Optimal administration strategies for EPO based on the model for erythropoiesis involving structured population equations

DIPLOMA THESIS
submitted by Laura Lippmann

at the

Universität Konstanz

Department of Mathematics and Statistics

in cooperation with the

RENAL RESEARCH INSTITUTE

Konstanz, June 2014

Supervisor and 1st Reviewer: Prof. Dr. Stefan Volkwein, University of Konstanz
2nd Reviewer: Prof. Dr. Franz Kappel, University of Graz
Abstract

This diploma thesis deals with a hyperbolic partial differential equation (PDE), developed by the Renal Research Institute in New York. It models the population density of CFU-E cells, which depends on time as well as on the maturity attribute of the cells, with respect to external administration of erythropoietin (EPO). CFU-E cells represent one level of development from stem cells to red blood cells and play a role by investigating optimal treatment of dialysis patients, amongst others. First, the PDE is normalized followed by a discretization of the maturity variable - representing the space variable in this context - which is carried out using Legendre polynomials. This leads to an ordinary differential equation (ODE). Utilizing a $L^2$-objective functional an optimal control problem is expressed. Since injections of EPO are only possible at several fixed days the control obtains a discrete nature. The theoretical background about solvability of the underlying ODE is illuminated and optimality criteria are formulated. Subsequently, different optimization techniques are examined, including the $\vartheta$-method for computing the state and the gradient and BFGS method combined with a modified Armijo step size strategy for the optimization process. Next, they are integrated in a model predictive control (MPC) framework. At the end of the thesis, numerical tests - some of which are based on actual problem settings occurring at the treatment of dialysis patients - are executed and results are presented.
Contents

| Introduction | 1 |
| 1 Governing hyperbolic differential equation | 3 |
| 1.1 Function spaces and semigroups | 5 |
| 2 Ordinary differential equations | 9 |
| 3 Optimal control problem | 13 |
| 3.1 Cauchy problem | 15 |
| 3.2 Discretization | 17 |
| 3.3 Optimality criteria | 26 |
| 4 Numerical methods | 35 |
| 4.1 Projected gradient method | 35 |
| 4.2 Projected BFGS method | 39 |
| 4.3 Termination condition | 41 |
| 4.4 Application of the $\vartheta$-method to the optimal control problem | 42 |
| 5 Model predictive control | 45 |
| 5.1 Application of MPC to the optimal control problem | 45 |
| 5.2 Suboptimal model predictive control | 50 |
| 6 Numerical results | 55 |
| 6.1 Solving the state equation | 55 |
| 6.2 Open loop problem | 58 |
| 6.3 Model predictive control | 72 |
| 7 Conclusion | 85 |

List of abbreviations 87

Bibliography 89
Introduction

The field of optimal control problems has experienced a tremendous enhancement over the last 50 years. With the mechanization in all sorts of areas - aeronautics, medicine, robotics, finance, to name but a few - computers became increasingly involved. The utilization of computing power facilitated complex optimization processes. In the course of this, various methods have been developed and improved to treat optimal control problems. In [2] several current research results on the topic of optimal control in combination with model reduction techniques are exposed.

The focus of research in optimal control is to explore the potential of influencing dynamics in a certain way. It combines optimization techniques as well as knowledge in the area of ordinary or partial differential equations (ODEs/PDEs). Mostly, the minimization of a certain objective functional is carried out which requires the solution of a so-called state equation for each iterated control. Due to possible constraints to the control, constrained optimization methods have been developed amongst others.

In [9] a model was developed to describe the evolution of blood cells depending on the concentration of erythropoietin (EPO) in blood. This thesis focuses on the partial differential equation specified there, which models the population of CFU-E cells under external administration of EPO. CFU-E cells represent one level of development between multipotent stem cells to erythrocytes. We aim at developing a method to optimally administer EPO to reproduce a given desired population. Application of this method - or actually an extended version including further stages of cells - may be a good possibility to improve the treatment of dialysis patients. For this purpose, an optimal control problem has to be formulated. The maturity attribute of the cells will be subjected to a discretization process and the resulting semi-discrete optimal control problem will be examined theoretically, based on [14]. In order to compute an optimal control numerically, the robust gradient method and the fast Broyden-Fletcher-Goldfarb-Shanno (BFGS)
method will be applied. An open loop solver will be established, mainly following [16, 23, 19]. This open loop solver will be embedded in a model predictive control (MPC) framework, based on the approach of [11, 10, 12], with a prediction horizon which is to be qualified.

In this thesis we restrict ourselves to the described simplified optimal control problem (compared to [9]) but want to emphasize that further research considering the model can be carried out. We want to mention model reduction techniques as well as sensitivity analysis. The intertwining of different aspects could disclose new perspectives and might enable targeted investigations of the full model.

Let us outline the structure of this thesis: In Chapter 1 we present the main equation which constitutes the starting point for all further considerations. Several definitions and theorems on function spaces and semigroups are compiled which are required in the following. Chapter 2 briefly presents the necessary background on ODEs to which the considered PDE is reduced by discretization of the maturity attribute. In Chapter 3 the main theoretical considerations about the specific PDE are assembled. First, we reformulate the PDE as a Cauchy problem which involves a normalization of the maturity attribute. Second, the spatial discretization is carried out, thus providing us with a semi-discrete optimal control problem. Third, the optimality criteria for this problem are formulated and finally the existence of an optimal control is proven. Several numerical approaches for the computation of an optimal control are presented in Chapter 4. These methods involve the computation of a descent direction, step size strategies and different ways how to solve the semi-discrete state equation. In Chapter 5 we explain the framework of MPC and discuss its application to the problem at hand. As an extension a suboptimal version of MPC is examined. Finally, we apply the presented optimization methods to the semi-discrete optimal control problem in Chapter 6. A few options for solving the state equation are illuminated before we turn to the optimization process. Here, the gradient method as well as the BFGS method are analyzed and compared with respect to speed, optimality and feasibility. Additionally, we have a closer look at the Armijo step size strategy. After thorough consideration of the open loop problem we address the MPC method and test its ability to react to perturbations.
The biological evolution of red blood cells runs through several levels of maturity ranging from multipotent stem cells to erythrocytes. In [9] a model was developed to describe the change of the population for different cell ages, subject to the existing amount of erythropoietin (EPO) as well as a feedback loop to model the amount of EPO being released by the kidneys depending on the population density of red blood cells. In this thesis, the coupling of the differential equations is not considered. Instead we focus on the model for CFU-E cells which represent the second level of cells. Furthermore, we assume that only external EPO is administered and no EPO is released by the kidneys. Throughout this thesis, the following linear hyperbolic partial differential equation (PDE) of first order will be the starting point for all further examinations:

\[
\begin{align*}
y_t(t, x) + y_x(t, x) &= \kappa(t, u; \mu)y(t, x) \quad \text{in } Q \text{ almost everywhere (a.e.),} \\
y(t, x) &= g(t) \quad \text{in } (t_0, t_f) \text{ a.e.,} \\
y(t_0, x) &= y_0(x) \quad \text{in } \Omega \text{ a.e.}
\end{align*}
\]

(1.1)

In equation (1.1) we have denoted the time interval with \( (t_0, t_f) \subset \mathbb{R} \) and the space domain with \( \Omega = (\underline{x}, \overline{x}) \subset \mathbb{R} \), which biologically defines the range of the maturity attribute. Let \( Q = (t_0, t_f) \times \Omega \) be the time-space cylinder. The value \( y(t, x) \) describes the population density of cells at time \( t \in (t_0, t_f) \) with maturity attribute \( x \in \Omega \).

The variable \( u \) is the control value which determines the concentration of EPO in blood. Assuming that administration of EPO is only possible at several fixed days \( \{t_1^*, \ldots, t_m^*\} \), the control has to be an element of \( \mathbb{R}^m \), so that each \( u_j, j = 1, \ldots, m \), specifies the amount of EPO injected at day \( t_j^* \) as a percentage of the maximal
possible amount $\text{EPO}_{\text{max}}$. The set of feasible controls is therefore chosen as the following subset of $U = \mathbb{R}^m$,

$$U_{\text{ad}} = \{ u \in \mathbb{R}^m \mid u_a \leq u \leq u_b \}, \quad (1.2)$$

where " \leq " is to be understood componentwise. A reasonable choice is $u_a = 0$ and $u_b = 1$.

Bounded shape functions $\chi_j$ are used to describe the non-differentiable time-dependent function $E(t)$ of the EPO concentration. The decay of EPO is exponential with a half-life $T_{1/2}$. This determines the coefficient $\lambda = \ln 2/T_{1/2}$ in the exponential functions

$$\chi_j(t) = \text{EPO}_{\text{max}} e^{-\lambda(t-t^*_j)}1_{[t^*_j, \infty)}$$

which are mainly dictated by the indicator functions $1_{[t^*_j, \infty)}$. The functions $\chi_j$ are summed up to model the change in the EPO concentration

$$E(t) = \frac{1}{TBV} \sum_{j=1}^{m} u_j \chi_j(t) \quad (1.3)$$

including the value of the total blood volume $TBV \in \mathbb{R}$.

The function $\kappa$ which appears in the state equation (1.1) has the shape

$$\kappa(t, u; \mu) = \beta - \alpha(t, u; \mu)$$

and consists of two terms, a constant proliferation rate with value $\beta$ and a time-, parameter- and control-dependent rate of apoptosis $\alpha$, which is the programmed cell death. The parameters $\mu = (\mu_1, \mu_2, \mu_3)$ are located in a parameter set

$$D_\mu = \{ \mu = (\mu_1, \mu_2, \mu_3) \in \mathbb{R}^3 \mid \mu_a \leq \mu \leq \mu_b \in \mathbb{R}^3 \}.$$ 

The rate of apoptosis $\alpha(t, u; \mu)$ depends on the time $t$ as well as on the control $u$. We use the parameter-dependent function

$$\alpha(\cdot, \cdot; \mu) : [t_0, t_f] \times \mathbb{R}^m \to \mathbb{R},$$

$$\alpha(t, u; \mu) = \frac{\mu_1}{1 + \exp(\mu_2 E(t) - \mu_3)}$$

$$= \frac{\mu_1}{1 + \exp(\mu_2 / TBV \sum_{i=1}^{m} u_i \chi_i(t) - \mu_3)} \quad (1.4)$$
to describe the mortality rate. This function describes a mortality rate which decreases monotonically with increasing EPO concentration.

Finally, equation (1.1) includes a boundary condition given by the function $g$. In the numerical tests in Chapter 6 we will choose $g \equiv S_0 e^{1.8}$ as constant, but now we treat it as an arbitrary function. $S_0$ denotes the number of cells committing to the erythroid lineage. The initial population density is given by $y_0 \in C(\bar{\Omega})$.

Using the method of characteristics we find that a unique global solution of equation (1.1) exists. For further details we refer to [9].

1.1 Function spaces and semigroups

Function spaces

Throughout this thesis we will reserve the letters $V$ and $H$ for the denotation of Hilbert spaces. Let now $\Omega \subset \mathbb{R}^n$ be a domain with sufficiently smooth boundary. For additional information on boundary properties we refer to [23, 14]. First we give a definition of a well-posed problem:

**Definition 1.1 (Well-posed problem).** A problem is called well-posed if

1. a solution exists,
2. the solution is unique and
3. it depends continuously on the data.

**Definition 1.2 (Lp spaces).** Let $1 \leq p < \infty$ and $\Omega \subset \mathbb{R}^n$. The $\| \cdot \|_p$-norm is defined for a Lebesgue-measurable function $y : \Omega \to \mathbb{R}$ as

$$
\|y\|_p := \left( \int_{\Omega} |y|^p \, dx \right)^{1/p}.
$$

We define the $L^p$ space of all equivalence classes with respect to the norm $\| \cdot \|_p$ as

$$
L^p(\Omega) := \{ y : \Omega \to \mathbb{R} \mid y \text{ measurable and } \|y\|_p < \infty \}.
$$

In this way $(L^p(\Omega), \| \cdot \|_p)$ is a Banach space. Let now $(X, \| \cdot \|_X)$ denote a Banach space and $[t_0, t_f] \subset \mathbb{R}$ a time interval. The space

$$
L^p(t_0, t_f; X) := \{ y : [t_0, t_f] \to X \mid y \text{ strongly measurable and } \|y\|_{L^p(t_0, t_f; X)} < \infty \}
$$
where the norm is given by

\[
\|y\|_{L^p(t_s,t_f;X)} := \left( \int_{t_s}^{t_f} \|y(t)\|^p_X \, dt \right)^{1/p}
\]

is a generalized \(L^p\) space.

**Definition 1.3 (Inner product on \(L^2\)).** We define the inner product on \(L^2(\Omega)\) for \(y, w \in L^2(\Omega)\) by

\[
\langle y, w \rangle_{L^2(\Omega)} := \int_{\Omega} yw \, dx.
\]

Endowed with this inner product, \((L^2(\Omega), \langle \cdot, \cdot \rangle_{L^2(\Omega)})\) is a Hilbert space.

**Remark.** Amongst all \(L^p\) spaces \(L^2\) is the only Hilbert space.

**Definition 1.4 (Fréchet differentiability).** Let \((U, \|\cdot\|_U), (V, \|\cdot\|_V)\) be Banach spaces and \(U \subset U\) an open subset. A function \(F : U \subset U \rightarrow V\) is called Fréchet differentiable at \(u \in U\) if an operator \(A \in L(U, V)\) exists with

\[
\frac{\|F(u + h) - F(u) - Ah\|_V}{\|h\|_U} \rightarrow 0 \quad \text{for} \quad \|h\|_U \rightarrow 0,
\]

for all \(h \in U\) with \(u + h \in U\). Then \(A\) is called the Fréchet derivative of \(F\) at \(u\) and we can write \(A = F'(u)\).

**Definition 1.5 (Test function).** Let \(C_0^\infty(\Omega)\) denote the space of all infinitely differentiable functions \(f : \Omega \rightarrow \mathbb{R}\) with compact support in \(\Omega\). The elements of \(C_0^\infty(\Omega)\) are called test functions.

**Definition 1.6 (Weak differentiability).** Let \(\alpha \in \mathbb{N}_0^n\) be a multi-index and \(y \in L^p(\Omega)\). If a function \(w \in L^p(\Omega)\) exists such that

\[
\int_{\Omega} w f \, dx = (-1)^{|\alpha|} \int_{\Omega} y (\partial^\alpha f) \, dx.
\]

holds for every test function \(f \in C_0^\infty(\Omega)\), then \(w\) is called a weak derivative of \(y\) of degree \(|\alpha|\) and is denoted by \(\partial^\alpha y\). The function \(y\) is called weakly differentiable of degree \(k\) if the weak derivative exists for every \(\alpha \in \mathbb{N}_0^n\) with \(|\alpha| \leq k\).

**Definition 1.7 (Sobolev spaces).** For \(s \in \mathbb{N}_0\) we define the Sobolev space

\[
W^{s,p}(\Omega) := \{ y \in L^p(\Omega) \mid \partial^\alpha y \in L^p(\Omega), 0 \leq \alpha \leq s \}
\]
which becomes a Banach space if endowed with the norm
\[ \| y \|_{W^{s,p}(\Omega)} = \left( \sum_{|\alpha| \leq s} \int_{\Omega} |\partial^\alpha y(x)|^p \, dx \right)^{1/p}. \]

Note that the derivatives \( \partial^\alpha y \) are to be understood in the weak sense. Particularly interesting is the case \( p = 2 \). This yields a Hilbert space which is denoted by
\[ H^s(\Omega) := W^{s,2}(\Omega). \]

Let us define the problem-specific spaces
\[ V := H^1(\Omega) = \{ y \in L^2(\Omega) \mid \partial_i y \in L^2(\Omega), i = 1, \ldots, n \} \]
and
\[ H := L^2(\Omega) \]
as well as
\[ Y := H^1(t_0, t_f; H) \cap L^2(t_0, t_f; V) \]
with \( \partial_i, i = 1, \ldots, n \), being the partial derivatives for the \( n \)-dimensional domain \( \Omega \). As mentioned before we assume that problem (1.1) admits a unique solution \( y \in Y \) for any parameter \( \mu \in D_\mu \).

**Semigroup theory**

We will only briefly touch the topic of semigroups. The subsequent definitions and theorems follow [5] which we also recommend for deeper studies. Let \( X \) be a Banach space and
\[ \partial_t y = Ay, \ t > 0; \quad y(0) = x \in X \]
a general Cauchy problem with a closed operator \( A \) and dense domain \( \overline{D(A)} = X \).

**Definition 1.8 (C₀-semigroup).** A family \( (T(t))_{t \geq 0} \) of linear and bounded operators on a Banach space \( X \) is called \( C_0 \)-semigroup if

(i) \( T(0) = I \),

(ii) \( T(t + s) = T(t)T(s) \),

(iii) The mapping \( [0, \infty) \to X, t \mapsto T(t)x \) is continuous for all \( x \in X \).
Definition 1.9 (Generator). The linear and unique operator
\[
A : \left\{ x \in X \mid \lim_{t \to 0} \frac{T(t)x - x}{t} \text{ exists} \right\} =: D \to X
\]
\[
Ax := \lim_{t \to 0} \frac{T(t)x - x}{t}
\]
is called generator of the semigroup \( T \) with domain \( D(A) = D \).

Theorem 1.10. Let \( X \) be a Banach space, \( A : D(A) \to X \) closed with \( \overline{D(A)} = X \).
The Cauchy problem (1.5) is well-posed if and only if \( A \) is a generator of a \( C_0 \)-semigroup \( T \) on \( X \).

Application to problem (1.1) In the special case of problem (1.1) with \( g \equiv 0 \),
one can use the semigroup approach in the following way (compare [9]):
Let \( \phi \in L^2(\Omega) \) denote the initial condition and define the time dependent operators
\[
S(t) : L^2(\Omega) \to L^2(\Omega), \quad \phi \mapsto y(t, \cdot; \mu; \phi)
\]
which map the initial condition onto the unique solution at time \( t \) belonging to \( \phi \).
Then the family \( (S(t))_{t \geq 0} \) is a \( C_0 \)-semigroup on \( L^2(\Omega) \) with generator \( A \) defined by
\[
A \phi = -\phi_x + \kappa \phi
\]
\[
\text{dom} A = \{ \phi \in L^2(\Omega) \mid \phi \text{ is absolutely continuous on } [\overline{\Omega}, \overline{\Omega}], \quad \phi(\overline{\Omega}) = 0, \phi_x - \kappa \phi \in L^2(\Omega) \}.
\]
In Chapter 3 we will encounter the generator \( A \) again by setting up a Cauchy problem corresponding to equation (1.1).
2 Ordinary differential equations

The later treatment of problem (1.1) will establish a space discretization leading to an ordinary differential equation of the type

\[
\frac{d}{dt}y(t) = F(t, y(t)), \quad t \in (t_0, t_f) \\
y(t_0) = y_0.
\] (2.1)

Therefore, some results on ordinary differential equations are necessary which are found, for example, in [6, 25, 7, 13]. Let \( Y \) be a Banach space such that \( y(t) \in Y \) for each \( t \in (t_0, t_f) \).

**Definition 2.1 (Space \( L(X, Y) \)).** Let \( X, Y \) be vector spaces. We denote by

\[ L(X, Y) := \{ T : X \to Y \mid T \text{ linear and continuous} \} \]

the space of linear continuous operators \( X \to Y \). We set \( L(X) := L(X, X) \).

**Definition 2.2 (Lipschitz continuity).** Let \( F : (t_0, t_f) \times Y \to Y \) be a continuous function. It is called globally Lipschitz continuous if there exists a constant \( L \geq 0 \) such that for all \( t \in (t_0, t_f) \) and \( y_1, y_2 \in Y \)

\[
\| F(t, y_1) - F(t, y_2) \| \leq L \| y_1 - y_2 \|
\]

with \( \| \cdot \| \) the norm on \( Y \).

**Theorem 2.3 (Picard - Lindelöf).** Let \( y_0 \in Y \) and let \( F : (t_0, t_f) \times Y \to Y \) be globally Lipschitz continuous with constant \( L \geq 0 \). Then there exists a unique solution \( y \in C^1((t_0, t_f); Y) \) to problem (2.1).

The proof of Theorem 2.3 uses the idea of transforming the differential equation (2.1) into an integral equation in terms of

\[
y(t) = y_0 + \int_{t_0}^{t} F(s, y(s)) \, ds
\]

\[= (Ty)(t)\]
and then applying the Banach fixed-point theorem to the problem $y = Ty$ to receive a unique solution.

**Remark 2.4.** A linear system has the form

$$\frac{d}{dt}y(t) = A(t)y(t) + f(t), \quad t \in (t_0, t_f)$$

$$y(0) = y_0. \quad (2.2)$$

Let $A \in C([t_0, t_f], L(Y))$ be a family of bounded operators. In this special case Picard-Lindelöf takes effect since $A$ yields the Lipschitz constant $L$ by defining $L := \sup\{\|A(t)\| \mid t \in [t_0, t_f]\}$ as the supremum over all operator norms. Therefore a unique solution to the differential equation (2.2) exists.

**Theorem 2.5 (Lemma of Gronwall).** Let $a, b \in C([t_0, t_f], \mathbb{R})$ be continuous functions. If

$$\frac{d}{dt}y(t) \leq a(t) + b(t)y(t) \quad \text{for all} \quad t \geq t_0$$

is satisfied then

$$y(t) \leq y(t_0)e^{\int_{t_0}^{t} b(\tau) \, d\tau} + \int_{t_0}^{t} a(s)e^{\int_{s}^{t} b(\tau) \, d\tau} \, ds$$

holds for $t \geq t_0$.

**One-step methods**

To solve problem (2.1) numerically, we will restrict ourselves to one-step methods. Assume that we have a partition of $(t_0, t_f)$ into an equidistant grid with grid size $\Delta t$ leading to the grid points $\{t_0 = t_0, t_1, \ldots, t_{N-1}, t_N = t_f\}$. Each numerical method will yield an approximative value $y_j \approx y(t_j)$ of the exact solution at time $t_j, j = 0, \ldots, N$. To describe the individual methods we introduce the function $\Psi_F$ which is used to calculate the numerical solution by

$$y_0 = y_0,$$

$$y^{j+1} = y^j + \Delta t\Psi_F(t_j, t_{j+1}, y^j, y^{j+1}, \Delta t), \quad j = 0, \ldots, N - 1.$$

The term “one-step method” is due to the fact that only the preceding value $y^j$ is used to determine $y^{j+1}$. We differentiate between explicit and implicit methods meaning that an explicit method requires an function $\Psi_F$ which does not depend on $y^{j+1}$. Otherwise the method is called implicit.
• The **Euler method** is the simplest method of numerical integration. It is an explicit method with the function \( \Psi_F(t_j, t_{j+1}, y^j, y^{j+1}, \Delta t) := F(t_j, y^j) \) which leads to the integration rule

\[
y^{j+1} = y^j + \Delta t F(t_j, y^j).
\]

• The **backward Euler method** is an implicit method which is specified by the function \( \Psi_F(t_j, t_{j+1}, y^j, y^{j+1}, \Delta t) := F(t_{j+1}, y^{j+1}) \) which leads to the integration rule

\[
y^{j+1} = y^j + \Delta t F(t_{j+1}, y^{j+1}).
\]

In this case the method requires in general the solution of an equation to receive the next iterate.

• The **\( \vartheta \)-method** is an interpolation between the Euler and the backward Euler method. Its integration rule is

\[
y^{j+1} = y^j + \Delta t \left( (1 - \vartheta) F(t_j, y^j) + \vartheta F(t_{j+1}, y^{j+1}) \right),
\]

with the parameter \( \vartheta \) in \([0, 1]\). Choosing \( \vartheta = 0 \) yields the Euler method and \( \vartheta = 1 \) the backward Euler method.

To measure the quality of the different methods the concepts of consistency, stability and convergence are introduced. Let \( y(t; t_a, y^a) \) denote the exact solution at time \( t \) for the problem

\[
\frac{d}{dt}y(t) = f(t, y(t)), t \in (t_a, t_f], \quad y(t_a) = y^a.
\]

We denote with \( y_{\Delta t}(t_a + \Delta t; t_a, y^a) = y^a + \Delta t \Psi_F(t_a, y^a, \Delta t) \) the numerical approximation for \( y(t_a + \Delta t; t_a, y^a) \).

**Definition 2.6 (Consistency).** By the term consistency we understand the degree to which the numerical method actually approximates the exact problem. By means of the truncation error

\[
\tau(t_a, y^a, \Delta t) = \frac{y(t_a + \Delta t; t_a, y^a) - y_{\Delta t}(t_a + \Delta t; t_a, y^a)}{\Delta t}
\]

we can determine if a method is consistent. We call the numerical method consistent of order \( p \) if

\[
\|\tau(t_a, y^a, \Delta t)\| = O(\Delta t^p) \quad \text{for} \quad \Delta t \to 0
\]
for all \((t_a, y^a)\) in a neighborhood of the solution trajectory \(\{(t, y(t)), t \in (t_o, t_f)\}\) with \(\|\cdot\|\) denoting the norm in \(Y\).

**Remark (Stability).** Stability is a property of the numerical method which ensures that small changes in the initial condition only cause small changes in the numerical solutions. In other words the errors which are produced by the numerical method have only bounded magnification. A precise definition can be found in [21].

**Definition 2.7 (Convergence).** A numerical method is called convergent of order \(p\) if
\[
\max_{j=0, \ldots, N} \|y(t_j) - y_j\| = O(\Delta t^p) \quad \text{for } \Delta t \to 0
\]
holds.

Convergence is of particular interest as one can determine by it if the numerical solution approaches the exact solution if the grid becomes infinitely fine.

**Remark.** The \(\vartheta\)-method is convergent of order 1 and for \(\vartheta = \frac{1}{2}\) we even receive convergence of order 2. In particular, both the Euler and the backward Euler method are convergent of order 1, compare [22].
3 Optimal control problem

With the preceding theoretical definitions and observations we have now the tools at hand to examine an optimal control problem including the original differential equation (1.1). Let us define an objective functional

\[ J(y,u) = \gamma_1 \frac{1}{2} \int_{t_0}^{t_f} \left| \int_{\Omega} y(t,x) \, dx - y_d(t) \right|^2 \, dt + \sum_{i=1}^{m} \sigma_i \frac{1}{2} \left| u_i - u_i^0 \right|^2, \quad (3.1) \]

which acts on the state \( y \) and on the control \( u \). The weights \( \gamma_1, \sigma_1, \ldots, \sigma_m \in \mathbb{R}_{\geq 0} \) will be specified later. The functional describes a form of costs which occur if the complete population \( \int_{\Omega} y \, dx \) does not match a given desired population \( y_d : [t_0, t_f] \to \mathbb{R} \). It is added up with the costs if the control \( u \) is not close to a given nominal control \( u^0 \in \mathbb{R}^m \). The goal is now to minimize the costs provided that the state equation holds and that all parameters are admissible. For this purpose we define the constraint function

\[
e(\cdot, \cdot; \mu) : X = Y \times U \to L^2(t_0, t_f; H) \times L^2(t_0, t_f) \times H
\]

\[
e(y, u; \mu) = \begin{cases} e_1(y, u; \mu) = y_t + y_x - \kappa(\cdot, u; \mu)y \\ e_2(y, u; \mu) = y(\cdot, x) - g \\ e_3(y, u; \mu) = y(t_0, \cdot) - y_o \end{cases} = 0
\]

which equals zero if and only if the state \( y \) solves the state equation (1.1) for a given control \( u \). The space appearing is \( X = Y \times U \) as well as the set of states and admissible controls \( X_{ad} = Y \times U_{ad} \) with \( U_{ad} \) as specified in (1.2). The optimal control problem which we will analyze is

\[
\min_{(y,u)\in Y \times U} J(y,u) \quad \text{subject to} \quad e(y, u; \mu) = 0, \\
u \in U_{ad}, y \in Y, \mu \in D_\mu. \quad (P_\mu)
\]
Lemma 3.1. The objective functional $J$ is continuously Fréchet differentiable with derivative

$$DJ(x)h = \gamma_1 \int_{t_0}^{t_f} \left( \int \Omega \ y(t, x) \, dx - y_d(t) \right) \int \Omega \ ˜y(t, x) \, dx \, dt + \sum_{i=1}^{m} \sigma_i (u_i - u^*_i) ˜u_i$$

where $x = (y, u) \in X$ is fixed and the direction $h = (˜y, ˜u) \in X$ is arbitrary.

Proof. We will straightforwardly show that

$$F(x, h) := \frac{1}{\|h\|_X} |J(x + h) - J(x) - DJ(x)h| \to 0, \quad \|h\|_X \to 0. \quad (3.2)$$

To do so, we rename the following terms

$$A(t) := \int \Omega y(t, x) \, dx - y_d(t), \quad B(t) := \int \Omega ˜y(t, x) \, dx, \quad C_i := u_i - u^*_i$$

thus being able to write (3.2) as

$$F(x, h) = \frac{1}{\|h\|_X} \left| \frac{\gamma_1}{2} \int_{t_0}^{t_f} (A(t) + B(t))^2 \, dt + \sum_{i=1}^{m} \frac{\sigma_i}{2} (C_i + ˜u_i)^2 \right. \left. - \left( \frac{\gamma_1}{2} \int_{t_0}^{t_f} A(t)^2 \, dt + \sum_{i=1}^{m} \frac{\sigma_i}{2} C_i^2 \right) - \left( \gamma_1 \int_{t_0}^{t_f} A(t)B(t) \, dt + \sum_{i=1}^{m} \sigma_i C_i ˜u_i \right) \right|.$$

Using $\|h\|_X = \sqrt{\|y\|_Y^2 + \|u\|_U^2}$ and the supremum norm $\|\sigma\|_\infty = \max_{i=1, \ldots, m} |\sigma_i|$, this leads to

$$F(x, h) = \frac{1}{\|h\|_X} \left| \frac{\gamma_1}{2} \int_{t_0}^{t_f} B(t)^2 \, dt + \sum_{i=1}^{m} \frac{\sigma_i}{2} u_i^2 \right| \leq \frac{1}{\|h\|_X} \left| \frac{\gamma_1}{2} \|\tilde{y}\|_Y^2 + \frac{\|\sigma\|_\infty}{2} \|\tilde{u}\|_U^2 \right| \leq \max \left\{ \frac{\gamma_1}{2}, \frac{\|\sigma\|_\infty}{2} \|\tilde{u}\|_U^2 \right\} \|h\|_X \to 0, \quad \|h\|_X \to 0$$

and thus the claim. \qed

There are several approaches to optimization problems. We will follow the "first discretize, then optimize" approach which is investigated in [14]. For this purpose a reformulation as a Cauchy problem is carried out, following [9].
Assume that the initial condition satisfies \( y_0 \in L^2(\Omega) \). Let us consider problem (1.1)
\[
\begin{align*}
y_t(t, x) &= -y_x(t, x) + \kappa(t, u; \mu)y(t, x) \quad \text{in } Q \text{ a.e.,} \\
y(t, x) &= g(t) \quad \text{in } (t_0, t_f) \text{ a.e.,} \\
y(t_0, x) &= y_0(x) \quad \text{in } \Omega \text{ a.e.}
\end{align*}
\]
and start with the homogeneous case \( g \equiv 0 \) and the function \( \kappa \) being time-independent. Since the state \( y \) depends both on the time \( t \) and the maturity attribute \( x \) we will abbreviate \( y(t) := y(t, \cdot) \) and look upon \( y(t) \) as a function acting on \( \Omega \). Defining the generator \( A(u) \) for \( u \in U_{ad} \) with its domain (dom) by
\[
\text{dom } A(u; \mu) = \{ \phi \in L^2(\Omega) \mid \phi \text{ is absolutely continuous on } [\underline{x}, \bar{x}], \\
\phi(\underline{x}) = 0, \phi_x - \kappa(u; \mu)\phi \in L^2(\Omega) \}
\]
\[
A(u; \mu) : \text{dom } A(u; \mu) \supset L^2(\Omega) \to L^2(\Omega), \quad A(u; \mu)\phi = -\phi_x + \kappa(u; \mu)\phi
\]
we can re-write (3.3) as
\[
\dot{y}(t) = A(u; \mu)y(t), \quad t \geq t_0, \quad y(t_0) = y_0.
\]

We have seen in Section 1.1 that the time-independent operator \( A(u; \mu) \) with the given domain and function rule is a generator of a \( C_0 \)-semigroup. Thus problem (3.4) is well-posed.

Now let \( g \not\equiv 0 \) which has to be taken into account by modifying a Cauchy problem. We have to define a new sequence of functions \( g_n \) which shall depend on \( x \) as well as on \( t \). These functions are designed to approach the boundary function \( g \) in the limit. Let
\[
\delta_n(x) = \begin{cases} 
-2n^2(x - \bar{x} - \frac{1}{n}) & \text{for } \underline{x} \leq x \leq \bar{x} + \frac{1}{n} \\
0 & \text{otherwise}
\end{cases}, \quad n = 1, 2, \ldots,
\]
be a sequence approximating the \( \delta \)-distribution at \( x = \bar{x} \). Using \( \delta_n \) we can define the functions \( g_n(t, x) = g(t)\delta_n(x) \) which provide us with a system of Cauchy problems
\[
\dot{y}(t) = A(u; \mu)y(t) + g_n(t, \cdot), \quad t \geq t_0, \quad y(t_0) = y_0, \quad n \in \mathbb{N}.
\]
As it is mentioned in [9] the solutions \( y_n \) of (3.5) tend in \( L^2(\Omega) \) to the solution of the original problem with \( g \not\equiv 0 \) as \( n \to \infty \).

The most general problem (3.3) with \( g \not\equiv 0 \) and \( \kappa \) being time-dependent yields a time-dependent operator \( A(t, u; \mu) \). The domain

\[
\text{dom } A(t, u; \mu) = \{ \phi \in L^2(\Omega) \mid \phi \text{ is absolutely continuous on } [x, \bar{x}], \phi(x) = 0, \phi_x - \kappa(t, u; \mu) \phi \in L^2(\Omega) \}
\]

is chosen in such a way that the problem

\[
\dot{y}(t) = A(t, u; \mu)y(t) + g_n(t, \cdot), \quad t \geq t_0, \quad y(t_0) = y_0, n \in \mathbb{N} \quad (3.6)
\]

is well-posed.

**Normalization**

Equation (3.3) is describing a model specifically for the CFU-E cells. Nevertheless it is close to the equations modelling the evolution of other types of red blood cells. The cells are distinguished by their maturity, thus leading to different space intervals in the models. To be able to work with each population class using the same numerical algorithms we normalize the maturity attribute which represents the structural variable in this context. We define the mapping

\[
h : [0, 1] \to [x, \bar{x}], \quad \xi \mapsto x + \omega \xi
\]

with \( \omega \) being the original range of the maturity attribute \( \omega = \bar{x} - x \). Likewise, the inverse is defined as

\[
h^{-1} : [x, \bar{x}] \to [0, 1], \quad x \mapsto \frac{1}{\omega}(x - x).
\]

So far, the basic space was \( L^2(\Omega) \) endowed with the natural norm \( \| \cdot \|_{L^2(\Omega)} \). To transfer the original range \( \omega \) to the normalized problem it is necessary to adapt the norm of the obtained Banach space \( L^2(0, 1) \). The new norm is given by

\[
\| \tilde{\phi} \|_\omega := \omega^{1/2} \| \tilde{\phi} \|_{L^2(0,1)} \text{ for a function } \tilde{\phi} \in L^2(0, 1) \text{ as well as the inner product } \langle \tilde{\phi}, \tilde{\psi} \rangle_\omega = \omega \langle \tilde{\phi}, \tilde{\psi} \rangle_{L^2(0,1)}. \text{ In this way, the space } (L^2(0, 1), \| \cdot \|_\omega) \text{ is a Banach space and is further on denoted with } L^2_\omega. \text{ An isomorphism } \Xi : L^2(0, 1) \to L^2_\omega \text{ is received:}
\]

\[
\Xi \phi = \phi \circ h, \quad \phi \in L^2(0, 1), \quad \text{and} \quad \Xi^{-1} \tilde{\phi} = \tilde{\phi} \circ h^{-1}, \quad \tilde{\phi} \in L^2_\omega.
\]
To normalize the problem the involved functions have to be normalized. Starting with the approximating $\delta$-distributions we get

$$\tilde{\delta}_n(\xi) = (\Xi\delta_n)(\xi) = \left\{ \begin{array}{ll} -2n^2(\omega\xi - \frac{1}{n}) & \text{for } 0 \leq \omega\xi \leq \frac{1}{n} \\ 0 & \text{otherwise} \end{array} \right\}, n = 1, 2, \ldots.$$  

The operator $\tilde{A}(t, u; \mu)$ is defined by

$$\tilde{A}(t, u; \mu) = \Xi A(t, u; \mu)\Xi^{-1} \text{ with } \text{dom } \tilde{A}(t, u; \mu) = \Xi(\text{dom } A(t, u; \mu))$$

$$\Rightarrow \tilde{A}(t, u; \mu)\tilde{\phi} = -\frac{1}{\omega}\tilde{\phi}' + \kappa(t, u; \mu)\tilde{\phi}, \text{ for } \tilde{\phi} \in \text{dom } \tilde{A}(t, u; \mu) \quad (3.7)$$

with precise domain

$$\text{dom } \tilde{A}(t, u; \mu) = \{ \tilde{\phi} \in L^2_{\omega} | \tilde{\phi} \text{ is absolutely continuous on } [0, 1], \tilde{\phi}(0) = 0, \frac{1}{\omega}\tilde{\phi}' - \kappa(t, u; \mu)\tilde{\phi} \in L^2_{\omega}) \}. $$

Note that $\kappa$ is piecewise continuous. For each interval in which $\kappa$ is continuous, $\tilde{A}(t, u; \mu)$ generates a $C_0$-semigroup and hence yields a well-posed problem. We denote with $':=\frac{d}{dx}$ the differentiation with respect to the normalized parameter $\xi$. Altogether the normalization of problem (3.6) reads

$$\dot{\tilde{y}}(t) = \tilde{A}(t, u; \mu)\tilde{y}(t) + g(t)\tilde{\delta}_n, \quad t \geq t_0, \quad \tilde{y}(t_0) = y_0 \circ h. \quad (3.8)$$

### 3.2 Discretization

By now we have a continuous formulation of the state equation as the Cauchy problem (3.8). The solution $\tilde{y}(t) \in L^2_{\omega}, t \in [t_0, t_f]$, is mostly not computable by numerical methods. Therefore, a finite dimensional subspace of $L^2_{\omega}$ is needed in which an approximative solution can be found. As $L^2_{\omega}$ is endowed with an inner product it is desirable to build an orthogonal basis respective to $\langle \cdot, \cdot \rangle_\omega$. To do so we choose the Legendre polynomials as a starting point to assemble a basis.

**Basis functions and subspace**

**Definition** The Legendre polynomials $L_j : [-1, 1] \rightarrow [-1, 1], j \in \mathbb{N}_0$, are given as solutions to a system of partial differential equations which can be found for example in [3]. A recursion formula is

$$(j + 1)L_{j+1}(x) = (2j + 1)xL_j(x) - jL_{j-1}(x), \quad x \in [-1, 1]$$
with initial polynomials
\[ L_0(x) = 1 \quad \text{and} \quad L_1(x) = x, \]
compare [18]. Transformation onto the interval \([0, 1]\) leads us to a definition of the basis functions
\[ e_j(\xi) = \omega^{-1/2}L_j(-1 + 2\xi), \quad 0 \leq \xi \leq 1, \quad j = 0, 1, 2, \ldots, \]
which are later used to build a finite dimensional subspace. The first functions are given by
\[ e_0(\xi) = \omega^{-1/2}, \quad e_1(\xi) = \omega^{-1/2}(-1 + 2\xi), \quad \ldots \]
The subspace which is spanned by the first \(N + 1\) functions \(e_j\) is from now on called
\[ X^N := \text{span}(e_0, \ldots, e_N) \subset L^2_{\omega}. \quad (3.9) \]

**Orthogonality** The most important property of the Legendre polynomials is their orthogonal behaviour respective to the \(\langle \cdot, \cdot \rangle_{L^2(-1, 1)}\) inner product which is transferred directly to the functions \(e_j, j \in \mathbb{N}_0\), respective to \(\langle \cdot, \cdot \rangle_{\omega}\). We end up with
\[ \langle e_k, e_j \rangle_{\omega} = \begin{cases} 0, & j \neq k \\ \omega \int_0^1 e_j(\xi)e_k(\xi) \, d\xi = \frac{1}{2j+1}, & j = k. \end{cases} \quad (3.10) \]
Here we also see that the functions \(e_j\) are orthogonal but not orthonormal.

**Projection** Having now defined the subspace \(X^N\) in (3.9) we have to explore how an orthogonal projection \(P^N : L^2_{\omega} \to X^N\) can be described. Every projection is characterized as a linear combination of the projection on each normalized basis vector
\[ P^N\tilde{\phi} = \sum_{j=0}^N \left< \tilde{\phi}, \frac{e_j}{\|e_j\|_{\omega}} \right>_{\omega} \frac{e_j}{\|e_j\|_{\omega}}, \quad \tilde{\phi} \in L^2_{\omega}. \]
Let us also introduce a notation for a projected vector
\[ P^N\tilde{\phi} = \sum_{j=0}^N \pi_j(\tilde{\phi})e_j =: \begin{pmatrix} \pi_0(\tilde{\phi}) \\ \vdots \\ \pi_N(\tilde{\phi}) \end{pmatrix} \in X^N. \]
By computing the norm and the inner product we obtain with
\[
\pi_j(\tilde{\phi}) = \left(P^N \tilde{\phi} \right)_j = \omega(2j + 1) \int_0^1 \tilde{\phi}(\xi) e_j(\xi) \, d\xi, \quad j = 0, \ldots, N, \tag{3.11}
\]
the projection of \(\tilde{\phi}\) on the \(j\)-th basis vector.

### Discretization of the Cauchy problem

We now aim to establish a discretization of problem (3.8). To do so we analyze the components of the operator \(\tilde{A}(t, u; \mu)\) which consist of a part belonging to the differentiation and a part incorporating the impact of \(\kappa\). For the first part we observe that the derivative of a function in \(X^N\) is again an element in \(X^N\) whereas the multiplication by an arbitrary function usually leads outside the subspace \(X^N\). Therefore it has to be projected back by \(P^N\). In our case \(\kappa\) does not depend on \(x\) and so \(\kappa(t, u; \mu)\tilde{\phi} \in X^N\) for \(\tilde{\phi} \in X^N\). Nevertheless, we will compute exemplarily the projection since the models for other types of cells involve \(x\)-dependent mortality rates. Moreover, the domain of \(\tilde{A}(t, u; \mu)\) requests amongst others that its elements fulfill \(\tilde{\phi}(0) = 0\) which is in general not given for elements of \(X^N\). Thus, a modification of the function rule (3.7) is needed, consisting of the product \(\delta^N\tilde{\phi}(0)\) which includes the approximated delta impulse \(\delta^N \in X^N\), which is given by
\[
\langle \delta^N, \tilde{\phi} \rangle_\omega = \tilde{\phi}(0), \quad \tilde{\phi} \in X^N. \tag{3.12}
\]
The existence and uniqueness of \(\delta^N\) can be proven by applying the Riesz representation theorem. The approximating generator \(\tilde{A}^N(t, u; \mu) : X^N \to X^N\), \(t \in (t_o, t_f), u \in U_{ad}\), is therefore defined by
\[
\tilde{A}^N(t, u; \mu)\tilde{\phi} = -\frac{1}{\omega} \tilde{\phi}^j + P^N \left( \kappa(t, u; \mu)\tilde{\phi} \right) - \delta^N \tilde{\phi}(0), \tag{3.13}
\]
for \(\tilde{\phi} \in X^N\), \(t \in (t_o, t_f), N \in \mathbb{N}\). Finally, the discretized Cauchy problem can be set up as
\[
\frac{d}{dt} \tilde{y}^N(t) = \tilde{A}^N(t, u; \mu)\tilde{y}^N(t) + \delta^N g(t), \quad \tilde{y}^N(t_o) = P^N(y_0 \circ h). \tag{3.14}
\]
Transformation matrices

The next task is to find a transformation matrix for the approximating generator given in (3.13) which is possible due to its linearity. As the generator splits into three different parts it is obvious to compute the transformation matrix for each term separately. Recall the definition of the numerical approximation of \( \tilde{A}(t, u; \mu) \) for \( \tilde{\phi} \in X^N, u \in U_{ad}, N \in \mathbb{N} \) and a fixed time \( t \in (t_o, t_f) \):

\[
\tilde{A}^N(t, u; \mu)\tilde{\phi} = -\omega^{-1}\tilde{\phi}' + P^N(\kappa(t, u; \mu)\tilde{\phi}) - \delta^N(0) =: (A_a(t, u; \mu) + A_b(t, u; \mu) - A_c(t, u; \mu))\tilde{\phi}. \tag{3.15}
\]

We can already see that only the second part \( A_b(t, u; \mu) \) actually depends on time, control and the parameter \( \mu \). Thereby computational benefit in matters of implementation is received since the other two parts can be computed offline. Following this observation we will further on omit the time parameter \( t \), control parameter \( u \) and the parameter \( \mu \) as arguments in \( A_a \equiv A_a(t, u; \mu) \) as well as in \( A_c \equiv A_c(t, u; \mu) \).

**\( A_a \):** The matrix \( A_a \) is mainly determined by the derivative of the basis functions. The derivatives of the Legendre polynomials can be found for example in [15] and dictate the derivatives of the basis functions:

\[
e_j'(\xi) = 2\omega^{-1/2}L_j'(1 + 2\xi) = 2 \left\{ \begin{array}{ll}
\sum_{\nu=0}^{k-1}(4\nu + 3)e_{2\nu+1}(\xi), & j = 2k \\
\sum_{\nu=0}^{k}(4\nu + 1)e_{2\nu}(\xi), & j = 2k + 1.
\end{array} \right.
\]

This already leads to a formulation of the the matrix structure of \( A_a \in \mathbb{R}^{(N+1) \times (N+1)} \):

The lower left triangle including the diagonal consists of zeros. To receive the other entries we consider two different cases:

- \( j = 2k \)

\[
A_a e_j(\xi) = \frac{1}{\omega} \sum_{\nu=0}^{k-1}(8\nu + 6)e_{2\nu+1}(\xi) = \frac{1}{\omega} [6e_1(\xi) + 14e_3(\xi) + \ldots + (4j - 2)e_{j-1}]
\]
• \( j = 2k + 1 \)

\[
A_ne_j(\xi) = \frac{1}{\omega} \sum_{\nu=0}^{k} (8\nu + 2)e_{2\nu}(\xi)
= \frac{1}{\omega} [2e_0(\xi) + 10e_2(\xi) + \ldots + (4j - 2)e_{j-1}]
\]

This leads to the following matrix:

\[
A_n = \frac{1}{\omega} \begin{pmatrix}
0 & 2 & 0 & 2 & 0 & \cdots & 2\delta_{N,\text{odd}} \\
0 & 0 & 6 & 0 & 6 & \cdots & 6\delta_{N,\text{even}} \\
0 & 0 & 0 & 10 & 0 & \cdots & 10\delta_{N,\text{odd}} \\
0 & 0 & 0 & 0 & 14 & \cdots & 14\delta_{N,\text{even}} \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & 0 & 0 & \cdots & 4N - 2 \\
0 & 0 & 0 & 0 & 0 & \cdots & 0
\end{pmatrix} \in \mathbb{R}^{(N+1)\times(N+1)}.
\]

Here \( \delta_{N,\text{odd}} \) denotes a delta-function which equals zero if \( N \) is an even number and one if \( N \) is odd.

\( A_b(t, u; \mu) \): The time- and control-dependent share of \( \tilde{A}^N \) is the second term, namely \( A_b(t, u; \mu)\tilde{\phi} = P^N(\kappa(t, u; \mu)\tilde{\phi}) \). As defined in equation (3.11) we find the image of a basis function \( e_j \) under \( A_b(t, u; \mu) \) as

\[
A_b(t, u; \mu)e_j = P^N(\kappa(t, u; \mu)e_j)
= \omega \left( \sum_{i=0}^{N} (2i + 1) \int_0^1 \kappa(t, u; \mu)e_j(\xi)e_i(\xi) \, d\xi \right) e_i.
\]

If we now recall the definition of \( \kappa \) which is influenced by equation (1.4) we discover that \( \kappa \) effectively only depends on the time \( t \) and control \( u \) but not on the maturity attribute \( x \). This leaves us with

\[
A_b(t, u; \mu)e_j = \omega \kappa(t, u; \mu) \left( \sum_{i=0}^{N} (2i + 1) \int_0^1 e_j(\xi)e_i(\xi) \, d\xi \right) e_i
\quad \overset{(3.10)}{=} \kappa(t, u; \mu)e_j
\]

which implies \( A_b(t, u; \mu) = \kappa(t, u; \mu) \text{Id}_{N+1} \) with the \((N + 1)\)-dimensional identity matrix \( \text{Id}_{N+1} \).
3 Optimal control problem

$A_c$: The last term $A_c\tilde{\phi} = \delta^N\tilde{\phi}(0)$, $\tilde{\phi} \in X^N$, includes the approximating delta pulse $\delta^N \in X^N$. Therefore it is necessary to first compute a representation of $\delta^N$. Let

$$\delta^N = \sum_{j=0}^N \pi_j(\delta^N)e_j$$

with $\pi_j(\delta^N)$ being the projection of $\delta^N$ on the $j$-th basis vector. As given in equation (3.12) the delta pulse is defined by $\langle \delta^N, \tilde{\phi} \rangle_\omega = \tilde{\phi}(0)$ for $\tilde{\phi} \in X^N$. Using the representation above we find for a basis function $e_k, k = 0, \ldots, N$:

$$e_k(0) \overset{\text{orth.}}{=} \pi_k(\delta^N) = \frac{1}{2k+1}$$

Utilizing the knowledge about properties of the Legendre polynomials, we find that they alternate at $x = -1$: $L_j(-1) = (-1)^j$. This immediately leads to

$$e_j(0) = \omega^{-1/2}L_j(-1) = \omega^{-1/2}(-1)^j$$

thus completing the representation of

$$\delta^N = \omega^{-1/2}(1, -3, 5, -7, \ldots)^T \in \mathbb{R}^{(N+1)}.$$

Last, we only need to assemble all components to get

$$A_c e_j = \delta^N e_j(0) = \omega^{-1} \sum_{k=0}^N (2k + 1)(-1)^{k+j}e_k.$$
3.2 Discretization

3.2.1 Semi-discrete optimal control problem

The formulation of the discrete optimal control problem is left to be done. The solution \( \tilde{y}_N(t) \in \mathbb{R}^{N+1}, t \in (t_0, t_f) \), of equation (3.14) demands an alternation of the objective functional defined in equation (3.1). The original objective functional contains the space-integral of the non-discretized solution. By defining the matrix \( D = \text{diag}(\frac{1}{2j+1}, j = 0, \ldots, N) \) the inner product of two functions \( \tilde{\phi}, \tilde{\psi} \in X^N \) can be displayed by

\[
\langle \tilde{\phi}, \tilde{\psi} \rangle_\omega = \omega \int_0^1 \left( \sum_{j=0}^N \pi_j(\tilde{\phi}) e_j(\xi) \right) \left( \sum_{j=0}^N \pi_j(\tilde{\psi}) e_j(\xi) \right) d\xi \]

orth.

\[
= \tilde{\phi}^T D \tilde{\psi}
\]

if we identify the vector \( \tilde{\phi} \in \mathbb{R}^{N+1} \) with entries \( \pi_j(\tilde{\phi}) \) respective to the basis functions \( e_j, j = 0, \ldots, N \), with the function \( \tilde{\phi} \in X^N \). Since \( \tilde{y}_N \) is the approximation for the state variable \( y \), it follows

\[
\int_{\Omega} y(t, x) dx \approx \omega \int_0^1 \sum_{j=0}^N \pi_j(\tilde{y}_N(t)) e_j(\xi) d\xi = \omega^{1/2} \langle \tilde{y}_N(t), e_0 \rangle_\omega = (\tilde{y}_N(t))^T \tilde{D} e_0
\]

with the matrix \( \tilde{D} = \omega^{1/2} D \). Replacing the integral by the discrete inner product the semi-discrete objective functional

\[
J^N : Y^N \times U \rightarrow \mathbb{R}
\]

\[
J^N(\tilde{y}_N, u) = \frac{\gamma_1}{2} \int_{t_0}^{t_f} |(\tilde{y}_N(t))^T \tilde{D} e_0 - y_d(t)|^2 dt + \sum_{i=1}^m \frac{\sigma_i}{2} |u_i - u_i^c|^2 \tag{3.16}
\]

is received with \( Y^N := H^1(t_0, t_f; X^N) \). In order to simplify the notation for the constraint given in equation (3.14) we introduce the semi-discrete constraint function

\[
e^N(\cdot, \cdot; \mu) : Y^N \times U \rightarrow Z^N
\]

\[
e^N(\tilde{y}_N, u; \mu) = \left\{ \begin{array}{l}
e^1_N(\tilde{y}_N, u; \mu) = \frac{d}{dt} \tilde{y}_N - \tilde{A}_N(\cdot, u; \mu) \tilde{y}_N - \delta^N g \\
e^2_N(\tilde{y}_N, u; \mu) = \tilde{y}_N(t_o) - P^N(y_o \circ h) \end{array} \right\} \neq 0
\]
which maps to $Z^N := L^2(t_o, t_f; X^N) \times X^N$. Finally, a semi-discrete optimal control problem approximating $(P_\mu)$ can be set up:

$$\min_{(\tilde{y}^N, u) \in Y^N \times U} J^N(\tilde{y}^N, u) \quad \text{subject to} \quad e^N(\tilde{y}^N, u; \mu) = 0, u \in U_{ad}, \quad \tilde{y}^N \in Y^N, \mu \in D_\mu$$

(P^N_\mu)

Note that the semi-discrete objective functional is continuously Fréchet differentiable which can be seen by an analogous process to the proof of Lemma 3.1. The derivative is given by

$$DJ^N(x^N)h^N = \gamma_1 \int_{t_o}^{t_f} \left( \tilde{y}^N(t)^T \hat{D}e_0 - y_d(t) \right) \tilde{y}^N(t)^T \hat{D}e_0 \, dt + \sum_{i=1}^{m} \sigma_i(u_i - u_i^0)(u_i)$$

for a point $x^N = (\tilde{y}^N, u)$ and a direction $h^N = (y^N_d, u_d)$.

### 3.2.2 Solvability of the semi-discretized state equation

We now aim to prove that equation (3.14) is uniquely solvable. In order to do so, we define $f_1 := \delta^N g \in L^2(t_o, t_f; X^N)$ which yields the equation

$$\frac{d}{dt} \tilde{y}^N(t) = \hat{A}^N(t, u; \mu)\tilde{y}^N(t) + f_1(t)$$

(3.17)

for a fixed control $u \in U_{ad}$. Furthermore, we define $f_2 := P^N(y_o \circ h)$ and the operator $T(u) : Y^N \to Z^N$ by

$$T(u)\tilde{y}^N = \left( \frac{d}{dt} \tilde{y}^N - \hat{A}^N(\cdot, u; \mu)\tilde{y}^N \right) = \left( \begin{array}{c} f_1 \\ f_2 \end{array} \right) \in Z^N.$$

We will prove even more than unique solvability. In fact we will show that the inverse $T(u)^{-1}$ exists and is continuous. Then the bijectivity of $T(u)$ corresponds to the unique solvability of the semi-discretized state equation.

Obviously $T(u)$ is a linear operator. Hence, it is sufficient to show that its kernel only contains zero to prove injectivity. Recall the split of $\hat{A}^N$, introduced in equation (3.15), into two constant matrices and one matrix which is multiplied with the function $\kappa$. This function is defined for every $u \in U_{ad}, U_{ad}$ compact, and is piecewise continuous for $t \in [t_o, t_f]$. Both, $U_{ad} \subset \mathbb{R}^m$ and $[t_o, t_f] \subset \mathbb{R}$ are
bounded and closed and therefore compact. Thus, $\kappa$ attains its maximum and we can estimate the maximum norm

$$
\| \tilde{A}^N(t, u; \mu) \| \leq C_1, \quad t \in [t_o, t_f], u \in U_{ad}
$$

(3.18)

for a global Lipschitz constant $C_1 > 0$ which is independent of $t$ and $u$. The function $\tilde{y}^N = 0$ clearly is a solution of $T(u)\tilde{y}^N = 0$. Uniqueness is obtained by applying Picard-Lindelöf piecewise for each interval in which $\kappa$ is continuous and taking the last state value as new initial value. This is possible since $F(t, \tilde{y}^N(t)) := \tilde{A}^N(t, u; \mu)\tilde{y}^N(t)$ fulfills the Lipschitz condition in each interval. Hence, $T(u)$ is injective. For an arbitrary element $f = (f_1, f_2) \in Z^N$ we have to find a $\tilde{y}^N$ satisfying the equation $T(u)\tilde{y}^N = f$ in order to show surjectivity. Since $\tilde{A}^N(\cdot, u; \mu) \in C([t_o, t_f], L(\mathbb{R}^{N+1}))$ is a bounded operator for a fixed control $u$, Picard-Lindelöf can be applied once more - compare Remark 2.4 - which in combination with the injectivity yields the unique solvability of the state equation. We have implicitly used the fact that the function $f_1 \in L^2(t_o, t_f; X^N)$ can be approximated by continuous functions and so Picard-Lindelöf is applicable, compare for example [8]. Altogether, we receive the invertibility of $T(u)$.

Now we show that $T(u)^{-1}$ is continuous. Let $\tilde{y}^N$ be a solution of $T(u)\tilde{y}^N = f$ for $f = (f_1, f_2) \in Z^N$ arbitrary. Equivalent $\tilde{y}^N = T(u)^{-1}f$ and so we need to show

$$
\| T(u)^{-1}f \|_{Y^N} \leq C\| f \|_{Z^N}
$$

$$
\Leftrightarrow \quad \| \tilde{y}^N \|_{Y^N} \leq C \left( \| f_1 \|_{L^2(t_o, t_f; X^N)}^2 + \| f_2 \|_{X^N}^2 \right)^{1/2}
$$

$$
= C \left( \left\| \frac{d}{dt}\tilde{y}^N - \tilde{A}^N(\cdot, u; \mu)\tilde{y}^N \right\|_{L^2(t_o, t_f; X^N)}^2 + \| \tilde{y}^N(t_o) \|_{X^N}^2 \right)^{1/2}
$$

with $\tilde{y}^N$ being uniquely determined by $f$ and $C > 0$ a constant. Note that $\| \cdot \|_{X^N}$ is another notation for $\| \cdot \|_{\omega}$ which we have used before. The norm

$$
\| \tilde{y}^N \|_{Y^N} = \| \tilde{y}^N \|_{H^1(t_o, t_f; X^N)} = \left( \int_{t_o}^{t_f} \| \tilde{y}^N(t) \|_{X^N}^2 \, dt + \int_{t_o}^{t_f} \left\| \frac{d}{dt}\tilde{y}^N(t) \right\|_{X^N}^2 \, dt \right)^{1/2}
$$

consists of two terms. To estimate the first term, we compute the inner product in $X^N$ of (3.17) with $\tilde{y}^N(t)$ and use that $\frac{1}{2} \frac{d}{dt} \| \tilde{y}^N(t) \|_{X^N}^2 = \langle \frac{d}{dt}\tilde{y}^N(t), \tilde{y}^N(t) \rangle_{X^N}$ for almost all $t$:

$$
\frac{1}{2} \frac{d}{dt} \| \tilde{y}^N(t) \|_{X^N}^2 = \langle \tilde{A}^N(t, u; \mu)\tilde{y}^N(t) + f_1(t), \tilde{y}^N(t) \rangle_{X^N}
$$
Using the boundedness of $\hat{A}^N$, the Cauchy-Schwarz inequality and $a^2 + b^2 \geq 2ab$, yields
\[
\frac{1}{2} \frac{d}{dt} \|\tilde{y}^N(t)\|_{X^N}^2 \leq C_2 \|\tilde{y}^N(t)\|_{X^N}^2 + \frac{1}{2} \|f_1(t)\|_{X^N}^2
\]
to which Gronwall’s inequality from Theorem 2.5 can be applied:
\[
\|\tilde{y}^N(t)\|_{X^N}^2 \leq C_3 \|\tilde{y}^N(t_0)\|_{X^N}^2 + C_4 \int_{t_0}^{t} \|f_1(\tau)\|_{X^N}^2 \, d\tau
\]
with all $C_i > 0$ being positive constants and $\int_{t_0}^{t} \|f_1(\tau)\|_{X^N}^2 \, d\tau$ is bounded since $f_1 \in L^2(t_0, t_f; X^N)$. Also, we have to note that $\tilde{y}_N(t) \in Y^N = H^1(t_0, t_f; X^N)$ and $H^1(t_0, t_f; X^N) \subset C([t_0, t_f]; X^N)$. Therefore, it is justified to compute $\tilde{y}_N(t_0)$. The initial condition satisfies $\tilde{y}_N(t_0) = f_2$ and so integration with respect to time yields the estimate
\[
\|\tilde{y}^N\|_{L^2(t_0, t_f; X^N)}^2 = \int_{t_0}^{t_f} \|\tilde{y}^N(t)\|_{X^N}^2 \, dt \overset{(3.19)}{\leq} C \left( \|f_2\|_{X^N}^2 + \|f_1\|_{L^2(t_0, t_f; X^N)}^2 \right).
\] (3.20)

Now we can perform the main estimate
\[
\|T(u)^{-1}f\|_{Y^N}^2 = \|\tilde{y}^N\|_{Y^N}^2 = \|\tilde{y}^N\|_{L^2(t_0, t_f; X^N)}^2 + \left\| \frac{d}{dt} \tilde{y}^N \right\|_{L^2(t_0, t_f; X^N)}^2 \overset{(3.18)}{\leq} C_1 \|\tilde{y}^N\|_{L^2(t_0, t_f; X^N)}^2 + \|\hat{A}^N(\cdot, u; \mu)\tilde{y}^N + f_1\|_{L^2(t_0, t_f; X^N)}^2 \overset{(3.20)}{\leq} C_2 \left( \|f_1\|_{L^2(t_0, t_f; X^N)}^2 + \|f_2\|_{X^N}^2 \right)
\]
with the appearing constants $C_i > 0$ chosen appropriately. Finally, this proves the continuity of $T(u)^{-1}$ as stated in the beginning.

### 3.3 Optimality criteria

We will re-formulate the semi-discrete optimal control problem $(P^N_\mu)$ using the Lagrange function. Utilizing this formalism, several new aspects about the problem can be expressed like the adjoint equation or the gradient of the reduced functional which will be introduced in Subsection 3.3.3. From now on, we assume that for every $u \in U_{ad}$, the state equation $e^N(\tilde{y}^N, u; \mu) = 0$ has a unique solution $\tilde{y}_N = \tilde{y}_N(u)$. We can formulate the definition of an optimal control:
3.3 Optimality criteria

Definition 3.2 (Optimal control). A control $\tilde{u} \in U_{ad}$ is called optimal for $(P_{\mu}^N)$ and $\tilde{y} = \tilde{y}^N(\tilde{u}) \in Y^N$ is called the associated optimal state if

$$J^N(\tilde{y}, \tilde{u}) \leq J^N(\tilde{y}^N(u), u) \text{ for all } u \in U_{ad}. $$

3.3.1 Lagrange function

Now we follow the notation proposed in [14]. The Lagrange function $L$ is defined by

$$L : Y^N \times U \times Z^N \to \mathbb{R}$$

$$L(\tilde{y}^N, u, p^N) = J^N(\tilde{y}^N, u) + \langle p^N, e^N(\tilde{y}^N, u; \mu) \rangle_{Z^N}.$$

The third argument $p^N = (\tilde{p}^N, \tilde{p}^N_0)$ is called the adjoint state. It will turn out in Section 3.3.2 that $\tilde{p}^N_0 = \tilde{p}^N(t_o)$ so that we will denote it anticipatorily with $p^N = : \tilde{p}^N$. The inner product $\langle \cdot, \cdot \rangle_{Z^N}$ is given by

$$\langle \tilde{p}^N, e^N(\tilde{y}^N, u; \mu) \rangle_{Z^N} = \langle \tilde{p}^N, e^N_1(\tilde{y}^N, u; \mu) \rangle_{L^2(t_o, t_f; X^N)} + \langle \tilde{p}^N(t_o), e^N_2(\tilde{y}^N, u; \mu) \rangle_{\omega}.$$

Assume that $(\tilde{y}, \tilde{u})$ is a pair of locally optimal solutions of Problem $(P_{\mu}^N)$. Then we can formulate necessary optimality conditions of first order which are called KKT-conditions, named after W. Karush, H. Kuhn and A. Tucker. They require certain assumptions, namely:

- $U_{ad} \subset U$ is nonempty, convex and closed.
- $J^N : Y^N \times U \to \mathbb{R}$ and $e^N : Y^N \times U \to Z^N$ are continuously Fréchet differentiable and $U, Y^N, Z^N$ are Banach spaces.
- For all $u \in V$ in a neighborhood $V \subset U$ of $U_{ad}$, the semi-discrete state equation $e^N(\tilde{y}^N, u) = 0$ has a unique solution $\tilde{y}^N = \tilde{y}^N(u) \in Y^N$.
- $e_{\tilde{y}^N}(\tilde{y}^N(u), u) \in L(Y^N, Z^N)$ has a bounded inverse for all $u \in V \supset U_{ad}$.

compare [14]. For the given problem the first three assumptions have already been shown to hold and the last assumption can be found in the proof of Theorem 3.5. If those assumptions hold, and $(\tilde{y}, \tilde{u})$ is an optimal solution of Problem $(P_{\mu}^N)$ then a Lagrange multiplier $\tilde{p} \in Z^N$, also called the adjoint state, exists such that the
optimality conditions hold, compare [14]:

\[
[KKT] \begin{cases}
    \langle \nabla_u L(\bar{y}, \bar{u}, \bar{p}), u - \bar{u} \rangle_U \geq 0, & \forall u \in U_{ad} \\
    \nabla_{\bar{y}N} L(\bar{y}, \bar{u}, \bar{p}) = 0, \\
    \nabla_{\bar{p}N} L(\bar{y}, \bar{u}, \bar{p}) = e^N(\bar{y}, \bar{u}; \mu) = 0.
\end{cases}
\] (3.21)

The first condition is affected by the restriction of the control \( u \in U_{ad} \subset U \). Unrestricted problems can demand that \( \bar{u} \) is a root of each directional derivative \( \nabla_u L(\bar{y}, \bar{u}, \bar{p})u_\delta, u_\delta \in U \). In the next section we will see how the optimal adjoint state \( \bar{p} \) is defined by the adjoint equation which is given by the second KKT-condition. The Fréchet derivatives in (3.21) are the following:

\[
\nabla_u L(\bar{y}^N, u, \bar{p}^N)u_\delta = \langle \nabla_u J^N(\bar{y}^N, u), u_\delta \rangle_{\mathbb{R}^m} + \langle \bar{p}^N, \nabla_u e^N(\bar{y}^N, u; \mu)u_\delta \rangle_{Z^N}
\]

\[
= \sum_{i=0}^{m} \sigma_i(u_i - u_i^0)(u_\delta)_i
\]

\[
+ \int_{t_0}^{t_f} \langle \nabla_u \alpha(t, u; \mu), u_\delta \rangle_{\mathbb{R}^m} (\bar{y}^N(t))^T D\bar{p}^N(t) \, dt \quad (3.22)
\]

and

\[
\nabla_{\bar{y}N} L(\bar{y}^N, u, \bar{p}^N)y_\delta = \nabla_{\bar{y}N} \left[ J^N(\bar{y}^N, u) + \langle \bar{p}^N, e^N(\bar{y}^N, u; \mu) \rangle_{(Z^N)^*, Z^N} \right] y_\delta
\]

\[
= \gamma_1 \int_{t_0}^{t_f} \left( (\bar{y}^N(t))^T \tilde{D}e_0 - y_\delta(t) \right) (y_\delta(t))^T \tilde{D}e_0 \, dt
\]

\[
+ \int_{t_0}^{t_f} \left( \frac{d}{dt} y_\delta(t) - \tilde{A}^N(t, u; \mu)y_\delta(t) \right)^T D\bar{p}^N(t) \, dt
\]

\[
+ (y_\delta(t_0))^T D\bar{p}^N(t_0). \quad (3.23)
\]

for directions \( u_\delta \in U_{ad} \) and \( y_\delta \in Y^N \).

### 3.3.2 Adjoint equation

To gain a description of the adjoint equation we use \( \nabla_{\bar{y}N} L(\bar{y}^N(u), u, \bar{p}^N(u))y_\delta = 0 \) which has to hold for every \( y_\delta \in Y^N \) assuming that \( u = \bar{u} \) is optimal. We will use the time derivative of the adjoint state in fact is an element of \( H^1(t_0, t_f; X^N) \). We will see in equation (3.25) that the time derivative is given by an element in \( L^2(t_0, t_f; X^N) \) and thus \( \bar{p}^N \in H^1(t_0, t_f; X^N) \).

We return to the original notation \( p = (\bar{p}^N, \tilde{p}_0^N) \) which leads to a replacement of \( \bar{p}^N(t_0) \) by \( \tilde{p}_0^N \) in (3.23). Then partial integration of (3.23) with respect to time
3.3 Optimality criteria

yields

\[
\gamma_1 \int_{t_0}^{t_f} \left( (\tilde{y}^N(t))^T \tilde{D}e_0 - y_0(t) \right) (y_\delta(t))^T \tilde{D}e_0 \ dt + (y_\delta(t_f))^T D\tilde{p}_N(t_f)
- (y_\delta(t_0))^T D\tilde{p}_N(t_0) - \int_{t_0}^{t_f} \left( \frac{d}{dt} \tilde{p}^N(t) + \tilde{A}^N(t, u; \mu)^T \tilde{p}^N(t) \right)^T Dy_\delta \ dt
+ (y_\delta(t_0))^T D\tilde{p}_0^N = 0, \quad y_\delta \in Y^N. \quad (3.24)
\]

Equation (3.24) has to hold for every \( y_\delta \in Y^N \). Therefore, we can choose particular directions \( y_\delta \) with \( y_\delta(t_f) = y_\delta(t_0) = 0 \). These directions yield the following equation

\[
\int_{t_0}^{t_f} \left[ \gamma_1 \omega^{1/2} \left( \tilde{y}^N(t)^T \tilde{D}e_0 - y_0(t) \right) e_0^T
- \left( \frac{d}{dt} \tilde{p}^N(t) + \tilde{A}^N(t, u; \mu)^T \tilde{p}^N(t) \right)^T \right] Dy_\delta \ dt = 0
\]

where we have used \( \tilde{D} = \omega^{1/2} D \). Finally, this determines the adjoint equation

\[
\frac{d}{dt} \tilde{p}^N(t) = -\tilde{A}^N(t, u; \mu)^T \tilde{p}^N(t) + \gamma_1 \omega^{1/2} \left( \tilde{y}^N(t)^T \tilde{D}e_0 - y_0(t) \right) e_0
\]

which holds almost everywhere. The right hand side of (3.25) is an element in \( L^2(t_0, t_f; X^N) \) since \( \tilde{A}^N(t, u; \mu) \) is piecewise continuous over time. Therefore, \( \frac{d}{dt} \tilde{p}^N \in L^2(t_0, t_f; X^N) \) holds. Inserting (3.25) into (3.24) leads to

\[
(y_\delta(t_f))^T D\tilde{p}_N(t_f) - (y_\delta(t_0))^T D\tilde{p}_N(t_0) + (y_\delta(t_0))^T D\tilde{p}_0^N = 0. \quad (3.26)
\]

Choosing \( y_\delta \) with \( y_\delta(t_f) = 0 \) in (3.26) produces \( \tilde{p}_0^N = \tilde{p}_N(t_0) \) as stated before. Thus, equation (3.26) yields the final condition

\[
\tilde{p}_N(t_f) = 0
\]

for the adjoint state.

3.3.3 Reduced problem

Assume that for every \( u \in U_{ad} \) there exists exactly one solution \( \tilde{y}^N(u) \) of Equation (3.14). So we can define a reduced objective functional \( \hat{J}^N \) which only depends on the control \( u \):

\[
\hat{J}^N(u; \mu) := J^N(\tilde{y}^N(u), u; \mu), \quad u \in U_{ad}
\]
An equivalent minimization problem is formulated as

$$\min_{u \in U} \hat{J}^N(u; \mu) \quad \text{subject to} \quad u \in U_{ad}, \mu \in D_{\mu}. \quad (\hat{P}_\mu^N)$$

To solve $(\hat{P}_\mu^N)$ numerically - applying for example the gradient method - it is necessary to derive a formula for $\nabla \hat{J}^N(u; \mu)$. There exists a relationship between the derivative of the reduced gradient and the derivative of the Lagrangian which is now presented following [14].

**Derivative of the gradient of the reduced cost functional**

We will show that the derivation of the reduced cost functional equals the derivation of the Lagrangian with respect to $u$. In order to do so, a simplified notation is used in this paragraph for the theoretical observations and computations. To indicate the dependence of the state which is uniquely determined by the given control we write $y(u)$. This defines a solution operator $U_{ad} \rightarrow Y^N, u \mapsto y(u)$. Now $u_\delta \in U_{ad}$ shall be an arbitrary direction, $p(u)$ the associated adjoint state, $e$ the constraint function, $J$ the objective functional and $L$ the corresponding Lagrangian. We further assume that $e_y(y(u), u)$ has a bounded inverse. We need three steps:

1. $y'(u)u_\delta = -\nabla_y e(y(u), u)^{-1}\nabla_u e(y(u), u)u_\delta$

2. $p(u) = -\nabla_y e(y(u), u)^{-*}\nabla_y J(y(u), u)$

3. $\hat{J}'(u)u_\delta = \nabla_u L(y(u), u, p(u))u_\delta$

where the derivative with respect to $u$ is denoted with a prime. The first two equations have to be used to show the third one which yields the needed derivative.

1. The first equation follows directly from differentiating the side condition $e \equiv 0$ in direction $u_\delta$ using the chain rule:

$$\nabla_y e(y(u), u)y'(u)u_\delta + \nabla_u e(y(u), u)u_\delta = 0.$$  

Note that $y'(u) \in L(U, Y^N)$.

2. The equation just defines the adjoint equation which is $\nabla_y e(y(u), u)^*p = -\nabla_y J(y(u), u)$ and is solved by the adjoint state $p(u)$. This is described more precisely in [14].
3. The derivative of the Lagrangian with respect to $u$ in direction $u_\delta$ is given by:

$$
\nabla_u L(y, u, p) u_\delta = \langle \nabla_u J(y, u), u_\delta \rangle_{\mathbb{R}^m} + \langle \nabla_u e(y, u) u_\delta, p \rangle_{Y^N} \\
= \langle \nabla_u J(y, u) + (\nabla_u e(y, u))^* p, u_\delta \rangle_{\mathbb{R}^m}.
$$

Choosing $y = y(u)$ we receive

$$
\nabla_u L(y(u), u, p) u_\delta = \langle \nabla_u J(y(u), u) + (\nabla_u e(y(u), u))^* p, u_\delta \rangle_{\mathbb{R}^m}. \quad (3.27)
$$

What is missing is the computation of the derivative of the reduced cost functional in direction $u_\delta$:

$$
\hat{J}'(u) u_\delta = \langle \nabla_y J(y(u), u), y'(u) u_\delta \rangle_{Y^N} + \langle \nabla_y e(y(u), u)^{-1} \nabla_u e(y(u), u) u_\delta, u_\delta \rangle_{Y^N} \\
\overset{1}{=} \langle \nabla_y J(y(u), u) - \nabla_y e(y(u), u)^{-1} \nabla_u e(y(u), u) u_\delta, u_\delta \rangle_{Y^N} \\
+ \langle \nabla_y J(y(u), u), u_\delta \rangle_{Y^N} \\
\overset{2}{=} \langle p(u), \nabla_u e(y(u), u) u_\delta \rangle_{Y^N} + \langle \nabla_u J(y(u), u), u_\delta \rangle_{Y^N} \\
= \langle \nabla_u J(y(u), u) + \nabla_u e(y(u), u)^* p(u), u_\delta \rangle_{\mathbb{R}^m}. \quad (3.28)
$$

From (3.27) and (3.28) we conclude:

$$
\hat{J}'(u) u_\delta = \nabla_u L(y(u), u, p(u)) u_\delta.
$$

This gives us a way to compute the gradient of the reduced objective functional. In Equation (3.22) the derivative is already given and thus the reduced gradient reads

$$
(\nabla \hat{J}^N(u; \mu))_i = \sigma_i (u_i - \bar{u}_i^0) + \int_{t_0}^{t_f} (\nabla_u \alpha(t, u; \mu))_i (\tilde{y}^N(t))^T D\tilde{p}^N(t) \, dt, \ i = 1, \ldots, m. \quad (3.29)
$$

Notice that the gradient of $\alpha$ at $\bar{u} = (\bar{u}_i) \in U_{ad}$ equals

$$
\nabla_u \alpha(t, \bar{u}; \mu) = \bar{\alpha}(t, \bar{u}; \mu) \begin{pmatrix} \chi_1(t), \ldots, \chi_m(t) \end{pmatrix} \in \mathbb{R}^{1 \times m} \quad (3.30)
$$
with the $\bar{u}$ dependent coefficient function

$$\bar{a}(t, \bar{u}; \mu) = -\frac{\mu_3}{TBV} \exp \left( \mu_2 \sum_{i=1}^{m} \bar{u}_i \chi_i(t) / TBV - \mu_3 \right).$$

Having now assembled different properties of the reduced problem, we can give an alternative formulation of the first KKT condition of (3.21).

**Definition 3.3 (Stationary control).** A control $u^* \in U_{ad}$ is called stationary for problem $(\hat{P}_N^\mu)$ if

$$\nabla \hat{J}_N(u^*)^T (u - u^*) \geq 0 \quad \text{for all} \quad u \in U_{ad}. \quad (3.31)$$

The first KKT condition in (3.21) is now equivalent to the following theorem which can also be found in [16]:

**Theorem 3.4.** Let $\hat{J}_N^\mu$ be continuously Fréchet differentiable on $U_{ad}$ and let $\bar{u}$ be a solution of problem $(\hat{P}_N^\mu)$. Then $\bar{u}$ satisfies (3.31).

**Proof.** The problem will be led back to a one-dimensional problem. Let $[a, b] \subset \mathbb{R}$ be a closed interval. For a continuously differentiable function $f : [a, b] \to \mathbb{R}$ with a local minimum in $x^* \in [a, b]$ we receive

$$f'(x^*)(x - x^*) \geq 0, \quad x \in [a, b]. \quad (3.32)$$

The proof follows by case-by-case analysis. Either $x^* \in (a, b)$ and thus $f'(x^*) = 0$ as in the unconstrained case or $x^* \in \{a, b\}$. In the second case the claim follows by examining the sign of the difference quotient which yields the derivative of $f$ in the limit. Assume now that $\bar{u}$ is a solution of problem $(\hat{P}_N^\mu)$ and let $u \in U_{ad}$ be arbitrary. The convexity of $U_{ad}$ as given in (1.2) with $u_a = 0, u_b = 1$ yields $\bar{u} + t(u - \bar{u}) \in U_{ad}$ for $t \in [0, 1]$. Hence the function $\phi$ defined on this line segment by the rule $\phi(t) = \hat{J}_N^\mu(\bar{u} + t(u - \bar{u}))$ has a local minimum at $t = 0$ and thus

$$0 \leq \phi'(0) = \nabla \hat{J}_N^\mu(\bar{u})^T (u - \bar{u}). \quad (3.32)$$

**Remark.** Note that the constraint function $e_N$ as well as the reduced objective functional $\hat{J}_N^\mu$ are twice continuously Fréchet differentiable. This implies amongst others the local Lipschitz continuity of $\nabla \hat{J}_N^\mu$ on the bounded and convex subset $U_{ad} \subset U$, compare for example [16].
3.3 Optimality criteria

**Theorem 3.5.** The reduced Problem \((\hat{\mathbf{P}}^N_\mu)\) possesses a solution \(u^* \in U_{ad}\). Hence, there is at least one solution to \((\hat{\mathbf{P}}^N_\mu)\).

**Proof.** The proof is mainly based on the theorem of Weierstrass which states that a real-valued continuous function on a compact subset of \(\mathbb{R}^m\) attains its minimum. We have to show that the control-to-state operator \(S : u \mapsto \tilde{y}^N\), with \(\tilde{y}^N\) being the unique solution of the semi-discrete state equation (3.14), is continuous. If we show that the constraint function \(e^N\) is continuously Fréchet differentiable and the derivative with respect to the state \(e^N(\tilde{y}^N, u) \in L(Y^N, Z^N)\) possesses a bounded inverse, then the implicit function theorem yields the continuity of the control-to-state operator \(S\), compare [14]. Hence, \(J^N\) and thus \(\hat{J}^N\) as a composition of the functionals \(J^N\) and \(S\) is continuous in \(u\). With the compactness of \(U_{ad} \subset U\) we obtain the claim immediately.

It is left to prove that \(e^N\) is continuously Fréchet differentiable and its derivative with respect to the state is continuously invertible. The Fréchet derivative at a point \((\tilde{y}^N, u) \in Z^N\) in a direction \((\hat{y}, \hat{u})\) is given by

\[
(e^N(\tilde{y}^N, u))'(\tilde{y}, \hat{u}) = \left( \frac{d}{dt} \tilde{y}(t) + \langle \nabla_u \alpha(t, u; \mu), \hat{u} \rangle \tilde{y}^N - \bar{A}^N(t, u; \mu) \tilde{y} \right)
\]

which can be seen by a straightforward computation as already carried out in the proof of Lemma 3.1.

The appearing \(\nabla_u \alpha(t, u; \mu)\) is given in equation (3.30). The derivative with respect to the state is given by \(e^N(\tilde{y}^N, u; \mu)\hat{y} = \left( \frac{d}{dt} \tilde{y}(t) - \bar{A}^N(t, u; \mu) \tilde{y} \right)\). Its invertibility is proven in exactly the same manner as it was done in Subsection 3.2.2.

To prepare the discussion of numerical methods in Chapter 4 we will distinguish inactive and active indices.

**Definition 3.6 (Active indices).** The set of admissible controls is determined by inequality constraints \(u_a \leq u \leq u_b\) for \(u \in U_{ad}\). The \(i\)-th constraint, \(i \in \{1, \ldots, m\}\), and accordingly the index \(i\) is called active if \(u_i = (u_a)_i\) or \(u_i = (u_b)_i\). Otherwise it is called inactive. The set of active indices at \(u\) will be denoted with \(\mathcal{A}(u)\) and the set of inactive indices with \(\mathcal{I}(u)\).

To distinguish between \(\mathcal{A}(u)\) and \(\mathcal{I}(u)\) will be crucial for some numerical methods described in Chapter 4.
In Chapter 3 the reduced optimal control problem \( \hat{P}_N^\mu \) was derived. The goal in this chapter now is to formulate numerical methods which are designed to compute an optimal control in an iterative way. Most methods consist of several steps including the search of a descent direction, the computation of an appropriate step size and the set up of suitable termination criteria. In this chapter we focus on the so-called open loop process which means that controls are computed iteratively using one of the algorithms defined consecutively. During the iteration process updates of parameters cannot be executed in contrast to the closed loop process explained in Chapter 5. Such, the term open loop arises from the assumption that all incoming data are known in advance and then the optimization process is started without the option to turn back and adapt parameters which might have changed due to external perturbations. Hence, it represents a one-way timeline. The open loop solver stops when a certain termination criterion is fulfilled. We will start with the gradient method which can be found for example in [16] or [24].

4.1 Projected gradient method

The main idea of the gradient method (GM) is to use the negative gradient of the reduced objective functional as the direction of steepest descent. The major advantage is the reduction of the objective function values by using the steepest descent method and the major disadvantage is that the speed of the iteration process might be extremely low.

4.1.1 Unconstrained case

First, suppose that \( U_{ad} = U \), meaning that the optimality problem \( \hat{P}_N^\mu \) is not restricted. Now let \( u_c \in U_{ad} \) be the current control. The next control \( u_+ \) will be computed by

\[
    u_+ = u_c + td
\]
with a direction $d \in U$ and a step size $t > 0$. For the steepest descent we use $d = -\nabla \hat{J}^N(u_c)$. In Algorithm 4.1 the steepest descent method is formulated.

**Algorithm 4.1** Steepest descent method with Armijo step size strategy

1: Given: $u^0$
2: while Termination criterion not fulfilled do
3: Compute the gradient $\nabla \hat{J}^N(u^j)$
4: Compute a step size $t_j$ with the Armijo rule, compare Algorithm 4.2
5: Compute the next control $u^{j+1} = u^j - t_j \nabla \hat{J}^N(u^j)$
6: $j = j + 1$
7: end while

It is indispensable to choose a proper step size $t$ to obtain a descent of the function values $\hat{J}^N(u_+) \leq \hat{J}^N(u_c)$. This will be carried out using the Armijo rule.

**Step size control using the Armijo rule**

The Armijo rule for the optimal step size is given by

$$\hat{J}^N(u_c + td) - \hat{J}^N(u_c) \leq \alpha t \nabla \hat{J}^N(u_c)^T d$$ (4.1)

for a parameter $\alpha \in (0, 1)$ chosen in advance. In general, the value $\alpha = 10^{-4}$ often appears to be a good choice, see [16]. To execute the search of the step size $t$ a backtracking strategy is carried out: Choose $\beta \in (0, 1)$ and an initial step size $t_0 > 0$ and then compute iteratively

$$t_n = \beta^n t_0, \quad n = 0, 1, \ldots,$$

until equation (4.1) holds for an $n \in \mathbb{N}_0$. This $t_n$ is chosen to compute the next control $u_+ = u_c - t_n \nabla \hat{J}^N(u_c)$. Formulating these steps in an algorithm ends in

**Algorithm 4.2** Armijo strategy

1: Given: $u, d, t_0, \alpha, \beta$
2: while $\hat{J}^N(u + td) - \hat{J}^N(u) > \alpha t \nabla \hat{J}^N(u)^T d$ do
3: Compute new step size: $t = \beta t$
4: Compute new control: $u = u + td$
5: Compute new state variable $\hat{y}^N(u)$ as the solution to the discretized state equation (3.14) defined by the control $u$
6: Compute the reduced cost functional $\hat{J}^N(u)$ by (3.16)
7: end while
For the gradient method with the Armijo step size strategy a global convergence result is found:

**Theorem 4.1.** Let $\hat{J}^N$ be continuously differentiable. Then Algorithm 4.1 either terminates with a stationary point $u^k$ or it generates an infinite sequence $(u^k)_{k \in \mathbb{N}}$ with

1. for all $k \in \mathbb{N}$: $\hat{J}^N(u^{k+1}) < \hat{J}^N(u^k)$ and
2. every limit point of $(u^k)_{k \in \mathbb{N}}$ is a stationary point of $\hat{J}^N$.

For the proof of Theorem 4.1 we refer to [24].

### 4.1.2 Constrained case

Now assume $U_{\text{ad}} \subsetneq U$ leading to a constrained optimization problem $(\hat{P}^N_\mu)$. Therefore, we have to guarantee that every iterate $u^k$ is admissible. The most basic way to do so is to include a projection in the used algorithms. Define the projection $P : U \rightarrow U_{\text{ad}}$ by

$$
(P(u))_i = \begin{cases}
(u_a)_i, & u_i < (u_a)_i \\
(u_a)_i, & (u_a)_i \leq u_i \leq (u_b)_i \\
(u_b)_i, & u_i > (u_b)_i
\end{cases}
$$

projecting $u \in U$ to its closest point in $U_{\text{ad}}$ in terms of the $l^2$-norm. Now we can formulate an equivalent expression of stationarity as given in Definition 3.3:

**Theorem 4.2.** Let $\hat{J}^N$ be continuously Fréchet differentiable. Then $u^* \in U_{\text{ad}}$ is stationary for problem $(\hat{P}^N_\mu)$ if and only if

$$u^* = P(u^* - \lambda \nabla \hat{J}^N(u^*))$$

for all $\lambda \geq 0$.

For the proof we refer to [16].

**Corollary 4.3.** Let $\hat{J}^N$ be continuously Fréchet differentiable on $U_{\text{ad}}$. If $u^*$ is stationary then $u^* - P(u^* - \nabla \hat{J}^N(u^*)) = 0$.

From this corollary a termination condition will be deduced. It is easy to see that for an inactive index $i \in I(u^*)$, this component in the gradient must equal zero, $(\nabla \hat{J}^N(u^*))_i = 0$, since this can be traced back to unconstrained optimality criteria.
Now the gradient method using the Armijo rule can be modified to be able to act on constrained problems. First, every new control depends on the computed step size as well as on the direction $d$ and can be written as

$$u(t, d) := P(u + td),$$

including the projection back onto the set of admissible controls. This syntax enables us to write down a modified Armijo condition for the step size $t$

$$\hat{J}^N(u(t, d)) - \hat{J}^N(u) \leq -\frac{\alpha}{t} \|u - u(t, d)\|^2 (4.3)$$

again taking $d = -\nabla \hat{J}^N(u)$ for the steepest descent and $\| \cdot \| := \| \cdot \|_{\mathbb{R}^m}$ since $U_{ad} \subset \mathbb{R}^m$. Typically, $\alpha$ is set to $10^{-4}$ as in the unconstrained case. Assuming that the gradient $\nabla \hat{J}^N$ in $U_{ad}$ is Lipschitz continuous with Lipschitz constant $L$ it can be shown that (4.3) holds for all $t \in \left(0, \frac{2(1-\alpha)}{L}\right)$, compare for example [16].

The pseudo code for the modified Armijo strategy is found in Algorithm 4.3.

**Algorithm 4.3 Modified Armijo strategy**

1. Given: $u, d, t_o, \alpha, \beta$
2. while $\hat{J}^N(u(t, d)) - \hat{J}^N(u) > -\frac{\alpha}{t} \|u - u(t, d)\|^2$ do
3. Compute new step size $t = \beta t$
4. Compute control $u(t, d) = P(u + td)$
5. Compute state variable $\tilde{y}^N(u(t, d))$ as the solution to the discretized state equation (3.14) defined by the control $u(t, d)$
6. Compute reduced cost functional at $u(t, d)$
7. end while

Using the modified Armijo strategy, the iterates computed by the projected gradient method are obtained according to Algorithm 4.1 with the following modifications: replace the step size in Line 4 by the step size yielded by the modified Armijo Algorithm 4.3 and define the new control $u^{j+1} := u(t, d)$. A statement about the convergence of the algorithm can be drawn for Lipschitz continuous functions.

**Theorem 4.4.** Assume that $\nabla \hat{J}^N$ is Lipschitz continuous on $U_{ad}$ and let $(u^k)_{k \in \mathbb{N}}$ be the sequence generated by the gradient projection method. Then every limit point of the sequence is a stationary point.

A proof can be found in [16].
4.2 Projected BFGS method

The BFGS method, which is named after Broyden, Fletcher, Goldfarb and Shanno, represents a Quasi-Newton method which approximates the Hessian $\nabla^2 \hat{J}^N(u)$. Using the BFGS method, we obtain the possibility to improve the descent direction by including knowledge about the curvature of the function. For further information we refer the reader to [16, 20].

4.2.1 Unconstrained case

Assume again that no constraints are given, i.e. $U_{ad} = U$. The matrix approximations have to satisfy the secant equation

$$B_{k+1} s^k = q^k$$

with

$$s^k := u^{k+1} - u^k \in \mathbb{R}^m, \quad q^k := \nabla \hat{J}^N(u^{k+1}) - \nabla \hat{J}^N(u^k) \in \mathbb{R}^m.$$ 

Given that $(q^k)^T s^k > 0$ and $B_k$ positive definite, the BFGS update is obtained by

$$B_{k+1} = B_k - \frac{B_k s^k (B_k s^k)^T}{(s^k)^T B_k s^k} + \frac{q^k (q^k)^T}{(q^k)^T s^k}, \quad B_0 = \|\nabla \hat{J}(u^0)\| \cdot \text{Id}_m. \quad (4.4)$$

Using $B_k$, the update of the control is realized as

$$u^{k+1} = u^k - t_k B_k^{-1} \nabla \hat{J}(u^k).$$

In numerical realization the linear system $B_k d_k = -\nabla \hat{J}(u^k)$ is solved first and the new control is obtained by $u^{k+1} = u^k + t_k d_k$. One possibility for the choice of $t_k$ is $t_k = 1$ which means that no step size control is executed. By choosing $t_k$ by a step size algorithm - for example the Armijo rule - globalization can be conducted.

The advantage of defining $B_0 = \|\nabla \hat{J}(u^0)\| \cdot \text{Id}_m$ is to start with a positive definite matrix and the normed steepest descent direction $d_0 = -\nabla \hat{J}^N(u^0)/\|\nabla \hat{J}^N(u^0)\|$.

**Lemma 4.5.** Let $B_k$ be positive semi-definite and $(q^k)^T s^k > 0$. Then $B_{k+1}$ given by equation (4.4) is positive semi-definite.

For the proof we refer to [16].

Let us assume that $\hat{J}^N$ is twice continuously differentiable, the level set $\{u \in U \mid \hat{J}^N(u) \leq \hat{J}^N(u^0)\}$ is convex and the matrix of second derivatives of $\hat{J}^N$ is bounded. Then, we find that the BFGS method together with the Armijo line
search generates iterates converging to the optimal control \( u^* \), compare [4].

In the case that the curvature condition \( (q^k)^T s^k > 0 \) does not hold we have several options how to carry out the update of the BFGS matrix. It is possible to reinitialize the matrix to the identity matrix which can additionally be multiplied by the norm of the gradient in order to obtain a normalized descent direction. Another possibility is to use the previously computed BFGS matrix to keep the iteration history. Both strategies are used in practice.

### 4.2.2 Constrained case

In the constrained case we assume again \( U_{ad} \subset U \) which leads to the constrained optimization problem \((\hat{P}_\mu^N)\). The projected BFGS-Armijo method consists of updates of the BFGS matrix as in the unconstrained case but includes projections of the differences \( s^k \) and \( q^k \) following the approach of [16]. Recall the definition of active and inactive sets as given in Definition 3.6 and introduce for \( \varepsilon > 0 \) the more general \( \varepsilon \)-active set at \( u \) as

\[
\mathcal{A}^\varepsilon(u) = \{ i \in \{1, \ldots, m\} \mid (u_b)_i - u_i \leq \varepsilon \text{ or } u_i - (u_a)_i \leq \varepsilon \}
\]

and the \( \varepsilon \)-inactive set at \( u \) as its complement \( \mathcal{T}^\varepsilon(u) = \{1, \ldots, m\} \setminus \mathcal{A}^\varepsilon(u) \). For any set \( S \subset \{1, \ldots, m\} \) of indices we define the projection

\[
(P_S u)_i = \begin{cases} 
  u_i, & i \in S \\
  0, & i \notin S 
\end{cases}
\]

and denote the transformation matrix with \( P_S \) as well. Likewise, \( P_{\mathcal{A}^\varepsilon(u)} \) denotes the projection on the \( \varepsilon \)-active indices of \( u \) and \( P_{\mathcal{T}^\varepsilon(u)} \) the projection on the \( \varepsilon \)-inactive indices of \( u \). Define the elements

\[
q^k_\# = P_{\mathcal{I}_{k+1}} (\nabla \hat{j}^N(u_{k+1}) - \nabla \hat{j}^N(u^k)) \quad \text{and} \quad s^k_\# = P_{\mathcal{I}_{k+1}} (u_{k+1} - u^k)
\]

with \( \mathcal{I}_{k+1} = \mathcal{T}^\varepsilon(u^{k+1}) \). These can be used for the update on the \( \varepsilon \)-inactive set

\[
\hat{B}_{k+1} = P_{\mathcal{I}_{k+1}} \hat{B}_k P_{\mathcal{I}_{k+1}} + \frac{q^k_\# (q^k_\#)^T}{(q^k_\#)^T s^k_\#} \frac{(B_k s^k_\#)(B_k s^k_\#)^T}{(s^k_\#)^T B_k s^k_\#} P_{\mathcal{I}_{k+1}}
\]

in case \( (q^k_\#)^T s^k_\# > 0 \) holds. If not, the same options are available as in the unconstrained case so either skip the update or reinitialize \( \hat{B} \) to \( P_{\mathcal{T}} \). The complete
4.3 Termination condition

The goal of the open loop solver is to generate iterates \( u^k \) until they are close enough to the optimal control. As stated in Corollary 4.3, a stationary control \( u^* \) fulfills \( u^* = P(u^* - \nabla \hat{J}^N(u^*)) \) or in other words

\[
u^* = u^*(1)\]

if we define \( u(\lambda) = P(u - \lambda \nabla \hat{J}^N(u)) \) with a step length parameter \( \lambda > 0 \). To conclude, it suggests itself to measure \( \|u - u(1)\|_{\mathbb{R}^m} \) to receive a suitable termination

\[
B_{k+1} = \mathcal{P}_{A_k} + \hat{B}_{k+1}, \quad B_0 = \|\nabla \hat{J}(u^0)\| \cdot \text{Id}_m. \tag{4.6}
\]
condition. In [16] it is explained that for a control $u$ which is sufficiently close to the stationary control $u^*$ the following holds:

$$(u - u(1))_i = \begin{cases} 
\nabla \hat{J}^N(u)_i, & i \in \mathcal{I}(u^*) \\
(u - u^*)_i = 0, & i \in \mathcal{A}(u^*). 
\end{cases}$$

Therefore, the difference $\|u - u(1)\|_{\mathcal{R}_m}$ is an indicator for the gradient which should be zero at $u^*$ in the unconstrained case.

We will follow the proposal of [16] to combine a desired absolute and a desired relative reduction of $\|u^0 - u^0(1)\|_{\mathcal{R}_m} =: r_0$ with $u^0$ denoting the initial control. With the definition of an absolute and a relative error tolerance $\varepsilon_{\text{abs}}$ and $\varepsilon_{\text{rel}}$ we demand that

$$\|u - u(1)\|_{\mathcal{R}_m} \leq \varepsilon_{\text{abs}} + \varepsilon_{\text{rel}} \cdot r_0 \quad (4.7)$$

holds for a computed control which will then terminate the iteration. Further, it is reasonable to establish a maximal iteration number $\max_{\text{iter}} \in \mathbb{N}$ to force an abort of the algorithm in case the optimality criterion (4.7) fails.

### 4.4 Application of the $\vartheta$-method to the optimal control problem

The algorithms presented in Section 4.1 and 4.2 can be carried out without adjustment to work on the concrete problem ($P^N_\mu$). It remains to establish a way to numerically solve the semi-discretized state equation (3.14). We will use the $\vartheta$-method which was presented in Chapter 2. To this end, we introduce an equidistant time grid with the grid size $\Delta t$ and the grid points $\{t_0 = t_0, t_1, \ldots, t_N = t_f\}$. The approximated states are denoted with $\tilde{y}_N^k \approx \tilde{y}_N(t_k)$. The application of the $\vartheta$-method given in (2.3) to the semi-discretized state equation for a given $u \in U_{\text{ad}}$

$$\frac{d}{dt} \tilde{y}_N(t) = \tilde{A}^N(t, u; \mu)\tilde{y}_N(t) + \delta^N g(t), \quad \tilde{y}_N(t_0) = P^N(y_0 \circ h)$$

leads to the following equation

$$\tilde{y}_{k+1}^N = \tilde{y}_k^N + \Delta t \left[ (1 - \vartheta)(\tilde{A}^N(t_k, u; \mu)\tilde{y}_k^N + \delta^N g(t_k)) + \vartheta (\tilde{A}^N(t_{k+1}, u; \mu)\tilde{y}_{k+1}^N + \delta^N g(t_{k+1})) \right]$$

42
4.4 Application of the $\vartheta$-method to the optimal control problem

for $\vartheta \in [0, 1]$. With the definitions

\begin{align*}
M^{\text{exp}}(t_k) &:= \text{Id}_{N+1} + \Delta t (1 - \vartheta) \tilde{A}^N(t_k, u; \mu), \\
M^{\text{imp}}(t_{k+1}) &:= \text{Id}_{N+1} - \Delta t \vartheta \tilde{A}^N(t_{k+1}, u; \mu), \\
G(t_k, t_{k+1}) &:= \Delta t \delta^N ((1 - \vartheta) g(t_k) + \vartheta g(t_{k+1}))
\end{align*}

the linear system

\begin{equation*}
M^{\text{imp}}(t_{k+1}) \tilde{y}_{k+1}^N = M^{\text{exp}}(t_k) \tilde{y}_k^N + G(t_k, t_{k+1})
\end{equation*}

is obtained. Its solution determines the value $\tilde{y}_{k+1}^N$. If $\vartheta = 0$ is chosen, then $M^{\text{imp}} \equiv \text{Id}_{N+1}$ and the system is fully explicit. For $\vartheta = 1$ we obtain a fully implicit system.
In Chapter 4 the open loop problem was discussed. An intrinsic property of this procedure is the missing possibility to react to external perturbations. Therefore a second approach is implemented, the model predictive control (MPC). Its main ideas are found for example in [12] or [11].

The act of driving a car is often used as a demonstrative example for explaining MPC. A driver on a twisting road is not able to foresee every event possibly emerging on his route. So he analyzes the part of the road which is visible to him and adjusts his driving behaviour to his perception. Some meters later he will correct his handling as now a new part of the road becomes visible to him. And so he reacts every few seconds to his changing surroundings.

Now we like to transfer this concept to the context of optimal control. The idea is to optimize the control \( u \) not on the full time interval \([t_0, t_f]\) but on various smaller time intervals \( I_i \subset [t_0, t_f] \). These time intervals shall be chosen such that they start always just before the next injection day. The reason to let the interval begin before the injection day and not directly on the injection day lies in the differential equation. Changing the control at time \( t \) theoretically leads to an immediate alternation of the state at time \( t \). So assume that a value for \( y(t^*_i) \) is found by optimizing the objective functional for a previous interval \( I_{i-1} \) and saved as a new initial condition for the optimization on \( I_i \). The control \( u_i \) now might be modified due to optimization but it cannot influence the already fixed value of \( y(t^*_i) \). Therefore one control is applied and then the interval is shifted to start just before the next injection day.

### 5.1 Application of MPC to the optimal control problem

The set of the injection days is called \( M_{\text{inj}} := \{t^*_1, \ldots, t^*_m\} \). Establish a suitable time distance

\[
\Delta t_M \geq \max_{i=2,\ldots,m} (t^*_i - t^*_{i-1})
\]
for the size of one prediction interval. By choosing \( \Delta t_M \) this way we ensure that the complete interval \((t_0, t_f)\) can be covered by prediction horizons. Independent of the value \( \Delta t_M \) the optimization problem has to be solved \( m \) times to receive an optimal control value \( u_i \) for each injection day \( t_i^*, i \in \{1, \ldots, m\} \).

Let \( \Delta t \) be a grid size for the time axis such that \( \Delta t_M = M \Delta t \) with \( M \in \mathbb{N} \). The interval \([t_0, t_f]\) is then subdivided into a grid with \( N_t := (t_f - t_0)/\Delta t + 1 \) grid points \( \{t_0 = t_0, t_1, t_2, \ldots, t_f = t_N\} \). For the \( i \)-th step, \( i = 1, \ldots, m \), the considered interval is \( I_i := [t_i^* - \Delta t, t_i^* - \Delta t + \Delta t_M] \). Determine \( M_i := M_{\text{inj}} \cap I_i \) which contains all injection days which lie in the \( i \)-th interval and denote these days with \( M_i = \{t_i^*, \ldots, t_{i+i}^*\} \). From the previous steps the values for the controls \( u_k \) for \( k \in \{1, \ldots, i - 1\} \) are already known. The state equation transfers to

\[
\begin{align*}
  y_i'(t, x) + y_i''(t, x) &= (\beta - \alpha_i(t, \hat{u}_i; \mu)) y_i'(t, x) \quad \text{in} \ I_i \times \Omega \ 	ext{a.e.}, \\
  y_i'(t, x) &= g(t) \quad \text{in} \ I_i \ 	ext{a.e.}, \\
  y_i(t_i^* - \Delta t, x) &= y_i^{-1}(t_i^* - \Delta t, x) \quad \text{in} \ \Omega \ 	ext{a.e.} 
\end{align*}
\]

with \( y_i^{-1}(t_i^* - \Delta t, x), x \in \Omega \), the optimal values from the last iteration. The control \( \hat{u}_i \) denotes \( \hat{u}_i = (u_i, \ldots, u_{i+i}) \). The only function which actually depends on the control \( \hat{u}_i \) is

\[
\alpha_i(t, \hat{u}_i; \mu) := \frac{\mu_1}{1 + \exp(\mu_2/\theta \sum_{k=1}^{i+i} u_k \chi_k(t) - \mu_3)},
\]

and \( u_1, \ldots, u_{i-1} \) already fixed by previous iterations. Note that all previous controls, which are not referring to days within \( I_i \), are still included in \( \alpha_i \) since the formerly injected EPO amount still influences the complete amount of EPO at a time \( t \in I_i \).

A new objective functional is needed:

\[
J(i, y_i, \hat{u}_i) = \frac{\gamma_1}{2} \int_{I_i} \left( \int_{\Omega} y_i'(t, x) \, dx - y_d(t) \right)^2 \, dt + \frac{\sigma_i}{2} \sum_{k=1}^{i+i} |u_k - u^0_k|^2 \\
+ \frac{\gamma_2}{2} \left( \int_{I_i} y_i(t_i^* - \Delta t + \Delta t_M, x) \, dx - y_d(t_i^* - \Delta t + \Delta t_M) \right)^2
\]

with parameters \( \gamma_1, \gamma_2, \sigma_1, \ldots, \sigma_m \in \mathbb{R}_{\geq 0} \). The last term in (5.2) represents stabilizing terminal constraints. It is a tool which might guarantee that the space
5.1 Application of MPC to the optimal control problem

The further approach runs parallel to Section 3.1-3.2. The differential equation (5.1) is transformed into a Cauchy problem with a normalized maturity attribute. Subsequently a numerical discretization using the Legendre polynomials is carried out yielding a $N + 1$-dimensional ordinary differential equation.

The semi-discretized and normed solution of the $i$-th problem shall now be called $(\tilde{y}^i)^N =: y^i \in Y^N$ to obtain better readability as well as we denote the operator $(\tilde{A}^i)^N(t, u; \mu) =: A^i(t, u; \mu)$.

The pseudo-algorithm which has to be followed for implementation of the MPC method is:

\begin{algorithm}[H]
\caption{Nonlinear model predictive control (NMPC)}
\begin{algorithmic}[1]
\Require initial condition $y^i(t^*_i - \Delta t) = y^{i-1}(t^*_i - \Delta t)$, controls $u_1, \ldots, u_{i-1}$, 
time grid with grid size $\Delta t$ on $[t_0, t_f]$, pre-allocated solution matrix $Y \in \mathbb{R}^{(N+1) \times (Nt+1)}$
\ForAll{$i = 1, \ldots, m$}
\State Define new time interval $I_i := [t^*_i - \Delta t, t^*_i - \Delta t + \Delta t_M]$ \label{line:1}
\State Compute $M_i := M_{inj} \cap I_i = \{t^*_i, \ldots, t^*_{i+1}\}$ \label{line:2}
\State Solve optimal control problem (5.6) on $I_i$ with initial condition $y^{i-1}(t^*_i - \Delta t)$ \label{line:3}
for $u_i, \ldots, u_{i+1}$
\State Compute the indices $a_i = t^*_i - t^{-} / \Delta t , b_i = (t^*_{i+1} - t^{-}) / \Delta t$ \label{line:4}
\State Save all state values belonging to $\{t^*_i - \Delta t, \ldots, t^*_{i+1} - \Delta t\}$ in the solution matrix $Y(:, a_i : b_i) = y^i(:, 1 : b_i - a_i + 1)$ \label{line:5}
\State Save $y^{i+1}(t^*_{i+1} - \Delta t) := y^i(t^*_{i+1} - \Delta t)$ as new initial condition \label{line:6}
\State Save $u_i$ \label{line:7}
\State Shift the interval to $[t^*_{i+1} - \Delta t, t^*_{i+1} - \Delta t + \Delta t_M]$ \label{line:8}
\EndFor
\end{algorithmic}
\end{algorithm}

The computation of the used functions is carried out in the following: The state is expressed as a sum over the basis functions $e_0, \ldots, e_N$ with

$$y^i(t) = \sum_{j=0}^{N} \pi_j(y^i(t))e_j = \begin{pmatrix} \pi_0(y^i(t)) \\ \vdots \\ \pi_N(y^i(t)) \end{pmatrix} = \begin{pmatrix} y^i_0(t) \\ \vdots \\ y^i_N(t) \end{pmatrix}$$

and $\pi_j(y^i(t))$ being the projection of $y^i(t)$ on the $j$-th basis vector. The matrix $D := \text{diag}(\frac{1}{2j+1}, j = 0, \ldots, N)$ is used to obtain an expression for the inner product in $L^2_\omega$. It is given by $\langle \varphi, \psi \rangle_{L^2_\omega} = \varphi^T D \psi$ for $\varphi, \psi \in X^N$. The semi-discrete version
of the $i$-th Cauchy problem is
\[
\frac{d}{dt} y^i(t) = A^i(t, u; \mu) y^i(t) + \delta^N g(t), \quad t \in I_i, \tag{5.3}
\]
\[
y^i(t^i_i - \Delta t) = y^{i-1}(t^i_i - \Delta t)
\]
with
\[
A^i(t, u; \mu) \tilde{\phi} = -\frac{1}{\omega} \tilde{\phi}' + P^N(\kappa^i(t, \bar{u}_i; \mu) \tilde{\phi}) - \delta^N \tilde{\phi}(0), \quad \tilde{\phi} \in X^N.
\]

Compared to the distribution of $A^N(t, u; \mu)$ in equation (3.15) the matrices are exactly the same, except $A_k(t, u; \mu)$ which is restricted on the prediction horizon $I_i$.

The semi-discrete objective functional which is applied is
\[
J^N(i, y^i, \bar{u}_i) = \frac{\gamma_1}{2} \int_{I_i} \left| (y^i(t))^T \tilde{D} e_0 - y_d(t) \right|^2 dt + \sum_{k=i}^{i+i_i} \sigma_k \left| u_k - u^0_k \right|^2 + \frac{\gamma_2}{2} \left| (y^i(t^i_i - \Delta t + \Delta t_M))^T \tilde{D} e_0 - y_d(t^i_i - \Delta t + \Delta t_M) \right|^2. \tag{5.4}
\]

The matrix $\tilde{D} = \omega^{1/2} D$ is the same as in Subsection 3.2.1. The new summand in (5.4) changes the end time condition of the adjoint equation. We receive the following semi-discrete adjoint equation:
\[
\frac{d}{dt} p^i(t) = -A^i(t, u; \mu)^T p^i(t) \\
+ \gamma_1 \omega^{1/2} \left( y^i(t)^T \tilde{D} e_0 - y_d(t) \right) e_0 \quad \text{in } I_i \text{ a.e.,}
\]
\[
p^i(t^i_i - \Delta t + \Delta t_M) = -\gamma_2 \left( y^i(t^i_i - \Delta t + \Delta t_M)^T \tilde{D} e_0 - y_d(t^i_i - \Delta t + \Delta t_M) \right) e_0. \tag{5.5}
\]

The reduced gradient is needed to compute a descent direction for the optimization problem:
\[
(\nabla \bar{u}_i \bar{J}^N(i, \bar{u}_i; \mu))_\nu = \sigma_\nu ((\bar{u}_i)_\nu - u^0_\nu) + \int_{I_i} (\nabla \bar{u}_i \alpha^i(t, \bar{u}_i; \mu))_\nu (y^i(t))^T D p^i(t) \, dt
\]
for $\nu \in \{i, \ldots, i+i_i\}$. This representation requires the computation of $\nabla \bar{u}_i \alpha^i(t, \bar{u}_i; \mu)$. As before, we determine the gradient to depend on the coefficient functions
\[
\bar{\alpha}^i(t, \bar{u}_i; \mu) = \frac{\mu_1 \mu_2 \exp(\mu_2 / TBV \sum_{k=1}^{i+i_i} \bar{u}_k \chi_k(t) - \mu_3)}{(1 + \exp(\mu_2 / TBV \sum_{k=1}^{i+i_i} \bar{u}_k \chi_k(t) - \mu_3))^2}
\]
5.1 Application of MPC to the optimal control problem

by

$$\nabla u_i \alpha^i(t, \hat{u}_i; \mu) = \bar{\alpha}^i(t, \hat{u}_i; \mu) \left( \chi_i(t), \ldots, \chi_{i+l}(t) \right) \in \mathbb{R}^u.$$ 

For a given $\mu \in D_\mu$ the $i$-th optimal control problem is:

$$\min J^N(i, y^i, \hat{u}_i) \quad \text{subject to (5.3) fulfilled, } \hat{u}_i \in [0, 1]^u, \quad y^i \in Y^N.$$ \hspace{1cm} (5.6)

Line 4 in Algorithm 5.1 requires the solution of an optimization problem which can be carried out using the same methods as discussed before, for example the projected gradient method or the projected BFGS method.

**MPC Feedback law**

Another name for the MPC strategy is closed loop method. This term results from the manner in which the computed optimal control is received. Assume that (5.6) possesses a unique global solution. Further, assume that the solving of the optimization problem (5.6) for each prediction horizon is terminated with an optimal solution $u_i^*$ which fulfills the optimality criteria. This means that

$$u_i^* = \arg \min_{\hat{u}_i} J^N(i, y^i, \hat{u}_i)$$

is uniquely determined by the initial value $y^i(t_i^* - \Delta t)$ and the solution method. We can see that the control can be understood as a function of the state itself. This yields the term ”closed loop” since the output values of the state are taken as input values for the next problem in the shape of initial values. In Definition 5.1 the feedback law is formulated:

**Definition 5.1 (Feedback law).** The MPC feedback law is defined as a mapping $\mu_M(y^i(t_i^* - \Delta t)) := u_i^*(1)$ with $u_i^* \in \mathbb{R}^u$ being the minimizing control for problem (5.6).

Here, $u_i^*(1)$ denotes the first component of $u_i^*$. The state equation is uniquely solvable for each prediction horizon. Thus, the trajectory of state values which is determined by the optimal control sequence can be indexed with the feedback law, $y_{\mu_M}(i) := y^i(u_i^*(1))$. The state $y_{\mu_M}(i)$ refers to the interval between the injection days $t_i^*$ and $t_{i+1}^*$. A corresponding optimal value function

$$V_M(y_0^i) := \inf \{ J^N(i, y^i, \hat{u}_i) \mid \hat{u}_i \in [0, 1]^u \}, \quad i = 1, \ldots, m,$$ \hspace{1cm} (5.7)
5 Model predictive control

for a prediction horizon of size \( M \Delta t \) can be defined for each initial value \( y_0^i \) of the state equation.

### 5.2 Suboptimal model predictive control

Model predictive control is frequently used to obtain a good approximation to the optimal solution as well as preserving the ability to react to unforeseen circumstances. The disadvantage of this approach is the rather long computing time since for every prediction horizon the solutions are iterated until the optimality condition is reached. For applications which have to provide a solution of the optimal control problem in real-time it might be necessary to settle for suboptimal but stable solutions in order to reduce the computational effort. Problem (5.6) is therefore only approximately solved.

One way to do so is incomplete optimization. The main ideas in this section follow the approach of [10]. The notation will be hold close to the notation used in this article.

Let us recall the objective functional used for MPC introduced in equation (5.4):

\[
J^N(i, y^i, \hat{u}_i) = \frac{\gamma_1}{2} \int_{I_i} \left( y^i(t) \right)^T \hat{D} e_0 - y_d(t) \right|^2 dt + \sum_{k=i}^{i+i_i} \frac{\sigma_k}{2} \left| u_k - u_k^\circ \right|^2 + \frac{\gamma_2}{2} \left( y^i(t_i^* - \Delta t + \Delta t_M) \right)^T \hat{D} e_0 - y_d(t_i^* - \Delta t + \Delta t_M) \right|^2.
\]

For basic treatment the last summand is further neglected. This will be justified later by numerical tests in Chapter 6. In addition it turns out that strictly positive values of \( \sigma \) yield rather low valued solutions \( \omega \int_0^1 \hat{y}^N(t) \, dx \) compared to the desired population \( y_d(t) \) due to the penalization of high EPO administration. In this thesis we will focus on the reachability of the desired population and therefore all \( \sigma_i \) will be set to zero. This leaves us with

\[
J^N(i, y^i, \hat{u}_i) = \frac{\gamma_1}{2} \int_{I_i} \left( y^i(t) \right)^T \hat{D} e_0 - y_d(t) \right|^2 dt.
\]  

For simplicity reasons in notation, we will regard the number of basis functions \( N + 1 \) as fixed and omit the index \( N \) in the space-discretized functions. In the same manner the control \( \hat{u}_i \) is from now on simply denoted with \( u \) as well as the state \( y^i := y \) when considering the \( i \)-th problem. The initial condition to the state
for the \(i\)-th problem which is defined by \(y^i(t^*_i - \Delta t) = y^{i-1}(t^*_i - \Delta t)\) is denoted with \(y_0\).

Furthermore, we will use the trapezoidal rule to carry out the integration on the equidistant time grid \(\{t^*_1 - \Delta t, t^*_1, \ldots, t^*_i - \Delta t + \Delta t_M\}\) with grid space \(\Delta t\).

Also, we will index the state \(y\) with the control \(u\) it belongs to, in symbols \(y_u\), to denote the dependency. In fact, the reduced objective functional which is considered only depends on the control and the initial condition \(y_0\). The iteration number \(i\) is determined by the first input in the objective functional (5.8). For first examinations we consider the control \(u\) as a function and will address the discrete nature of the control at the end of this section. By identifying the control \(u\) with a piecewise constant function attaining the components of \(u\) in the corresponding time intervals the theoretical insights below can be applied.

Let now the grid size \(\Delta t\) be fixed and the size of the prediction horizon obtained as a multiple of \(\Delta t\), i.e. \(\Delta t_M = M\Delta t, M \in \mathbb{N}\). With the new notation introduced above we derive:

\[
J_M(i, y_0, u) = \sum_{k=0}^{M} \ell(i, y_u(k), u(k))
\]

The function \(\ell\) is obtained by carrying out the trapezoidal rule on (5.8)

\[
\frac{\gamma_1}{2} \int \left| y_u(t)^T \hat{D} e_0 - y_d(t) \right|^2 dt \approx \frac{\gamma_1}{2} \Delta t \left[ \frac{1}{2} \left( y_u^T(0) \hat{D} e_0 - y_d(0) \right) 
+ \frac{1}{2} \left( y_u^T(M) \hat{D} e_0 - y_d(M) \right) 
+ \sum_{k=1}^{M-1} \left( y_u^T(k) \hat{D} e_0 - y_d(k) \right) \right] 
=: \sum_{k=0}^{M} \ell(i, y_u(k), u(k))
\]

where in the \(i\)-th iteration \(k\) stands for the discrete point in time \((t^*_i - \Delta t) + k\Delta t = \hat{k}\).

We now focus on the objective functional \(J_M\) which approximates the continuous functional \(J^N_i(y^i_u, \hat{u}_i)\) by the trapezoidal integration rule. The theoretical goal of minimizing the cost functional for an infinite horizon can be formulated as

\[
\inf_{u \in U_{ad}} J_\infty(i, y_0, u) = \inf_{u \in U_{ad}} \sum_{k=0}^{\infty} \ell(i, y_u(k), u(k))
\]
The state equation is solved with an ODE solver. In our test samples we will use a one-step solver which can be expressed by a function rule for an explicit solver with

\[ y(n + 1) = f(y(n), u(n)) \]

or for an implicit solver with

\[ y(n + 1) = f(y(n + 1), u(n + 1)) \]

or it can be an interpolation between both.

**Suboptimal feedback law**

Assume again that (5.6) possesses a unique global solution. The control which is determined by incomplete optimization does not necessarily fulfill the optimality conditions but is generally suboptimal. Nevertheless, it shall fulfill a certain stability requirement which will be tested numerically rather than theoretically proven. For that purpose the suboptimal feedback law for a prediction horizon of size \( M\Delta t \) is introduced.

**Definition 5.2 (Suboptimal feedback law).** The suboptimal feedback law \( \tilde{\mu}_M \) is the numerical solution of the incomplete optimization algorithm and maps the initial condition onto the first suboptimal control component \( \tilde{\mu}_M(y_0) := \tilde{\mu}(0) \).

With the suboptimal feedback law \( \tilde{\mu}_M \) the corresponding trajectory of state values on the respective time interval will be denoted by \( y_{\tilde{\mu}_M} \). It is necessary to differentiate between already applied suboptimal controls and the control sequence which is delivered by incomplete optimization for one prediction horizon. We will index the iteration number \( i \) in the control \( \tilde{\mu}_i \) and denote with \( \tilde{\mu}_i(k), k = 0, \ldots, M - 1 \), the values of the control which are found by incomplete optimization at time points \( (t_i^* - \Delta t) + k\Delta t \). The state values \( y_{\tilde{\mu}_i}(0), \ldots, y_{\tilde{\mu}_i}(M - 1) \) correspond to these control values. In contrast to this, \( y_{\tilde{\mu}_M} \) denotes the trajectory which is given by the suboptimal feedback law, i.e. the values which are obtained by applying the suboptimal control values one by one. The following equalities are implied by the definition of the feedback law and the used iterative solver of the state equation:

\[
\begin{align*}
\tilde{\mu}_M(y_{\tilde{\mu}_M}(i)) &= \tilde{\mu}_i(0) \quad & \text{(explicit)} \\
\tilde{\mu}_M(y_{\tilde{\mu}_M}(i + 1)) &= \tilde{\mu}_i(1) \quad & \text{(implicit)}
\end{align*}
\]
and
\[ y_{\bar{\mu}M}(i+1) = y_{\bar{u}_i}(1) = y_{\bar{u}_{i+1}}(0) \]

Following the definitions for complete optimization a suboptimal value function
\[ \tilde{V}_M(i) := J_M(i, y_{\bar{\mu}M}(i), \tilde{u}_i) = \sum_{k=0}^{M-1} \ell(i, y_{\bar{u}_i}(k), \tilde{u}_i(k)) \]
is defined. The suboptimality of \( \tilde{V} \) implies \( \tilde{V}_M(i) \geq V(y_{\bar{\mu}M}(i)) \) with the optimal value function \( V \). Remembering the primary goal of working with an infinite prediction horizon we define the optimal value function with infinite horizon according to (5.7) as
\[ V^{\mu}_\infty(y_0) := \sum_{k=0}^{\infty} \ell(y_{\mu M}(k), \mu_M(y_{\mu M}(k))) \] (5.9)
with \( \mu_M \) the optimal feedback law. Since it is not directly computable it is sufficient to calculate an upper bound.

The following theorem is taken directly from [10] where also a proof can be found:

**Theorem 5.3.** Consider a closed loop trajectory \( y_{\bar{\mu}M} \) and assume there exists \( \alpha_{\text{sub}} \in (0,1) \) such that for all \( i \in \mathbb{N}_0 \) the inequality
\[ \tilde{V}_M(i) \geq \tilde{V}_M(i+1) + \alpha_{\text{sub}} \ell(y_{\bar{\mu}M}(i), \mu_M(y_{\bar{\mu}M}(i))) \] (5.10)
holds. Then we obtain the estimate
\[ V^{\mu}_\infty(y_{\bar{\mu}M}(0)) \leq \frac{\tilde{V}_M(0)}{\alpha_{\text{sub}}} \] (5.11)

Theorem 5.3 is constructive in the way that the suboptimal control given by the suboptimal feedback law \( \tilde{\mu}_M \) is determined as the control which fulfills inequality (5.10) for a fixed \( \alpha_{\text{sub}} \). This means that, assuming a suboptimal control \( \tilde{u}_{i-1} \) is already given, all values \( \tilde{u}_{i-1}(k), k = 0, \ldots, M-1 \), are saved as well as the corresponding trajectory \( y_{\bar{u}_{i-1}} \). These values are needed to compute the value of the cost functional \( J_M(i-1, y_{\bar{\mu}M}(i-1), \tilde{u}_{i-1}) \). For solving the \( i \)-th problem an initial try \( \tilde{u}_{i}^{(0)}(k), k = 0, \ldots, M-1 \), is needed. Now the optimization algorithm is applied to \( \tilde{u}_{i}^{(0)} \), yielding iterates \( \tilde{u}_{i}^{(n)}, n = 1, 2, \ldots \) until the following holds:
\[ J_M(i, y_{\bar{\mu}M}(0), \tilde{u}_{i}^{(n)}) \leq J_M(i-1, y_{\bar{\mu}M}(i-1), \tilde{u}_{i-1}) - \alpha_{\text{sub}} \ell(i-1, y_{\bar{\mu}M}(i-1), \tilde{u}_{i-1}(0)) \] (5.12)
Then the suboptimal control is set to $\tilde{u}_i := \tilde{u}_i^{(n)}$. In general, the choice of $\alpha_{\text{sub}}$ is not an easy task.

So far we have not addressed the fact that the specific control we apply to our optimization problem is defined as the relative amount of EPO which is injected at several fixed injection days. In this case it is not a continuous control which can be adjusted at any point in time. But this is easily manageable by redefining the function $\ell$. It was originally specified by the trapezoidal integration rule to the sum of $\sum_{k=0}^{K} \ell(i, y_u(k), u(k))$ with $K$ being the time index of the next injection day. Transferring this, the function $\ell$ which is used for implementation matches the discretization of $\frac{\alpha_i}{2} \int_{t_i}^{t_{i+1}} |y_u(t)^T \tilde{D}e_0 - y_d(t)|^2 \, dt$ for the $i$-th problem. Denoting again the number of injection days in the interval $I_i$ with $\{t_{i*}, \ldots, t_{i*l}\}$ yields $i_l$ summands in the discrete objective functional $J_{i_l}(i, y_0, u) = \sum_{k=1}^{i_l} \ell(i, y_u(k), u(k))$. 
6 Numerical results

In the previous chapters we have prepared the necessary theoretical background about the numerical methods which can be used for solving the state equation and providing candidates for the optimal control. Now we will evaluate the arising problems and benefits. First, we will compare the different options in solving the state equation (3.14) by varying $\vartheta \in [0,1]$, each defining a specific $\vartheta$-method which was described in Chapter 2. Second, the optimization problem ($\hat{P}_\mu^N$) will be considered regarding the gradient method as well as the BFGS method with and without step size control. Third, the complete and incomplete MPC method will be examined and compared to the open loop process.

If not stated otherwise we will use the following configuration of variables throughout the tests: The control boundaries are set to $u_a = 0 \in \mathbb{R}^m$ (no injection) and $u_b = 1 \in \mathbb{R}^m$ (full amount of injection). The values appearing in the function $\kappa$ are $\beta = 0.6$, $TBV = 5000$, $\mu = (\mu_1, \mu_2, \mu_3) = (1.5, 2.5, 0.3)$, $EPO_{\text{max}} = 20000$. The half-life $T_{1/2} = 10/24$ is given as a fraction of a day. The maturity interval is given by $\Omega = (x, \bar{x}) = (0, 5)$. The boundary condition $g$ will be chosen as the constant function $g(t) \equiv S_0 e^{1.8}$ with $S_0 = 6 \cdot 10^6$ being the number of stem cells committing to the erythroid lineage. The initial condition is as well given by $y(0) \equiv S_0 e^{1.8}$. The time discretization rate is $\Delta t = 0.01$. We will focus on the set of injection days $M_{\text{inj}} = \{0, 2, 4, 7, 9, 11, \ldots \}$ which refers to the assumption that the patient is treated on Mondays, Wednesdays and Fridays.

6.1 Solving the state equation

In this thesis the 'first discretize, then optimize' approach was chosen for optimization. To recall the semi-discretized state equation (3.14) which was deduced in Section 3.2 we present it once more:

$$\frac{d}{dt} \hat{y}^N(t) = \hat{A}^N(t, u; \mu) \hat{y}^N(t) + \delta^N g(t), \quad \hat{y}^N(t_0) = P^N(y_0 \circ h).$$
Various methods for numerically solving the equation were discussed in Chapter 2. We will focus on the \(\vartheta\)-methods with \(\vartheta \in [0, 1]\) and set the length of the maturity domain as \(\omega = \overline{x} - \underline{x}\). As we are interested in the actual population of cells which is given by \(\omega \int_{0}^{1} \tilde{y}^{N}(t) \, d\xi\) and thus independent of their maturity \(\underline{x} \in [\underline{x}, \overline{x}]\) we will have a look at the time development of the population for a time interval \([t_0 = 0, t_f = 3]\). In Figure 6.1 we can see a comparison of different combinations of \(\vartheta\) and \(N\) for full injection amounts, i.e. \(u = 1\).

![Graphs showing population development](image)

Figure 6.1: The blue line displays the time development of the population of cells dependent on the discretization dimension \(N\) and the \(\vartheta\)-method. The instability of the explicit Euler method \((\vartheta = 0)\) can be seen in 6.1b whereas the implicit Euler method \((\vartheta = 1)\) proves to be stable, compare 6.1c,6.1d.

We witness an increasing instability with an increasing value \(N\) in combination with the explicit Euler method \((\vartheta = 0)\). In contrast to this the implicit Euler method \((\vartheta = 1)\) proves to be stable for high dimensions. The instability in the explicit Euler method can be reduced by minimizing \(\Delta t\). But this results in longer computation times as seen in Table 6.1 compared to a time of 0.193s for the implicit Euler method with \(\vartheta = 1, \Delta t = 0.01, N = 100\). We used a time discretiza-
6.1 Solving the state equation

<table>
<thead>
<tr>
<th>N</th>
<th>Δt</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.01</td>
<td>0.008s</td>
</tr>
<tr>
<td>60</td>
<td>0.001</td>
<td>0.221s</td>
</tr>
<tr>
<td>100</td>
<td>0.0005</td>
<td>1.169s</td>
</tr>
</tbody>
</table>

Table 6.1: Computation times for the explicit Euler method for a combination of a number of basis functions $N$ and time discretization $Δt$ which yield stability.

The behaviour of the change in population dependent on the injection amount can be seen if computed over several days, compare Figure 6.2. The development which can be observed was expected like this. If no EPO is injected as seen in the blue line in Figure 6.2 the population first drops and then becomes constant after approximately five days. The population which is received by injecting the highest possible amount of EPO on every injection day (yellow line) rises between two injection days and then drops a little when the EPO concentration becomes too low to have much influence on the proliferation rate $\alpha$. Over the weekend, i.e. days 4-7, the population suffers a bigger loss. The red line denotes a control which starts with full injection on day $t^*_0$ and ends with no injection on the last possible injection day $t^*_9$ and decreases constantly in between. It yields an interpolation in between the two other populations.
6 Numerical results

Figure 6.2: The upper graph shows the time development of the population of CFU-E cells dependent on the injection of EPO which is given in the lower graph. Full injection (yellow line) results in increasing population aside from a drop over the weekend, for example at days 4-7. No injection (blue line) yields a drop from the initial population to a constant population starting approximately at day 5. The red line shows a constantly decreasing injection amount and results in a population as well as in an EPO concentration which is an interpolation between the two other graphs.

6.2 Open loop problem

In this section we will turn to the optimization problem ($\mathbf{P}_\mu^N$) and apply the open loop method which was described in Chapter 4. The discrete objective functional

$$J^N(\tilde{y}^N, u) = \frac{\gamma_1}{2} \int_{t_0}^t \left| (\tilde{y}^N(t))^T \tilde{D}e_0 - y_d(t) \right|^2 \, dt + \sum_{i=1}^m \frac{\sigma_i}{2} |u_i - u_i^0|^2$$

(6.1)

which was first introduced in (3.16) shall be minimized. It contains the parameters $\gamma_1 > 0$ and $\sigma_1, \ldots, \sigma_m \geq 0$. First, we will choose $\gamma_1 = \int_{t_0}^t y_d(t) \, dt$ and $\sigma_i = 1$ for $i = 1, \ldots, m$ to have the two terms on the right hand side of (6.1) in approximately the same range. The injection days are given by $\{0, 2, 4, 7, 9, 11, \ldots\}$ which refers to the assumption that erythropoietin is injected on Mondays, Wednesdays and
Fridays. The nominal control \( u^o \) is chosen to be \( u^o_i = 0, i = 1, \ldots, m \), to favor a low injection amount or in other words low treatment expenses.

### 6.2.1 Choice of parameters \( \gamma_1 \) and \( \sigma \)

Setting \( t_f = 1.9 \) leads to a one-dimensional reduced optimization problem. There is only one injection day in \([0, 1.9]\) and so only one control parameter \( u \in \mathbb{R} \). To evaluate if the choices for \( \gamma_1 \) and \( \sigma \) made above are justified we apply the gradient method to the minimization problem. The reduced functional \( \hat{J}^N \) only depends on the control and is therefore representable as a graph. The negative reduced gradient

\[
(\nabla \hat{J}^N(u; \mu))_i = \sigma_i (u_i - u^o_i) + \int_{t_o}^{t_f} (\nabla_u \alpha(t, u; \mu))(\tilde{y}^N(t)) \cdot (\hat{\rho}^N(t))dt, \quad i = 1, \ldots, m,
\]

which is used to obtain a descent direction is one-dimensional as well. In Figures 6.3a, 6.3b we present two graphs to illustrate the action of \( \sigma = 1 \) on the optimization problem. Furthermore, the effects of \( \sigma = 1 \) are investigated for a seven-dimensional problem in Figures 6.4a, 6.4b which results by choosing \([t_o, t_f] = [0, 15]\). The step size for each optimization step was obtained by Armijo with a maximal iteration number of \( \max_{\text{armijo}} = 10 \). The maximal iteration number for the whole optimization method \( \max_{\text{iter}} = 50 \) and the main termination criterion \( \|u - u(1)\| \leq \varepsilon_{\text{abs}} + \varepsilon_{\text{rel}} \) were explained in Section 4.3. In general, the range of the population is approximately in \([1 \cdot 10^8, 4.7 \cdot 10^8]\). So, a desired population \( y_d \) is only reasonable to lie in the above interval since other populations are not being reached by any choice of control parameters. What we now expect is that a high desired population is not likely to be reached because the objective functional penalizes a high control value by the second term in (6.1) whereas a low desired population is more likely to be reached for fixed \( \sigma = 1 \). The initial control is now chosen as \( u = 1 \). In addition to the one-dimensional cost functional we compute an optimal population and optimal control for a fifteen day treatment but using the same values for \( \sigma \) and \( y_d \). Recall that the EPO function which was introduced in (1.3) was given by

\[
E(t) = \frac{1}{TBV} \sum_{j=1}^{m} u_j \chi_j(t)
\]

and ranges in the interval \([0, 4] \subset \mathbb{R}\) for the given parameters and possible control values.
Figure 6.3: The two graphs show an optimization for the desired populations $y_d = 1.5 \cdot 10^8$ (6.3a) and for $y_d = 3.5 \cdot 10^8$ (6.4a), always with $\sigma = 1$ and $[t_o, t_f] = [0, 1.9]$. The objective functional (blue line) for the one-dimensional problem is given, the objective function values of the iterated control values (red stars) and the optimal control which is found by the gradient algorithm (green circle).

$\sigma = 1$ : In Figures 6.3 and 6.4 we see several graphs illustrating the impact of the value $\sigma = 1$ on the optimization process.

In Figure 6.3 the objective functionals for $t_f = 1.9$ are plotted once for $y_d = 1.5 \cdot 10^8$ (6.3a) and once for $y_d = 3.5 \cdot 10^8$ (6.3b). The minimum is optically found for very low control values. This can be explained by the high costs in the objective functional $J_N$ for higher control values.

In Figures 6.4a and 6.4b two optimizations have been carried out for fifteen days and thus a seven-dimensional problem with $u \in \mathbb{R}^7$. The objective functional can not be displayed graphically but the gradient method yields optimal control values which also range close to zero (note the y-axis limits) as seen in the lower section. The maximal value for EPO is four as explained before. The populations resulting from these control values behave as expected, for $y_d = 1.5 \cdot 10^8$ the actual population (blue line) is closer to the desired population (red line) than for $y_d = 3.5 \cdot 10^8$. But in both cases the actual population is far below the desired population which might medically not be acceptable.
6.2 Open loop problem

Figure 6.4: The two graphs show an optimization for a desired population \( y_d = 1.5 \cdot 10^8 \) (6.4a) and for \( y_d = 3.5 \cdot 10^8 \) (6.4b), always with \( \sigma = 1 \) and \([t_0, t_f] = [0, 15] \). The desired population \( y_d \) is marked with a red line in the upper section and the actual population with a blue line. In the lower section the corresponding EPO-values are shown with a blue line.

\( \sigma = 0 \): In Figure 6.5 a value \( \sigma = 0 \) has been chosen for the objective functional given in equation (6.1). We find that very high control values are attained, several times even the maximal possible value \( u = 1 \). The population influenced by these control values fits the curve of the desired population much better but at the price of high medical treatment costs.

To sum up, the gradient method seems to find good minimal values of the objective functional for different parameters \( \sigma \). Considering the medical purpose of preserving a desired population of red blood cells, the value \( \sigma = 1 \) seems to underestimate the value of the patient’s health. As could be seen in Figure 6.4, this results in populations far below the desired population \( y_d \). During the further procedure we will focus on the ability of the different optimization algorithms to approximate the desired population and neglect the cost factor for the hormone EPO. This means, we will now set \( \sigma = 0 \) to obtain better visualization of the action of the optimal control on the actual population. The value \( \gamma_1 \) is kept fixed. Hence, the values of the objective functional remain comparable for different desired populations \( y_d \).
6 Numerical results

Figure 6.5: In the left graph (6.5a) a one-dimensional objective functional is drawn (blue line) for $\sigma = 0$ and a desired population $y_d = 3.5 \cdot 10^8$. The optimal control value which is found by the algorithm is marked with a green circle.

In the right graph (6.5b) the optimization was carried out for fifteen days and again $\sigma = 0$ and $y_d = 3.5 \cdot 10^8$. The desired population is given by the red line, the actual population by the blue line in the upper section and the corresponding EPO concentration by the blue line in the lower section.

6.2.2 Armijo step size strategy

In Chapter 4 two methods have been discussed for the computation of a descent direction: the projected gradient method and the projected BFGS method. To receive globalization, both methods will be combined with a step size control strategy. We have chosen the modified Armijo strategy which is explained in Algorithm 4.3. The parameter $\alpha$ appearing in the modified Armijo condition (4.3) is set to $\alpha = 10^{-4}$ and the backtracking parameter to $\beta = 0.5$. We will now consider the two-dimensional problem which arises if $t_f \in [2, 4)$. Setting $t_f = 3$ we can compare the results for a normed descent direction $d = -\nabla J^N(u)/\|\nabla J^N(u)\|$ and for $d = -\nabla J^N(u)$. We take now the desired population $y_d = 2.5 \cdot 10^8$ and start for different initial controls $u_0$ using the gradient method.

Testing different combinations we can conclude that it is reasonable to use the Armijo step size control as well as normalizing the descent direction. This combination leads to a convergence of the algorithm and usually takes the least number of iterations. In contrast, working without step size control mostly does not lead to convergence of the algorithm. This is easily comprehensible since the Armijo strategy forces a step size which results in minimizing the value of the objective functional at the next step. Without step size control the next control value might
6.2 Open loop problem

as well lead to an increasing value of the objective functional. An example with an initial control \( u_0 = (0, 0)^T \) is given below:

- without step size control

<table>
<thead>
<tr>
<th>Descent direction</th>
<th>( | \nabla \hat{J}^N(u^*) | )</th>
<th>( \hat{J}^N(u^*) )</th>
<th>Iterations</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d = -\nabla \hat{J}^N(u) )</td>
<td>1.10 \cdot 10^{-4}</td>
<td>1.04 \cdot 10^{-3}</td>
<td>80</td>
<td>3.2s</td>
</tr>
<tr>
<td>( d = -\frac{\nabla \hat{J}^N(u)}{| \nabla \hat{J}^N(u) |} )</td>
<td>2.07 \cdot 10^{-2}</td>
<td>9.42 \cdot 10^{-3}</td>
<td>100</td>
<td>4.0s</td>
</tr>
</tbody>
</table>

- with step size control

<table>
<thead>
<tr>
<th>Descent direction</th>
<th>( | \nabla \hat{J}^N(u^*) | )</th>
<th>( \hat{J}^N(u^*) )</th>
<th>Iterations</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d = -\nabla \hat{J}^N(u) )</td>
<td>1.10 \cdot 10^{-4}</td>
<td>1.04 \cdot 10^{-3}</td>
<td>80</td>
<td>4.5s</td>
</tr>
<tr>
<td>( d = -\frac{\nabla \hat{J}^N(u)}{| \nabla \hat{J}^N(u) |} )</td>
<td>6.89 \cdot 10^{-5}</td>
<td>1.04 \cdot 10^{-3}</td>
<td>12</td>
<td>1.8s</td>
</tr>
</tbody>
</table>

To accommodate these tests we will now choose a normed descent direction and apply the Armijo step size strategy in each iteration.

In the Armijo algorithm we have implemented a maximal number of iterations \( \max_{armijo} \). The choice of this maximal number is a crucial point since a high number might lead to quite long computation times whereas a small number might lead to no convergence and step sizes which do not fulfill the Armijo condition. As an example we have chosen a desired population \( y_d = 1.5 \cdot 10^8 \) and a time interval \( [t_0, t_f] = [0, 1.9] \). We compare different values for \( \max_{armijo} \) in combination with the gradient method and a maximal iteration number for the open loop algorithm of \( \max_{iter} = 100 \).

<table>
<thead>
<tr>
<th>( \max_{armijo} )</th>
<th>( | \nabla \hat{J}^N(u^*) | )</th>
<th>Iterations</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1.18 \cdot 10^{-3}</td>
<td>100</td>
<td>15.2s</td>
</tr>
<tr>
<td>15</td>
<td>8.97 \cdot 10^{-5}</td>
<td>29</td>
<td>5.5s</td>
</tr>
<tr>
<td>20</td>
<td>6.63 \cdot 10^{-4}</td>
<td>100</td>
<td>26.2s</td>
</tr>
</tbody>
</table>

We see that only for \( \max_{armijo} = 15 \) the open loop algorithm terminates with an optimal control. In the other two cases the sufficient decrease condition is not fulfilled, which means that the value for \( \alpha \) is too big. In the first case the iterates oscillate between the same values because the step size is too large. In contrast, in the third case the step sizes are in the range of \( 10^{-7} \) and therefore become too small. The iterates move too slow towards the optimal control so that the open
6 Numerical results

loop algorithm terminates before convergence is received. In this case convergence is obtained after 759 iterations and 196s.

As a good choice of $\text{max}_{\text{armijo}}$ will always depend on the particular situation we will choose $\text{max}_{\text{armijo}} = 10$ in the following which has in other cases proven to be a good choice and leads to comparatively small CPU times. The value $\alpha = 10^{-4}$ will stay fixed.

**Correlation between Armijo iterations and active indices**

The optimal control lies in $U_{ad}$ which is a compact subset of $\mathbb{R}^m$. For extreme values of desired populations $y_d$ ($y_d$ close to $1 \cdot 10^8$ or $4.7 \cdot 10^8$) it might occur that indices of the iterated controls become active, meaning that $u_i = (u_a)_i$, or $u_i = (u_b)_i$. We will now contemplate the connection between the modified Armijo step size strategy and active indices in the optimal control found by the gradient method. If one index $i$ of a current control $u$ is active and the same index in the direction $d_i$ is still leading outside $U_{ad}$ then $u(t, d)_i = u_i$ due to the projection into the admissible set. So, if two subsequent iterated controls possess the same active index it is not taken into account anymore to determine the step size. We will use the $\varepsilon$-active set to avoid computational inaccuracies. An index is $\varepsilon$-active if $u_i - (u_a)_i \leq \varepsilon$ or $(u_b)_i - u_i \leq \varepsilon$. We choose $\varepsilon = 0.01$ in the following.

We want to empirically investigate the correlation between the number of iterations the modified Armijo step size strategy takes per average and the number of $\varepsilon$-active indices in the optimal control found by the algorithm. As an example we choose $[t_0, t_f] = [0, 10]$ and an initial control $u_0 = u_b \in \mathbb{R}^5$. In Figure 6.6 we see the number of $\varepsilon$-active indices in the optimal control $u^*$ given by the red line dependent on the desired population $y_d$. The blue line denotes the average number of iterations the modified Armijo strategy needs to find a new step size. We find that for extreme values of desired populations, $y_d \leq 1.2 \cdot 10^8$, $y_d \geq 3.5 \cdot 10^8$, a great number of indices of the optimal control $u^*$ are $\varepsilon$-active. In these cases Armijo tends to a low number of iterations, in average approximately less than three. In contrast, having an optimal control with exclusively $\varepsilon$-inactive indices as given for $y_d \in [1.5, 2.5] \cdot 10^8$, the mean number of Armijo iterations tends to rise above four. If examined closer we find that after a certain number of iterations the algorithm fails to converge and thus the maximal number of iterations is needed. In these cases the gradient is already quite small and the graph of $\hat{J}^N$ seems to be flat. The peaks of the blue line for $y_d \geq 3.5 \cdot 10^8$ appear if no optimal control was found by the algorithm within the maximal number of iterations which was here set to 40. In these cases the number of $\varepsilon$-active indices of the optimal control might differ
6.2 Open loop problem

$$[t_0, t_f] = [0, 10], \quad u_0 = 1 \in \mathbb{R}^5$$

desired population (in $10^8$)

| desired population (in $10^8$) | mean number of Armijo iterations (max = 10) | number of active indices for optimal control $u^* \in \mathbb{R}^7$
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

![Figure 6.6: Shown is the dependency of the number of $\varepsilon$-active indices of the optimal control $u^*$ (red line) and the dependency of the average number of iterations of the modified Armijo step size strategy (blue line) for each optimization problem. The problem is determined by the desired population $y_d$, the time interval $[t_0, t_f]$ and the initial control $u_0$. The optimization was carried out by the gradient method.](image)

The optimization was carried out by the gradient method.

A little from the proposed red line. But assuming that the last control computed by the algorithm is already close to the optimal control there should not be many fluctuations.

### 6.2.3 BFGS method

The second method for finding a descent direction was the BFGS method discussed in Section 4.2. The update of the BFGS matrix as defined in equation (4.6) contains some freedom in connection with $\varepsilon$-active indices. The rows and columns of $B_k$ referring to $\varepsilon$-active indices can be replaced by the equivalent rows and columns of the identity matrix or additionally being multiplied by the norm of the reduced gradient. The latter case leads the BFGS method back to the gradient method for the $\varepsilon$-active indices with normed directions. But since we want to keep information from both the gradient and the curvature we will not use the normed directions. It makes sense to scale the BFGS matrix if there is a very large gradient component for an $\varepsilon$-active index whose influence in the search direction can be reduced by normalizing this component. But as we work with rather small
6 Numerical results

gradients there is no need to multiply the BFGS matrix with the gradient’s norm.

We will now investigate if an additional step size control in the BFGS method is desirable or if the combination of approximated Hessian $B_k$ and gradient $\nabla \hat{J}^N(u_k)$ already yields a good length of the direction vector $d_k$. Recall that $d_k$ is given by $B_kd_k = -\nabla \hat{J}^N(u_k)$. We present a simple example which shows the necessity of step size control. Take the configuration $[t_0, t_f] = [0, 3]$ which yields a two-dimensional problem. We apply the BFGS method once without and once with step size control to the problem.

![Figure 6.7: The contour lines for the objective functional for a desired population $y_d = 2.4 \cdot 10^6$ are given with their values presented in the color bar. The red asterisks show the iterates computed with the BFGS method and the green circle marks the optimal control found by the algorithm for an initial control $u_0 = (1, 1)^T$. In 6.7a the BFGS method was used without step size control and in 6.7b the Armijo step size strategy was applied.](image)

In Figure 6.7 the red line denotes the iterates given by the BFGS method for a maximal iteration number of $\text{max}_{\text{iter}} = 100$. The contour lines show the values for the objective functional and the green circle marks the optimal control found by the algorithm. In case the curvature condition fails we have chosen the update $\hat{B} = P_L$ as explained in Subsection 4.2.2. We see in 6.7b that the minimum is found after 12 iterations whereas in 6.7a the algorithm stopped as the maximal number of iterations is reached. The computed directions in this case mostly have norms close or greater than one. For a domain of size $U_{ad} = [0, 1]^2$ this means that the next control iterate will be close to the boundary. We have created an infinite loop.

The other possibility for handling the case, in which the curvature condition fails, was to skip the update and use the old BFGS matrix. In the test described above
this leads to convergence after 32 steps. But for a desired population \( y_d = 2.2 \cdot 10^8 \) we end up with an infinite loop once again. Applying step size control in this case leads to convergence after 16 iterations.

Of course, BFGS endowed with the Armijo strategy might lead to slower performances since in every Armijo iteration a new state and a function value has to be computed in order to test the new step size. However, we ensure globalization and with this alternation we avoid infinite loops as described above. We will in the following use the update \( B = \mathcal{P}_T \) in case the curvature condition fails.

### 6.2.4 Comparison between gradient method and BFGS method

We are now interested in the difference between the two optimization methods. In the following section we will compare the gradient method with the BFGS method with regard to speed, number of iterations and optimality of the solution.

Considering a time interval \([t_0, t_f] = [0, 15]\) yields a 7-dimensional optimal control problem. Given different values of desired populations \( y_d \) and a fixed initial control \( u_0 = 0 \) we will compare the results of GM and BFGS.

<table>
<thead>
<tr>
<th>( y_d )</th>
<th>Method</th>
<th>( | \mathcal{P}_{T^<em>}(u^</em>) \nabla \hat{J}^N(u^*) | )</th>
<th>Iterations</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 1.7 \cdot 10^8 )</td>
<td>GM</td>
<td>( 1.01 \cdot 10^{-4} )</td>
<td>10</td>
<td>7.9s</td>
</tr>
<tr>
<td></td>
<td>BFGS</td>
<td>( 1.21 \cdot 10^{-4} )</td>
<td>7</td>
<td>2.1s</td>
</tr>
<tr>
<td>( 2.7 \cdot 10^8 )</td>
<td>GM</td>
<td>( 9.79 \cdot 10^{-5} )</td>
<td>15</td>
<td>13.6s</td>
</tr>
<tr>
<td></td>
<td>BFGS</td>
<td>( 1.91 \cdot 10^{-4} )</td>
<td>8</td>
<td>2.4s</td>
</tr>
</tbody>
</table>

For higher values of the desired population we will choose \( u_0 = 1 \) as initial control to start comparatively close to the optimal control.

<table>
<thead>
<tr>
<th>( y_d )</th>
<th>Method</th>
<th>( | \mathcal{P}_{T^<em>}(u^</em>) \nabla \hat{J}^N(u^*) | )</th>
<th>Iterations</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 3.3 \cdot 10^8 )</td>
<td>GM</td>
<td>( 4.35 \cdot 10^{-5} )</td>
<td>9</td>
<td>4.3s</td>
</tr>
<tr>
<td></td>
<td>BFGS</td>
<td>( 5.65 \cdot 10^{-5} )</td>
<td>7</td>
<td>2.0s</td>
</tr>
</tbody>
</table>

Note here that \( \| \mathcal{P}_{T^*}(u^*) \nabla \hat{J}^N(u^*) \| \) computes only the components of the gradient for inactive indices. If active indices would be taken into account the gradient might be clearly greater than zero. Giving an example, for \( y_d = 3.3 \cdot 10^8 \) the norm of the gradient for all indices is ten times as big as the norm at the inactive components. As the active indices of the optimal control do not contribute to the search direction which is influenced by the gradient we are not interested in
their share of the gradient’s norm. In order to improve readability we will use the gradient’s norm $\|\nabla \hat{J}(u^*)\|$ and implicate that only inactive indices are computed as in $\|P_{\omega\epsilon}(u^*) \nabla \hat{J}(u^*)\|$. 

In all examples we see a clear superiority of the BFGS method regarding speed and number of iterations needed until the optimality criterion is fulfilled. However, the gradient method yields smaller gradients and thus controls which even better fulfill the optimality criterion for which we used values as $\epsilon_{\text{abs}} = \epsilon_{\text{rel}} = 10^{-4}$. If we set those values to zero we can explore which controls are found by the two methods after the same number of iterations, say $\text{max}_\text{iter} = 20$. The new situation is given in the following table.

<table>
<thead>
<tr>
<th>$y_d$</th>
<th>$|\nabla \hat{J}(u^*)|$</th>
<th>GM CPU time</th>
<th>$|\nabla \hat{J}(u^*)|$</th>
<th>BFGS CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1.7 \cdot 10^8$</td>
<td>$7.14 \cdot 10^{-5}$</td>
<td>19.6s</td>
<td>$6.52 \cdot 10^{-10}$</td>
<td>11.9s</td>
</tr>
<tr>
<td>$2.7 \cdot 10^8$</td>
<td>$8.54 \cdot 10^{-5}$</td>
<td>19.1s</td>
<td>$2.09 \cdot 10^{-5}$</td>
<td>9.1s</td>
</tr>
<tr>
<td>$3.3 \cdot 10^8$</td>
<td>$6.41 \cdot 10^{-4}$</td>
<td>14.3s</td>
<td>$2.59 \cdot 10^{-6}$</td>
<td>15.1s</td>
</tr>
</tbody>
</table>

The BFGS method tends to deliver smaller gradients than the gradient method after the same number of iterations. In these examples the CPU times either are similar or the BFGS method is faster.

**Choice of the initial control**

The choice of the initial control plays an important role for optimization. It is beneficial if the initial control is already close to the optimal control. Taking again the values from above but an initial control $u_0 = 0$ we receive the following table:

<table>
<thead>
<tr>
<th>$y_d$</th>
<th>Method</th>
<th>$|\nabla \hat{J}(u^*)|$</th>
<th>Iterations</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3.3 \cdot 10^8$</td>
<td>GM</td>
<td>$5.66 \cdot 10^{-5}$</td>
<td>9</td>
<td>3.6s</td>
</tr>
<tr>
<td></td>
<td>BFGS</td>
<td>$1.88 \cdot 10^{-4}$</td>
<td>26</td>
<td>8.4s</td>
</tr>
</tbody>
</table>

The BFGS method is clearly inferior in this example. But also for the gradient method we can compare results according to the initial control as done in the following table for $y_d = 2.7 \cdot 10^8$:

<table>
<thead>
<tr>
<th>Method</th>
<th>$u_0$</th>
<th>$|\nabla \hat{J}(u^*)|$</th>
<th>Iterations</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>GM</td>
<td>$0 \in \mathbb{R}^7$</td>
<td>$9.79 \cdot 10^{-5}$</td>
<td>15</td>
<td>13.6s</td>
</tr>
<tr>
<td></td>
<td>$1 \in \mathbb{R}^7$</td>
<td>$1.01 \cdot 10^{-4}$</td>
<td>29</td>
<td>28.4s</td>
</tr>
</tbody>
</table>
The term $1 \in \mathbb{R}^7$ denotes that every component $u_i$ is equal to $1$ for $i = 1, \ldots, 7$. We see that the CPU time as well as the number of iterations approximately doubles by choosing $u_0 = 1$ instead of $u_0 = 0$. Choosing a smaller desired population yields even greater differences. By $y_d = 1.1 \cdot 10^8$ we obtain for $u_0 = 1$ a CPU time as well as a number of iterations approximately seven times as much as for $u_0 = 0$. It is therefore reasonable to start at a control which appears to be close to an optimal control independent of the optimization method.

Another aspect about the initial control is the quality of the optimal control which is found by the algorithm. Comparing these for the one-dimensional problem with $y_d = 2.3 \cdot 10^8$ and with $[t_a, t_f] = [0, 3]$ a two-dimensional problem we end up with:

<table>
<thead>
<tr>
<th>Method</th>
<th>$u_0$</th>
<th>Iterations</th>
<th>Optimal control $u^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GM</td>
<td>$(0,0)^T \in \mathbb{R}^2$</td>
<td>13</td>
<td>$(0.4453, 0.2276)^T$</td>
</tr>
<tr>
<td></td>
<td>$(1,1)^T \in \mathbb{R}^2$</td>
<td>37</td>
<td>$(0.4437, 0.2310)^T$</td>
</tr>
</tbody>
</table>

Obviously the control which terminates the algorithm depends on the initial control. Of course, the termination depends on the requested accuracy in $\varepsilon_{\text{abs}}$ and $\varepsilon_{\text{rel}}$ which we have chosen as $10^{-4}$. If chosen even smaller the optimal controls $u^*$ move closer together. In Figure 6.8 a contour plot of the function values as well as of the iterates is given. Figure 6.8a shows the iterates for the initial control $u_0 = (0, 0)^T \in \mathbb{R}^2$ and Figure 6.8b for $u_0 = (1, 1)^T \in \mathbb{R}^2$.

![Figure 6.8](image_url)

Figure 6.8: The contour lines for the objective functional for $y_d = 2.3 \cdot 10^8$ are given with their values presented in the color bar. The red asterisks show the iterates computed with the gradient method and the green circle marks the optimal control found by the algorithm for an initial control $u_0 = (0, 0)^T \in \mathbb{R}^2$ in 6.8a and $u_0 = (1, 1)^T \in \mathbb{R}^2$ in 6.8b.
In Figure 6.9 the iterates computed by the BFGS method are presented. We see that the optimal control is found in less iterations compared to using the gradient method.

Figure 6.9: The contour lines for the objective functional for $y_d = 2.3 \cdot 10^8$ are given with their values presented in the color bar. The red asterisks show the iterates computed with the BFGS method and the green circle marks the optimal control found by the algorithm for an initial control $u_0 = (0, 0)^T \in \mathbb{R}^2$ in 6.9a and $u_0 = (1, 1)^T \in \mathbb{R}^2$ in 6.9b.

We can conclude that the BFGS method is faster than the gradient method. Using the same number of iterations, the BFGS method in general yields controls which are closer to the optimal control than the gradient method. It is more difficult to find a good initial control or the right way to update the matrices in case the curvature condition fails for the BFGS method. For iterates which are far from the optimal control we find that the BFGS method behaves similar to the gradient method. When approaching the minimum the approximation of the Hessian gets better and the method speeds up.

Here lies the difficulty in the BFGS method: its advantage in speed can be exploited best if we already knew the minimum and could therefore start close to it. For arbitrary initial points the method is not faster than the gradient method which is in general robust but rather slow.

### 6.2.5 Varying particular parameters

The problem $(P_N^\mu)$ involves certain parameters which have been left untouched so far. In this section we investigate the impact of changes in the half-life of EPO as well as changes in the injection days on the optimal control.
**Half-life**  The half-life chosen in the beginning was $T_{1/2} = 10/24$. Assuming that the half-life of the hormone could be altered we try a shorter half-life $0.5 \cdot T_{1/2}$ and a longer half-life $2 \cdot T_{1/2}$. In Figure 6.10 the effect of the change in $T_{1/2}$ is shown.

In Figure 6.10a at almost every injection day the maximal amount of EPO was used whereas in Figure 6.10b only about one quarter of the maximal amount was injected. We see that the population curve adapts better to the constant desired population for a longer half-life than for a shorter half-life of EPO. In the case of $2 \cdot T_{1/2}$, we even see that the mortality rate is too high to be compensated by the proliferation rate. After day 7 the population does not reach the desired population anymore. Altogether, we have a downward tendency. Concluding, if different hormones are available for the treatment, hormones with longer half-life are more desirable.

**Injection days**  Another possibility is the change of the set of injection days. Assume that an injection is possible everyday as seen in Figure 6.11a. Another possibility is that the patient’s treatment which was assumed to be on Monday, Wednesday and Friday is changed to Monday, Tuesday and Friday. This effect is seen in Figure 6.11b.

Optically one recognizes a good adaption of the computed to the desired population in both cases. In case of injections everyday the control attains very low values as expected and the population curve does not oscillate as much as in the
6 Numerical results

Figure 6.11: The upper part shows the population (blue line) and the desired population (red line). The lower part shows the corresponding EPO concentration computed by the gradient method. In Figure 6.11a injection of EPO is possible everyday and in Figure 6.11b the injections are Monday, Tuesday and Friday.

In the first case the value of the objective functional is only about one-tenth of the value of the objective functional in the second case. To compare quantitatively, we compute the value $\int_{t_0}^{t_f} \left| (\hat{y}^N(t))^T \tilde{D}e_0 - y_d(t) \right|^2 \, dt$ for the two cases: in the first case we get $8.23 \cdot 10^{14}$ and in the second case $1.24 \cdot 10^{16}$. In the case of injections everyday the population is therefore closer to the desired population. But in both cases the optimization method is able to react to the change of the control influence.

6.3 Model predictive control

In this section we turn towards the solutions of equation (5.6). In each iteration an open loop problem is solved whose properties have already been discussed in Section 6.2. We now evaluate the influence of the size of the prediction horizon $\Delta t_M$, a good choice for the value $\gamma_2$ and reactions of the algorithm to external perturbations. Recall the definition of the semi-discrete objective functional, presented in (5.4),

$$J^N(i, y^i, \hat{u}^i) = \frac{\gamma_1}{2} \int_{t_i}^{t_f} \left| (y^i(t))^T \tilde{D}e_0 - y_d(t) \right|^2 \, dt$$

$$+ \frac{\gamma_2}{2} \left| (y^i(t^*_i - \Delta t + \Delta t_M))^T \tilde{D}e_0 - y_d(t^*_i - \Delta t + \Delta t_M) \right|^2$$
which is minimized in every iteration. We have already chosen the value \( \sigma_i = 0 \) for \( i = 1, \ldots, m \) as decided in Section 6.2. The grid distance \( \Delta t = 0.01 \) stays fixed if not stated otherwise.

We use the gradient method to compute the optimal controls for each prediction horizon. In this case we have added another termination criterion for the while-loop. In some cases the iterates produced by GM combined with Armijo tend to oscillate between two values and end in an infinite loop if the maximal number of Armijo steps is badly chosen. Therefore, the loop is terminated if the condition

\[ \|u^k - u^{k+2}\| \leq 10^{-5} \]

is fulfilled. Here \( u^k \) denotes the \( k \)-th iterate in the while-loop.

### 6.3.1 Choice of parameter \( \gamma_2 \)

Consider now a prediction horizon of size \( \Delta t_M = 5 \). This means that at least two injection days lie in each prediction horizon if again \( M_{inj} = \{0, 2, 4, 7, 9, 11, \ldots\} \) is assumed. The value for \( \gamma_1 \) is chosen as \( \gamma_1 = (\int_{t_o}^{t_f} y_d(t) \, dt)^{-1} \). Take an interval \([t_o, t_f] = [0, 20] \) and a desired population \( y_d = 2 \cdot 10^8 \). Then \( \gamma_1 = 6.25 \cdot 10^{-20} \) and we will try exemplarily \( \gamma_2 \in \{10^{-30}, 1\} \).

![Figure 6.12: The upper part shows the population (blue line) and the desired population (red line) as a result of the closed loop for 20 days with a prediction horizon of \( \Delta t_M = 5 \). The lower part shows the corresponding EPO concentration computed by the gradient method. In Figure 6.12a \( \gamma_2 \) is chosen as 1 and in Figure 6.12b we choose \( \gamma_2 = 1 \cdot 10^{-30} \).](image-url)
In Figure 6.12a we see that $\gamma_2 = 1$ yields extremely bad values for the control and so also for the state. In contrary, for $\gamma_2 = 10^{-30}$ we see in Figure 6.12b a good adaption of the computed population to the desired population. To explore the differences in the computed controls we take a look at the first two prediction horizons and the computed optimal controls for this time interval in Figures 6.13 and 6.14.

We clearly see the differences for the computed optimal controls for the first (Figure 6.13) and second prediction horizon (Figure 6.14). In case of $\gamma_2 = 1$ the algorithm strongly favors a state such that the last value is close to the desired population. We see in Figures 6.13a and 6.14a that the last state value is approximately $2 \cdot 10^8$. For the lower value $\gamma_2 = 10^{-30}$ we see for example in Figure 6.14b that the optimization does not force the state to end close to the desired population.

Concluding, the results are better for very small values of $\gamma_2$. This can be explained by the discrete nature of the control. In each prediction horizon the values are found according to the goal to have a final value as close as possible to the desired population if $\gamma_2$ is big. Then the first control value is applied and the horizon is moved. But since the final point moves as well the situation has completely changed due to the discrete injection days. We see that the problem for the horizon itself is quite optimally solved in both cases but the combination of
6.3 Model predictive control

Figure 6.14: The upper part in each figure shows the population (blue line) and the desired population (red line). The lower part shows the corresponding EPO concentration computed by the gradient method. We see the optimal controls for the second prediction horizon and the corresponding state for $\gamma_2 = 1$ in 6.14a and for $\gamma_2 = 10^{-30}$ in 6.14b. Different horizons leads to an overall non-optimal solution for great values for $\gamma_2$. Therefore, we will from now on choose $\gamma_2 = 0$.

6.3.2 Size of prediction horizon

The choice for the size of the prediction horizon $\Delta t_M$ has been left open so far. We will now take an optimization problem with $y_d = 2.5 \cdot 10^8$ and $[t_0, t_f] = [0, 15]$ and perform the closed loop process with the BFGS method. The initial control is $u_0 = 0$. We will compute the reduced gradient for each optimal control obtained by the MPC algorithm.

<table>
<thead>
<tr>
<th>$\Delta t_M$</th>
<th>$|\nabla \hat{J}<em>N(u</em>{opt})|$</th>
<th>$\hat{J}<em>N(u</em>{opt})$</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>$6.28 \cdot 10^{-4}$</td>
<td>$3.94 \cdot 10^{-4}$</td>
<td>3.7s</td>
</tr>
<tr>
<td>10</td>
<td>$3.00 \cdot 10^{-4}$</td>
<td>$3.88 \cdot 10^{-4}$</td>
<td>8.5s</td>
</tr>
<tr>
<td>20</td>
<td>$2.19 \cdot 10^{-4}$</td>
<td>$3.76 \cdot 10^{-4}$</td>
<td>21.9s</td>
</tr>
<tr>
<td>30</td>
<td>$2.76 \cdot 10^{-4}$</td>
<td>$3.81 \cdot 10^{-4}$</td>
<td>38.8s</td>
</tr>
<tr>
<td>100</td>
<td>$8.37 \cdot 10^{-5}$</td>
<td>$3.71 \cdot 10^{-4}$</td>
<td>723.2s</td>
</tr>
</tbody>
</table>

For comparison, the open loop solver takes 2.3s for this fifteen day problem and ends up with $\|\nabla \hat{J}_N(u_{opt})\| = 7.68 \cdot 10^{-5}$ and $\hat{J}_N(u_{opt}) = 3.70 \cdot 10^{-4}$. We see that the closed loop performs better for larger prediction horizons $\Delta t_M$ but takes more computational effort. Note here, that outliers are possible as seen for $\Delta t_M = 20$. It...
is possible that a small prediction horizon is a better choice for particular problems but in general, a large horizon yields the best results. In application one has to weigh up accuracy against speed and choose the size of the prediction horizon according to the individual problem.

![Graph showing EPO concentrations over time with different prediction horizons](image)

Figure 6.15: The different EPO concentrations are outlined as optimal solutions for a fifteen day treatment and a desired population $y_d = 2.5 \cdot 10^8$. The purple line shows the solution of the open loop problem and the other three lines the solutions of the closed loop problem with prediction horizons $\Delta t_M \in \{5, 10, 20\}$. The optimization was carried out with the BFGS-Armijo algorithm.

In Figure 6.15 we see that the computed optimal controls are quite comparable in the situation described above.

### 6.3.3 Shift in desired population

Assume now a similar situation as before but take a desired population $y_d = 2 \cdot 10^8$. At day 7 the desired population $y_d$ rises by 30%. We want to compare the optimal controls computed by closed and open loop. For the open loop solver, the increase in the desired population is known in advance. For the closed loop solver there is no existing knowledge about the increase in the desired population until day 7. If we compare the open loop and the closed loop solver we see in Figure 6.16 that the control computed by the open loop solver yields a state which better adapts the desired population than the closed loop solver with a prediction horizon $\Delta t_M = 5$. 
Figure 6.16: The upper part in each figure shows the population (blue line) and the desired population (red line). The lower part shows the corresponding EPO concentration computed by the BFGS-Armijo method. 6.16a shows the solution for the closed loop and a prediction horizon $\Delta t_M = 5$ and 6.16b for the open loop problem. In 6.16b the dotted light blue line marks additionally the population and the EPO concentration for the case in which the desired population is not shifted.

In the following table the different values for the objective functional and its gradient are presented:

<table>
<thead>
<tr>
<th>Method</th>
<th>$|\nabla \hat{J}^N(u_{opt})|$</th>
<th>$\hat{J}^N(u_{opt})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>open loop</td>
<td>$1.85 \cdot 10^{-4}$</td>
<td>$5.82 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>closed loop</td>
<td>$8.84 \cdot 10^{-4}$</td>
<td>$6.23 \cdot 10^{-4}$</td>
</tr>
</tbody>
</table>

Let’s take a closer look at the computed EPO concentrations as given in Figure 6.17. We see that the EPO concentration yielded by the open loop solver is already higher at day 4 where the desired population has not yet increased. The three other concentrations from the closed loop solver have to compensate the missing concentration at day 7 and therefore we see higher values than for the open loop concentration. However, the differences are rather small.

We find that the open loop population is better adjusted to the desired population compared to the one computed by the closed loop. So assuming we already know each shift in advance, of course the open loop method is better suited to solve the optimal control problem. But since the desired population may alter in the
6 Numerical results

Figure 6.17: The different EPO concentrations are outlined as optimal solutions for a fifteen day treatment and a desired population $y_d = 2.0 \cdot 10^8$ with an increase by 30% at day 7. Shown is day 3-8. The purple line shows the solution of the open loop problem and the other three lines the solutions of the closed loop problem with prediction horizons $\Delta t_M \in \{5, 10, 20\}$. The optimization was carried out with the BFGS-Armijo algorithm.

Running process, the closed loop also yields extremely good values and is more adaptable to sudden changes in the problem specifications.

6.3.4 Skip injection day

In reality, it occurs that a patient does not keep the appointment for the next injection. Therefore, we want to investigate the effects if one injection day is omitted. Assume a treatment period of $[t_o, t_f] = [0, 20]$ and a desired population of $y_d = 2 \cdot 10^8$. We compare the computed optimal control sequence of the MPC method with prediction horizon $\Delta t_M = 10$ to the ideal open loop method. Assume that the patient does not come at day 9 which refers to the fifth injection. The open loop solution shows the ideal curve if the skipped day was known in advance.

In Figure 6.18 the effects of the skipped injection are presented. In the upper section of the figure the populations are given for the MPC method (dark blue line), the open loop method (dotted light blue line) and the desired population (red line). The lower section displays the EPO concentrations computed by the
Figure 6.18: The upper section shows the population computed by the MPC method (dark blue line) with a prediction horizon $\Delta t_M = 10$ for the case that injection day $t^*_5$ was omitted. The dotted light blue line gives the ideal population computed by the open loop method in case the skipped injection is known in advance. The red line marks the desired population $y_d = 2 \cdot 10^8$. In the lower section the corresponding EPO values are given for the MPC method (dark blue line) and the open loop (dotted light blue line). The skipped injection is marked with a red asterisk.

We see that at injection day $t^*_4$ the ideal dose (given by the dotted light blue line) is almost double than the dose computed by MPC (dark blue line). Therefore, the ideal population does not suffer such a big loss as the population which is computed by the MPC method which reflects realistic conditions. But the important information which is taken from this graph is the ability of the MPC method to compensate quite fast the omitted injection. Starting at day 13 we do not see great differences in the two populations apart from the EPO concentration which was administered at day 18. But this effect appears due to the optimization process: The open loop problem only optimizes the control in the interval $[0, 20]$ while the control for day 18 computed by the MPC method refers to the interval $[18 - \Delta t, 28 - \Delta t]$ with the discretization rate $\Delta t$. 

MPC method (dark blue line) and the open loop method (dotted light blue line). The injection day which was skipped is marked with a red asterisk.
In the example described above we have assumed that the omitted injection was at day 9 which corresponds to a Wednesday in our model. Suppose now that either a Monday or a Friday is skipped and thus a long injection pause is forced. We find in this case that the EPO concentration reduces fast enough that the ideal population at the next injection day ends approximately at the same point as the population computed by the MPC method. The high half-life of EPO reduces the possibility to influence the population in a favored way over a longer time period.

### 6.3.5 Administration of wrong dose rate

Another case that might occur in reality is that a wrong dose rate was administered by mistake. We assume a treatment period of \([t_0, t_f] = [0, 20]\) and a desired population \(y_d = 2.5 \times 10^8\). At day 4 which corresponds to the third injection day \(t^*_3\) only half the dose which was computed by the MPC method with \(\Delta t_M = 10\) was administered.

In Figure 6.19 the population determined by the wrong dose (blue line) is shown in the upper section, compared to the population which would have turned out if the correct dose rate was administered (dotted light blue line). We see again that the MPC method is able to compensate the missing concentration afterwards. The differences in the EPO amounts at the end of the interval appear due to the optimization strategy as explained in the last section. The open loop solver only optimizes the control for the interval \([0, 20]\) whereas the MPC control at day 18 is computed for the interval \([10 - \Delta t, 28 - \Delta t]\) with the discretization rate \(\Delta t\).

### 6.3.6 Suboptimal model predictive control

In Section 5.2 we have presented a way to speed up the MPC method by using suboptimal solutions of the optimal control problem. Inequality (5.12),

\[
J_M(i, y_{\tilde{\mu}_M}(0), \tilde{u}^{(n)}_i) \leq J_M(i - 1, y_{\tilde{\mu}_M}(i - 1), \tilde{u}_{i-1}) - \alpha_{\text{sub}}(i - 1, y_{\tilde{\mu}_M}(i - 1), \tilde{u}_{i-1}(0)),
\]

gives the termination criterion which determines the suboptimal control. We will now test the usefulness of including this termination criterion to the algorithm and its effects on CPU time and optimality of the solution. Consider a nine-dimensional problem with \([t_0, t_f] = [0, 20]\) and different desired populations \(y_d\). We will examine which termination criterion is fulfilled in each iteration. We take into account
Figure 6.19: The upper section shows the population computed by the MPC method (dark blue line) with a prediction horizon $\Delta t_M = 10$ for the case that at injection day $t_3^*$ only half of the computed EPO was administered. The dotted light blue line gives the ideal reference population computed by the open loop method for correct administration. The red line marks the desired population $y_d = 2.5 \cdot 10^8$. In the lower section the corresponding EPO values are given for the MPC method (dark blue line) and the open loop (dotted light blue line). The injection day at which the wrong dose was administered is marked with a red asterisk.

A. the suboptimality criterion (5.12),

B. the optimality criterion $\|u - u(1)\| \leq \varepsilon_{\text{abs}} + \varepsilon_{\text{rel}} r_0$ which was given in (4.7) and

C. a maximal iteration number $\max_{\text{iter}} = 100$

in each of the eight iterations. Note that the first control is identified by the optimal MPC algorithm so that we are able to test inequality (5.12) for the subsequent iterations. In the following tables the reason for the termination of the iteration is marked with an "x". Take the BFGS method, $\alpha_{\text{sub}} = 0.5$ and we find for a prediction horizon $\Delta t_M = 10$:
6 Numerical results

• $y_d = 1.8 \cdot 10^8$

<table>
<thead>
<tr>
<th>Iteration</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td>A</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

• $y_d = 2.5 \cdot 10^8$

<table>
<thead>
<tr>
<th>Iteration</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td>A</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

• $y_d = 3.5 \cdot 10^8$

<table>
<thead>
<tr>
<th>Iteration</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td>A</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Taking a larger prediction horizon, for example $\Delta t_M = 20$, we find that the situation changes and we can not predict if the suboptimality criterion is rather fulfilled than the optimality criterion. We see in this example that sometimes the suboptimality criterion represents a stronger condition than the actual optimality criterion. As it was stated in [10], the suboptimality criterion is not necessarily feasible. So we choose now to include both, the optimality as well as the suboptimality criterion, as termination criteria in our algorithm.

**Choice of $\alpha_{sub}$**

The parameter $\alpha_{sub}$ which determines the termination of the algorithm due to the suboptimality criterion is an element in $(0, 1]$. The larger it is chosen the harder the suboptimality criterion is satisfied. We expect that for decreasing values of $\alpha_{sub}$ the CPU time reduces to the price of less optimal controls.

We take again $[t_0, t_f] = [0, 20]$ which yields a nine-dimensional problem. Let us fix a desired population $y_d = 2.5 \cdot 10^8$ and the BFGS method. Applying the optimal MPC algorithm we receive the following table:
The open loop solver represents an ideal way to solve an optimization problem if everything was known in advance. Thus, we can compute the value of the objective functional which is given by $\hat{J}(u^*) = 2.37 \cdot 10^{-4}$ in order to compare the values which were determined by the MPC method. It seems in this case that $\Delta t_M = 15$ is a good choice for the prediction horizon. But in every case the value of the objective functional which is computed by the MPC method is bounded below by the value computed by the open loop method.

We are now able to compare these values to the ones obtained by the suboptimal MPC algorithm. The values of the objective functional as well as the corresponding CPU times are given in the next two tables:

- **Value of the objective functional:**

<table>
<thead>
<tr>
<th>$\Delta t_M$</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{J}(u^*_{\text{opt}})$ (in $10^{-4}$)</td>
<td>3.20</td>
<td>2.81</td>
<td>2.77</td>
<td>2.76</td>
<td>2.96</td>
</tr>
<tr>
<td>CPU time</td>
<td>5.3s</td>
<td>9.1s</td>
<td>21.4s</td>
<td>29.0s</td>
<td>64.5s</td>
</tr>
</tbody>
</table>

In almost every column the value drops with the increase of $\alpha_{\text{sub}}$ but it is only a minor difference. As in the optimal MPC method the prediction horizon $\Delta t_M = 15$ represents a preferable choice in this case.

- **CPU times:**

<table>
<thead>
<tr>
<th>$\Delta t_M$</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU time</td>
<td>3.4s</td>
<td>7.8s</td>
<td>17.2s</td>
<td>25.0s</td>
<td>60.6s</td>
</tr>
<tr>
<td>$\alpha_{\text{sub}}$</td>
<td>0.1</td>
<td>0.4</td>
<td>0.7</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>3.10</td>
<td>3.06</td>
<td>2.70</td>
<td>2.91</td>
<td>3.08</td>
</tr>
<tr>
<td>0.1</td>
<td>3.10</td>
<td>3.06</td>
<td>2.76</td>
<td>2.91</td>
<td>3.02</td>
</tr>
<tr>
<td>0.4</td>
<td>3.10</td>
<td>2.90</td>
<td>2.77</td>
<td>2.83</td>
<td>2.96</td>
</tr>
<tr>
<td>0.7</td>
<td>3.20</td>
<td>2.81</td>
<td>2.77</td>
<td>2.76</td>
<td>2.96</td>
</tr>
<tr>
<td>1.0</td>
<td>3.20</td>
<td>2.81</td>
<td>2.77</td>
<td>2.76</td>
<td>2.96</td>
</tr>
</tbody>
</table>
With the increase of $\alpha_{\text{sub}}$ the computation times increase as well. This is as expected since a larger value of $\alpha_{\text{sub}}$ yields a stronger termination condition. But compared to the CPU times of the optimal MPC algorithm we do not receive a great improvement in matters of computational effort.

The wanting monotony in the decrease of CPU time respectively increase in the objective functional value for the suboptimal MPC method might be up to the non-convex structure of the objective functional. In some cases the iteration is terminated because the suboptimality condition is fulfilled. The found control yields the initial condition for the next iteration which is performed on the following prediction horizon. But since this control might be another one than the optimal MPC algorithm has determined, the path of iterates in the next interval can differ from the one computed by the optimal MPC algorithm. This could be an explanation that the values are not monotonous as expected. Altogether, the suboptimal MPC method does not appear to be the appropriate choice to reduce the CPU time since the time saving is rather negligible.
7 Conclusion

In this thesis a specific state equation which models the change of the population density of CFU-E cells subject to the injection amount of the hormone erythropoietin (EPO) was considered. The CFU-E cells represent one stage of maturation of red blood cells. To be able to easily extend the numerical strategies to other cell types the model equation was normalized. Subsequently, the normalized equation was reformulated as a Cauchy problem. Using Legendre polynomials a discretization in space was carried out. This semi-discretized state equation was used to formulate an optimal control problem. The chosen objective functional which should be minimized measured the deviation of the computed population to a desired population and the medical therapy costs. Theoretical examinations of the optimal control problem showed the existence of an optimal control.

In order to treat the optimal control problem numerically, several numerical methods were considered, namely the gradient method or the BFGS method endowed with the Armijo step size strategy. For the solution of the state equation the \( \vartheta \)-method was explained. These methods were applied in an open loop process as well as in a closed loop framework, the model predictive control (MPC). Finally, the implementation of these methods was carried out.

First, the \( \vartheta \)-method which was used to solve the semi-discretized state equation was examined. As expected, the implicit method showed great stability in contrary to the explicit version. Therefore, we decided to use the implicit Euler method which is equivalent to \( \vartheta = 1 \). Since the objective functional consists of two weighted terms, we investigated the influence of different weights. Using the gradient method the minimum was found in every case and the optimization algorithm showed reasonable results. But the effort of considerably reducing the medical treatment costs resulted in bad approximations of the desired population. Hence, the weights for the medical treatment costs were set to zero for further computations. Comparing the gradient method and the BFGS method we found that the gradient method is very robust but rather slow. In comparison, the BFGS method converged locally very fast if the initial point was close to the optimum.
The Armijo step size strategy yielded good step sizes in most cases. On occasion the Armijo condition was not fulfilled if the objective functional became too flat. We chose to neglect stabilizing terminal constraints for the MPC method since they resulted in non-optimal solutions due to the discrete structure of the control. The choice of the size of the prediction horizon was not an easy task. The optimality of the solution increased with its size as well as the computing time. However, it was not a monotonous correlation and thus a small horizon might be a good choice for particular problems. The MPC method was also capable of quickly reacting to unexpected changes in the problem setting. In contrary to these good results, the suboptimal MPC method did not yield large computation time savings and is therefore not recommendable.

Concluding, the formulated and implemented numerical methods deliver good results for the problem at hand. However, choosing another set in the wide variety of parameters which are included in this optimal control problem - apart from the combination which we had fixed for the numerical computations - might change the outcomes or the feasibility of the numerical methods. This requires a sensitivity analysis in order to measure their individual influences. In addition, the full model - which includes models for different degrees of maturity as well as a feedback law for internal EPO release - should be examined to obtain a more realistic assessment of the proposed medical treatment. Furthermore, it is possible to expand the model by taking the influence of iron concentration into account and thus obtaining a more precise description of the underlying biological process. Hence, it seems obvious to consider forms of model reduction as proper orthogonal decomposition or reduced basis methods when turning to the more complex problem. Combining those different methods might enable to get a good mathematical description of the development of red blood cells as well as the ability to optimize medical treatment with low computing time.
# List of abbreviations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y$</td>
<td>Population density of cells</td>
</tr>
<tr>
<td>$x \in [\bar{x}, \underline{x}]$</td>
<td>Maturity parameter</td>
</tr>
<tr>
<td>$\omega = \bar{x} - \underline{x}$</td>
<td>Length of maturity domain</td>
</tr>
<tr>
<td>$[t_o, t_f]$</td>
<td>Time interval</td>
</tr>
<tr>
<td>$\Omega = [\bar{x}, \underline{x}]$</td>
<td>Space domain</td>
</tr>
<tr>
<td>$\beta \in \mathbb{R}^+$</td>
<td>Proliferation rate</td>
</tr>
<tr>
<td>$\alpha(t; u; \mu)$</td>
<td>Mortality rate</td>
</tr>
<tr>
<td>$Q = (t_o, t_f) \times \Omega$</td>
<td>Time-space cylinder</td>
</tr>
<tr>
<td>$u \in \mathbb{R}^m$</td>
<td>Control variable</td>
</tr>
<tr>
<td>${t_1^<em>, \ldots, t_m^</em>}$</td>
<td>Injection days</td>
</tr>
<tr>
<td>$TBV \in \mathbb{R}$</td>
<td>Total blood volume</td>
</tr>
<tr>
<td>$D_\mu \subset \mathbb{R}^3$</td>
<td>Parameter set</td>
</tr>
<tr>
<td>$g$</td>
<td>Constant boundary function</td>
</tr>
<tr>
<td>$y_o$</td>
<td>Initial population density</td>
</tr>
<tr>
<td>$V = H^1(\Omega)$</td>
<td>Problem-specific space</td>
</tr>
<tr>
<td>$H = L^2(\Omega)$</td>
<td>Problem-specific space</td>
</tr>
<tr>
<td>$Y = H^1(t_o, t_f; H) \cap L^2(t_o, t_f; V)$</td>
<td>Problem-specific space</td>
</tr>
<tr>
<td>$U = \mathbb{R}^m$</td>
<td>Control space</td>
</tr>
<tr>
<td>$U_{ad} \subset \mathbb{R}^m$</td>
<td>Set of admissible controls</td>
</tr>
<tr>
<td>$X = Y \times U$</td>
<td>State and control set</td>
</tr>
<tr>
<td>$X_{ad} \subset X$</td>
<td>Admissible state and control set</td>
</tr>
<tr>
<td>$Z = L^2(t_o, t_f; H) \times L^2(t_o, t_f) \times H$</td>
<td>Target set of constraint function</td>
</tr>
</tbody>
</table>
## Conclusion

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J : X \rightarrow \mathbb{R}$</td>
<td>Objective functional for the optimal control problem</td>
</tr>
<tr>
<td>$\gamma_1, \sigma_1, \ldots, \sigma_m$</td>
<td>Positive weights</td>
</tr>
<tr>
<td>$e : X \rightarrow Z$</td>
<td>Constraint function</td>
</tr>
<tr>
<td>$L^2_\omega = (L^2(0,1), | \cdot |_\omega)$</td>
<td>Space of normalized functions</td>
</tr>
<tr>
<td>$|\tilde{\phi}|_\omega := \omega^{1/2}</td>
<td>\tilde{\phi}</td>
</tr>
<tr>
<td>$X^N \subset L^2_\omega$</td>
<td>Finite dimensional subspace</td>
</tr>
<tr>
<td>$Y^N = H^1(t_0, t_f; X^N)$</td>
<td>Space of semi-discretized states</td>
</tr>
<tr>
<td>$Z^N = L^2(t_0, t_f; X^N) \times X^N$</td>
<td>Target set of discrete constraint function</td>
</tr>
<tr>
<td>$P^N : L^2_\omega \rightarrow X^N$</td>
<td>Orthogonal projection</td>
</tr>
<tr>
<td>$J^N : Y^N \times U \rightarrow \mathbb{R}$</td>
<td>Discrete objective functional</td>
</tr>
<tr>
<td>$e^N : Y^N \times U \rightarrow Z^N$</td>
<td>Discrete constraint function</td>
</tr>
</tbody>
</table>


Acknowledgements

First of all, I would like to thank Prof. Dr. Stefan Volkwein for offering me the possibility to write this thesis at his chair and introducing me to the interesting subject of optimal control problems. Thank you for supervising and discussing the problems which occurred during this thesis and helping with your experience in theoretical and practical questions.

Second, thanks to Prof. Dr. Franz Kappel, who kindly agreed to review this thesis and helped together with Dr. Doris Fürtinger to answer specific questions on the model.

I would like to thank Sabrina Rogg, Carmen Gräßle and Dennis Beermann for thoroughly proofreading and for discussing theoretical and practical aspects about the optimal control problem.

Then, I would like to thank my parents who enabled my studies in the first place and have supported me over the years in all situations. I am very grateful for having you. And last, thanks to Jan Schmidt for always being there for me, no matter what happens.

This diploma thesis has been performed in cooperation with the Renal Research Institute in New York.