int_{x_{i-1}}^{x_i} \lim_{h\to 0} \frac{1}{h}\big(L(x, f(x) + h\eta(x), f'(x) + h\eta'(x)) \\ &\quad - L(x, f(x), f'(x))\big)\,\mathrm{d}x \\ &= \sum_{i=1}^{m} \int_{x_{i-1}}^{x_i} L_f(x, f(x), f'(x))\eta(x) + L_{f'}(x, f(x), f'(x))\eta'(x)\,\mathrm{d}x \\ &= \int_a^b L_f(x, f(x), f'(x))\eta(x) + L_{f'}(x, f(x), f'(x))\eta'(x)\,\mathrm{d}x\,. \end{aligned}$$

To show that the term really converges uniformly we look at the expression where the limit is applied and a *clever zero* is added, which leads to

$$\begin{aligned} &\frac{1}{h}[L(x,f + h\eta, f' + h\eta') - L(x, f, f')] \qquad (6.5) \\ &\quad = \frac{1}{h}\int_0^h \frac{\mathrm{d}}{\mathrm{d}s}\left[L(x, f + s\eta, f' + s\eta') - L(x, f, f')\right]\,\mathrm{d}s \\ &\quad = \frac{1}{h}\int_0^h L_f(x, f + s\eta, f' + s\eta')\eta(x) + L_{f'}(x, f + s\eta, f' + s\eta')\eta'(x)\,\mathrm{d}s \\ &\quad = \frac{1}{h}\int_0^h L_f(x, f + s\eta, f' + s\eta')\eta(x) + L_{f'}(x, f + s\eta, f' + s\eta')\eta'(x)\,\mathrm{d}s \\ &\quad + L_f(x, f, f')\eta(x) + L_{f'}(x, f, f')\eta'(x) \\ &\quad - \frac{1}{h}\int_0^h L_f(x, f, f')\eta(x) + L_{f'}(x, f, f')\eta'(x)\,\mathrm{d}s\ \eta(x) \\ &\quad = L_f(x, f, f')\eta(x) + L_{f'}(x, f, f')\eta'(x) \\ &\quad + \frac{1}{h}\int_0^h L_f(x, f + s\eta, f' + s\eta') - L_f(x, f, f')\,\mathrm{d}s\ \eta(x) \qquad (6.6) \\ &\quad + \frac{1}{h}\int_0^h L_{f'}(x, f + s\eta, f' + s\eta') - L_{f'}(x, f, f')\,\mathrm{d}s\ \eta'(x)\,. \end{aligned}$$

As we consider $h \to 0$, we may restrict s to the interval $\left[-\frac{\varepsilon}{2}, \frac{\varepsilon}{2}\right]$. As $[x_{i-1}, x_i]$ is compact, f, η, f', η' continuous and therefore also their linear combinations, we may conclude, using the Extreme Value Theorem [14, p. 226], that these linear combinations are limited and therefore

$$\begin{aligned} &f(x) + s\eta(x) \in [-c, c], \\ &f'(x) + s\eta'(x) \in [-c', c'], \quad s \in \left[-\frac{\varepsilon}{2}, \frac{\varepsilon}{2}\right], \quad x \in [x_{i-1}, x_i], \quad i = 1, \ldots, m, \end{aligned}$$

with $c, c' \in \mathbb{R}^+$. Using this, we may obtain

$$\left\{(x, f(x)+s\eta(x), f'(x)+s\eta'(x)) \,\middle|\, x \in [x_{i-1}, x_i],\, s \in \left[-\frac{\varepsilon}{2}, \frac{\varepsilon}{2}\right]\right\} \subset [x_{i-1}, x_i] \times [-c, c] \times [-c', c'].$$

As this area is compact, we may conclude from the continuity of L_F and $L_{f'}$ that they are uniformly continuous as well. This leads us, for $x \in [x_{i-1}, x_i]$ and $\tilde{\varepsilon} > 0$, to the estimation

$$|L_f(x, f(x)+s\eta(x), f'(x)+s\eta'(x)) - L_f(x, f(x), f'(x))| < \tilde{\varepsilon}$$

if

i) $|s|(|\eta(x)| + |\eta'(x)|) < \delta(\tilde{\varepsilon})$

ii) $|s| < \frac{\varepsilon}{2}$

are valid. Using the norm, we can say

$$|\eta(x)| + |\eta'(x)| \leq \|\eta\|_{0,[x_{i-1},x_i]} + \|\eta'\|_{0,[x_{i-1},x_i]} \leq \|\eta\|_{1,pw,[a,b]} = \|\eta\|_{1,pw}.$$

The conditions i) and ii) are fulfilled, if $|s| < \min\left\{\frac{\varepsilon}{2}, \frac{\delta(\tilde{\varepsilon})}{|\eta(x)|+|\eta'(x)|}\right\}$ and therefore especially when

$$|s| < \min\left\{\frac{\varepsilon}{2}, \frac{\delta(\tilde{\varepsilon})}{\|\eta\|_{1,pw}}\right\}.$$

As the same estimation is valid for $L_{f'}$, this leads us to an estimation for Equation (6.6). We obtain

$$\begin{aligned}
&\frac{1}{h}[L(x,f(x)+h\eta(x), f'(x)+h\eta(x)') - L(x, f(x), f'(x))] \\
&\quad = L_f(x, f(x), f'(x))\eta(x) + L_{f'}(x, f(x), f'(x))\eta'(x) \\
&\quad + \frac{1}{h}\int_0^h L_f(x, f(x)+s\eta(x), f'(x)+s\eta'(x)) - L_f(x, f(x), f'(x))\,\mathrm{d}s\ \eta(x) \\
&\quad + \frac{1}{h}\int_0^h L_{f'}(x, f(x)+s\eta(x), f'(x)+s\eta'(x)) - L_{f'}(x, f(x), f'(x))\,\mathrm{d}s\ \eta'(x) \\
&\quad \leq L_f(x, f(x), f'(x))\eta(x) + L_{f'}(x, f(x), f'(x))\eta'(x) \\
&\quad + \frac{1}{h}\int_0^t \tilde{\varepsilon}\,\mathrm{d}s\,\eta(x) + \frac{1}{h}\int_0^t \tilde{\varepsilon}\,\mathrm{d}s\ \eta'(x) \\
&\quad = L_f(x, f, f')\eta(x) + L_{f'}(x, f, f')\eta'(x) + \tilde{\varepsilon}(\eta(x) + \eta'(x))
\end{aligned}$$

for $0 < h < \min\left\{\frac{\varepsilon}{2}, \frac{\delta(\tilde{\varepsilon}}{\|h\|_{1,pw}}\right\}$. Using the definition of Equation (6.6), we get

$$\begin{aligned}
&\left|\frac{1}{h}(L(x, f(x)+h\eta(x), f'(x)+h\eta'(x)) - L(x, f(x), f'(x))\ ) \right. \\
&\qquad \left. - (L_f(x, f(x), f'(x))\eta(x) + L_{f'}(x, f(x), f'(x))\eta'(x))\right| \\
&\qquad \leq \tilde{\varepsilon}(\eta(x) + \eta'(x)) \leq \tilde{\varepsilon}\|\eta\|_{1,pw} < \hat{\varepsilon} \in \mathbb{R}^+
\end{aligned}$$

for $0 < \tilde{\varepsilon} < \frac{\hat{\varepsilon}}{\|\eta\|_{1,pw}}$. This fits the definition of the uniformly convergence, which means, that

$$\begin{aligned}
&\lim_{t\to 0}\left(\frac{1}{t}(L(x, f(x)+h\eta(x), f'(x)+h\eta'(x)) - L(x, f(x), f'(x)))\right) \\
&= L_f(x, f(x), f'(x))\eta(x) + F_{f'}(x, f(x), f'(x))\eta'(x), \quad x \in [x_{i-1}, x_i],\, i = 1, \ldots, m
\end{aligned}$$

converges uniformly. Therefore, the assumption from the beginning of the proof is valid. ■

6.2.2. The Euler-Lagrange Equation

As we have introduced the basic idea of variational analysis, we want to use it to determine criteria for extrema of a functional. One main result is the Euler-Lagrange-Equation. The criterion to search for an extremum is based on it. In the beginning of this subsection, we will give a definition on how we expect an extremum of a functional to look like and then look for criteria.

We now want to define the local minimum of a functional.

Definition 6.2.6 (*Local Minimizer of a Functional*)

A function $f \in D \subset C^{1,pw}[a,b]$ represents a local minimizer of a functional J, if

$$J(f) \leq J(\tilde{f}), \quad \forall \tilde{f} \in D \text{ with } \|f - \tilde{f}\|_{1,pw} < d,$$

with $d > 0$ fixed.

To see the analogy to classical optimization problems on functions in $\mathbb{R}$, we include another result from a lecture of Annette A'Campo-Neuen [2].

Theorem 6.2.2

Let $J : D \to \mathbb{R}$, $D \subset C^{1,pw}[a,b]$, be a functional which is Gâteaux differentiable on complete D. If $f \in D$ is a local minimizer or a local maximizer the equation

$$\mathrm{d}J(f,\eta) = 0$$

must be valid for all $\eta \in C_0^1[a,b]$.

Proof: We may fix $f \in D$, $\eta \in C_0^1[a,b]$. Then, we just have to look at the function $g : (-\varepsilon, \varepsilon) \to \mathbb{R}$ from the introduction. This leads us to

$$\mathrm{d}J(f,\eta) = 0 \Leftrightarrow g'(0) = 0.$$

$g'(x_0) = 0$ exactly fits the necessary condition for a local extremum of g in $x_0 = 0$. ■

As solving the equation $\mathrm{d}J(f,\eta) = 0$ for an infinite number of functions η is no option, we have to look for an alternative criterion. This will guide us to the Euler-Lagrange-Equation.

To be able to proof the Euler-Lagrange-Equation, we first need a couple of auxiliary results.

Lemma 6.2.1

Let $f \in C^{pw}[a,b]$. If it is valid that

$$\int_a^b f(x)\eta(x)\,\mathrm{d}x = 0 \tag{6.7}$$

for all $\eta \in C_0^\infty[a,b]$, we may conclude that $f(x) = 0$ for all $x \in [a,b]$.

Proof: We assume that there is a function $f \in C^{pw}[a,b]$, $f(x) \neq 0$ for a specific $x \in [a,b]$, that fits Equation (6.7). Because of the continuity of f, there is an open interval I with

$$f(x) \neq 0, \quad \forall x \in I.$$

Now, we choose $\eta \in C_0^{pw}[a,b]$, as it fits $\operatorname{supp}(f) \subset I$ and

$$\operatorname{sgn}(f(x)) = \operatorname{sgn}(\eta(x)), \quad \forall x \in \operatorname{supp}(f)\,,$$

whereby

$$\operatorname{supp}(f) = \{x \in [a,b] \,|\, f(x) \neq 0\}.$$

Assuming this, we know

$$f(x)\eta(x) \geq 0, \quad \forall x \in [a,b]$$

and $f(x)\eta(x)$ continuous. Equation (6.7) then implies that $f(x)\eta(x) = 0$ on $[a,b]$. This is a contradiction as we assumed that there are $x \in I$ wherefore $\eta(x) \neq 0$ and $f(x) \neq 0$. ■

Lemma 6.2.2

Let $f \in C^{pw}[a,b]$. If the equation

$$\int_a^b f(x)\eta'(x)\,\mathrm{d}x = 0$$

is fulfilled for all $\eta \in C_0^{1,pw}[a,b]$, we may conclude that $f(x) = c$, $c \in \mathbb{R}$ for all $x \in [a,b]$.

Proof: We choose the $c \in \mathbb{R}$ as

$$c = \frac{1}{b-a}\int_a^b f(x)\,\mathrm{d}x = \frac{1}{b-a}\sum_{i=1}^{m}\int_{x_{i-1}}^{x_i} f(x)\,\mathrm{d}x$$

and

$$\eta(x) = \int_a^x (f(s) - c)\,\mathrm{d}s.$$

Then, we know that $\eta \in C[a,b]$, $\eta(a) = \eta(b) = 0$ and

$$\eta'(x) = f(x) - c\,,$$

whereby on the borders the one sided derivatives have to be used. This implies that $\eta \in C_0^1[a,b]$. Using the assumption from this Lemma, it must be valid that

$$\begin{aligned}\int_a^b (f(x) - c)\eta'(x)\,\mathrm{d}x &= \int_a^b f(x)\eta'(x)\,\mathrm{d}x - c\int_a^b \eta'(x)\,\mathrm{d}x \\ &= 0 - c\,(\eta(b) - \eta(a)) = 0\,.\end{aligned}$$

Then, we know that

$$\int_a^b (f(x) - c)\eta'(x)\,\mathrm{d}x = \int_a^b f(x)\eta'(x)\,\mathrm{d}x\,.$$

Using the way we constructed our η, we obtain

$$\begin{aligned}\int_a^b f(x)\eta'(x)\,\mathrm{d}x &= \int_a^b (f(x)-c)\eta'(x)\,\mathrm{d}x = \int_a^b (f(x)-c)(f(x)-c)\,\mathrm{d}x \\ &= \int_a^b (f(x)-c)^2\,\mathrm{d}x = \sum_{i=1}^{m}\int_{x_{i-1}}^{x_i}(f(x)-c)^2\,\mathrm{d}x\,.\end{aligned}$$

This shows that $f(x) = c \in \mathbb{R}$ for $x \in [a,b]$. ■

Lemma 6.2.3 (*Fundamental Lemma of Calculus of Variations*)

Let $f, g \in C^{pw}[a,b]$. If now

$$\int_a^b f(x)\eta(x) + g(x)\eta'(x)\,\mathrm{d}x = 0$$

is valid for all $\eta \in C_0^{1,pw}[a,b]$, we know $g \in C^{1,pw}[a,b] \subset C[a,b]$ and $g' = f$ piecewise.

Proof: We choose

$$F(x) = \int_a^x f(s)\,\mathrm{d}s.$$

Then $F \in C[a,b]$ and for $x \in [x_{i-1}, x_i]$, $i = 1, \ldots, m$ it is valid that

$$F'(x) = f(x),$$

whereby the one sided derivatives have to be taken on the boundaries. Therefore, we know that $F \in C^1[a,b]$. Using partial integration, we obtain

$$\begin{aligned}\int_a^b f(x)\eta(x)\,\mathrm{d}x &= \int_a^b F'(x)\eta(x)\,\mathrm{d}x \\ &= F(x)\eta(x)\Big|_a^b - \int_a^b F(x)\eta'(x)\,\mathrm{d}x \\ &= (F(b)\eta(b) - F(a)\eta(a)) - \int_a^b F(x)\eta'(x)\,\mathrm{d}x \\ &= 0 - \int_a^b F(x)\eta'(x)\,\mathrm{d}x \\ &= -\int_a^b F(x)\eta'(x)\,\mathrm{d}x \quad ,\forall \eta \in C_0^{1,pw}[a,b].\end{aligned}$$

Then, it is true that

$$\begin{aligned}\int_a^b f(x)\eta(x) + g(x)\eta'(x)\,\mathrm{d}x &= \int_a^b f(x)\eta(x)\,\mathrm{d}x + \int_a^b g(x)\eta'(x)\,\mathrm{d}x \\ &= -\int_a^b F(x)\eta'(x)\,\mathrm{d}x + \int_a^b g(x)\eta'(x)\,\mathrm{d}x \\ &= \int_a^b (g(x) - F(x))\eta'(x)\,\mathrm{d}x, \quad \forall \eta \in C_0^{1,pw}[a,b].\end{aligned}$$

As $g \in C^{pw}[a,b]$ and therefore also $g - F \in C^{pw}[a,b]$ we may apply Lemma 6.2.2 to this expression and obtain

$$g(x) - F(x) = c \Leftrightarrow g(x) = c + F(x).$$

Per construction, we see that $g \in C^{1,pw}[a,b]$ and get

$$g'(x) = f(x)$$

piecewise. ■

Remark 6.2.2

The piecewise equality of f and g' may be understood as that for $f \in C[x_{i-1}, x_i]$ follows that $g'(x) = (x)$ for $x \in [x_{i-1}, x_i]$, $i = 1, \ldots, m$.

Remark 6.2.3

Having a look on the proof of Lemma 6.2.3, we see that the statement $g \in C^{1,pw}[a,b]$ follows from the construction $g'(x) = f(x)$. Therefore, if $g, f \in C[a,b]$ and the assumption is valid, we may conclude that $g \in C^1[a,b]$ and therefore $g = f'$ on $[a,b]$.

Using all these Lemmas, we are able to proof that the Euler-Lagrange-Equation is a valid criterion for a function being a local minimizer or a local maximizer. We reach the final result.

Before that, we have to assume that, for $f \in D \subset C^{1,pw}[a,b]$, the first variation $\delta J(f)\eta$ exists for all $\eta \in C_0^{1,pw}[a,b]$. Then, we may conclude that for all $\eta \in C_0^{1,pw}[a,b]$ and for all $h \in (-\varepsilon, \varepsilon)$, $f + h\eta \in D$, whereby $\varepsilon > 0$ might depend on η. This can be seen as the first variation exists, wherefore $J(f + h\eta)$ must be defined. As $J : D \to \mathbb{R}$, this is just the case if $f + h\eta \in D$.

Theorem 6.2.3 (*Euler-Lagrange Equation*)

If $f \in D \subset C^{1,pw}[a,b]$ is a local minimizer for the functional defined in Equation 6.4 and the Lagrange equation $L : [a,b] \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is continuous as well as continuously differentiable in the two last variables, we know

i) $L_{f'}(\cdot, f, f') \in C^{1,pw}[a,b]$

ii) $\dfrac{\mathrm{d}}{\mathrm{d}x} L_{f'}(\cdot, f, f') - L_f(\cdot, f, f') = 0$ *piecewise on* $[a,b]$.

For $f \in C^1[a,b]$, we get $L_{f'}(\cdot, f, f') \in C^1[a,b]$ and $\frac{\mathrm{d}}{\mathrm{d}x} L_{f'}(\cdot, f, f') - L_f(\cdot, f, f') = 0$ on $[a,b]$.

Proof: For sufficiently small $h > 0$, it must be valid that

$$\|h\eta\| < d, \quad h \in (-\varepsilon, \varepsilon),\, d > 0\,.$$

If f is a local minimizer, the function

$$g(t) = J(f + t\eta), \quad t \in \mathbb{R}$$

must have a local minima in $t = 0$ for all $\eta \in C_0^{1,pw}[a,b]$. As shown in Theorem 6.2.1, the Gâteaux-differential exists and as seen in the proof of Theorem 6.2.2,

$$g'(0) = 0$$

must be valid. This leads us to

$$\delta J(f)\eta = \int_a^b L_f(x,f,f')\eta(x) + L_{f'}(x,f,f')\eta'(x)\,\mathrm{d}x = 0, \quad \forall \eta \in C_0^{1,pw}[a,b]\,.$$

Using Lemma 6.2.3 we conclude

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}x} L_{f'}(\cdot,f,f') = L_f(\cdot,f,f') \\ \Leftrightarrow &\frac{\mathrm{d}}{\mathrm{d}x} L_{f'}(\cdot,f,f') - L_f(\cdot,f,f') = 0\,. \end{aligned}$$

The lemma also says that $L_{f'}(\cdot,f,f') \in C^{1,pw}[a,b]$. The statement that, for $f \in C^1[a,b]$, $g'(x) = f(x)$ for $x \in [a,b]$, immediately follows from Remark 6.2.3. ■

6.2.3. Second Order Condition

In the previous subsection, we derived a necessary criterion to decide whether a point may be a local minimum or maximum. As we are looking for local minimizers, we need a second condition that ensures, we really obtained a local minimizer.

Before we define such a criterion, we need to obtain the second variation.

Definition 6.2.7 (*Second variation*)

Let the functional

$$J(f) = \int_a^b L(x,f,f')\,\mathrm{d}x$$

be defined on $D \subset C^{1,pw}[a,b]$, whereby L is two times partially continuous differentiable in f and f'. If for all $f \in D \subset C^{1,pw}[a,b]$, $\eta \in C_0^{1,pw}[a,b]$ and $h \in (-\varepsilon,\varepsilon)$ the function $g(h) = J(f+h\eta)$ is defined, then

$$\frac{\mathrm{d}^2}{\mathrm{d}h^2} g(h)\bigg|_{h=0} = \delta^2 J(f)(\eta,\eta)$$

is the second variation of J at f in direction η. For the given assumptions it is valid that

$$\delta^2 J(f)(\eta,\eta) = \int_a^b L_{ff}(x,f,f')\eta(x)^2 + 2L_{ff'}(x,f,f')\eta(x)\eta(x)' + L_{f'f'}(x,f,f')\eta'(x)^2\,\mathrm{d}x\,.$$

Proof: The interchangeability of taking the limit and determining the integral may be shown as in Theorem 6.2.1. Therefore, this step is skipped. Knowing about this, we obtain, using the

multidimensional chain rule,

$$\begin{aligned}
&\frac{\mathrm{d}^2}{\mathrm{d}h^2}\int_a^b L(x, f+h\eta, f'+h\eta')\,\mathrm{d}x\bigg|_{h=0} = \int_a^b \frac{\mathrm{d}^2}{\mathrm{d}h^2} L(x, f+h\eta, f'+h\eta')\,\mathrm{d}x\bigg|_{h=0} \\
&= \int_a^b \frac{\mathrm{d}}{\mathrm{d}h}\left(L_f(x, f+h\eta, f'+h\eta')\eta(x) + L_{f'}(x, f+h\eta, f'+h\eta')\eta'(x)\right)\,\mathrm{d}x\bigg|_{h=0} \\
&= \int_a^b L_{ff}(x,f,f')\eta^2(x) + L_{ff'}(x,f,f')\eta(x)\eta'(x) \\
&+ L_{ff'}(x,f,f')\eta(x)\eta'(x) + L_{f'f'}(x,f,f')\eta'(x)^2\,\mathrm{d}x \\
&= \int_a^b L_{ff}(x,f,f')\eta^2(x) + 2L_{ff'}(x,f,f')\eta(x)\eta'(x) + L_{f'f'}(x,f,f')\eta'(x)^2\,\mathrm{d}x\,.
\end{aligned}$$

■

Knowing the second variation, we are able to obtain a criterion to check whether an extremum is a local maximum or a local minimum.

Theorem 6.2.4 (*Second-Variation Condition*)

If $f \in D \subset C^{1,pw}[a,b]$ is a local minimizer of a functional of the shape of Equation (6.4), it must be valid that

$$\delta^2 J(f) \geq 0\,,$$

for all $\eta \in C_0^{1,pw}[a,b]$. [6, p. 226]

Proof: Using Taylor's expansion around 0, we may write $g(h)$ as

$$\begin{aligned}
g(h) &= J(f+h\eta) \\
&= g(0) + hg'(0) + \frac{h^2}{2}g''(0) + \mathcal{O}(h^3)\,.
\end{aligned}$$

As we know $g'(0) = 0$ for a local extremum, we get

$$g(h) - g(0) = \frac{h^2}{2}g''(0) + \mathcal{O}(h^3)\,.$$

Then we know

$$\lim_{h\to 0}\frac{\mathcal{O}(h^3)}{h^2} = 0\,.$$

Using the knowledge above the limit and assuming $g''(0) \neq 0$, we get for sufficiently small $h > 0$

$$\left|\frac{\mathcal{O}(h^3)}{h^2}\right| < \frac{1}{2}\left|g''(0)\right| \Leftrightarrow \left|\mathcal{O}(h^3)\right| < \frac{1}{2}h^2\left|g''(0)\right|\,.$$

Therefore, we know that the sign of $g(h) - g(0)$ for sufficiently small $h > 0$ is given by

$$\operatorname{sgn}\left(g(h) - g(0)\right) = \operatorname{sgn}\left(\frac{h^2}{2}g''(0) + \mathcal{O}(h^3)\right) = \operatorname{sgn}\left(g''(0)\right)\,.$$

■

6.3. Solving the Problem of the Brachistochrone using Calculus of Variations

As we have introduced the calculus of variations in the previous section, we would like to use it, to solve the problem of the brachistochrone. We will do this in this section. In the first part, we will do this analytically in the second using a numerical way.

6.3.1. Analytical Solution

We saw in Theorem 6.2.3 that if a function minimizes a functional, it has to fit the Euler-Lagrange-Equation, which is given by

$$0 = L_y - \frac{\mathrm{d}}{\mathrm{d}x} L_{y'}.$$

In section 6.1.1, we derived as the cost functional

$$J(f) = \int_0^{x_B} \frac{1}{\sqrt{2 \cdot g}} \frac{\sqrt{1 + y'(x)^2}}{\sqrt{-y(x)}} \, \mathrm{d}x \, .$$

As this fits the shape of the cost functional in Equation 6.4, we know that in our problem the Lagrangian function L is given by

$$L(x, y, y') = \frac{1}{\sqrt{2 \cdot g}} \frac{\sqrt{1 + y'(x)^2}}{\sqrt{-y(x)}} \, .$$

Multiplying with y' and adding a clever zero, we see that the Euler-Lagrange Equation is equivalent to

$$\begin{aligned} 0 &= y'(L_y - \frac{\mathrm{d}}{\mathrm{d}x} L_{y'}) = y' L_y - y' \frac{\mathrm{d}}{\mathrm{d}x} L_{y'} \\ \Leftrightarrow 0 &= y' L_y + L_{y'} y'' - L_{y'} y'' - y' \frac{\mathrm{d}}{\mathrm{d}x} L_{y'} \, . \end{aligned}$$

As multidimensional chain rule tells us that

$$\frac{\mathrm{d}}{\mathrm{d}x} L = L_y \cdot y' + L_{y'} \cdot y''$$

and we know

$$\frac{\mathrm{d}}{\mathrm{d}x} L_{y'} \cdot y' = y' \frac{\mathrm{d}}{\mathrm{d}x} L_{y'} + y'' L_{y'} \, ,$$

from the product rule, we may write the expression as

$$\begin{aligned} 0 &= y' L_y + L_{y'} y'' - L_{y'} y'' - y' \frac{\mathrm{d}}{\mathrm{d}x} L_{y'} \\ \Leftrightarrow 0 &= \frac{\mathrm{d}}{\mathrm{d}x} \left(L - L_{y'} \cdot y' \right) \, . \end{aligned}$$

Therefore,

$$L - L_{y'}y' = \text{const} =: \tilde{C}\,.$$

Inserting the definition of L, we get

$$\begin{aligned}
\tilde{C} &= \frac{1}{\sqrt{2\cdot g}}\left[\frac{\sqrt{1+y'^2}}{\sqrt{-y}} - \frac{y'^2}{\sqrt{1+y'^2}\sqrt{-y}}\right] \\
\Leftrightarrow C &:= \sqrt{2\cdot g}\cdot\tilde{C} = \frac{\sqrt{1+y'^2}}{\sqrt{-y}} - \frac{y'^2}{\sqrt{1+y'^2}\sqrt{-y}} \\
\Leftrightarrow C &\cdot \sqrt{-y}\cdot\sqrt{1+y'^2} = (1+y'^2) - y'^2 = 1\,.
\end{aligned}$$

This leads us to

$$\sqrt{-y}\sqrt{1+y'^2} = \frac{1}{C} \Leftrightarrow -y(1+y'^2) = \frac{1}{C^2} =: 2r$$

and we get the differential equation

$$-y(x)\cdot(1+y'(x)^2) = 2r \Leftrightarrow y(x)(1+y'(x)^2) = -2r\,.$$

This is the same equation we derived in section 1 and had the parametric solution

$$x(t) = r(1-\sin(t)), \quad y(t) = -r(1-\cos(t))\,,$$

we already know. [30, pp. 384 – 386]

This solution is much shorter than the solution Bernoulli chose. Moreover, it is gained by just solving the conditions we obtained from the calculus of variations.

6.3.2. Numerical Solution

We saw in the analytical solution that solving the differential equation we obtained by solving the Euler-Lagrange equation is not trivial to solve and we have to switch to a parametric solution. Instead of doing this, we have a look on solving it numerically. Different to other papers like [3] or [22], that focused on directly minimizing the cost functional, we want to just look on the Euler-Lagrange-Equation, which, as you may see, simplifies the task.

Before we may solve the Euler-Lagrange Equation, we assume a few things. We assume that our coordinate system looks like shown in Figure 6.5. Doing this, the cost functional, we derived in the beginning, changes to

$$T(y) = \frac{1}{\sqrt{2g}}\int_0^{x_B}\frac{\sqrt{1+y'(x)^2}}{\sqrt{y(x)}}\,\mathrm{d}x\,.$$

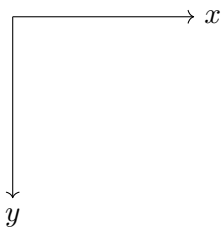


Figure 6.5.: Chosen CS

As $\frac{1}{\sqrt{2g}}$ just scales the term, we may ignore it to find an optimal solution. To obtain the Euler-Lagrange Equation, we have to identify the L from the cost functional. In our case we have

$$L(x, y, y') = \frac{\sqrt{1 + y'(x)^2}}{\sqrt{y(x)}}\,.$$

This leads to the Euler-Lagrange Equation

$$\begin{aligned}
\mathrm{EL}(L, y, y', x) &= \frac{\mathrm{d}}{\mathrm{d}x} L_{f'}(x, y, y') - L_y(x, y, y') = 0 \\
&\Leftrightarrow \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{y'(x)}{\sqrt{y'(x)^2 + 1} \cdot \sqrt{y(x)}}\right) + \frac{\sqrt{y'(x)^2 + 1}}{2 \cdot (y(x))^{\frac{3}{2}}} = 0 \\
&\Leftrightarrow \frac{y'(x)^2 + 2 \cdot y''(x) y(x) + 1}{2\left((y'(x)^2 + 1) y(x)\right)^{\frac{3}{2}}} = 0\,.
\end{aligned}$$

We may simplify this to

$$y'(x)^2 + 2 \cdot y''(x) y(x) + 1 = 0\,,$$

which is the equation we have to solve. As the expression has to be zero for all $x \in [0, x_B]$, we have to approximate the solution in $N > 0$ positions. Let the positions, where we approximate, be given by

$$0 = x_1 < x_2 < \cdots < x_N = x_B\,.$$

We name the approximation of the minimizing function

$$y(x_i) = y_i, \quad i = 0, \ldots, N\,,$$

whereby $y_0 = 0$, $y_N = y_B$. In order for some terms to cancel out, we choose for the approximation of the first derivative the forward difference quotient

$$y_i' = \frac{y_{i+1} - y_i}{h}$$

and for the second derivative the central difference quotient

$$y_i'' = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}\,.$$

As mentioned, several terms cancel out and we get the approximation of our Euler-Lagrange Equation in a position x_i with

$$\begin{aligned}
\frac{h^2 - 3y_i^2 + 2y_i y_{i-1} + y_{i+1}^2}{h^2} &= 0 \\
\Leftrightarrow h^2 - 3y_i^2 + 2y_i y_{i-1} + y_{i+1}^2 &= 0\,.
\end{aligned}$$

As we know y_0 and y_N and do not have to determine them, we may obtain a nonlinear equation system by

$$h^2 - 3y_i^2 + 2y_i y_{i-1} + y_{i+1}^2 = 0, \quad \forall i \in \{1, \ldots, N-1\}\,.$$

In our case, the system of equations is solved using the Python library `Scipy` and in particular using a function that uses the Krylov approximation for the inverse Jacobian

used in the function. This method suits the size of the system of equations and the case that we have a kind of tridiagonal shape.

Looking on the results, we first consider the problem, with

$$(x_A, y_A) = (0, 0), \quad (x_B, y_B) = (2, -1)\,.$$

Solving the equation system for $N = 100$, we get the results in Figure 6.6, whereby the dashed line is the exact solution and the solid line is the approximated function. For $N = 100$ the maximum residuum is 0.0225191. As to see in Figure 6.7 the approximation also works for different (x_B, y_B).

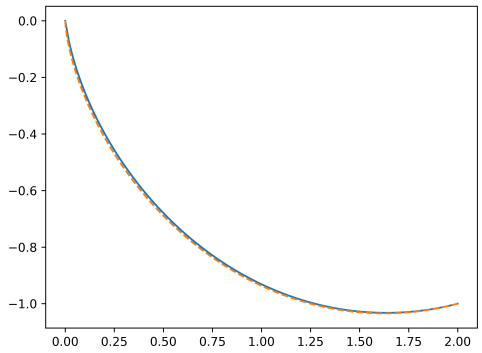


Figure 6.6.: Approximated Brachistochrone, dashed: exact, solid: approximated

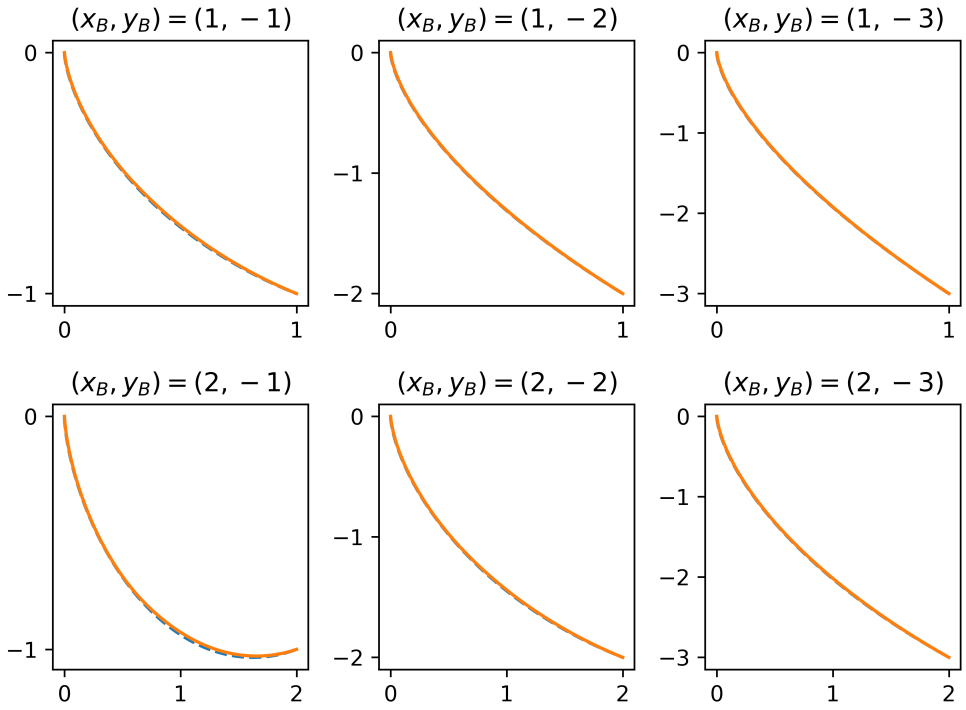


Figure 6.7.: Approximated Brachistochrones, dashed: exact, solid: approximation

7. A Variational Perspective on Witsenhausen's Counterexample

After introducing variational analysis in the previous sections and applying it on the well known example from Bernoulli, we will come back to the real world problem, we want to solve. In this section, we may see that Witsenhausen's counterexample may be seen as a typical problem from variational analysis.

7.1. From Variational Analysis to a numerical Criterion

Before we may use variational analysis to obtain an approximation for the optimal function, we need to show that Witsenhausen's counterexample may be seen as a problem in variational analysis. When this is done, we will start to derive a criterion, we may evaluate numerically.

7.1.1. Showing, Witsenhausen's Counterexample may be handled using Variational Analysis

First, we show that Witsenhausen's counterexample may be seen as part of problems from variational analysis. Therefore, we first look on the original problem, stated in the 60s of the last millennium.

Before we do this, we introduce a new notation. From now on

$$E_{z_1,z_2,\dots}[\cdot]$$

means the expected value of $\cdot$ for the random variables $z_1, z_2, \dots$. This notation is used, as in this chapter, not necessarily all random variables, are meant by an expected value.

Remark 7.1.1

For the random variables $x \sim \mathcal{N}(0,\sigma^2)$, $\nu \sim \mathcal{N}(0,1)$ and $k > 0$, the cost functional derived by Witsenhausen is given by

$$J(f,g) = E_{x,\nu}[k^2(f(x)-x)^2 + (f(x) - g(f(x)+\nu))^2]\,.$$

Knowing for a fixed f the optimal g_f^*, we obtain the cost functional by

$$J(f) = E_{x,\nu}[k^2(f(x)-x)^2 + (f(x) - g_f^*(f(x)+\nu))^2]\,.$$

One may ask, how this may be seen as a shape of a functional seen in the theory of variational analysis. The following lemma will help us to derive such a form.

Lemma 7.1.1 (*Witsenhausen's cost Functional fits needs for Variational Analysis*)

For bounded f and g, the cost functional

$$J(f,g) = E_{x,\nu}[k^2(f(x)-x)^2 + (f(x) - g(f(x)+\nu))^2]$$

introduced by Witsenhausen, may be approximated up to a precision $\varepsilon > 0$ by

$$\tilde{J}(f,g) = \int_{x_l(\varepsilon)}^{x_u(\varepsilon)} f_x(x) \left(k^2(f(x)-x)^2 + E_\nu[(f(x) - g(f(x)+\nu))^2]\right) \mathrm{d}x\,,$$

whereby $x_l(\varepsilon)$, $x_u(\varepsilon)$ determine the needed borders such as $|\tilde{J}(f) - J(f)| < \varepsilon$ and f_x means the density of the random variable x.

Proof: As $x \sim \mathcal{N}(0,\sigma^2)$, with density f_x, we know

$$J(f,g) = \int_{-\infty}^{+\infty} f_x(x) \left(k^2(f(x)-x)^2 + E_\nu[(f(x) - g(f(x)+\nu))^2]\right) \mathrm{d}x\,.$$

Moreover, from the properties of the normal distribution, we may conclude that $\lim_{|x|\to\infty} f_x(x) = 0$. Using the properties of the integral, we get

$$\begin{aligned}
J(f,g) &= \int_{-\infty}^{+\infty} f_x(x) \left(k^2(f(x)-x)^2 + E_\nu[(f(x) - g(f(x)+\nu))^2]\right) \mathrm{d}x \\
&= \int_{x_l(\varepsilon)}^{x_u(\varepsilon)} f_x(x) \left(k^2(f(x)-x)^2 + E_\nu[(f(x) - g(f(x)+\nu))^2]\right) \mathrm{d}x \\
&+ \int_{x_u(\varepsilon)}^{+\infty} f_x(x) \left(k^2(f(x)-x)^2 + E_\nu[(f(x) - g(f(x)+\nu))^2]\right) \mathrm{d}x \\
&+ \int_{-\infty}^{x_l(\varepsilon)} f_x(x) \left(k^2(f(x)-x)^2 + E_\nu[(f(x) - g(f(x)+\nu))^2]\right) \mathrm{d}x \\
\Leftrightarrow \quad J(f,g) - \tilde{J}(f,g) &= \int_{x_u(\varepsilon)}^{+\infty} f_x(x) \left(k^2(f(x)-x)^2 + E_\nu[(f(x) - g(f(x)+\nu))^2]\right) \mathrm{d}x \\
&+ \int_{-\infty}^{x_l(\varepsilon)} f_x(x) \left(k^2(f(x)-x)^2 + E_\nu[(f(x) - g(f(x)+\nu))^2]\right) \mathrm{d}x \\
&=: \mathrm{ER}(x_l(\varepsilon), x_u(\varepsilon))\,.
\end{aligned}$$

As f and g are chosen as sufficiently bounded and we saw that f_x tends to zero in the limit, we may see, that we find $x_l(\varepsilon)$ and $x_u(\varepsilon)$ such as

$$\begin{aligned}
\left| \int_{x_u(\varepsilon)}^{+\infty} f_x(x) \left(k^2(f(x)-x)^2 + E_\nu[(f(x) - g(f(x)+\nu))^2]\right) \mathrm{d}x \right| &< \frac{\varepsilon}{2} \\
\left| \int_{-\infty}^{x_l(\varepsilon)} f_x(x) \left(k^2(f(x)-x)^2 + E_\nu[(f(x) - g(f(x)+\nu))^2]\right) \mathrm{d}x \right| &< \frac{\varepsilon}{2}\,.
\end{aligned}$$

Therefore,

$$|\mathrm{ER}(x_l(\varepsilon), x_u(\varepsilon))| < 2 \cdot \frac{\varepsilon}{2} = \varepsilon \,.$$

■

Knowing this Lemma, we may derive the form needed, by using the properties of the conditional expectation.

Theorem 7.1.1

For bounded f, the cost functional J may be approximated up to a precision $\varepsilon > 0$ by

$$\tilde{J}(f) = \int_{x_l(\varepsilon)}^{x_u(\varepsilon)} f_x(x) \left(k^2(f(x) - x)^2 + E_\nu[(f(x) - g_f^*(f(x) + \nu))^2]\right) \mathrm{d}x \,,$$

with f_x density of x and $x_l(\varepsilon)$, $x_u(\varepsilon)$ the chosen integral boundaries as the needed precision is achieved.

Proof: For f and g chosen as they are bounded, we already know from Lemma 7.1.1, that we may get an approximation with an error smaller than a given $\varepsilon > 0$. Now, we just have a bounded f and determine the g_f^* given on the f. Therefore, we have to show that our g_f^* is bounded on the interval $[x_l(\varepsilon), x_u(\varepsilon)]$. We know that g_f^* is given by

$$g_f^*(y) = E_{x,\nu}[f(x) \mid f(x) + \nu = y] \,.$$

As our f is bounded, we know that

$$\|f\|_\infty < \infty \,.$$

Using that for $X \in L^p$ for $p \in [1, \infty]$ also $E[X|G] \in L^p$, for G sub σ-algebra of the before used σ-algebra [32], we get

$$E_{x,\nu}[f(x) \mid f(x) + \nu] \in L^\infty \,,$$

which means g_f^* is bounded and therefore, we may apply Lemma 7.1.1. ■

7.1.2. Deriving a numerical Criterion for local Minimizers

Based on the results from the last section, we want to derive a criterion for a local minimizer in the case of Witsenhausen's counterexample.

As we saw that the given cost functional may be handled using variational analysis, we use the Euler-Lagrange equation as a necessary criterion for a local minimizer. Therefore, we first extract the Lagrangian function.

Remark 7.1.2

In Witsenhausen's counterexample, the Lagrangian function is given by

$$L(x, f, f') = f_x(x) \left(k^2(f(x) - x)^2 + E_\nu[(f(x) - g(f(x) + \nu))^2]\right) \,.$$

As in the end the expected value in our Lagrangian is given by an improper integral, we again have to find a way to handle it.

Remark 7.1.3

We may write the expected value as an integral with

$$E_\nu[(f(x) - g(f(x) + \nu))^2] = \int_{-\infty}^{+\infty} f_\nu(\nu)(f(x) - g(f(x) + \nu))^2 \, \mathrm{d}\nu \,,$$

whereby f_ν is the density of ν. Knowing $\lim_{|\nu|\to\infty} f_\nu(\nu) = 0$ and f as well as g_f^* is bounded, we see that we may find boundaries $\nu_l(\varepsilon)$ and $\nu_u(\varepsilon)$ such as

$$\left| E_\nu[(f(x) - g(f(x) + \nu))^2] - \int_{\nu_l(\varepsilon)}^{\nu_o(\varepsilon)} f_\nu(\nu)(f(x) - g(f(x) + \nu))^2 \, \mathrm{d}\nu \right| < \varepsilon \,.$$

As we now have boundaries, we can approximate the integral expression using Gauss-Legendre quadrature. Let w_i be the Gauss-Legendre weights and ν_i the supporting points, each scaled to the integration area $[\nu_l(\varepsilon), \nu_o(\varepsilon)]$, then we get as an approximation for n steps

$$\tilde{L}(x, f, f') = f_x(x) \left(k^2(f(x) - x)^2 + \sum_{i=1}^{n} w_i f_\nu(\nu_i)(f(x) - g(f(x) + \nu_i))^2 \right)$$

that we may use in further steps. We choose the n as the approximation fulfills the precision $\tilde{\varepsilon} \geq \varepsilon$, such as

$$\left| L(x, f, f') - \tilde{L}(x, f, f') \right| \leq \tilde{\varepsilon} \,.$$

Having the results from the prior remarks, we are able to define a approximative criterion for local minimizers.

Theorem 7.1.2 (*Approximated Criterion for local Minimizers*)

If a function f that is bounded is a local minimizer, it has to fulfill the condition

$$k^2(f(x)-x)+\left(\sum_{i=1}^{n} f_\nu(\nu_i) w_i (g'(f(x) + \nu_i) - 1)(g(f(x) + \nu_i) - f(x)) \right) = 0, \quad \forall x \in [x_l(\varepsilon), x_u(\varepsilon)]$$

up to a precision $\hat{\varepsilon}$.

Proof: In Remark 7.1.3, we have seen, we may approximate the Euler-Lagrange equation for Witsenhausen's counterexample properly by

$$f_x(x) \left(k^2(x - f(x))^2 + \sum_{i=1}^{n} w_i f_\nu(\nu_i)(f(x) - g(f(x) + \nu_i))^2 \right) .$$

As this term does not depend on f' and we have a sum instead of an integral, deriving the approximated Euler-Lagrange equation from this term becomes much easier. Partially differentiating

for $f(x)$ leads to

$$2f_x(x)\left(-k^2(x-f(x))+\left(\sum_{i=1}^{n} f_\nu(\nu_i)w_i(g'(f(x)+\nu_i)-1)(g(f(x)+\nu_i)-f(x))\right)\right).$$

Therefore, we get for the approximated Euler-Lagrange equation with

$$\begin{aligned} 0 &= \frac{\mathrm{d}}{\mathrm{d}x}\tilde{L}_{f'}(\cdot,f,f') - \tilde{L}_f(\cdot,f,f') \\ \Leftrightarrow 0 &= -2f_x(x)\left(-k^2(x-f(x))+\left(\sum_{i=1}^{n} f_\nu(\nu_i)w_i(g'(f(x)+\nu_i)-1)(g(f(x)+\nu_i)-f(x))\right)\right) \\ \Leftrightarrow 0 &= k^2(f(x)-x)+\sum_{i=1}^{n}(f_\nu(\nu_i)w_i(g'(f(x)+\nu_i)-1)(g(f(x)+\nu_i)-f(x))) \,. \end{aligned} \tag{7.1}$$

As the error for the derivative may become bigger than ε, we have to allow an deviation from the Euler-Lagrange equation with $\hat{\varepsilon} > \varepsilon$. ■

In the following sections, for a given function f, the approximated Euler-Lagrange value from Equation 7.1 at position x will be called $\mathrm{EL}(f,x)$. Further, for fixed f we call

$$\max|EL| := \max_{x\in\mathbb{R}}|EL(f,x)|\,.$$

7.2. Euler-Lagrange Values of known Attempts to the Counterexample

Before we might look on new attempts to minimize the cost functional, we want to have a look on previous results and if they fulfill the criterion gained from variational analysis. We will look on three well known attempts and focus on the well known benchmark, wherein

$$\sigma = 5, \quad k = 0.2\,.$$

7.2.1. Witsenhausen's Attempt

The function pointed out in Witsenhausen's counterexample is given as

$$f_W(x) = \sigma \cdot \mathrm{sgn}(x)$$

and reached a cost value of 0.4042. Looking on the Euler-Lagrange value for this function, we get a maximum absolute value of 0.9999 in the interval $[-30, 30]$. The resulting plot may be seen in Figure 7.1. In the plot it is easy to see that an increasing absolute x value leads to an increasing Euler-Lagrange value as well.

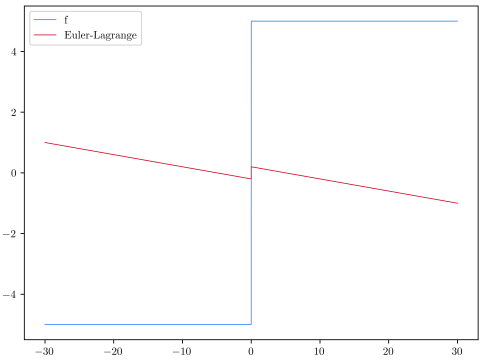


Figure 7.1.: EL for Witsenhausen's f

7.2.2. Deng's and Ho's Attempt

In 1995 Deng and Ho published a paper wherein they obtained a 2-step function given by

$$f_{DH}(x) = \begin{cases} -9.048 & x < -6.41 \\ -3.168 & -6.41 \leq x < 0 \\ 3.168 & 0 \leq x < 6.41 \\ 9.048 & x \geq 6.41 \end{cases}$$

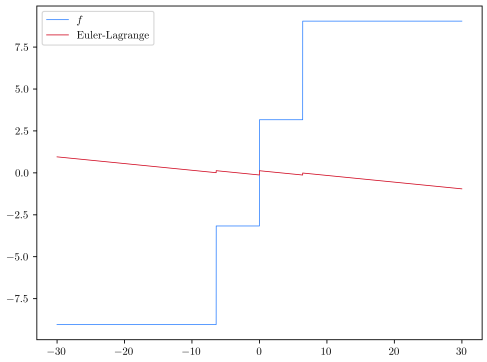


Figure 7.2.: EL for Deng's and Ho's f

which reached a cost value of 0.1901[1]. The maximum absolute value for the Euler-Lagrange value is given by 0.9539 which means a light improvement to the function given by Witsenhausen. It may be seen in Figure 7.2 that for increasing absolute x the Euler-Lagrange value increases slower, which is an explanation for the improved Euler-Lagrange value in the given interval.

7.2.3. 3.5-step function from Lau's, Lee's and Ho's attempt

In 2001 Lau, Lee and Ho published a paper, where they added segments and slopes to given step functions. In this paper they obtained the best result initially using a 3.5-step function, which is given by

$$f_{LLH}(x) = \begin{cases} 0 & 0 \leq |x| < 3.25 \\ \operatorname{sgn}(x) \cdot 6.5 & 3.25 \leq |x| < 9.9 \\ \operatorname{sgn}(x) \cdot 13.2 & 9.9 \leq |x| < 16.65 \\ \operatorname{sgn}(x) \cdot 19.9 & 16.65 \leq |x| \end{cases}$$

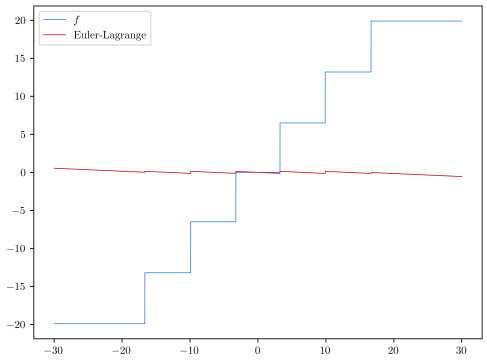


Figure 7.3.: EL for 3.5-step function

and results in a cost value of 0.1713. Again also the maximum absolute Euler-Lagrange value decreases to now 0.9486 in the interval $[-30, 30]$, which may be seen in Figure 7.3.

In the end, we may say that approaches obtaining a lower value for the cost functional also obtain lower absolute values for the Euler-Lagrange equation. But we can also see that none of the functions looked at, resulted in a Euler-Lagrange equation near zero on the interval considered.

[1]Determined by Monte Carlo simulation with standard deviation 0.01

8. Applying Variational Methods to Witsenhausen's Counterexample

We have seen that the problem stated by Witsenhausen may be seen as a problem from variational analysis. Moreover, we know the Euler-Lagrange value is a criterion that can be used to determine, whether a function can might be a local minima or not. We want to use this criterion and obtain a method using it.

In this chapter, we will introduce the method and focus on the benchmarking case wherefore $\sigma = 5$ and $k = 0.2$.

8.1. Concept for a Methodology based on Variational Analysis

As we saw in the previous section, it seems like the Euler-Lagrange value of a function has an impact on the value of the cost functional. Therefore, it seems reasonable to build a function out of basis functions for which the Euler-Lagrange value is zero.

This means two steps for us to perform:

1. Define an initial function and make it fit the Euler-Lagrange equation (from now on we will call this process *rooting*).
2. Stack multiple of those *rooted* basis function beyond each other to determine our final function.

Before we consider the method to apply, we remind one result Witsenhausen pointed out in his publication.

Remark 8.1.1

We know from previous chapters that for the well known problem, pointed out, exists an optimal solution f^*, for which it is valid, that

$$E_x[f^*(x)] = 0\,.$$

This remark and knowing, previous papers most of the time yielded even functions, lead us to the assumption that the function we are looking for also is even. For this reason, we just consider even functions and just optimize the functions for $x \geq 0$, which as a side effect speeds up the calculation immensely.

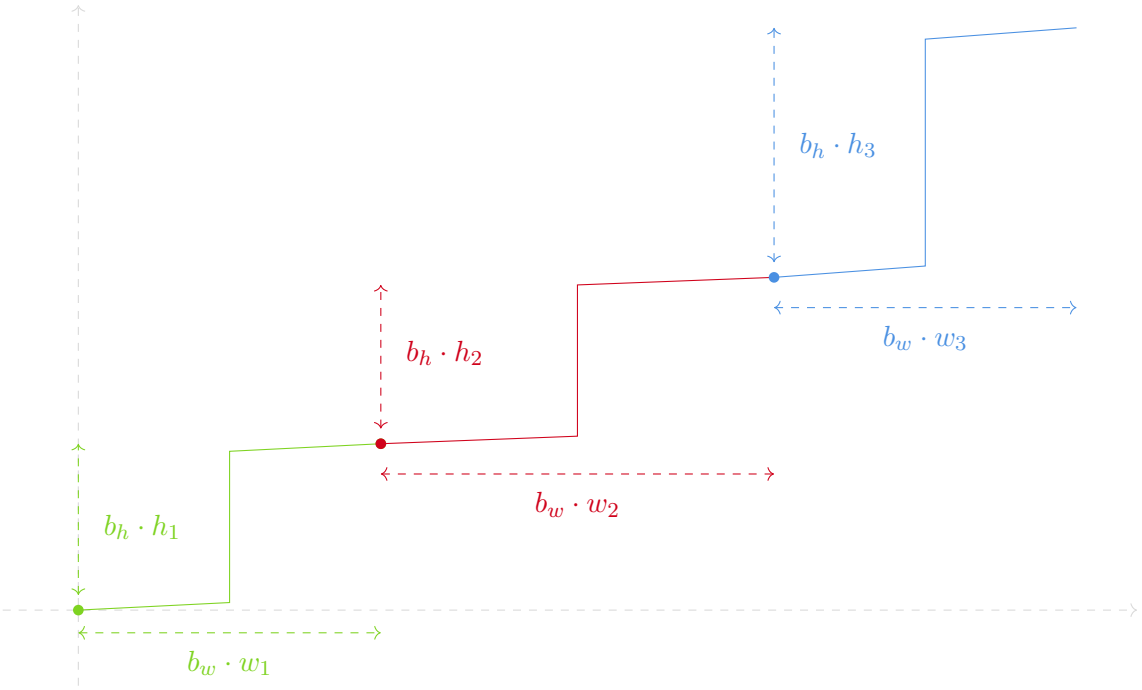


Figure 8.1.: Construction of optimized step function

Having the main idea and just optimizing for $x \geq 0$, our function, gained by stacking the basis functions might look like in Figure 8.1. There, we see the basis function (even if the functions in the Figure do not fit the Euler-Lagrange equation) in green, red and blue. Those *step like* basis function are scaled in x- and y-direction to minimize the cost functional and then stacked beyond each other, as the end point of the i-th basis function is the first of the $(i+1)$-th basis function.

We generalize our idea. We assume our basis function has width b_w and height b_h. To perform an optimization on the cost value from Witsenhausen's publication, we add parameters to each basis function. This means, the i-th stacked basis function is scaled by w_i in the x- and h_i in the y-direction. Moreover, we have to add an displacement to each basis function as the first value of the added basis function has to be the last of the previous (or zero in the case the stacked function is the first). Therefore, the x-displacement d_x^i and the y-displacement d_y^i are chosen with

$$d_x^i = \sum_{j=1}^{i-1} w_i \cdot b_w, \quad d_x^1 = 0$$
$$d_y^i = \sum_{j=1}^{i-1} h_i \cdot b_h, \quad d_y^1 = 0\,.$$

In the sections following, we will now focus on how to determine such a basis function

and then stack it using different algorithms.

8.2. Gaining a Basis Function by Rooting a 2-step Function

In this section, we obtain a basis function by *rooting* a 2-step function and extracting one step.

Therefore, we first define a 2-step function, as to see in Figure 8.2. We initially choose step width 5 and step height 8, as the form of the steps is relevant, not the width or height. As this shape does not fit the Euler-Lagrange equation, we have to *root* it. Therefore, we perform two steps.

1. Fix two points (marked in purple) in each jump discontinuity and *root* the function.
2. *Root* function without fixed points.

This means in step one the chosen method tries to reach

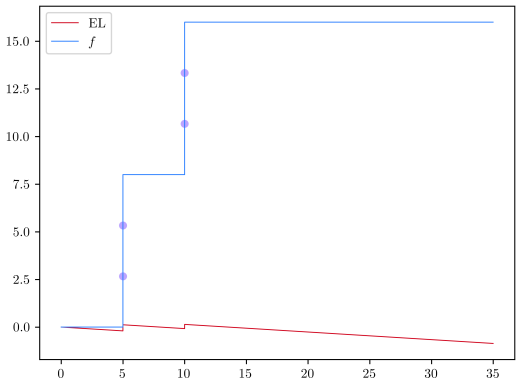


Figure 8.2.: Initial 2-step function

$$EL(f_i, x_i) = 0, \quad x_i \in \{x_1, \dots, x_n\} \backslash \{x_{k_1}, x_{k_2}, \dots, x_{k_2 j}\}, \quad 0 \leq x_1 < x_2 < \dots < x_n \leq x_u(\varepsilon)\,,$$

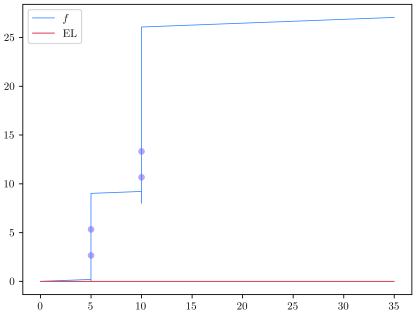
whereby the f_i are the f-values, we want to root, the x_i the grid positions, we want the f_i to optimize on and the x_{k_j} the j fixed positions we chose before. The grid is chosen, as a desired step width is approximately fit and still all the fixed positions are hit. In the second step, we then optimize on all grid positions, as we want our method to find f_i as

$$EL(f_i, x_i) = 0, \quad x_i \in \{x_1, \dots, x_n\}, \quad 0 \leq x_1 < x_2 < \dots < x_n \leq x_u(\varepsilon)\,.$$

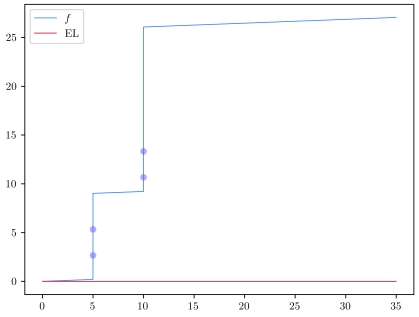
In the first iteration, the points have to be fixed, as otherwise the solution did not converge against a step function. The result from step one may be seen in Figure 8.3, where also the reached Euler-Lagrange values may be seen. As to see in the plot, this still leads to wiggles in the approximated function. Performing step two, solves the problem, keeps the 2-step form and leads to Euler-Lagrange values near zero, what might be seen in Figure 8.3 (b).

The extracted *step* may be seen in Figure 8.4. We will use the extracted *step* further. Moreover, in the plot might be seen that differently to the previous initial step function the rooted function now is strictly monotonously increasing, which is condition for a function minimizing the cost functional, like it was shown in [31] and mentioned before. In the next sections, we will use this basis function to obtain a function minimizing the cost functional pointed out.

Figure 8.3.: Optimization steps performed



(a) Optimized with fixed points



(b) Optimized without fixed points

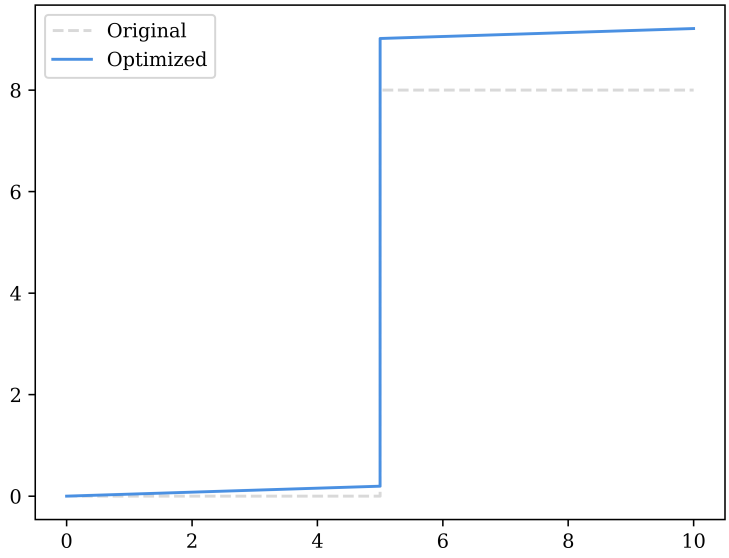


Figure 8.4.: Extracted step from 2-step function

8.3. Combine Basis Functions using `scipy` Built-Ins

In the first attempt, we perform the optimization of the parameters

$$w_i,\, h_i, \quad i = 1, \ldots, n$$

for n stacked basis functions by just using `scipy` built-ins.

Therefore, we perform two steps:

1. Optimization using *differential evolution*
2. Optimization using *BFGS*

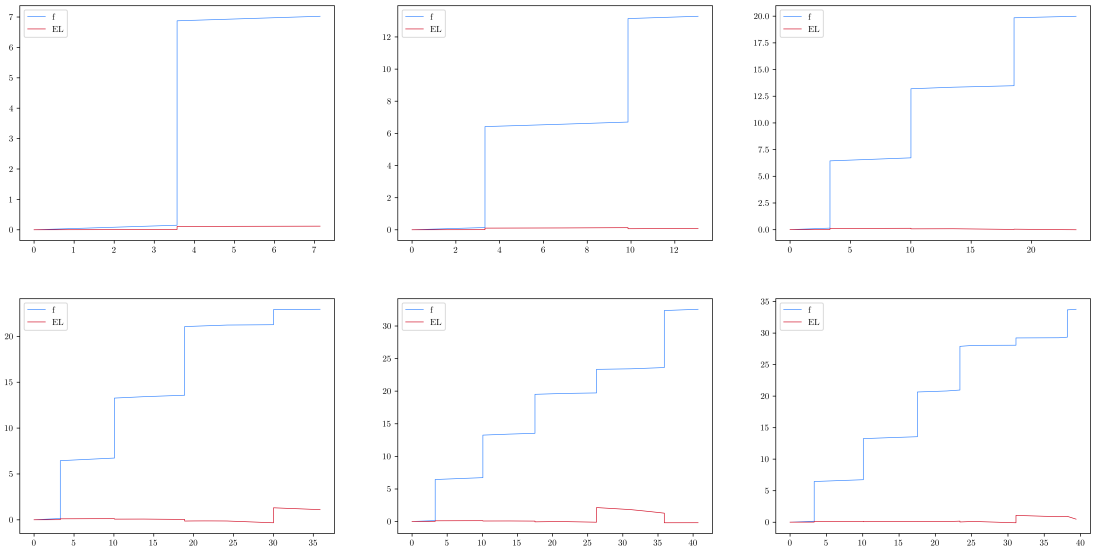


Figure 8.5.: Functions and Euler-Lagrange values obtained by `scipy` built-ins

In the first step, it is assumed that

$$w_i, h_i \in [0, 2], \quad i = 1, \ldots, n\,,$$

which leads to the boundaries for the *differential evolution* search for the optimum. After deriving the parameters from step one, they are used for further optimization in step two by using *BFGS*. As mentioned, the implementations of those methods are taken from the well known Python library `scipy`.

Steps	$\max \lvert EL \rvert$	J
1	0.0919	0.20534669
2	0.0936	0.16771291
3	0.4610	0.16733274
4	1.3919	0.16734572
5	0.8361	0.16729103
6	2.4673	0.16720541

Table 8.1.: Values reached by `scipy` built-ins

The obtained functions and their corresponding Euler-Lagranges values may be seen in Figure 8.5. Moreover we report the achieved cost values and the maximum absolute Euler-Lagrange value over the interval the stacked basis functions were optimized. Those values are shown in Table 8.4.

Even when the achieved cost value 0.16714 outperforms any result published before 2001, for more than 3 stacked basis functions the maximum absolute value of the Euler-Lagrange equation increases drastically. Moreover, we observe in this case the appearance of one particularly flat step that does not fit to the pattern of the others. This seems to be caused by the chosen optimization method, which makes it necessary to choose another.

8.4. Combine Basis Functions using a Grid Search Method

As we saw in the previous section, another method is necessary to optimize the stacking of the basis functions. This method will be presented in this section, along with the results.

Grid search itself may be called an inefficient method of finding a global minimum. Even with 4 stacked basis functions, each with two parameters, and considering 5 possible values for each parameter, there are $390,625$ parameter combinations to consider. Performing the evaluations with a precision of 10^{-6} one operation takes about 0.74846 seconds. For $390,625$ combinations this leads to a compute time of about 81.21 hours just for the first iteration in the grid search.

The parameters obtained using `scipy` built-in methods are listed in Table 8.2. Ignoring the flat steps, the values appear to lie within the interval $[0.95, 1.7]$. Furthermore, step

No.	w_1^{opt}	w_2^{opt}	w_3^{opt}	w_4^{opt}	h_1^{opt}	h_2^{opt}	h_3^{opt}	h_4^{opt}
1	1.0709				1.0641			
2	0.9996	0.9607			0.9938	1.0188		
3	0.9923	1.0202	1.5423		0.9966	1.0240	1.0069	
4	0.9871	1.0496	1.5898	1.7522	0.9977	1.0347	1.1845	0.2606

Table 8.2.: Scaling factors obtained with `scipy` built-ins

height and width tend to increase or at least do not decrease significantly with increasing x. This observations lead us to the assumption that

$$\begin{aligned} w_{i+1}^{opt} - w_i^{opt} &\geq -0.15\,, \\ h_{i+1}^{opt} - h_i^{opt} &\geq -0.15\,, \quad i = 1, \ldots, n\,. \end{aligned}$$

Moreover, we assume the best parameters to fulfill

$$w_i^{opt}, h_i^{opt} \in [0.7, 1.7], \quad i = 1, \ldots, n\,.$$

Using $[0.5, 0.75]$ as the area to search each parameter in and the first assumption as a filter criterion, we may reduce the number of combinations that has to be tested immensely. As this in the second iteration of the grid search leads to a huge number of possible combinations (for 5 stacked basis functions there are $702,240$ combinations to consider) and further restrictions do not seem to deteriorate the result, the selection of considered combinations is restricted to the assumptions

$$\begin{aligned} w_{i+1}^{opt} - w_i^{opt} &\geq 0\,, \\ h_{i+1}^{opt} - h_i^{opt} &\geq 0\,, i = 1, \ldots, n \end{aligned}$$

and

$$w_i^{opt}, h_i^{opt} \in [0.5, 1.5], \quad i = 1, \ldots, n\,.$$

Using this more restricted assumptions as filter, in the first grid search iteration, we get for 1 to 5 stacked basis functions the number of considered combinations listed in Table 8.3. Beside the grid search, the before introduced `scipy` built-in methods are used in additional optimization steps. This leads to a 3-step optimization method performing:

1. Optimization using *grid search*.
2. Optimization using *differential evolution*.
3. Optimization using *BFGS*.

No.	Before	After
1	25	25
2	625	225
3	15625	1225
4	390625	4900
5	9765625	15876

Table 8.3.: No. combinations before/after filtering

The resulting grid search algorithm is described in the Nassi-Shneiderman diagram in Figure 8.6. We perform the optimization using both the less as well as the more restricted filters. Due to the

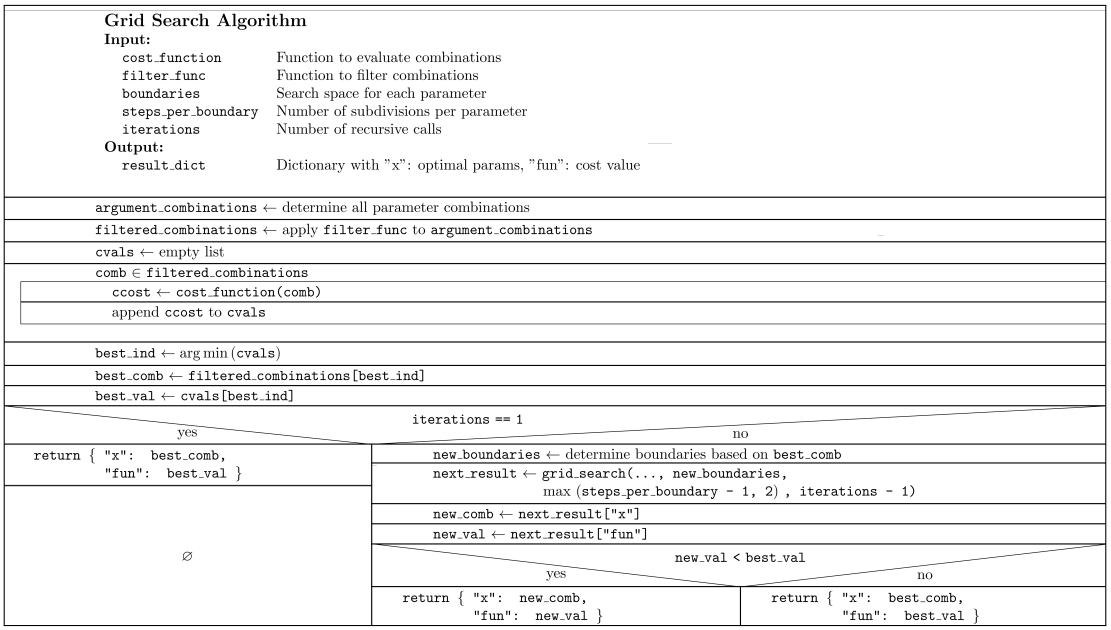


Figure 8.6.: Nassi-Shneiderman diagram for the grid search algorithm

needed compute time for the less restricted parameters, the optimization is just performed for $n = 1, \ldots, 4$ stacked basis functions. For the more restricted parameter set, the optimization is performed for $n = 1, \ldots, 5$ stacked basis functions.

As to see in Table 8.4, the reached values for the Euler-Lagrange value as well as

Steps	max $\lvert EL \rvert$	J
1	0.0915	0.20535737
2	0.0904	0.16759650
3	0.0898	0.16714794
4	0.2686	0.16713079
5	–	–

(a) Grid search less restricted

Steps	max $\lvert EL \rvert$	J
1	0.0919	0.20534150
2	0.0889	0.16758893
3	0.0880	0.16713560
4	0.2227	0.16713515
5	0.5846	0.16713460

(b) Grid search more restricted

Table 8.4.: Values reached by grid search in different parameter configurations

the cost value are nearly similar for both parameter restrictions. Moreover, the best reached value 0.16713079 outperforms any value determined before and in 2001 [19]. How the obtained value compares to more recent results can be seen in Table 8.5. The corresponding plots for each parameter configuration are presented in Figure 8.8 and Figure 8.7.

We see that the values obtained already reach the top 5 of the values obtained before. We also see that the Euler-Lagrange value decreases to 0.2686 in the case of the best cost value reached. The Euler-Lagrange as well as the value reached still should be reduced what we will do in the sections following.

Author	Year	J
Tseng et al. [28]	2017	0.166897
Mehmetoglu et al. [21]	2014	0.16692291
Karlsson et al. [17]	2011	0.16692462
Li et al. [20]	2009	0.1670790
Value just presented	**2025**	**0.16713079**
Lee et al. [19]	2001	0.167313205
Baglietto et al. [4]	2001	0.1701

Table 8.5.: Comparison to prior results

8.5. Refining Step Profiles through Smoothing Functions

In the previous Section, we stacked the basis functions obtained before and already reached a minimum cost value of 0.16713079 with a Euler-Lagrange value of 0.2686. This values will be reduced in this section.

In their publication Tseng et al. found parameter configurations wherefore the optimal step shape seems to be not an affine function but slightly curved [28]. Especially for bigger values of k this seems to happen. As the general shape for different k seems to have similarities, this could be the case for smaller k, too. This lead to the idea that both named values might be minimized by adding a *smoothing function* upon the step functions

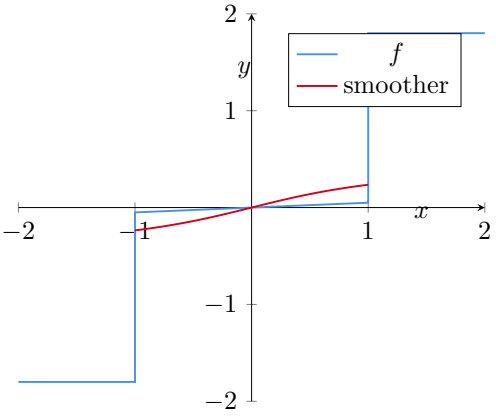


Figure 8.9.: Idea adding a *smoother*

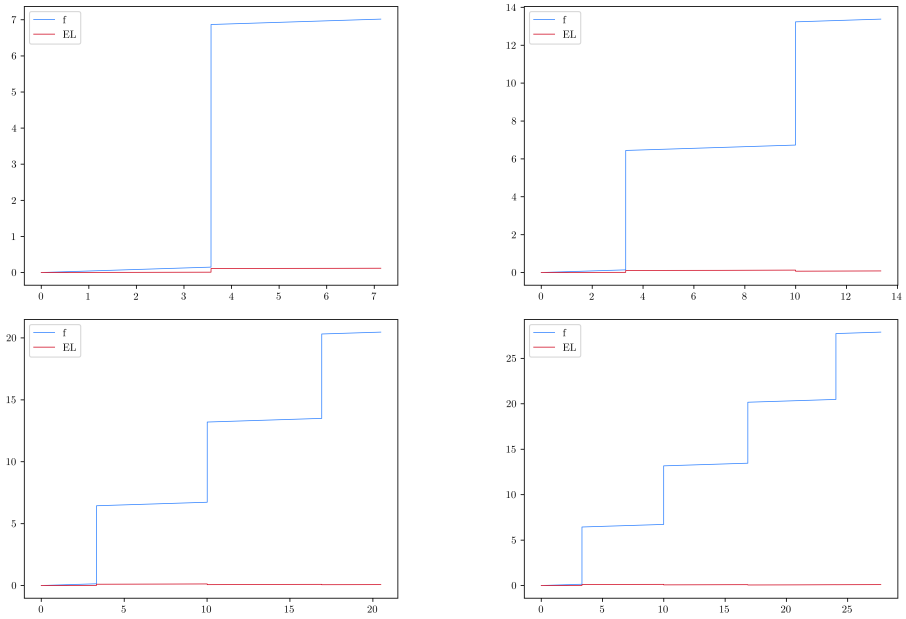


Figure 8.7.: Functions and Euler-Lagrange values obtained by grid search (less restricted parameter set)

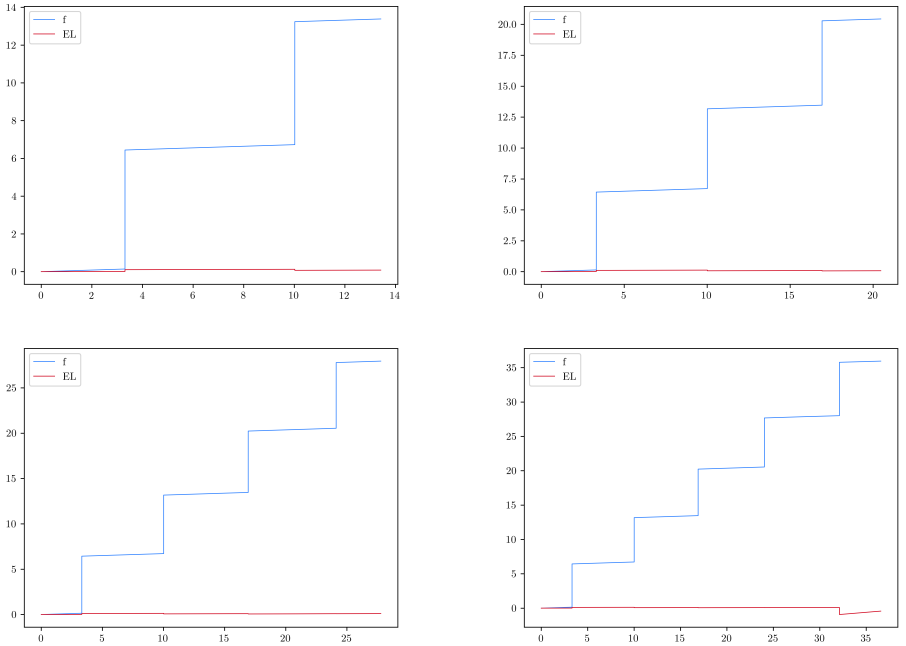


Figure 8.8.: Functions and Euler-Lagrange values obtained by grid search (more restricted parameter set)

determined before. As *smoothing functions*, we choose 11 different functions. Those functions are listed in Equation 8.1 named as $s_1, \ldots, s_{10}$ and shown in Figure 8.10, whereby Φ names the distribution function of the normal distribution.

Again, we assume that the optimal function is even and for this reason just optimize for $x \geq 0$. To simplify the optimization we introduce two parameters for each step covered. Therefore, for our *smoothers* s_i we just consider $x \in [-1, 1]$ and $x = 0$ as the center of a step plateau as it it shown in Figure 8.9. Then, we choose a parameter $\alpha_j \in [-0.1, 0.1]$ for the height of the added *smoother* and β_j to cut just a part of the *smoother* to add it on to the plateau. In formulas this may be written as

$$\alpha_j \cdot s_i(\beta_j \cdot x), \qquad \alpha_j \in [-1, 1],\ \beta_j \in [-0.1, 0.1],\ x \in [-1, 1]\,.$$

By just considering $x \geq 0$, we run into the situation that the first plateau as well as

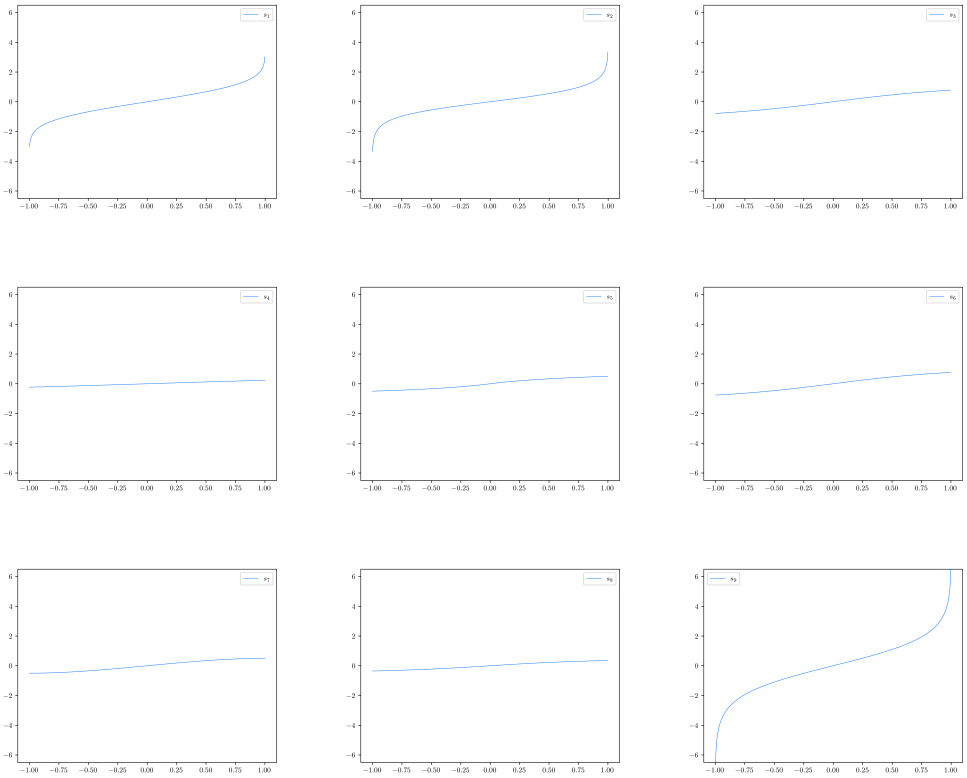


Figure 8.10.: Smoother functions $s_1, \ldots, s_9$

the last is just a *half* plateau. This might be easier to understand looking on Figure 8.9 again. The first plateau is associated with the *smoother* in the interval $[-1, 0]$ the last with the interval $[0, 1]$. Therefore, we just consider $x \in [-1, 0]$ respectively $[0, 1]$ as input for the smoothing functions for the first/last plateau.

Steps	$\max \lvert EL \rvert$	J
2	0.0837	0.16741087
3	0.2181	0.16694973
4	1.3885	0.16694962
5	2.5931	0.16694900

(a) Smoother: s_1

Steps	$\max \lvert EL \rvert$	J
2	0.0838	0.16741097
3	0.1701	0.16695058
4	1.2070	0.16695070
5	2.6739	0.16694965

(b) Smoother: s_2

Steps	$\max \lvert EL \rvert$	J
2	0.0779	0.16741069
3	0.1612	0.16694980
4	1.2085	0.16694891
5	2.6405	0.16694818

(c) Smoother: s_3

Steps	$\max \lvert EL \rvert$	J
2	0.0775	0.16741090
3	0.1695	0.16694922
4	1.2082	0.16694872
5	2.6210	0.16694821

(d) Smoother: s_4

Steps	$\max \lvert EL \rvert$	J
2	0.0738	0.16741425
3	0.1730	0.16694937
4	1.2091	0.16695002
5	2.6387	0.16695053

(e) Smoother: s_5

Steps	$\max \lvert EL \rvert$	J
2	0.0779	0.16741070
3	0.1681	0.16694921
4	1.2071	0.16694892
5	2.6868	0.16694824

(f) Smoother: s_6

Steps	$\max \lvert EL \rvert$	J
2	0.0781	0.16741072
3	0.1717	0.16694920
4	1.2080	0.16694895
5	2.6471	0.16694828

(g) Smoother: s_7

Steps	$\max \lvert EL \rvert$	J
2	0.0781	0.16741077
3	0.1490	0.16695000
4	1.2076	0.16694884
5	2.6261	0.16694824

(h) Smoother: s_8

Steps	$\max \lvert EL \rvert$	J
2	0.0782	0.16740780
3	0.2153	0.16695086
4	1.2068	0.16694926
5	2.7474	0.16694826

(i) Smoother: s_9

Steps	$\max \lvert EL \rvert$	J
2	0.0780	0.16741067
3	0.1680	0.16694920
4	1.2073	0.16694905
5	2.6230	0.16694851

(j) Smoother: s_{10}

Table 8.6.: Cost and Euler-Lagrange values reached by performing smoothing on previous known step function

$$
\begin{aligned}
s_1(x) &= \Phi(0.5 \cdot (x+1)) & s_2(x) &= \operatorname{arctanh}(x) \\
s_3(x) &= \arctan(x) & s_4(x) &= \frac{1}{1+\exp(-x)} - 0.5 \\
s_5(x) &= \frac{x}{1+|x|} & s_6(x) &= \tanh(x) \\
s_7(x) &= 3x^2 - 2x^3 - 0.5 & s_8(x) &= \frac{x}{\sqrt{1+x^2}} \\
s_9(x) &= \log\left(\frac{x}{1-x}\right) & s_{10}(x) &= \arctan(x) - 0.05x
\end{aligned}
\tag{8.1}
$$

Those parameters α_j, β_j, $j = 1, \ldots, n$ have to be optimized for all n plateaus in a function.

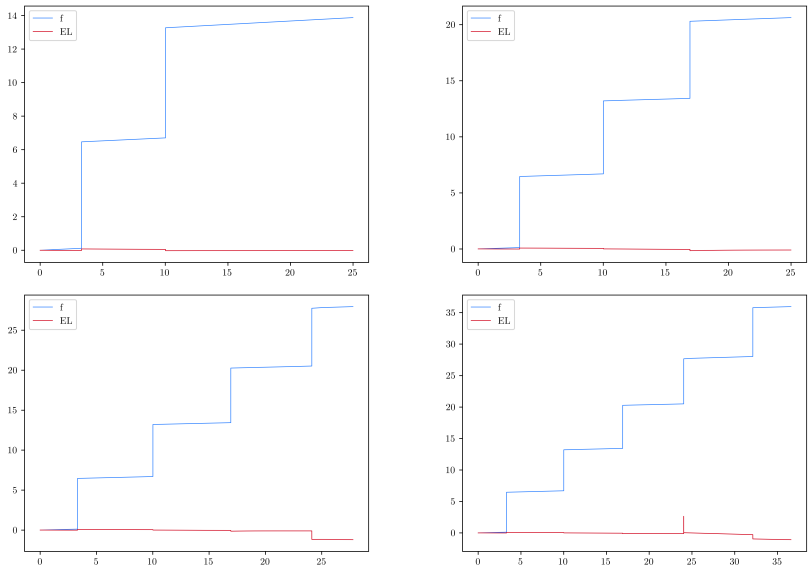


Figure 8.11.: Plots of functions obtained by using smoother s_3

This will be done by using `scipy` built-ins. This means performing the steps:

1. Perform optimization using *differential evolution*.
2. Perform optimization using *BFGS*.

The results of the different smoothing functions are listed in Table 8.6. It might be seen that smoother s_3 reaches the best cost value with 0.16694818 for 5 stacked steps. This outperforms the value gained in the previous sec-

Author	Year	J
Tseng et al. [28]	2017	0.166897
Mehmetoglu et al. [21]	2014	0.16692291
Karlsson et al. [17]	2011	0.16692462
Value just presented	**2025**	**0.16694818**
Li et al. [20]	2009	0.1670790
Lee et al. [19]	2001	0.167313205
Baglietto et al. [4]	2001	0.1701

Table 8.7.: Comparison to prior results

tion and any value obtained before 2009 [20].
How this result performs compared to previous results is shown in Table 8.7. Moreover, the functions obtained by using this smoother are shown in Figure 8.13.

Against the expectation, for the lower cost values the Euler-Lagrange values increased for most of the compared values. Therefore, we extend the search idea from this section and present it in the next.

8.6. Combining Search for Optimal Stacked Basis Functions and Smoothing of the Step Functions

In the two previous sections, we used different optimization approaches:

1. First, we determined an optimal step function, by stacking basis steps.
2. Afterwards, we added a smoothing function onto those step functions.

As the optimal step function to smooth could be a different one than the optimal *unsmooth* step function, we optimize step 1 and 2 at once. Therefore we perform:

1. Gain best stacking parameters as before.
2. Gain best smoothing parameters as before.
3. Vary gained parameters simultaneously.

For b^{opt}, the optimal parameters for stacking the basis function and s^{opt}, the optimal parameters for smoothing the step function, we perform the variation of them using grid search. Therefore, we choose the input ranges

$$b_i^{\text{opt}} \pm 0.08,\ s_i^{\text{opt}} \pm 0.08, \qquad b_i^{\text{opt}} \in b^{\text{opt}}, s_i^{\text{opt}} \in s_i^{\text{opt}}\ .$$

We perform this optimization for the smoothers s_1 and s_3. The results for both smoothers are shown in Table 8.8 (a) and 8.8 (b). The plots in Figure 8.12 and 8.13.

Steps	$\max \lvert EL \rvert$	J
1	0.0940	0.20510975
2	0.0779	0.16740211
3	0.0776	0.16692911
4	0.6999	0.16692930
5	12.3134	0.16692928

(a) Combinated search with s_1

Steps	$\max \lvert EL \rvert$	J
1	0.0884	0.20510987
2	0.0764	0.16740023
3	0.0763	0.16692968
4	0.7084	0.16692912
5	12.1954	0.16692859

(b) Combinated search with s_3

Table 8.8.: Values reached by combinated optimization

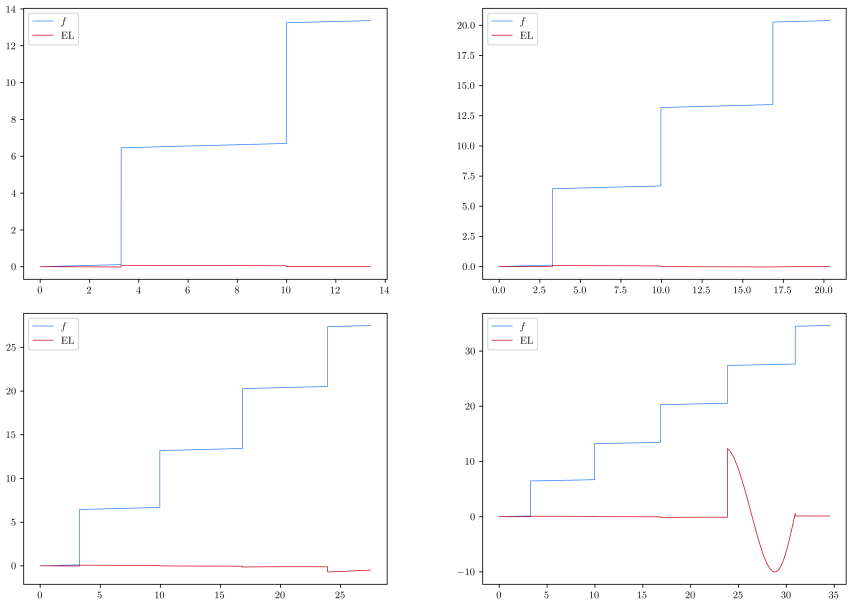


Figure 8.12.: Plots of functions obtained by optimizing stacking and smoothing process with s_1

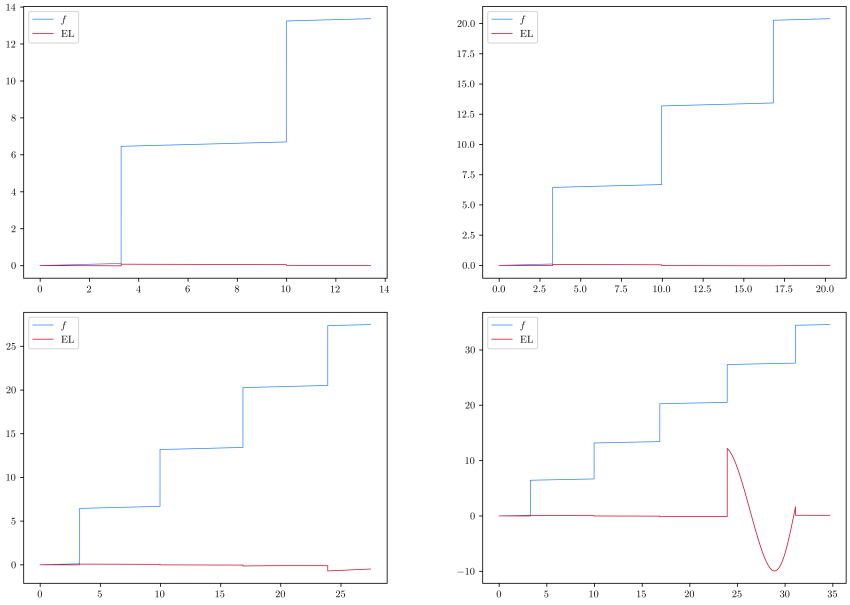


Figure 8.13.: Plots of functions obtained by optimizing stacking and smoothing process with s_3

The values gained using this optimization method, outperform the values we reached using the prior method and yields for $\sigma = 5, k = 0.2$ the value 0.16692859. This value reaches, as shown in Table 8.9, number 4 in the global ranking for the named benchmark. The difference to the currently known best is given by around $3.159 \cdot 10^{-5}$.

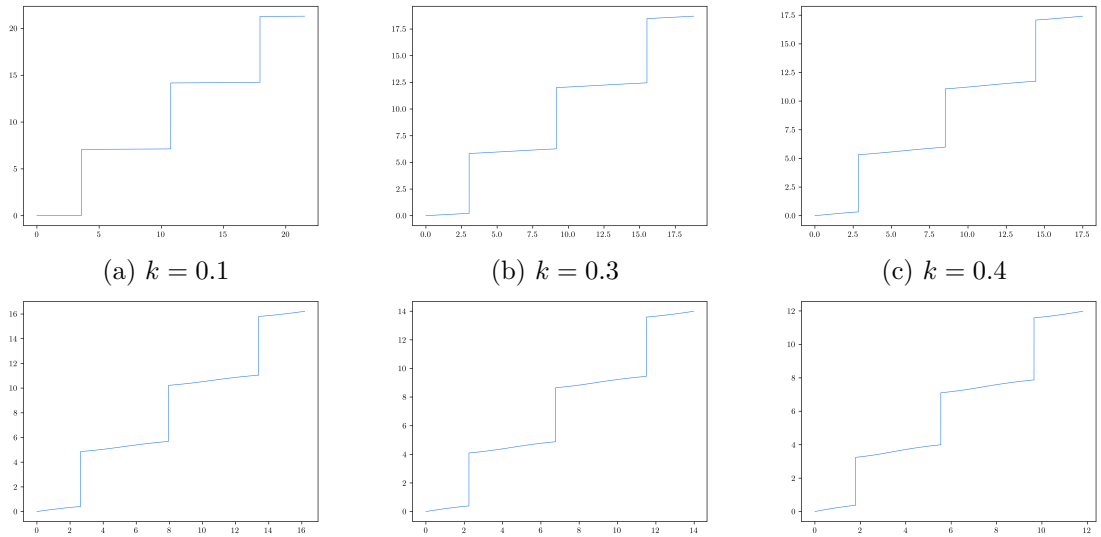
However, it is surprising that the Euler-Lagrange value becomes $\gg 0$ for such good results. Especially for more than 3 steps, it seems like the 4-th step initiates a huge increase of the value. This was not the case when we were not using the smoothing functions. Why this happens or if there are other irregularities regarding the Euler-Lagrange value could be topic of further works.

Author	Year	J
Tseng et al. [28]	2017	0.166897
Mehmetoglu et al. [21]	2014	0.16692291
Karlsson et al. [17]	2011	0.16692462
Value just presented	**2025**	**0.16692859**
Li et al. [20]	2009	0.1670790
Lee et al. [19]	2001	0.167313205
Baglietto et al. [4]	2001	0.1701

Table 8.9.: Comparison to prior results

8.7. Evaluating the Algorithm for different k

As it was done in [28] we want to consider different parameter combinations for σ, k. We keep $\sigma = 5$ fixed and then vary k in the interval $[0.1, 1.5]$. We just focus on 3-step functions.



(a) $k = 0.1$ (b) $k = 0.3$ (c) $k = 0.4$

(d) $k = 0.5$ (e) $k = 0.7$ (f) $k = 0.9$

Figure 8.14.: Reached functions for various k

The algorithm is applied as described in Section 8.6. This means, the basis functions determined for $k = 0.2$ are chosen. Therefore the expected result is that the algorithm

performs better for k near 0.2. To be able to compare the results gained, we use the values Tseng et al. reached in [28].

The expected behavior occurs and in Table 8.10 we see that for $k > 0.7$ the reached value differs at least by 0.31 to the values gained in [28]. On the other hand, the algorithm for all tested $k \in [0.1, 0.7]$ deviates from the test values by a maximum of 0.054542.

Similar to the results in [28], in our results also the slope as well as the curvature of the steps seems to increase with k increasing, as to see in Figure 8.14.

k	Tseng et al. [28]	Our result
0.1	0.052292	0.053621
0.2	0.166897	0.166928
0.3	0.314824	0.314867
0.4	0.477652	0.477801
0.5	0.640974	0.642165
0.7	0.916458	0.971008
0.9	0.961454	1.274045
1.5	0.961498	1.696058

Table 8.10.: Values reached for various k

9. Conclusion and Outlook

This master's thesis focused on two major topics:

1. Obtaining an efficient method to evaluate the cost functional.
2. Obtaining a method optimizing functions to minimize the cost functional.

First, the method for evaluating the cost functional was developed. Therefore, in Chapter 5 a representation, based on the Fisher information, Witsenhausen developed, was used. Therefore, first a adaptive method choosing the grid points was developed and adapted into the method. Then, the integration was build especially for the given cost function. The integrand was expressed using spline interpolation. As derivatives and integrations that occur then can be expressed analytically, this reduces the computational effort. To reduce compute time further, integral evaluations that have to be done thousands of times for each cost evaluation, were ported onto the GPU. As for Witsenhausen's counterexample often step functions are used as controllers, the integration method was improved for functions including discontinuities. This was realized by creating an algorithm looking for discontinuities and considering them for the integration. As no integration method for discontinuous functions was implemented for the used `PyTorch` package, a specialized Gauss-Legendre integration method, running on the GPU was created. Performing those steps, a method was gained, evaluating the cost functional in less than a second for 8 decimal places. Moreover, the method performed for up to 15 decimal places for known benchmark values.

To perform the function optimization, first in Chapter 6 the needed theory on variational analysis was introduced. Then, this theory was used in Chapter 7 to show that Witsenhausen's counterexample is a problem from variational analysis. Moreover, a necessary criterion for a local minimizer of the known cost functional was obtained. This criterion was then adapted to become a numerical criterion that might be used in the numerical optimization. Afterwards, in Chapter 8, this criterion was used to determine a step shaped basis function, fulfilling the gained criterion. Then, those gained basis functions were stacked to gain a n-step function. To perform an optimization of the *stacking parameters* a grid search was implemented, applying a filter function on the considered parameter combinations. Using this grid search method already results in the 5-th best value known. Considering results from [28] lead to the idea to use *smoother* step function. Therefore, various functions were used to smooth the step function. Performing an optimization of the stacked basis function and the added smoothers at once, lead to the 4-th best value known for the usual benchmark. At the same time, the reached value just differs by around $3.212 \cdot 10^{-5}$ to the currently known best value.

Finally, it might be said that a method was developed that is able to evaluate the cost functional fast and up to a high precision. Moreover, an optimization method was gained that reached the 4-th best value known, with a difference of around $3.159 \cdot 10^{-5}$ to the currently known best.

9.1. Outlook

The methods presented in this thesis have yielded promising results, but there is still room for improvement and exploration of new approaches.

The first question which might be interesting to discuss is, why it is possible to improve the reached cost value while the maximum deviation from the gained necessary criterion increases. This was observed in Chapter 8 and does not fit into the expected behavior. Here, one could also ask if the reached results could be improved, if the optimization would reach a function, fulfilling the necessary criterion at all.

Moreover, the idea of adding the gained necessary criterion as a penalty term to other optimization attempts could be considered. This could lead to better results, as the gained functions should fit the necessary criterion and therefore could be local minimizers. This idea also overcomes the fixed basis functions and gives more flexibility to the optimization.

Another idea, keeping the basis function idea, could be based on choosing smoothed basis functions, as they fulfill the necessary criterion. This could address the problem that adding the smoothers, sometimes leads to a massive increasing of the deviation from the necessary criterion.

Bibliography

[1] Mei (May) Deng and Yu-Chi Ho. "An ordinal optimization approach to optimal control problems". In: *Automatica* (1999), pp. 331–338.

[2] Annette A'Campo-Neuen. *Vorlesung: Differentialgleichungen*. Universität Basel, 2020.

[3] John C. Vassberg Antony Jameson. "Studies of alternative numrtical optimization methods applied to the brachistochrone problem". In: (2000).

[4] M. Baglietto, T. Parisini, and R. Zoppoli. "Numerical solutions to the Witsenhausen counterexample by approximating networks". In: *IEEE Transactions on Automatic Control* (2001), pp. 1471–1477.

[5] Rajesh Bansal and Tamer Başar. "Stochastic teams with nonclassical information revisited: When is an affine law optimal?" In: *Automatic Control, IEEE Transactions on* (1987), pp. 554–559.

[6] Bruce van Brunt. *The Calculus of Variations*. Springer Verlag, 2004.

[7] Wei Cao et al. "On Nonparametric Estimation of the Fisher Information". In: *2020 IEEE International Symposium on Information Theory (ISIT)*. 2020, pp. 2216–2221.

[8] Pedro Comesaña, Fernando Pérez-González, and Chaouki T. Abdallah. "Witsenhausen's Counterexample and Its Links with Multimedia Security Problems". In: *Digital Forensics and Watermarking*. Springer Berlin Heidelberg, 2012, pp. 479–493.

[9] *Cubic Spline Interpolation - Wikiversity*. Accessed: 2025-06-03. URL: https://en.wikiversity.org/wiki/Cubic_Spline_Interpolation.

[10] Oliver Deiser. *Analysis 2*. Accessed: 2025-06-01. Online Publication, 2024. URL: https://www.aleph1.info/?call=Puc&permalink=analysis2.

[11] Reiner Dreizler and Eberhard Engel. *Density Functional Theory: An Advanced Course*. Springer Verlag, 2011.

[12] "Ein Streit unter Brüdern führte zum wichtigsten Prinzip der Physik". In: *Spektrum: Die fabelhafte Welt der Mathematik* (2024).

[13] Encyclopedia of Mathematics. *Real analytic function*. Accessed: 2025-08-11. URL: https://encyclopediaofmath.org/wiki/Real_analytic_function.

[14] Harro Heuser. *Lehrbuch der Analysis: Teil 1*. Vieweg+Teubner Verlag, 2003.

[15] Y.-C. Ho and J.T. Lee. "Granular optimization: An approach to function optimization". In: *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*. 2000, 103–111 vol.1.

[16] Investopedia. *Nash Equilibrium*. Accessed: 2025-08-13. 2025. URL: https://www.investopedia.com/terms/n/nash-equilibrium.asp.

[17] Johannes Karlsson et al. “Iterative source-channel coding approach to Witsenhausen’s counterexample”. In: *Proceedings of the 2011 American Control Conference*. 2011, pp. 5348–5353.

[18] Hansjörg Kielhöfer. *Variationsrechnung: Eine Einführung in die Theorie einer unabhängigen Variablen mit Beispielen und Aufgaben*. Vieweg+Teubner Verlag, 2010.

[19] J.T. Lee, E. Lau, and Yu-Chi Ho. “The Witsenhausen counterexample: a hierarchical search approach for nonconvex optimization problems”. In: *IEEE Transactions on Automatic Control* (2001), pp. 382–397.

[20] Na Li, Jason R. Marden, and Jeff S. Shamma. “Learning approaches to the Witsenhausen counterexample from a view of potential games”. In: *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. 2009, pp. 157–162.

[21] Mustafa Mehmetoglu, Emrah Akyol, and Kenneth Rose. “A deterministic annealing approach to Witsenhausen’s counterexample”. In: *2014 IEEE International Symposium on Information Theory*. 2014, pp. 3032–3036.

[22] Aditya Mittal. “Numerical Solution to the Brachistochrone Problem”. In: (2008).

[23] Lothar. Papula. *Mathematische Formelsammlung [E-Book] : für Ingenieure und Naturwissenschaftler*. Vieweg+Teubner Verlag / GWV Fachverlage GmbH, Wiesbaden, 2009.

[24] Hans Josef Pesch. *Schlüsseltechnologie Mathematik: Einblicke in aktuelle Anwendungen der Mathematik*. Teubner Verlag, 2002.

[25] Hossein Pishro-Nik. *Introduction to probability, statistics, and Random Processes*. Kappa Research, LLC, 2014.

[26] *PyTorch*. https://pytorch.org/projects/pytorch/. Accessed: 2025-08-06. 2025.

[27] *scipy.interpolate.CubicSpline — SciPy v1.13.0 Manual*. Accessed: 2025-06-03. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.CubicSpline.html.

[28] Shih-Hao. Tseng and Ao Tang. “A Local Search Algorithm for the Witsenhausen’s Counterexample”. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)* (2017), pp. 5014–5019.

[29] Hans S. Witsenhausen. “A counterexample in stochastic optimum control”. In: *SIAM Journal on Control and Optimization* (1968), pp. 131–147.

[30] Martin Wohlgemuth, ed. *Mathematisch für fortgeschrittene Anfänger. Weitere beliebte Beiträge von Matroids Matheplanet*. Spektrum Verlag Heidelberg, 2010.

[31] Yihong Wu and Sergio Verdú. “Witsenhausen’s counterexample: A view from optimal transport theory”. In: (2011), pp. 5732–5737.

[32] Gordan Žitković. *Lecture Notes on: Theory of Probability I*. accessed: 2025-06-18). 2023. URL: https://web.ma.utexas.edu/users/gordanz/notes/conditional%5C_expectation.pdf.