ulm university universität ulm

A Numerical Approach to Model Reduction for Optimal Control of Multiscale ODE

Dissertation zur Erlangung des Doktorgrades Dr. rer. nat der Fakultät für Mathematik und Wirtschaftswissenschaften der Universität Ulm vorgelegt von *Marcel Rehberg* aus Grevesmühlen.

Mai 2013

amtierender Dekan: Prof. Dr. Dieter Rautenbach
1. Gutachter: Prof. Dr. Dirk Lebiedz
2. Gutachter: Prof. Dr. Knut Graichen
Tag der Promotion: tba

Abstract

The topic of this thesis is model (order) reduction in the context of numerical optimal control. Complex mathematical models based on ordinary differential equations, can often be reduced in order to decrease computational complexity and enable their use in control algorithms. A feature that can be exploited for the purpose of model reduction is time scale separation, which means that the system dynamics are comprised of fast and slow processes. Fast states relax to a slow invariant manifold in the state space, which is parametrized by the slow states. This manifold is approximated numerically and the fast species can be eliminated from the full model and thus also from the optimal control problem. To this end multivariate interpolation based on radial basis functions is used and compared with an online approach that solves the model reduction problem ad hoc. Singular perturbation theory is employed to illustrate the theoretical background and illustrate how reduced models influence the solution of the optimal control problem. It is shown via numerical experiments that the application of the model reduction techniques can lead to significant savings in computation time without compromising the quality of the optimal control solution.

Zusammenfassung

Das Thema dieser Arbeit ist Model(ordnungs)reduktion im Kontext von numerischer Optimalsteuerung. Komplexe mathematische Modelle, basierend auf gewöhnlichen Differentialgleichungen, können oftmals reduziert werden, was aufgrund des verringerten Rechenaufwands ihre Anwendung in Steuerungsalgorithmen ermöglicht. Zeitskalenseparation ist eine Eigenschaft, die zur Modelreduction ausgenutzt werden kann. Dabei ist die Systemdynamik aus schnellen und langsamen Prozessen zusammengesetzt. Schnelle Zustände relaxieren auf eine langsame, invariante Manigfaltigkeit im Zustandsraum, die von den langsamen Spezies parametrisiert ist. Diese Manigfaltigkeit kann numerisch approximiert werden womit die schnellen Zustände aus dem vollen Modell und aus dem Optimalsteuerungsproblem entfernt werden können. Zu diesem Zweck wird multivariate Interpolation basierend auf radialen Basisfunktionen angewandt. Dieser Ansatz wird mit einem online Verfahren verglichen, bei dem das Modellreduktionsproblem ad hoc gelöst wird. Um den theoretischen Hintergrund zu illustrieren, und um um reduzierte Optimalsteuerungsprobleme zu analysieren wird singuläre Störungstheorie verwendet. Numerische Experimente zeigen, dass die Anwendung von Modellreduktionstechniken zu signifikanten Einsparungen bei der Rechenzeit führen kann, ohne die Qualität der optimalen Lösung zu sehr zu verschlechtern.

Contents

CHAPTER 1

Introduction

In this work the numerical solution of optimal control problems constrained by ordinary differential equations with considerable time scale separation is studied. This involves applying techniques from mathematical control theory, singular perturbation theory, approximation theory, and numerical analysis as well as scientific computing.

Time scale separation is a natural occurring feature of many real world systems and means that different states of the system evolve with significantly different speeds. This can be exploited analytically and helps to understand the structure of solutions of initial value, boundary value, and optimal control problems. Moreover, under certain circumstances time scale separation allows to reduce a model to the slower moving states which in turn can facilitate numerical algorithms that work faster with the reduced model. The development and integration of such algorithms plays an important role in this work.

A classic example of mathematical models with explicit time scale separation are *singularly perturbed* problems in which the fast state variables are characterized by multiplication of the (left-hand side) derivative with a *small* parameter. These systems exhibit the interesting property that the dynamics can be decomposed into a slow and a fast part and each of this parts can be regarded independently. We study initial value, boundary value, and optimal control problems based on singularly perturbed systems and present results that highlight the decomposition of the solution into slow and fast varying components.

Another major feature of dynamic systems with two time scale behavior is the presence of lower dimensional integral manifolds or *slow invariant manifolds* (SIM) that represent the relaxed fast states parametrized by the slow states. This manifold can be computed numerically and employed to reduce the order of the model in question.

To utilize this manifold in the context of numerical optimal control, multidimensional interpolation based on radial basis functions is used. Alternatively, the model reduction problem can be solved ad-hoc during the ongoing solution of the optimal control problem.

Many of the theoretical results, especially from singular perturbation and control theory are illustrated with the help of two descriptive examples, which are also used for the final assessment of the newly developed numerical procedures. This way the two examples are extensively analyzed from different angles. Furthermore, newly introduced concepts are illustrated with familiar systems and a comparison between theoretical and numerical results is enabled.

The main contribution of this thesis lies in the bundling and channeling of findings and tools from different fields of applied mathematics to the problem of reduced optimal control. Besides clarifying what optimal control results are principally obtainable with reduced model the focus is on numerical implementation. A fast interpolation routine based on *radial basis functions* and *partition of unity* is the main reason for the observed speed up in the computation of optimal control results.

1. Outline

In Chapter 2 singular perturbation theory is outlined. The classic analytical approach is presented and contrasted with a geometric approach that investigates integral manifolds. In both cases expressions for the solution of the system with dependence on a small parameter can be derived. We study initial value problems first and later boundary value problems, since they emerge in optimal control applications.

Mathematical (optimal) control theory is one cornerstone of this work and important results are highlighted in Chapter 3. The well known controllability statements for linear and nonlinear systems are given and the Pontryagin minimum principle as one central result for optimal control is discussed in greater detail. Picking up the second chapter the implications of singular perturbation theory for (optimal) control theory are presented. The last section of the chapter is devoted to the numerical approach to control problems.

The topic of Chapter 4 is multidimensional interpolation. After a short introduction to the problem we quickly turn to the *radial basis function* (RBF) method and develop the analytical framework necessary to state central convergence and stability results. Since the interpolation is crucial for the computational performance of the optimal control procedure the development of fast algorithms and implementations plays a major role in the later half of the chapter.

Chapter 5 deals with model reduction. After a short introduction to the general concept our approach based on slow invariant manifolds is described. This manifolds can be approximated numerically and the involved procedures are explained in detail. Some connections to singularly perturbed systems are also inspected.

Finally, in Chapter 6 the results and techniques introduced in the previous chapters are brought together and are applied to three example optimal control problems. It is shown that for two of the three examples the model reduction allows to solve the optimal control problem significantly faster without loosing the main characteristics of the full control solution. The third example displays a negative result inasmuch as numerical instabilities prevent the successful application of the model reduction toolbox.

The final Chapter 7 concludes this thesis with a summary, some final remarks, and an outlook to possible future work.

2. Notation, Language, and Ordinary Differential Equations

Here we are going to clarify some notational aspects and conventions used in this work. Also basic results regarding the solvability of ordinary differential equations and properties of this solution are given here, because they are featured prominently throughout the thesis.

2.1. Notation and Language. Vectors of real numbers are abundantly used and denoted by plain letter symbols without additional markers. They are generally assumed to be columns. For the frequently needed vector $x \in \mathbb{R}^d$, $d \in \mathbb{N}$ we will use $\chi_1, \chi_2, \ldots, \chi_d$ for its components. For two vectors x_1 and x_2 we will add an additional superscript, for example as in $\chi_1^{(2)}$ for the first component of x_2. Transposes of vectors and matrices are indicated with $(\cdot)^{\mathrm{T}}$. Zero vectors and matrices are denoted with 0, the dimensions can usually be concluded from the context. Unit vectors and matrices are given by $I_k \in \mathbb{R}^n$ and $I \in \mathbb{R}^{n\times n}$, respectively. The $k \in \{1, 2, \ldots, n\}$ indicates the position of the 1.

Let H be a (pre-)Hilbert space. For elements $x, y \in H$ we denote the inner product with

$$(x, y)_H .$$

The subscript H is omitted if the space it is referring to, is clear from the context.

Multi-indices are a convenient way of expanding index notation to the multivariate case.

DEFINITION 1 (Multi-index). The vector $\alpha \in \mathbb{N}_0^d$, $d \in \mathbb{N}$ is called a **multi-index**. Let α and β be multi-indices and $x \in \mathbb{R}^d$. We define

- $\alpha \pm \beta := (\alpha_1 \pm \beta_1, \alpha_2 \pm \beta_2, \dots, \alpha_n \pm \beta_n)$;
- $|\alpha| := \alpha_1 + \alpha_2 + \cdots + \alpha_n$ (**order**);
- $x^\alpha := \chi_1^{\alpha_1} \chi_2^{\alpha_2} \cdots \chi_d^{\alpha_d}$.

The symbol $C^k(\Omega)$ is used to denote the space of k-times differentiable functions over a domain $\Omega \subset \mathbb{R}^d$. The argument Ω is omitted if it is clear from the context. Functions in C^∞ are called smooth.

We use D as the exclusive operator for describing total and partial derivatives. A variety of sub- and superscripts is used to denote the relevant cases. Let $f : \mathbb{R}^d \to \mathbb{R}^m$, $g : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, and $h : \mathbb{R}^d \to \mathbb{R}^d$ be smooth functions. The different notations are

- $\mathrm{D}_x\, h(x) := \frac{\mathrm{d}}{\mathrm{d}x} h(x)$ for the total derivative of h with respect to x. For single valued functions we omit the argument of D and write $\mathrm{D}\, h$;
- $\mathrm{D}_i\, f := \frac{\partial f}{\partial \chi_i}$, $i \in \{1, 2, \dots, d\}$ for the partial derivative of f with respect to the i-th input variable;
- $\mathrm{D}_x\, g\big(x, h(x)\big) = \mathrm{D}_1\, g + \mathrm{D}_2\, g\, \mathrm{D}_h$ for combinations;
- a superscript $j \in \mathbb{N}$ to denote higher derivatives as in $\mathrm{D}_2^4\, f$ which would be the fourth order partial derivative of f with respect to the second variable;
- multiple subscripts for repeated differentiation as in $\mathrm{D}_{2\,3}$ which is f differentiated with respect to the third and second input variable;
- $\mathrm{D}^\alpha := \frac{\partial^{\alpha_1} f}{\partial \chi_1^{\alpha_1}} \frac{\partial^{\alpha_2} f}{\partial \chi_2^{\alpha_2}} \cdots \frac{\partial^{\alpha_d} f}{\partial \chi_d^{\alpha_d}}$, where α is a multi-index, for repeated partial derivatives with respect to different input variables and with different orders;
- $$\mathrm{D}_*\, f := \begin{pmatrix} \frac{\partial f_1}{\partial \chi_1} & \frac{\partial f_1}{\partial \chi_2} & \cdots & \frac{\partial f_1}{\partial \chi_d} \\ \frac{\partial f_2}{\partial \chi_1} & \frac{\partial f_2}{\partial \chi_2} & \cdots & \frac{\partial f_2}{\partial \chi_d} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_m}{\partial \chi_1} & \frac{\partial f_m}{\partial \chi_2} & \cdots & \frac{\partial f_m}{\partial \chi_d} \end{pmatrix}$$
  for the Jacobian of f. Alternatively, if f depends on several r multivariate variables $x_1, x_2, \dots, x_r$ we use D_{x_k} for the partial Jacbian with respect to x_k.
- $\mathrm{D}_*^2\, g$ for the Hessian of g;

2.2. Ordinary Differential Equations. Ordinary differential equations play a fundamental role in this work. We regard the initial value problem

$$\mathrm{D}_t\, x(t) = f\big(t, x(t)\big), \quad x(\tau) = \xi, \tag{1}$$

where $x : \mathbb{R} \to \mathbb{R}^{n_x}$, $f : \mathbb{R} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ on the interval $t \in [0, T]$. For this problem to have a solution the right hand side function f must be *Lipschitz* continuous on a domain of interest, with respect to its second argument.

DEFINITION 2 (Lipschitz continuity, [**67**, Definition 4.1]). Let $\Omega \subset \mathbb{R} \times \mathbb{R}^{n_x}$, $f : \Omega \to \mathbb{R}^{n_x}$. Then f is said to be **Lipschitz continuous** on Ω with respect to its second argument if a constant $L \in \mathbb{R}$, $L > 0$ exists such that for all (t, x_1), $(t, x_2) \in \Omega$ it holds that

$$\|f(t, x_1) - f(t, x_2)\| \leq L \|x_1 - x_2\|.$$

If the right hand side function f is Lipschitz continuous on a domain Ω a solution of the initial value problem (1) exists.

THEOREM 3 ([**67**, Satz 4.7]). *Let f from (1) be continuous with respect to t and Lipschitz continuous with respect to x on a domain*

$$\Omega = \{(t, x) \,|\, t \in [0, T],\ x \in \mathbb{R}^{n_x}\}.$$

Then, problem (1) has a unique solution $x(t)$ on $[0, T]$ for every $(\tau, \xi) \in \Omega$.

This is the basic existence and uniqueness result for initial value problems subject to ordinary differential equations. Often we are confronted with problems depending on constant parameters $p \in \mathbb{R}^{n_p}$, i.e. f is extended and we have

$$\mathrm{D}_t\, x(t) = f(t, x(t), p), \quad x(\tau) = \xi. \tag{2}$$

The solution x is now also regarded as function of ξ and p: $x(t, \xi, p)$. It is often convenient to write the solution of (2) in closed form

$$\phi : \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_x},\ \phi(t, \tau, \xi, p) := x(t),$$

and collect all dependencies in one function. Here, ϕ is the state $x(t)$ after solving the initial value problem $x(\tau) = \xi$ with the parameters p. If f is partially differentiable with respect to x and p then ϕ is too.

THEOREM 4 ([**39**, Satz 13.2]). *Let f from (2) be continuous with respect to t and partially differentiable with respect to x and p on a domain*

$$\Omega = \{(t, x, p) \,|\, t \in [0, T],\ x \in \mathbb{R}^{n_x},\ p \in \mathbb{R}^{n_p}\}.$$

The solution $\phi(t, \tau, \xi, p)$ of (2) is partially differentiable with respect to ξ and p for all $(t, \xi, p) \in \Omega$ and $\tau < t$.

REMARK 1. The partial derivatives with respect to initial values and parameters are also called sensitivities. If (2) is written in integral form

$$\phi(t, \tau, \xi, p) = \xi + \int_\tau^t f(\sigma, \phi(\sigma, \tau, \xi, p), p)\mathrm{d}\sigma$$

a differential equation for the initial value sensitivity $\mathrm{D}_\xi\, \phi(t, \tau, \xi, p)$ can be obtained via simply differentiating the above equation:

$$\mathrm{D}_\xi\, \phi(t, \tau, \xi, p) = 1 + \int_\tau^t \mathrm{D}_x\, f(\sigma, \phi(\sigma, \tau, \xi, p), p)\, \mathrm{D}_\xi\, \phi(\sigma, \tau, \xi, p)\mathrm{d}\sigma,$$

or in differential form (omitting the arguments for clarity)

$$\mathrm{D}_t\, \mathrm{D}_\xi\, \phi = \mathrm{D}_x\, f(\sigma, \phi, p)\, \mathrm{D}_\xi\, \phi, \quad \mathrm{D}_\xi\, \phi(\tau, \tau, \xi, p) = 1.$$

Analogously,

$$\mathrm{D}_t\, \mathrm{D}_p\, \phi = \mathrm{D}_x\, f(\sigma, \phi, p)\, \mathrm{D}_p\, \phi, \quad \mathrm{D}_p\, \phi(\tau, \tau, \xi, p) = 0.$$

These are the so called *sensitivity* or *adjoint* differential equations to problem (2).

Proofs for this general results are left out and can be found in any basic textbook dealing with ordinary differential equations, and especially in the cited references. With the basic results in place we are now going to take a look at systems with time scale separation.

CHAPTER 2

Singular Perturbation Theory

Systems, natural or artificial, that are composed of sub-processes which are evolving on different time scales are abundant. Mathematical models that try to capture such systems are inheriting the time scale properties, if they are reaching a certain level of accuracy. If models based on *ordinary differential equations* (ODEs) are used, then multiple time scales often lead to stiff systems and special care must be taken to solve those systems fast and reliably [**37**]. A classic approach to ODE systems with explicit time scale separation is singular perturbation theory which is the main topic of this chapter. We are going to highlight the important results, first for initial then for boundary value problems.

1. Singularly Perturbed Initial Value Problems

The following material is based on [**40**]. The classic singularly perturbed initial value problem has the form

$$\begin{aligned} \mathrm{D}_t\, x &= f(t,x,y,\varepsilon), \quad x(0) = \xi(\varepsilon) \\ \varepsilon\, \mathrm{D}_t\, y &= g(t,x,y,\varepsilon), \quad y(0) = \eta(\varepsilon) \end{aligned} \tag{3}$$

where $0 < \varepsilon \leq 1$. The overall state vector $z(t) = \big(x(t), y(t)\big)^{\mathrm{T}} \in \mathbb{R}^{n_x+n_y}$ is decomposed into the so called slow and fast states $x(t)$ and $y(t)$ respectively. We only regard the problem on the finite interval $t \in [0,T]$. Further assumptions regarding the right-hand side functions f and g are given later. The parameter ε in equation (3) represents the explicit time scale separation. The eigenvalues of the Jacobian of $\frac{1}{\varepsilon} g(t,x,y,\varepsilon)$ with respect to x and y are of the order $\mathcal{O}(\varepsilon)$ which represents the time scale the fast modes evolve on. Accordingly, t is called the slow time whereas $\tau = \frac{t}{\varepsilon}$ is the fast time and the transformed fast system

$$\begin{aligned} \mathrm{D}_\tau\, x &= \varepsilon f(\tau,x,y,\varepsilon), \quad x(0) = \xi(\varepsilon) \\ \mathrm{D}_\tau\, y &= g(\tau,x,y,\varepsilon), \quad y(0) = \eta(\varepsilon) \end{aligned} \tag{4}$$

is obtained. For $\varepsilon > 0$ both systems are equivalent, still (3) is better suited to analyze the slow states whereas (4) increases the resolution for the fast variables.

Several additional systems are associated with either (3) and (4) and are analyzed in the following. If we set $\varepsilon = 0$ two vastly different problems are obtained. System (3) becomes

$$\begin{aligned} \mathrm{D}_t\, x &= f(t,x,y,0), \quad x(0) = \xi(0) \\ 0 &= g(t,x,y,0), \end{aligned} \tag{5}$$

a differential algebraic equation also known as the *reduced problem*. Note that the initial conditions for y are relaxed, since in general they cannot be fulfilled anymore. A possible interpretation is that the fast states immediately relax to a stationary state for each x. The fast system (4) becomes

$$\begin{aligned} \mathrm{D}_\tau\, x &= 0, \quad x(0) = \xi(0) \\ \mathrm{D}_\tau\, y &= g(\tau,x,y,0), \quad y(0) = \eta(0), \end{aligned}$$

called the *inner problem*. Both states retain their initial values. The slow states can be regarded as constant whereas the fast states are governed by a dynamical equation.

The convergence of solutions of (3) to solutions of (5) for $\varepsilon \to 0$ is one of the central questions of singular perturbation analysis. If convergence happens on some closed interval $t \in [0, T]$ the problem is said to *degenerate regularly*. In that case we might expect that the solutions are smooth with respect to ε. This is reflected by including ε as argument of the solutions $x = x(t, \varepsilon)$ and $y = y(t, \varepsilon)$. If we impose the smoothness condition on (3) we arrive at the *outer problem*

$$\begin{aligned} \mathrm{D}_t\, x^* &= f(t, x^*, y^*, \varepsilon), \quad x^*(0, \varepsilon) = \xi^*(\varepsilon), \\ \varepsilon\, \mathrm{D}_t\, y^* &= g(t, x^*, y^*, \varepsilon), \end{aligned} \tag{6}$$

subject to the constraints that $x^*(t, \varepsilon)$ and $y^*(t, \varepsilon)$ and their derivatives with respect to ε are continuous at $\varepsilon = 0$ and $x^*(t, 0)$, $y^*(t, 0)$ are the solution of (5). The initial value $\xi^*(\varepsilon)$ is a smooth function of $\varepsilon \geq 0$ and we assume $|\xi(\varepsilon) - \xi^*(\varepsilon)| < \rho$ for some $\rho > 0$. The reason for allowing a different initial value for the outer problem becomes apparent later. There is no initial value for $y^*(t, \varepsilon)$, instead the smoothness conditions define $y^*(0, \varepsilon)$ implicitly. If solutions $x^*(t, \varepsilon)$ and $y^*(t, \varepsilon)$ of the outer problem do exist yet another system can be introduced by regarding $X := x - x^*$ and $Y := y - y^*$ and we get the following initial value problem

$$\begin{aligned} \mathrm{D}_t\, X &= \hat{f}(t, X, Y, \varepsilon), \quad X(0) = \hat{\xi}(\varepsilon), \\ \varepsilon\, \mathrm{D}_t\, Y &= \hat{g}(t, X, Y, \varepsilon), \quad Y(0) = \hat{\eta}(\varepsilon) \end{aligned} \tag{7}$$

with

$$\begin{aligned} \hat{f} &= f(t, x^* + X, y^* + Y, \varepsilon) - f(t, x^*, y^*, \varepsilon), \quad \hat{\xi}(\varepsilon) = \xi(\varepsilon) - \xi^*(\varepsilon), \\ \hat{g} &= g(t, x^* + X, y^* + Y, \varepsilon) - g(t, x^*, y^*, \varepsilon), \quad \hat{\eta}(\varepsilon) = \eta(\varepsilon) - \eta^*(\varepsilon). \end{aligned}$$

The new variables $X(t, \varepsilon)$ and $Y(t, \varepsilon)$ are called *boundary layer corrections* and account for the difference between the solution of the full system and the outer solution. Interestingly, we have

$$\hat{f}(t, 0, 0, \varepsilon) = 0, \quad \hat{g}(t, 0, 0, \varepsilon) = 0,\ t \in [0, T],$$

so 0 is a stationary point for the boundary layer correction. The study of the stability properties of this stationary point will show that under certain assumptions the boundary layer correction converges to 0 and the long term evolution of (3) is mainly governed by the outer solution. Finally, the transformation $\tau = \frac{t}{\varepsilon}$ puts (7) into a regular perturbed form

$$\begin{aligned} \mathrm{D}_\tau\, X &= \varepsilon \hat{f}(\varepsilon\tau, X, Y, \varepsilon), \quad X(0) = \hat{\xi}(\varepsilon), \\ \mathrm{D}_\tau\, Y &= \hat{g}(\varepsilon\tau, X, Y, \varepsilon), \quad Y(0) = \hat{\eta}(\varepsilon). \end{aligned} \tag{8}$$

Before we go into details a simple system will serve as a descriptive example for singular perturbed problems.

Example 1. Consider the nonlinear system

$$\begin{aligned} \mathrm{D}_\sigma\, s &= -k_1 e s + k_{-1} c, \\ \mathrm{D}_\sigma\, e &= -k_1 e s + (k_{-1} + k_2) c, \\ \mathrm{D}_\sigma\, c &= k_1 e s - (k_{-1} + k_2) c, \\ \mathrm{D}_\sigma\, p &= k_2 c \end{aligned} \tag{9}$$

describing the enzymatic reaction

$$\mathrm{S} + \mathrm{E} \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \mathrm{SE} \xrightarrow{k_2} \mathrm{P} + \mathrm{E}$$

of the substrate S to the product P via the complex SE of substrate and enzyme E. The quantities in equation (9) are the relative concentrations of the involved species: $s = [\mathrm{S}]$, $e = [\mathrm{E}]$, $c = [\mathrm{SE}]$, and $p = [\mathrm{P}]$. We assume that at the beginning of the reaction there is zero substrate-enzyme-complex and product, so that

$$s(0) = s_0, \quad e(0) = e_0, \quad c(0) = 0, \quad p(0) = 0.$$

The differential equation for p is decoupled from the rest of the system, p can be obtained through integration once c is known. Also

$$\mathrm{D}_\sigma e + \mathrm{D}_\sigma c = 0 \quad \Rightarrow \quad e(\sigma) + c(\sigma) = e_0 \quad \Leftrightarrow \quad e(\sigma) = e_0 - c(\sigma),$$

because the enzyme is not consumed in the reaction. The remaining system is

$$\begin{aligned} \mathrm{D}_\sigma s &= -k_1 e_0 s + (k_1 s + k_{-1})c, \quad s(0) = s_0, \\ \mathrm{D}_\sigma c &= k_1 e_0 s - (k_1 s + k_{-1} + k_2)c, \quad c(0) = 0. \end{aligned}$$

The next step consists of introducing new variables (nondimensionalisation)

$$t = k_1 e_0 \sigma, \quad x(t) = \frac{s(\sigma)}{s_0}, \text{and} \quad y(t) = \frac{c(\sigma)}{e_0}$$

and parameters

$$\beta = \frac{k_2}{k_1 s_0}, \quad K = \frac{k_{-1} + k_2}{k_1 s_0}, \quad \text{and} \quad \varepsilon = \frac{e_0}{s_0}$$

to obtain

$$\begin{aligned} \mathrm{D}_t x &= -x + (x + K - \beta)y, \quad x(0) = 1, \\ \varepsilon \, \mathrm{D}_t y &= x - (x + K)y, \quad y(0) = 0. \end{aligned} \tag{10}$$

Note, that a usual assumption for enzymatic reactions is that $s_0 \gg e_0$ so that $\varepsilon \ll 1$.

For $K = \beta = 1$ Figures 2.1 and 2.2 show the solution for $\varepsilon = 10^{-1}$ and $\varepsilon = 10^{-3}$ respectively. On the slow time scale (Figure 2.1) $y_{0.1}(t)$ and $y_{0.001}(t)$ are essentially confined to the same path after an initial transit. The curve for $\varepsilon = 10^{-3}$ shows a jump at $t = 0$ and seems to start at 0.5, which is the result of the described fast relaxation and can be linked to the boundary layer correction. On the fast time scale (depicted in Figure 2.2) we zoom in into the fast initial transient. It should be noted that the τ-scale is different for both systems, if it is converted back to t. For example, $\tau = 5$ corresponds to $t = 0.5$ for $\varepsilon = 10^{-1}$ but to $t = 0.005$ for $\varepsilon = 10^{-3}$. Especially for the smaller ε the corresponding x seems to be constant.

After this introductory example we take a closer look at the limit process $\varepsilon \to 0$. Our goal is to expand the solution of (3) into a power series in ε around $\varepsilon = 0$. As long as $\varepsilon > 0$, Lipschitz continuity of f and g guarantee the existence and uniqueness of a solution of either (3) or (4), (Theorem 3). However, a few more restrictions are necessary to state the first general result.

- A1 The reduced system (5) has a continuous solution $x(t, 0) = x_0(t)$, $y(t, 0) = y_0(t)$ on $t \in [0, T]$.
- A2 The right hand side functions f, g are in C^{R+2} with respect to (t, x, y, ε) in a neighborhood of $\big(t, x_0(t), y_0(t), \varepsilon\big)$, $t \in [0, T]$, $\varepsilon \in [0, \varepsilon_0]$, $\varepsilon_0 > 0$, $R \in \mathbb{N}$. The initial value functions ξ and η are in $C^{R+2}([0, \varepsilon_0])$.
- A3 The Jacobian

  $$g_y(t) = \mathrm{D}_y \, g\big(t, x_0(t), y_0(t), 0\big) \in \mathbb{R}^{n_y \times n_y}$$

  is nonsingular for $t \in [0, T]$.

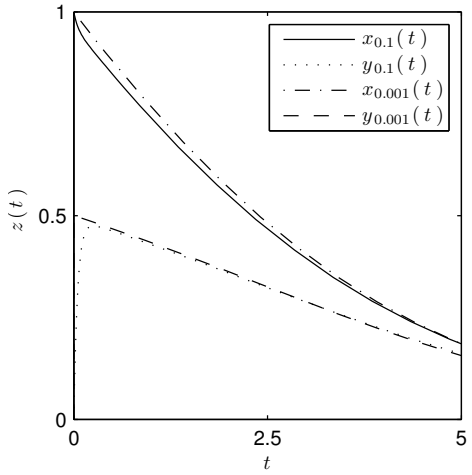

FIGURE 2.1. The enzyme example for two different values of ε, on the slow time scale t.

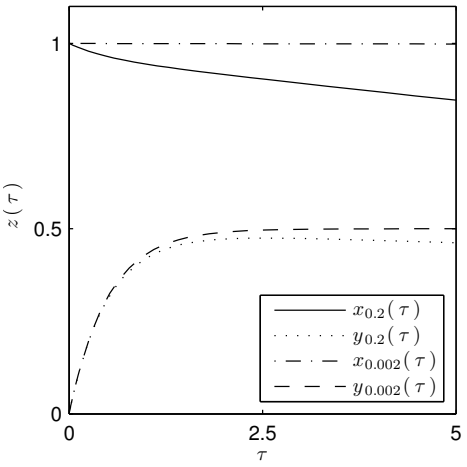

FIGURE 2.2. The enzyme example for two different values of ε on the fast time scale $\tau = \frac{t}{\varepsilon}$.

A4 There exists a smooth and nonsingular matrix $P(t) \in \mathbb{R}^{n_y \times n_y}$ such that the matrix $A(t) \in \mathbb{R}^{n_y \times n_y}$ given by

$$A(t) = P^{-1}(t) g_y(t) P(t)$$

is block triangular. For the square blocks $A_{i,j}$, $i, j = 1, 2, \ldots, N$, where N is the number of blocks, it holds that

$$A_{i,j} = \begin{cases} 0 & i > j \\ \text{nonsingular} & i = j. \end{cases}$$

This means the matrices on the main diagonal of A are invertible. Additionally, for functions $\phi_i(t, s, \varepsilon)$, $\psi_i(t, s, \varepsilon)$ defined as fundamental matrices of the systems

$$\varepsilon \, \mathrm{D}_t \, \phi_i = A_{i,i} \phi_i, \quad \phi(s, s, \varepsilon) = I, \ i = 1, 2, \ldots, M$$
$$\varepsilon \, \mathrm{D}_t \, \psi_i = A_{j,j} \psi_j, \quad \psi(s, s, \varepsilon) = I, \ j = M+1, M+2, \ldots, N$$

there is a constant $K > 0$ independent of t, s and $\varepsilon \in [0, \varepsilon_0]$ such that

$$\|\phi(t, s, \varepsilon)\| \leq K, \ 0 \leq s \leq t \leq T, \ i = 1, 2, \ldots, M$$
$$\|\psi(t, s, \varepsilon)\| \leq K, \ 0 \leq t \leq s \leq T, \ j = M+1, M+2, \ldots, N$$

holds, where $M \leq N$. Here, s is the initial time.

A5 There is a $\mu > 0$ such that the Jacobian $g_y(t)$ has k, $1 \leq k \leq n_y$ eigenvalues $\lambda_i(t)$, $i = 1, 2, \ldots, k$ with

$$\mathrm{Re}\big(\lambda_i(t)\big) \leq -\mu, \quad t \in [0, T].$$

The remaining $n_y - k$ eigenvalues satisfy

$$\mathrm{Re}\big(\lambda_j(t)\big) > -\mu, \quad t \in [0, T], \ j = k+1, k+2, \ldots, n.$$

REMARK 2. The important and interesting assumptions are the ones that put constrains on the Jacobian g_y. Assumptions A1, A3, A4 are essential for the outer solution to exist and behave "properly". First, A3 guarantees that $x_0(t)$ and $y_0(t)$ are unique and smooth with respect to t and moreover no turning point or bifurcation behavior is possible. The technical assumption A4 is fulfilled if the eigenvalues of

$A_{11}, A_{22}, \ldots, A_{M,M}$ and $-A_{M+1,M+1}, -A_{M+2,M+2}, \ldots - A_{N,N}$ have negative real parts. The important point is that the inequalities for ϕ_i, $i = 1, 2, \ldots, M$ and ψ_j, $j = M+1, M+2, \ldots, N$ hold because this restricts the eigenvalues of the blocks changing sign on the interval $t \in [0, T]$. Otherwise, since the triangularization is supposed to be smooth, a block would become singular and the estimation of the norm would break. The last assumption A5 is important for the stability properties of the stationary point of the boundary layer correction as $\varepsilon \to 0$.

Some of the above assumptions are very technical however they allow very general systems to fit into the singular perturbation setting. A very stringent but common assumption is

A6 The Jacobian $g_y(t)$ has only eigenvalues with negative real part.

In this case the stationary point 0 of the boundary layer correction is always stable and convergence is guaranteed for appropriate initial values. The assumptions A3–A5 can be replaced by A6. A less demanding replacement is

A7 The Jacobian $g_y(t)$ has k, $1 \leq k \leq n_y$ eigenvalues $\lambda_i(t)$, $i = 1, 2, \ldots, k$ with
$$\operatorname{Re}\big(\lambda_i(t)\big) \leq -\mu, \ t \in [0, T],$$
where $\mu > 0$. The remaining $n_y - k$ eigenvalues satisfy
$$\operatorname{Re}\big(\lambda_i(t)\big) \geq \mu \quad t \in [0, T].$$

THEOREM 5 ([**40**, Theorem 1]). *Let assumptions A1–A5 hold (or alternatively A1, A2, and A6 or A7). For each $\varepsilon > 0$ there is a k-dimensional manifold $S(\varepsilon) \subset \mathbb{R}^{n_y}$ such that (3) has a unique solution $x = x(t, \varepsilon)$, $y = y(t, \varepsilon)$ on $[0, T]$ if $\eta(\varepsilon) \in S(\varepsilon)$. The solutions admit a series expansion in ε:*
$$x(t, \varepsilon) = x^*(t, \varepsilon) + X(t/\varepsilon, \varepsilon)$$
$$y(t, \varepsilon) = y^*(t, \varepsilon) + Y(t/\varepsilon, \varepsilon),$$
with
$$x^*(t, \varepsilon) - \sum_{r=0}^{R} x_r^*(t)\varepsilon^r = \mathcal{O}(\varepsilon^{R+1}),$$
$$y^*(t, \varepsilon) - \sum_{r=0}^{R} y_r^*(t)\varepsilon^r = \mathcal{O}(\varepsilon^{R+1}),$$
where $x^*(t, \varepsilon)$, $y^*(t, \varepsilon)$ are solutions of (6) with $\xi^*(\varepsilon) = x^*(0, \varepsilon)$ implicitly defined. The boundary layer corrections have the representation
$$X(t/\varepsilon, \varepsilon) - \sum_{r=0}^{R} X_r(t/\varepsilon)\varepsilon^r = \mathcal{O}(\varepsilon^{R+1}),$$
$$Y(t/\varepsilon, \varepsilon) - \sum_{r=0}^{R} Y_r(t/\varepsilon)\varepsilon^r = \mathcal{O}(\varepsilon^{R+1}),$$
and $X(t/\varepsilon, \varepsilon)$, $Y(t/\varepsilon, \varepsilon)$ are solutions of (7). For X and Y the following estimate with constants $K_1, \delta_1, \varepsilon_0 > 0$ holds:
$$|X(\tau, \varepsilon)| + |Y(\tau, \varepsilon)| \leq K_1 \, |\eta(\varepsilon) - y^*(0, \varepsilon)| \, \mathrm{e}^{-\delta_1 \tau}, \ \tau \in [0, T/\varepsilon], \ \varepsilon \in (0, \varepsilon_0].$$

The theorem makes the notion of two time scales precise in as much as it gives the solution of the overall problem (3) as the sum of solutions of two subproblems. One is the fast boundary layer correction (7). The other one is the outer problem (6) which describes the long term evolution. This is because the boundary layer correction diminishes exponentially depending on the difference of the initial value $\eta(\varepsilon)$ for the fast variables y and the value of the outer solution $y^*(0, \varepsilon)$ which is

supposed to lie on some k-dimensional manifold. The speed of convergence hinges on ε. For $\varepsilon \to 0$ the boundary layer correction approaches 0 infinitely fast. This is what was observed in Example 1: The initial transient for $\varepsilon = 10^{-3}$ almost looked like a jump in $y_{0.001}$ on the slow time scale (see Figure 2.1). An idea for model simplification would be to ignore the boundary layer correction and just work with the outer solution if one is only interested in the long term behavior of the system.

REMARK 3. A remark is in order with regard to the initial values. The functions $\xi(\varepsilon)$ and $\eta(\varepsilon)$ are given and are therefore readily available. The theorem states that the initial value $\xi^*(\varepsilon)$ for the outer problem is implicitly defined. First, one must note that the outer problem itself has a solution for a general $\xi^*(\varepsilon)$ as long as this function is differentiable with high enough order and "close" to $\xi(\varepsilon)$ (see **[40]**, Lemma 1). The key to the "right" ξ^* for the application of the theorem is to look at the two initial values for the boundary layer correction:

$$\hat{\xi}(\varepsilon) = \xi(\varepsilon) - \xi^*(\varepsilon) \text{ and } \hat{\eta}(\varepsilon) = \eta(\varepsilon) - y^*\big(0, \varepsilon, \xi^*(\varepsilon)\big).$$

The only unknown function is $\xi^*(\varepsilon)$ because $y^*(0, \varepsilon, \xi^*\big(\varepsilon)\big)$ is determined as the solution of the outer problem, which of course depends implicitly on $\xi^*(\varepsilon)$. For the boundary layer solution to exist over the whole time interval it is necessary that both initial values $\hat{\xi}$ and $\hat{\eta}$ lie on a k-dimensional manifold embedded into $\mathbb{R}^{n_x+n_y}$ (see **[40]**, Lemma 2). This manifold is uniquely defined by ε and $\hat{\eta}^*(\varepsilon) \in \mathbb{R}^k$ (obtained after a transformation of $\hat{\eta}$ that enables partitioning of Y into stable and unstable components) and corresponds to stable directions of the Jacobian $\mathrm{D}_Y\, \hat{g}$ which is supposed to have k negative eigenvalues. In other words the manifold $S(\varepsilon)$ from Theorem 5 is parametrized through a linear combination of the components of $\eta(\varepsilon)$ with k terms. Thereby the points $\xi^*(\varepsilon)$ are determined. In praxis one can iteratively develop expansions for the manifold $S(\varepsilon)$ and $\xi^*(\varepsilon)$.

The coefficients in the series expansion of Theorem 5 can be computed in the following way.

LEMMA 6 (Computation of x_r^* and y_r^*). *Let $\xi^*(\varepsilon) = \sum_{r=0}^{R+1} \xi_r^* \varepsilon^r + \mathcal{O}(\varepsilon^{R+2})$ be given. The coefficients $x_r^*(t)$ and $y_r^*(t)$, $r = 0, 1, \ldots, R$ of the expansion for $x^*(t, \varepsilon)$ and $y^*(t, \varepsilon)$ from Theorem 5 are given through*

$$\begin{aligned} \mathrm{D}_t\, x_0^* &= f(x_0^*, y_0^*, 0), \quad x_0^*(0) = \xi_0^*, \\ 0 &= g(x_0^*, y_0^*, 0), \end{aligned}$$

for $r = 0$ and

$$\begin{aligned} \mathrm{D}_t\, x_r^* &= \mathrm{D}_2\, f(t, x_0^*, y_0^*, 0) x_r^* + \mathrm{D}_3\, f(x_0^*, y_0^*, 0) y_r^* + p_r(t), \quad x_r^*(0) = \xi_r^*(0) \\ \mathrm{D}_t\, y_{r-1}^* &= \mathrm{D}_2\, g(t, x_0^*, y_0^*, 0) x_r^* + \mathrm{D}_3\, g(t, x_0^*, y_0^*, 0) y_r^* + q(t), \end{aligned}$$

for $r = 1, 2, \ldots, R$, where $p_r(t)$ and $q_r(t)$ are polynomials in $x_1^*, x_2^*, \ldots, x_{r-1}^*$ and $y_1^*, y_2^*, \ldots, y_{r-1}^*$ with coefficients depending on t, x_0^* and y_0^*.

PROOF. We are seeking expansions of the form

$$x^*(t, \varepsilon) = x_0^*(t) + \sum_{r=1}^{R} x_r^*(t) \varepsilon^r \text{ and } y^*(t, \varepsilon) = y_0^*(t) + \sum_{r=1}^{R} y_r^*(t) \varepsilon^r.$$

The zeroth order coefficients are thus simply $x_0^*(t) = x^*(t, 0)$ and $y_0^*(t) = y^*(t, 0)$. Setting $\varepsilon = 0$ in (6) gives the stated equations. For the initial value we have

$$x_0^*(0) = x^*(0, 0) = \xi^*(0) = \xi_0^*.$$

The coefficients for $r = 1, 2, \ldots, R$ are obtained by iteratively differentiating both sides of (3) with respect to ε and evaluate the resulting expression at $\varepsilon = 0$.

We find

$$\begin{aligned}
&\mathrm{D}_\varepsilon^r\big(\mathrm{D}_t\, x(t,\varepsilon)\big)\big|_{\varepsilon=0} = \mathrm{D}_\varepsilon^r\, f\big(t, x(t,\varepsilon), y(t,\varepsilon),\varepsilon\big)\big|_{\varepsilon=0}\\
\Leftrightarrow\quad &\mathrm{D}_t\, x_r^* = \mathrm{D}_\varepsilon^{r-1}\Big(\big(\mathrm{D}_2\, f\, \mathrm{D}_\varepsilon\, x(t,\varepsilon)\big)\big|_{\varepsilon=0} + \big(\mathrm{D}_3\, f\, \mathrm{D}_\varepsilon\, y(t,\varepsilon)\big)\big|_{\varepsilon=0} + \mathrm{D}_4\, f\big|_{\varepsilon=0}\Big)\\
\Leftrightarrow\quad &\mathrm{D}_t\, x_r^* = \mathrm{D}_2\, f^*\, \mathrm{D}_\varepsilon^r\, x(t,\varepsilon)\big|_{\varepsilon=0} + \mathrm{D}_3\, f^*\, \mathrm{D}_\varepsilon^r\, y(t,\varepsilon)\big|_{\varepsilon=0} + p_r(t,\varepsilon)\\
\Leftrightarrow\quad &\mathrm{D}_t\, x_r^* = \mathrm{D}_2\, f^* x_r^* + \mathrm{D}_3\, f^* y_r^* + p_r(t)
\end{aligned}$$

The remainder $p_r(t)$ is a polynomial in x_r^* and y_r^*, $r = 0, 1, \ldots, r-1$ and involves higher order differentials of f evaluated at x_0^* and y_0^*. The Jacobians are $\mathrm{D}_2\, f^* := \mathrm{D}_2\, f(t, x_0^*, y_0^*, 0)$ and $\mathrm{D}_3\, f^* := \mathrm{D}_3\, f(t, x_0^*, y_0^*, 0)$, respectively.

For y_r^* the general Leibniz rule is employed and gives

$$\begin{aligned}
\mathrm{D}_\varepsilon^r\,(\varepsilon\, \mathrm{D}_t\, y(t,\varepsilon))\big|_{\varepsilon=0} &= \left(\sum_{k=0}^{r} \binom{r}{k} \mathrm{D}_\varepsilon^k\, \varepsilon\, \mathrm{D}_\varepsilon^{r-k}\, \mathrm{D}_t\, y(t,\varepsilon)\right)\Big|_{\varepsilon=0}\\
&= \big(\varepsilon\, \mathrm{D}_\varepsilon^r\, y(t,\varepsilon)\big)\big|_{\varepsilon=0} + \binom{r}{1} \mathrm{D}_\varepsilon^{r-1}\, \mathrm{D}_t\, y(t,\varepsilon)\big|_{\varepsilon=0}\\
&= r\, \mathrm{D}_t\, y_{r-1} = \mathrm{D}_2\, g^* x_r^* + \mathrm{D}_3\, g^* y_r^* + q_r(t).
\end{aligned}$$

The last equality holds by the same argument as in the x_r^* case with analog definitions for g^*. □

Remark 4. The equations for $y_r^*(t)$ are algebraic and the solvability of the equation depends on the Jacobian $g_y(t)$ which is therefore required to be non-singular. Implicitly a map $y_r^* = h_r(x_r^*, y_{r-1}^*)$, $r = 1, 2, \ldots, R$ exists where $y_0 = h_0(x_0)$ is defined through the algebraic equation of the reduced problem (5) and one can say that iteratively the fast state is parametrized by the slow state. We will come back to this view soon.

Lemma 7 (Computation of $X_r(\tau)$ and $Y_r(\tau)$). *Let $\xi^*(\varepsilon) = \sum_{r=0}^{R+1} \xi_r^* \varepsilon^r + \mathcal{O}(\varepsilon^{R+2})$ be given. For $r = 0$ the coefficients of the expansion in Theorem 5 are*

$$\begin{aligned}
&\mathrm{D}_\tau\, X_0 = 0, \quad X_0(0) = \hat{\xi}(0),\\
&\mathrm{D}_\tau\, Y_0 = \hat{g}(0, X_0, Y_0, 0), \quad Y_0(0) = \hat{\eta}(0).
\end{aligned}$$

For $r = 1, 2, \ldots, R$ the coefficients are given through

$$\begin{aligned}
&\mathrm{D}_\tau^r\, X_r = P_r(\tau), \quad X_r(0) = \hat{\xi}_r,\\
&\mathrm{D}_\tau^r\, Y_r = \mathrm{D}_2\, \hat{g}\big(0, X_0(\tau), Y_0(\tau), 0\big) X_r + \mathrm{D}_3\, \hat{g}\big(0, X_0(\tau), Y_0(\tau), 0\big) Y_r + Q_r(\tau),\\
&Y_r(0) = \hat{\eta}_r,
\end{aligned}$$

where $\hat{\xi} = \sum_{r=0}^{R} \hat{\xi}_r \varepsilon^r$, $\hat{\eta} = \sum_{r=0}^{R} \hat{\eta}_r \varepsilon^r$ and $P_r(\tau)$, $Q_r(\tau)$ are polynomials in $X_1, X_2, \ldots, X_{r-1}$ and $Y_1, Y_2, \ldots, Y_{r-1}$ with coefficients depending on τ, X_0 and Y_0.

Proof. The coefficients are obtained in a similar way to the coefficients of the outer solution, in this case by differentiating (8) iteratively with respect to ε and then set $\varepsilon = 0$. Note that the Leibniz rule is applied to the right hand side for $\mathrm{D}_\tau\, X$ and the highest order term in the sum

$$\Big(\varepsilon\big(\mathrm{D}_\varepsilon^r\, f(\varepsilon\tau, X, Y, \varepsilon)\big)\Big)\big|_{\varepsilon=0} = 0$$

and the right hand side for $\mathrm{D}_\tau\, X_r$ does not depend on X_r. □

1.1. Geometric Singular Perturbation Theory. From our viewpoint, a more interesting result is connected to the geometric approach to the singular perturbation problem. It is already indicated in Remark 4 that in the outer solution the fast variables are determined in a way by the slow variables. And as it turns out, the fast variables are confined to move on a so called *slow manifold*. This approach is known as geometric singular perturbation theory and is based on the work by Fenichel, [**21**]. Compared to the analytic approach geometric singular perturbation theory analyzes not only single trajectories but families of solutions which form manifolds in the phase space. The starting point is the manifold defined by the zero set of algebraic equation in (5). For $\varepsilon > 0$ but small enough this manifold persists and is perturbed smoothly with respect to ε.

The bulk of the presented material is based on [**42**]. Geometric singular perturbation is also concerned with systems (3) or (4), respectively, but we drop the explicit dependence of f and g on t i.e. regard right hand sides $f = f(x, y, \varepsilon)$ and $g = g(x, y, \varepsilon)$ and initial values that do not depend on ε. The slow and fast systems are thus

$$\begin{aligned} \mathrm{D}_t\, x(t) &= f(x, y, \varepsilon), \\ \varepsilon\, \mathrm{D}_t\, y(t) &= g(x, y, \varepsilon), \end{aligned} \tag{11}$$

and

$$\begin{aligned} \mathrm{D}_\tau\, x(\tau) &= \varepsilon f(x, y, \varepsilon), \\ \mathrm{D}_\tau\, y(\tau) &= g(x, y, \varepsilon) \end{aligned} \tag{12}$$

respectively.

The foundation of the geometric approach is a manifold given by $0 = g(x, y, 0)$.

Definition 8 (Critical Manifold $\mathcal{M}_0$, [**42**, Section 1.2]). Let D be a compact domain in $\mathbb{R}^{n_x}$. If there is a function $y = h_0(x)$ solving $0 = g(x, y, 0)$ on D then the n_y-dimensional critical manifold $\mathcal{M}_0$ is defined by

$$\mathcal{M}_0 = \{\begin{pmatrix} x & y \end{pmatrix}^{\mathrm{T}} \,|\, y = h_0(x),\ x \in D\}.$$

Under the assumption A3 that g_y is nonsingular the function h_0 always exists locally. The key feature of the manifold is invariance with regard to the flow in (11).

Definition 9 (Local Invariance, [**42**, Definition 2]). A set $\mathcal{M}$ is locally invariant under (11) (or (12)) if it has a neighborhood V such that for all $\xi \in \mathcal{M}$, $\phi(t, 0, \xi) \subset V$ follows that $\phi(t, 0, \xi) \subset \mathcal{M}$.

Locally invariant means thus that if $x(t)$ is on $\mathcal{M}$ it can not leave the set without also leaving the neighborhood V. A few more assumptions are needed to state the central theorem.

- A8 The set $\mathcal{M}_0$ is a compact manifold and normally hyperbolic, i.e. g_y has (exactly) n_y eigenvalues with real part unequal to 0.
- A9 The set $\mathcal{M}_0$ is given by a function $h_0(x) \in C^{R+2}$ as described in Definition 8, the domain $D \subset \mathbb{R}^{n_x}$ is compact and its boundary $\partial D \subset \mathbb{R}^{n_x - 1}$ is a smooth submanifold.

Assumption A8 can be linked to the former A7, which states that the Jacobian g_y has $1 \leq k \leq n_y$ negative and $n_y - k$ positive eigenvalues. The second assumption (A9) is made for convenience. As already stated, a function h_0 can always be found locally through the use of the implicit function theorem. If necessary such local solutions could be pieced together to form a global map. Additionally, we retain A1 which was concerned with the smoothness of f and g. Under these assumptions the following theorem can be stated.

THEOREM 10 ([**44**, Theorem 2.1], Fenichel, asymptotically stable slow manifolds). *Let A1, A8 and A9 hold. For any sufficiently small ε, there is a function h that is defined on D such that the graph*

$$\mathcal{M}_\varepsilon = \{\begin{pmatrix} x & y \end{pmatrix}^{\mathrm{T}} \,|\, y = h(x,\varepsilon), x \in D\}$$

is locally invariant under (11) *(or* (12) *respectively). The function h admits an asymptotic expansion,*

$$h(x,\varepsilon) = \sum_{r=0}^{R} h_r(x)\varepsilon^r + \mathcal{O}(\varepsilon^{R+1}).$$

REMARK 5. So far initial values did not play a role. As we mentioned we are interested in families of solutions. In geometric singular perturbation theory initial values come into play via the notion of locally invariant stable and unstable manifolds that are transverse to $\mathcal{M}_0$ and also perturb regularly and smoothly with ε. The stable manifold is the manifold $S(\varepsilon)$ from Theorem 5, i.e. solutions with initial values on it will converge to $\mathcal{M}_\varepsilon$ exponentially for $\tau \to \infty$.

The theorem expresses that the critical manifold $\mathcal{M}_0$ persists for $\varepsilon > 0$ small enough and its perturbation is regular which means $\mathcal{M}_\varepsilon$ is within $\mathcal{O}(\varepsilon)$ of $\mathcal{M}_0$ and diffeomorphic to it. Also $\mathcal{M}_\varepsilon$ is locally invariant under (3). This is a compelling result with regard to model reduction since if one can get hold of h it could be used to reduce the system to the slow variables by regarding

$$\mathrm{D}_t\, x = f\big(x, h(x,\varepsilon), \varepsilon\big).$$

The state space for the system is $\mathbb{R}^{n_x}$ compared to $\mathbb{R}^{n_x+n_y}$ for the full system.

The coefficients of the series expansions for $h(x,\varepsilon)$ from Theorem 10 can be computed algebraically.

LEMMA 11 (Computation of h_r). *The coefficients in the expansion of $h(x,\varepsilon)$ in theorem* 10 *are given by*

$$g\big(x, h_0(x), 0\big) = 0$$

for $r = 0$ and

$$\mathrm{D}_2\, g\big(x, h_0(x), 0\big) h_r(x) = \mathrm{D}_1\, h_{r-1}(x) f\big(x, h_0(x), 0\big) + q$$

for $r > 0$. The function q is a polynomial in $h_0, h_1, \ldots, h_{r-1}$.

PROOF. Differentiating $\varepsilon y = \varepsilon h(x,\varepsilon)$ with respect to t gives

$$g\big(x, h(x,\varepsilon), \varepsilon\big) = \varepsilon\, \mathrm{D}_1\, h(x,\varepsilon) f\big(x, h(x,\varepsilon), \varepsilon\big). \tag{13}$$

Now, we expand this equation asymptotically with respect to ε. For $r = 0$ this leaves us

$$g\big(x, h(x,0), 0\big) = g(x, h_0, 0) = 0.$$

For $r > 0$ we first regard the left hand member of the invariance equation (13):

$$\begin{aligned}\mathrm{D}_\varepsilon^r\, g\big(x, h(x,\varepsilon), \varepsilon\big)\big|_{\varepsilon=0} &= \mathrm{D}_\varepsilon^{r-1}\Big(\mathrm{D}_2\, g\big(x, h(x,\varepsilon), \varepsilon\big)\, \mathrm{D}_\varepsilon\, h(x,\varepsilon) + \\ &\qquad \mathrm{D}_3\, g\big(x, h(x,\varepsilon), \varepsilon\big)\Big)\big|_{\varepsilon=0} \\ &= \mathrm{D}_2\, g(x, h_0, 0) h_r + p(h_0, h_1, \ldots, h_{r-1}).\end{aligned}$$

Similarly, the right hand side can be expanded to obtain

$$\begin{aligned}\mathrm{D}_\varepsilon^r\Big(\varepsilon\, \mathrm{D}_1\, h(x,\varepsilon) f\big(x, h(x,\varepsilon), \varepsilon\big)\Big)\big|_{\varepsilon=0} &= \mathrm{D}_\varepsilon^{r-1}\Big(\mathrm{D}_1\, h(x,\varepsilon) f\big(x, h(x,\varepsilon), \varepsilon\big)\big|_{\varepsilon=0} + \\ &\qquad \varepsilon(\ldots)\big|_{\varepsilon=0}\Big) \\ &= \mathrm{D}_x\, h_{r-1}(x) f(x, h_0, 0) + \hat{q}(h_0, h_1, \ldots, h_{r-2}).\end{aligned}$$

Combining the results gives the assertion with $q = \hat{q} - p$. □

REMARK 6. The remainder terms p, $\hat{q}$, and q could be given explicitly with the help of Faà di Bruno's formula for higher order chain derivatives [**41**]. But the given abbreviated form already shows the main point: Asymptotic expansions exist and the coefficients can be computed iteratively and algebraically.

1.2. Zero Derivative Principle. We already mentioned that the manifold $\mathcal{M}_\varepsilon$ could be used for model reduction. It involves evaluating the increasingly complex algebraic equations for h_r. Therefore we present another approach to the system (11) that aims at computing the expansion of h in an algorithmic fashion that is more suited for practical purposes. To this end we regard the solution $y(t, \varepsilon)$ of (11), (12) as a function of the initial value $\xi \in \mathbb{R}^{n_x}$ for $x(t)$ and the small parameter ε. Hence we have $y = y(t, \xi, \varepsilon)$. We assume that the assumptions needed for the singular perturbation approach are fulfilled, and $y(t, \xi, \varepsilon)$ is a smooth function of all its arguments. The idea of the so called zero-derivative principle [**92, 30, 47**] is to demand for a given point $\xi \in \mathbb{R}^{n_x}$ that subsequently

$$\mathrm{D}_t^{r+1}\, y(t_0, \xi, \varepsilon) = 0, \quad r = 0, 1, \ldots, \quad t_0 \in [0, T] \text{ and } x(t_0, \varepsilon) = \xi. \tag{14}$$

holds, which implies less and less movement in the fast directions. For further analysis we shorten the notation of $y(t, \xi, \varepsilon) = h(x, \varepsilon)$ to $y(\xi, \varepsilon)$ since only autonomous systems are regarded and thus h does not depend on t and define the functional iteration

$$F_r(y^*) := y^* + \left(\mathrm{D}_t^{r+1}\, y\right)(x, y^*)$$

with fixed points $y^* = \hat{h}_r(\xi, \varepsilon)$.

THEOREM 12 ([**92**, Theorem 2.1]). *For each $r = 0, 1, \ldots$ there is an $\varepsilon_r > 0$ such that, for $\varepsilon \in (0, \varepsilon_r]$ condition (14) can be solved uniquely for y to yield an n_y-dimensional manifold $\mathcal{M}_r$ which is the graph of $\hat{h}_r$. Moreover, the asymptotic expansions of $\hat{h}_r$ and h from Theorem 10 agree up to terms of order r.*

REMARK 7. Another way to make the connection between h from Theorem 10 and $\hat{h}_r$ is to directly apply the zero-derivative principle to the second equation of (11). This corresponds to disregarding terms with a factor of ε^{r+1}. For $r = 0$ or setting $\varepsilon = 0$ we have

$$0 = g(x_0, y, \varepsilon),$$

the essential first order approximation of the slow manifold. More interestingly for $r = 1$ we find

$$\begin{aligned} \varepsilon\, \mathrm{D}_t^2\, y &= \mathrm{D}_1\, g\, \mathrm{D}_t\, x + \mathrm{D}_2\, g\, \mathrm{D}_t\, y \\ &= \mathrm{D}_1\, gf + \mathrm{D}_2\, g \frac{1}{\varepsilon} g \end{aligned}$$

which is equivalent to

$$\varepsilon^2\, \mathrm{D}_t^2\, y = \varepsilon\, \mathrm{D}_1\, gf + \mathrm{D}_2\, gg.$$

Disregarding ε^2 or setting $\mathrm{D}_t^2\, y = 0$ gives rise to the first order approximation of the manifold. This also shows another motivation for this approach: The second order time derivative of y evolves on a time scale that is $\mathcal{O}(\varepsilon^2)$.

By using the above defined fixed-point equation one can approximate the manifold $\mathcal{M}_\varepsilon$ to arbitrary order just by regarding time derivatives of y. It is much more feasible to implement the fixed-point iteration in a numerical procedure compared to solving the algebraic equations for $h_r(x)$.

There is also a relationship between the analytical approach and the manifold given by $h(x,\varepsilon)$ **[79]**. Because of the invariance of the manifold $\mathcal{M}_\varepsilon$ with respect to (11) for initial values $\big(\xi, h(\xi,\varepsilon)\big) \in \mathcal{M}_\varepsilon$ it follows that

$$\begin{pmatrix} x(t,\varepsilon) \\ y(t,\varepsilon) \end{pmatrix} = \begin{pmatrix} x(t,\varepsilon) \\ h(x(t,\varepsilon),\varepsilon) \end{pmatrix} \text{ if } x(0,\varepsilon) = \xi \text{ and } y(0,\varepsilon) = h(\xi,\varepsilon).$$

This means $y(t,\varepsilon)$ can be identified with h evaluated along $x(t,\varepsilon)$.

The outer solution $y^*(t,\varepsilon)$ approximates the fast modes on the manifold $\mathcal{M}_\varepsilon$. If we use $h(x,\varepsilon)$ for model reduction purposes this means we get an approximation of the outer solution which describes (as expected) the long term evolution of the system. No information on the fast boundary layer correction can be obtained from such a reduced system because it corresponds to the full system starting on the slow manifold and thus the boundary layer correction is zero.

REMARK 8. That using $h(x,\varepsilon)$ will approximate $x^*(t,\varepsilon)$ is indicated if one tries to compute coefficients for the expansion of $x^*(t,\varepsilon)$ for the reduced system and the full system. Let $x(t,\varepsilon)$ and $y(t,\varepsilon)$ be the solution of (11) with $x(0,\varepsilon) = \xi$ and $y(0,\varepsilon) = h(\xi,\varepsilon)$. We seek outer expansions of

$$\mathrm{D}_t\, \tilde{x}(t,\varepsilon) = f\big(\tilde{x}(t,\varepsilon), h(\tilde{x}(t,\varepsilon),\varepsilon\big), \quad \tilde{x}(0,\varepsilon) = \xi$$

and

$$\begin{aligned} \mathrm{D}_t\, x(t,\varepsilon) &= f\big(x(t,\varepsilon), y(t,\varepsilon),\varepsilon\big), \quad x(0,\varepsilon) = \xi, \\ \varepsilon\, \mathrm{D}_t\, y(t,\varepsilon) &= g\big(x(t,\varepsilon), y(t,\varepsilon),\varepsilon\big), \quad y(0,\varepsilon) = h(\xi,\varepsilon). \end{aligned}$$

For $r = 0$ we get

$$\begin{aligned} \mathrm{D}_t\, \tilde{x}_0^* &= f\big(\tilde{x}_0^*, h_0(\tilde{x}_0^*), 0\big), \quad \tilde{x}_0(0) = \xi \\ 0 &= g\big(\tilde{x}_0^*, h_0(\tilde{x}_0^*), 0\big), \end{aligned}$$

and

$$\begin{aligned} \mathrm{D}_t\, x_0^* &= f(x_0^*, y_0^*, 0), \quad x_0(0) = \xi, \\ 0 &= g(x_0^*, y_0^*, 0). \end{aligned}$$

Obviously both systems are equivalent and $x_0^*(t) = \tilde{x}_0^*(t)$.

For $r = 1$ we have

$$\begin{aligned} \mathrm{D}_t\, \tilde{x}_1^* &= \mathrm{D}_1\, f\tilde{x}_1^* + \mathrm{D}_2\, f(\mathrm{D}_1\, h_0\tilde{x}_1^* + \underbrace{\mathrm{D}_2\, h}_{=h_1}) \\ &= \mathrm{D}_1\, f\tilde{x}_1^* + \mathrm{D}_2\, f\big(\mathrm{D}_1\, h_0\tilde{x}_1^* + \mathrm{D}_2\, g^{-1}(\mathrm{D}_1\, h_0 f - \mathrm{D}_3\, g)\big). \end{aligned}$$

On the other hand we find

$$\begin{aligned} & \mathrm{D}_t\, y_0^* = \mathrm{D}_1\, gx_1^* + \mathrm{D}_2\, gy_1^* \\ \Leftrightarrow \quad & y_1^* = \mathrm{D}_2\, g^{-1}(\mathrm{D}_t\, y_0^* - \mathrm{D}_1\, gx_1^* - \mathrm{D}_3\, g), \end{aligned}$$

then noting

$$\mathrm{D}_t\, y_0 = \mathrm{D}_t\, y_0\big(x_0(t)\big) = \mathrm{D}_1\, y_0^*\, \mathrm{D}_t\, x_0^* = \mathrm{D}_1\, h_0 f$$

and (differentiating $0 = g\big(x, h(x,\varepsilon),\varepsilon\big)$ with respect to x)

$$\mathrm{D}_x\, h_0(x,\varepsilon) = -\,\mathrm{D}_2\, g^{-1}\, \mathrm{D}_1\, g$$

we obtain

$$\begin{aligned} \mathrm{D}_t\, x_1^* &= \mathrm{D}_1\, fx_1^* + \mathrm{D}_2\, fy_1^* \\ &= \mathrm{D}_1\, fx_1^* + \mathrm{D}_2\, f\big(\mathrm{D}_1\, h_0 x_1^* + \mathrm{D}_2\, g^{-1}(\mathrm{D}_1\, h_0 f - \mathrm{D}_3\, g)\big). \end{aligned}$$

which equals the above result.

We conclude this section with an example.

EXAMPLE 2. Consider

$$\begin{aligned} \mathrm{D}_t\, x &= f(x,y) = -2x - y, \quad x(0) = \xi, \\ \varepsilon\, \mathrm{D}_t\, y &= g(x,y) = -x - 2y, \quad y(0) = \eta. \end{aligned} \tag{15}$$

We are going to derive zero and first order approximations for x^*, y^*, X, Y and h. First, we notice that $\mathrm{D}_2\, g = -2$ is trivially nonsingular and has eigenvalue -2. The reduced problem

$$\begin{aligned} \mathrm{D}_t\, x &= -2x - y, \quad x(0,0) = \xi, \\ 0 &= -x - 2y \quad \Leftrightarrow \quad y = -\frac{1}{2}x \end{aligned}$$

has a solution for all $\xi \in \mathbb{R}$ because the remaining equation for x is

$$\mathrm{D}_t\, x = -\frac{3}{2}x, \quad x(0,0) = \xi.$$

Assumptions A1–A7 are fulfilled and we proceed with computing the coefficients of the various expansions.

For h_0 we have to solve the algebraic equation

$$0 = -x - 2h_0(x) \quad \Leftrightarrow \quad h_0(x) = -\frac{1}{2}x.$$

Substituting this into (15) and disregarding y corresponds to the reduced problem.

The zeroth order coefficients of the outer expansion are obtained from

$$\begin{aligned} \mathrm{D}_t\, x_0^* &= -2x_0^* - y_0^*, \quad x_0^*(0) = \xi, \\ 0 &= -x_0^* - 2y_0^* \end{aligned}$$

which gives

$$y_0^*(t) = -\frac{1}{2}x_0^*(t)$$

and

$$\mathrm{D}_t\, x_0^* = -\frac{3}{2}x_0^*, \quad x_0^*(0) = \xi \quad \Leftrightarrow \quad x_0^*(t) = \xi \mathrm{e}^{-\frac{3}{2}t}.$$

For the boundary layer correction we need to calculate initial values first. Since $X(0,\varepsilon) = x(0,\varepsilon) - x^*(0,\varepsilon)$ we get $X_0(0) = x(0,0) - x^*(0,0) = \xi - \xi = 0$ and $Y(0,\varepsilon) = y(0,\varepsilon) - y^*(0,\varepsilon)$ leads to $Y_0(0) = \eta - y_0^*(0) = \eta + \frac{1}{2}\xi$. The corresponding system on the stretched time scale $\tau = t/\varepsilon$ is

$$\begin{aligned} \mathrm{D}_\tau\, X_0 &= 0, \quad X_0(0) = 0, \\ \mathrm{D}_\tau\, Y_0 &= g\big(0, x_0^*(0) + X_0(\tau), y_0^*(0) + Y_0(\tau)\big) - g\big(0, x_0^*(0), y_0^*(0)\big) \\ &= -\big(\xi + X_0(\tau)\big) - 2\big(-\frac{1}{2}\xi + Y_0(\tau)\big) - (\xi + \frac{1}{2}\xi) \\ &= -2Y_0, \quad Y_0(0) = \eta + \tfrac{1}{2}\xi. \end{aligned}$$

Solutions are

$$X_0(\tau) = 0 \text{ and } Y_0(\tau) = (\eta + \tfrac{1}{2})\mathrm{e}^{-2\tau}.$$

Because the outer solution for $x(t,\varepsilon)$ starts at ξ the zeroth order boundary layer correction is zero. The correction Y_0 starts with the difference between the initial value of the full system η and the initial value $-\frac{1}{2}\xi$ for the outer solution y^*. Also it converges exponentially to zero for $\tau \to \infty$ as expected.

We continue with

$$-2h_1(x) = \mathrm{D}_x\, h_0(x) f\big(x, h_0(x)\big) \quad \Leftrightarrow \quad h_1(x) = -\frac{3}{8}x$$

and obtain

$$h(x,\varepsilon) = h_0(x) + h_1(x)\varepsilon + \mathcal{O}(\varepsilon^2) = -\frac{1}{2}x - \frac{3}{8}x\varepsilon + \mathcal{O}(\varepsilon^2).$$

To get x_1^* and y_1^* we have to solve the system

$$\begin{aligned} \mathrm{D}_t\, x_1^* &= \mathrm{D}_1\, f(x_0^*, y_0^*, 0)x_1^* + \mathrm{D}_2\, f(x_0^*, y_0^*, 0)y_1^* \\ &= -2x_1^* - y_1^*, \quad x_1^*(0) =?, \\ \mathrm{D}_t\, y_0^* &= \mathrm{D}_1\, g(x_0^*, y_0^*, 0)x_1^* + \mathrm{D}_2\, g(x_0^*, y_0^*, 0)y_1^* \\ \Leftrightarrow \quad \frac{3}{4}\xi \mathrm{e}^{-\frac{3}{2}t} &= -x_1^* - 2y_1^*. \end{aligned} \tag{16}$$

The last equation can be solved and

$$y_1^* = -\frac{1}{2}x_1^* - \frac{3}{8}\xi \mathrm{e}^{-\frac{3}{2}t}.$$

The initial value $x_1^*(0)$ still remains to be determined. To this end we try to match the boundary layer correction for $\tau \to \infty$ with the outer solution. In general we have $X_r(0) = \xi_r - x_r^*(0)$ and since ξ does not depend on ε and therefore $\xi_r = 0$ for $r > 0$ it follows $x_r^* = -X_r(0)$. We are forced to regard the coefficients of the inner expansion and have

$$\begin{aligned} \mathrm{D}_\tau\, X_1 =& f\big(x_0^*(0) + X_0(\tau), y_0^*(0) + Y_0(\tau)\big) - f\big(x_0^*(0), y_0^*(0)\big) \\ =& -2(\xi + 0) - \big(-\frac{1}{2}\xi + Y_0(\tau)\big) - (-2\xi + \frac{1}{2}\xi) \\ =& -(\eta + \frac{1}{2}\xi)\mathrm{e}^{-2\tau}, \quad X_1(0) = -x_1^*(0), \\ \mathrm{D}_\tau\, Y_1 =& \mathrm{D}_1\, g\big(x_0^*(0) + X_0(\tau), y_0^*(0) + Y_0(\tau)\big)X_1 + \\ & \mathrm{D}_2\, g\big(x_0^*(0) + X_0(\tau), y_0^*(0) + Y_0(\tau)\big)Y_1 \\ =& -X_1 - 2Y_1, \quad Y_1(0) = \eta_1 - y_1^*(0) = -y_1^*(0). \end{aligned}$$

We know (Theorem 5) $\lim_{\tau\to\infty} X_1(\tau) = 0$ and

$$X_1(\tau) = X_1(0) + \int_0^\tau \mathrm{D}_\tau\, X_1(\sigma)\mathrm{d}\sigma,$$

therefore we choose

$$X_1(0) = -\int_0^\infty \mathrm{D}_\tau\, X_1(\sigma)\mathrm{d}\sigma = \int_0^\infty \mathrm{e}^{-2\sigma}(\eta + \frac{1}{2}\xi)\mathrm{d}\sigma = \frac{1}{2}\eta + \frac{1}{4}\xi.$$

With $x_1^*(0) = -X_1(0)$ we can continue the outer expansion. From (16) we get

$$x_1^*(t) = \mathrm{e}^{-\frac{3}{2}t}\left(-\frac{1}{2}\eta - \frac{1}{4}\xi + \frac{3}{8}\xi t\right)$$

and

$$y_1^*(t) = \mathrm{e}^{-\frac{3}{2}t}\left(\frac{1}{4}\eta - \frac{1}{4}\xi - \frac{3}{16}\xi t\right).$$

The complete first order expansion of the outer solution is thus

$$\begin{aligned} x^*(t,\varepsilon) &= \xi \mathrm{e}^{-\frac{3}{2}t} + \mathrm{e}^{-\frac{3}{2}t}\left(-\frac{1}{2}\eta - \frac{1}{4}\xi + \frac{3}{8}\xi t\right)\varepsilon + \mathcal{O}(\varepsilon^2), \\ y^*(t,\varepsilon) &= -\frac{1}{2}\xi \mathrm{e}^{-\frac{3}{2}t} + \mathrm{e}^{-\frac{3}{2}t}\left(\frac{1}{4}\eta - \frac{1}{4}\xi - \frac{3}{16}\xi t\right)\varepsilon + \mathcal{O}(\varepsilon^2) \end{aligned}$$

Finally, with $X_1(0) = -x_1^*(0)$ and $Y_1(0) = -y_1^*(0)$ the first order inner coefficients are

$$X_1 = \mathrm{e}^{-2\tau}\left(\frac{1}{2}\eta + \frac{1}{4}\xi\right)$$

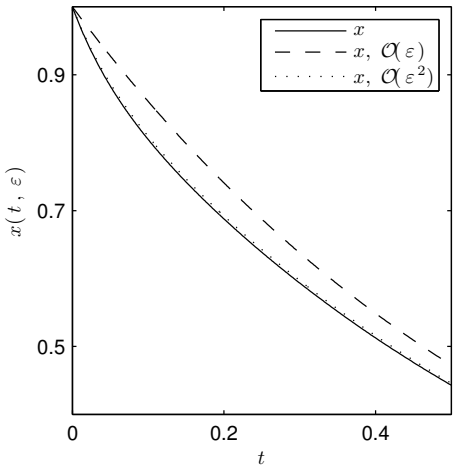

FIGURE 2.3. Approximation of $x(t,\varepsilon)$ with the full expansion, i.e. outer solution plus boundary layer correction.

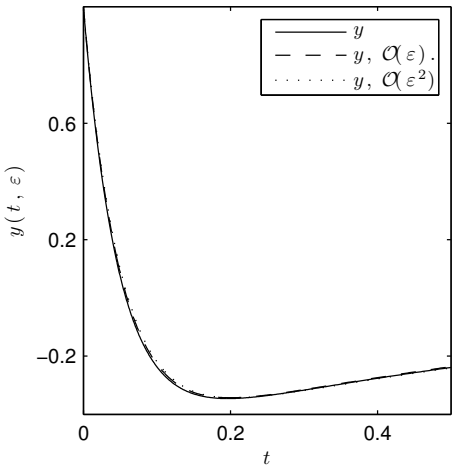

FIGURE 2.4. Approximation of $x(t,\varepsilon)$ with the full expansion, i.e. outer solution plus boundary layer correction.

and

$$Y_1 = \mathrm{e}^{-2\tau}\left(\frac{1}{4}\eta - \frac{1}{4}\xi + \frac{1}{2}\tau\right)$$

so

$$X(\tau,\varepsilon) = \mathrm{e}^{-2\tau}\left(\frac{1}{2}\eta + \frac{1}{4}\xi\right)\varepsilon + \mathcal{O}(\varepsilon^2)$$
$$Y(\tau,\varepsilon) = \mathrm{e}^{-2\tau}\left(\eta + \frac{1}{2}\xi\right) + \mathrm{e}^{-2\tau}\left(\frac{1}{4}\eta - \frac{1}{4}\xi + \frac{1}{2}\tau\right)\varepsilon + \mathcal{O}(\varepsilon^2).$$

For $\xi = \eta = 1$ and $\varepsilon = 0.1$ Figures 2.3 and 2.4 show the solutions and approximations of the slow and fast variable, respectively. The approximations are

$$x(t,\varepsilon) = x_0^*(t) + X_0(t/\varepsilon) + \big(x_1^*(t) + X_1(t/\varepsilon)\big)\varepsilon + \mathcal{O}(\varepsilon^2)$$

for $x(t,\varepsilon)$ and similarly for $y(t,\varepsilon)$. Especially for x the improvement of including the first order terms is visible. The plot of y shows also that the width of the initial correction layer is of order $\varepsilon = 0.1$.

Figures 2.5 and 2.6 show only the outer approximations. For x only the first order expansion captures the long term dynamics nicely.

Lastly, we consider the reduced system $\mathrm{D}_t\, x = f\big(x, h(x,\varepsilon)\big)$ where the zeroth or first order expansion of h is used. Figures 2.7 and 2.8 show the results. Surprisingly, the first order approximations lead to slightly worse results in the slow and fast variables. However, a good approximation could not be expected since the fast variable did not initially start on the slow manifold $\mathcal{M}_\varepsilon$. If we set $\xi = 1$ and $\eta = h_0(\xi)$ we get the result from Figures 2.9 and 2.10. In this case the first order approximation of h leads to a better approximation of x. The fast variable approaches the first order h and stays close to it while $h_0\big(x(t)\big)$ is significantly different from $y(t)$. This difference of course leads to the increasing error in the slow variable obtained from solving $\mathrm{D}_t\, x = f\big(x, h_0(x)\big)$.

First of all, the example shows that even for simple linear systems the computations to obtain the full expansion, consisting of the outer and inner expansions for slow and fast variables are rather tedious. They involve solving differential equations which in the general nonlinear case will not be feasible analytically. The

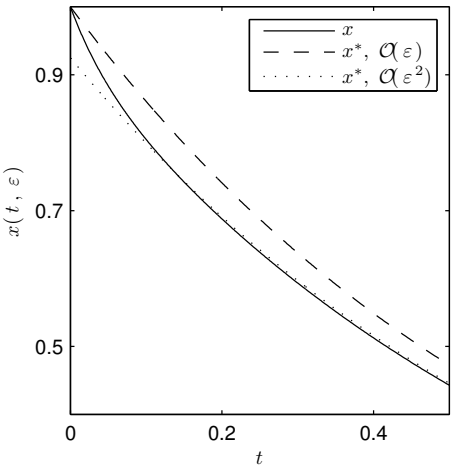

FIGURE 2.5. Approximation of $x(t,\varepsilon)$ with the outer expansion.

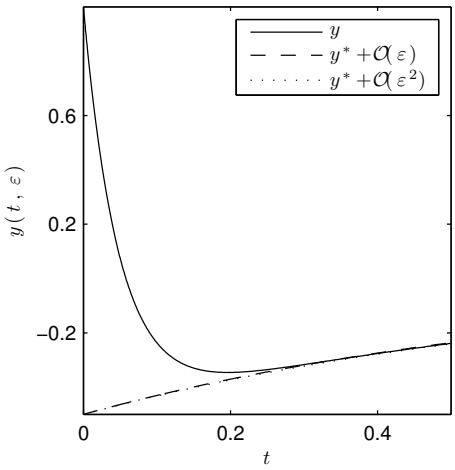

FIGURE 2.6. Approximation of $y(t,\varepsilon)$ with the outer expansion.

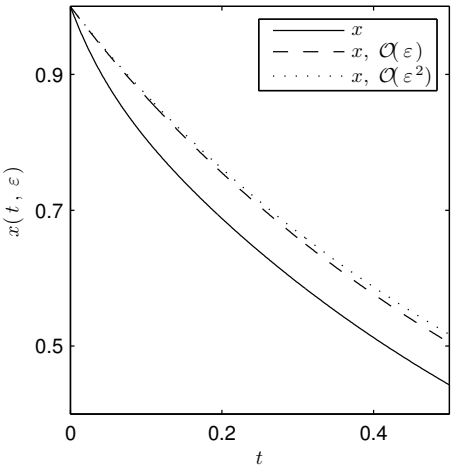

FIGURE 2.7. Approximation of $x(t,\varepsilon)$ obtained from the reduced system $\mathrm{D}_t\, x = f\big(x, h(x,\varepsilon)\big)$, $\eta = 1$.

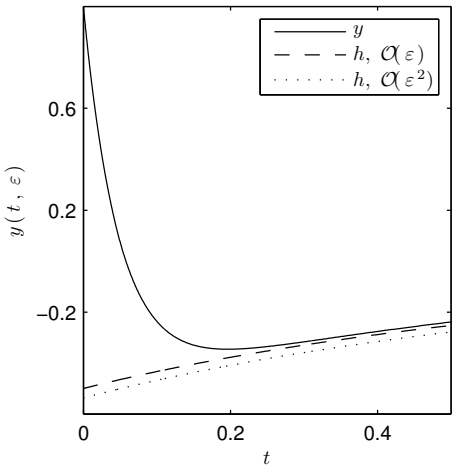

FIGURE 2.8. Approximation of $y(t,\varepsilon)$ through $y = h(x(t),\varepsilon)$, $\eta = 1$.

expansions for h can be carried out algebraically. However, we ignore the boundary layer correction completely and this can lead to significantly different results for the slow variable.

2. Singularly Perturbed Boundary Value Problems

After dealing with singular perturbed initial value problems we now turn to boundary value problems. Later they come into play when handling optimal control problems involving different time scales. They are analyzed similarly to initial value problems. Boundary layer corrections, however, now emerge at both ends of the integration interval.

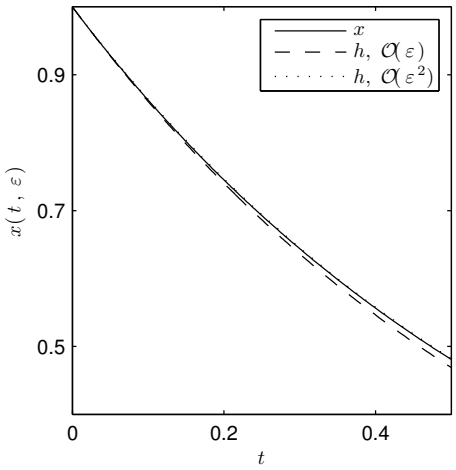

FIGURE 2.9. Approximation of $x(t,\varepsilon)$ obtained from the reduced system $\mathrm{D}_t = f\big(x, h(x,\varepsilon)\big)$, $\eta = h(\xi)$.

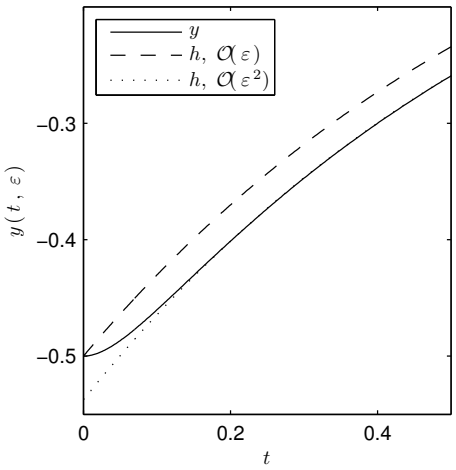

FIGURE 2.10. Approximation of $y(t,\varepsilon)$ through $y = h(x(t),\varepsilon)$, obtained from the reduced system $\mathrm{D}_t\, x = f\big(x, h(x,\varepsilon)\big)$, $\eta = h(\xi)$.

We regard the following singularly perturbed boundary value problem [**40**]:

$$\begin{aligned} \mathrm{D}_t\, x &= f(t,x,y,\varepsilon),\\ \varepsilon\, \mathrm{D}_t\, y &= g(t,x,y,\varepsilon),\\ \mathcal{L}\big(x(0),y(0),\varepsilon\big) &= 0,\\ \mathcal{R}\big(x(1),y(1),\varepsilon\big) &= 0. \end{aligned} \tag{17}$$

For general (smooth) functions $\mathcal{L}$ and $\mathcal{R}$ it is difficult to determine a reduced outer problem. This already becomes apparent when we look at the first order approximation ($\varepsilon = 0$)

$$\begin{aligned} \mathrm{D}_t\, x &= f(t,x,y,0),\\ 0 &= g(t,x,y,0),\\ \mathcal{L}\big(x(0),y(0),0\big) &= 0,\\ \mathcal{R}\big(x(1),y(1),0\big) &= 0. \end{aligned} \tag{18}$$

Here, $\mathcal{L} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \times \mathbb{R} \to \mathbb{R}^{n_\mathcal{L}}$ and $\mathcal{R} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \times \mathbb{R} \to \mathbb{R}^{n_\mathcal{R}}$, with $n_\mathcal{L} + n_\mathcal{R} = n_x + n_y$. In general the problem is overdetermined. The fast variables are subject to three algebraic equations at $t = 0$ and $t = 1$. There will be no solution unless some boundary conditions are relaxed. In the last section we removed the initial condition for y from the problem, but here $\mathcal{L}$ and $\mathcal{R}$ are nonlinear functions and general cancellation laws can not be stated. The situation is different if simple initial and end values are considered, i.e.

$$\mathcal{L}\big(x(0),y(0),\varepsilon\big) = \begin{pmatrix} \xi_\mathcal{L} \\ \eta_\mathcal{L} \end{pmatrix} \text{ and } \mathcal{R}\big(x(1),y(1),\varepsilon\big) = \begin{pmatrix} \xi_\mathcal{R} \\ \eta_\mathcal{R} \end{pmatrix}.$$

It is natural to relax the boundary conditions for y since it is expected to relax fast onto some kind of slow manifold $\mathcal{M}_\varepsilon$ in the vicinity of $\mathcal{M}_0 = \{\begin{pmatrix} x & y \end{pmatrix}^{\mathrm{T}} \,|\, 0 = g(t,x,y,0)\}$. The slow state x can still fulfill its boundary conditions since it is governed by a differential equation.

We need the following assumptions to be able to state the main result.

B1 The functions f, g, $\mathcal{L}$, and $\mathcal{R}$ from problem (17) are C^{R+2}, $R \in \mathbb{N}$ functions of their arguments on any domain of interest.

B2 For problem (18) there is at least one consistent set of boundary value conditions $\mathcal{L}^*$ and $\mathcal{R}^*$, i.e. the algebraic part of the problem has a solution. Moreover, the remaining differential-algebraic boundary value problem

$$\begin{aligned} \mathrm{D}_t\, x &= f(t, x, y, 0), \\ 0 &= g(t, x, y, 0), \\ \mathcal{L}^*\big(x(0), y(0), 0\big) &= 0, \\ \mathcal{R}^*\big(x(1), y(1), 0\big) &= 0. \end{aligned}$$

has continuous solutions $x_0(t)$ and $y_0(t)$ for $t \in [0, 1]$.

B3 The Jacobian

$$g_y(t) = \mathrm{D}_y\, g\big(t, x_0(t), y_0(t), 0\big) \in \mathbb{R}^{n_y \times n_y}, \quad t \in [0, 1]$$

is nonsingular.

B4 The Jacobian g_y has $k_{\mathcal{L}}$ eigenvalues with negative real part and $k_{\mathcal{R}}$ eigenvalues with positive real part on $0 \leq t \leq 1$ and $1 \leq k_{\mathcal{L}} + k_{\mathcal{R}} \leq n_y$.

Remark 9. The assumptions are similar to A1-A7 for the singular perturbed initial value problem. The most opaque requirement is B2 which can be hard to check depending on the problem. The last point concerns the stability of both boundary layer corrections on both sides of the interval. The right-hand side correction is stable in backward time, hence the need for eigenvalues with positive real part.

We proceed similarly to the initial value problem from the last section and state the outer problem

$$\begin{aligned} \mathrm{D}_t\, x^* &= f(t, x^*, y^*, \varepsilon), \\ \varepsilon\, \mathrm{D}_t\, y^* &= g(t, x^*, y^*, \varepsilon), \\ \mathcal{L}^*\big(x^*(0, \varepsilon), y^*(0, \varepsilon), \varepsilon\big) &= 0, \\ \mathcal{R}^*\big(x^*(1, \varepsilon), y^*(1, \varepsilon), \varepsilon\big) &= 0, \end{aligned}$$

where we demand solutions $x^*(t, \varepsilon)$ and $y^*(t, \varepsilon)$ to be continuous with respect to ε as $\varepsilon \to 0$ and converge to $x_0(t)$ and $y_0(t)$.

Next, define $X_{\mathcal{L}}(\tau, \varepsilon) = x(\varepsilon\tau, \varepsilon) - x^*(\varepsilon\tau, \varepsilon)$ and $Y_{\mathcal{L}}(\tau, \varepsilon) = x(\varepsilon\tau, \varepsilon) - x^*(\varepsilon\tau, \varepsilon)$ with right hand sides

$$\begin{aligned} \mathrm{D}_\tau\, X_{\mathcal{L}}(\tau, \varepsilon) &= -\varepsilon f\big(\varepsilon\tau, x^*(\varepsilon\tau, \varepsilon) + X_{\mathcal{L}}(\tau, \varepsilon), y^*(\varepsilon\tau, \varepsilon) + Y_{\mathcal{L}}(\tau, \varepsilon), \varepsilon\big), \\ \mathrm{D}_\tau\, Y_{\mathcal{L}}(\tau, \varepsilon) &= -\varepsilon g\big(\varepsilon\tau, x^*(\varepsilon\tau, \varepsilon) + X_{\mathcal{L}}(\tau, \varepsilon), y^*(\varepsilon\tau, \varepsilon) + Y_{\mathcal{L}}(\tau, \varepsilon), \varepsilon\big), \end{aligned}$$

and

$$X_{\mathcal{L}}(0, \varepsilon) = x(0, \varepsilon) - x^*(0, \varepsilon) \text{ and } Y_{\mathcal{L}}(0, \varepsilon) = y(0, \varepsilon) - y^*(0, \varepsilon).$$

We use the transformation $\tau = t/\varepsilon$ to zoom in into the initial boundary layer correction. For the boundary layer at the end of the time interval we introduce the new time variables $s = 1 - t$ and $\sigma = s/\varepsilon$ and deal with $X_{\mathcal{R}}(\sigma, \varepsilon) = x(1 - \varepsilon\sigma, \varepsilon) - x^*(1 - \varepsilon\sigma, \varepsilon)$ and $Y_{\mathcal{R}}(\sigma, \varepsilon) = y(1 - \varepsilon\sigma, \varepsilon) - y^*(1 - \varepsilon\sigma, \varepsilon)$. The governing equations are

$$\begin{aligned} \mathrm{D}_\tau\, X_{\mathcal{R}}(\sigma, \varepsilon) &= -\varepsilon f\big(1 - \varepsilon\sigma, x^*(1 - \varepsilon\sigma, \varepsilon) + X_{\mathcal{R}}(\sigma, \varepsilon), y^*(1 - \varepsilon\sigma, \varepsilon) + Y_{\mathcal{R}}(\sigma, \varepsilon), \varepsilon\big), \\ \mathrm{D}_\tau\, Y_{\mathcal{R}}(\sigma, \varepsilon) &= -\varepsilon g\big(1 - \varepsilon\sigma, x^*(1 - \varepsilon\sigma, \varepsilon) + X_{\mathcal{R}}(\sigma, \varepsilon), y^*(1 - \varepsilon\sigma, \varepsilon) + Y_{\mathcal{R}}(\sigma, \varepsilon), \varepsilon\big), \end{aligned}$$

with initial values

$$X_{\mathcal{R}}(0, \varepsilon) = x(1, \varepsilon) - x^*(1, \varepsilon) \text{ and } Y_{\mathcal{R}}(0, \varepsilon) = y(1, \varepsilon) - y^*(1, \varepsilon).$$

For both boundary layer solutions we demand convergence to 0 for $\tau \to \infty$ and $\sigma \to \infty$ respectively.

With all the assumptions and subproblems in place we can state the main theorem.

THEOREM 13 ([**40**, Section 2]). *Let assumptions B1–B3 hold. Then there exist an $\varepsilon_0 > 0$, a $k_{\mathcal{L}}$-dimensional manifold $S_{\mathcal{L}}(\varepsilon)$, and a $k_{\mathcal{R}}$-dimensional manifold $S_{\mathcal{R}}(\varepsilon)$ such that for each point in $S_{\mathcal{L}} \cap \{\mathcal{L}(x, y, \varepsilon) = 0\}$ and $S_{\mathcal{R}} \cap \{\mathcal{R}(x, y, \varepsilon) = 0\}$ there is a solution of* (17) *which can be represented through*

$$\begin{aligned} x(t, \varepsilon) &= x^*(t, \varepsilon) + X_{\mathcal{L}}(t/\varepsilon, \varepsilon) + X_{\mathcal{R}}(s/\varepsilon, \varepsilon) \\ y(t, \varepsilon) &= y^*(t, \varepsilon) + Y_{\mathcal{L}}(t/\varepsilon, \varepsilon) + Y_{\mathcal{R}}(s/\varepsilon, \varepsilon) \end{aligned}$$

for $t \in [0, 1]$ and $\varepsilon \in (0, \varepsilon_0)$.

The boundary layer corrections $X_{\mathcal{L}}(\tau, \varepsilon)$ and $Y_{\mathcal{L}}(\tau, \varepsilon)$ decay exponentially with $\tau \to \infty$ as do $X_{\mathcal{R}}(\sigma, \varepsilon)$ and $Y_{\mathcal{R}}(\sigma, \varepsilon)$ for $\sigma \to \infty$.

REMARK 10. All functions can be expanded into series with respect to ε. Matching the outer solutions and the boundary layer corrections has to be done on both sides of the time interval. The boundary conditions $\mathcal{L}^*$ and $\mathcal{R}^*$ have ε-expansions too and are determined through a matching condition similarly to the initial value problem case from last section.

There is also a geometrical view on boundary layer problems. Again it involves perturbations of the manifold of critical points $g(x, y, \varepsilon) = 0$, [**86**]. We will only give a very short introduction because the geometrical details are not needed later when we deal with singularly perturbed optimal control problems. Again we regard autonomous systems

$$\begin{aligned} \mathrm{D}_t\, x &= f(x, y, \varepsilon), \\ \varepsilon\, \mathrm{D}_t\, y &= g(x, y, \varepsilon) \\ \mathcal{L}\big(x(0), y(0), \varepsilon\big) &= 0 \\ \mathcal{R}\big(x(1), y(1), \varepsilon\big) &= 0. \end{aligned} \tag{19}$$

In general there might exist several domains $D_k \subset \mathbb{R}^{n_x}$, $k = 1, 2, \ldots, N$ and several slow manifolds $\mathcal{M}_\varepsilon^k$ and they can be represented by functions $h_k(x, \varepsilon)$ such that $\mathcal{M}_\varepsilon^k = \{\begin{pmatrix} x & y \end{pmatrix}^{\mathrm{T}} \mid y = h_k(x, \varepsilon)\}$. These slow invariant manifolds $\mathcal{M}_\varepsilon^k$ are an intrinsic property of the ODE system from (19) and not connected to the initial or in this case boundary values. Solutions, if they exist, will approach the slow manifolds and stay close to it for "most of the time". This "most of the time" means that there are boundary layers and maybe even interior layers of width $\mathcal{O}(\varepsilon)$. Interior layers may occur if manifolds are normally hyperbolic and the unstable manifold is intersected transversely by trajectories. The solutions "jump" from one slow manifold to another. This corresponds to heteroclinic orbits that connect the stationary points that make up the critical manifolds $\mathcal{M}_0^k$. The critical manifolds are assumed to be hyperbolic, i.e. all eigenvalues have strictly positive or strictly negative real part. And only if there are unstable directions the described jumps from one slow manifold to another might occur.

3. Summary

In this chapter we collected the basic results in singular perturbation theory with regard to initial and boundary value problems. Singular perturbation theory is concerned with problems based on ordinary differential equations where a small parameter $\varepsilon > 0$ occurs in such a way that if $\varepsilon \to 0$ the problem formulation becomes singular. In the case of initial value problems some of the initial values

can not be attained any more. One interpretation is that slow and fast processes are present in the system and slow states move on a time scale $\mathcal{O}(1)$ compared to fast states which are $\mathcal{O}(\varepsilon)$.

The system decouples for $\varepsilon \to 0$ and the slowly varying outer problem can be analyzed independently from the fast varying boundary layer correction. The solution of singularly perturbed initial value problems is continuous and differentiable with respect to the small parameter and series representations for the outer solution and boundary layer correction can be derived.

A geometric view of the problem is introduced, based on invariant manifolds in the state space of a singularly perturbed system. The critical manifold $\mathcal{M}_0$ is defined by the zero set of the fast right hand side and a function $y = h(x)$ exists and describes how the fast states are parametrized by the slow states. The critical manifold persists under perturbation with ε and again a ε-series can be derived.

Finally, boundary value problems are regarded. They are treated similarly to initial value problems, albeit now boundary layer corrections are present at both end points of the time interval.

CHAPTER 3

Mathematical Control Theory

Mathematical models are often used to describe and represent real world systems that can be controlled, i.e. systems whose state can be influenced through a certain input path. Naturally, a variety of question arises: How are such systems represented mathematically? What type of problems can be dealt with? What mathematical limitations are there with respect to states of a system that can be reached or inputs that can be applied. Can optimal inputs to reach a certain state of a system be computed? If so, how can this be done numerically? In this chapter we briefly answer some of these questions. Our focus will be on optimal control problems and numerical algorithms to solve these problems.

1. Introduction to Mathematical Control Theory

Mathematical control theory and optimal control theory are well established branches of applied mathematics. Therefore most of the material presented here is standard and can be found in many textbooks. This exposition is based on **[84]**. Other textbooks are **[61, 80, 12, 1]**.

As introduced in Chapter 1 we concentrate on dynamical systems based on ordinary differential equations. Given a time interval $\mathcal{T} = [T_0, T_1] \subset \mathbb{R}$ and a metric state space $\mathcal{X} \subset \mathbb{R}^{n_x}$ the transition map that takes a state $x(\tau) \in \mathcal{X}$, $\tau \in \mathcal{T}$ to $x(t)$, $t \in \mathcal{T}$, $t > \tau$ is provided through the initial value problem

$$\mathrm{D}_t\, x(t) = f\big(t, x(t)\big), \quad x(\tau) = \xi, \tag{20}$$

where $x : \mathbb{R} \to \mathbb{R}^{n_x}$. We saw that under certain assumptions (Lipschitz continuity of f, nonemptiness of $\mathcal{X}$ and $\mathcal{T}$, see Theorem 3) this problem has a unique solution at least in a neighborhood of τ. The system (20) can not be controlled other then by choosing the initial value ξ, because once τ and ξ are fixed the state $x(t)$ is determined. The natural extension to allow for control inputs is to represent them as functions over $\mathcal{T}$. Given a control-value or input-value set $U \subset \mathbb{R}^{n_u}$ the most liberal definition of the control space $\mathcal{U}$ consists of all functions (called controls) that map $\mathcal{T}$ onto U:

$$\mathcal{U} = \{u \,|\, u : \mathcal{T} \to U\}.$$

Later we will constrain $\mathcal{U}$ for example by demanding certain degrees of differentiability, confining U, or imposing more general path constraints on $u(t)$. The input or control u is incorporated into the right hand side function of system (20) and we have the controlled system

$$\mathrm{D}_t\, x(t) = f\big(t, x(t), u(t)\big), \quad x(\tau) = \xi. \tag{21}$$

Its solution will depend on t, τ, ξ, and now also on u and we write $\phi : \mathcal{T} \times \mathcal{T} \times \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ for the solution of the initial value problem (21) where $\phi(t, \tau, \xi, u)$ is the state the system has reached at time t starting at τ with initial value ξ and using the control $u : [\tau, t] \to U$. The domain of ϕ is abbreviated

$$\mathcal{D}_\phi := \{(t, \tau, x, u) \,|\, t, \tau \in \mathcal{T},\ t > \tau,\ x \in \mathcal{X},\ u \in \mathcal{U}\}.$$

A control system Σ is thus given by the 4-tupel $\Sigma = (\mathcal{T}, \mathcal{X}, \mathcal{U}, \phi)$ where ϕ represents the transition map and is defined at least on a nonempty subset of $\mathcal{D}_\phi$. For a solution to exist we first of all demand that U is an open subset of $\mathbb{R}^{n_u}$ and

$$f : \mathbb{R} \times \mathcal{X} \times U \to \mathbb{R}^{n_x}$$

is continuously partially differentiable with respect to all of its arguments. Moreover, we define L_U^∞, the set of all measurable, essentially bounded controls $u : \mathcal{T} \to U$. A function $u : \mathcal{T} \to U$ is essentially bounded on $\mathcal{T}$ if there exists a compact subset $K \subset U$ such that $u(t) \in K$ for almost all $t \in \mathcal{T}$ [**84**, Appendix C, Remark C.1.2]. Since U is a metric space, with

$$\mathrm{d}_\infty(u, v) := \operatorname{ess\,sup}_{t \in \mathcal{T}} |u(t) - v(t)|$$

a metric on L_U^∞ can be defined. For $U = \mathbb{R}^{n_u}$ the space is complete with respect to $\mathrm{d}_\infty(\cdot, \cdot)$. For $u \in L_U^\infty$, there exists a unique solution to problem (21) on a nonempty time interval.

THEOREM 14 ([**84**, Lemma 2.6.2]). *Let f be continuously differentiable with respect to its arguments. For any $\tau \in \mathcal{T}$, $u \in L_U^\infty$, and any $\xi \in \mathcal{X}$ there is a $t \in \mathcal{T}$, $t > \tau$ such that the initial value problem* (21) *has a unique solution $\phi(t, \tau, x, u)$ on $[\tau, t]$.*

The theorem is a straightforward extension of the usual existence theorem (Theorem 3) for initial value problems. If a solution of the initial value problem can be extended to the whole time interval for a certain control u this control is called admissible.

DEFINITION 15 (Admissible Control). Given a control system $\Sigma = (\mathcal{T}, \mathcal{X}, \mathcal{U}, \phi)$, a control $u(t) \in \mathcal{U}$ is **admissible** for $\xi \in \mathcal{X}$ if $\phi(t, \tau, x, u) \in \mathcal{X}$ is the unique solution of (21) for all $\tau, t \in \mathcal{T}$, $t > \tau$.

The concept of admissibility rejects controls that would lead to the system (21) having no solution at all or solutions outside the predefined state space $\mathcal{X}$.

Let $\mathcal{U}_a(x) \subset \mathcal{U}$ be the set of admissible controls for the initial value $x(\tau) = \xi$ for $\tau \in \mathcal{T}$. We define

$$\phi : \mathcal{X} \times \mathcal{U}_a \to \mathcal{X}, \quad \phi(x, u) := \phi(t, \tau, x, u)$$

as the unique solution of (21) for any fixed $\tau, t \in \mathcal{T}$, $t > \tau$. Then $\phi(x, u)$ is a continuous function of both of its arguments. Moreover it is even differentiable.

THEOREM 16 ([**84**, Theorem 1]). *Let $\Sigma = (\mathcal{T}, \mathcal{X}, \mathcal{U}, \phi)$ be a control system. Pick any interval $[\tau, t] \in \mathcal{T}$ and let $\phi(t) = \phi(t, \tau, x, u)$ be a solution of* (21) *on $[\tau, t]$. Consider the solution $\lambda : [\tau, t] \to \mathbb{R}^{n_x}$ of the variational equation*

$$\mathrm{D}_t \lambda(t) = \mathrm{D}_1 f\big(x(t), u(t)\big)\lambda(t) + \mathrm{D}_2 f\big(x(t), u(t)\big)\mu(t), \quad \lambda(0) = \lambda_0$$

where $\lambda_0 \in \mathbb{R}^{n_x}$ and $\mu \in L_U^\infty([\tau, t])$. Then it holds that

1. *The set*
$$\mathcal{D}_{\tau,t} = \{(x, u) \,|\, (\tau, t, x, u) \in D_\phi\}$$
is open in $\mathcal{X} \times L_U^\infty$ and $\phi(x, u)$ is continuously differentiable where the derivative with respect to u is meant to be in the Fréchet sense.
2. *The directional derivative of $\phi(x, u)$ in the direction λ_0 and μ is given by the solution of the variational equation $\lambda(t)$ with initial value λ_0.*

REMARK 11. We state another result more informally here: For any x and any admissible u let $\{u_j\}_{j=1}^\infty$ be a bounded sequence of controls, i.e. $u_j(t) \in K$ for almost all j and t where $K \subset U$ is bounded. Then

$$\lim_{j \to \infty} \phi(t, \tau, x, u_j) = \phi(t, \tau, x, u)$$

if $u_j \to u$ for $j \to \infty$ pointwise almost everywhere. This result enables us for example to approximate arbitrary controls u via piecewise constant controls to any degree of accuracy in the resulting trajectory $x(t)$.

The theorem extends the notion of sensitivity to control systems. If the system is at least continuously partially differentiable then small changes in the control will only lead to small changes in the state. Even directional derivatives can be computed which may serve as basis for gradient based algorithms that aim at finding an optimal control in some sense.

1.1. Reachability and Controllability. After introducing control systems and the basic smoothness properties we continue with the classic notion of reachability and controlability. Both terms capture the same essential question: "Given an initial state ξ which states of a system can be reached on a certain time interval?".

DEFINITION 17 (Reachability/Controllability, [**84**, Definition 3.1.1]). Let Σ be a control system. An event is a pair $(x,t) \in \mathcal{X} \times \mathcal{T}$.

- The event (z,t) can be **reached** from (x,τ) iff there exists a $u \in \mathcal{U}$ such that
$$z = \phi(t,\tau,x,u).$$
The other way around (x,τ) can be **controlled** to (z,t). We write $x \rightsquigarrow z$.
- One says that (z,t) can be **reached in time T** if there exist $\tau, t \in \mathcal{T}$, $T = t - \tau$ and $u \in \mathcal{U}$ such that $z = \phi(t,\tau,x,u)$. Equivalently (x,τ) can be **controlled to z in time T**. We write $x \underset{t}{\rightsquigarrow} z$.

Less formally we will also say z can be reached from x or x can be controlled to z without explicitly referring to the respective events.

DEFINITION 18 (Controllability, [**84**, Definition 3.1.6]). Let $\Sigma = (\mathcal{T}, \mathcal{X}, \mathcal{U}, x)$ be a control system. It is called **(completely) controllable** on $[t,\tau]$ if for each $z, x \in \mathcal{X}$ it holds that z is reachable from x. Analogously a system is said to be **controllable in time T** if z can be reached from x in time T. If x can be controlled to z regardless of the time interval, the system is just **(completely) controllable**.

For time invariant systems the following definitions for sets of reachable points are reasonable.

DEFINITION 19 (Reachable set, [**84**, Definition 3.2.1]). Let $\Sigma = (\mathcal{T}, \mathcal{X}, \mathcal{U}, \phi)$ be a time invariant system. Additionally let $t \in \mathcal{T}$ and $x \in \mathcal{X}$. The set
$$\mathcal{R}_t(x) := \{z \in \mathcal{X} \mid z = \phi(t,\tau,x,u), u \in \mathcal{U}_a\}$$
is called the **reachable set** from x in time t. The reachable set from x is
$$\mathcal{R}(x) := \bigcup_{t \in \mathcal{T}} \mathcal{R}_t.$$
Let $\mathcal{S}$ be a subset of $\mathcal{X}$ then we also define
$$\mathcal{R}_t(\mathcal{S}) := \bigcup_{x \in \mathcal{S}} \mathcal{R}_t(x)$$
and
$$\mathcal{R}(\mathcal{S}) := \bigcup_{x \in \mathcal{S}} \mathcal{R}(x).$$

Obviously, a system is completely controllable if $\mathcal{R}(x) = \mathcal{X}$ for all $x \in \mathcal{X}$. For time variant systems the set $\mathcal{R}(x)$ (or $\mathcal{R}_t(x)$ respectively) will also depend on the starting time τ and we have $\mathcal{R}(x,\tau)$ (or $\mathcal{R}_t(x,\tau)$, $t \geq \tau$ respectively).

With all the basic definitions in place the aim now is to deduce for a given system Σ if it is controllable. For systems governed by differential equations the most important part for this analysis is played by the right hand side function f. Unfortunately, as often in systems theory there is a big difference between linear and nonlinear problems. The linear case is well understood with linear, time invariant systems being the most simple class. The nonlinear case is complex: Either linearizations are used to obtain local properties, or more complex tools like Lie brackets have to be employed. Even then, proving controllability for certain systems remains a challenge.

1.1.1. *Controllability for Linear Systems.* Let us regard the simple, linear, time invariant system given by $\mathcal{T} = \mathbb{R}$, $\mathcal{X} = \mathbb{R}^{n_x}$, $U = \mathbb{R}^{n_u}$, and the transition map $\phi(t, \tau, \xi, u)$ through

$$\mathrm{D}_t\, x(t) = Ax(t) + Bu(t)$$

where $A \in \mathbb{R}^{n_x \times n_x}$ and $B \in \mathbb{R}^{n_x \times n_u}$ are matrices. We collect some basic properties of this system with regard to controllability.

LEMMA 20 ([**84**, Lemma 3.1.5, Lemma 3.1.7]). *Let Σ be a linear, time invariant system.*

1. *If $x_1 \underset{t}{\rightsquigarrow} z_1$ and $x_2 \underset{t}{\rightsquigarrow} z_2$ it follows that $(x_1 + \alpha x_2) \underset{t}{\rightsquigarrow} (z_1 + \alpha z_2)$ for all $\alpha \in \mathbb{R}$.*
2. *The system Σ is controllable in time t iff $0 \underset{t}{\rightsquigarrow} x$ for all $x \in \mathcal{X}$*
3. *The system Σ is controllable in time t iff $x \underset{t}{\rightsquigarrow} 0$ for all $x \in \mathcal{X}$.*

The first point is a natural consequence of the linearity of the system. The later points illustrate the fact that in case of linear systems reaching 0 from any state and the reverse reaching any state from 0 are equivalent. This is not generally true.

THEOREM 21 ([**84**, Corollary 3.2.7]). *Let x be a stationary point, i.e. there exists a $u \in \mathcal{U}$ such that $0 = f\big(x, u(t)\big)$ for all $t \in \mathcal{T}$. Further let $\mathcal{R}_t(x)$ be a subspace of $\mathcal{X}$ for each t then $\mathcal{R}_\varepsilon(x) = \mathcal{R}(x)$, $\varepsilon > 0$.*

The subspace requirement is fulfilled for linear systems if $x = 0$ which is not a limitation because such systems are translation invariant. The theorem states that any state that can be reached from 0 can be reached in any given time $\varepsilon > 0$. The reason is that there is no constraint on the magnitude of the control. With arbitrarily large controls all states within $\mathcal{R}(x)$ can be reached. This is of course unrealistic and we will later consider problems where U is a proper closed subset of $\mathbb{R}^{n_u}$. For now we continue with another observation that will lead the way to a general result on the controllability of time invariant linear systems.

LEMMA 22 ([**84**, Corollary 3.2.10]). *A time invariant linear system is controllable iff $\mathcal{R}(0) = \mathcal{R}_\varepsilon(0) = \mathcal{X}$, for all $\varepsilon > 0$.*

This lemma is a conclusion of Lemma 20 and Theorem 21. Finally, we present a verifiable condition for controllability of time invariant linear systems that involves the matrices A and B. Therefore we introduce the matrix

$$R = R(A, B) := [B, AB, A^2B, \ldots, A^{n-1}B]$$

where the last part means that the columns of R are the collection of the columns of the matrices in the expression. Note that higher exponents of A do not have to be considered due to the Cayley-Hamilton theorem (any matrix A fulfills its own characteristic polynomial) which states that A^{n+1} is a linear combination of A^i for $i < n$. In any event the following theorem is the main result.

THEOREM 23 ([**84**, Theorem 2]). *The linear, time invariant system* Σ *is controllable iff the rank of the controllability matrix is* n_x*, i.e.* $\operatorname{rank} R(A,B) = n_x$.

This result is also known as the Kalmann characterization of controllability. If this condition is fulfilled for a system, every state can be attained from any other state in any given time period greater than 0. Surprisingly, an algebraic condition suffices to make a statement about a continuous time, dynamical system.

EXAMPLE 3. The following example, taken from [**69**, Example 4.2] will be revisited later and serves as an illustration here. It is based on a voltage regulator with five states. Let

$$A = \begin{pmatrix} -\frac{1}{5} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & -\frac{1}{2} & \frac{8}{5} & 0 & 0 \\ 0 & 0 & -\frac{5}{7} & \frac{30}{7} & 0 \\ 0 & 0 & 0 & -\frac{5}{4} & \frac{15}{4} \\ 0 & 0 & 0 & 0 & -\frac{1}{2} \end{pmatrix} \text{ and } B = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{3}{2} \end{pmatrix}.$$

The ODE system is given by

$$M \, \mathrm{D}_t \, x = Ax + Bu.$$

where M is a diagonal matrix with entries $\operatorname{diag}(M) = (1, 1, \varepsilon, \varepsilon, \varepsilon)$ and $\varepsilon > 0$. The ε represents a small parasitic parameter. Its significance will also be made clear later. For now we choose $\varepsilon = 1$ and M can be safely ignored. Using `MATLAB` to compute $R(A,B)$ we get

$$R(A,B) = \begin{pmatrix} 0 & 0 & 0 & 0 & 19.2857 \\ 0 & 0 & 0 & 38.5714 & -114.3367 \\ 0 & 0 & 24.1071 & -59.4069 & 101.1947 \\ 0 & 5.6250 & -9.8438 & 13.7109 & -17.8418 \\ 1.5000 & -0.7500 & 0.3750 & -0.1875 & 0.0938 \end{pmatrix},$$

a triangular matrix and therefore $\operatorname{rank} R(A,B) = 5$ which means the system is fully controllable.

We already mentioned that this result is based on the fact that $U = \mathbb{R}^{n_u}$. If the controls are bounded the reachable set gets smaller. If $U \subset \mathbb{R}^{n_u}$ is a bounded neighborhood of 0, we redefine the reachable set as follows:

$$\mathcal{R}_t^U(x) := \{z \,|\, z = \phi(t, 0, x, u),\ u(t) \in U \text{ almost everywhere}\},$$

i.e. the states that can be reached from x in time t only using controls that have values in U. Analogously, $\mathcal{R}^U(x) = \bigcup_{t>0} \mathcal{R}_t^U(x)$.

THEOREM 24 ([**84**, Theorem 6]). *Let* Σ *be a time invariant system and* U *be a bounded neighborhood of* 0*. Then,* $\mathcal{R}^{\mathcal{U}}(0) = \mathbb{R}^{n_x}$ *iff*

- $\operatorname{rank} R(A,B) = n_x$*, and*
- *the matrix* A *has no eigenvalues with negative real part.*

The second condition is a rather strong one, however negative eigenvalues would imply that at least in some directions of the state space the states would tend to 0 and if the control action possible with U is too small, controllability is lost.

EXAMPLE 4. Picking up Example 3 the matrix A has eigenvalues

$$\lambda_1 = -\frac{1}{5},\ \lambda_2 = -\frac{1}{2},\ \lambda_3 = -\frac{5}{7},\ \lambda_4 = -\frac{5}{4},\ \text{and } \lambda_5 = -\frac{1}{2}.$$

All eigenvalues have negative real part so whenever $U \neq \mathbb{R}$ the system is not controllable. Note that this does not exclude certain states to be reached by constrained controls but only that not all of $\mathcal{X} = \mathbb{R}^5$ can be attained.

So far we dealt with time invariant systems. The time varying case is the next step on the ladder of complexity. In this case $A = A(t)$ and $B = B(t)$ have entries that depend on $t \in \mathcal{T}$. The smoothness of this functions plays a vital role for stating results about controllability. Two cases are considered: $A(t)$ and $B(t)$ are smooth on $\mathcal{T}$ and $A(t)$ and $B(t)$ are even analytical on $\mathcal{T}$. We extend this and speak of smooth or analytical systems Σ if the respective matrices have said properties. With $B_0 = B$ we define the recursion

$$B_{i+1} := A(t)B_i(t) - \mathrm{D}_t\, B_i(t).$$

THEOREM 25 ([**84**, Proposition 3.5.16, Corollary 3.5.18]). *Let Σ be a smooth system on $\mathcal{T}$ and pick any sub-interval $[\tau, \sigma] \subset \mathcal{T}$. If there is a $t \in [\tau, \sigma]$ and a $k \in \mathbb{N}$ such that*

$$\operatorname{rank}[B_0(t), B_1(t), \ldots, B_k(t)] = n$$

then Σ is controllable.

If Σ is even analytical then Σ is controllable on any non-empty subinterval of $\mathcal{T}$ iff for any fixed $t \in \mathcal{T}$ the rank condition holds for some $k \in \mathbb{N}$.

REMARK 12. For time invariant systems we find $B_1 = AB - 0$ and more general $B_i = A^{i-1}B - 0$ so that the rank condition is an extension to the rank condition given above.

1.1.2. *Controllability for Nonlinear Systems.* Nonlinear systems can be treated in two distinct ways: Consider linearizations along trajectories and deal with the full nonlinear system.

Local controllability is a concept based on first order information. Therefore we regard systems that are at least C^1, i.e. the right hand side f is continuously differentiable with respect to its arguments. Moreover, we introduce the equilibrium state x^*, for which holds that there is a control u^* such that $x^* = \phi(t, \tau, x^*, u^*)$ for each $t > \tau$. Again, this adds up to $f(x^*, u^*)$ being 0. The uniform distance between two functions of t on an interval $[\tau, \sigma] \subset \mathcal{T}$ is given by

$$d_\infty(x, y) := \sup\{\|x(t) - y(t)\|\,,\ t \in [\tau, \sigma]\}.$$

For general nonlinear systems (complete) controllability is a property that is hard to grasp and we have to retire to a weaker concept.

DEFINITION 26 (Local controllability, [**84**, Definition 3.7.4]). Let Σ be a system and $x(t)$ a trajectory on an interval $[\tau, \sigma] \subset \mathcal{T}$. With $x(\tau) = \xi$ and $x(\sigma) = x_1$. The system Σ is **locally controllable along $x(t)$** if for $\varepsilon > 0$ there is a $\delta > 0$ such that for each $\eta, y_1 \in \mathcal{X}$ with $\|\xi - \eta\| < \delta$ and $\|x_1 - y_1\| < \delta$ there is a trajectory $y(t)$ with $y(\tau) = \eta$ and $y(\sigma) = y_1$ and

$$d_\infty(x, y) < \varepsilon.$$

If x^* is an equilibrium state and $x(t) = x^*$ for all $t \in [\tau, \sigma]$, $T = \sigma - \tau$ then Σ is called **locally controllable** (in time T) at x^*, without reference to the trajectory.

A system is thus locally controllable along a trajectory $x(t)$ if states near the initial state ξ can be controlled to a state near the final state x_1 without deviating to much from the trajectory $x(t)$. For equilibrium states we do not rely on the trajectory since it is trivial and the definition can be reformulated using an environment $\mathcal{V} \subset \mathcal{X}$ that contains x^*, y_1 and y_2. A system is said to be locally controllable at x^* (in time T) if y_2 can be reached from y_1 (in time T) by a trajectory $y(t)$ without $y(t)$ leaving $\mathcal{V}$.

REMARK 13. The notion of local controllability is not useful for linear systems. If a linear system is controllable on an interval $[\tau, \sigma]$ it means that any state $x_1 \in \mathcal{X}$ can be reached from any initial state $\xi \in \mathcal{X}$ in any nontrivial amount of time. This

indicates also that any trajectory can be tracked arbitrary closely and thereby the definition of local controllability is fulfilled.

THEOREM 27 ([**84**, Theorem 7]). *Let Σ be a C^1 system.*

(1) *Let $x(t)$ be a trajectory to the control $u(t)$ on an interval $[\tau,\sigma] \subset \mathcal{T}$. If the linearized system with $A(t) = \mathrm{D}_x f\big(x(t),u(t)\big)$ and $B(t) = \mathrm{D}_u f\big(x(t),u(t)\big)$ is controllable on $[\tau,\sigma]$ then Σ is locally controllable along $x(t)$.*
(2) *Let Σ be time invariant and x^* an equilibrium point. For any $\varepsilon > 0$ and any $u \in \mathcal{U}$ such that $f(x^*,u) = 0$ a sufficient condition for Σ to be locally controllable at x^* in time ε is that the linearization of Σ at (x^*,u) is controllable.*

1.1.3. *Accessibility for Nonlinear Systems.* The basic idea behind controllability concepts for nonlinear systems lies in regarding the control input of the differential equation

$$\mathrm{D}_t\, x = f(x,u)$$

as constant on small time intervals. We will write $f_u(x) := f(x,u)$ for $u = \text{const}$. The result of using f_u as the right hand side for an initial value $x(0) = \xi$ will be denoted by

$$\mathrm{e}^{tf_u}\xi = \phi(t,0,\xi,u)$$

which is merely a notational device and only for linear systems with $f = A$ the matrix exponential e^{tA} represents the solution of the system. The advantage of this notation is that the result of following successive different right hand sides for different time intervals and using the result of over one interval as the initial value for the next interval can be easily written. Suppose we have f and g and start at ξ using f for one time unit and then g for two time units. The result is denoted as

$$x_1 = \mathrm{e}^{2g}\mathrm{e}^{1f}\xi$$

and means solving $\mathrm{D}_t\, x = f(x)$ on $[0,1]$ with initial value ξ and using $x(1)$ as initial value for solving $\mathrm{D}_t\, x = g(x)$ on $[0,2]$ to obtain the final result x_1. For nonlinear systems in general the order in which f and g are used will play an important role and $\mathrm{e}^{2g}\mathrm{e}^{1f}\xi \neq \mathrm{e}^{1f}\mathrm{e}^{2g}\xi$. With regard to control systems we could ask the question which states are accessible from ξ through a combination of $f_u(x)$ with $u \in U$, i.e.

$$x_1 = \mathrm{e}^{\varepsilon_k f_{u_k}}\mathrm{e}^{\varepsilon_{k-1} f_{u_{k-1}}}\cdots\mathrm{e}^{\varepsilon_1 f_{u_1}}\xi$$

with $\varepsilon_k > 0$.

To access this idea and characterize the directions one can move to from x using $f_u(x)$ we need Lie algebra techniques. To this end let $\mathcal{O} \subset \mathcal{X} \subset R^{n_x}$ be open with $x \in \mathcal{O}$. The collection of all smooth vector fields i.e. infinitely many times differentiable functions $f : \mathcal{O} \to \mathbb{R}^{n_x}$ will be denoted by $\mathbb{V}(\mathcal{O})$. Furthermore we need the space of smooth functions $g : \mathcal{O} \to \mathbb{R}$ which is denoted by $\mathbb{F}(\mathcal{O})$.

DEFINITION 28 (Lie bracket, [**84**, Definition 4.1.1]). The **Lie bracket** of $f,g \in \mathbb{V}(\mathcal{O})$ is

$$[f,g] := \mathrm{D}_*\, gf - \mathrm{D}_*\, fg \in \mathbb{V}(\mathcal{O}).$$

DEFINITION 29 (Lie algebra, [**84**, Definition 4.1.3]). A linear subspace $\mathcal{S}$ of $\mathbb{V}(\mathcal{O})$ is called **Lie algebra** if it is closed under the Lie bracket operation, i.e. $f,g \in \mathcal{S} \Rightarrow [f,g] \in \mathcal{S}$.

We call $\mathcal{A}_{\mathrm{LA}}$ the Lie algebra generated by $\mathcal{A} \subset \mathbb{V}(\mathcal{O})$ if it is the intersection of all Lie algebras that contain $\mathcal{A}$. This set is the smallest Lie algebra that contains $\mathcal{A}$ and can be generated thusly: Let $\mathcal{A}_0 = \mathcal{A}$ and then recursively:

$$\mathcal{A}_{i+1} := \{[f,g] \,|\, f,g \in \mathcal{A}_i\}, \quad i = 0,1,2,\ldots .$$

Additionally, $\mathcal{A}_\infty = \bigcup_{i\geq 0} \mathcal{A}_i$. The linear span of $\mathcal{A}_\infty$ is equal to $\mathcal{A}_{\mathrm{LA}}$. If we evaluate every vector field in $\mathcal{A}_{\mathrm{LA}}$ at x we get a subspace of $\mathbb{R}^{n_x}$ which is defined by

$$\mathcal{A}_{\mathrm{LA}}(x) := \{X(x) \,|\, X \in \mathcal{A}_{\mathrm{LA}}\}.$$

If we now regard the k-tuple $F = (f_1, f_2, \ldots, f_k)$ with elements from $\mathcal{A} \subset \mathbb{V}(\mathcal{O})$ we say that this tuple is nonsingular at $x \in \mathcal{O}$ if there exists a vector of time points $T = (t_1, t_2, \ldots, t_k)$ such that the map

$$F_F^\xi = \mathrm{e}^{t_k f_k} \cdots \mathrm{e}^{t_2 f_2} \mathrm{e}^{t_1 f_1} \xi$$

has a Jacobian of rank k at T. Note that $F_F^\xi(T)$ is a function that maps a subset of $\mathbb{R}^k$ containing 0 to $\mathcal{O}$. The rank condition implies that all points in a vicinity of ξ can be reached by means of combining the vector fields from $\mathcal{A}$ because all directions are present in the linearization.

Lemma 30 ([**84**, Lemma 4.2.8]). *If $\mathcal{A}_{\mathrm{LA}}(x) = \mathbb{R}^{n_x}$ then there exists a nonsingular n_x-tuple at x. Additionally, for $\varepsilon > 0$ there is $T \in \mathbb{R}_+^{n_x}$ such that $T < \varepsilon$ and there are $f = (f_1, f_2, \ldots, f_{n_x})$ in $\mathcal{A}$ such that the Jacobian $\mathrm{D}_* F_F^x(T)$ has rank n_x.*

Finally we will make the connection to control systems.

Definition 31 (Accessibility rank condition, [**84**, Definition 4.3.2]). Let Σ be a control system. Define

$$\mathcal{A} := \{f_u(x) \,|\, u \in U\}.$$

The Lie algebra $\mathcal{A}_{\mathrm{LA}}$ is called the accessibility Lie algebra of system Σ. The **accessibility rank condition** at ξ holds if $\mathcal{A}_{\mathrm{LA}}(\xi) = \mathbb{R}^{n_x}$.

Nonlinear controllability amounts to the statement that the set of points that can be reached from x is nonempty. Sometimes this is also coined accessibility to distinguish it from the much stronger controllability that we used earlier on. We define the set of points reachable from x without leaving a neighborhood $\mathcal{V} \subset \mathcal{X}$ of x in time t by

$$\mathcal{R}_{\mathcal{V}}^t(x) := \{x_1 \,|\, \exists u \in \mathcal{U} :\ \phi(s, 0, x, u) \in \mathcal{V}\ \forall s \in [0, t] \text{ and } x_1 = \phi(t, 0, x, u)\}.$$

Theorem 32 ([**84**, Theorem 9]). *Let Σ be a system and let the accessibility rank condition hold at x. Then for each neighborhood $\mathcal{V}$ of x and each $t > 0$*

$$\operatorname{int} \mathcal{R}_{\mathcal{V}}^t(x) \neq \emptyset.$$

Remark 14. In order to check the accessibility rank condition for a control system one has to generate n_x linearly independent (at x) vector fields through ongoing bracketing, i.e. every vector field X_i, $i = 1, 2, \ldots, n_x$ is generated through $[[\ldots[f_{u_1}, f_{u_2}], f_{u_3}], \ldots, f_{u_\ell}]$ with possibly different ℓ for each X_i. Moreover ℓ can become arbitrary large and the verification of the accessibility condition might be impossible.

Remark 15. Control-affine systems are based on differential equations of the form

$$\mathrm{D}_t\, x = g_0(x) + g_1(x)u_1 + g_2(x)u_2 + \cdots + g_{n_u}(x)u_{n_u}$$

and for them $\mathcal{A}_{\mathrm{LA}} = \{g_0, g_1, \ldots, g_{n_u}\}_{\mathrm{LA}}$. For time invariant linear systems $g_0 = Ax$ and $g_i = b_i$, $i = 1, 2, \ldots, n_u$ where b_i is the i-th column of B. We find

$$[Ax, Ax] = 0, \quad [b_i, b_j] = 0, \quad \text{and } [b_i, Ax] = Ab_i,\ i, j = 1, 2, \ldots, n_u.$$

Thus iterative brackets of the form $[[\ldots[b_i, Ax], Ax], \ldots, Ax]$ are equal to $A^{\ell-1}b_i$. All other brackets will be 0. Hence if

$$\operatorname{rank}[Ax, b_1, \ldots, b_{n_u}, Ab_1, \ldots, Ab_{n_u}, \ldots, A^{n_x-1}b_1, \ldots, A^{n_x-1}b_{n_u}] = n_x$$

at a point x the accessibility rank condition holds and is very similar to the Kalman rank condition presented above albeit weaker because it contains the additional term Ax.

We close this section with an example taken from [**68**].

EXAMPLE 5. We return to the enzyme example, see Example 1. A possible control scenario consists of assuming that substrate can be introduced into the system. And with $u(t) : U \to \mathbb{R}$, where $U \subset \mathbb{R}$ is a subset containing zero (10) becomes

$$\begin{aligned} \mathrm{D}_t\, x &= -x + (x + K - \beta)y + u(t), \quad x(0) = 1, \\ \varepsilon\, \mathrm{D}_t\, y &= x - (x + K)y, \quad y(0) = 0. \end{aligned}$$

This is a control-affine system and as pointed out in Remark 15 it suffices to regard

$$\mathcal{A} = \{g_0, g_1, \ldots, g_{n_u}\} = \left\{ \begin{pmatrix} -x + (x + K - \beta)y \\ x - (x + K)y \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}.$$

To check the accessibility rank condition we have to compute $\mathcal{A}_{\mathrm{LA}}(\xi)$ possibly through iterated bracketing as introduced after Definition 29. For $\xi = \big(x(0), y(0)\big) = (1, 0)$ it is sufficient to regard

$$\mathcal{A}_0(\xi) = \mathcal{A}(\xi) = \left\{ \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$$

because $\operatorname{rank} \mathcal{A}_0(\xi) = 2 = n_x$ already and the condition is fulfilled which implies that the accessibility region around ξ has a nonempty interior and states in a neighborhood of ξ can be reached using suitable controls.

If we choose $\xi = (0, 0)$ we find

$$\mathcal{A}_0(\xi) = \mathcal{A}(\xi) = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$$

with $\operatorname{rank} \mathcal{A}_0(\xi) = 1$ and the rank condition fails. For $\mathcal{A}_1$ we compute the nontrivial $[g_0, g_1] = -[g_1, g_0]$ and get

$$\mathrm{D}_*\, g_0 = \begin{pmatrix} -1 + y & x + K - \beta \\ 1 - y & -(x + K) \end{pmatrix}$$

and $\mathrm{D}_*\, g_1 = 0$ so

$$[g_0, g_1] = \begin{pmatrix} 1 - y \\ y - 1 \end{pmatrix}.$$

Finally,

$$(\mathcal{A}_0 \cup \mathcal{A}_1)(\xi) = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right\}$$

with $\operatorname{rank}(\mathcal{A}_0 \cup \mathcal{A}_1)(\xi) = 2$ and the rank condition is fulfilled.

2. Optimal Control

2.1. Introduction to Optimal Control. Up to this point we were interested in the question whether a given control system is susceptible to control inputs and can be steered into an arbitrary state. The question that is asked in optimal control is, how control can be achieved in an optimal fashion. Optimal in this case relates to a given cost or value functional that maps state and control into the real numbers. Optimal control problems can be viewed as infinite-dimensional optimization problems over the space of admissible controls and trajectories where

the differential equation enters the problem as a constraint. The basic problem can be formulated as

$$\begin{aligned} &\min_{x,u} J(x,u) \\ \text{s.t. } &\mathrm{D}_t\, x = f(x,u) \end{aligned} \tag{22}$$

where we have a system $\Sigma = (\mathcal{T}, \mathcal{X}, \mathcal{U}, \phi)$ and $J : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$. The formulation as a minimization problem is merely conventional. If J is to be maximized we could regard

$$\min_{x,u} -J(x,u)$$

instead.

There are two distinct approaches to problem (22), both of which aim at replacing the infinite dynamic optimization problem with a (hopefully) more accessible finite dimensional static one. In *dynamic programming* [**4**] a partial differential equation in x and t is obtained which is optimized pointwise (for (x,t)) with respect to $u(t)$ and whose solution is the optimal trajectory. The *Pontryagin minimum principle* [**74**] is based on a reformulation using Lagrange multipliers and results in a couple of necessary optimality conditions part of which define a boundary value problem.

There are many textbooks on the subject, among them [**61, 12, 1, 20**]. Much of the material covered here can be found in there in one form or the other. Our exposition is strongly based on a lecture on the subject given by Dirk Lebiedz in the winter term 2011 at Freiburg University, [**32**].

2.2. Problem Formulation. The most general optimal control problem we are going to deal with in the theoretical part of this chapter is

$$\begin{aligned} &\min_{x,u,T} \Theta\big(x(T)\big) + \int_0^T \theta\big(t, x(t), u(t)\big)\mathrm{d}t \\ \text{s.t. } &\mathrm{D}_t\, x = f(t,x,u), \\ &x(0) = \xi, \quad \psi\big(x(T)\big) = 0, \\ &u(t) \in U, \quad U \subset \mathbb{R}^{n_u}, \\ &t \in [0,T], \end{aligned} \tag{23}$$

where $f : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$, $\Theta : \mathbb{R}^{n_x} \to \mathbb{R}$, $\theta : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$, and $\psi : \mathbb{R}^{n_x} \to \mathbb{R}^r$, $0 \le r \le n$ are smooth functions of their arguments and U is nonempty and convex. The final time T can enter the problem as an optimization variable and leads to the notion of time optimal control. Problems with both, final and intermediate time costs present, i.e. Θ and θ are not equal to zero, are called *Bolza problems*. If $\Theta \equiv 0$ (23) is dubbed *Lagrange problem* and θ is called Lagrange functional accordingly. The other way around, if $\theta \equiv 0$ we speak of a *Mayer problem*. It is possible to transfer one problem formulation into another, e.g. free end-time problems can be converted to fixed end-time problems or Lagrange problems can be turned to Mayer problems. This is advantageous for theoretical and numerical purposes since in principle only one type of problem has to be considered.

Example 6. A Mayer problem with fixed end time will later serve as the basis for our numerical algorithm. To this end consider (23) and introduce the new time variable $s \in [0,1]$, defined by $t = sT$. The new variables $\tilde{x}(s) = x(sT)$ and

$\tilde{u}(s) = u(sT)$ lead to

$$\begin{aligned} &\min_{\tilde{x},\tilde{u}} \Theta\big(\tilde{x}(1)\big) + T\int_0^1 \theta\big(sT, \tilde{x}(s), \tilde{u}(s)\big)\mathrm{d}s, \\ \text{s.t. } &\mathrm{D}_s\,\tilde{x} = Tf\big(sT, \tilde{x}(s), \tilde{u}(s)\big), \\ &\tilde{x}(0) = \xi, \quad \psi\big(\tilde{x}(1)\big) = 0, \\ &\tilde{u}(s) \in U, \quad U \subset \mathbb{R}^{n_u}, \\ &s \in [0,1]. \end{aligned}$$

Some of the equations still contain T. To replace this we add the state $z(s) \equiv T$ for which

$$\mathrm{D}_s\, z(s) = 0, \quad z(0) = z(1) \text{ free.}$$

The free end-time is exchanged for free initial and final values of the new state $z(s)$. Since z appears in the objective, the initial and final value are subject to optimization. The most simple case

$$\min_{T,x,u} T$$

is converted to

$$\min_{z,x,u} z(1).$$

Now we are confronted with a fixed end time Bolza type problem. To not clutter the notation we disregard the above transformation to fixed end-time and start anew. To convert to a Mayer problem consider the additional state variable

$$z(t) = \int_0^t \theta\big(s, x(s), u(s)\big)\mathrm{d}s.$$

Differentiating with respect to t results in

$$\mathrm{D}_t\, z(t) = \theta(t, x, u), \quad z(0) = 0,$$

and with the extended state vector $\tilde{x}(t) = \big(z(t), x(t)\big)^{\mathrm{T}} \in \mathbb{R} \times \mathcal{X}$ we have the Mayer problem

$$\begin{aligned} &\min_{\tilde{x},\tilde{u}} \Theta\big(x(T)\big) + z(T)\big), \\ \text{s.t. } &\mathrm{D}_t\,\tilde{x} = \begin{pmatrix} \theta\big(t, x(t), u(t)\big) \\ f\big(t, x(t), u(t)\big) \end{pmatrix}, \\ &\tilde{x}(0) = \begin{pmatrix} 0 \\ x \end{pmatrix}, \quad \tilde{\psi}\big(\tilde{x}(T)\big) = \psi\big(x(T)\big), \\ &u(t) \in U, \quad U \subset \mathbb{R}^{n_u}, \\ &t \in [0,T], \quad T \text{ fixed.} \end{aligned}$$

2.3. Pontryagin Minimum Principle. We are going to focus on the Pontryagin minimum principle and dismiss the dynamic programming approach because the former offers more flexibility in handling additional constraints and it will again play a vital role later when dealing with optimal control of reduced models.

For notational convenience we write

$$J(u) = \Theta\big(\phi(T, 0, \xi, u)\big) + \int_0^T \theta\Big(t, \phi\big(t, 0, \xi, u(t)\big), u(t)\Big)\mathrm{d}t,$$

where ϕ is the solution of the differential equation starting at ξ using the control u.

DEFINITION 33 (Optimality, [**84**, Definition 9.2.1]). The control u^* is **optimal for** ξ if it is admissible for ξ, $\psi\big(x_{u^*}(T)\big) = 0$, and

$$J(u^*) \leq J(u), \quad \forall u \in \mathcal{U}.$$

The idea of characterizing a possible optimal solution is to view problem (23) as an infinite-dimensional constrained optimization problem where the dynamic equation is a part of the constraints. Much in analogy to the Lagrange method for finite-dimensional constrained problems Lagrange multipliers are introduced to couple the constraints and the objective functions. However, here these multipliers are functions, too.

REMARK 16 (Digression into the method of Lagrange multipliers, [**66**].). Let $f : \Omega \to \mathbb{R}$ be a function of x where $\Omega \subset \mathbb{R}^{n_x}$ is an open subset. Moreover, let $g : \Omega \to \mathbb{R}$ also be a function of x and let f and g be at least differentiable. The problem of finding $x \in \Omega$ such that

$$\begin{aligned} &\operatorname*{crit}_{x\in\Omega} f(x), \\ &\text{s.t. } g(x) = 0, \end{aligned} \tag{24}$$

where crit $f(x)$ describes the set of critical points (minimum or maximum) of $f(x)$, can be approached using the Langrange multiplier method. The key observation is:

THEOREM 34 ([**66**, Satz 7.3.6]). *For every solution x^* of* (24) *with* $\mathrm{D}_x\, g \neq 0$ *there exists a $\lambda \in \mathbb{R}$ such that*

$$\mathrm{D}_x\, f(x^*) + \lambda\, \mathrm{D}_x\, g(x^*) = 0.$$

The number λ is called Lagrange multiplier. The condition stated in the theorem can be interpreted in the way that in x^* the curve $g(x) = 0$ and $f(x) = f(x^*)$ are tangent.

In the case that several constraints $\{g_i\}_{i=1}^k$, $1 \leq k \leq n_x$ are present we consider the function $G : \Omega \to \mathbb{R}^k$, $G(x) = \big(g_1(x), g_2(x), \ldots, g_k(x)\big)^{\mathrm{T}}$. If the Jacobian $\mathrm{D}_*\, G$ has rank k then there is a vector $\lambda \in \mathbb{R}^{n_x}$ such that

$$\mathrm{D}_x\, f(x^*) + \lambda^{\mathrm{T}}\, \mathrm{D}_*\, G(x^*) = 0$$

at critical points of the Lagrange function

$$L(x, \lambda) := f(x) + \lambda^{\mathrm{T}} G(x).$$

If $\operatorname{rank} \mathrm{D}_*\, G(x^*) < k$ there is a nonzero λ such that $\lambda^{\mathrm{T}}\, \mathrm{D}_*\, G(x^*) = 0$ and x^* is a critical point of

$$L(x, \lambda_0, \lambda) := \lambda_0 f(x) + \lambda^{\mathrm{T}} G(x), \tag{25}$$

where $\lambda_0 \in \mathbb{R}$. Finally, one can summarize: There exists a $\lambda_0 \in \mathbb{R}$ and a vector $\lambda \in \mathbb{R}^k$ not both zero such that

$$\lambda_0\, \mathrm{D}_x\, f(x^*) + \lambda^{\mathrm{T}}\, \mathrm{D}_*\, G(x^*) = 0$$

at solutions x^* of

$$\begin{aligned} &\operatorname*{crit}_{x\in\Omega} f(x), \\ &\text{s.t. } G(x) = 0. \end{aligned}$$

The nonlinear equation (25) can be used as a starting point for analytical or numerical methods that aim at solving the constrained optimization problem. Note that the Lagrange method only states necessary conditions for an optimum to occur. In praxis the type of extrema has to be determined through different approaches.

Also for optimal control the objective and the constraint function, in this case the right hand side function of the ODE, are combined in a single function and a multiplier is added.

DEFINITION 35 (Hamiltonian). The function $H : \mathcal{T} \times \mathcal{X} \times U \times \mathbb{R} \times \mathbb{R}^{n_x} \to \mathbb{R}$ defined by

$$H(t, x, u, \eta_0, \eta) := \eta_0 \theta(t, x, u) + \eta^{\mathrm{T}} f(t, x, u)$$

is called **Hamiltonian** function associated with the optimal control problem (23).

For the following we only assume that f and the cost functions are C^1 in x and u. If controls are bounded it might happen that the optimal control $u^*(t)$ is not continuous.

THEOREM 36 (Pontryagin Minimum Principle, [**32**, Satz 10.1]). *Let $x^*(t)$ and $u^*(t)$ be an optimal solution of (23) and let $\mathrm{D}_* \psi\big(x^*(T)\big)$ have full rank. Then there exist $\lambda_0 \geq 0$, a piecewise continuous function $\lambda : [0, T] \to \mathbb{R}^{n_x}$, and a vector $\nu \in \mathbb{R}^r$ such that $(\lambda_0, \lambda(t), \nu) \neq 0$ for almost all $t \in [0, T]$. The following conditions apply:*

(1) *For all $t \in [0, T]$ at which $u^*(t)$ is continuous it holds that*

$$H\big(t, x^*(t), u^*(t), \lambda_0, \lambda(t)\big) = \min_{u \in U} H\big(t, x^*(t), u(t), \lambda_0, \lambda(t)\big).$$

(2) *For all $t \in [0, T]$ at which $u^*(t)$ is continuous it holds that*

$$\mathrm{D}_t \lambda(t)^{\mathrm{T}} = -\mathrm{D}_x H\big(t, x^*(t), u^*(t), \lambda_0, \lambda(t)\big).$$

*This equation is called the **adjoint** equation.*

(3) *At the final time T*

$$\lambda(T)^{\mathrm{T}} = \lambda_0 \mathrm{D}_x \Theta\big(x^*(T)\big) + \nu^{\mathrm{T}} \mathrm{D}_x \psi\big(x^*(T)\big),$$

*which is known as **transversality condition**.*

(4) *For time invariant systems with time invariant Lagrange term and fixed end time*

$$H\big(t, x^*(t), u^*(t), \lambda_0, \lambda(t)\big) = \text{const}.$$

(5) *If the end time T is free it holds at the optimal T^* that*

$$H\big(T^*, x^*(T^*), u^*(T^*), \lambda_0, \lambda(T^*)\big) = 0.$$

REMARK 17. If $\lambda_0 > 0$ (as in most cases, for example if their are no end-point constraints, i.e. $\psi\big(x(T)\big) \equiv 0$) we set $\tilde{\lambda}_0 = 1$ and rescale $\tilde{\lambda}(t) = \lambda(t)/\lambda_0$ and regard $H(t, x, u, 1, \tilde{\lambda})$.

REMARK 18. The theorem only states necessary conditions that have to be fulfilled along an optimal pair of a control $u^*(t)$ and trajectory $x^*(t)$. Only under stronger conditions they become sufficient.

THEOREM 37 ([**32**, Satz 10.3]). *Let $x^*(t)$ and $u^*(t)$ be admissible for problem (23) and fulfill the conditions of Theorem 36 with $\lambda_0 = 1$ and $\lambda(t)$ and ν accordingly. If*

(1) *ψ is affine-linear,*
(2) *Θ convex and*
(3) *$H^*(t, x, \lambda) = \min_{u \in U} H(t, x, u, \lambda_0, \lambda)$ is convex in x for every t, λ,*

then x^*, u^* are an optimal solution of problem (23).

The condition $\lambda_0 = 1$ is no restriction since under the assumptions of the theorem $\lambda_0 \neq 0$ anyway and H can be rescaled such that $\lambda_0 = 1$ holds.

The minimum principle is a central result in optimal control theory. The first condition gave the theorem its name. Note that the minimum is pointwise over the finite-dimensional space U which replaces the optimization over the infinite-dimensional space $\mathcal{U}$. The adjoint equations can be given explicitly in terms of the right hand side f and the cost function. Differentiating H with respect to x gives us

$$\mathrm{D}_x H(t, x, u, \eta_0, \eta) = \eta_0 \, \mathrm{D}_x \, \theta(t, x, u) + \eta^{\mathrm{T}} \, \mathrm{D}_x \, f(t, x, u)$$

and therefore

$$\begin{aligned} \mathrm{D}_t \, \lambda(t)^{\mathrm{T}} &= - \mathrm{D}_x \, H\big(t, x^*(t), u^*(t), \lambda_0, \lambda(t)\big) \\ &= -\lambda_0 \, \mathrm{D}_x \, \theta(t, x^*, u^*) + \lambda(t)^{\mathrm{T}} \, \mathrm{D}_x \, f(t, x^*, u^*). \end{aligned}$$

Also, one can easily see that $\mathrm{D}_t \, x = f(t, x, u) = \mathrm{D}_\lambda \, H(t, x, u, \lambda_0, \lambda)$. Assuming (an) optimal control(s) $u^*(t)$ exist we can use the minimum condition to define

$$u^*(t, x, \lambda) := \operatorname*{argmin}_{u \in U} H(t, x, u, \lambda_0, \lambda).$$

This minimum must not necessarily be unique. For the optimal control $u^*(t)$ we have

$$u^*(t) = u^*\big(t, x^*(t), \lambda(t)\big)$$

where $x^*(t)$ is the optimal trajectory. Using the rest of the statements of the minimum principle we arrive at the boundary value problem

$$\begin{aligned} \mathrm{D}_t \, x &= \mathrm{D}_\lambda \, H\Big(t, x, u^*\big(t, x^*(t), \lambda(t)\big), \lambda_0, \lambda\Big), \\ \mathrm{D}_t \, \lambda^{\mathrm{T}} &= - \mathrm{D}_x \, H\Big(t, x, u^*\big(t, x^*(t), \lambda(t)\big), \lambda_0, \lambda\Big), \\ x(0) &= \xi, \quad \psi\big(x(T)\big) = 0, \\ \lambda(T)^{\mathrm{T}} &= \lambda_0 \, \mathrm{D}_x \, \Theta\big(x^*(T)\big) + \nu^{\mathrm{T}} \, \mathrm{D}_x \, \psi\big(x^*(T)\big). \end{aligned}$$

This two-point boundary value problem serves as basis for so-called indirect methods that aim at computing an optimal control. In case $U = \mathbb{R}^{n_u}$ which we assume henceforth and H differentiable with respect to u the equation

$$0 = \mathrm{D}_u \, H\big(t, x^*(t), u, \lambda_0, \lambda(t)\big)$$

can be solved pointwise to get $u^*(t) = u^*\big(t, x^*(t), \lambda(t)\big)$, which is used in the boundary value problem.

Definition 38 (C^k Regularity of the Hamiltonian). Let $x^*(t)$ and $u^*(t)$ be the optimal solution of problem (23). If there is an $\varepsilon > 0$ such that

$$u^*(t, x, \lambda) = \operatorname*{argmin}_{u \in U} H(t, x, u, \lambda_0, \lambda)$$

is unique on the set

$$D_\varepsilon = \{(t, z, \eta) \,|\, t \in [0, T], \ \|z - x^*(t)\| < \varepsilon, \ \|\eta - \lambda(t)\| < \varepsilon\}$$

we call H **regular** (with respect to $x^*(t)$). If additionally $u^*(t, x, \lambda)$ is in C^k we call H $\boldsymbol{C^k}$**-regular**.

One consequence of C^k regularity of H is that $u^*(t)$, $x^*(t)$ and $\lambda(t)$ are also C^k functions.

Definition 39 (Legendre-Clebsch Condition). Let $x^*(t)$ and $u^*(t)$ be an optimal solution of (23) and the conditions of Theorem 36 fulfilled. If

$$\mathrm{D}_u^2 \, H\big(t, x^*(t), u^*(t), \lambda_0, \lambda(t)\big) \geq 0$$

for all $t \in [0, T]$ the optimal solution is said to satisfy a **Legendre-Clebsch condition**.

Together with the C^k regularity of the Hamiltonian the Legendre-Clebsch condition ensures that $u^*(t,x,\lambda)$ is the unique minimizing control for problem (23). Again, we take advantage of the fact that the minimization of H is pointwise over U.

If $U=[a,b]\subset\mathbb{R}$, where $a,b\in\mathbb{R}$, is a bounded interval there might be parts of the optimal control $u^*(t)$ that breach that boundary. The restriction to one control $u(t)\in\mathbb{R}$ is made to simplify the notation. The statements we are about to make carry over to the case $n_u>1$. If the Hamiltonian is C^k regular (with $k\geq 2$) this amounts to

$$u^*(t,x,\lambda)=\operatorname*{argmin}_{u\in U} H(t,x,u,\lambda_0,\lambda)$$

probably having solutions outside $[a,b]$. Let $i,j,\ell,r=1,2,\ldots$. In that case there will be inner intervals $[t_i,t_j]$, $t_i<t_j$ such that

$$a<u(t)<b,\quad t\in[t_i,t_j]$$

and boundary intervals $[t_\ell,t_r]$, $t_\ell<t_r$, and $[t_i,t_j]\cap[t_\ell,t_r]=\emptyset$ with

$$u(t)=a \text{ or } u(t)=b,\quad t\in[t_\ell,t_r].$$

The point t_ℓ is an entry-point for $u(t)$ if there is a $\varepsilon>0$ such that $u(t_\ell)=a$ or $u(t_\ell)=b$ respectively and

$$u(t)<a \text{ or } u(t)<b,\quad t\in[t_\ell-\varepsilon,t_\ell).$$

Exit points t_r where the control leaves the boundary are similarly defined. Since we assume C^k regularity it can be shown that $u(t)$ is continuous at entry and exit points and this fact can be used in analytical or numerical computations to determine those points. For example for an optimal control that takes its maximum value b at the beginning of the overall time interval $[0,T]$ one has

$$u^*(t)=\begin{cases} b & 0\leq t\leq t_1\\ u^*\big(x^*(t),\lambda(t)\big) & t_1<t\leq T.\end{cases}$$

with the additional condition that

$$u^*\big(x^*(t_1),\lambda(t_1)\big)=b$$

from which t_1 might be determined. For general nonlinear optimal problems with control constraints there is no way to determine the number of time intervals on which the control reaches the boundary.

The regularity of the Hamiltonian turned out to be a useful property, however many control systems that emerge from the simulation of real world systems are lacking regularity. An example are control systems based on

$$\mathrm{D}_t\, x=f(x)+g(x)u,\quad f:\mathbb{R}^{n_x}\to\mathbb{R}^{n_x},\ g:\mathbb{R}^{n_x}\to\mathbb{R}^{n_x\times n_u}$$

and cost function

$$\theta(x,u)=a(x)+b(x)u,\quad a:\mathbb{R}^{n_x}\to\mathbb{R},\ b:\mathbb{R}^{n_x}\to\mathbb{R}^{1\times n_u}$$

that are linear in the controls $u(t)\in U\subset\mathbb{R}^{n_u}$, where U is convex and compact. Let there be an optimal solution $x^*(t)$, $u^*(t)$ and the condition of the minimum principle fulfilled with costate $\lambda(t)$ and $\lambda_0=1$. We remove λ_0 from the list of arguments of H which is given by

$$H(x,u,\lambda)=a(x)+\lambda^{\mathrm{T}}f(x)+\big(b(x)+\lambda^{\mathrm{T}}g(x)\big)u.$$

The linearity in u transfers to the Hamiltonian.

DEFINITION 40 (Switching Function). The function $\sigma(x,\lambda) : \mathcal{X} \times \mathbb{R}^{n_x} \to \mathbb{R}$ defined by

$$\sigma(x,\lambda) := \mathrm{D}_u H(x,\lambda,u) = b(x) + \lambda^{\mathrm{T}} g$$

is called switching function. We also write $\sigma(t) := \sigma\big(x(t),\lambda(t)\big)$.

The switching function is thus the term of the Hamiltonian that is multiplied by u. Obviously minimizing the Hamiltonian pointwise with respect to $u \in U$ reduces to

$$u^*(t) = \min_{u\in U} \sigma(t)u$$

which is a linear optimization problem in u. For the sake of simplicity we set $n_u = 1$ and $U = [a,b]$ again. The optimal control structure is obviously

$$u^*(t) = \begin{cases} a, & \sigma(t) > 0 \\ \in (a,b) & \sigma(t) = 0 \\ b, & \sigma(t) < 0. \end{cases}$$

Depending on the smoothness of the involved functions, $\sigma(t)$ will at least be continuous. Still, u will have discontinuities if $\sigma(t)$ changes sign at isolated roots $\sigma(t_i) = 0$. This also means H is discontinuous and thus not regular. This is the reason why for $\sigma(t) = 0$ the value of u can not be determined via the condition $\mathrm{D}_u H = 0$ as in the regular case, because H is not differentiable.

DEFINITION 41 (Switching points, Bang-Bang Control, Singular Arc). Let the time interval $[t_1,t_2] \subset [0,T]$, $t_1 < t_2$ be given. If

(1) $\sigma(t)$ has only isolated or no roots in $[t_1,t_2]$ the optimal control $u^*(t)$ defined above is called **bang-bang** on $[t_1,t_2]$. The roots of $\sigma(t)$ are called **switching points**.
(2) $\sigma(t) \equiv 0$ on $[t_1,t_2]$ the control $u^*(t)$ is called **singular** or **singular arc** on $[t_1,t_2]$.

Of course there might be several singular arcs on the overall time interval $[0,T]$. Also, singular arcs may appear or disappear depending on T and the boundary values a and b. The value of $u^*(t)$ on a singular arc may be obtained through repeatedly differentiating $\sigma(t)$ with respect to t until it depends on u, i.e. there is a $k > 0$ such that $\mathrm{D}_u \mathrm{D}_t^k \sigma(t) \neq 0$. We define

$$\sigma^0(t) := \sigma\big(t, x(t), \lambda(t)\big) \text{ and } \sigma^{k+1}(t) := \mathrm{D}_t \sigma^k(t) = \mathrm{D}_1 \sigma^k \mathrm{D}_t x + \mathrm{D}_2 \sigma^k \mathrm{D}_t \lambda.$$

If there is a k such that $\mathrm{D}_u \sigma^k \neq 0$, and if the resulting equation can indeed be solved for u, can not be determined beforehand and depends strongly on the underlying system.

THEOREM 42 (Order of Singular Control, Generalized Legendre-Clebsch Condition, [**32**, Satz 11.4]).

(1) *Let $k^* \geq 0$ be the index for which $\mathrm{D}_u \sigma^k(t) \neq 0$ holds for the first time. Then $k^* = 2q$ is even and $q \geq 1$ is called the order of the singular control.*
(2) *Let $x^*(t)$ and $u^*(t)$ be an optimal solution with according costate $\lambda(t)$. If q is the order of the singular control it holds that*

$$(-1)^q \mathrm{D}_u \Big(\mathrm{D}_t^{2q} \mathrm{D}_u H\big(x^*(t), u^*(t), \lambda(t)\big)\Big) \geq 0.$$

The inequality is called a generalized Legendre-Clebsch condition.

We conclude this section with an example:

EXAMPLE 7. We return to the enzyme kinetic Example 5 and consider the optimal control problem

$$\begin{aligned} &\min \int_0^5 -50y + u^2 \mathrm{d}t \\ \text{subject to: } &\dot{x} = -x + (x + \frac{1}{2})y + u, \\ &\varepsilon \dot{y} = x - (x+1)y, \\ &x(0) = 1, \quad y(0) = \eta, \\ &u \in \mathbb{R}. \end{aligned}$$

The parameter values $K = \beta = 1$, the final time T, and the scaling of the objective bear no meaning and are chosen for simplicity and clarity. The objective function aims at maximizing the output of y on the time horizon while minimizing the action of the control. Assuming $\lambda_0 = 1$, we define $\lambda(t) := \big(\lambda_1(t), \lambda_2(t)\big)^{\mathrm{T}}$ and write down the Hamiltonian

$$H(x,u,\lambda) = -50y + u^2 + \lambda_1 \left(-x + \left(x + \frac{1}{2} \right) y + u \right) + \frac{\lambda_2}{\varepsilon} \left(x - (x+1)y \right).$$

The differential equations for the costate $\lambda(t)$ are given by $\mathrm{D}_{x,y} H$ and we have

$$\begin{aligned} \mathrm{D}_t \lambda_1 &= -\lambda_1(y-1) - \frac{\lambda_2}{\varepsilon}(1-y), \\ \mathrm{D}_t \lambda_2 &= -\lambda_1(x + \frac{1}{2}) - \frac{\lambda_2}{\varepsilon}(-x-1) + 50. \end{aligned}$$

The final value $\lambda(T)$ is obtained by employing the transversality condition but since there are neither final-time state constraints nor final-time costs we simply have $\lambda(T) = 0$. If we further assume C^k regularity of H with $k \geq 2$ we might try to minimize H by solving

$$\mathrm{D}_u H(x,u,\lambda) = 2u + \lambda_1 = 0 \quad \Rightarrow \quad u^* = -\frac{\lambda_1}{2}.$$

Hence $u(T) = 0$, which is common for problems with no endpoint constraints and a regularity term u^2 in the Lagrange objective. Checking the Legendre-Clebsch condition $\mathrm{D}_u^2 H = 2 > 0$ tells us that u^* is indeed a minimizer of H. Substituting $u^*(t) = \frac{\lambda_1(t)}{2}$ back into the differential equation for x we have a two point boundary value problem in x, y, λ_1, and λ_2.

Due to the nonlinearity there is no analytical solution, however, a numerical solution could be obtained with the help of a boundary value problem solver. Using the `MATLAB` solver `bvp4c` the problem with $\varepsilon = 10^{-2}$ is solved. The final objective value is -188.8. In Figure 3.1 the optimal state trajectories are shown. The costate trajectories are given in Figure 3.2. More interestingly, the optimal control is given in Figure 3.3. As expected $u^*(T) = 0$, moreover, the maximum control input is reached at $t = 0$. An interpretation with the system characteristics in mind is as follows: The reaction $\mathrm{S} + \mathrm{E}$ to SE is fast compared to the reaction from SE to $\mathrm{P} + \mathrm{E}$. The second state $y(t)$ represents this through the initial transient (fast build up) and slow degradation over time. A larger $u(t)$ towards the end of the time interval would create a large contribution to the objective value via the u^2 term without increasing the output y to much. Looking at Figure 3.1 we see that $y(t)$ is nearly constant except for t small, although $x(t)$ increases to values over 6. The high inflow of x in the beginning and its decrease over time while "waiting" for the reaction to happen seems to be a reasonable approach given the objective.

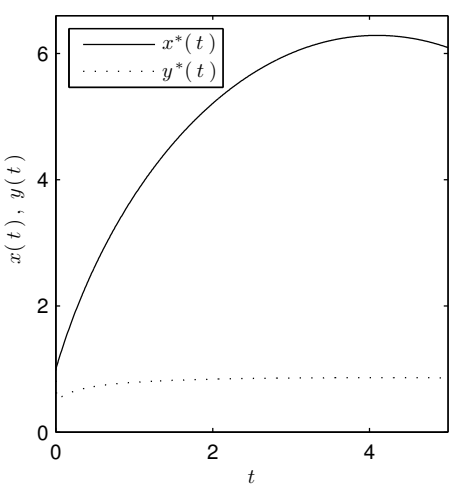

FIGURE 3.1. Optimal trajectories $x^*(t)$, $y^*(t)$ for the enzyme kinetic example with $\varepsilon = 10^{-2}$.

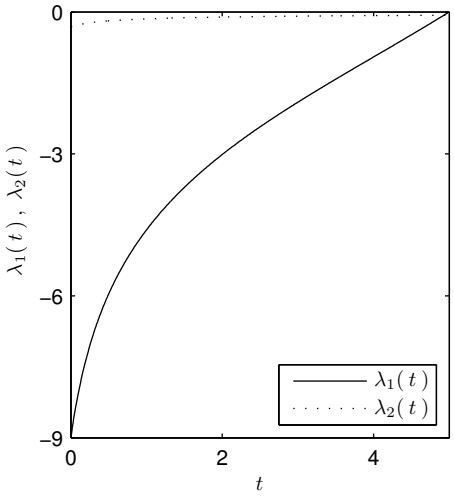

FIGURE 3.2. Costates $\lambda_1(t)$ and $\lambda_2(t)$ to the optimal trajectories for the enzyme kinetic example with $\varepsilon = 10^{-2}$.

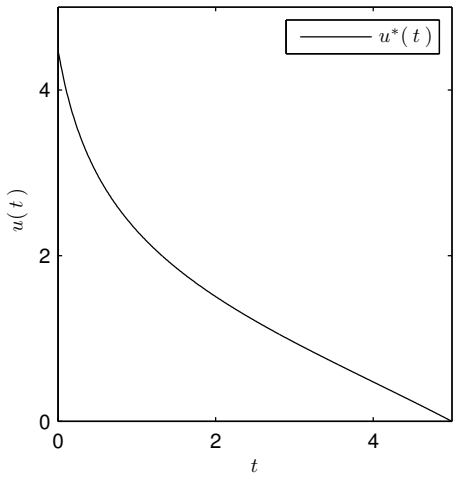

FIGURE 3.3. Optimal control $u^*(t) = \frac{\lambda_1(t)}{2}$ for the enzyme kinetic example with $\varepsilon = 10^{-2}$.

2.4. Control Theory for Two Timescale Systems. In this subsection we conclude the study of singular perturbed systems from Chapter 2 and consider singularly perturbed optimal control problems. These problems have drawn considerable attention in the past and a couple of reviews dealing with the subject have been published: **[46, 88, 70, 17]**.

All the aspects of mathematical control theory, like controllability, accessibility, and optimal control can be reviewed from the singular perturbation angle. Of course, as long as $\varepsilon > 0$ the various results continue to hold, still, exploiting the time scale structure often gives additional insight. Often, properties of the full control system can be deduced from independently analyzing the reduced fast and slow subproblems.

2.4.1. *Controllability for Linear Two Timescale Systems.* To illustrate some of the specifics of singularly perturbed control problems we regard the linear system

$$\begin{aligned} \mathrm{D}_t\, x &= A_{11}(t)x + A_{12}(t)y + B_1 u \\ \varepsilon\, \mathrm{D}_t\, y &= A_{21}(t)x + A_{22}(t)y + B_2 u \end{aligned} \tag{26}$$

where A_{11}, A_{12}, A_{21}, A_{22}, B_1, and B_2 are (possibly time varying) matrices of appropriate size. We are going to drop the argument t in favor of a compact notation. This short exposition is mainly based on [**46**]. Obviously, if $\varepsilon = 0$ we can solve the second equation if $A_{22}(t)$ is non-singular for all $t > 0$ and obtain

$$z = -A_{22}^{-1}(A_{21}x + B_2 u)$$

which gives the reduced system

$$\mathrm{D}_t\, x_0 = (A_{11} - A_{12}A_{22}^{-1}A_{21})x_0 + (B_1 - A_{22}^{-1}B_2)u$$

or zero order approximation. A system for the zeroth order boundary layer $Y(t)$ solution is obtained by looking at the difference

$$Y(t) = y - \left(-A_{22}^{-1}(A_{21}x + B_2 u)\right) = y + A_{22}^{-1}(A_{21}x + B_2 u).$$

The following theorem on the controllability of (26) is deduced after applying a transformation that decouples slow and fast motions.

THEOREM 43 (Controllability, [**46**, Theorem 3.1]). *There is an $\varepsilon_0 > 0$ such that for $\varepsilon \in (0, \varepsilon_0]$ the linear control system* (26) *is controllable (in the sense of Definition* 17*) if*

(1) *the slow subsystem*

$$\mathrm{D}_t\, x^* = (A_{11} - A_{12}A_{22}^{-1}A_{21})x^* + (B_1 + A_{12}A_{22}^{-1}B_2)u$$

is controllable and

(2) *the boundary layer controllability condition*

$$\operatorname{rank}[B_2, A_{22}B_2, \ldots, A_{22}^{-1}B_2] = n_y$$

holds for all $t \geq 0$.

In principle we have to check the controllability of the time-varying slow system and a time-invariant zeroth order fast subsystem.

REMARK 19. The conditions of the theorem are only sufficient but not necessary. They fail if the fast subsystem is controlled through the slow system.

EXAMPLE 8. We now apply the theorem to the linear problem from Example 3. It has the system matrices

$$A_{11} = \begin{pmatrix} -\frac{1}{5} & \frac{1}{2} \\ 0 & -\frac{1}{2} \end{pmatrix}, \quad A_{12} = \begin{pmatrix} 0 & 0 & 0 \\ \frac{8}{5} & 0 & 0 \end{pmatrix},$$
$$A_{21} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad A_{22} = \begin{pmatrix} -\frac{5}{7} & \frac{30}{7} & 0 \\ 0 & -\frac{5}{4} & \frac{15}{4} \\ 0 & 0 & -\frac{1}{2} \end{pmatrix}$$

and

$$B_1 = 0 \text{ as well as } B_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{3}{2} \end{pmatrix}$$

Since the slow subsystem

$$\mathrm{D}_t\, x^* = \underbrace{\begin{pmatrix} -\frac{1}{5} & \frac{1}{2} \\ 0 & -\frac{1}{2} \end{pmatrix}}_{A} x^* + \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{432}{5} \end{pmatrix}}_{B} u$$

is not time varying we have to check the rank condition from Theorem 23 and get (omitting zero columns)

$$R(A,B) = \begin{pmatrix} 0 & -\frac{216}{5} \\ -\frac{432}{5} & \frac{216}{5} \end{pmatrix}$$

which has rank $n_x = 2$ and thus the first condition of the theorem is fulfilled. The second condition requires for

$$R(B_2, A_{22}) = \begin{pmatrix} 0 & 0 & \frac{675}{28} \\ 0 & -\frac{45}{8} & -\frac{315}{32} \\ \frac{3}{2} & -\frac{3}{4} & \frac{3}{8} \end{pmatrix}$$

to have rank $n_y = 3$ which is the case. Thereby the second condition of the theorem holds and there is a ε_0 small enough such that the full system is controllable.

2.4.2. *Optimal Control.* We skip the analysis of nonlinear controllability and reachability and proceed with singularly perturbed optimal control problems. We will see that the structure of outer solutions and boundary layer corrections persists also for the optimal control and the objective function.

The general approach to nonlinear singularly perturbed optimal control problems via the Pontryagin minimum principle is to derive a singularly perturbed boundary value problem which then can be treated with methods presented in Chapter 2, especially Section 2.

To illustrate the main ideas we review the nonlinear state regulator problem [**72**], see also [**88**]:

$$\begin{aligned} &\min_u J(u) = \Theta(x(1), \varepsilon y(1), \varepsilon) + \int_0^1 \theta(x,y,u,\varepsilon)\mathrm{d}t \\ \text{subject to: } &\dot{x} = f(x,y,u,\varepsilon), \quad x(0) = \xi, \\ &\varepsilon\dot{y} = g(x,y,u,\varepsilon), \quad y(0) = \eta, \\ &\psi\big(x(T), Y(T)\big) = 0 \end{aligned} \tag{27}$$

which is analog to the general problem (23) we dealt with before. For convenience all functions are supposed to be C^∞ functions of their arguments on any domain of interest. The $\varepsilon y(1)$ in the final time cost avoids Θ to depend on the fast variable y for $\varepsilon = 0$. This enables to view $y(t)$ as additional control in the reduced problem ($\varepsilon = 0$) because then $y(t)$ is an additional function constrained by algebraic equations only.

Using the Pontryagin minimum principle (Theorem 36) with the Hamiltonian (assuming $\lambda_0 = 1$)

$$H(x, y, \lambda_x, \lambda_y, u, \varepsilon) = \theta + \lambda_x^{\mathrm{T}} f + \lambda_y^{\mathrm{T}} g$$

we get (additionally to the primal dynamic equations) the following ODE system for the adjoint variables λ_x and λ_y:

$$\begin{aligned} \mathrm{D}_t\,\lambda_x &= -\,\mathrm{D}_x\,H, \quad \lambda_x(1) = \mathrm{D}_x\,\Theta(x(1), \varepsilon y(1), \varepsilon) + \nu_x^{\mathrm{T}}\,\mathrm{D}_x\,\psi\big(x(1), y(1)\big), \\ \varepsilon\,\mathrm{D}_t\,\lambda_y &= -\,\mathrm{D}_y\,H, \quad \lambda_y(1) = \varepsilon\,\mathrm{D}_y\,\Theta(x(1), \varepsilon y(1), \varepsilon) + \nu_y^{\mathrm{T}}\,\mathrm{D}_y\,\psi\big(x(1), y(1)\big). \end{aligned} \tag{28}$$

The singular perturbation transfers to the co-state equations.

We note that there are no restrictions on the value of the control u thus $\mathrm{D}_u\,H = 0$ is a necessary condition for a minimum to occur. Moreover, we assume the strong Legendre-Clebsch condition (39), $\mathrm{D}_u^2\,H$ is positive definite, to hold. In that case a (locally) optimal control $u(t)$, that minimizes the cost functional $J(u)$ exists [**12**] for $\varepsilon > 0$. It also allows to solve $\mathrm{D}_u\,H = 0$ (locally) for $u = \omega(x, y, \lambda_x, \lambda_y, \varepsilon)$ and replace it in (27) and (28). We now have a singularly perturbed boundary value problem.

In Section 2 of Chapter 2 we dealt with such problems and saw that the main challenge with problems of this type is to determine a reasonable reduced problem,

i.e. setting $\varepsilon = 0$ in (27) and (28). In general, not all boundary values can be satisfied and some of them have to be relaxed. For $y(0)$ and $\lambda_y(1)$, the boundary values associated with the fast modes, the choice is obvious since they can not be fulfilled for $\varepsilon = 0$. Their values are determined by algebraic equations. The final state constraint $\psi\big(x(1), y(1)\big)$ is too general to be able to state simple cancellation rules. Only if they are also simple point constraints, like $x(1) = \xi_T$ and $y(1) = \eta_T$, the conditions on $y(1)$ and $\lambda_y(1)$ have to be eliminated from the problem.

Back in Section 2 of Chapter 2 we stated a couple of assumptions for a singularly perturbed boundary value problem to have a continuous solution (with respect to $\varepsilon \to 0$): B1–B4. If these assumptions hold for the problem compiled above, Theorem 13 tells us that series solutions for x, y, λ_x, and λ_y exist and that boundary layer corrections emerge at both ends of the time interval. If one would plug in these representations into $u = \omega(x, y, \lambda_x, \lambda_y, \varepsilon)$ and further $J(u)$ they can also be developed into ε-series. Thus the optimal control u^* is a combination of two fast varying boundary layer corrections and slow varying outer solution.

The assumptions B1–B4 can be refined for control problems.

B1' The reduced problem

$$\begin{aligned}
\mathrm{D}_t\, x &= f(x, y, \omega, 0), \quad x(0) = \xi,\\
\mathrm{D}_t\, \lambda_x &= -\mathrm{D}_x\, H(x, y, \lambda_x, \lambda_y, \omega, 0),\\
\lambda_x(1) &= \mathrm{D}_x\, \Theta(x(1), \varepsilon y(1), \varepsilon) + \nu_x^{\mathrm{T}}\, \mathrm{D}_x\, \psi\big(x(1), y(1)\big),\\
0 &= \mathrm{D}_{\lambda_y}\, H(x, y, \lambda_x, \lambda_y, \omega, 0) = g(x, y, \omega, 0),\\
0 &= -\mathrm{D}_y\, H(x, y, \lambda_x, \lambda_y, \omega, 0),
\end{aligned}$$

has a unique solution $x^0(t)$, $y^0(t)$, $\lambda_x^0(t)$, and $\lambda_y^0(t)$.

B2' The Jacobian

$$H_{y,\lambda_y} := \begin{pmatrix} \mathrm{D}_{y\lambda_y}\, H + \mathrm{D}_{\lambda_y\lambda_y}\, H \\ -\mathrm{D}_{yy}\, H - \mathrm{D}_{y\lambda_y}\, H \end{pmatrix} \in \mathbb{R}^{2n_y \times 2n_y}$$

evaluated along $x^0(t)$, $y^0(t)$, $\lambda_x^0(t)$, and $\lambda_y^0(t)$ has no purely imaginary eigenvalues, moreover we require it to have exactly n_y eigenvalues with positive and n_y eigenvalues with negative real part.

The assumption B1' corresponds to B2 whereas B2' is equivalent to B3 and B4.

The Jacobian H_{y,λ_y} can be analyzed in more detail. To this end we define $z(t) = \big(y(t), \lambda_y(t)\big)^{\mathrm{T}}$ and further $\omega(z) = \omega(x, y, \lambda_x, \lambda_y, 0)$, $H(z, u) = H(x, y, \lambda_x, \lambda_y, u, 0)$, and

$$E = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}, \quad I \in \mathbb{R}^{n_y \times n_y},$$

so the algebraic equations from B1' can be rewritten as

$$0 = E\, \mathrm{D}_1\, H\big(z, \omega(z)\big).$$

The Jacobian H_{y,λ_y} is thus given by

$$\mathrm{D}_z\, E\, \mathrm{D}_1\, H\big(z, \omega(z)\big) = E\Big(\mathrm{D}_{11}\, H\big(z, \omega(z)\big) + \mathrm{D}_{21}\, H\big(z, \omega(z)\big)\, \mathrm{D}_z\, \omega(z)\Big).$$

We notice that differentiating the optimality condition $\mathrm{D}_u\, H\big(z, \omega(z)\big) = 0$ with respect to z gives

$$\begin{aligned}
&\mathrm{D}_{22}\, H\big(z, \omega(z)\big)\, \mathrm{D}_z\, \omega(z) + \mathrm{D}_{12}\, H\big(z, \omega(z)\big) = 0\\
&\qquad\qquad \Leftrightarrow \quad \mathrm{D}_z\, \omega(z) = -\,\mathrm{D}_{22}\, H\big(z, \omega(z)\big)^{-1}\, \mathrm{D}_{12}\, H\big(z, \omega(z)\big).
\end{aligned}$$

Finally, $\mathrm{D}_{21}\, H\big(z,\omega(z)\big) = \mathrm{D}_{12}\, H\big(z,\omega(z)\big)^{\mathrm{T}}$ and

$$H_{y,\lambda_y} = E\Big(\underbrace{\mathrm{D}_{11}\, H\big(z,\omega(z)\big) - \mathrm{D}_{12}\, H\big(z,\omega(z)\big)^{\mathrm{T}}\, \mathrm{D}_{22}\, H\big(z,\omega(z)\big)^{-1}\, \mathrm{D}_{12}\, H\big(z,\omega(z)\big)}_{H_z}\Big).$$

The inner matrix H_z is real symmetric since $\mathrm{D}_{11}\, H$ as well as $\mathrm{D}_{12}\, H^{\mathrm{T}}\, \mathrm{D}_{22}\, H\, \mathrm{D}_{12}\, H$ are symmetric ($\mathrm{D}_{22}\, H$ is symmetric). From the symmetry we have that H_z has a full set of eigenvalues λ_i, $i = 1, 2, \ldots, n_y$ and the multiplication with E leads to H_{y,λ_y} having $2n_y$ eigenvalues $\pm\lambda_i$. The second part of condition B2' is therefore automatically fulfilled if all eigenvalues of H_{y,λ_y} have nonzero real part. Interestingly, for optimal control problems the partitioning of the eigenvalues is a consequence of the co-state condition of the minimum principle. In other words, singularly perturbed boundary value problems derived from optimal control problems are well behaved.

If B1' and B2' hold, the solution of the full problem converges to the solution of the reduced problem for $\varepsilon \to 0$ and the following series representations can be stated [**72, 40**]:

$$\begin{aligned} x(t,\varepsilon) &= x^*(t,\varepsilon) + X_L(t/\varepsilon,\varepsilon) + X_R(s/\varepsilon,\varepsilon), \\ y(t,\varepsilon) &= y^*(t,\varepsilon) + Y_L(t/\varepsilon,\varepsilon) + Y_R(s/\varepsilon,\varepsilon), \\ u(t,\varepsilon) &= u^*(t,\varepsilon) + U_L(t/\varepsilon,\varepsilon) + U_R(s/\varepsilon,\varepsilon), \\ J(u) &= J^*(t,\varepsilon) + J_L(t/\varepsilon,\varepsilon) + J_R(s/\varepsilon,\varepsilon), \end{aligned} \tag{29}$$

with $s = 1 - t$. Boundary layer corrections emerge at both ends of the time interval and appropriate series representations can be found for all right-hand-side terms in (29). The eigenvalue condition on H_{y,λ_y} is necessary for the stability of the left and right hand boundary layer corrections to be stable in forward and backward time, respectively.

REMARK 20. We only considered one of the most simple nonlinear singularly perturbed optimal control problems. Specifically we excluded constrained and Mayer problems, problems with singular Hessians, and time optimal control. In this cases often only examples can be treated completely since the convergence of the full problem to the reduced problem for $\varepsilon \to 0$ can not be guaranteed for a general problem class and for the asymptotic development into ε-series the involved Hamiltonians and related boundary value problems are not smooth enough. For examples and further analysis see [**88, 17**].

REMARK 21. Another approach to obtain ε-series expansion is the so-called *direct scheme* [**5**]. The idea is to formally plug in the expansions (29) into the control problem (27) and collect same order terms separately for the three time scales t, t/ε, and s/T. For each order of approximation, i.e. ε^k, $k = 0, 1, 2, \ldots$, three optimal problems are obtained: One for the outer solution and one for each boundary layer correction.

The main difference lies in the fact that the expansion in ε is performed before applying the minimum principle. One can show that both approaches, i.e. the direct scheme and the boundary value problem solution are the same if the control problem can be expanded into ε asymptotically at all.

So far we only talked about the classic analytical approach to singularly perturbed optimal control problems, involving expansions of outer solutions and boundary layer corrections. We are now focusing on the geometric point of view (see Section 1.1). There we saw that the fast variable is restricted to a slow manifold given by $y = h(x,\varepsilon)$. For control problems the right hand side functions also depend on the control input u. We define the new critical manifold (compare Definition 8)

DEFINITION 44 (Critical Manifold $\mathcal{M}_0$ for control systems). Let D be a compact domain in $\mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$. If there is a function $y = h_0(x, u)$ solving $0 = g(x, y, u, 0)$ on D then by

$$\mathcal{M}_0 = \{\begin{pmatrix} x & y & u \end{pmatrix}^{\mathrm{T}} \mid y = h_0(x, u),\ (x, u) \in D\}$$

the n_y-dimensional **critical manifold** $\mathcal{M}_0$ is defined.

If additionally the assumptions A8 and A9 (see page 12) are accordingly fulfilled for $(x, u) \in D$ then there is a function $h(x, u, \varepsilon)$ such that

$$\mathcal{M}_\varepsilon = \{\begin{pmatrix} x & y & u \end{pmatrix}^{\mathrm{T}} \mid y = h(x, u, \varepsilon),\ (x, u) \in D\}$$

is (locally) invariant under the flux created by

$$\begin{aligned} \mathrm{D}_t\, x &= f(x, y, u, \varepsilon) \\ \varepsilon\, \mathrm{D}_t\, y &= g(x, y, u, \varepsilon). \end{aligned}$$

The function $h(x, u, \varepsilon)$ can be expanded into an ε-series with u being a pointwise input. We now consider the reduced optimal control problem

$$\begin{aligned} \min_u J(u) &= \Theta(x(1), \varepsilon) + \int_0^1 \theta(x, h(x, u), u, \varepsilon)\mathrm{d}t \\ \text{subject to: } \dot{x} &= f(x, h(x, u, \varepsilon), u, \varepsilon), \quad x(0) = \xi. \end{aligned}$$

Tackling singularly perturbed optimal control problems with analytical methods is not an easy task. General and applicable statements, especially about nonlinear systems are rare. Even if all assumptions are fulfilled, the asymptotic expansions and the matching of outer solutions and boundary layer corrections is a tedious undertaking. The more important are numerical methods, to which we come in the next section.

3. Numerical Methods

Numerical methods to solve optimal control problems can be broadly divided into two major categories: Direct and indirect methods. Indirect methods consist of using either the dynamic programming approach or the minimum principle to derive an equivalent problem and then use numerical algorithms. Deducing the boundary value problem and solving it with `MATLAB` in Example 7 was prototypical for that approach. Direct methods aim at solving the optimal control problem via directly (hence the name) discretizing $u(t)$ and the dynamic equations for the state and thereby obtaining a finite-dimensional optimization problem which then can be solved numerically. The difference between those two methods is often stated as *first optimize, then discretize* (indirect) versus *first discretize, then optimize* (direct). Both approaches rely on robust and efficient numerical optimization algorithms (in the indirect approach to solve the BVP), so that the "backend" is often the same. We will concentrate on direct methods and then give a short introduction into constrained numerical optimization.

Optimal control problems arising from real world systems often feature additional parameters, that are subject to optimization, general objectives, and more complex constraints and do not fit into the setting proposed by (23). For example, initial values for the states might also be given by an equation of the form $\vartheta\big(x(0)\big) = 0$. More general, states and controls can be subject to nonlinear path (in)equality constraints, i.e. there are functions

$$\begin{aligned} v(x(t), u(t), T, p) &= 0 \\ w(x(t), u(t), T, p) &\leq 0 \end{aligned}$$

where $v : \mathcal{X} \times U \times \mathcal{T} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_v}$, $w : \mathcal{X} \times U \times \mathcal{T} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_w}$, and $p \in \mathbb{R}^{n_p}$ are additional parameters and are also featured in the right hand side function and objective. The problem we aim to solve is

$$
\begin{aligned}
&\min_{x,u,T,p} J(t,x,u,p,T) = \Theta(T,x(T),u(T),p) + \int_0^T \theta(t,x(t),u(t),p)\mathrm{d}t \\
&\text{s.t. } \mathrm{D}_t\, x(t) = f(t,x(t),u(t),p), \\
&\quad v(t,x(t),u(t),T,p) = 0, \\
&\quad w(t,x(t),u(t),T,p) \leq 0, \\
&\quad t \in [0,T].
\end{aligned} \tag{30}
$$

All functions are supposed to be at least twice continuously differentiable.

Dealing with this problem theoretically is quite challenging. One central approach is to extend the minimum principle to obtain necessary conditions [**38**]. See also [**65**] for a very general form. Extensions of the minimum principle are proven for various different settings, i.e. the concrete form of the constraints allowed. In general, one has to assume that the constraints v and w along an optimal solution fulfill constraint qualification conditions involving rank conditions on the Jacobians of v and w with respect to u and/or x. Besides the Hamiltonian

$$H(t,x,u,\lambda_0,\lambda) = \lambda_0 \Theta(t,x,u,p) + \lambda f(t,x,u,p)$$

additional multipliers μ, ν and the Lagrangian

$$L(t,x,u,\lambda_0,\lambda,\mu,\nu) = H(t,x,u,\lambda_0,\lambda) + \mu v(t,x,u,T,p) + \nu w(t,x,u,T,p)$$

are introduced. Optimal solutions are then necessarily critical points of the Lagrangian with multipliers not equal to zero. A final value problem for $\lambda(t)$ is obtained via $\mathrm{D}_t\, \lambda = \mathrm{D}_x\, L^*$ and a transversality condition on $\lambda(T)$. We are not stating more details here and proceed with numerical methods that are used to solve the problem.

3.1. Multiple Shooting. The ultimate goal of direct methods is to derive a finite dimensional nonlinear program that can be solved numerically. This means in essence discretization of the functions involved in the problem, i.e. eventually regarding the functions only on finitely many time points $t_i \in [0,T]$, $i = 1,2,\ldots$. There are several methods available to achieve this.

Single shooting [**15**] consists of parametrizing $u(t)$ with finitely many parameters such that

$$u(t) = u(t,\alpha), \quad \alpha \in \mathbb{R}^{n_\alpha}$$

where $u(t,\alpha)$ is completely defined by α for each $t \in [0,T]$. A common example are piecewise constant controls. To this end the parameters would be fixed time points t_i, $i = 1,2,\ldots,n_t$, $0 = t_1 < t_2 < \cdots < t_{n_w} = T$ and constants $u_i \in U$, $i = 1,2,\ldots,n_t - 1$ such that

$$u(t) = u_i \in U \quad t \in [t_i, t_{i+1}),\ i = 1,2,\ldots,n_t - 1.$$

Remark 22. The parametrization for u is arbitrary as long as the number of parameters needed is finite. Common are the mentioned piecewise constant approximations and higher order (piecewise) polynomials, e.g. cubic splines that could be used to ensure a differentiable control solution. Depending on the problem at hand other approaches are possible, for example

$$u(t) = u(t,\alpha) = \sum_{j=-n_\alpha}^{n_\alpha} \alpha_j \mathrm{e}^{2\pi j i t}, \quad t \in [0,1],$$

where $u(t)$ is approximated through a finite Fourier sum and the parameters are the amplitudes of the involved harmonic functions.

The initial value problem $\mathrm{D}_t\, x(t) = f(t, x(t), u(t), p),\ x(0) = \xi$ on $[0, T]$ is replaced by $\mathrm{D}_t\, x(t) = f(t, x(t), u(t, \alpha), p),\ x(0) = \xi$ and can be solved for admissible α by using an appropriate numerical initial value solver that computes $x(T) = \phi(T, 0, \xi, \alpha)$. Problem (30) is replaced with

$$
\begin{aligned}
&\min_{x,\alpha,p,T} J(t, x(t, x), u(t, \alpha), p, T)\\
\text{s. t. }\ &\mathrm{D}_t\, x(t) = f(t, x(t), u(t, \alpha), p),\\
&v(0, x(0), u(0, \alpha), T, p) = 0,\\
&w(0, x(0), u(0, \alpha), T, p) \le 0,\\
&v(t, x(T), u(T, \alpha), T, p) = 0,\\
&w(t, x(T), u(T, \alpha), T, p) \le 0,\\
&t \in [0, T].
\end{aligned}
\tag{31}
$$

Note that the path constraints can only be checked for $t = 0$ and $t = T$. Problem (31) is an optimization problem over a finite dimensional search space and the result is an initial value ξ^*, control parameters α^*, an optimal final time T^*, and optimal parameters p^*. If the solution of the finite problem is a good approximation of the solution of the infinite problem depends critically on the parametrization of u. Moreover, it turns out that the nonlinear optimization problem (31) is often numerically unstable depending on the sensitivity of the differential equation with respect to ξ and α. If the solution of the initial value problem is (numerically) unstable for certain input α the relatively long integration from $[0, T]$ will lead to difficulties.

Among other things these stability issues lead to development of multiple shooting **[9, 15, 85]**. The idea is to divide the time domain of interest $[0, T]$ into n_t subintervals $0 = t_1 < t_2 < \cdots < t_{n_t} = T$ where t_i might be free. At this time points, also called multiple shooting nodes we introduce the approximate solution $x_i \approx x(t_i)$. The control is replaced by a parametrized version, however, parameters might vary on each multiple shooting interval, meaning we have

$$
u(t) = \begin{cases} u_i(t, \alpha_i) & t \in [t_i, t_{i+1}),\ i = 1, 2, \ldots, n_t - 1,\\ u_i(t, \alpha_{n_t}) & t = T,\\ 0 & \text{else,} \end{cases}
$$

where each $\alpha_i \in \mathbb{R}^{n_\alpha}$. For example if again piecewise constant controls are desired each α_i would be a single real number. The numerical solution on the i-th multiple shooting interval can now be obtained by integrating the initial value problem

$$
\mathrm{D}_t\, x(t) = f(t, x(t), u_i(t, \alpha_i), p), \quad x(t_i) = \xi_i,\ t \in [t_i, t_{i+1}],\ i = 1, 2, \ldots, n_t - 1,
$$

which is completely defined if t_i, t_{i+1}, α_i, and ξ_i are known. Let us denote the solution of this problem by $x_i(t)$, $i = 1, 2, \ldots, n_t - 1$ and let $x_{n_t}(t) = x_{n_t}$. To obtain a continuous solution for the state over the whole time interval we add the following continuity constraints to the problem:

$$
x_i(t_{i+1}) = x_{i+1}(t_{i+1}) = \xi_{i+1}, \quad i = 1, 2, \ldots, n_t - 1.
$$

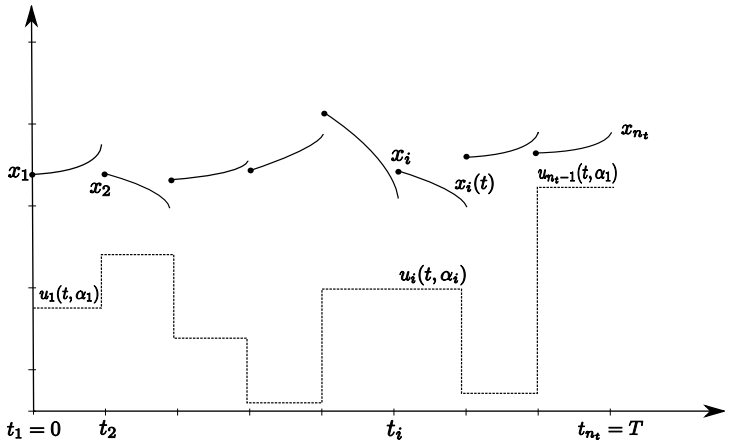

FIGURE 3.4. Multiple shooting.

Note that $x_1(t_1) = x_1(0)$ is (possibly) already constrained in the original problem (30) as is $x_{n_t-1}(t_{n_t}) = x_{n_t-1}(T) = x_{n_t}$. Finally, we arrive at

$$
\begin{aligned}
&\min_{\xi_i,\alpha_i,p,t_i,T} \sum_{i=1}^{n_t} J(t, x_i(t), u_i(t,\alpha_i), p, T) \\
\text{s. t. } & \mathrm{D}_t\, x_i(t) = f(t, x_i(t), u_i(t,\alpha_i), p), \quad i = 1, 2, \ldots, n_t - 1, \\
& x_i(t_i) = \xi_i, \quad i = 1, 2, \ldots, n_t - 1, \\
& x_i(t_{i+1}) = x_{i+1}, \quad i = 1, 2, \ldots, n_t - 1, \\
& v(t_i, x_i(t_i), u_i(t_i,\alpha_i), T, p) = 0, \quad i = 1, 2, \ldots, n_t, \\
& w(t_i, x_i(t_i), u(t_i,\alpha_i), T, p), \leq 0, \quad i = 1, 2, \ldots, n_t, \\
& \sum_{i=1}^{n_t-1} t_{i+1} - t_i = T, \\
& 0 = t_0 < t_1 < t_2 < \cdots < t_{n_t} = T, \\
& g_v(t_1, t_2, \ldots, t_{n_t}, T) = 0, \\
& g_w(t_1, t_2, \ldots, t_{n_t}, T) \leq 0.
\end{aligned}
\tag{32}
$$

REMARK 23. The last four constraints are dealing with the multiple shooting nodes. The first condition ensures that the final time T is reached if the length of all multiple shooting intervals is summed up. In the next condition the strict inequalities are necessary because in the case $t_i = t_{i+1}$ and $u_{i-1}(t, \alpha_{i-1}) = u_i(t, \alpha_i) = u_{i+1}(t, \alpha_{i+1})$, $i = 1, 2, \ldots, n_t - 2$, $t \in [t_{i-1}, t_{i+2}]$ the position of $t_i = t_{i+1}$ would be arbitrary within $[t_{i-1}, t_{i+2}]$ which, depending on the concrete implementation, might lead to failures in the optimization because $t_i = t_{i+1}$ is undetermined. The time constraint g_v and g_w are introduced to implement additional constraints on the position of the multiple shooting nodes, e.g. only allowing an equidistant grid of time points or a minimal length for the multiple shooting intervals. They do not have to be present.

Problem (32) is a finite nonlinear program because all optimization variables are finite-dimensional. Figure 3.4 serves as illustration of the approach. In the state of the algorithm depicted, the continuity conditions for x are not yet fulfilled. In comparison to the single-shooting approach the number of optimization variables has increased as has the number of constraints. This might increase the computation time. However, the aim of multiple shooting is not so much speed but stability.

The initial value problems only have to be solved on the comparatively short multiple shooting intervals. The granularity of the discretization is easily adjustable to achieve a stable solution or to adopt a control structure (e.g. switching points) that are inherited from the underlying control problem. The path constraints defined by v and w can only be guaranteed to be fulfilled at the multiple shooting nodes. This is still better than in single shooting where without extension path constraints can not be handled.

For a successful numerical solution of (32) two main ingredients are needed: First, a numerical integration routine to solve the initial value problems that preferably is also able to compute derivatives of the solution of an ODE with respect to initial values and parameters. And secondly, an NLP solver.

3.1.1. *Numerical Solution of Initial Value Problems and Automatic Differentiation.* Consider the initial value problem

$$\mathrm{D}_t\, x(t) = f(t, x(t), p), \quad x(\tau) = \xi, \quad t \in [\tau, \sigma], \tag{33}$$

with $f : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_x}$, $\sigma > \tau$. Let the right hand side be at least twice continuously differentiable with respect to x and p. The initial value problems resulting from the multiple shooting approach fit into this setting by subsuming the control function parameters α_i and the parameter vector p into the "new" parameter vector p in problem (33). We are not only interested in the solution $x(\sigma)$ but also in the derivatives $\mathrm{D}_\xi\, x(\sigma)$, $\mathrm{D}_\sigma\, x(\sigma)$ and $\mathrm{D}_p\, x(\sigma)$. The later derivatives are often called parameter sensitivities of the ODE (33). To facilitate the computation of the sensitivity of the solution with respect to σ we introduce

$$\Delta t = \sigma - \tau$$

and each time point $t \in [\tau, \sigma]$ can now be represented through

$$t = \tau + \Delta t \tau, \quad \tau \in [0, 1].$$

Using this equation we transform (33) to the new time scale $\tau \in [0, 1]$ and have

$$\mathrm{D}_\tau\, x(\tau) = \Delta t f(\tau + \Delta t \tau, x(\tau), p), \quad x(0) = x, \quad \tau \in [0, 1].$$

abusing the notation and reusing x for the new state variable on the new time scale. The result of the transformation is that the length of the time interval Δt is an explicit parameter of the right hand side function and the derivative $\mathrm{D}_{\Delta t}\, x(1)$ can be treated the same way as $\mathrm{D}_p\, x(1)$. Note that in the context of multiple shooting the transformation also has to be applied to the control function as well as the time constraint functions. For convenience we regard the new problem

$$\mathrm{D}_t\, x(t) = f(t, x(t), p), \quad x(0) = \xi, \quad t \in [0, 1], \tag{34}$$

We proceed with describing a *backward differentiation formula* (BDF) based integration routine which is employed in the numerical tool that is used to solve our optimal control problems **[37, 15]**. BDF methods are linear multistep methods that are particularly suited to tackle stiff problems. We introduce the time grid $t_j = jh$, $j = 0, 1, \ldots, n_t$ with a time step h such that $t_{n_t} = 1$, i.e. $h = \frac{1}{n_t}$. The approximate solution at time t_j is $x_j \approx x(t_j)$. The approximation is done via a (variable) order polynomial that is used to interpolate the last s solution values, $s \in \{1, 2, \ldots, S\}$ where S is the maximum order of the method. The general form of a BDF is

$$\sum_{k=0}^{S} \alpha_k x_{n-k} = h\beta f(t_n, x_n, p), \quad n = S, S+1, \ldots, n_t.$$

The coefficients α_k and β are chosen to guarantee a maximal approximation order. The method is implicit for $\beta \neq 0$, because the unknown new approximation x_n is

argument of the possibly nonlinear right-hand side function f. Typically a Newton type method is used to solve for x_n.

From Theorem 4 we know that if f is partially differentiable the solution of the initial value problem is a differentiable function of x and p and the desired sensitivities exist and are bounded. If one writes the solution of problem (34) in the form

$$x(t, x, p) = \xi + \int_0^1 f(\tau, x(\tau, \xi, p), p)\mathrm{d}\tau,$$

a differential equation for the sensitivities can be formulated by formally differentiating the integral equation with respect to x and p. We obtain

$$\mathrm{D}_\xi\, x(t, x, p) = 1 + \int_0^1 \mathrm{D}_2\, f(\tau, x(\tau, \xi, p), p)\, \mathrm{D}_\xi\, x(\tau, \xi, p)\mathrm{d}\tau$$

and

$$\mathrm{D}_p\, x(t, \xi, p) = \int_0^1 \mathrm{D}_2\, f(\tau, x(\tau, \xi, p), p)\, \mathrm{D}_p\, x(\tau, \xi, p) + \mathrm{D}_3\, f(\tau, x(\tau, \xi, p), p)\mathrm{d}\tau.$$

Writing $s_0(t) = \mathrm{D}_\xi\, x(t, \xi, p)$ and $s(t) = \mathrm{D}_p\, x(t, \xi, p)$ we have

$$\mathrm{D}_t \begin{pmatrix} s_0(t) \\ s(t) \end{pmatrix} = \begin{pmatrix} \mathrm{D}_2\, f(t, x(t, \xi, p), p) s_0(t) \\ \mathrm{D}_2\, f(t, x(t, \xi, p), p) s(t) + \mathrm{D}_3\, f(t, x(t, \xi, p), p) \end{pmatrix}, \quad \begin{pmatrix} s_0(0) \\ s(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

which are known as the adjoint or variational differential equation to the initial value problem (34). They can in principle be solved using the BDF method described above.

However, another method that is used in practice because of its computational advantages is *internal numerical differentiation* [**6, 7**]. The basic idea is to differentiate the BDF algorithm itself with respect to initial values and parameters. This way internal states of the integrator can be reused, for example adaptive time grids.

Regardless of the method used to compute sensitivities there is a need for the Jacobians of f with respect to x and p. Also for the not augmented BDF method differential information is needed if a Newton method is used to solve the nonlinear system of equations. In general these could be generated using finite differences at the cost of extra evaluations of the right hand side function f. Depending on the problem, besides the increased cost, finite differences might be unstable or even unreliable. An alternative is to use analytic expressions, however, this would mean the user has to provide them and for large systems this can become very tedious even if symbolic computing tools are used. An advanced method is the use of automatic or algorithmic differentiation (AD) [**35**]. To this end the elementary operations that are needed to evaluate the right hand side f are recorded and can then be differentiated. Differential information of any order can be obtained by use of the chain rule. For the simple example $f\big(h(x)\big)$ we have

$$\mathrm{D}_x\, f\big(h(x)\big) = \mathrm{D}_h\, f\, \mathrm{D}_x\, h.$$

With AD the chain rule is either evaluated from left to right (forward mode) or right to left (reverse mode). Note that this is not equivalent to symbolic differentiation with the help of a symbolic mathematics program (like Mathematica). There f would be regarded as a single expression whereas AD regards the elementary operations that make up the right hand side. Another advantage of AD is that sparsity information for Jacobian or Hessian matrices can be obtained easily, probably saving storage and computation time.

In actual implementations sophisticated approximation error controlling strategies are used as well as adaptive time grids and adaptive method order and other improvements that increase the robustness and efficiency of the BDF method. The

implementation we use was written by Dominik Skanda for his PhD thesis [**83**]. It uses `CppAD` [**3**] for automatic differentiation and includes many advanced techniques, that were not included in this description.

3.1.2. *Nonlinear Programming.* Eventually, the NLP (32) has to be solved to obtain an approximated optimal control solution to problem (30). We will only provide a very short overview over the most important results with a focus on interior point methods. There is a bulk of literature available, e.g. textbooks like [**71, 31, 60**]. We generally follow the review [**25**].

We rephrase the problem we are dealing with in general terms to facilitate the discussion. Consider

$$\begin{aligned} &\min_x f(x) \\ \text{s.t. } &v(x) = 0, \\ &w(x) \geq 0, \end{aligned} \tag{35}$$

where $x \in \mathbb{R}^{n_x}$ and $f : \mathbb{R}^{n_x} \to \mathbb{R}$, $v : \mathbb{R}^{n_x} \to \mathbb{R}^{n_v}$, and $w : \mathbb{R}^{n_x} \to \mathbb{R}^{n_w}$ are at least twice continuously differentiable. For notational convenience we write $\nabla f(x) := \mathrm{D}_x f(x)^{\mathrm{T}} \in \mathbb{R}^{n_x \times 1}$ for the gradient of f at x and $\nabla^2 f(x)$ for the Hessian of f at x. The Jacobian $\mathrm{D}_* v$ will be denoted by $J_v(x)$, analogously J_w is defined while the combined Jacobian $J(x) := [J_v, J_w](x)$ is defined thusly. Additionally, to simplify the notation we will write $x \cdot y$ for the componentwise multiplication of two vectors $x, y \in \mathbb{R}^{n_x}$.

Definition 45 (Feasible Region, [**25**, Definition 2.1]). Let the constraint functions $w(x)$ and $v(x)$ be given. The set

$$\Omega_F = \{x \in \mathbb{R}^{n_x} \mid v(x) = 0 \text{ and } w(x) \geq 0\}$$

is called the **feasible region**

The next definition describes conditions under which "interesting" points of (35) might exist.

Definition 46 (KKT Points, [**25**, Definition 6.1]). Let problem (35) be given. The (first-order) **KKT conditions** hold at a point $x^* \in \mathbb{R}^{n_x}$ if there is $\lambda^* = (\lambda_v, \lambda_w)^{\mathrm{T}} \in \mathbb{R}^{n_x}$, $\lambda_v \in \mathbb{R}^{n_v}$, $\lambda_w \in \mathbb{R}^{n_w}$ such that

(1) $x \in \Omega_F$, (feasibility)
(2) $\nabla f(x^*) - J(x^*)^{\mathrm{T}} \lambda = 0$, (stationarity),
(3) $\lambda_v \geq 0$ (nonnegativity of the inequality multipliers), and
(4) $v(x^*) \cdot \lambda_w^* = 0$ (complementarity for the inequality constraints).

Additionally, x^* is called a **KKT point**.

Remark 24. The second condition of the theorem becomes more clear if we look at the Lagrangian

$$\mathcal{L}(x, \lambda) = f(x) - \lambda^{\mathrm{T}} \begin{pmatrix} v(x) \\ w(x) \end{pmatrix},$$

i.e. KKT points are stationary points of the associated Lagrangian function to problem (35). The Lagrangian resembles a remarkable similarity to the Hamiltonian encountered in optimal control and indeed the two problems are very similar, both include minimizing an objective function with respect to a variable subject to constraints. The difference lies in the dimensionality of the search space, finite versus infinite.

The KKT conditions are based on the linearization of the Lagrangian and the constraint functions. However, on their own they are not necessary conditions for

an optimum to occur. More assumptions are needed and to state them we define

$$\mathcal{A}(x) := \{j \mid w_j(x) = 0\},$$

the set of indices of active constraints: Inequality constraints for which equality holds.

DEFINITION 47 (LICQ, [**25**, Definition 6.3]). Consider problem (35). The **linear independence constraint qualification** hold at a feasible point x^* if the combined Jacobian

$$[J_{\mathcal{A}}(x^*), J_v(x^*)]$$

has full rank. Here, $J_{\mathcal{A}}(x^*)$, is the Jacobian of the active inequality constraints.

The LICQ condition is fulfilled if the gradients of the active constraint functions are linearly independent. Now necessary conditions can be stated.

THEOREM 48 (First Order Necessary Conditions, [**25**, Lemma 6.5]). *Let x^* be a local minimizer of* (35)*. If LICQ holds at x^* then the KKT conditions are satisfied for x^*.*

The other way around if LICQ holds, which can be checked numerically then KKT points are candidates for a local minimum of f. Finally, to state sufficient conditions we define the Hessian of the Langrange function with respect to x:

$$H(x, \lambda) = \mathrm{D}_x^2 \mathcal{L}(x, \lambda) = \nabla^2 x - \sum_{j=1}^{n_v} \lambda_j^{(v)} \nabla^2 v_j(x) - \sum_{j=1}^{n_w} \lambda_j^{(v)} \nabla^2 w_j(x),$$

where $\lambda^{(v)}$ and $\lambda^{(w)}$ are the Lagrange multipliers with respect to the equality and inequality constraints, respectively.

THEOREM 49 (Second Order Sufficient Conditions, [**25**, Theorem 6.10]). *For problem* (35) *the point x^* is a local, isolated minimizer if*

1. *x^* is feasible and LICQ holds for x^*,*
2. *x^* is a KKT point and $\lambda_j^* > 0$ for all $j \in \mathcal{A}(x^*)$ (strict complementarity),*
3. *for all vectors $p \neq 0$ satisfying $J_{\mathcal{A}}(x^*)p = 0$ there is a $\omega > 0$ such that*

$$p^{\mathrm{T}} H(x^*, \lambda *) p \geq \omega \|p\|^2$$

holds.

REMARK 25. The condition on the Hessian $H(x, \lambda)$ can be restated in terms of the reduced Hessian

$$N_{\mathcal{A}}^{\mathrm{T}} H(x, \lambda) N_{\mathcal{A}}$$

where $N_{\mathcal{A}}$ is a matrix of basis vectors for the null space of $[J_{\mathcal{A}}(x^*), J_v(x^*)]$.

With this we end the discussion of the theoretical background and now turn to the algorithmic side of the problem. We use the interior point solver `IPOPT` [**89**] later. To outline the algorithm it implements we consider the simplified problem

$$\begin{aligned} &\min_x f(x) \\ \text{s.t. } &v(x) = 0 \\ &x \geq 0. \end{aligned} \tag{36}$$

We assume f and v to be at least twice continuously differentiable. Additionally we assume at least one local isolated minimizer x^* exists and that the LICQ conditions are satisfied at x^*. General nonlinear inequality constraints $w(x) \geq 0$ can be treated through the use of a slack variable $s \in \mathbb{R}^{n_w}$ as in

$$w(x) - s = 0, \quad s \geq 0.$$

The general inequality is replaced by an equality constraint and a bound constraint for s which is now also subject to optimization.

The basic idea of an interior point method is to replace the constrained problem (36) with a series of problems only constrained by equality constraints by augmenting the objective with a so called barrier term. Common is

$$\begin{aligned} \min_x \theta(x,\mu) &= f(x) - \mu \sum_{j=1}^{n_x} \log \chi_j \\ \text{s.t. } v(x) &= 0, \end{aligned}$$

where $\mu > 0$ is called the barrier parameter, which is consecutively driven towards zero. Obviously, $\log \chi_j$ will approach infinity if the component χ_j converges to zero and thus the bound constraint can not be broken. The homotopy problem

$$\begin{aligned} \nabla f(x) + J_v(x)^{\mathrm{T}} \lambda - z &= 0, \\ v(x) &= 0, \\ x \cdot \lambda - \mu 1_{n_x} &= 0 \end{aligned} \tag{37}$$

is equivalent to the KKT conditions for the original problem (36) for $\mu = 0$ and additionally $x, z > 0$ has to hold. In the third equation 1_{n_x} is the vector $(1, 1, \ldots, 1)^{\mathrm{T}} \in \mathbb{R}^{n_x}$. Here λ are the Lagrange multipliers for the equality constraints and z for the zero bound. Since we assume LICQ to hold this are necessary conditions for an optimum and a solution of (37) can be numerically obtained by applying a version of Newtons method for a fixed $\mu_j > 0$. Therefore we have an outer iteration over μ (using the index j) and inner iterations of the Newton method (using index k). In each iteration of inner problem we have to solve the linear system

$$\begin{pmatrix} W_k & A_k & -I \\ A_k^{\mathrm{T}} & 0 & 0 \\ Z_k & 0 & X_k \end{pmatrix} \begin{pmatrix} d_k^x \\ d_k^\lambda \\ d_k^z \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + A_k \lambda_k - z_k \\ v(x_k) \\ x_k \cdot z_k - \mu_j e \end{pmatrix}$$

where $A_k = J_v(x_k)$, $W_k = \nabla^2 \mathcal{L}(x_k, \lambda_k, z_k)$. The new iterates are generally computed through

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k d_k^x, \\ \lambda_{k+1} &= \lambda_k + \alpha_k d_k^\lambda, \\ z_{k+1} &= z_k + \alpha_k d_k^z, \end{aligned}$$

with a step size α_k, which is computed through a sophisticated filter-line-search approach, **[89]**. It can be shown, **[25]** that decreasing μ monotonously, will produce a differentiable path (with respect to μ) through the interior of the feasible region of solutions to the barrier problem that converges to a local solution of the original problem (36).

Practical implementations like `IPOPT` employ various additional tweaks and subalgorithms like second order corrections, feasibility restoration phases, and more to ensure a robust and efficient convergence to a feasible minimizer. These tweaks are a topic of its own and we are not presenting any details here.

3.1.3. *Implementation Details of the Multiple Shooting Algorithm.* The multiple shooting algorithm outlined above is implemented in the software package `DOT` by Markus Esenwein, **[18]**. The software can solve multi-stage Mayer problems subject to time dependent ODEs with nonlinear equality and inequality constraints on path and controls as well as nonlinear constraints on the multiple-shooting nodes. Multi-stage means that there might be several right hand sides f_j and according time intervals $T_j = [t_0^j, t_1^j]$ with $t_1^j = t_0^{j+1}$, $j = 1, 2, \ldots, n$ possibly depending on the same controls and/or parameters. They arise for example when modeling chemical

control processes with several reaction stages, like heating phase, reaction phase, and cooling phase.

`DOT` uses the BDF integrator from Dominik Skanda [**83**] and `IPOPT` to solve the two main problems of integrating the initial value problem and optimizing the NLP. The controls are approximated by arbitrary order polynomials on each multiple shooting interval. The optimization variables with respect to the control are thus the polynomial coefficients.

To solve the NLP various functions have to be differentiated once or twice with respect to the optimization variables. Although, finite differences could be used to approximate the sensitivities the stability and efficiency is highly increased if exact gradients, Jacobians and Hessians are provided. Automatic differentiation via `CppAD` [**3**] (as described in Section 3.1.1) is used for this purpose for the objective and the constraint functions. Since the constraints involve the solution of the ODE the sensitivity generating capabilities of the BDF integrator are used to obtain accurate derivatives of the solution of the ODE with respect to the control parameters, initial values, and the length of the multiple shooting intervals.

4. Summary

In this chapter we provide an overview of the theory of mathematical control with an emphasis on optimal control and concluded with some remarks on how such problems might be solved numerically. The control systems we consider here are based on ordinary differential equations to provide a dynamical system map that takes an initial state $\xi \in \mathcal{X} \subset \mathbb{R}^{n_x}$ to some final state x_1 via the time dependent path $x(t)$, $t \in \mathcal{T} \subset \mathbb{R}$. The dynamical system is subject to external input $u \in \mathcal{U}$, with values in $U \subset \mathbb{R}^{n_u}$, i.e. the ODE is given by

$$\mathrm{D}_t\, x = f(x, u).$$

We introduce the notion of controllability which means every initial state $\xi \in \mathcal{X}$ can be controlled to every final state $x_1 \in \mathcal{X}$ and see that for time independent linear systems, i.e. $f = Ax + Bu$, with $U = \mathbb{R}^{n_u}$ controllability is equivalent to the condition that the matrix

$$[B, AB, A^2B, \ldots, A^{n_x-1}B]$$

has rank n_x. For strict subsets $U \subset \mathbb{R}^{n_u}$ controllability depends on the stability properties of the matrix A, i.e. all eigenvalues have to have positive or zero real parts. A similar albeit less practical condition can be stated for time dependent linear systems. For nonlinear systems the problem is more complex and the concept of controllability has to be weakened to the less strict concept of reachability, which describes sets of points that can be reached from an initial state ξ. Using Lie algebra techniques a rank condition can be given that if fulfilled is equivalent to the reachable set of a point x having nonempty interior.

Optimal control problems are control systems subject to an objective functional, i.e. a performance measure $J : \mathcal{X} \times U \to \mathbb{R}$, $J\big(x(t), u(t)\big) \mapsto \mathbb{R}$ that has to be minimized. One approach to problems of this type is the Pontryagin minimum principle which replaces the optimization over an infinite search space (the space of admissible control functions) to a pointwise minimization of the Hamiltonian function H associated with an optimal control problem. To this end a costate variable $\lambda : \mathbb{R} \to \mathbb{R}^{n_x}$ is introduced and using the conditions of the minimum principle a boundary value problem is obtained for x and λ that has to be fulfilled by an optimal solution. The minimum principle gives necessary conditions that have to be realized along an optimal trajectory $x^*(t)$, $\lambda^*(t)$ and $u^*(t)$.

Although the minimum principle can be used to solve optimal control problems in practice direct numerical algorithms are employed mainly. They aim at discretizing the control function space and approximate the optimal control problem with an NLP. Our approach is multiple shooting where the overall time interval is divided into subintervals on which the control is parametrized finitely, e.g. piecewise constant. The initial values for the state equations are also subject to the optimization and additional continuity constraints guarantee the continuity of the solution $x^*(t)$. As a result two main subproblems remain: Integration of the initial value problems and solving the NLP. For the integration a BDF method is used while the NLP is solved using the interior point software `IPOPT`. All the differential information that is needed in the integrator and `IPOPT` is generated using automatic differentiation with `CppAD`.

CHAPTER 4

Interpolation

1. Introduction

Interpolation in general describes the process of obtaining information about a mapping from a finite number of samples. The defining feature of interpolation (versus approximation) is that the samples are reproduced exactly by the interpolation process.

EXAMPLE 9 (Lagrange Polynomial). Let $f : \mathbb{R} \to \mathbb{R}$ be a (unknown) function. Further we are given $x_i \in \mathbb{R}$, $i = 1, 2, \ldots, n$, $x_1 < x_2 < \cdots < x_n$, and $f_i = f(x_i)$. The objective is to construct $s_f : \mathbb{R} \to \mathbb{R}$ such that $f_i = s_f(x_i)$ holds. One possible solution is

$$s_f(x) = \sum_{i=0}^{n} f_i \ell_i(x) \tag{38}$$

with the Lagrange basis polynomials

$$\ell_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}.$$

Since $\ell_k(x_i) = 0$ for $k \neq i$ and $\ell_i(x_i) = 1$ the interpolation condition is fulfilled.

The interpolation of function values on the real line, also called univariate interpolation is a cornerstone of numerical mathematics, **[14, 85]**. Therefore, we will only highlight a few results that we need later when dealing with multivariate interpolation.

We will now state the general form of the interpolation problem considered here. To this end let W be a linear (function) space and W^* its dual. Further let $\mathcal{L} = \{L_i\}_{i=1}^n \subset W^*$ be a finite set of bounded linear functionals, and $F = \{f_i\}_{i=1}^n$ a set of real numbers. The aim of interpolation is to find a $s_\mathcal{L} \in W$ such that

$$L_i(s_\mathcal{L}) = f_i, \quad i = 1, 2, \ldots, n \tag{39}$$

holds. The existence of an interpolation function is highlighted in the next theorem.

THEOREM 50 ([**14**, Theorem 2.2.2]). *Let W be a n-dimensional linear space and $\mathcal{L} = \{L_i\}_{i=1}^n \subset W^*$ elements of its dual. The interpolation problem* (39) *has a solution iff the L_i are linearly independent in W^*. The solution is unique.*

It is clear that if we choose $s_i \in W$, $i = 1, 2, \ldots, n$ linearly independent, the unknown interpolation function $s_\mathcal{L}$ can be uniquely represented as a linear combination $s_\mathcal{L} = \sum_{i=1}^n a_i s_i$ with $a_i \in \mathbb{R}$. The interpolation condition (39) takes the form

$$L_i(s_\mathcal{L}) = \sum_{j=1}^{n} a_j s_j = f_i, \quad i = 1, 2, \ldots, n, \tag{40}$$

and the solution of the interpolation problem is coupled to the regularity of the system of linear equations in (40). In other words the dimension of the interpolation space W must be the same as the number of functionals.

EXAMPLE 10. Picking up Example 9 we have as functionals the evaluation of f at the nodes x_i:

$$L_i(f) = f(x_i) = f_i, \quad i = 1, 2, \ldots, n.$$

We choose $W = \pi_n(\mathbb{R})$, the space of polynomials with maximum degree n and coefficients in $\mathbb{R}$. The Lagrange polynomials form a basis of that space, and using the interpolation formula (38) the unique polynomial of degree n is obtained that fulfills

$$L_i(s_{\pi_n}) = s_{\pi_n}(x_i) = f_i.$$

Other important and frequently used functionals are

$$L_i(f) = \mathrm{D}_x^i f(x_0), \quad i = 0, 1, \ldots, n$$

which give rise to the Taylor polynomial and

$$L_i^j(f) = \begin{cases} f(x_i) & i = 1, 2, \ldots, n,\ j = 0, \\ \mathrm{D}_x f(x_i) & i = 1, 2, \ldots, n,\ j = 1, \end{cases}$$

which is known as Hermite interpolation.

2. Multivariate Interpolation

In contrast to the interpolation problem in one variable the multivariate counterpart proves to be more complex and difficult. The major obstacle to overcome is to determine the space of interpolation functions that in a certain way optimally fits with the given data. Often, there are many different interpolation spaces or their construction depends on the functionals that are interpolated. A major part of this material is based on [**91**].

DEFINITION 51 (Haar Space, [**91**, Definition 2.1]). Let $\Omega \subseteq \mathbb{R}^d$ contain at least n points, moreover let $W \subseteq C(\Omega)$ be an n-dimensional linear space. If for arbitrary distinct points $x_i \in \Omega$ and arbitrary $f_i \in \mathbb{R}$, $i = 1, 2, \ldots, n$ there exists exactly one function $s \in W$ such that $s(x_i) = f_i$, $i = 1, 2, \ldots, n$ then W is called a **Haar space** of dimension n.

LEMMA 52 ([**91**, Theorem 2.2]). *If W is an n-dimensional Haar space then the following statements are equivalent*

- *Every $s \in W \setminus \{0\}$ has at most $n - 1$ zeros.*
- *For arbitrary distinct points x_i, $i = 1, 2, \ldots, n$ and any basis $\{s_i\}_{i=1}^n$ of W it holds that $\det\big(s_i(x_j)\big)_{i=1,\,j=1}^n \neq 0$.*

The crucial and somewhat surprising theorem is the following

THEOREM 53 (Mairhuber-Curtis, [**91**, Theorem 2.3]). *If $\Omega \subseteq \mathbb{R}^d$, $d \geq 2$ contains an interior point then there exists no Haar space on Ω with dimension $n \geq 2$.*

PROOF. Suppose $W = \operatorname{span}\{s_i\}_{i=1}^n$ is a Haar space on Ω. Since Ω contains an interior point, say x_0 there is a nonempty ball $B(x_0, \delta)$ with $\delta > 0$ such that we can fix $n-3$ distinct points $x_i \in B(x_0, \delta)$, $i = 3, 4, \ldots, n$. Now we construct continuous curves $x_1(t)$ and $x_2(t)$, $t \in [0, 1]$ with the following properties: $x_1(0) = x_2(1)$, $x_1(1) = x_2(0)$, $x_1(t) \neq x_2(t)$ for $t \neq 0, 1$, and $x_1(t), x_2(t) \neq x_i$ $i = 3, 4, \ldots, n$ for all t. In other words the curves do not intersect with each other (except at $t = 0$ and $t = 1$) and the other x_i. This is possible since $d \geq 2$. Now we have a look at the determinant

$$D(t) = \det\big(s_i(x_j)\big)_{i=1,\,j=1}^n.$$

Because W is supposed to be a Haar space $D(t)$ is continuous and nonzero for all t. However, it also holds that $D(0) = -D(1)$ because the first two rows of $D(t)$ are exchanged. This is a contradiction and W can not be Haar space. □

REMARK 26. The theorem excludes the possibility that for regions $\Omega \subset \mathbb{R}^d$ with $d \geq 2$ there exists a space of interpolation functions that can be used for all configurations of sample points to solve the interpolation problem. For multivariate interpolation two possibilities are left: Construct the space of interpolation functions based on the configuration of the sample points or only allow a certain configuration of sample points such that a given interpolation space can be employed.

REMARK 27. If we soften the requirement, that the interpolation space has exactly the same dimension as the number of functionals that are interpolated then we may find interpolation spaces that are unisolvent for all sets of sample points **[29]**, but it is yet impossible to determine the minimum dimension needed for the interpolation space given a certain amount of sample points.

In the application that we have in mind we have first order derivative information available at the sample points of the function f that we try to interpolate. So we are looking for an extension of Hermite interpolation to dimensions $d \geq 2$. This is not consistently defined. In this work we will constrain ourselves to the case that is usually called Hermite interpolation of total degree **[59]**. To this end let $X = \{x_k\}_{k=1}^N$ be a set of pairwise distinct points from $\mathbb{R}^d$, further let $m_1, m_2, \ldots, m_N \in \mathbb{N}_0$, then Hermite interpolation of total degree is defined by the functionals

$$L_{k,\alpha}(f) = \delta_{x_k} \circ \mathrm{D}^\alpha f = \mathrm{D}^\alpha f(x_k), \quad 0 \leq |\alpha| \leq m_k,\ k = 1, 2, \ldots, N.$$

For $m_k = 0$, $k = 1, 2, \ldots, N$ we get Lagrange interpolation. In our case $m_k = 1$ for all $k = 1, 2, \ldots, N$, which implies all first order partial derivatives are available. It also means we have $N + Nd = N(d + 1)$ functionals to interpolate.

The classic approach to multivariate interpolation are tensor based methods. The main idea is to treat each dimension independently and solve d univariate interpolation problems. Thus this approach is independent from the dimensionality of the overall problem and the univariate interpolation process that is used. The main drawback is that the interpolation nodes are confined to regular d-dimensional grids. This is problematic in two aspects: Sometimes the function that should be interpolated can not be evaluated on a complete grid or data might be missing at some nodes. Also, if a higher resolution or higher accuracy is only needed in part of a domain still the granularity of the whole grid has to be increased which, depending on the input dimension and the refinement means a huge increase in the problem size. This is due to the exponential dependence of the number of points in a tensor grid on the dimension, which is a problem in itself, because it essentially restricts the method to only a few dimensions.

2.1. Multivariate Polynomials. In the univariate setting polynomial based methods are predominant, mainly in the form of splines. This is due to the already mentioned fact that the involved interpolation spaces are Haar spaces and that efficient and robust implementations are possible. Theorem 53 tells us that we do not have this favorable situation in the multivariate case. Nevertheless, polynomial interpolation can be extended in various directions.

An overview over the historical development and different approaches for global polynomial interpolation can be found in the two review papers **[28, 29]**. Given the general problem two main strategies are possible. First, given a certain interpolation space of polynomials determine the sets of nodes for which the interpolation problem does have a solution. Often general conditions on the distribution of the nodes can be formulated. The other way around, given a set of nodes determine or construct a polynomial space such that the interpolation problem has a solution.

In [**10**] a framework is developed that allows to construct the minimum dimensional polynomial space that allows the unique solution to the interpolation problem given a set of points $X \subset \mathbb{R}^d$. The described method can be extended to interpolating general linear functionals L_k, including the Hermite case.

3. Radial Basis Function Interpolation

Radial basis function (RBF) interpolation is a method of multivariate interpolation that is truly dimension and grid independent and can thus be used for very general multivariate interpolation problems. It is the method we use for the purpose of interpolating the slow manifold in the optimal control problems we solve, see Chapters 5 and 6.

3.1. Positive Definite Functions. A central concept of RBF interpolation is that of a positive definite function.

Definition 54 (Positive definite function, [**91**, Definition 6.1]). A continuous function $\Phi : \mathbb{R}^d \to \mathbb{C}$ is said to be **positive semi-definite** if for all $N \in \mathbb{N}$, all sets of pairwise distinct nodes $X = \{x_k\}_{k=1}^N$, and all $\alpha \in \mathbb{C}^N$ it holds that

$$\sum_{\ell=1}^{N}\sum_{k=1}^{N} \alpha_\ell \overline{\alpha_k} \Phi(x_\ell - x_k) \geq 0.$$

If the quadratic form is positive for all $\alpha \in \mathbb{C}\setminus\{0\}$ then Φ is called **positive definite**.

The reason why this property is compelling lies in the fact that if we consider Lagrange interpolation, i.e. $L_k(f) = \delta_{x_k} f = f(x_k) = f_k$, with the set of pairwise distinct centers $X = \{x_k\}_{k=1}^N$ the interpolant

$$s_{f,X}(x) = \sum_{k=1}^{N} \lambda_k \Phi(x - x_k)$$

with Φ positive definite leads to the interpolation condition

$$\delta_{x_\ell} s_{f,X} = s_{f,X}(x_\ell) = \sum_{k=1}^{N} \lambda_k \Phi(x_\ell - x_k) = \delta_{x_\ell} f = f_\ell.$$

We obtain the linear system

$$A\lambda = F,$$

where

$$A_{\ell,k} = \Phi(x_\ell - x_k), \quad \lambda = \begin{pmatrix}\lambda_1 & \lambda_2 & \dots & \lambda_N\end{pmatrix}^{\mathrm{T}}, \quad F = \begin{pmatrix} f_1 & f_2 & \dots & f_N\end{pmatrix}^{\mathrm{T}}$$

with A positive definite since $\alpha^{\mathrm{T}} A \overline{\alpha} > 0$ holds for all $\alpha \in \mathbb{C}^N \setminus \{0\}$ by definition. That guaranties a unique solution to the interpolation problem for all sets of pairwise distinct nodes.

We will go on to characterize positive definite functions a little more. Some basic properties are collected in the next theorem.

Theorem 55 ([**91**, Theorem 6.2]). *If the $\Phi(x)$ are positive definite functions, then the following statements hold:*

(1) $\Phi(0) \geq 0$.
(2) $\Phi(-x) = \overline{\Phi(x)}$, $\forall x \in \mathbb{R}^d$.
(3) $|\Phi(x)| \leq \Phi(0)$, $\forall x \in \mathbb{R}^d$.
(4) $\Phi(0) = 0 \Leftrightarrow \Phi \equiv 0$.

(5) *If $\Phi_1, \Phi_2, \ldots, \Phi_n$ are positive semi-definite and $c_j \geq 0, j = 1, 2, \ldots, n \Rightarrow \Phi = \sum_{j=1}^{n} c_j \Phi_j$ is also positive semi-definite. If Φ_k is positive definite and $c_k > 0$ for at least one $k \in \{1, 2, \ldots, n\} \Rightarrow \Phi$ is positive definite.*
(6) *The product of two positive definite functions is also positive definite.*

We introduced positive definite functions with values in $\mathbb{C}$ since later we will employ Fourier transforms for their characterization. From the above theorem, property (2) it follows that positive definite functions can be real-valued only if they are even. In that case it is sufficient to use real coefficients in the quadratic form.

THEOREM 56 ([**91**, Theorem 6.3]). *Let $\Phi : \mathbb{R}^d \to \mathbb{R}$ be continuous. Then Φ is positive definite iff it is even and for all $N \in \mathbb{N}$, for all $\alpha \in \mathbb{R}^N \setminus \{0\}$, and for all pairwise distinct $\{x_k\}_{k=1}^N$ it holds that*

$$\sum_{\ell=1}^{N} \sum_{k=1}^{N} \alpha_\ell \alpha_k \Phi(x_\ell - x_k) > 0.$$

We will proceed with providing tools that will help to decide whether a given function is positive definite or not. It turns out that the Fourier transform plays a viable role in this. Therefore, the L^p spaces, $1 \leq p \leq \infty$, are defined as usual: $L^p(\Omega)$ for $\Omega \subset \mathbb{R}^d$ is the set of all measurable functions $f : \Omega \to \mathbb{R}$ for which the norm

$$\|f\|_{L^p(\Omega)} := \int_\Omega |f(x)|^p \, \mathrm{d}x, \quad 1 \leq p < \infty$$

is finite. For $p = \infty$ the norm is given by

$$\|f\|_{L^p(\Omega)} := \operatorname{ess\,sup}_{x \in \Omega} |f(x)| ,$$

which means there is a constant $K > 0$ such that $|f(x)| \leq K$ almost everywhere on Ω. Such a function is also said to be *essentially bounded*. Now we are ready to define the Fourier transform.

DEFINITION 57 (Fourier transform). Let $f \in L^1(\mathbb{R}^d)$. Its **Fourier transform** is defined by

$$\widehat{f}(x) := (2\pi)^{-d/2} \int_{\mathbb{R}^d} f(\omega) \mathrm{e}^{-\mathrm{i} x^{\mathrm{T}} \omega} \mathrm{d}\omega.$$

The **inverse Fourier transform** is given by

$$\check{f}(x) := (2\pi)^{-d/2} \int_{\mathbb{R}^d} f(\omega) \mathrm{e}^{\mathrm{i} x^{\mathrm{T}} \omega} \mathrm{d}\omega.$$

If we now assume that $\Phi \in C(\mathbb{R}^d) \cap L^1(\mathbb{R}^d)$ has an integrable Fourier transform in $L^1(\mathbb{R}^d)$ we have

$$\Phi(x) = (2\pi)^{-d/2} \int_{\mathbb{R}^d} \check{\Phi}(\omega) \mathrm{e}^{\mathrm{i} x^{\mathrm{T}} \omega} \mathrm{d}\omega.$$

The quadratic form from Definition 54 can thus be restated as

$$\begin{aligned} \sum_{\ell=1}^{N} \sum_{k=1}^{N} \alpha_\ell \alpha_k \Phi(x_\ell - x_k) &= (2\pi)^{-d/2} \sum_{\ell=1}^{N} \sum_{k=1}^{N} \alpha_\ell \alpha_k \int_{\mathbb{R}^d} \check{\Phi}(\omega) \mathrm{e}^{\mathrm{i} x^{\mathrm{T}} \omega} \mathrm{d}\omega \\ &= (2\pi)^{-d/2} \int_{\mathbb{R}^d} \check{\Phi}(\omega) \left| \sum_{\ell=1}^{N} \alpha_\ell \mathrm{e}^{\mathrm{i} x^{\mathrm{T}} \omega} \right|^2 \mathrm{d}\omega, \end{aligned}$$

which is nonnegative if $\check{\Phi}$ is nonnegative. This means a function is positive semi-definite if its Fourier transform is nonnegative.

REMARK 28. In a more general form, i.e. if the function in question is not integrable, a general Borel measure μ has to be used. The main result then is: A function is positive semi-definite iff it is the Fourier transform of a finite nonnegative Borel measure μ. In other words we have to look at the distributional Fourier transform of Φ. The details of this approach are left out because in our case we can restrict ourselves to the case of an integrable function with a measure with Lebesgue density.

For the sake of interpolation we are interested only in functions that are positive definite, otherwise there might be cases in which the interpolation matrix is singular for a certain configuration of points and a solution of the interpolation problem would not exist. Hence we proceed with stating results that are concerned with distinguishing positive and positive semi-definite functions. The first theorem shows how positive definite functions can be constructed from general functions f.

THEOREM 58 ([**91**, Corollary 6.9]). *Let $f \in L^1(\mathbb{R}^d)$ be continuous, nonnegative, and nonvanishing. Then*

$$\Phi(x) := \int_{\mathbb{R}^d} f(\omega) \mathrm{e}^{\mathrm{i}x^{\mathrm{T}}\omega} \mathrm{d}\omega$$

is positive definite.

In praxis we are more often confronted with the problem to decide whether a given function is positive definite or not.

THEOREM 59 ([**91**, Theorem 6.11]). *Let $\Phi \in L^1(\mathbb{R}^d)$ be continuous. Then Φ is positive definite iff it is bounded and its Fourier transform is nonnegative and nonvanishing.*

EXAMPLE 11. A popular choice for a positive definite function is the Gaussian

$$\Phi(x) = \mathrm{e}^{-c\|x\|_2^2}, \quad c > 0, \quad x \in \mathbb{R}^d$$

where c is used as a scaling parameter.

PROOF. Obviously, Φ is bounded for all $x \in \mathbb{R}^d$. Additionally, denote with $G(x)$ the Gaussian with $c = \frac{1}{2}$. It holds that $\widehat{G} = G$ and therefore $G(x)$ is positive definite since it is nonnegative and nonvanishing on all $\mathbb{R}^d$. Lastly, $\Phi(x) = G(\sqrt{2c}x)$ and with the scaling property of the Fourier transform we find $\widehat{\Phi} = (\frac{1}{\sqrt{2c}})^d \widehat{G}(\frac{1}{\sqrt{2c}}\omega)$, which again is nonnegative and nonvanishing and thus concludes the proof. □

The Gaussian is an example of a radial function which are the most commonly used type of functions for the purpose of interpolation. Their main advantage is that they only have to be evaluated for values in $\mathbb{R}$ and a simple norm computation suffices to extend them to the multivariate setting. Let $\mathbb{R}_0^+ := \{x \,|\, x \in \mathbb{R},\ x \geq 0\}$.

DEFINITION 60 (Radial Functions, [**91**, Definition 6.15]). A function $\Phi : \mathbb{R}^d \to \mathbb{R}$ is called **radial** if there is a function $\phi : \mathbb{R}_0^+ \to \mathbb{R}$ such that $\Phi(x) = \phi(\|x\|_2)$.

DEFINITION 61 ([**91**, Definition 6.16]). A function $\phi : \mathbb{R}_0^+ \to \mathbb{R}$ is called **positive definite** on $\mathbb{R}^d$ if $\Phi(x) = \phi(\|x\|_2)$, $x \in \mathbb{R}^d$ is positive definite.

Since $\mathbb{R}^\ell$, $\ell < d$ is a subspace of $\mathbb{R}^d$ any ϕ that is positive definite on $\mathbb{R}^d$ is also positive definite on $\mathbb{R}^\ell$. Another advantage is that also the characterization as positive definite through Fourier transforms is reduced to the univariate setting.

THEOREM 62 ([**91**, Theorem 6.18]). *Consider $\phi \in C[0,\infty)$ satisfying $r \to r^{d-1}\phi(r) \in L^1[0,\infty)$. Then ϕ is positive definite on $\mathbb{R}^d$ iff it is bounded and*

$$\widehat{\phi}_d(r) := r^{-(d-2)/2} \int_0^\infty \phi(t) t^{d/2} J_{d-2}(rt) \mathrm{d}t$$

is nonnegative and nonvanishing.

The function J_z is the Bessel function of first kind with order $z \in \mathbb{C}$, which comes into play while evaluating the Fourier integral on the $d-1$ dimensional sphere $S^{d-1} = \{x \,|\, x \in \mathbb{R}^d, \|x\|_2 = 1\}$.

3.2. Native Spaces and Error Estimates. We proceed with yet another view on positive definite functions with the aim of embedding the interpolation problem in a Hilbert space setting and finally being able to state error estimates in appropriate norms.

Definition 63 (Positive definite kernel, [**91**, Definition 6.24]). A continuous function $\Phi : \Omega \times \Omega \to \mathbb{R}$ is said to be a **positive semi-definite** kernel on $\Omega \subset \mathbb{R}^d$ if for all $N \in \mathbb{N}$, all sets of pairwise distinct nodes $X = \{x_k\}_{k=1}^N \subset \Omega$, and all $\alpha \in \mathbb{C}^N \setminus \{0\}$ it holds that

$$\sum_{\ell=1}^{N} \sum_{k=1}^{N} \alpha_\ell \alpha_k \Phi(x_\ell, x_k) \geq 0.$$

If strict inequality holds, then Φ is called **positive definite**.

Kernels allow more general functions to act in the interpolation, the already introduced radial functions fit in by $\Phi(x, y) := \phi(\|x - y\|_2)$. The general interpolant can now be written as

$$s(x) = \sum_{k=1}^{N} \lambda_k \Phi(x, x_k).$$

The other generalization in the definition is that the function Φ might be restricted to a domain $\Omega \subset \mathbb{R}^d$, which might even be a finite set. In that case N is only allowed to be chosen in a way such that a set of pairwise distinct nodes can be found. There are indeed positive definite functions with compact support, i.e. Ω is a proper subset of $\mathbb{R}^d$, but we will not pursue this topic any further here, since we will focus on the Gaussian kernel.

Definition 64 (Reproducing kernel, [**91**, Definition 10.1]). Let $\mathcal{F}$ be a real Hilbert space of functions $f : \Omega \to \mathbb{R}$. If for $\Phi : \Omega \times \Omega \to \mathbb{R}$ it holds that

(1) $\Phi(\cdot, y) \in \mathcal{F}\ \forall y \in \Omega$,
(2) $f(y) = (f, \Phi(\cdot, y))_{\mathcal{F}}\ \forall f \in \mathcal{F}, y \in \Omega$,

then Φ is called a **reproducing kernel** of $\mathcal{F}$.

Remark 29. The reproducing kernel Φ of a Hilbert space is unique in the sense that if we would have two different Φ_1 and Φ_2 we find $(f, \Phi_1(\cdot, y) - \Phi_2(\cdot, y))_{\mathcal{F}} = 0$ for all f and y. Then choosing $f = \Phi_1(\cdot, y) - \Phi_2(\cdot, y)$ shows $\Phi_1(\cdot, y) - \Phi_2(\cdot, y) = 0$ for every fixed y.

The next theorem collects some basic properties of reproducing kernel Hilbert spaces.

Theorem 65 ([**91**, Theorem 10.2, Theorem 10.3]). *If $\mathcal{F}$ is a Hilbert space with reproducing kernel Φ then the following statements hold:*

(1) *The pointwise evaluation functionals δ_y are continuous for all $y \in \Omega$, as such $\delta_y \in \mathcal{F}^*$.*
(2) $\Phi(x, y) = (\Phi(\cdot, x), \Phi(\cdot, y))_{\mathcal{F}} = (\delta_x, \delta_y)_{\mathcal{F}^*}$.
(3) $\Phi(x, y) = \Phi(y, x)$ *for* $x, y \in \Omega$.
(4) *If f_n converges to f in the Hilbert space norm, for $f_n, f \in \mathcal{F}$, then f_n converges to f also pointwise.*

The statements are consequences of the fact that the Riesz representer for δ_y is the reproducing kernel, i.e. $\delta_y(f) = (f, \Phi(\cdot, y))_{\mathcal{F}}$. Until here we did not assume that Φ is positive (semi) definite and it turns out that this is not needed since it comes into play as a natural property of a reproducing kernel.

THEOREM 66 ([**91**, Theorem 10.4]). *Let $\mathcal{F}$ be a reproducing-kernel Hilbert function space with $\Phi : \Omega \times \Omega \to \mathbb{R}$ the reproducing kernel. Then Φ is positive semi definite. Additionally Φ is positive definite iff the point evaluation functionals are linearly independent in $\mathcal{F}^*$.*

The connection between positive definiteness of Φ and the linear independence of δ_y in $\mathcal{F}^*$ is something we observed already. In the beginning of the chapter we concluded that the Lagrange interpolation problem can be solved only if the nodes are pairwise distinct (i.e. the point evaluation functionals are linearly independent). In that case only a positive definite function will be sufficient to generate an invertible interpolation matrix.

The previous theorems were concerned with the situation that we have a Hilbert space and its reproducing kernel given. More interesting for practical purposes is the answer to the question: If we have a positive definite function at hand, can we describe or construct some associated function space? To answer that question we start with Φ positive definite on $\Omega \subset \mathbb{R}$. Since $\Phi(\cdot, y) \in \mathcal{F}$ for all $y \in \Omega$ it seems natural to define the linear space

$$F_\Phi(\Omega) := \operatorname{span}\{\Phi(\cdot, y) \,|\, y \in \Omega\}$$

with bilinear form

$$\left(\sum_{\ell=1}^{N} \alpha_\ell \Phi(\cdot, x_\ell), \sum_{k=1}^{M} \beta_k \Phi(\cdot, y_k)\right)_\Phi := \sum_{\ell=1}^{N}\sum_{k=1}^{M} \alpha_\ell \beta_k \Phi(x_\ell, x_k). \tag{41}$$

The linear space F_Φ together with the bilinear form constitutes a pre-Hilbert space.

THEOREM 67 ([**91**, Theorem 10.7]). *Let $\Phi : \Omega \times \Omega \to \mathbb{R}$ be a positive definite kernel, then $(\cdot,\cdot)_\Phi$ from (41) is an inner product on $F_\Phi(\Omega)$, which is a pre-Hilbert space.*

We now consider the completion $\mathcal{F}_\Phi(\Omega)$ of $F_\Phi(\Omega)$ with respect to the $\|\cdot\|_\Phi$-norm. Since the point-evaluation functionals are continuous on F_Φ their extension on $\mathcal{F}_\Phi$ is also continuous and can be used to define function values for the abstract elements of the completion of F_Φ and we write

$$\delta_x(f) = f(x) := (f, \Phi(\cdot, x))_\Phi$$

for all $f \in \mathcal{F}_\Phi(\Omega)$. Technically, we have to define an operator

$$R : \mathcal{F}_\Phi(\Omega) \to C(\Omega), \quad R(f)(x) := (f, \Phi(\cdot, x))_\Phi$$

to identify the abstract elements of $\mathcal{F}_\Phi$ with continuous functions. For this operator it can be shown that its output are continuous functions for all $f \in \mathcal{F}_\Phi$ and that it is injective. Now the native space of Φ can be defined.

DEFINITION 68 (Native space, [**91**, Definition 10.9]). The reproducing-kernel Hilbert function space corresponding to the positive definite kernel $\Phi : \Omega \times \Omega \to \mathbb{R}$ is defined by

$$\mathcal{N}_\Phi(\Omega) := R\big(\mathcal{F}(\Omega)\big).$$

The inner product is given by

$$(f, g)_{\mathcal{N}_\Phi(\Omega)} := \left(R^{-1}(f), R^{-1}(g)\right)_\Phi.$$

THEOREM 69 ([**91**, Theorem 10.11]). *Let Φ be a positive definite kernel. Additionally, let $\mathcal{G}$ be a reproducing-kernel Hilbert space with Φ the reproducing kernel. Then $\mathcal{G}$ is the native space $\mathcal{N}_\Phi(\Omega)$.*

Finally, we return to the characterization of positive definite functions via Fourier transforms and state a result that employs them to describe also the native space.

THEOREM 70 ([**91**, Theorem 10.12]). *Let $\Phi \in C(\mathbb{R}^d) \cap L^1(\mathbb{R}^d)$ be a real-valued positive definite function. Let*

$$\mathcal{G} := \left\{ f \in L^2(\mathbb{R}^d) \cap C(\mathbb{R}^d) \mid \frac{\widehat{f}}{\sqrt{\widehat{\Phi}}} \in L^2(\mathbb{R}^d) \right\}$$

be a space of functions and

$$(f,g)_{\mathcal{G}} := (2\pi)^{-d/2} \left(\frac{\widehat{f}}{\sqrt{\widehat{\Phi}}}, \frac{\widehat{g}}{\sqrt{\widehat{\Phi}}} \right)_{L^2(\mathbb{R}^d)} = (2\pi)^{-d/2} \int_{\mathbb{R}^d} \frac{\widehat{f}(\omega)\overline{\widehat{g}(\omega)}}{\widehat{\Phi}(\omega)} \mathrm{d}\omega$$

a bilinear form. Then $\mathcal{G}$ is a real Hilbert space with inner product $(\cdot,\cdot)_{\mathcal{G}}$ and reproducing kernel $\Phi(\cdot - \cdot)$, i.e. the native space $\mathcal{N}_\Phi(\mathbb{R}^d)$. Every $f \in \mathcal{N}_\Phi$ can be recovered from $\widehat{f} \in L^1 \cap L^2$.

REMARK 30. Note that the theorem explicitly assumes Φ to be a positive definite function as opposed to a kernel, i.e. Φ is translation invariant. Furthermore, no restriction to sets $\Omega \subset \mathbb{R}^d$ is possible. Still, the theorem allows to highlight a connection of native spaces of positive definite basis functions to classic Sobolev spaces. Looking at the definition of the Sobolev space of order s, $s > d/2$ also in terms of Fourier transforms

$$H^s(\mathbb{R}^d) = \{ f \in L^2(\mathbb{R}^d) \cap C(\mathbb{R}^d) \mid \widehat{f}(\cdot)(1 + \|\cdot\|_2^2)^{s/2} \in L^2(\mathbb{R}^d) \}$$

we see that it is equivalent to the native space of a positive definite kernel if its Fourier transform fulfills

$$c_1(1 + \|\omega\|_2^2)^{-s} \leq \widehat{\Phi}(\omega) \leq c_2(1 + \|\omega\|_2^2)$$

for $0 < c_1 \leq c_2$. This connection is more deeply explored in [**19**].

The last theorem allows us to describe the functions, that the native space consists of in terms of the properties of their Fourier transform. We are of course interested what that means for the Gaussian.

EXAMPLE 12. The Gaussian $\Phi(x) = \phi(\|x\|_2) = \mathrm{e}^{-c\|x\|_2^2}$ most certainly fulfills the assumptions put forward in Theorem 70 and we have (see Example 11)

$$\widehat{\Phi} = \left(\frac{1}{\sqrt{2c}} \right)^d \mathrm{e}^{-\frac{1}{2}\left\| \frac{1}{\sqrt{2c}}\omega \right\|_2}.$$

The native space thus consists of all continuous functions with a Fourier transform that decays exponentially. This space is rather small and excludes many smooth and even analytic functions like polynomials for example. Still, as we shall see later the Gaussian basis functions are a good choice for interpolation.

Now, that we have set up a proper environment we are ready to state some results concerning the interpolation error

$$\|f - s_{f,X}\|_{L^\infty(\Omega)},$$

between a function f and its interpolant $s_{f,X}$, including error estimates for the derivatives. First we will provide errors for interpolating functions from the native space of the basis function Φ.

The error of the interpolation will be stated in terms of the fill distance $h_{X,\Omega}$ which is a measure of how well X covers Ω.

DEFINITION 71 (Fill distance, [**91**, Definition 1.4]). Let $\Omega \subset \mathbb{R}^d$ be bounded and $X = \{x_k\}_{k=1}^N \subset \Omega$ be given. The **fill distance** $h_{X,\Omega}$ of X in Ω is given by

$$h_{X,\Omega} := \sup_{x\in\Omega} \min_{x_k\in X} \|x - x_k\|_2 .$$

The fill distance can be interpreted as the diameter of the largest data-free hole in Ω. Convergence of an interpolant to a function f will be measured in terms of $h_{X,\Omega}$. Additionally, we need the function space $C^k_\nu(\mathbb{R}^d)$:

$$C^k_\nu(\mathbb{R}^d) = \{f \in C^k(\mathbb{R}^d) \,|\, \mathrm{D}^\alpha f(x) = \mathcal{O}(\|x\|_2^\nu), \text{ as } \|x\|_2 \to 0, \ |\alpha| = k\}.$$

The general result is

THEOREM 72 ([**91**, Theorem 11.11]). *Let $\Phi \in C^k_\nu(\mathbb{R}^d)$ be positive definite and $\Omega \subset \mathbb{R}^d$ bounded and satisfying an interior cone condition* [**91**, Definition 3.6]. *For $\alpha \in \mathbb{N}_0^d$ with $|\alpha| \le \frac{k}{2}$ and $X = \{x\}_{k=1}^N \subset \Omega$ satisfying $h_{X,\Omega} \le h_0$ the error bound*

$$\|\mathrm{D}^\alpha f - \mathrm{D}^\alpha s_{f,X}\|_{L^\infty(\Omega)} \le C h_{X,\Omega}^{(k+\nu)/2-|\alpha|} \|f\|_{\mathcal{N}_\Phi(\Omega)}$$

holds.

The first thing to notice is that the error of the interpolant can be divided into one part depending on the underlying domain Ω and the set of node points X and another part depending on the norm of f in the native space norm. The interpolation order also depends on the smoothness properties of the basis function. Obviously, the Gaussian is in $C^k_\nu(\mathbb{R}^d)$ for all $\nu, k \in \mathbb{N}$ and therefore arbitrarily high algebraic convergence orders can be reached. This is an indicator that the order achieved in the theorem can be qualitatively enhanced and indeed spectral orders can be proven.

THEOREM 73 ([**91**, Theorem 11.22]). *Let Ω be a cube in $\mathbb{R}^d$. Further, let $\Phi = \phi(\|\cdot\|_2)$ be radial and satisfy $\left|\mathrm{D}^m \phi(\sqrt{\cdot})\right| \le M^m$, $M > 0$ fixed. Then, there exists a constant $d > 0$ such that*

$$\|f - s_{f,X}\|_{L^\infty(\Omega)} \le \exp\left(\frac{d \log(h_{X,\Omega})}{h_{X,\Omega}}\right) |f|_{\mathcal{N}_\Phi(\Omega)}$$

holds, if $h_{X,\Omega} < h_0$.

EXAMPLE 13. The Gaussian is a prime example for this behavior. We have $\phi(r) = \mathrm{e}^{-cr^2}$, $c > 0$ and hence $\mathrm{D}^m \phi(r) = (-1)^m c^m \phi(r)$, $m \in \mathbb{N}$. The assumptions of the theorem hold with $M = c$ and the stated error bound is valid. An example plot for interpolating the 1-dimensional function

$$y = f(x) = 3(1-x)^2 \mathrm{e}^{-x^2} - 10\left(\frac{x}{5} - x^3\right)\mathrm{e}^{-x^2} - \frac{1}{3}\mathrm{e}^{-(x+1)^2}$$

on an equidistant grid with $N = 10$ points can be found in Figure 4.1. The scale factor was set to $c = 1$ for $s_1(x)$ and $c = 10$ for $s_{10}(x)$. Figure 4.2 gives a convergence plot for the same function for $c = 1$ and $c = 2$ on an equidistant grid with N between 4 and 60. Exponential convergence can albeit only be observed on an interval for the number of nodes (and therefore fill distance). Depending on the value of the scaling parameter, the error increases sharply if a certain threshold is exceeded. This problem is connected to the condition of the interpolation matrix which gets worse as $h \to 0$ and we take a closer look at that soon. The first plot also shows that rescaling of the basis function plays an important role. For $c \to 0$ all entries of the interpolation matrix will tend to 1, thus rendering it singular. In the contrary for $c \to \infty$ we have $A_{k,k} \to 1$ and $A_{k,\ell} \to 0, k \neq \ell$, i.e. we get the unit matrix which is easily invertible but also leads to sharp peaks of the interpolant.

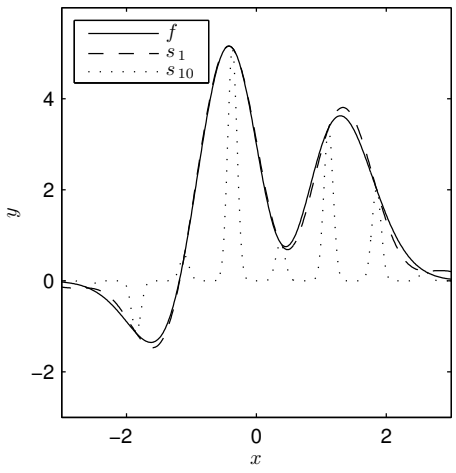

FIGURE 4.1. Influence of the shape parameter in Gaussian RBF interpolation on the quality of the approximation of s to f.

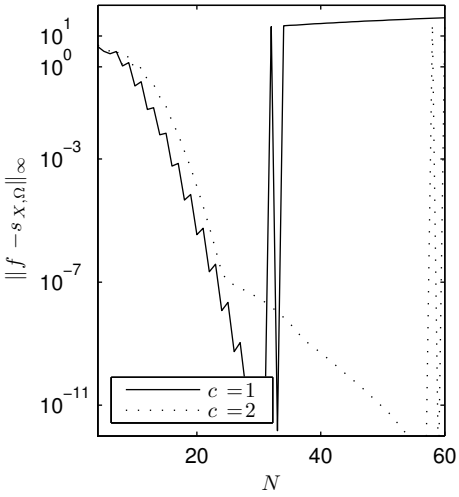

FIGURE 4.2. Exponential convergence of Gaussian RBF interpolation until numerical instability occurs.

The spectral convergence is a consequence of stricter estimates for the constants of Theorem 72 which are also based on the properties of the domain Ω and only domains Ω in form of a cube are possible.

The error estimates so far were given for interpolating functions from the native space. However, in practice it is often not easy to decide if a given function is in the native space of some basis function. Sometimes this can be circumvented by linking the native space to a classic function space, like a Sobolev space, [**19**]. On the other hand, the interpolation often also converges for functions that are not from the native space, although error estimates might not be available in that case. For univariate interpolation with the Gaussian the following result holds.

THEOREM 74 ([**73**, Theorem 4.1]). *Let $\{x_k\}_{k=1}^N$ be equally spaced nodes on the interval $[-1,1]$. Interpolation with Gaussian basis functions is convergent for functions that are analytic inside the Runge region of polynomial interpolation, i.e. inside the region encircled by*

$$2\log 2 = \mathrm{Re}\big((z+1)\log(z+1) - (z-1)\log(z-1)\big), \quad z \in \mathbb{C}.$$

If polynomial interpolation does not converge for a function f, neither will interpolation with Gaussian basis functions.

The theorem is an expression of the deep connection of polynomial and RBF interpolation. For the limit $c \to 0$ of the shape parameter the Gaussian interpolant converges to a polynomial interpolant [**24**]. Although the result is only stated for the univariate case it assures us that interpolation with Gaussians is a reasonable method also for functions not from the native space.

3.2.1. *Stability.* We already observed that numerical instabilities occur in RBF interpolation if the fill distance gets too small. The reason is that the condition of the interpolation matrix gets worse and the linear system can not be solved accurately on a computer. Now we want to look closer at the problem. Therefore, we need another measure for the distribution of node points in a domain Ω.

DEFINITION 75 (Separation Distance, [**91**, Definition 4.6]). Given a set of nodes $X = \{x_k\}_{k=1}^N$ the **separation distance** is defined as

$$q_X := \frac{1}{2} \min_{\ell \neq k} \|x_\ell - x_k\|_2 .$$

DEFINITION 76 (Quasi-Uniform Set, [**91**, Definition 4.6]). Let $\Omega \subset \mathbb{R}^d$ and $X = \{x_k\}_{k=1}^N \subset \Omega$ be a set of nodes with fill distance $h_{X,\Omega}$ and separation distance q_X. The set X is called **quasi-uniform** with respect to a constant $c_q > 0$ if

$$q_X \leq h_{X,\Omega} \leq c_q q_X$$

holds.

The separation distance is half the distance between the two closest pairwise disjunct points. In comparison to the fill distance $h_{X,\Omega}$ which is a measure how well Ω is covered with data points, q_X is suited much better for stability analysis because only two very close points can make the interpolation problem ill-conditioned. Fill distance and separation distance can be very different for a given set of nodes and domain Ω. Of course for any given Ω and X we find a constant c_q such that the inequality in the definition of quasi-uniform is fulfilled. However, if we consider a series of sets X that fill out Ω more and more we demand that all sets are quasi-uniform with respect to the same constant c_q. For example for $\Omega = [0,1]^d$ the equidistant grid $X_h = h\mathbb{Z}^d \cap \Omega$, $h > 0$ has separation distance $q_X = h/2$ and fill distance $h_{X_h,\Omega} = \sqrt{d}h/2$, so the quasi-uniform constant is $c_q = \sqrt{d}$. This property of a set of nodes will play a role again when we discuss implementation details later.

For now we come back to the problem of stability which we already linked to the condition of the interpolation matrix $A_{\phi,X}$ which we define to be

$$\operatorname{cond} A_{\phi,X} := \frac{\lambda_{\max}(A_{\phi,X})}{\lambda_{\min}(A_{\phi,X})},$$

where $\lambda_{\max}(A_{\phi,X})$ and $\lambda_{\min}(A_{\phi,X})$ are the maximum and minimum eigenvalue of $A_{\phi,X}$ respectively. This is consistent with more general definitions involving singular values because A is symmetric and positive definite in our case.

To estimate the condition number an estimate of the two eigenvalues is needed. Therefore we assume $X = \{x_k\}_{k=1}^N$ to be a set of nodes in $\Omega \subset \mathbb{R}^d$ and Φ a positive definite function. For the maximum eigenvalue $\lambda_{\max}$ we start with Gershgorin's theorem which tells us that we find an index $\ell \in \{1, 2, \ldots, N\}$ such that

$$|\lambda_{\max} - \Phi(x_\ell - x_\ell)| \leq \sum_{\substack{k=1 \\ k \neq \ell}} |\Phi(x_\ell - x_k)|$$

holds. Using the triangle inequality and $\Phi(0) \geq \Phi(x)$ for all x from Theorem 55 we obtain

$$\lambda_{\max} \leq N\Phi(0).$$

That means the maximum eigenvalue grows like $\mathcal{O}(N)$ or if the data is quasi-uniform like $\mathcal{O}(h_{X,\Omega}^{-d})$.

It turns out that the smallest eigenvalue is much more crucial for the stability of the interpolation process. Its estimation is also more complex, we omit the details and content ourselves with stating the lower bound for the Gaussian. We define the constants

$$M_d = 12 \left(\frac{\pi \Gamma^2(d/2+1)}{9} \right)^{1/(d+1)} \text{ and } C_d = \frac{1}{2\Gamma(d/2+1)} \left(\frac{M_d}{2^{3/2}} \right)^d .$$

THEOREM 77 ([**91**, Corollary 12.4]). *Let $X = \{x_k\}_{k=1}^N$ be a set of nodes, $X \subset \Omega \subset \mathbb{R}^d$ and $\Phi = \mathrm{e}^{-c\|x\|_2^2}$. Then the minimum eigenvalue of the interpolation matrix $A_{\Phi,X}$ can be bounded by*

$$\begin{aligned}\lambda_{\min}\left(A_{\Phi,X}\right) &\geq C_d(2c)^{-d/2}\exp\left(-\frac{M_d^2}{cq_X^2}\right)q_X^d \\ &\geq C_d(2c)^{-d/2}\exp\left(-\frac{40.71d^2}{cq_X^2}\right)q_X^d.\end{aligned}$$

So $\lambda_{\min}$ decreases exponentially with q_X, which means that the condition number of $A_{\Phi,X}$ will increase exponentially with q_X. For quasi-uniform data, where q_X and $h_{X,\Omega}$ are equivalent it follows that with $h_{X,\Omega} \to 0$ the interpolation error and the condition number behave exponentially (decrease and grow respectively). This explains why numerical instability occurs for moderately fine grids already. In the estimate for $\lambda_{\min}$ the scale parameter c is present in a form that suggests that larger c could counterbalance the effect of q_X becoming small. And indeed numerical experiments show that this is, to some extend, the case. However, on the other hand the interpolation error also grows with c. Hence, a careful choice of a "good" c can balance the trade of between accuracy and stability and plays a crucial role for the practicability of the method.

3.3. Shape Parameter Optimization. As discussed in the last section and illustrated in figures 4.1 and 4.2 the shape parameter is of great importance for the interpolation process. Choosing a suitable value is not an easy task and depends on the function f that is interpolated, the input dimension d and of course mainly on the set of interpolation nodes X. There are some heuristic ad-hoc rules that can be found throughout the literature on RBF interpolation [**78**], for example

- $c = 0.815\delta$, where $\delta = (1/N)\sum_{k=1}^N \delta_k$ and δ_k is the distance between the k-th data point and its nearest neighbor;
- $c = 1.25\delta/\sqrt{N}$ where in this case δ is the diameter of the smallest ball containing all data points.

The problem with approaches of this kind is twofold. First they do not take the function f into account. The interpolation of a function with a couple of sharp peaks might benefit from a smaller shape parameter. The other problem is numerical stability. Although these rules aim at preventing the ill-conditioning of the matrix they might fail in the case of unorthodox sets of nodes (they do not take the separation distance into account) or they are too conservative and lead to a larger interpolation error than necessary. In conclusion, we are looking for a method that adapts to all influencing factors and can be feasibly computed. A classic approach is leave-one-out optimization, see Algorithm 1. The basic idea of the approach is to interpolate data $F = \{f_k\}_{k=1}^N$ on a set of nodes $X = \{x_k\}_{k=1}^N$ with one x_ℓ removed which then can be used to compute an error between the known data value f_ℓ and the interpolated value $s^{(\ell)}_{\tilde{F},\tilde{X}}(x_\ell)$, where $\tilde{F} := F\setminus\{f_\ell\}$ and $\tilde{X} := X\setminus\{x_\ell\}$. This is done for every node point x_k, $k = 1, 2, \ldots, N$ and an error vector E is obtained. Minimizing the norm of this error vector with respect to the shape parameter gives an estimate for a good choice of c. The leave-one-out approach fulfills our requirements regarding adaption to the function and numerical stability since it uses actual data and interpolates during the process. However, it is computational expensive as for every new choice of c N linear systems of size $(N-1)\times(N-1)$ have to be solved to construct $s^{(\ell)}_{\tilde{F},\tilde{X}}$ for each c. Using a classic LU decomposition, this leads to a complexity of $\mathcal{O}(N^4)$. In [**78**] an alternative formula for calculating the error vector E is derived. We trace the basic steps. For this the following notation

Algorithm 1 Leave-one-out

Input: Set of nodes $X = \{x_k\}_{k=1}^N \subset \mathbb{R}^d$, data: $F = \{f_k\}_{k=1}^N \subset \mathbb{R}$.
procedure LEAVEONEOUT(X, F)
  repeat
    Choose shape parameter c
    for $k = 1 : N$ **do**
      $\tilde{X} \leftarrow X \setminus \{x_k\}$
      $\tilde{F} \leftarrow F \setminus \{f_k\}$
      Interpolate $\tilde{F}$ on $\tilde{X}$ to obtain $s^{(k)}_{\tilde{F},\tilde{X}}$, i.e. solve $A^{(k)}\lambda^{(k)} = \tilde{F}$.
      $E_k = \left| s^{(k)}_{\tilde{F},\tilde{X}}(x_k) - f_k \right|$
    end for
  until $\|E\|_2 = \min$
end procedure

is needed: The superscript (k) to a vector always means that the k-th component is removed. For a matrix $A \in \mathbb{R}^{N\times N}$ it means that the k-th row and column are removed. It holds or such a matrix that

$$Ay = z \Rightarrow A^{(k)}y^{(k)} = z^{(k)}, \tag{42}$$

if $y \in \mathbb{R}^N$ with $y_k = 0$. Lastly, $s^{(k)}_{\tilde{F},\tilde{X}}$ is the interpolant without using the k-th point, i.e. interpolating $\tilde{F} = F^{(k)}$ on $\tilde{X} = X^{(k)}$.

THEOREM 78 ([**78**, Section 3]). *Let $X = \{x_k\}_{k=1}^N \subset \mathbb{R}^d$ be a set of nodes and $F = \{f_k\}_{k=1}^N \subset \mathbb{R}$ a set of data values. The leave-one-out error vector E for the RBF interpolation process*

$$s_{F,X}(x) = \sum_{k=1}^{N} \lambda_k \Phi(x - x_k)$$

on X is given by

$$E_\ell = \frac{\lambda_\ell}{Y_{\ell,\ell}}, \quad \ell = 1, 2, \ldots, N,$$

where $Y_{\ell,\ell}$ is the ℓ-th diagonal component of the solution to the system

$$AY = I$$

with the interpolation matrix A and the $N \times N$ unit matrix I.

PROOF. We consider the vector

$$b^\ell = \lambda - \frac{\lambda_\ell}{Y_{\ell,\ell}} Y_{:,\ell}$$

where $Y_{:,\ell}$ is the ℓ-th column of Y. The diagonal elements of Y are nonzero otherwise with (42) we would have $A^{(\ell)}Y^{(\ell)}_{:,\ell} = 0$ which would in turn imply that $Y = 0$, because $A^{(\ell)}$ is non-singular (as an interpolation matrix), a contradiction to the assumption that Y is the solution to the system $AY = I$. We find

$$\begin{aligned} Ab^\ell = A\lambda - \frac{\lambda_\ell}{Y_{\ell,\ell}} AY_{:,\ell} &= f - \frac{\lambda_\ell}{Y_{\ell,\ell}} e_\ell \\ &= \begin{pmatrix} f_1 & f_2 & \cdots & f_{\ell-1} & f_\ell - \frac{\lambda_\ell}{Y_{\ell,\ell}} & f_{\ell+1} & \cdots & f_N \end{pmatrix}^{\mathrm{T}}. \end{aligned}$$

The ℓ-th component of b^ℓ is 0 and therefore with (42) it follows that $A^{(\ell)}b^{(\ell)} = f^{(\ell)}$ and hence $\lambda^{(\ell)} = b^{(\ell)}$ because $A^{(\ell)}\lambda^{(\ell)} = f^{(\ell)}$, too. With that we conclude

$$\begin{aligned} s^{(\ell)}_{\tilde{F},\tilde{X}}(x_\ell) &= \sum_{\substack{k=1\\k\neq\ell}}^{N} \lambda_k^{(\ell)} \Phi(x_\ell - x_k) \\ &= \sum_{\substack{k=1\\k\neq\ell}}^{N} b_k^{\ell} \Phi(x_\ell - x_k) \\ &= \sum_{k=1}^{N} b_k^{\ell} \Phi(x_\ell - x_k) \\ &= (Ab^\ell)_\ell \\ &= f_\ell - \frac{\lambda_\ell}{Y_{\ell,\ell}}. \end{aligned}$$

which implies

$$E_\ell = f_\ell - s^{(\ell)}_{\tilde{F},\tilde{X}}(x_\ell) = \frac{\lambda_\ell}{Y_{\ell,\ell}}.$$

□

To compute E the solution to the full interpolation problem $A\lambda = F$ and $AY = I$ is needed which amounts to solve the large system $A(\lambda \,|\, Y) = (F \,|\, I)$. Only one factorization of A is sufficient and the complexity is down to $\mathcal{O}(N^3)$. A revised algorithm can be found in Algorithm 2. Since λ_ℓ and $Y_{\ell,\ell}$ are results of the

Algorithm 2 Leave-one-out (modified)

Input: Set of nodes $X = \{x_k\}_{k=1}^N \subset \mathbb{R}^d$, data: $F = \{f_k\}_{k=1}^N \subset \mathbb{R}$.
procedure LEAVEONEOUTMOD(X, F)
  repeat
    Choose shape parameter c.
    Solve $A(\lambda \,|\, Y) = I$.
    Get error vector through $E_\ell = \frac{\lambda_\ell}{Y_{\ell,\ell}}$.
  until $\|E\|_2 + 10\varepsilon_p \operatorname{cond} A = \min$
end procedure

factorization of the same matrix A, the robustness towards a bad condition number is not necessarily given and further precautions are needed. Rippa suggests to cut off shape parameters for which the condition number of A is larger than $1/(10\varepsilon_p)$ (ε_p machine epsilon). We use a slightly modified approach where we augment the objective function like $\|E\|_2 + 10\varepsilon_p \operatorname{cond} A$. Finally, to compute the minimum, in accordance with Rippa we use Brents Algorithm **[75]**, a derivative free method.

EXAMPLE 14. We return to Example 13. Figure 4.3 shows the optimized shape parameter in dependence of the number of points N used for interpolation. Figure 4.4 shows the interpolation error for the optimized shape parameter. The second plot shows how the optimization approach leads to a smaller error. The condition number safeguard albeit leads to a larger error if the number of interpolation nodes grows and hence the fill distance gets smaller. However, it also prevents numerical instability quiet reliably. Still, one can conclude that there is a certain range for the fill distance that is optimal with regard to the interpolation error, and shape parameter optimization is only efficient in that range. The first plot also shows how

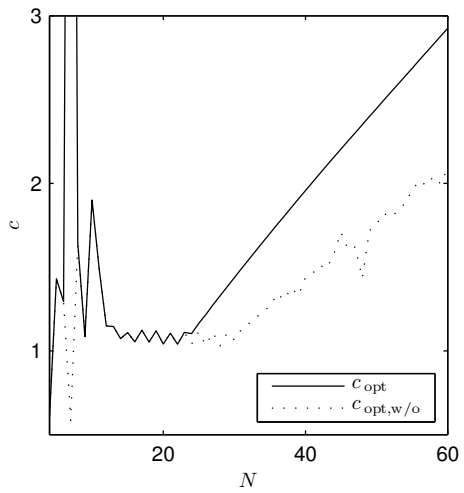

FIGURE 4.3. The shape parameter with safeguard (c_{opt}) and without ($c_{\mathrm{opt,w/o}}$).

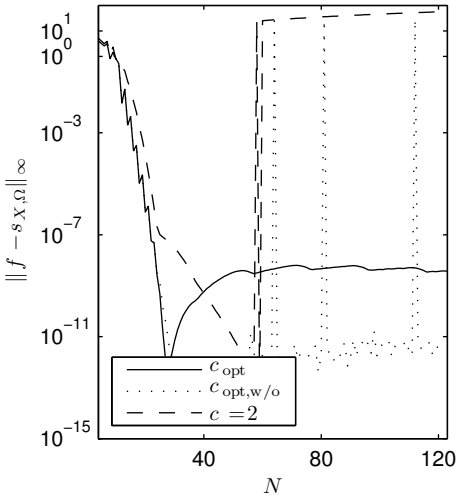

FIGURE 4.4. Convergence plot for optimized shape parameter with safeguard (c_{opt}) and without $c_{\mathrm{opt,w/o}}$.

the safeguarding kicks in once a certain number of nodes is reached. The shape parameter then continues to grow linearly with the number of nodes.

3.4. Hermite Interpolation. One way to increase the accuracy of the interpolation is to increase the number of data points. However, this might have a negative influence on the stability of the process. Another approach is to use higher order information, i.e. include information about the derivative of the function that is interpolated. As mentioned in the beginning of the chapter, we have first order derivative information available at every node in our application. Including this information was called Hermite interpolation of total degree. The extension of RBF interpolation to this problem is fairly straight forward. The functionals in question are given by

$$L_k = \delta_{x_k} \circ \mathrm{D}^{\alpha_k}, \quad k = 1, 2, \ldots, N, \quad \alpha_k \in \mathbb{N}^d, \tag{43}$$

so for a function $f : \mathbb{R}^d \to \mathbb{R}$ we have

$$L_k(f) = \mathrm{D}^{\alpha_k} f(x_k).$$

As stated in Theorem 50 the functionals L_k have to be linearly independent in the dual space of the interpolation function space for the problem to have a solution. This is the case if they are pairwise distinct so either $x_k \neq x_\ell$ or $\alpha_k \neq \alpha_\ell$ if $k \neq \ell$. In general if we have a positive definite kernel $\Phi(y, x)$, we can conclude that the functionals (43) are in the dual of the native space $\mathcal{N}_\Phi^*$. The basis functions are thus given by the according Riesz representers

$$\Phi_k = \mathrm{D}_2^{\alpha_k} \Phi(\cdot, x_k), \quad k = 1, 2, \ldots, N. \tag{44}$$

The symbol 2 indicates that the derivative is to be taken with respect to the second argument. This is natural since the data is specified with respect to x, too. The following theorem summarizes the previous ideas.

THEOREM 79 ([**91**, Theorem 16.4]). *Let $\Phi \in L^1(\mathbb{R}^d) \cap C^{2k}(\mathbb{R}^d)$ be a symmetric, positive definite kernel. If the functionals* (43)*, with $|\alpha_k| \leq k$ are pairwise distinct, then they are linearly independent over $\mathcal{N}_\Phi(\mathbb{R}^d)$.*

REMARK 31. The requirement that $\Phi \in L^1$ is necessary since the proof relies on the application of the Fourier transform to evaluate the norm of the linear combination of the basis functions.

The interpolation condition in this case is

$$L_\ell(s_{f,X}) = \sum_{k=1}^{N} \lambda_k L_\ell\big(\mathrm{D}_2^{\alpha_k}\,\Phi(\cdot, x_k)\big) = \sum_{k=1}^{N} \lambda_k\,\mathrm{D}_1^{\alpha_\ell}\,\mathrm{D}_2^{\alpha_k}\,\Phi(x_\ell, x_k).$$

Hence the entries of the interpolation matrix are $A_{\ell,k} = \mathrm{D}_1^{\alpha_\ell}\,\mathrm{D}_2^{\alpha_k}\,\Phi(x_\ell, x_k)$. This matrix is again symmetric and positive definite, and thus a unique solution always exists. This follows from

$$L_\ell^y L_k^x \Phi(y, x) = (L_\ell^y \Phi(\cdot, y), L_k^x \Phi(\cdot, x))_{\mathcal{N}_\Phi(\Omega)}$$

as long as the L_k are linearly independent.

REMARK 32. It is somewhat surprising that if we only evaluate the Hermite interpolator we already need the first derivative of the basis functions. This could be circumvented if we would use $s_{f,X}(x) = \sum_{k=1}^{N} \lambda_k \Phi_k(x, x_k)$. The entries of the interpolation matrix would be of the form $\mathrm{D}_1^{\alpha_l}\,\Phi(x_\ell, x_k)$, i.e. the derivative is only needed for creating the interpolator. However, it would not be symmetric nor positive definite in the general case and there are example node and data sets were there is no solution for the problem.

REMARK 33. It can be shown that the above described RBF interpolant is the norm minimizing approximation of the functionals $L_k(f)$ for f in the native space $\mathcal{N}_\Phi(\mathbb{R}^d)$.

3.4.1. *Partition of Unity.* A fast evaluation of the interpolation function is very important for the use in the optimal control framework, see Chapter 3. However, the naive approach involves computing the sum over all interpolation nodes for each new input x and thus has a complexity of $\mathcal{O}(N)$. Especially in the multivariate setting the number of points grows exponentially with the number of dimensions which leads to a costly evaluation but also creation of the interpolation function. It is therefore necessary to find a way to bound the computational cost, at best to $\mathcal{O}(1)$. A general way to achieve this goal is to localize the problem, i.e. somehow make the creation and evaluation of the interpolation object only depend on a subset of the domain Ω and therefore also only on a subset of X.

One root for the problem is that the Gaussian basis function we intend to use has global support and so we can not easily divide the domain Ω. One approach is partition of unity. The main idea is to cover Ω with (slightly) overlapping patches $\{\Omega_j\}_{j=1}^M$, $\bigcup_{j=1}^M \Omega_j = \Omega$ and then interpolate the data on each patch independently to obtain $s_{F,X}^j$, $j = 1, 2, \ldots, M$. For evaluation we employ a sufficiently smooth function with the following properties:

$$\omega(x) = \sum_{j=1}^{M} \omega_j(x) = 1,\ \forall x \in \Omega \text{ and } \omega_j(x) = 0,\ \forall \notin \Omega_j.$$

If we define an index function

$$J(x) = \{j \,|\, x \in \Omega_j\}$$

the global interpolant is given by

$$s_{F,X}(x) = \sum_{j \in J(x)} \omega_j(x) s_{F,X}^j(x).$$

The advantage lies in the fact that for the evaluation only the interpolation objects for the patches containing the input point x have to be evaluated. The detailed

algorithms are given in Algorithms 3 and 4 for interpolation and evaluation, respectively. Before we turn to some more implementation details we quickly state

Algorithm 3 Partition of unity, interpolation

Input: Set of nodes $X = \{x_k\}_{k=1}^N \subset \mathbb{R}^d$, data: $F = \{f_k\}_{k=1}^N \subset \mathbb{R}$.
procedure PARTOFUNITYINTERP(X, F)
Obtain estimate of Ω from X (e.g. bounding box).
Create overlapping covering $\{\Omega_j\}_{j=1}^M$ (e.g. fixed grid).
Interpolate data on each patch to obtain $s_{F,X}^j$.
end procedure

Algorithm 4 Partition of unity, evaluation

Input: $x \in \Omega$
procedure PARTOFUNITYEVAL(x)
Find all patches x lies in: $J(x)$.
Evaluate according interpolation objects: $s_{F,X}^j(x)\ \forall j \in J(x)$.
Return: $s_{F,X}(x) = \sum_{j \in J(x)} \omega_j(x) s_{F,X}^j(x)$
end procedure

some facts regarding the convergence and error of the method. First we need to define some requirements for the function ω.

DEFINITION 80 (k-stable partition of unity, [**91**, Definition 15.16]). Let $\Omega \subset \mathbb{R}^d$ be a bounded set. Additionally, let $\{\Omega_j\}_{j=1}^M$ be an open and bounded covering of Ω, which means that all Ω_j are open and bounded and $\Omega \subset \bigcup_{j=1}^M \Omega_j$. Set $\delta_j = \sup_{x,y \in \Omega_j} \|x - y\|_2$. A family of nonnegative functions $\{\omega_j\}_{j=1}^M$ with $\omega_j \in C^k(\mathbb{R}^d)$ is called a ***k*-stable partition of unity** with respect to $\{\Omega_j\}$ if

(1) $\operatorname{supp} \omega_j \subset \Omega_j$
(2) $\sum_{j=1}^M \omega_j(x) = 1\ \forall x \in \Omega$
(3) for all $\alpha \in \mathbb{N}_0^d$ with $|\alpha| \leq k$ there is a constant $c_\alpha > 0$ such that

$$\|\mathrm{D}^\alpha \omega_j\|_{L_\infty(\Omega_j)} \leq c_\alpha / \delta_j^{|\alpha|}, \quad j = 1, 2, \ldots, M.$$

The first and second point are fairly obvious, the third point is needed to get an upper bound for the error introduced through the weighting of the interpolation functions with ω. Another set of requirements is needed for the covering $\{\Omega_j\}_{j=1}^M$.

DEFINITION 81 (Regular covering, [**91**, Definition 15.18]). Let $\Omega \subset \mathbb{R}^d$ be bounded and $X = \{x_k\}_{k=1}^N \subset \Omega$ be given. An open and bounded covering $\{\Omega_j\}_{j=1}^M$ is called **regular** for (Ω, X) if

(1) There is a global constant K such that for every $x \in \Omega$ the inequality $|J(x)| < K$ holds.
(2) Every patch satisfies an interior cone condition [**91**, Definition 3.6].

The first condition ensures that the sum for the partition of unity has at maximum K terms, the second condition is necessary to apply the error estimates for the radial basis function interpolation to each patch. With the definitions in place we are able to state the error for the global partition of unity method.

THEOREM 82 ([**91**, Theorem 15.19]). *Let Ω be a cube in $\mathbb{R}^d$ and $X = \{x_k\}_{k=1}^N \subset \Omega$ a set of nodes. Further, let $\Phi = \phi(\|\cdot\|_2)$ be radial and satisfy $\left|\mathrm{D}^m \phi(\sqrt{\cdot})\right| \leq$*

$M^m, M > 0$ fixed. Let $\{\Omega_j\}_{j=1}^M$ be a regular covering for (Ω, X) and let $\{\omega_j\}_{j=1}^M$ be k-stable for $\{\Omega_j\}_{j=1}^M$. The interpolation error between $f \in \mathcal{N}_\Phi(\Omega)$ and the partition of unity interpolant $s_{f,X} = \sum_j \omega_j s_{f,X_j}$ can be bounded like

$$\|f - s_{f,X}\|_{L^\infty(\Omega)} \leq \exp\left(\frac{c\log(h_{X,\Omega})}{h_{X,\Omega}}\right) |f|_{\mathcal{N}_\Phi(\Omega)}$$

if $h_{X,\Omega} < h_0$ with $c > 0$.

The theorem guarantees the same order of convergence for the interpolation using the partition of unity approach and the naive global interpolant.

Stability is in general not improved through the partition of unity method because the important factor, the separation distance, is not changed. Still, a little advantage might be that the linear systems, that have to be solved, are smaller, and more important a shape parameter optimization can be done for each subdomain independently. This way, if the nodes are not uniformly distributed, parts of the domain with a lower separation distance can be separated from parts with a higher one. Different shape parameters can be applied and finer adjustment to the node set is possible.

The goal of the partition of unity approach is to reduce the computational effort to build and especially evaluate the interpolant. The main ingredient for a concrete implementation is a data structure for the node points which links them also to the subdomains Ω_j and the functions $\{\omega_j\}_{j=1}^M$. The data structure has to handle two operations:

- Containment query, i.e. given an input point $x \subset \Omega$ return all indices of the patches Ω_j x is contained in, more formally implement $J(x) = \{j \,|\, x \in \Omega_j\}$.
- Given an index j return all nodes that are in Ω_j, i.e. return $X_j = \{x \,|\, x \in \Omega_j\}$.

The second operation is only needed for evaluation whereas the first is also needed to build the data structure. Both operations have to be performed as fast as possible because both are needed for evaluation. Since we can determine the node set in our application we restrict ourselves to quasi-uniform data sets. In this case a reasonable choice for a data structure is based on the fixed-grid idea. The domain Ω is covered with rectangular, axis parallel boxes and for each box a list of the nodes, that fall into it can be saved. When the side length of the boxes and the coordinates of the edges are known the containment query can be done in $\mathcal{O}(1)$ regarding the overall number of points N.

For the partition of unity approach we need overlapping boxes or patches. We use a multidimensional data structure, thus from now on we do not use linear indices $j = 1, 2, \ldots, M$ but multi-indices, i.e. each box is identified by a multi-index $\beta = (\beta_1, \beta_2, \ldots, \beta_d)$ where β_i is the coordinate of the box in the i-th dimension and $\beta = (1, 1, \ldots, 1)$ is the lower left corner box. A two dimensional example is given in Figure 4.5.

The algorithm for creating the data structure is given in Algorithm 5. The user has to provide an overlap factor $\gamma \in (0, 0.5)$ and the approximate number of points per patch K. The 0.5 maximum for γ prevents that more than two patches can overlay in one dimension. The first step is to estimate the bounding box $B \in \mathbb{R}^d$, $b \in \mathbb{R}^d$:

$$B_i = \max\{\chi_i \,|\, \chi_i \in x,\ x \in X\} \text{ and } b_i = \min\{\chi_i \,|\, \chi_i \in x,\ x \in X\}$$

for $X \subset \Omega$ which is needed to compute the optimal size of the boxes. Next, the number of patches per dimension if they would not overlay is calculated. Given the

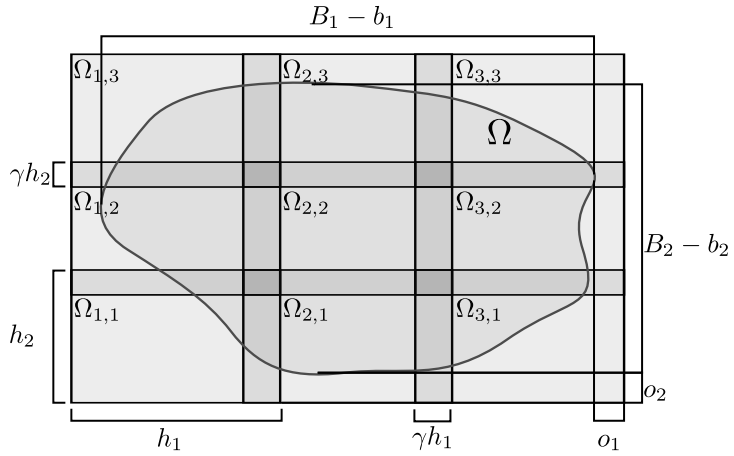

FIGURE 4.5. Two dimensional example for a domain Ω covered with overlapping boxes $\{\Omega_\beta\}$.

number of points $|X|$ and assuming uniformly distributed nodes this comes down to

$$\tilde{M} = \left\lfloor \left(\frac{|X|}{K} \right)^{1/d} \right\rfloor.$$

Now the length of one box is given by

$$h_i = \frac{B_i - b_i}{\tilde{M}}.$$

To get the number M of boxes per dimension, if the overlap is considered we first note that the length of one box will not change and stays h_i. That means, that in general the covering will not be exact and there will be an offset for the boxes on the edge. If this offset is given by o_i. The diameter of the set X plus two times the offset o_i of the covering is equal to the sum of the length of $M-1$ boxes minus the overlap, plus the length of the last full box in a row. The following holds:

$$\begin{aligned} & 2o_i + (B_i - b_i) = h_i + \sum_{j=1}^{M_i - 1} [h_i - \gamma h_i] \\ \Leftrightarrow \quad & M_i = \frac{2o_i + (B_i - b_i) - \gamma h_i}{h_i - \gamma h_i} \\ \overset{B_i - b_i = h_i \tilde{M}}{\Leftrightarrow} \quad & 2o_i = (M_i(1-\gamma) - \tilde{M} + \gamma) h_i. \end{aligned}$$

Since $0 < o_i < \frac{h_i}{2}$ (otherwise one more box would fit) on the right hand side $(M(1-\gamma) - \tilde{M} + \gamma) \in (0,1)$ and hence

$$\frac{\tilde{M} - \gamma}{1 - \gamma} < M < \frac{1 + \tilde{M} - \gamma}{1 - \gamma}.$$

Finally, we use

$$M = \left\lceil \frac{\tilde{M} - \gamma}{1 - \gamma} \right\rceil.$$

With these parameters in place the corners of the boxes are fixed and for a given point $x \in \Omega$ the indices $\{\beta_s\}_{s=1}^{2^d}$ of the boxes Ω_{β_s} it is contained in, can be calculated, see Algorithm 6. The 2^d stems from the fact that in each dimension a point can be

Algorithm 5 Fixed grid, creation

Input: Set of nodes X, overlap factor $\gamma \in (0, .5)$, approximate number of points per box K.
procedure InitGrid(X, γ, K)
Compute bounding box: $B_i \leftarrow \max_{\chi_i} X$, $b_i \leftarrow \min_{\chi_i} X$.
Compute number of boxes in each dimension, if adjacent: $\tilde{M} \leftarrow \left\lfloor \left(\frac{|X|}{K}\right)^{1/d} \right\rfloor$.
Compute number of boxes, with overlap: $M \leftarrow \left\lceil \frac{\tilde{M}-\gamma}{1-\gamma} \right\rceil$.
Compute diameter of boxes: $h_i \leftarrow \frac{B_i - b_i}{\tilde{M}}$.
Compute offset of edge boxes: $o_i \leftarrow \frac{h_i}{2}(M(1-\gamma) - \tilde{M} + \gamma)$.
for $k = 1 : |X|$ **do**
Get indices of boxes for x_k: $\{\beta_s\}_{s=1}^{2^d} \leftarrow$ `getIndex`(x_k).
$X_{\beta_s} \leftarrow X_{\beta_s} \cup \{x_k\}$, $s = 1, 2, \ldots, 2^d$
end for
end procedure

contained in a maximum of two boxes, since $\gamma \in (0, 0.5)$. The left (lower) corners of the boxes are given by

$$L_i^j = b_i - o_i + (j-1)(h_i - \gamma h_i),\ j = 1, 2, \ldots, M,\ i = 1, 2, \ldots, d \tag{45}$$

and the right (upper) corners by

$$R_i^j = b_i - o_i + h_i + (j-1)(h_i - \gamma h_i),\ j = 1, 2, \ldots, M,\ i = 1, 2, \ldots, d.$$

We start with determining the leftmost box a point $x \in \Omega$ can be in (per dimension). Therefore, we observe that we are looking for the largest index j_i^l of $L_i^{j_i^l}$ such that

$$\chi_i - L_i^{j_i^l} \leq 0.$$

Otherwise, the point would be to the left of the corner. Inserting (45) we get

$$j_i^l - 1 \leq \frac{o_i - b_i + \chi_i}{h_i - \gamma h_i}.$$

The largest j_i is thus given by

$$j_i^l = \left\lfloor \frac{o_i - b_i + \chi_i}{h_i - \gamma h_i} \right\rfloor + 1.$$

Analog considerations (we are looking for the smallest index such that the point is on the left side of a right corner) lead to

$$j_i^r = \left\lceil \frac{o_i - b_i - h_i + \chi_i}{h_i - \gamma h_i} \right\rceil + 1.$$

If all j_i^l and j_i^r are computed the multi indices for the boxes a point x is contained in, are given by all unique combinations of the j_i^l and j_i^r in each dimension. For example in two dimensions let $j_1^l = 2$, $j_2^l = 3$, $j_1^r = 2$, and $j_2^r = 2$. The box indices are then $\beta_1 = (2, 3)$ and $\beta_2 = (2, 2)$. For points close to the boundary we might get indices that are negative or larger than M, which can be safely ignored. Also in praxis if a point is exactly on the boundary of one patch, its index is ignored too, because the subdomains Ω_β are supposed to be open. The index computation is obviously $\mathcal{O}(1)$ regarding the number of points in X. However, it depends exponentially on the input dimension d and thus the fixed grid idea can only be used for small and moderate space dimensions.

Algorithm 6 Get box index for input point.

Input: x
procedure GETINDEX(x)
Compute index of first box: $L_i \leftarrow \left\lfloor \frac{o_i - B_i + \chi_i}{h_i - \gamma h_i} \right\rfloor$.
Compute index of second box: $R_i \leftarrow \left\lceil \frac{o_i - B_i + \chi_i}{h_i - \gamma h_i} \right\rceil$.
return $\{\beta_s\}_{s=1}^{2^d} \leftarrow \{(L_1, L_2, \ldots, L_d), (L_1, L_2, \ldots, R_d), \ldots, (R_1, R_2, \ldots, R_d)\}$
end procedure

THEOREM 83. *The covering $\{\Omega_j\}_{j=1}^M$ of $\Omega \subset \mathbb{R}^d$ obtained by the fixed-grid method (see Algorithm 5) is regular.*

PROOF. The covering has to comply with both conditions of definition 81. First, since the overlap $\gamma \in (0, 0.5)$ any point $x \in \Omega$ can only be contained in two boxes per dimension and therefore only in $K = 2^d$ boxes globally. Secondly, since the subdomains Ω_j are rectangular boxes, they fulfill a cone condition trivially. □

The last missing ingredient for the partition of unity approach are the functions ω_j. Every family of functions $\psi_j : \Omega \to \mathbb{R}$, $j = 1, 2, \ldots, M$ gives rise to a feasible ω_j through normalization:

$$\omega_j(x) = \frac{\psi_j(x)}{\sum_{s=1}^M \psi_s(x)}.$$

Obviously,

$$\begin{aligned}\sum_{j=1}^M \omega_j(x) &= \sum_{j=1}^M \frac{\psi_j(x)}{\sum_{s=1}^M \psi_s(x)} \\ &= \frac{1}{\sum_{s=1}^M \psi_s(x)} \sum_{j=1}^M \psi_j(x) \\ &= 1.\end{aligned}$$

The global interpolant should be at least two times continuously differentiable because we may need second order derivatives during the numerical optimal control procedure. In [**87**] a composition of a univariate and multivariate polynomials is suggested:

$$\psi_j = \begin{cases} p \circ q_j(x) & x \in \Omega_j \\ 0 & \text{else.} \end{cases} \tag{46}$$

The polynomial $p : [0, 1] \to [0, 1]$ fulfills $p(0) = 1$, $p(1) = 0$ and the spline like conditions $\mathrm{D}^k p(0) = \mathrm{D}^k p(1) = 0$, $k = 1, 2$ and thereby guarantees a continuous transition from one patch to the other. The polynomial of minimal degree that fulfill these conditions is given by

$$p(r) = -6r^5 + 15r^4 - 10r^3 + 1.$$

The $q_j : \Omega_j \to [0, 1]$ are 0 for points on the boundary of Ω_j and 1 for the midpoint. They are based on a tensor product of second order univariate polynomials. It is given by

$$q_j(x) = 1 - \prod_{i=1}^d \frac{4(x_i - L_i^j)(R_i^j - x_i)}{(L_i^j - R_i^j)^2},\ j = 1, 2, \ldots, M.$$

THEOREM 84. *Let $\{\Omega_j\}_{j=1}^M$ be a fixed-grid covering of $\Omega \subset \mathbb{R}^d$. The family of functions*

$$\omega_j = \frac{\psi_j(x)}{\sum_{s=1}^M \psi_s(x)}$$

with ψ_s from (46) is a k-stable partition of unity for any $k \in \mathbb{N}$.

PROOF. We have to check the conditions of Definition 80.

(1) Because ψ is continuous, $\psi_j(x) = 1$ for the midpoint of Ω_j, and $\psi_j(x) = 0$ for $x \notin \Omega_j$ the support of ω_j is a subset of Ω_j.
(2) The equation $\sum_{j=1}^M \omega_j = 1$ for all $x \in \Omega$ holds, because of the normalization approach.
(3) The functions ψ_s are smooth and bounded on Ω_s. This will also transfer to ω_j since $\psi_s(x) > 0$ for $x \in \Omega_s$. Additionally, $\Omega \subset \bigcup_{s=1}^M \Omega_s$ and thus $x \in \Omega$ means $x \in \Omega_s$ for at least one $s = 1, 2, \ldots, M$ which guarantees that the sum $\sum_{s=1}^M \psi_s(x) > 0$ for all $x \in \Omega$, too. Overall, there exists a constant $c_\alpha > 0$ such that the stated bound on the derivative of ω_j holds for all $k \geq |\alpha|$.

□

REMARK 34. Using the fixed grid structure and the family of functions ω_j just presented and combining it with the Gaussian basis function $\phi(x) = \mathrm{e}^{-c\|x\|_2}$ in a partition of unity method will lead to a global interpolant that retains the exponential convergence for interpolation to functions from the native space $\mathcal{N}_\phi(\Omega)$ (Theorem 82).

EXAMPLE 15. Again we return to the previous examples. Figure 4.6 compares the interpolation error for the naive (meaning without partition of unity), the Lagrange partition of unity, and the Hermite partition of unity interpolator. Figure 4.7 does the same for the error of the derivative. Unfortunately, as soon as more than one box is involved the error of the partition of unity approach increases significantly. Also there is a lot more variation in the error regarding the number of points, which would make the choice for the right number more difficult. Also it should be noted that including derivative information into the interpolation process increases the accuracy by several orders of magnitude. Especially if only one box is present, i.e. no error due to the partition of unity, the error is close to 1×10^{-15} and nearly reaches the precision limit of the computer.

Figures 4.8 and 4.9 show wall times for creation (including the shape parameter optimization) and evaluation of the naive and partition of unity interpolators. Both operations profit from the partition of unity approach. The creation time for the naive interpolator increases superlinearly, which can be expected since the unavoidable cost of solving a linear system is $\mathcal{O}(N^3)$. Linear systems are also solved in the partition of unity approach, however their size is capped at around $K \times K$ (or $K(d+1) \times K(d+1)$ in the Hermite case). It also seems that the shape parameter optimization is more stable for larger N for the partition of unity, there is less variations in the runtime. A similar result can be found for the evaluation. As expected the cost is essentially constant for the partition of unity approach as soon as a minimum number of points N is exceeded. Also interesting is that even for small N the overhead for the partition of unity (mainly containment query) is negligible for most use cases as is the cost for the Hermite approach. This might change if more input dimensions are considered since the number of basis functions and accordingly terms in the sum increases with d like $K(d+1)$.

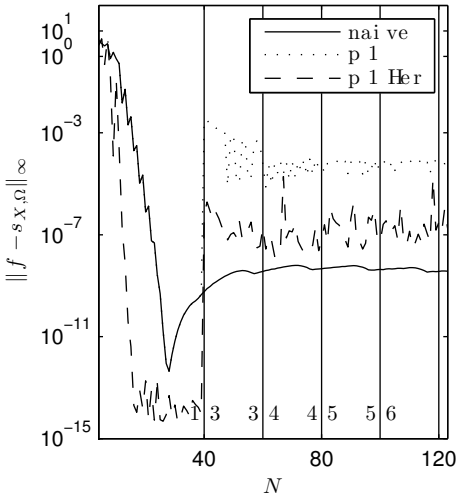

FIGURE 4.6. Interpolation error of the naive, albeit shape parameter optimized (naive) and the partition of unity interpolator (p 1, p 1 Her for the version including derivative information). For the partition of unity an overlap factor of $\gamma = 0.1$ and approximate number of points per patch $K = 20$ was used. The vertical lines and the numbers indicate a change in the number of boxes used.

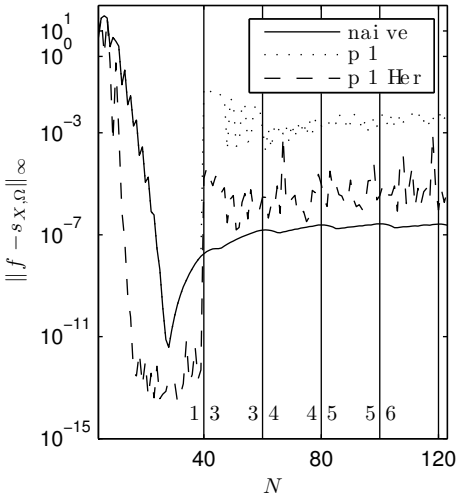

FIGURE 4.7. Error of the derivative, using interpolation without partition of unity (naive), with partition of unity (p 1), and with partition of unity and derivative information (p 1 Her).

The decreased accuracy is caused by the transition between the boxes. Near the boundary of each box the interpolation error of the according $s^j_{\tilde{F},\tilde{X}}$ is increasing because the function values of f do not tend to 0 whereas the interpolators do. Also, the ω_j introduce an additional error that adds to the error of the interpolation function. If high accuracy is of importance the direct naive implementation is to be preferred, however, one has to consider the performance bonus of the partition of unity approach. A fast, constant time evaluation is critical for our application. Therefore we will solely use the partition of unity implementation.

3.4.2. *Implementation Details.* The presented concepts and algorithms for RBF interpolation are implemented in a set of `C++` classes. The source code, a basic documentation, and examples can be found at `https://github.com/mosesx/NDInterpolator`.

All implemented interpolators are independent of input dimension and node configuration and thereby retaining the definitive features of the RBF approach. There are two main classes: `NDInterpolator` and `MultiNDInterpolator`, the first represents the concept of an interpolator with one output dimension, the second with several output dimensions, i.e. several functions that are interpolated on the same set of nodes. This approach saves memory because the set of nodes has to be saved within the interpolator object since it is needed during evaluation. All interpolators are available as naive, partition of unity, Lagrange, and Hermite versions, see Table 4.1 for an overview. All versions implement the described shape parameter optimization and support evaluating derivatives up to and including

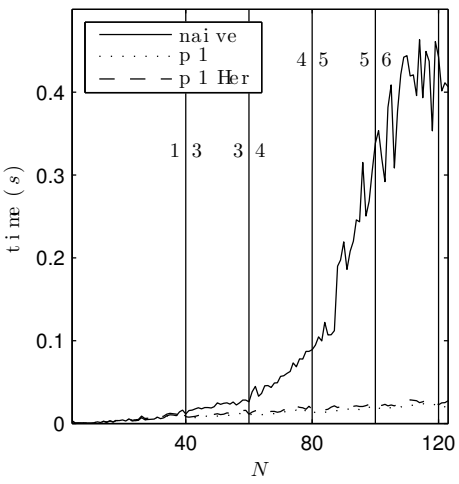

FIGURE 4.8. Wall time for the creation of the interpolator, without partition of unity (naive), with partition of unity (p 1), and Hermite with partition of unity (p 1 Her). The partition of unity was created with overlap $\gamma = 0.1$ and approximated number of points per box $K = 20$.

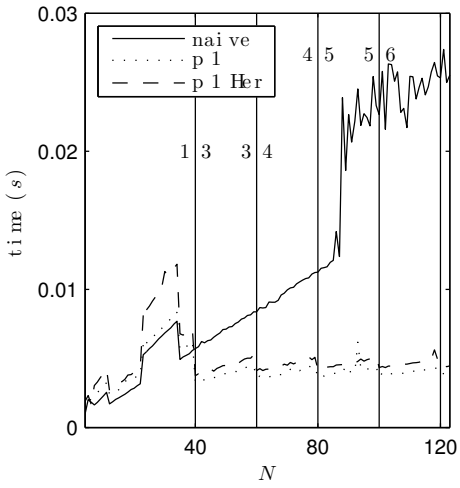

FIGURE 4.9. Wall time for 1000 evaluations of the interpolator, without partition of unity (naive), with partition of unity (p 1), and Hermite with partition of unity (p 1 Her).

TABLE 4.1. Available `C++` classes.

	$\mathbb{R}^d \to \mathbb{R}$	$\mathbb{R}^d \to \mathbb{R}^p$
naive/Lagrange	`RBFInterpolator`	`MultiRBFInterpolator`
naive/Hermite	`RBFInterpolatorH`	`MultiRBFInterpolatorH`
part. unity/Lagrange	`RBFInterpolatorPU`	`MultiRBFInterpolatorPU`
part. unity/Hermite	`RBFInterpolatorPUH`	`MultiRBFInterpolatorPUH`

second order. The implementation is also independent from the basis function used. Any radial, positive definite function could be included and in fact the inverse multiquadric is available, however not as thoroughly tested as the Gaussian.

For the evaluation and as well creation of the Hermite interpolator numerous derivatives of the Gaussian basis function $\phi = \mathrm{e}^{-c\|x\|_2^2}$ and the partition of unity weight functions ω_j have to be evaluated. For evaluation, since the basis function is radial, it can be regarded as a composition of the $\mathbb{R} \to \mathbb{R}$ function $\phi(r) = \mathrm{e}^{-cr}$ and the $\mathbb{R}^d \to \mathbb{R}$ function $x \mapsto \|x\|_2^2$. Partial derivatives

$$\mathrm{D}^\alpha \phi(\|x\|_2^2)$$

can then be obtained up to any order through multiple applications of the chain rule or more formally through the use of Faà di Bruno's formula **[13]**. In the implementation we make use of the trivial recursion

$$\mathrm{D}^k \phi(r) = (-c)^k \phi(r).$$

One has to be more careful with the basis functions of the Hermite interpolator from (44). In this case the norm computation has to be regarded as a function of two variables $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, $(x, y) \mapsto \|x - y\|_2^2$. Consequently, derivatives with

respect to χ_i and y_i might differ, for example

$$\mathrm{D}_{\chi_i} \|x-y\|_2^2 = 2(\chi_i - y_i) \neq \mathrm{D}_{y_i} \|x-y\|_2^2 = -2(\chi_i - y_i).$$

The weight function can always be regarded as functions of one variable because it only comes into play during evaluation of the interpolator. The derivative of the partition of unity sum $\sum_{j=1}^{M} \omega_j(x) s_j(x)$ involves lengthy evaluations of the product and chain rule (the ω_j itself are composite functions). The resulting polynomial expressions are evaluated using Horner's rule. Detailed formulas might be extracted directly from the source code.

A couple of third party libraries are used in the implementation. Most of them are from the boost project `http://www.boost.org/`. First of all `ublas` [**90**], a `BLAS` compatible library that provides the basic vector and matrix containers and operations on them. Next, `Multi-Array` [**27**], a dimension independent implementation of a random access container for generic objects. It is used to provide a multidimensional storage and access to the single interpolator objects in the partition of unity approach. For each box an interpolator object is instantiated independently and stored in the multi-array. The solving of the linear equation system to obtain the coefficients λ and the optimal c in the shape parameter optimization is done with the `LAPACK` (`http://www.netlib.org/lapack/`) routine `dgesv`, accessed through a bindings library implemented for the `ublas` data structures [**11**]. Lastly, the Brent algorithm for the derivative free optimization of the shape parameter is part of the boost math toolbox [**62**]. For more details see the already mentioned repository at `https://github.com/mosesx/NDInterpolator`.

4. Summary

In this chapter we give an overview about interpolation in a multivariate environment, a topic that differs from its univariate counterpart significantly due to the lack of Haar spaces for more than one dimension. We introduce some general approaches and then detail the theoretical background and some practical considerations for our method of choice: Interpolation with positive definite (radial) basis functions. They enjoy the remarkable property that they are truly dimension and grid independent, i.e. they can be used to interpolate data on any node configuration in any input dimension. More general, arbitrary functionals can be approximated if they are linearly independent in the dual space of the native space of the basis function. A fact, which we use for Hermite interpolation of total degree. Moreover, for the Gaussian basis function on which we focus it can be shown that the interpolation error converges exponentially fast to 0 with the fill distance $h_{X,\Omega}$. On the other hand the method has numerical stability issues and the condition of the problem worsens if the distance between the interpolation nodes gets smaller. Optimizing the shape parameter of the basis functions can counterbalance the instability to a certain extend, and a numerical algorithm for automatically choosing a good shape parameter is presented. Another issue is that the computational complexity increases with the number of nodes. A powerful but yet simple approach to overcome this shortcoming is partition of unity where the interpolation domain is divided in overlapping subdomains and the interpolation problem can be solved on each subdomain independently. The global interpolant is then synthesized by "gluing" the individual results from each subdomain together with a smooth weighting function. With this method the evaluation time can be regarded as constant with respect to the number of nodes used for interpolation, a property that is very important for the application in optimal control.

CHAPTER 5

Model Order Reduction in the Context of Optimal Control

1. Introduction to Model Order Reduction

For the practical application of optimal control accurate models with small prediction error are needed to enable the computation of a (feedback) control that is close or at least consistent with the true optimal control for the process under consideration **[64]**. The desired accuracy can often only be provided by large scale nonlinear models which make solving the NLP derived via the multiple shooting approach a time consuming affair. Additionally, real world systems are subject to random perturbations and as such the prediction of any model, accurate as it may be will only be an estimation of the true state. Therefore a control action that is solely based on a precomputed optimal control trajectory is bound to fail where the magnitude of the failure depends on the system at hand and its sensitivity with respect to perturbations. Depending on this sensitivity and the control horizon it may be necessary to incorporate measurements of the real system state into the mathematical model more or less frequently.

One approach to do so is nonlinear model predictive control (NMPC), **[23, 16]**. To this end assume that we have a mathematical formulation of an optimal control problem on a control horizon $[0, T]$ for a real world system given. Denote the mathematical state variable with $x(t) \in \mathbb{R}^{n_x}$, the control with $u(t) \in \mathbb{R}^{n_u}$, and parameters $p \in \mathbb{R}^{n_p}$. Further let $\mathrm{D}_t\, x(t) = f(x(t), u(t), p)$ be the model equation. The real world system state $\tilde{x}(t)$ and parameters $\tilde{p}$ can be monitored or sampled with a sampling frequency $\delta \in \mathbb{R}$ and we write $\tilde{x}_i = \tilde{x}(i\delta)$ and $\tilde{p}_i = \tilde{p}(i\delta)$, $i = 1, 2, \ldots$ for the measurements of the state and the parameters at time $i\delta \in [0, T]$ (assuming they are all measurable). Assume that we have arrived at the i-th sampling interval and $\tilde{x}_i$ and $\tilde{p}_i$ are available. Now the problem

$$\begin{aligned} &\min J(x(t), u(t), \tilde{p}) \\ \text{s.t. } &\mathrm{D}_t\, x = f(x, u, \tilde{p}_i), \\ &x(i\delta) = \tilde{x}_i, \end{aligned}$$

is solved on the remaining time horizon $t \in [i\delta, T]$ to obtain $u^*(t)$. This control is applied to the system until the next sampling point is reached, i.e. for $t \in [i\delta, (i+1)\delta]$. Now a new measurement is performed and a new control problem with $\tilde{x}_{i+1}$ and $\tilde{p}_{i+1}$ as initial values and parameters gets solved. This procedure is repeated until T is reached. The final time T might be updated too, in each step, according to $T = T + \delta$ to represent continuous operation.

Crucial for the application of NMPC is that the individual optimal control problems can be solved relatively fast. Relatively here means with respect to the sampling time δ and the system's dynamic. Let $\tau < \delta$ be the (maximum) computation time that is needed to solve the optimal control problem. We need that

$$\|\tilde{x}(i\delta + \tau) - \tilde{x}_i\|$$

is small enough so that the prediction based on $x(i\delta) = \tilde{x}_i$ is still a valid approximation of the real initial system state. A key strategy is to exploit that the optimal control problems on subsequent sampling intervals are close to each other.

The computation time spent for solving an optimal control problem with the multiple shooting approach is divided among two main operations: Solving the NLP and solving the initial value problems for each multiple shooting interval. Due to the added continuity constraints at the multiple shooting nodes the size (in terms of number of optimization variables) of the NLP can get rather large. One step of an interior point NLP solver mainly consists of solving a linear system and doing a line search. For optimal control problems the matrices involved (gradients and Hessians of the constraints) are sparse because only for adjacent multiple shooting nodes the optimization variables depend on each other so that derivatives are often zero. Also, the continuity constraints are linear and thus second order derivatives are zero. This structural information can be exploited **[57, 58]** and also enables the use of efficient (sparse) matrix algorithms and solving the NLP is usually not the most time consuming part in a multiple shooting approach.

For most optimal control problems solving the initial value problems contributes far more to the total computation time. This is especially true if nonlinear stiff models are used because for a reasonable accuracy many time steps of the integrator are needed. Additionally in a typical BDF method a nonlinear system of equations has to be solved, often by some variation of Newton's method which increases the cost even more.

REMARK 35. We are not going to worry about storage efficiency and assume that all the problems we are dealing with can be fitted into the memory and solved online without additional trickery.

At this point model reduction comes into play. The main goal is to decrease the evaluation cost of the model without losing to much of its descriptive capabilities. In this case we would like to decrease the stiffness of the model to facilitate a faster integration. Model reduction can be divided into model order reduction which aims at decreasing the dimension of the state space and model simplification which tries to simplify the evaluation of the model equations **[64]**. Both approaches can be combined and essentially strive to capture the most important features of the dynamic process at the cost of an error in the reduced compared to the full model. The trade off between lost accuracy and benefits of the reduced model always has to be considered and carefully balanced depending on the application at hand.

2. Model Order Reduction

We will exclusively focus on model order reduction. At its core lies the reduction of the state space dimension, however, we will later see, that the approach we are favoring will also decrease the stiffness of the initial value problem(s).

Many different methods are proposed to reduce the order of a model **[64]**, still the underlying principle is general and does not depend on the concrete method. They all aim at partitioning the state variables after their contribution to the systems dynamics. A probably nonlinear coordinate transformation may facilitate the partitioning by making the division between important and non-important contributions more obvious. Model reduction methods often differ in how they measure "contribution". To describe the basic approach, we introduce the control system

$$\mathrm{D}_t\, \tilde{z} = \tilde{f}(\tilde{z}, u),\ \tilde{z}(0) = \tilde{z}_0 \tag{47}$$

with state $\tilde{z}(t) \in \mathbb{R}^{n_z}$ and control $u(t) \in \mathbb{R}^{n_u}$ essentially bounded on $[0, T]$. The right hand side $\tilde{f} : \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_z}$ is assumed to be in C^∞. In that case the

initial value problem (47) has a unique solution $\tilde{z}$, see Theorem 14. The general approach to model order reduction can be summarized in the following steps [**64**]:

(1) Find a diffeomorphism $T : \mathbb{R}^{n_z} \to \mathbb{R}^{n_z}$ that maps $\tilde{z}$ via
$$\tilde{z} - \tilde{z}^* = T(z) \Leftrightarrow z = \tilde{z}^* + T^{-1}(\tilde{z}),$$
onto the new state $z(t) \in \mathbb{R}^{n_z}$, where $\tilde{z}^*$ is a possibly nonzero set point. The aim of this coordinate change is to separate directions in the phase space of (47) that have strong contributions to the dynamics from those that only contribute in a minor way.

(2) Decompose the new state space into $x(t) \in \mathbb{R}^{n_x}$ and $y(t) \in \mathbb{R}^{n_y}$ such that $z = (x, y)$ and $n_z = n_x + n_y$. Here x will play the role of the dominant states, also called *reaction progress variables*.

(3) Assemble new dynamic systems for x and y from
$$\mathrm{D}_t\, z = \big(\mathrm{D}_z\, T(z)\big)^{-1} \tilde{f}(\tilde{z}^* + T(z), u)$$
and obtain
$$\begin{aligned} \mathrm{D}_t\, x &= f(x, y, u), \quad x(0) = \xi, \\ \mathrm{D}_t\, y &= g(x, y, u), \quad y(0) = \eta. \end{aligned}$$
The smoothness of the right hand sides $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \to \mathbb{R}^x$ and $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$ is determined by the smoothness of T and T^{-1}. Since T is assumed to be a diffeomorphism, at least we can expect that the differential equations for x and y are properly defined and uniquely solvable.

(4) Eliminate the dynamic equation for y by one of the following methods:

Truncation: Set $y = 0$ for the reduced dynamic system
$$\mathrm{D}_t\, \tilde{x} = f(\tilde{x}, 0, u), \quad \tilde{x}(0) = \xi$$
with $\tilde{x} \approx x$ and state space dimension n_x.

Residualization: Set $\dot{y} = 0$ to obtain the differential-algebraic system
$$\begin{aligned} \mathrm{D}_t\, \tilde{x} &= f(\tilde{x}, \tilde{y}, u), \quad \tilde{x}(0) = \xi, \\ 0 &= g(\tilde{x}, \tilde{y}, u). \end{aligned}$$
The dimension of the model is not reduced.

Slaving: Obtain a map $\tilde{y} = \phi(x, u)$ either from the residualization approach by solving the algebraic equation explicitly or through an independent method. Using
$$\mathrm{D}_t\, \tilde{x} = f(\tilde{x}, \phi(\tilde{x}, u), u), \quad \tilde{x}(0) = \xi$$
leads to a reduced model with state space dimension n_x.

Remark 36. The crucial ingredient to the above algorithm is the availability of the map T. Deducing it from the (nonlinear) system equation (47) alone is usually not possible. In some cases physical insight into the process that is modeled or other additional external knowledge can be used to partition the state $\tilde{z}$ into (x, y). Quite common is the usage of information gained from sampling the system (47) systematically for different inputs $u(t)$. This approaches often lead to linear maps T.

Several model order reduction methods have been proposed in the past and we proceed to give a short description of some of them.

Nonlinear balancing is an analytical method based on the theory of nonlinear Hankel operators and their attributed singular value functions aimed at obtaining a nonlinear map T [**26**]. In practice, empirical balancing that incorporates samples

of the systems behavior for different inputs and initial values can be used [**36**]. In that case the map T is linear.

Proper orthogonal decomposition (POD) is based on sampling representative trajectories of (47), called snapshots [**45**]. Similarly to balancing a linear transformation matrix T is obtained using singular value decomposition. For a focus in the context of optimal control see [**49**].

Additional approaches include combinations of balancing and POD [**50**] and moment matching for nonlinear systems [**2**].

3. Slow Invariant Manifolds for Model Reduction

The method for model order reduction that we are using is based on an optimization principle and is applicable for control systems that involve processes evolving on significantly different time scales, that is, some of the states "move" considerably faster than others (on parts of the overall time interval). It was developed in the context of the simulation of combustion and other complex chemical reactions but is independent from the underlying problem and can be applied to general nonlinear systems based on ordinary differential equations.

Dirk Lebiedz developed the method, [**52**]. Later it was refined theoretically and numerically in [**77, 54, 56**]. The following is mainly based on the PhD thesis of Jochen Siehr [**81**].

In Chapter 2 on singularly perturbed systems we saw that the fast modes relax to an invariant manifold in the state space, that is parametrized by slow modes (or reaction progress variables). The fast states (or *unrepresented species*) relax into the direction of the manifold much faster than in any other direction (for example to an equilibrium point). We already hinted that if one has this slow manifold available it could be used for model reduction by means of a slaving approach. For singular perturbed systems the existence of the manifold and its properties could be characterized accurately, often in terms of the small parameter ε. For general nonlinear systems without ε explicitly present the situation is more involved.

Consider the system

$$\mathrm{D}_t\, z(t) = f\big(z(t)\big), \quad z(0) = z_0,\ t \in [0,T] \tag{48}$$

with $z(t) \in \mathbb{R}^{n_z}$ and f smooth. Further assume the solution of the initial value problem exists for all $t \in [0,T]$. Time scales present in a system can be identified via analyzing the eigenvalues of the Jacobian

$$J_f(t) = \mathrm{D}_z\, f(z)\big|_{z=z(t)}.$$

Definition 85 (Time Scale, [**51**, Section 6]). Denote the eigenvalues of $J_f(t)$ with $\lambda_i(t)$, $i = 1, 2, \ldots, n_z$. Then local time scales for each state according to the linearized system can be defined by

$$\tau_i(t) := \frac{1}{|\mathrm{Re}\,\lambda(t)|}.$$

Fast time scales are connected to large eigenvalues of $J_f(t)$ and vice versa. One can speak of time scale separation if there is one or more significant gap(s) in the ordered series

$$\tau_1 \leq \tau_2 \leq \cdots \leq \tau_{n_z}.$$

Remark 37. In the context of singularly perturbed systems n_y rows of the Jacobian $J_f(t)$ get scaled by $\frac{1}{\varepsilon}$ and there will be n_y according eigenvalues of order $\mathcal{O}(\frac{1}{\varepsilon})$ which in turn result in n_y time scales of order $\mathcal{O}(\varepsilon)$ compared to the time scales of the slow modes which are of order $\mathcal{O}(1)$ with respect to ε. This emphasizes the role of the parameter ε as an explicit representation of the time scale separation.

The manifold object we try to identify is often called *slow invariant manifold* (SIM). *Slow* because if the fast states have relaxed close to the manifold they move along it on the same time scale as the slow variables. *Invariant* refers to the (positive) invariance (see Definition 9) of the manifold with respect to the flow generated by (48). An additional assumption that is needed for model reduction is that the SIM is attracting, i.e. all trajectories starting δ-close to the SIM will converge towards it. This is more restrictive than what we allowed for in the case of singularly perturbed systems where we assumed for the manifold to be normally hyperbolic (Assumption A9, page 12). One is hard pressed to find a formal and well accepted definition of SIMs for systems not explicitly in singular perturbed form **[33, 34]**. Often the existence of a manifold parametrized by the slow states is assumed **[82]**, in that case the defining equation

$$y = h(x)$$

can be differentiated with respect to t to obtain

$$\begin{aligned} & \mathrm{D}_t\, y = \mathrm{D}_x\, h(x)\, \mathrm{D}_t\, x \\ \Leftrightarrow \quad & g(x,y) = \mathrm{D}_x\, h(x) f(x,y) \\ \Leftrightarrow \quad & 0 = \mathrm{D}_x\, h(x) f(x,y) - g(x,y). \end{aligned}$$

The last equation can be solved approximately using functional iteration (either analytically, if feasible, or numerically). Another approach is to assume that the general nonlinear system

$$\mathrm{D}_t\, z(t) = f\big(z(t)\big)$$

can be transformed into singular perturbed form

$$\begin{aligned} \mathrm{D}_t\, x(t) &= f\big(x(t), y(t)\big), \\ \varepsilon\, \mathrm{D}_t\, y(t) &= g\big(x(t), y(t)\big), \end{aligned}$$

and corresponding slow and fast states can be identified in the original state vector $z(t)$. This transformation could be incorporated into the map T from the general model reduction pattern above. The transformation to a singular perturbed system is usually not carried out (because the mapping is not available) but merely assumed to exist for theoretical justification. In general it is not clear under which circumstances a nonlinear system can be transformed into a singular perturbed form. In **[63]** geometric conditions on a mapping for control systems where an ε can be identified in the right hand side are derived. In practice the transforming approach stays merely a theoretical justification and the SIM is sought after in the original state coordinates.

A central question is how the reaction progress variables might be chosen. A time scale analysis using the notion of time scale introduced above will reveal if a significant time scale separation is present (for certain example trajectories) and how many slow and fast variables there are, i.e. the number of reaction progress variables should correspond to the number of large time scales. For further analysis the contribution of the individual states $z_i(t)$, $i = 1, 2, \ldots, n_z$ to the slow and fast directions has to be considered. There are several methods available, all aiming at identifying the contribution of a species to the slow and fast time scale movements. Locally, this can be done via decomposing the Jacobian $J(t)$ for example using Schur decomposition or singular value decomposition (SVD) **[43, 53]**. This way, linear transformations are obtained that (partially) decouple the states from each other and each new state has a time scale associated with it. Partitioning the eigen- or singular values can be used to define slow and fast subspaces. Now, from the transformation matrices the contribution of each state in the original coordinates to this slow and fast subspaces can be computed and based on that the

state space might be divided. For an overview on further approaches see [**81**]. In practice, reaction progress variables are often chosen based on physical insight into the process, that is modeled or on considerations concerning the use of the reduced model.

Further complications arise for example for systems with several time scales, which means the eigenvalues of $J(t)$ feature more than two clusters. In that case the state variable z can be partitioned into several hierarchical sets of fast variables. To this end, I_j are index sets with $I_1 \subset I_2 \subset \cdots \subset I_{n_f} = \{1, 2, \ldots, n_y\}$ where $n_f \leq n_y$ is the number of different fast time scales. Also, $I_j \cap I_i = \emptyset$ for $j \neq i$. We assume that the I_j are ordered according to the time scale magnitude with I_1 containing the indices of the fastest and I_{n_f} the indices of the slowest variables. Now there exist manifolds M^j, $M^1 \subset M^2 \subset \cdots \subset M^{n_f}$ such that y_i, $i = 1, 2, \ldots, n_y$ relax onto this manifolds subsequently. In other words the manifold M^1 is parametrized by the slow states x and all, except the most fastest, y_i. We have

$$y_i = h_i^1(x, y_k), \quad i \in I_1,\ k \in \bigcup_{s=2}^{n_f} I_s,$$

ongoing

$$y_i = h_i^2(x, y_k) \quad i \in I_1 \cup I_2,\ k \in \bigcup_{s=3}^{n_f} I_s,$$

and finally

$$y_i = h(x), \quad i = 1, 2, \ldots, n_y.$$

Example 16. Imagine the system

$$\begin{aligned} \mathrm{D}_t\, x &= -2x + y_1 \\ \varepsilon_1\, \mathrm{D}_t\, y_1 &= -2y_1 + 1y_2 + x \\ \varepsilon_2\, \mathrm{D}_t\, y_2 &= -1y_2 + y_1 \end{aligned} \tag{49}$$

being given with $0 < \varepsilon_2 \ll \varepsilon_1 \ll 1$. Obviously, for all initial values $x(0)$, $y_1(0)$, $y_2(0) \geq 0$ the origin in the phase space $\mathbb{R}^3$ is a stable equilibrium point. The state y_2 will relax first on a 2-dimensional manifold M^1 in the 3-dimensional state space parametrized by x and y_1. Then y_1 and y_2 will relax onto a 1 dimensional manifold M^2 parametrized by x before finally the 0-dimensional equilibrium point is reached.

Figure 5.1 shows example trajectories in the state space. Although the plot is crowded, on closer inspection one can see that in y_2 direction all trajectories move to an inclined plane and only then on this plane into the direction of the central diagonal which represents the 1-dimensional manifold.

The difficulty with multiple time scales is how to chose the reaction progress variables. Depending on the accuracy demands more or less reaction progress variables might be used. One can also think of adaptive strategies, but research in this direction is still ongoing.

3.1. Formulation as Optimization Problem. After introducing the SIM for general nonlinear systems and discussing some of its properties the aim now is to identify the manifold at least pointwise, i.e. given a point $x \in \mathbb{R}^{n_x}$ what is the corresponding point $y = h(x)$ on the manifold. To this end we assume that

- the system features two distinct time scales (slow and fast),
- the overall state vector $z(t) \in \mathbb{R}^{n_z}$ is decomposed into reaction progress variables $x(t) \in \mathbb{R}^{n_x}$ and unrepresented fast states $y(t) \in \mathbb{R}^{n_y}$ such that $z(t) = \big(x(t), y(t)\big)^{\mathrm{T}}$, and

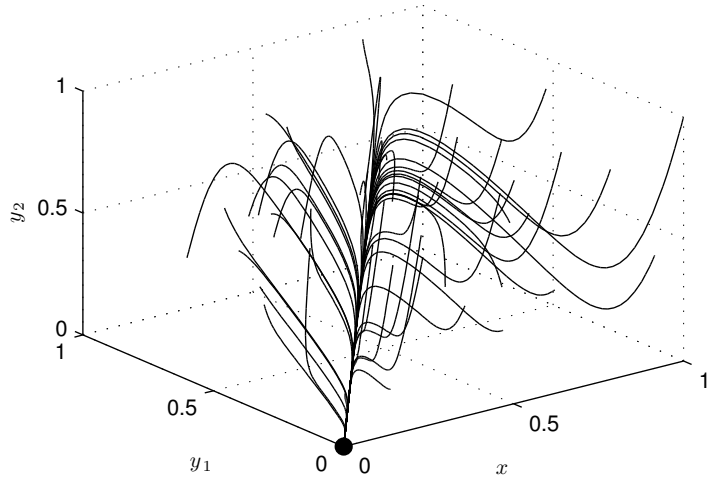

FIGURE 5.1. Random trajectories of the three time scale system (49) with $\varepsilon_1 = 0.1$ and $\varepsilon_2 = 0.01$. The equilibrium point is marked in the lower middle corner.

- a SIM M parametrized by x exists over a domain $D_x \subset \mathbb{R}^{n_x}$ of interest and there is a $h : D_x \to \mathbb{R}^{n_y}$, at least twice continuously differentiable, such that $M = \{(x, y) \,|\, y = h(x),\ x \in D_x\}$.

The pointwise approximation problem will be formulated as an optimization problem. The basic idea is that trajectories on the slow manifold fulfill a minimum principle of some kind. For example for chemical reaction systems it was argued that trajectories on the SIM minimize chemical entropy production, **[52]**.

We aim at approximating $y^* = h(x)$ and regard the problem

$$
\begin{aligned}
&\min_{x(t),y(t)} \Theta\big(x(t), y(t)\big) \\
\text{s.t. } &\mathrm{D}_t\, x(t) = f\big(x(t), y(t)\big), \\
&\mathrm{D}_t\, y(t) = g\big(x(t), y(t)\big), \\
&x(t^*) - r = 0, \\
&v\big(x(t), y(t)\big) = 0, \\
&w\big(x(t), y(t)\big) \le 0, \\
&t^* \in [0, T].
\end{aligned}
\tag{50}
$$

Given an objective functional $\Theta : C^1[0,T] \times C^1[0,T] \to \mathbb{R}$ the solution of the problem are trajectories since the differential equations enter the problem as constraints. The functions v and w may be used to constrain the search space further. For example, often for mathematical models of chemical reactions negative values for the states are prohibited as well as certain mass balances have to be fulfilled. Additionally, the reaction progress variables are fixed at some time point t^* to a value r. Since we want to approximate $y^* = h(x)$ we use $y^* = h\big(x^*(t^*)\big) = h(r)$, where $x^*(t)$ is the solution of problem (50).

There have been some suggestions for the choice of Θ, **[76]**, often based on the idea that trajectories on the SIM have minimal curvature. Let

$$J(t) = [\mathrm{D}_*\, f\big(x(t), y(t)\big), \mathrm{D}_*\, g\big(x(t), y(t)\big)]$$

be the Jacobian of the system and

$$F(t) = \begin{pmatrix} f\big(x(t), y(t)\big) \\ g(x(t), y(t) \end{pmatrix}$$

the complete right-hand-side vector. We are using

$$\Theta\big(x(t), y(t)\big) = \int_0^T \|J(t)F(t)\|_2^2 \,\mathrm{d}t \tag{51}$$

and the "local" version

$$\Theta\big(x(t), y(t)\big) = \|J(t^*)F(t^*)\|_2^2 . \tag{52}$$

The term inside the norm can be interpreted as the second derivative of $\big(x(t), y(t)\big)$ with respect to t. Especially the first choice resembles a measure of the length of the curves $\big(x(t), y(t)\big)$, $t \in [0, T]$, which is expected to be minimal for trajectories on the SIM, **[55]**.

Another parameter that has to be chosen is t^*. Theoretically, the value of the reaction progress variables can be fixed at any point in time within $[0, T]$, however, there are two prevalent choices, namely $t^* = 0$ (*forward mode*) and $t^* = T$ (*reverse mode*). If $t^* = 0$ is used, this amounts to asking for initial values for $y(t)$ that minimize the curvature of the trajectory $\big(x(t), y(t)\big)$. We saw so far that this would mean that the initial fast transient for $y(t)$ would be eliminated since it would cause a large contribution to the objective function. The choice $t^* = T$ is justifiable in a similar way: Going backward in time solutions that are not starting on the manifold are usually unstable, which again would lead to large contributions to Θ. Only if $y(T)$ is already on the SIM, $y(t)$, $t < T$ will stay on it since it is assumed to be invariant. The integral version emphasizes the invariance property of the manifold since whole trajectory pieces are regarded that have to fulfill the differential equation.

3.2. Application to Singularly Perturbed Systems. Consider the singularly perturbed system

$$\begin{aligned} \mathrm{D}_t\, x &= f(x, y), \\ \varepsilon\, \mathrm{D}_t\, y &= g(x, y), \end{aligned} \tag{53}$$

that we dwelt on in Chapter 2 for a bit. Under certain circumstances a SIM exists and on a proper domain there is a function $y = h(x, \varepsilon)$ that defines the SIM. The assumptions we need are collected in Section 1 of Chapter 2. We are going to assume here that all eigenvalues of the Jacobian $g_y(t)$ of g with respect to y have negative real part (A6). If we apply the local objective (52) to system (53) we obtain (assuming $\varepsilon > 0$)

$$\begin{aligned} \|J(t^*)F(t^*)\|_2^2 &= \left\| \begin{pmatrix} \mathrm{D}_1 f & \mathrm{D}_2 f \\ \frac{1}{\varepsilon}\mathrm{D}_1 g & \frac{1}{\varepsilon}\mathrm{D}_2 g \end{pmatrix} \begin{pmatrix} f \\ \frac{1}{\varepsilon} g \end{pmatrix} \right\|_2^2 \\ &= \left\| (\mathrm{D}_1 f) f + \frac{1}{\varepsilon}(\mathrm{D}_2 f) g \right\|_2^2 + \underbrace{\left\| \frac{1}{\varepsilon}(\mathrm{D}_1 g) f + \frac{1}{\varepsilon^2}(\mathrm{D}_2 g) g \right\|_2^2}_{J_g} . \end{aligned} \tag{54}$$

The second term in the last line contains an expression that is equal (except scaling with $\frac{1}{\varepsilon}$) to the condition of the zero derivative principle from Section 1.2 (Chapter 2) for order $m = 1$. There, Theorem 12 proposes that states y^* that fulfill

$$J_g = \frac{1}{\varepsilon}(\mathrm{D}_1 g) f + \frac{1}{\varepsilon^2}(\mathrm{D}_2 g) g = 0$$

are approximations $y^* = h(r) + \mathcal{O}(\varepsilon^2)$, where r is the value the reaction progress variable is fixed to. In other words the (unique) y^* that minimizes J_g (read, makes

it zero) in the objective function (54) is the second order approximation of the point on the SIM for a fixed x. The first term in the last line of (54) has no apparent explanation in the setting of singularly perturbed systems. It is minimized for y^* that lead to less variation in the slow variables. Note that because of the scaling with $\frac{1}{\varepsilon}$ the second term in general will be more significant for ε small.

The global objective function (51) is a based on minimizing curvature, as already mentioned above. In terms of singularly perturbed systems we aim at minimizing the contribution of the boundary layer correction. Likewise to the local formulation, points close to the manifold will minimize J_g in (54). The difference lies in the fact that actual pieces of trajectories are regarded and these pieces of trajectories have to be close to the manifold in order to minimize the objective. This notion is based on the invariance of the manifold. If we find a point y^* on the manifold the trajectory through that point will also be on the manifold and thus minimize the integral.

For stable systems where the Jacobian g_y only has eigenvalues with negative real parts this setting suggests to choose $t^* = T$, i.e. the reverse mode. That way we have to solve a final value problem, which can be transformed to an initial value problem on a reversed time scale $s = T - t$. We have

$$\begin{aligned} \mathrm{D}_s\, x &= -f(x,y), \quad x(0) = r \\ \varepsilon\, \mathrm{D}_s\, y &= -g(x,y), \quad y(0) = y^*. \end{aligned}$$

On this new time scale the eigenvalues of g_y have positive real parts and the boundary layer solutions $X(\tau)$, $Y(\tau)$ become unstable which means for every $y(0) = y^*$ that is not on the SIM the solution $y(s)$ will exponentially move away from the SIM. This would lead to a large contribution to the objective function. Also, from this analysis it seems beneficial to increase the integration horizon as much as possible because the instability might manifest itself significantly only after a critical amount of time for points that are very close to the manifold. In **[55]** it is shown that the method identifies the SIM of linear models exactly for $T \to \infty$ in the reverse mode. Hence, in practice one would like to chose T as large as possible, however, the choice of T is often limited by the stability of the shooting approach that is used to solve the optimization problem (see next section). For singular perturbation systems it seems reasonable to choose T in the order of ε since the boundary layer is of that width.

3.3. Numerical Methods. Eventually, the optimization problem (50) has to be solved numerically. The PhD thesis of Jochen Siehr **[81]** is largely devoted to this subject, and we are only going to highlight a couple of interesting points. Problem (50) is a semi-infinite optimization problem similar to the optimal control problems that were treated in Chapter 3 as in both cases the solutions are functions and thus come from an infinite dimensional search space. Accordingly, the same methods used for solving optimal control problems, namely single and multiple shooting could be used to discretize problem (50) and reduce it to an NLP. In the software package `MoRe` developed by Jochen Siehr the NLP is either solved using `IPOPT` **[89]** if the global, integral objective (51) is used, or a generalized Gauss-Newton method **[8]** is employed in case of (52). The generalized Gauss-Newton approach is combined with an active set strategy and a filter method to handle constraints and choosing appropriate step sizes.

3.3.1. *Sensitivity Generation.* We aim to replace $y(t)$ with $h\big(x(t)\big)$ in the optimal control problem. Whenever in the numerical procedure to solve the reduced optimal control problem derivatives of the right hand side or constraint functions with respect to x are needed we have to provide $\mathrm{D}_x\, h(x)$. For example

$$\mathrm{D}_x\, f(x, h(x), u) = \mathrm{D}_1\, f(x, h(x), u) + \mathrm{D}_2\, f(x, h(x), u)\, \mathrm{D}_x\, h(x).$$

In the context of the finite (discretized) model reduction problem this means we are interested in the derivative of the optimal solution (x^*, y^*) with respect to r which are the values the reaction progress variables are fixed to. If this derivative exists at all and how it may be computed will be discussed immediately, for now we assume $x^* = x^*(r)$ and $y^* = y^*(r)$. Depending on the discretization method that was chosen, the optimization variables (x^*, y^*) do not correspond to the point $y = h(r)$ on the manifold but are given through an extra mapping

$$y = h(r) = \eta\big(x^*(r), y^*(r)\big),$$

hence

$$\mathrm{D}_r\, y = \mathrm{D}_r\, h(r) = \mathrm{D}_1\, \eta\big(x^*(r), y^*(r)\big)\, \mathrm{D}_r\, x^*(r) + \mathrm{D}_2\, \eta\big(x^*(r), y^*(r)\big)\, \mathrm{D}_r\, y^*(r).$$

The partial derivatives of η are known from the discretization method and the sensitivities $\mathrm{D}_r\, x^*$ and $\mathrm{D}_r\, y^*$ are available at a low extra cost from the solution of the NLP.

Essentially, the solution of the NLP is a root of a function $K(x, y, \lambda) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_\lambda}$ representing the KKT conditions (Definition 46) with Lagrange multipliers λ. This function is augmented by the parameter r and now $K := K(x, y, \lambda, r)$. We seek solutions of the problem $K(x, y, \lambda, r) = 0$ in x, y and λ. If LICQ (Definition 47) and second order sufficient conditions (Theorem 49) hold for a solution $x^*(r^*)$, $y^*(r^*)$, and $\lambda^*(r^*)$ then x^*, y^*, and λ^* are differentiable functions of r around $r = r^*$, [**22**]. We find the linear system

$$\begin{aligned}\mathrm{D}_r\, K(x^*(r), y^*(r), \lambda^*(r), r) = {} & \mathrm{D}_1\, K(x^*(r), y^*(r), \lambda^*(r), r)\, \mathrm{D}_r\, x^*(r) + \\ & \mathrm{D}_2\, K(x^*(r), y^*(r), \lambda^*(r), r)\, \mathrm{D}_r\, y^*(r) + \\ & \mathrm{D}_3\, K(x^*, y^*(r), \lambda^*(r), r)\, \mathrm{D}_r\, \lambda^*(r) \\ = {} & -\mathrm{D}_4\, K(x^*(r), y^*(r), \lambda^*(r), r),\end{aligned}$$

for the sensitivities $\mathrm{D}_r\, x^*$, $\mathrm{D}_r\, y^*$ and $\mathrm{D}_r\, \lambda^*$. The matrix

$$\mathcal{K} = [\mathrm{D}_1\, K(x^*, y^*, \lambda^*, r), \mathrm{D}_2\, K(x^*, y^*, \lambda^*, r), \mathrm{D}_4\, K(x^*, y^*, \lambda^*, r)]$$

is called the KKT matrix. Let

$$\mathcal{R} = \begin{pmatrix} \mathrm{D}_r\, x^*(r) \\ \mathrm{D}_r\, y^*(r) \\ \mathrm{D}_r\, \lambda^*(r) \end{pmatrix}$$

be the matrix of sensitivities. The linear equation from above can be shortened to

$$\mathcal{K}\mathcal{R} = -\mathrm{D}_r\, K.$$

Under the assumptions put forward above, the matrix $\mathcal{K}$ has full rank and the sensitivities can be obtained by solving the linear system.

3.3.2. *Warm Starts and Path Following.* When using model order reduction in the context of optimal control, problem (50) has to be solved for numerous different input values r. Since $h(x)$ is assumed to be at least C^2 we can expect that for two fixed values x_1, x_2 where $\|x_1 - x_2\|$ is small also $\|y_1^* - y_2^*\|$ is small for the corresponding $y_1^* = h(x_1)$ and $y_2^* = h(x_2)$. Thus, the NLPs for obtaining y_1 and y_2 are close, too, in the sense that according Lagrange multipliers λ_1 and λ_2 are close. This can be exploited by using warm starts of the NLP solver, which means Lagrange multipliers as well as other internal variables and states of the NLP solver from one run are used to initialize the algorithm for the next run on a close problem.This strategy can greatly reduce the number of NLP iterations needed.

A more sophisticated approach is again to consider the optimal solution x^*, y^* and Lagrange multipliers λ^* as differentiable functions of the parameter r. We write $c(r) = \big(x^*(r), y^*(r), \lambda^*(r)\big)^{\mathrm{T}}$. Two parameter values r_0, r_{f} are then assumed to be

connected through a differentiable path $c(r) \in \mathbb{R}^{n_x+n_y+n_\lambda}$. A possible strategy to get from r_0 to r_f in terms of the solution $c(r)$ is to approximately following this path with a predictor corrector scheme in which a prediction for the solution at some point $r + hd$, where $h \in (0, 1]$ and $d \in \mathbb{R}^{n_x}$, $\|d\| = 1$ is made and this prediction is used to start the optimization procedure at that point as correction of the prediction. The prediction can generally be based on interpolation of the path $c(r)$. To this end define a grid r_i, $i = 0, 1, \ldots, n_r$, $r_n = r_\mathrm{f}$ and step sizes $h_i = \|r_{i+1} - r_i\|$, $i = 1, 2, \ldots, n_r$. Additionally, let c_i be the approximation of $c(r)$ at r_i. A first order predictor in direction $d = r_2 - r_1$ is given by

$$c_{i+1} = c_i + h\,\mathrm{D}_r\, c_i d.$$

For $h = 1$ a full step is taken and r_f is reached with one step. Remember that after each step the predictor c_{i+1} is corrected by solving the optimization problem (50) with c_i as initial guess for the optimization variables. A step size strategy based on the aspired number of NLP iterations in each correction step can be developed and is implemented in the software package `MoRe` by Jochen Siehr, [**81**, Chapter 8].

3.4. Application to Control Systems. We consider control systems

$$\begin{aligned} \mathrm{D}_t\, x &= f(x, y, u), \\ \mathrm{D}_t\, y &= g(x, y, u), \end{aligned} \tag{55}$$

where $x(t) \in \mathbb{R}^{n_x}$ and $y(t) \in \mathbb{R}^{n_y}$ are the slow and fast states, respectively. The control $u(t) \in U \subset \mathbb{R}^{n_u}$ is a function $u : \mathbb{R} \to \mathbb{R}^{n_u}$. The following is assumed to hold:

- H1 The right hand sides f and g are smooth functions of their arguments on any domain of interest.
- H2 A unique solution of (55) exists for all initial values $x(0)$, $y(0)$ and all admissible $u(t)$ on any time interval $[0, T]$.
- H3 A SIM (depending on u) exists for all admissible $u : \mathbb{R} \to \mathbb{R}^{n_u}$.

For singular perturbed systems we saw that the manifold can be described by a function $y = h(x, u)$ that depends on x and u. For general systems the third assumption allows us at least to write $y = h^u(x)$ for the manifold equation for a fixed u, i.e. for each admissible u we obtain a different set of right hand sides and thus manifolds.

Remark 38. The third assumption H3 is strong, however natural: If it does not hold the model reduction approach we pursue here is bound to fail. If the SIM ceases to exist for some $u(t)$, then in general using the reduced model in the optimal control context will lead to unreliable and wrong results (even if the numerical model reduction method will deliver a result).

For all practical purposes we build up on assumption H3 and the properties of our multiple shooting approach to solve the reduced optimal control problem. On the j-th multiple shooting interval, $j = 1, 2, \ldots, n$, we approximate $u(t)$ with a parametrized version $u_j(t, \alpha_i)$, $t \in [t_j, t_{j+1}]$, $\alpha_j \in \mathbb{R}^{n_\alpha}$. Each parameter $\alpha_j \in \mathbb{R}^{n_\alpha}$ constitutes different right hand sides $f_j(x, y, \alpha_j) = f\big(x, y, u_j(t, \alpha_j)\big)$ and $g_j(x, y, \alpha_j) = g\big(x, y, u_j(t, \alpha_i)\big)$. As mentioned before, sensitivities of $h(x)$ with respect to the optimization variables, which this time also include the α_j are necessary for the numerical solution of the optimal control problem. To this end we extend the system with the constant state $\tilde{x}(t) : \mathbb{R} \to \mathbb{R}^{n_\alpha}$ where

$$\mathrm{D}_t\, \tilde{x}(t) = 0, \quad \tilde{x}(0) = \alpha_j.$$

This way, from the model reduction point of view the underlying ODE system is

$$\begin{aligned} \mathrm{D}_t\, \tilde{x} &= 0, \\ \mathrm{D}_t\, x &= f_j(x, y, \tilde{x}), \\ \mathrm{D}_t\, y &= g_j(x, y, \tilde{x}), \end{aligned}$$

which is a normal (read not a control) system and the model reduction thus can be performed without modification and $y = h(x, \tilde{x})$ as well as $\mathrm{D}_x\, h$ and $\mathrm{D}_{\tilde{x}}\, h$ can be approximated using the techniques outlined above.

REMARK 39. Note that the introduction of the constant state variable $\tilde{x}$ is not necessary if the local objective function (52) is used since only the value $u(t^*)$ is needed which could be fixed to the desired value, i.e. $u(t^*) = u^*$ would enter the optimization problem as another constraint. However, the proposed setting includes this case and allows for a general treatment regardless of the concrete objective function in use.

EXAMPLE 17. To illustrate the ideas of this section we consider again the enzyme control system (see Examples 1, 5, and 7). We will use piecewise constant controls and therefore $n_\alpha = 1$ and $\alpha_j \in \mathbb{R}$. The parametrized control is $u_j = \alpha_j$ for all $t \in [t_j, t_{j+1}]$. The control system

$$\begin{aligned} \mathrm{D}_t\, x &= -x + (x + K - \lambda)y + u(t), \quad x(0) = \xi, \\ \varepsilon\, \mathrm{D}_t\, y &= x - (x + K)y, \quad y(0) = \eta. \end{aligned} \tag{56}$$

becomes

$$\begin{aligned} \mathrm{D}_t\, \tilde{x} &= 0, \quad \tilde{x}(0) = \alpha_j, \\ \mathrm{D}_t\, x &= -x + (x + K - \lambda)y + \tilde{x}, \quad x(0) = \xi, \\ \varepsilon\, \mathrm{D}_t\, y &= x - (x + K)y, \quad y(0) = \eta. \end{aligned} \tag{57}$$

For the unaugmented original system (56) and (57) series expansions for the respective manifold functions can be analytically computed. For system (56) we use $h^u(x, u)$ and for (57) $h^c(\tilde{x}, x, \varepsilon)$. The series coefficients up to first order are

$$\begin{aligned} h_0^c(\tilde{x}, x) = h_0^u(x, u) &= \frac{x}{x+1}, \\ h_1^c(\tilde{x}, x) &= \frac{(2x+2)\tilde{x} - x}{2x^4 + 8x^3 + 12x^2 + 8x + 2}, \\ h_1^u(x, u) &= \frac{(2x+2)u - x}{2x^4 + 8x^3 + 12x^2 + 8x + 2}. \end{aligned}$$

REMARK 40. The first order coefficients are the same except for an exchange of symbols. The manifold only depends on x and u in a local manner. The past, i.e. how a point (x, u) was reached is not important. Only if h^c and h^u are regarded along trajectories and controls $\tilde{x}(t)$, $x(t)$ and $u(t)$ there will be a difference.

We aim at a three-way comparison of manifolds: Numerically computed using the integral objective (51) ($h^i(\tilde{x}, x)$) versus numerically computed using the local objective (52) ($h^l(\tilde{x}, x)$) versus analytically computed to first order from (57) ($h^c(\tilde{x}, x)$). The manifold is approximated on the domain $(\tilde{x}, x) \in [0, 6] \times [0, 6]$ with 20 points in each direction. For illustration purposes a random trajectory with initial value $x(0) = 0$, $y(0) = 1$ and $\tilde{x}(t) = u(t) = 4.5$ is included.

For $\varepsilon = 1$, i.e. no time scale separation Figures 5.2–5.5 show h^i, h^l, h^c and $\left\|h^l - h^c\right\|$, respectively. Both numerical approximations show artifacts of instability which seem to be more grave for h^l. This shows that the lacking time scale separation poses problems for the model reduction algorithm, especially for small values of x and $\tilde{x}$. The analytical approximation is smooth. Note however, that

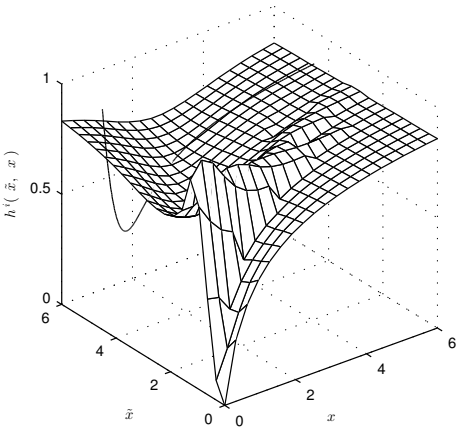

FIGURE 5.2. Plot of the numerical approximation $h^i(\tilde{x}, x)$ of the SIM using the integral objective (51) with $T = t^* = 5$.

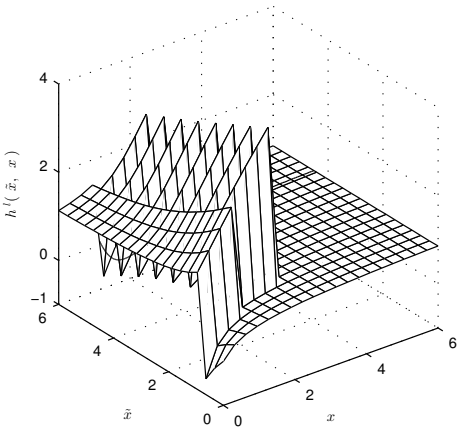

FIGURE 5.3. Plot of the numerical approximation $h^l(\tilde{x}, x)$ of the SIM using the local objective (52).

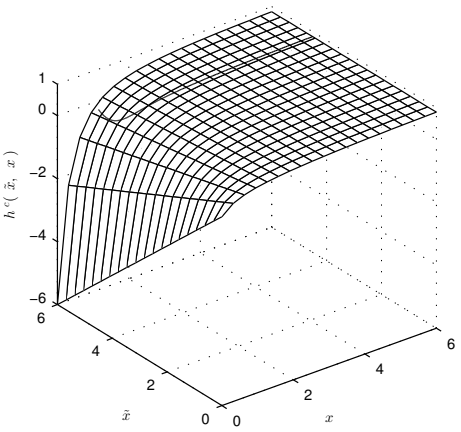

FIGURE 5.4. Plot of the analytical approximation $h^c(\tilde{x}, x)$ of the SIM.

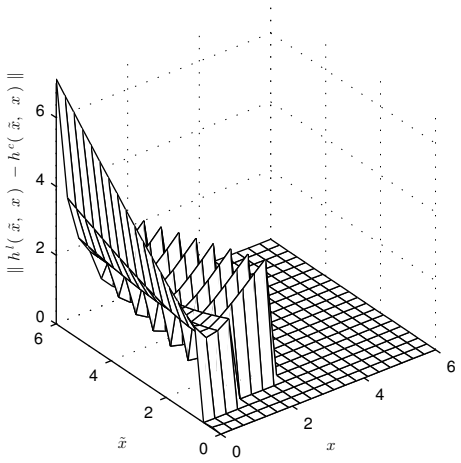

FIGURE 5.5. Plot of the error $\left\|h^i(\tilde{x}, x) - h^c(\tilde{x}, x)\right\|$.

the SIM is negative for small x and $\tilde{x}$. In more complex situations this might pose a problem because negative concentrations contradict model assumptions and can lead to errors and instabilities. This case is artificial in as much that there is no time scale separation to begin with and also the analytical approximation does not represent a SIM of the system. The comparison serves only to illustrate that the numerical model reduction algorithm works (albeit with problems) and produces results even in the case of no time scale separation.

For $\varepsilon = 0.1$ the situation is different. The numerical approximations (see Figures 5.6 and 5.7) are smooth, but different for small x. The analytical version (Figure 5.8) resembles the approximation obtained with the local objective and thus the error between the two (Figure 5.9) is relatively small (maximum about 0.6). Both, the analytical and local objective based manifold have negative values for x and $\tilde{x}$ small. This is different for the manifold obtained via the integral based

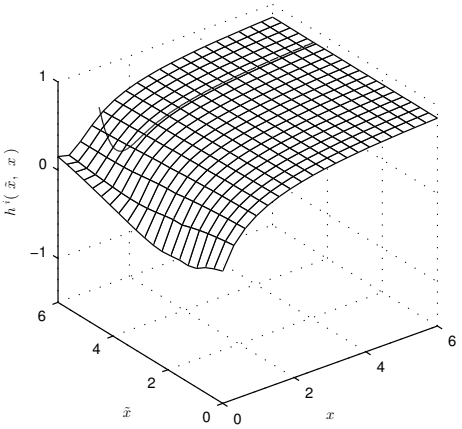

FIGURE 5.6. Plot of the numerical approximation $h^i(\tilde{x}, x)$ of the SIM using the integral objective (51) with $T = t^* = 0.5$.

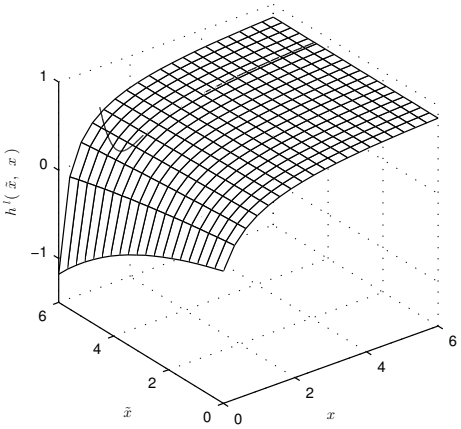

FIGURE 5.7. Plot of the numerical approximation $h^l(\tilde{x}, x)$ of the SIM using the local objective (52).

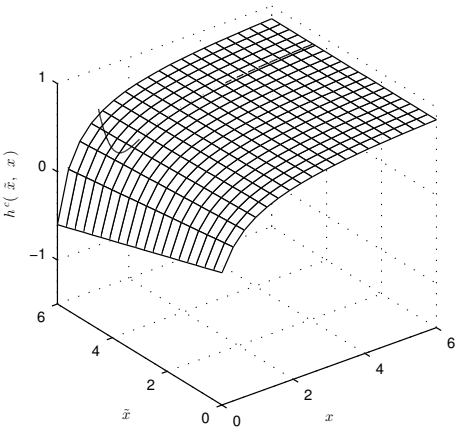

FIGURE 5.8. Plot of the analytical approximation $h^c(\tilde{x}, x)$ of the SIM.

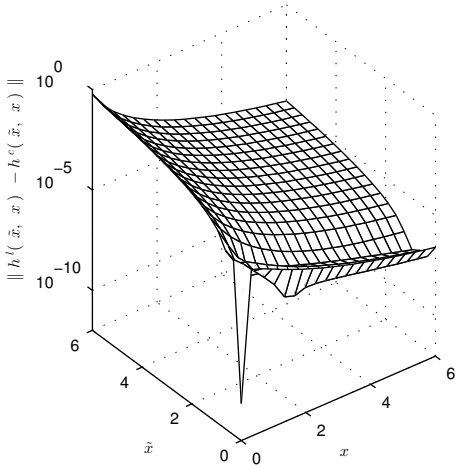

FIGURE 5.9. Plot of the error $\left\|h^i(\tilde{x}, x) - h^c(\tilde{x}, x)\right\|$.

objective. One may speculate that curvature for trajectories starting above zero is smaller since they reach the plateau of the manifold pretty fast compared to trajectories traveling along the strong bend of the manifold for small x and $\tilde{x}$.

3.4.1. *Evaluation of the Manifold.* Eventually, we have to evaluate the manifold map $y = h(x, u)$ for arbitrary points $(x, u) \subset \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$. We do not regard ε as argument of h here, since it is either a fixed parameter for the numerical solution of the optimal control problem in case of singularly perturbed problems or we regard general problems without explicit dependence on a small parameter ε. We will discuss two alternatives: Solving the model reduction problem online, i.e. whenever an evaluation of $h(x, u)$ is needed while solving the optimal control problem, and interpolation of offline precomputed data obtained by evaluating $h(x, u)$ on a discrete set of points $C \subset \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$. Both methods have intrinsic advantages and disadvantages. The online method can be easily applied since no preparation steps

are needed. However, calculating $h(x,u)$ is costly and involves the solution of an NLP which could slow down the overall computation. On the contrary, the interpolation approach provides a direct, fast, and easy to evaluate object, that promises a larger speed up. However, it suffers from the need to precompute the manifold data and building the interpolation object, both tasks take a considerable amount of time. This makes this approach only effective if the optimal control problem has to be solved very often (e.g. in NMPC) so that the time spend in the preliminary stages is outweighed by the overall performance gain. Furthermore, especially for higher dimensional problems, the storage needed for the interpolation data might be to large to be handled properly for given hardware resources.

If an evaluation of $h(x,u)$ is needed for a certain (ξ, u_0) we solve the model reduction problem (50) with the slow states and control fixed to (ξ, u_0). For performance reasons only the local formulation (52) is feasible because integration of the full model is dispensed with and the generalized Gauss-Newton method can be used for efficient solution of the minimization problem. Although it seems that this approach contradicts the purpose of model reduction, since the full right hand side still has to be evaluated, there is a computational advantage due to the decreased stiffness of the reduced model. In addition the points at which $h(x,u)$ has to be evaluated will typically be close to each other which makes it possible to use warm starts for the Gauss-Newton procedure or even the mentioned path following algorithms. Thus, in general, only very few iterations will be needed to solve the model reduction problem. This is in principle similar to solving an explicitly given differential-algebraic equation where usually inside the integration routine in each time step only a few iterations of a nonlinear equation solver are needed for the algebraic part of the dynamic problem.

As already described above, a set C of discrete points in the domain of interest $C \subset \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$ is established and the model reduction problem is solved for each $z = (x,u) \in C$. Additionally, sensitivities of $h(x,u)$ with respect to x and u are computed. Finally, we obtain a data set consisting of function values and directional derivatives of h. This data set is used to build an interpolation function $\phi(x,u) \approx h(x,u)$ using Hermite RBF interpolation as introduced in Chapter 4. For fast evaluation we rely on the partition of unity approach. The interpolation object ϕ is used to replace h in the model reduction approach.

Naturally, the question about the error between the full model and the reduced model comes up. We compare

$$\begin{aligned} \mathrm{D}_t\, x_\mathrm{f} &= f(x_\mathrm{f}, y_\mathrm{f}), \quad x_\mathrm{f}(0) = \xi, \\ \mathrm{D}_t\, y_\mathrm{f} &= g(x_\mathrm{f}, y_\mathrm{f}), \quad y_\mathrm{f}(0) = \eta, \end{aligned}$$

with

$$\begin{aligned} \mathrm{D}_t\, x_\mathrm{r} &= f\big(x_\mathrm{r}, h(x_\mathrm{r})\big), \quad x_\mathrm{r}(0) = \xi, \\ y_\mathrm{r} &= h(x_\mathrm{r}), \end{aligned} \tag{58}$$

and define the local errors

$$E_x(t) = \|x_\mathrm{f}(t) - x_\mathrm{r}(t)\|_2$$

and

$$E_y(t) = \big\|y_\mathrm{r}(t) - h\big(x_\mathrm{r}(t)\big)\big\|_2 .$$

For singularly perturbed systems we dealt with this question extensively in Chapter 2, see especially Remark 8 and Example 2. The solution $x_\mathrm{r}(t)$ of the reduced system along the manifold corresponds to the outer solution of the full system if the initial values match correctly. That means the outer solution of the full system does not start at ξ but at some value ξ^* that is only defined implicitly. Remember $x(t,\varepsilon) = x^*(t,\varepsilon) + X(t/\varepsilon, \varepsilon)$, where only $x(0,\varepsilon) = \xi$ is explicitly known. Only in the

case that $x^*(0,\varepsilon) = \xi$ and therefore $X(0,\varepsilon) = 0$ the outer solution $y^*(t)$ matches $y(t) = h\big(x^*(t)\big)$ exactly. Still, the error is given mainly through the magnitude of the boundary layer correction, which converges to 0 exponentially on the fast time scale (see Theorem 5). The crucial estimate is

$$\|X(\tau,\varepsilon)\|_2 + \|Y(\tau,\varepsilon)\|_2 \leq K_1 \|\eta(\varepsilon) - y^*(0,\varepsilon)\|_2 \, \mathrm{e}^{-\delta_1 \tau}, \ \tau \in [0, T/\varepsilon], \ \varepsilon \in (0, \varepsilon_0],$$

where $\tau = \frac{t}{\varepsilon}$ and K_1 and δ are constants. We have

$$\|E_y(t)\|_2 \approx \|Y(t/\varepsilon,\varepsilon)\|_2 .$$

Using the reduced system it must be clear that we disregard the boundary layer correction completely.

For general systems error estimates are not available however the concepts transfer. Fast modes relax onto the manifold exponentially fast and thus conceptually we have again that the error consists of ignoring the boundary layer correction.

For the offline approach we also have to regard the interpolation process as an error source. Two things can go wrong here: First, the SIM might not be in the native space of the basis function of the RBF interpolation and second, the interpolation process itself will be erroneous.

We are going to use the Gaussian basis function for our interpolation of h. The native space $\mathcal{N}_\phi$ of the Gaussian is rather small and includes only functions that are characterized through a Fourier transform that decays at least exponentially. Even if we assume that the SIM is smooth this will in general not suffice to conclude that it is contained in $\mathcal{N}_\phi$. However, we also saw that Gaussian RBF interpolation will converge for functions not from the native space, (see for example Theorem 74). However, for that case no error estimates are available. Another approach would be to multiply h with a bump function such that the result would be a member of $\mathcal{N}_\phi$. Numerical experiments show that there is no significant benefit in using a bump function. Moreover it introduces the problem of handling the boundary, which can get especially problematic if there is a problem intrinsic border, for example the model is only valid for positive states. In that case, the interpolation for points close to zero is seriously perturbed by the bump function.

To analyze the error caused by the interpolation let $\phi_h(x)$ be the RBF interpolation of $h(x)$ on a set of points $X = \{x_j\}_{j=1}^N$ on a domain $\Omega \subset \mathbb{R}^{n_x}$ with $X \subset \Omega$. The interpolation error was defined in terms of the fill distance $h_{X,\Omega}$ (Definition 71). We saw that for the RBF interpolant with Gaussian basis function the error decayed exponentially with the fill distance. The influence of the interpolation error on the solution of the differential equation system (58) can be estimated with Gronwall's inequality.

THEOREM 86. *Let $\phi_h(x)$ be an interpolation of $h(x)$ as described. For the solutions of the initial value problems*

$$\mathrm{D}_t\, x_1 = f\big(x_1, \phi(x_1)\big), \quad x_1(0) = \xi \ \textit{and} \ \mathrm{D}_t\, x_2 = f\big(x_2, h(x_2)\big), \quad x_2(0) = \xi$$

on the interval $[0,T]$ it holds that

$$\|x_1(t) - x_2(t)\| \leq L_{f,\Omega} M T \mathrm{e}^{L_{f,\Omega} t}$$

where $L_{f,\Omega}$ is the global Lipschitz constants of f with respect to the first and second argument of f on Ω and $M = \max_{x\in\Omega} \|\phi(x) - h(x)\|$ the maximum error of the interpolation on Ω.

PROOF. With the Lipschitz continuity of f with respect to y it follows that

$$\big\|f\big(x,\phi(x)\big) - f\big(x,h(x)\big)\big\| < L_{f,\Omega} \|\phi(x) - h(x)\| < L_{f,\Omega} M$$

and the difference in f can be bounded. Now define

$$f_1(x) := f\big(x, \phi(x)\big) \text{ and } f_2(x) := f\big(x, h(x)\big),$$

so that

$$\|f_1(x) - f_2(x)\| < L_{f,\Omega} M \tag{59}$$

for all $x \in \Omega$.

We proceed with the usual proof utilizing Gronwalls Lemma [**67**, Satz 4.9]:

$$\|x_1(t) - x_2(t)\| = \left\| \xi + \int_0^t f_1\big(x_1(\tau)\big)\mathrm{d}\tau - \xi - \int_0^t f_2\big(x_2(\tau)\big)\mathrm{d}\tau \right\|$$

(we omit the argument τ from here on)

$$\begin{aligned}
&\leq \int_0^t \|f_1(x_1) - f_1(x_2) + f_1(x_2) - f_2(x_2)\| \,\mathrm{d}\tau \\
&\leq \int_0^t \|f_1(x_1) - f_1(x_2)\| \,\mathrm{d}\tau + \int_0^t \|f_1(x_2) - f_2(x_2)\| \,\mathrm{d}\tau \\
&\overset{(59)}{\leq} L_{f,\Omega} \int_0^t \|x_1 - x_2\| \,\mathrm{d}\tau + T L_{f,\Omega} M \\
&\overset{\text{Gronwall}}{\leq} T L_{f,\Omega} M \mathrm{e}^{L_{f,\Omega} t}.
\end{aligned}$$

□

As one would expect the error due to the interpolation approaches 0 (at least theoretically) if the interpolation error itself is reduced. We saw that for numerical computations the interpolation suffers from instability issues if the fill distance gets too small. Therefore the set of interpolation nodes X has to be carefully chosen as to minimize the interpolation and therefore the simulation error.

Example 18. To give an example for the computational advantages of model reduction we are picking up Examples 3 and 4. For reference: It is a linear singularly perturbed system with five states and one control. Of the five states two are slow, i.e. $x(t) = \big(x_1(t), x_2(t)\big)^{\mathrm{T}}$ and three are fast, i.e. $y = \big(y_1(t), y_2(t), y_3(t)\big)^{\mathrm{T}}$. For our numerical experiments we set $u(t) = 0$. We compare the integration of the full model with the integration of the reduced model in online and offline mode. We use the subscripts f, d, and i to refer to the full, the offline reduced and the online reduced system respectively. The error will be recorded pointwise, i.e. we look at $\|x_{\mathrm{f}}(t) - x_{\mathrm{d}}(t)\|_2$, $\|x_{\mathrm{f}}(t) - x_{\mathrm{i}}(t)\|_2$, and $\|x_{\mathrm{d}}(t) - x_{\mathrm{i}}(t)\|_2$. For numerical integration we use the BDF integrator introduced in Section 3.1.1. Numerical efficiency is accessed through integrator statistics (number of steps, number of rejected steps, etc.) and timings. Here we only consider pure integration timings, a more thorough examination of the subject is given in the next chapter.

To also include a comparison between the different objectives in the optimization problem for model reduction (integral based (51) and local (52)) we used the global objective to compute the interpolation data. In all cases the backward mode was used, i.e. $t^* = T$.

For the reduced model initial values for the fast variable can not be set because they are determined through $y = h(x)$. However, they play an important role for the integration of the full system, therefore we will use a set of two different values (for the fast variables)

$$x(0) = (-10, 10)^{\mathrm{T}}$$

and

$$\eta_1 = y(0) = (10, 10, 10)^{\mathrm{T}}, \quad \eta_2 = y(0) = (0, 0, 0).$$

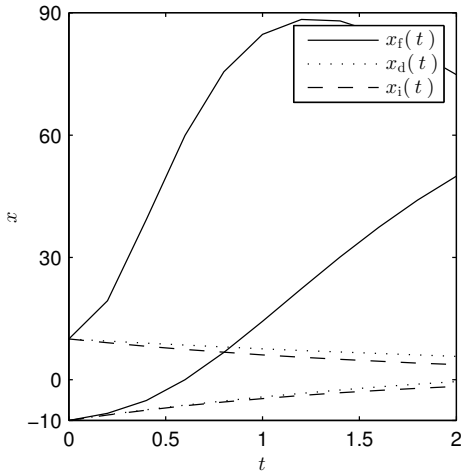

FIGURE 5.10. Trajectories $x_\mathrm{f}(t)$, $x_\mathrm{d}(t)$, and $x_\mathrm{i}(t)$ for $y(0) = \eta_1 = (10, 10, 10)^\mathrm{T}$ and $\varepsilon = 2 \times 10^{-1}$.

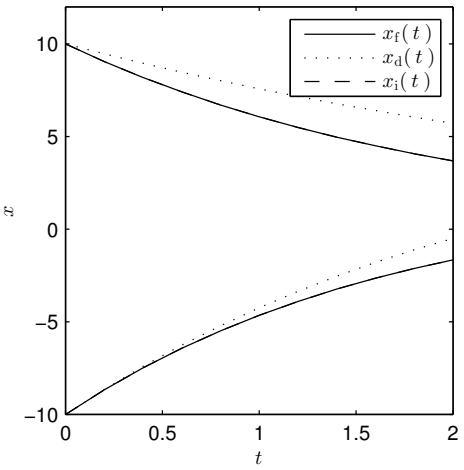

FIGURE 5.11. Trajectories $x_\mathrm{f}(t)$, $x_\mathrm{d}(t)$, and $x_\mathrm{i}(t)$ for $y(0) = \eta_2 = (0, 0, 0)^\mathrm{T}$ and $\varepsilon = 2 \times 10^{-1}$. The lines for x_f and x_i overlap.

TABLE 5.1. Integration statistics for $\varepsilon = 2 \times 10^{-1}$.

	full, η_1	full, η_2	offline	online
accepted steps	49	18	16	18
rejected steps	5	5	2	2
time (s)	0.001	3×10^{-4}	2×10^{-4}	0.007

The second choice is the first order approximation of $h(x(0), 0)$.

We start with $\varepsilon = 2 \times 10^{-1}$, for the model reduction $T = 2$. The time scale separation seems to be too small as can be seen in Figure 5.10. For the initial value η_1 the model reduction error is fairly large. Only if we start at η_2 so that the fast variables are close to the SIM the error $\|x_\mathrm{f}(t) - x_\mathrm{i}(t)\|_2$ is negligible, see Figure 5.11. Surprisingly, there is a significant difference between $x_\mathrm{i}(t)$ and $x_\mathrm{d}(t)$. This is not due to the interpolation error but the difference in the model reduction results. It seems that the local optimization problem (52) is not able to capture the SIM accurately enough or with a high enough order. This was observed in several examples (not shown here), however, a clear explanation is lacking. Theoretically, the order of the approximation is determined by the order of the derivative term in the objective, which is the same for both formulations. There is although a bit of ambiguity about the integration horizon. As we saw a large T would most likely increase the quality of the approximation although its influence can not be pinned down exactly.

Only of minor interest are the integration statistics given in Table 5.1 because the resulting trajectories are erroneous for most initial values anyway. Still, they show the principle we wish to exploit: For the reduced model the number of integration steps is reduced compared to the full model if the initial values for the fast modes are not close to the SIM. This gives the interpolation approach a significant time advantage for η_1.

For $\varepsilon = 2 \times 10^{-3}$ there is a larger time scale separation. The results are plotted in Figures 5.12 and 5.13. Clearly, the error between the full and reduced model is

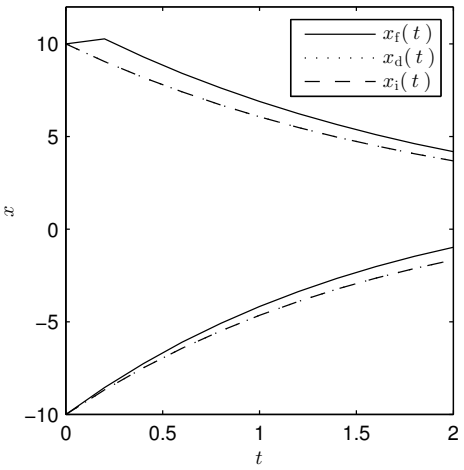

FIGURE 5.12. Trajectories $x_\mathrm{f}(t)$, $x_\mathrm{d}(t)$, and $x_\mathrm{i}(t)$ for $y(0) = \eta_1 = (10, 10, 10)^\mathrm{T}$) and $\varepsilon = 2 \times 10^{-3}$.

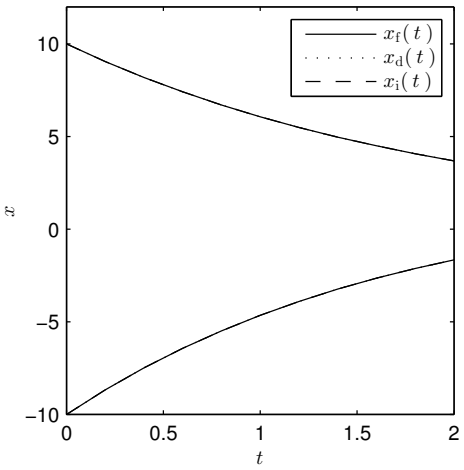

FIGURE 5.13. Trajectories $x_\mathrm{f}(t)$, $x_\mathrm{d}(t)$, and $x_\mathrm{i}(t)$ for $y(0) = \eta_2 = (0, 0, 0)^\mathrm{T}$) and $\varepsilon = 2 \times 10^{-3}$. All lines overlap.

TABLE 5.2. Integration statistics for $\varepsilon = 2 \times 10^{-3}$.

	full, η_1	full, η_2	offline	online
accepted steps	108	18	18	18
rejected steps	10	5	2	2
time (s)	0.002	5×10^{-4}	4×10^{-4}	0.01

smaller than for $\varepsilon = 2 \times 10^{-1}$, although there still is a significant boundary layer if $\eta_1(0)$ is used as initial value. Using the interpolation does not result in a noticeable difference compared to the online integration this time.

The dependence on the initial values is also visible in the integration statistics, see Table 5.2. For η_1 the integration of the full system needs around 6 times more steps than the integration of the reduced system with either method, offline or online. Using the first order approximation of the manifold for $y(0)$, i.e. η_2 shows that also the full system looses its stiffness and the same small number of steps are needed for the integration. The time spend for the integration is not only a function of the integrator steps but also of the cost for function evaluation. It seems that using the online approach is rather expensive and not competitive for pure integration. The interpolation approach delivers a performance that is better or on par with the full system even for η_2.

4. Summary

In this chapter we introduce model order reduction as a way to speed up the simulation and numerical computations to obtain an optimal control solution which is for example needed for model predictive control. We see that solving the initial value problem in the multiple shooting approach is one large contributor to the computation time. Reducing the order of the model helps to decrease this time because of the reduced number of variables that has to be considered but also and mainly because of the reduced stiffness of the model which results in fewer steps of the numerical integration procedure.

The model reduction approach focused on here is based on the idea of a *slow invariant manifold* (SIM) that is assumed to exist in the phase space of a system with significant time scale separation. The state space is partitioned into slow and fast variables x and y respectively and the SIM is then parametrized by the slow variables, i.e. there is a map $y = h(x)$. The SIM can be theoretically identified in the case of singularly perturbed systems, for general nonlinear systems a clear definition is not available but the concept persists.

For general systems the SIM can only be approximated numerically. Our approach is based on formulating the model reduction problem as an optimization problem. To this end the value of the slow variables, also called reaction progress variables are fixed and the corresponding values of the fast variables are sought as minimizers of some objective functional. Two objectives were introduced, an integral based formulation that aims at minimizing the curvature of trajectories and a local formulation that minimizes the second derivative of the solution $x(t)$ and $y(t)$ with respect to time. If singularly perturbed systems are considered, the objectives can be linked to the zero derivative principle.

The numerical solution of the optimization problem for model reduction involves reducing the (in case of the integral objective) infinite problem to an NLP. Also, for the purpose of optimal control sensitivities of h with respect to x are needed which can be obtained cheaply if the NLP is regarded as parametrized by the value of the reaction progress variable. Moreover, this sensitivities can be put to use in path following strategies that aim to speed up the solution of neighboring model reduction problems, which are bound to turn up in the context of optimal control.

Finally, the model reduction has to be connected with the integration of the reduced model, i.e. $y = h(x, u)$ has to be evaluated fast and reliably whenever it occurs during the integration routine. Offline and online evaluation is considered. The offline approach consists of precomputing the manifold (and sensitivities) on a discrete set of points and using RBF interpolation to obtain a smooth and easy to evaluate object. Online evaluation solves the model reduction problem for each input value. It takes advantage of the local formulation, the general Gauss-Newton method with warm starts or path following.

CHAPTER 6

Numerical Results

1. Introduction and General Remarks

In this chapter all the tools (model reduction, interpolation and numerical optimal control) that were discussed in earlier chapters are united and used to showcase benefits and disadvantages of their use with the help of three examples. To facilitate the discussion we will use the subscript f when referring to the full model and r for the reduced model. We are looking at the problems

$$\begin{aligned} &\min_{x_\mathrm{f},u_\mathrm{f}} J_\mathrm{f}(u_\mathrm{f}) = \Theta\big(x_\mathrm{f}(T)\big) + \int_0^T \theta\big(x_\mathrm{f}(t),u_\mathrm{f}(t)\big)\mathrm{d}t \\ \text{s.t. } &\mathrm{D}_t\, x_\mathrm{f} = f(x_\mathrm{f},y,u_\mathrm{f}), \\ &\mathrm{D}_t\, y = g(x_\mathrm{f},y,u_\mathrm{f}), \end{aligned}$$

and

$$\begin{aligned} &\min_{x_\mathrm{r},u_\mathrm{r}} J_\mathrm{r}(u_\mathrm{r}) = \Theta\big(x_\mathrm{r}(T)\big) + \int_0^T \theta\big(x_\mathrm{r}(t),u_\mathrm{r}(t)\big)\mathrm{d}t \\ \text{s.t. } &\mathrm{D}_t\, x_\mathrm{r} = f(t,x_\mathrm{r},h(x_\mathrm{r},u_\mathrm{r}),u_\mathrm{r}), \end{aligned}$$

When dealing with reduced models in the context of optimal control there are two key aspects:

- What is the performance of the optimal control $u_\mathrm{r}^*(t)$ obtained with the reduced model when used in the full model?
- What, if any, is the time benefit for using the reduced model to compute the optimal control?

The first point addresses the usefulness of $u_\mathrm{r}(t)$ to control the full dimensional system which is the ultimate goal of computing the optimal control strategy. The second question aims at the numerical efficiency of the approach. Both points have to be fulfilled in order to make model reduction in optimal control scenarios a valid strategy. The reduced solution should be close enough to the solution for the full model and it should be computable significantly faster.

The full control $u_\mathrm{f}(t)$ consists of two boundary layer corrections $U_L(t)$, $U_\mathrm{r}(t)$ and an outer solution part $u^*(t)$:

$$u_\mathrm{f}(t) = u^*(t) + U_L(t) + U_\mathrm{r}(t).$$

For singularly perturbed systems each part of the decomposition of $u_\mathrm{f}(t)$ can be expanded into an ε series and we have

$$J_\mathrm{f}\big(u_\mathrm{f}(t)\big) = J_\mathrm{f}^*(t) + J_L(t) + J_\mathrm{r}(t).$$

The reduced optimal control lacks the boundary layer corrections, thus

$$u_\mathrm{r}(t) \approx u^*(t)$$

and

$$J_\mathrm{r}\big(u_\mathrm{r}(t)\big) \approx J_\mathrm{f}^*(t).$$

Hence comparing J_r and J_f directly is usually not meaningful because the contribution of the boundary layer corrections can not be quantified without analytical

insight. Also, as stated above, since the full system is what we aim to control the performance of the reduced system is only of minor interest to us. One obvious measure for the quality of u_f is the objective value

$$J(u_\mathrm{r}) = \Theta\big(x(T)\big) + \int_0^T \theta\big(x(t), u_\mathrm{r}(t)\big)\mathrm{d}t$$

where x is the solution of

$$\begin{aligned}\mathrm{D}_t\, x &= f(x, y, u_\mathrm{r}),\\ \mathrm{D}_t\, y &= g(x, y, u_\mathrm{r}).\end{aligned}$$

Note that x and y will in general neither be equal to x_f and y_f, nor x_r and $h(x_\mathrm{r}, u_\mathrm{r})$. We expect $J_\mathrm{f}(u_\mathrm{f}) \leq J(u_\mathrm{r})$ because u_r will not contain the boundary layer correction and is thus suboptimal for the full system.

For the computational comparison one has to clarify first what should be compared. When starting with the full problem formulation there are several steps until the final u_f and u_r can be computed:

(1) Select reaction progress variables x and fast variables y.
(2) Implement the system for the model reduction algorithm.
(3) Tune the model reduction algorithm:
Online: Set tolerances for the general Gauss-Newton method.
Offline: Chose the objective (integral based/local); if integral based, chose T and t^*; Define the domain $\Omega \subset \mathbb{R}^{n_x}$ and grid $X \subset \Omega$ for the interpolation; Tune the interpolator (number of points per partition of unity patch, overlap factor); Build the interpolation object.
(4) Solve the full and reduced optimal control problem for u_f and u_r, respectively.

Each step takes a considerable amount of time. Points 1–3 are preparation steps specific to the reduced system only. The time spent there is hard to measure since especially the tuning of the model reduction algorithm and interpolation is not yet automated and has to be done manually. Building the interpolation object includes optimizing the shape parameter and solving the linear system to obtain the interpolation coefficients. The time spent there can be easily measured. The last step is where the model reduction should pay of in terms of time consumption and its duration can be recorded accurately. Let us denote the time spent in step 4 for the full and the reduced system respectively with σ_f and σ_r. The difference $\sigma = \sigma_\mathrm{f} - \sigma_\mathrm{r}$ is positive if the reduced optimal control problem can be solved faster than the full problem. Usually, even if $\sigma > 0$ it will not cover the preparation time invested in steps 1–3, thus the model reduction will only pay off if the optimal control problem has to be solved several times (for example in NMPC, see the introduction of Chapter 5). Typically, there will be certain number n_s of repetitions necessary such that $n_s\sigma$ is larger than the time used for preparation.

Remark 41. Storage requirements (either hard drive or RAM) do not play a role in our considerations of computational performance. Although there seems to be an advantage because of the reduced number of variables in the reduced system this view does not take into account additional storage needs for model reduction (online) or interpolation data (offline). Especially the interpolation approach needs to fit all the interpolation data (function values and derivatives) into the RAM which can be prohibitive for higher dimensional problems because the amount of data grows like $\mathcal{O}(N^{n_y})$ where N is the number of interpolation nodes. All in all, storage is very problem dependent and we merely assume that enough storage is available.

REMARK 42. The problem formulations we are considering in this chapter do not contain boundary value constraints, e.g. final time point constraints like $x_{\mathrm{f}}(T) = 42$. We already mentioned that this is a difficult topic. Initial and final values for the fast (and therefore reduced) variables y can not be attained as long as they are not on the SIM. Theoretically, the slow variables x_{r} in the reduced system can be subject to boundary or path constraints, however, these would only be satisfied in the reduced setting. One can not guarantee that using u_{r} for the full system will lead to trajectories $x(t)$ that fulfill the additional constraints.

2. Enzyme Example

This example is based on Michaelis-Menten enzyme kinetics in singularly perturbed form we already used in Examples 1, 5, 7, and 17. The (full) problem was introduced in Example 7:

$$\begin{aligned} &\min \int_0^5 -50y + u^2 \mathrm{d}t \\ \text{s. t. } &\mathrm{D}_t\, x = -x + (x + 0.5)y + u, \\ &\varepsilon\, \mathrm{D}_t\, y = x - (x + 1.0)y, \\ &x(0) = 1, \quad y(0) = \eta. \end{aligned} \tag{60}$$

The control and the objective are artificial and not related to a realistic model scenario. The reduced problem is obtained by replacing y with $h(x, u)$, eliminating the ODE for y and the initial condition η. The reduced problem is thus

$$\begin{aligned} &\min \int_0^5 -50h(x, u) + u^2 \mathrm{d}t \\ \text{subject to: } &\mathrm{D}_t\, x = -x + (x + 0.5)h(x, u) + u, \\ &x(0) = 1. \end{aligned} \tag{61}$$

We set $\varepsilon = 10^{-2}$ and use the multiple shooting approach described in Section 3.1. To obtain initial values for x and y at the multiple shooting nodes we integrate the uncontrolled system ($u(t) = 0$) numerically starting at $x(0) = 0$ and $y(0) = \eta$. The overall time interval is divided into 40 equidistant multiple shooting intervals. The termination tolerance for `IPOPT` was set to 10^{-4} and the integration tolerance of the BDF-integrator to 10^{-6}. The discretized full problem has 204 variables whereas the reduced problem has 163. Lastly, bounds are introduced for the state variables and the control. We use the lower bounds $x_{\mathrm{l}} = y_{\mathrm{l}} = u_{\mathrm{l}} = 0$ and the upper bounds $x_{\mathrm{u}} = y_{\mathrm{u}} = u_{\mathrm{u}} = 9$ on all multiple shooting nodes.

In Example 7 we computed a control solution to the full problem via an indirect approach. We denote this solution $u^*(t)$ in this section.

We go ahead and analyze the runtime of the various numerical solution attempts. The performance of the full system (60) depends on the initial value η. For $\eta = 0$ we have an average runtime of $\sigma_{\mathrm{f}} = 4.4\,\mathrm{s}$ and 44 NLP iterations. For $\eta = 0.5$, which is the first order approximation $h_0(1, 0)$ we find $\sigma_{\mathrm{f}} = 4.8\,\mathrm{s}$ and 50 NLP iterations and lastly for $\eta = 1$ we get $\sigma_{\mathrm{f}} = 39\,\mathrm{s}$ and 3.9 iterations. Surprisingly, the first order approximation $y(0) = 0.5$ fares the worst in terms of number of NLP iterations and time needed.

Next we used the online computation of h, i.e. the local objective (52) in combination with the general Gauss-Newton procedure. The algorithm clocks in at $\sigma_{\mathrm{d}} = 3.8\,\mathrm{s}$ and 44 iterations, which means no significant performance gain compared to the full problem. Note however that since the initial value $y(0)$ is not part of the problem anymore, the runtime does not depend on it for the reduced model. It is interesting to look at the number of iterations of the general Gauss-Newton method

TABLE 6.1. Summary of various statistics concerning the solution of problem (60) and (61) with $y(0) = 0$. The steps statistics refer to the integration and are the sums over all NLP iterations and multiple shooting intervals.

problem	time	NLP iter	time per iter	steps	rej. steps
(60)	4.4 s	44	0.100 s	73 053	12 496
(61) online	3.8 s	44	0.087 s	9674	2247
(61) offline (median)	1.7 s	52	0.033 s	-	-
(61) offline (best)	0.9 s	38	0.025 s	8245	1878

in the subproblem of approximating the manifold. The maximum is 4 iterations but 59 % of the calls terminate after 2 and 37 % only after 1 iterations (3 iterations: 3 %, 4 iterations: 0.7 %) The call to the model reduction routine is by now done through an external library which produces a lot of overhead, for example in terms of right hand side function evaluations. For example, for every x used in the integrator the reduced right hand side $f\big(x, h(x, u)\big)$ is evaluated. The same evaluation is finally made in the model reduction routine without sharing the result.

For the interpolation approach we use the global objective (51) in reverse mode with $T = t^* = 0.075$ together with the shooting approach for discretizing the NLP. The final time is chosen in such a way that it is of the same order as ε and as large as possible (approximated with numerical experiments). Next, one needs to select a reasonable set of nodes. Although the RBF method is grid independent for convenience we use a Cartesian grid on $[-0.5, -0.5] \times [10, 10]$. The performance of the interpolator depends of course on the number of points but also on the number of points per patch and overlap in the partition of unity approach. To assess the influence we scatter searched the region $\{20, 30, 35, 40\} \times \{0.025, 0.05, 0.1, 0.15\} \times \{5, 10, 15\}$ for points in each direction, overlap and points per patch respectively. Median runtime was 1.7 s and median number of NLP iterations 52. Moreover, the fastest combination took 0.9 s and only 38 iterations compared to the slowest which needed 3.8 s and 105 iterations. It is apparent that using the interpolation approach in this case is beneficial from an performance point of view. In the best case it is more than 5 times faster than the full problem.

The main reason for the speed up is not so much the reduction in the number of optimization variables but mainly the reduced stiffness along the manifold h. In the integration routine larger step sizes are possible which greatly reduces the computational effort. This especially pays off in the multiple shooting approach since the initial values at the multiple shooting nodes are subject to optimization and they might be set away from the SIM for the full system in each iteration of the NLP solver leading to transient behavior of the fast trajectories on each interval and therefore forces the integrator to use small steps. An overview of example integrator statistics is given in Table 6.1 as well as the result of the performance tests. Despite the huge difference in accepted steps between the full model and either one of the reduced models there also is a difference between the online and offline method. It seems that the SIM returned via the integral based objective (51) reduces stiffness even more. Another thing that should be noted is the time spend in each iteration. The difference between the full and reduced models could be explained by the larger number of steps the integrator has to take in each NLP iteration. But the online method needs about the same time per iteration whereas the offline method is on average 3 times faster. This shows that the direct evaluation of the SIM in the online method is much more expensive than the evaluation of the interpolation object in the offline method (as expected).

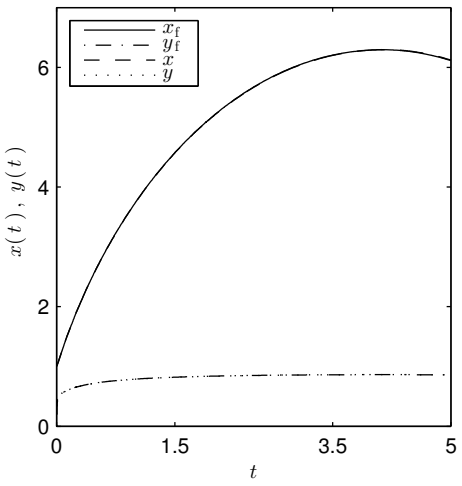

FIGURE 6.1. Example trajectories for the enzyme example for $\varepsilon = 1 \times 10^{-2}$ and $y(0) = 1$. Trajectories with subscript f are obtained using u_f with the full system, x and y are obtained using u_r with the full system. All trajectories overlap.

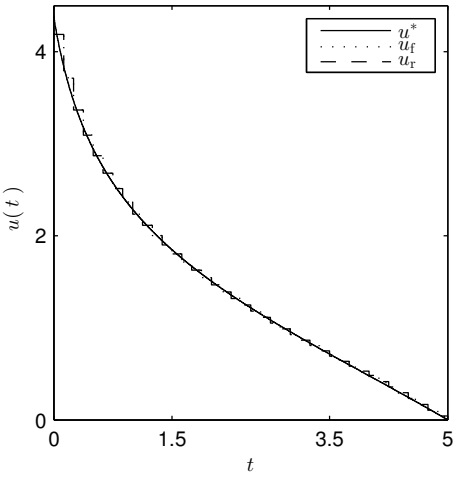

FIGURE 6.2. Example controls computed from the full problem (60), subscript f and the reduced problem (61), subscript r. As comparison the "exact" solution u^* from Example 7 is also included. We used $\varepsilon = 1 \times 10^{-2}$ and $y(0) = 1$. Both controls from the multiple shooting method overlap.

If we use the reduced control u_r, computed online or offline, for the full system the objective values $J(u_r)$ are virtually equal to the objectives $J(u_f)$ of the full system for all three different initial values for $y(0)$ that were tested. For example, for $y(0) = 0$ we have $J(u_f) = 188.7818$ for the full model. If we use the reduced control as input to the full system we also get $J(u_r) = 188.7812$ (online). The solution $u^*(t)$ from Example 7 produces $J(u^*) = 188.7852$. As one would expect $J(u^*) < J(u_f) < J(u_r)$, however only by a very small margin. In this example the error of the reduction is negligible which can also be concluded from the system itself. Example trajectories and controls for this case ($y(0) = 0$) are plotted in Figure 6.1 and 6.2 respectively. In both plots the subscript f refers to the results obtained using the computed control from the full system (60), whereas no subscript refers to the results using the control computed from the reduced problem (61).

We continue with the analysis of the runtime. To this end we are going to focus on the solutions obtained with the interpolator. The fastest solution of the optimal control problem took $\sigma_i = 0.9\,\mathrm{s}$ compared to $\sigma_f = 4.8\,\mathrm{s}$ (in the worst case). The difference $\sigma = \sigma_f - \sigma_i = 3.9\,\mathrm{s}$ is significant however, has to be compared to the time invested into the model reduction.

One big chunk is the initial preparation, like implementation of the model with the `MoRe` toolkit, numerical experiments to determine parameters (objective function and discretization method, T, t^*), and initial tests of the interpolation. This time can hardly be measured but with a little experience a test problem can be set up in reasonable time. If singularly perturbed systems are considered an automated procedure is conceivable that systematically would take the necessary steps.

Next, the scatter search with its 48 iterations needed around 400 s. This only has to be done once if one has an idea which region of interpolation parameters

would be fruitful to search in. The most time consuming part of the scatter search is the computation of the interpolation data. If a grid X with 40^2 interpolation nodes is used, one model reduction run takes about 370 s. To reduce this time we decomposed X into pairwise disjoint subsets X_i and solved the model reduction problem on each subset in parallel. With each X_i containing roughly 50 nodes and doing 8 jobs in parallel the runtime is down to 100 s. The job size, i.e. number of points in X_i is critical. If it is too small, the cost of managing and scheduling the parallel jobs gets disproportionally large. Also, because of the disruption of the continuation strategy the computational disadvantage of starting the `MoRe` algorithm new for every X_i might become noticeable. For example, using only 20 points in each X_i needs around 140 s, the same time needed for 200 nodes per X_i. The computational advantage of doing the model reduction in parallel becomes more important if higher dimensional slow state spaces are of interest.

The interpolation itself is comparable fast. Using the fastest set of interpolation data (30 points per direction) it takes only around 0.09 s to build the interpolation object. This means, if the model reduction data is pre-computed and available, the time needed to interpolate this data and to solve the optimal control problem combined is significantly smaller than the time used for solving the full problem.

In the introduction to this chapter we stated that two things have to be fulfilled in order to make model reduction in the context of optimal control a worthwhile approach. For one, the quality of the control computed from the reduced model must give a good enough performance if used for the full model and secondly, there must be a significant speed up. For the this specific example the mission is accomplished, the reduced control produces results that are nearly identical and using the interpolation, the computation of the reduced control can be nearly 5 times faster.

3. Voltage Regulator Example

In this section we revisit the voltage regulator from Examples 3, 4, and 18, originally based on Example 4.2 from [**69**]. The problem is given by

$$
\begin{aligned}
&\min \frac{1}{2}\int_0^2 x_1^2 + u^2 \mathrm{d}t \\
\text{s.t. } &\mathrm{D}_t\, x_1 = -\frac{1}{5}x_1 + \frac{1}{2}x_2, \\
&\mathrm{D}_t\, x_2 = -\frac{1}{2}x_2 + \frac{8}{5}y_1, \\
&\varepsilon\, \mathrm{D}_t\, y_1 = -\frac{5}{7}y_1 + \frac{30}{7}y_2, \\
&\varepsilon\, \mathrm{D}_t\, y_2 = -\frac{5}{4}y_2 + \frac{15}{4}y_3, \\
&\varepsilon\, \mathrm{D}_t\, y_3 = -\frac{1}{2}y_3 + \frac{3}{2}u.
\end{aligned} \tag{62}
$$

The problem, although in essence linear-quadratic, has some interesting features: The coupling between the slow and fast subsystems is only through the one fast state y_1 which means that only one state has to be reproduced during the optimization,

TABLE 6.2. Final objective values of problems (62) and (63) for a selection initial values and $\varepsilon = 0.2$.

$(x_1, x_2, y_1, y_2, y_3)$	$J(u_\mathrm{f})$	$J(u_\mathrm{r})$, online	$J(u_\mathrm{r})$, offline
$(-10, 0, 0, 0, 0)$	32.9	35.1	34.9
$(-10, 0, 10, 0, 10)$	25.1	521.7	612.7
$(-10, 0, 0, 10, 10)$	24.7	648.0	750.1
$(-10, 10, 10, 10, 10)$	20.4	782.6	830.3
$(-10, 10, 0, 0, 10)$	20.5	521.4	559.4
$(-10, 10, 10, 0, 10)$	20.0	579.1	619.5

i.e. we are only interested in $y_1 = h(x_1, x_2, u)$. The reduced problem is

$$\begin{aligned} &\min \frac{1}{2}\int_0^2 x_1^2 + u^2 \mathrm{d}t \\ \text{subject to: } &\mathrm{D}_t\, x_1 = -\frac{1}{5}x_1 + \frac{1}{2}x_2, \\ &\mathrm{D}_t\, x_2 = -\frac{1}{2}x_2 + \frac{8}{5}h(x_1, x_2, u). \end{aligned} \tag{63}$$

We saw in Example 3 that the system is fully controllable if $U = \mathbb{R}^{n_u}$. In case $U \neq \mathbb{R}^{n_u}$ controllability is lost (Example 4). However, this refers to the question if all states from $\mathbb{R}^{n_x} = \mathbb{R}^5$ could be reached using a constrained control. Our control problem (62) does not ask to drive some or all states to a certain value but aims at minimizing the control action and the divergence of x_1 from zero.

As in Example 18, we start with $\varepsilon = 2 \times 10^{-1}$ and solve both problems on 10 multiple shooting intervals with the `IPOPT` tolerance set to 10^{-3}. We already saw that for pure integration the time scale separation is not large enough and the model reduction error is intolerable. Nevertheless, we proceed to show how the reduced control might fail in the case of insufficient time scale difference.

A selection of initial values for the full system was used. An overview is given in Table 6.2. Note, that this selection leads to a set of two initial values for the reduced system, namely $\xi_1 = (-10, 0)^\mathrm{T}$ and $\xi_2 = (-10, 10)^\mathrm{T}$. The initial values for the state variables at the multiple shooting nodes are obtained through integrating the ode system with the initial control $u(t) = 0$. Bounds are introduced as follows: $x_1 \in [-20, 20]$, $x_2 \in [10, 50]$, $y_3, y_4, y_5 \in [-10^8, 10^8]$, and $u \in [-15, 15]$.

Using the control computed from the reduced problem for the full problem leads to extremely large objective values in comparison (as expected), see Table 6.2, and thus renders the model reduction unusable in this case. Only for ξ_1 and $\eta_1 = \big(y_1(0), y_2(0), y_3(0)\big) = (0, 0, 0)$ the final objectives are close because η_1 is the first order approximation of $h(\xi_1)$ and hence the fast variables start already on the SIM. Figures 6.3 and 6.4 compare trajectories from the full system using the full and reduced controls u_f and u_r respectively computed with the online method.

Additionally there is no runtime advantage: The full system needs on average 1.1 s compared to the online approach which needs 1.4 s which is also the median timing of the offline method.

If we increase the spectral gap by setting $\varepsilon = 2 \times 10^{-3}$ the results are much more favorable. First of all the computed input from the reduced model is very close to solution of the full problem. Therefore also the objective values are similar. See Figures 6.5 and 6.6 for an example. With the reduced model we find a significant computational advantage, as documented in Table 6.3. The runtime for the full problem depends strongly on the initial values and varies between 2.3 s to 5.1 s which is between 5 to 10 times slower compared to the fastest solution of the

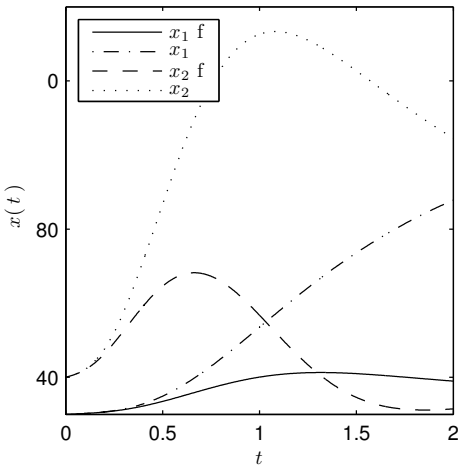

FIGURE 6.3. Example trajectories using the control from the full problem (62), indicated with f, and the reduced problem (63) with $\varepsilon = 0.2$, ξ_1, and η_3.

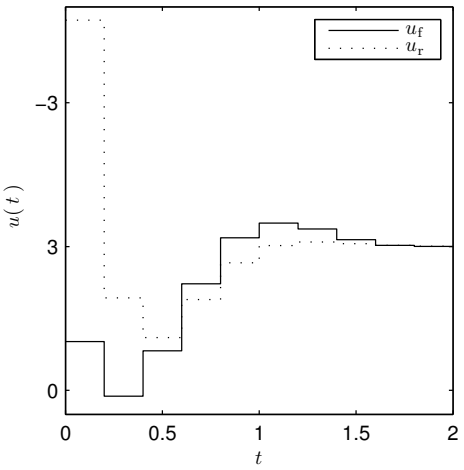

FIGURE 6.4. Example controls computed from the full problem (62), subscript f and the reduced problem (63), subscript r with $\varepsilon = 0.2$, ξ_1, η_3.

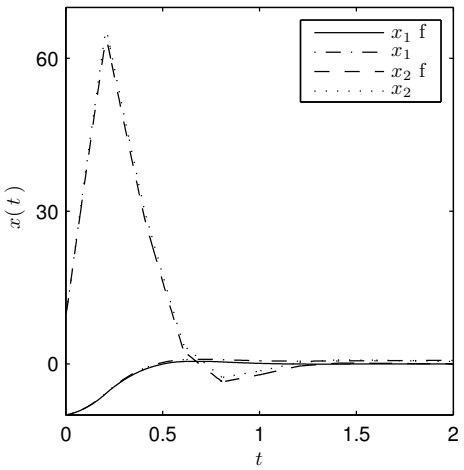

FIGURE 6.5. Example trajectories using the control from the full problem (62), indicated with f, and the reduced problem (63) with $\varepsilon = 2 \times 10^{-3}$.

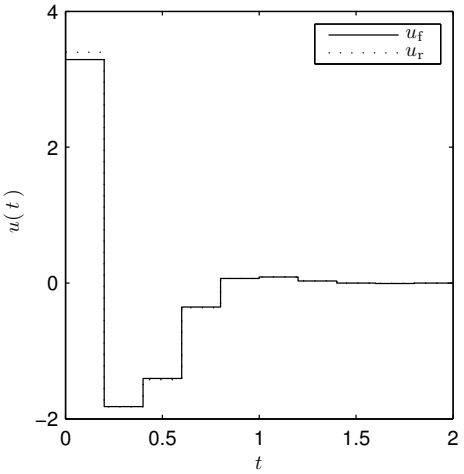

FIGURE 6.6. Example controls computed from the full problem (62), subscript f and the reduced problem (63), subscript r with $\varepsilon = 2 \times 10^{-3}$.

reduced problem with the offline method. The online approach is around 2 to 3 times faster. A further advantage worth mentioning in this context is that the time needed for the reduced problem is much less dependent on the initial values, which makes the computation more reliable in online control scenarios where the next input has to be computed within a given time frame.

As in the enzyme example we systematically tried various parameter combinations (points per dimension, overlap and points per patch) for the interpolator. We already mentioned the best and median runtime values, however it should also be noted that bad parameter combinations can decrease the algorithmic performance

TABLE 6.3. Summary of various statistics concerning the solution of problem (62) and (63) for $\varepsilon = 0.2 \times 10^{-3}$. Timings are averages over all initial values.

problem	time	NLP iter	time per iter
(62)	3.7 s	37.7	0.1 s
(61) online	1.6 s	16.5	0.1 s
(63) offline (median)	1.3 s	18	0.07 s
(63) offline (best)	0.5 s	16	0.03 s

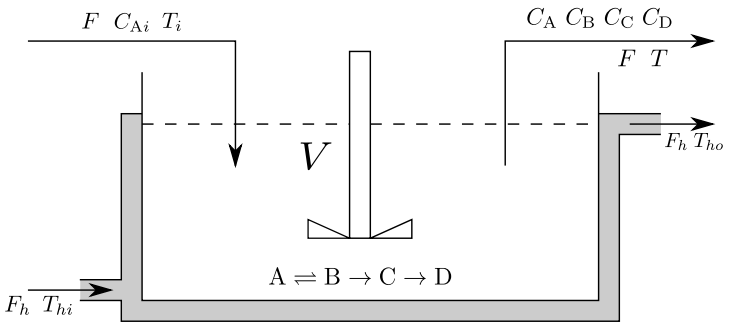

FIGURE 6.7. CSTR with heating jacket. See the text for an explanation of the symbols.

significantly. The maximum time needed for both ε values and sets of initial values was over 16 s. In a considerable number of cases the problem could not even be solved. This shows that the interpolator approach has to be tuned carefully but further analysis reveals that at least in this case the best configuration is the same for both initial values and ε.

4. CSTR Example

The last example of this chapter shows some of the limits of the approach. It has not been used in this work before, therefore we are going to introduce the background model. The example can be found in [**48**]. The model is based on a chemical reaction taking place in a *continuous stirred tank reactor* (CSTR). A CSTR is an idealized reactor type in which a chemical reaction takes place in a tank reactor where it is assumed that all components are perfectly mixed. The raw materials enter the reactor at a certain inflow rate and also the content of the reactor is removed with a certain outflow rate. Often the reactor can be cooled or heated with the help of a heating jacket. Common control variables are the inflow and outflow and the temperature of the heating jacket. The example we are referring to is depicted in Figure 6.7. The reaction taking place is

$$\mathrm{A} \rightleftharpoons \mathrm{B} \longrightarrow \mathrm{C} \longrightarrow \mathrm{D}$$

where A is the starting reactant and B is the desired product. The other species, C and D are unwanted by-products. The state variables are the concentrations

$$C_{\mathrm{A}}(t),\ C_{\mathrm{B}}(t),\ C_{\mathrm{C}}(t), C_{\mathrm{D}}(t)$$

of the chemicals in the reactor and the temperature $T(t)$ of the mixture (in the tank). The tank has the volume V which could be a state variable, however since we assume that the inflow equals the outflow, denoted with F the volume is constant. The concentration of A in the inflow stream is given by C_{Ai} and its temperature is T_{i}.

We are not considering the dynamics of the heating jacket but consider only the heat transfer to the reactor given by

$$Q = ha(T_{\mathrm{ho}} - T)$$

where h is the heat transfer coefficient and a is the surface area used for heat exchange. The heat transfer coefficient h is a parameter depending on the materials used for the heating jacket and the heating fluid. Using only Q is equivalent with assuming that the heat transfer is instantaneous, i.e. T will equal T_{ho} immediately.

Arrhenius kinetics are used to express the dependence of the reaction on the temperature T. To this end let T^0 be the nominal temperature of the reactor. The reaction rate constants are given by

$$k_i = k_i^0 \exp\left(\frac{E_i}{R}\left(\frac{1}{T^0} - \frac{1}{T}\right)\right), \quad i = 1, 2, 3,$$

where E_i is the activation energy and R is the ideal gas constant. The differential equations read

$$\begin{aligned}
\mathrm{D}_t C_{\mathrm{A}} =& \frac{F}{V}(C_{\mathrm{Ai}} - C_{\mathrm{A}}) - k_1\left(C_{\mathrm{A}} - \frac{C_{\mathrm{B}}}{\kappa_1}\right),\\
\mathrm{D}_t C_{\mathrm{B}} =& -\frac{F}{V}C_{\mathrm{B}} + k_1\left(C_{\mathrm{A}} - \frac{C_{\mathrm{B}}}{\kappa_1}\right) - k_2 C_{\mathrm{B}},\\
\mathrm{D}_t C_{\mathrm{C}} =& -\frac{F}{V}C_{\mathrm{C}} + k_2 C_{\mathrm{B}} - k_3 C_{\mathrm{C}},\\
\mathrm{D}_t C_{\mathrm{D}} =& -\frac{F}{V}C_{\mathrm{D}} + k_3 C_{\mathrm{C}},\\
\mathrm{D}_t T =& -\frac{F}{V}(T_i - T) - k_1\left(C_{\mathrm{A}} - \frac{C_{\mathrm{B}}}{\kappa_1}\right)\frac{\Delta H_{r1}}{\rho c_p} -\\
& k_2 C_{\mathrm{B}}\frac{\Delta H_{r2}}{\rho c_p} - k_3 C_{\mathrm{C}}\frac{\Delta H_{r3}}{\rho c_p} - \frac{Q}{\rho V c_p}.
\end{aligned} \tag{64}$$

For an overview over the states and their stable steady state values and the nominal parameters and see Table 6.4. The differential equation for the temperature depends on the reaction heats

$$\Delta H_{ri} = \Delta H_{ri}^0 + \Delta c_{pi}(T - T^0), \quad i = 1, 2, 3$$

where

$$\Delta c_{p1} = c_{p\mathrm{B}} - c_{p\mathrm{A}},\ \Delta c_{p2} = c_{p\mathrm{C}} - c_{p\mathrm{B}},\ \Delta c_{p3} = c_{p\mathrm{D}} - c_{p\mathrm{C}}.$$

Finally, the reaction equilibrium constant κ_1 is given through

$$\log\left(\frac{\kappa_1}{\kappa_1^0}\right) = \left(\frac{\Delta H_{r1}^0 - \Delta c_{p1} T^0}{R}\right)\left(\frac{1}{T^0} - \frac{1}{T}\right) + \left(\frac{\Delta c_{p1}}{R}\right)\log\left(\frac{T}{T^0}\right).$$

When using the nominal values from Table 6.4 the reactor works at a stable set point.

Obviously, the right hand side is highly nonlinear and more complex as in the previous examples. The first step to apply model reduction techniques is to chose fast and slow states. In this case looking at the pre-exponential factors for all three reactions (Table 6.4) we find that $k_2^0 \ll k_1^0, k_3^0$ and therefore the first and third reaction are much faster than the second. This means A and C are converted to B and D much faster than B to C and we have $x = (C_{\mathrm{B}}, C_{\mathrm{D}}, T)$ and $y = (C_{\mathrm{A}}, C_{\mathrm{C}})$.

A control scenario is introduced into the setting by demanding a certain percentage change $\alpha > 0$ in the set point for B. In the original example there are two control variables: The inflow F and the heat duty Q. We restrict ourselves to

TABLE 6.4. Parameters and nominal values for the CSTR example.

parameter	description	nominal value
C_{Ai}	molecular concentration, A, inflow	$10\,\mathrm{mol\,l^{-1}}$
C_{A}	molecular concentration, A	$2.325\,\mathrm{mol\,l^{-1}}$
C_{B}	molecular concentration, B	$5.756\,\mathrm{mol\,l^{-1}}$
C_{C}	molecular concentration, C	$0.003\,\mathrm{mol\,l^{-1}}$
C_{D}	molecular concentration, D	$1.916\,\mathrm{mol\,l^{-1}}$
T	reactor temperature	$300\,\mathrm{K}$
c_p	specific heat capacity, liquid	$5\,\mathrm{kJ\,kg^{-1}\,K^{-1}}$
$c_{p\mathrm{A}}$	molar heat capacity, A	$120\,\mathrm{J\,mol^{-1}\,K^{-1}}$
$c_{p\mathrm{B}}$	molar heat capacity, B	$80\,\mathrm{J\,mol^{-1}\,K^{-1}}$
$c_{p\mathrm{C}}$	molar heat capacity, C	$70\,\mathrm{J\,mol^{-1}\,K^{-1}}$
$c_{p\mathrm{D}}$	molar heat capacity, D	$140\,\mathrm{J\,mol^{-1}\,K^{-1}}$
E_1	activation energy, 1st reaction	$45\,\mathrm{kJ\,mol^{-1}}$
E_2	activation energy, 2nd reaction	$35\,\mathrm{kJ\,mol^{-1}}$
E_3	activation energy, 3rd reaction	$40\,\mathrm{kJ\,mol^{-1}}$
F	inflow rate	$3\,\mathrm{l\,min^{-1}}$
k_1^0	pre-exponential factor, T^0 value, 1st reaction	$100\,\mathrm{min^{-1}}$
k_2^0	pre-exponential factor, T^0 value, 2nd reaction	$0.1\,\mathrm{min^{-1}}$
k_3^0	pre-exponential factor, T^0 value, 3rd reaction	$200\,\mathrm{min^{-1}}$
T_i	temperature inflow stream	$290\,\mathrm{K}$
Q	heat duty	$58.42\,\mathrm{kJ\,min^{-1}}$
V	reactor volume	$10\,\mathrm{l}$
κ_1^0	equilibrium constant, 1st reaction	2.5
ρ	liquid density	$0.8\,\mathrm{kg\,l^{-1}}$
ΔH_{r1}^0	heat of reaction, T^0 value, 1st reaction	$-6\,\mathrm{kJ\,mol^{-1}}$
ΔH_{r2}^0	heat of reaction, T^0 value, 2nd reaction	$-5\,\mathrm{kJ\,mol^{-1}}$
ΔH_{r3}^0	heat of reaction, T^0 value, 3rd reaction	$-2\,\mathrm{kJ\,mol^{-1}}$

one input variable, namely the inflow, thus we replace F with $u(t)$ in (64). As a possible objective we use

$$\min_u \int_0^{90} C_{\mathrm{B}}(t) - \alpha 5.756 \,\mathrm{d}t. \tag{65}$$

For example $\alpha = 1.05$ would correspond to a 5 % increase in the set point for B. Additionally, $U = [1, 10]$ and all initial values correspond to the stable stationary point.

We start with comparing the full solution of the problem to the solution produced with the online approach and come back to the offline method later. Using 9 multiple shooting intervals, we get the results depicted in Figures 6.8 and 6.9. The objective values $J(u_{\mathrm{f}})$ and $J(u_{\mathrm{r}})$ are both approximately 0.072 and thus the reduced model can be used as basis to compute useful controls. This continues to hold if other values for α in (65) and different numbers of multiple shooting nodes.

However, the second point of increased efficiency is not fulfilled here. For the example given, the solution based on the full model needs 1.63 s which is significantly less than the 5.94 s for the reduced model. Most of the time is spend evaluating the NLP function, i.e. solving the initial value problems in the multiple shooting approach. This takes around 5 of the nearly 6 seconds. This is surprising, since if we compare the integrator statistics, see Table 6.5, the reduced model is clearly more efficiently solved. This means that evaluating the right hand side function

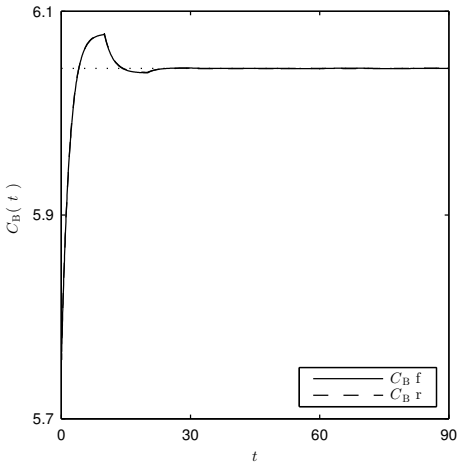

FIGURE 6.8. Example trajectories for the CSTR example. The dotted line represents the new set point. The trajectories overlap.

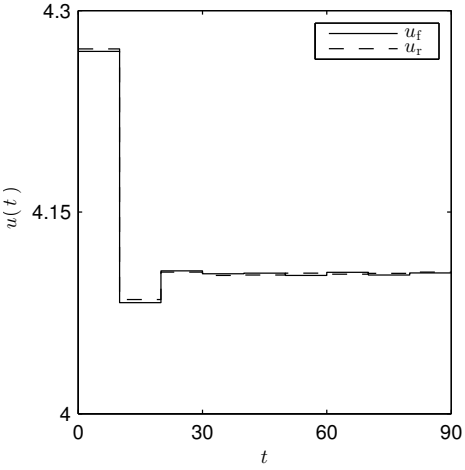

FIGURE 6.9. Control for the CSTR example. The lines partially overlap.

TABLE 6.5. Integrator statistics for the CSTR example using the full model and the online approach.

problem	steps	rej. steps	rej. Newton steps	Jacobian updates
full	16 627	2089	6	779
offline	7106	1093	0	413

f inside the integrator is much more expensive for the reduced model than for the full model. The generalized Gauss-Newton algorithm inside the online model reduction needs a median number of 1 step per call, however, Figure 6.10 shows that as many as over 40 steps are sometimes necessary to find a solution. We already mentioned that for the offline method, the full right hand side of the differential equation has to be evaluated in order to compute $y = h(x)$. In this case, the right hand side is complex and expensive to evaluate and hence even if the integration needs less steps for the reduced model the cost of evaluating the model equation eats this advantage away in the offline model reduction. One way to overcome this problem is to integrate the model reduction into the integrator. This would save a considerable amount of function and derivative evaluations because the information could be used for integration and model reduction.

Getting rid of the evaluation of a probably complex model equation is one of the benefits of using the offline approach. We did not include it in the discussion of our results for the CSTR, yet since it fails to deliver reliable results here. Even for the simple case of integrating the stable stationary point in absence of a control the interpolation error (in the order of 1×10^{-4}) leads to a loss of significance in the computation of the right hand side for B and the integrator eventually fails. This can be observed with different interpolator parameters, however it could of course be possible that a certain combination of overlap factor, points per patch, and other parameters would lead to a feasible interpolation. A problem with testing different configurations for this problem is albeit the time needed to generate and even interpolate the model reduction data. The function h is $\mathbb{R}^4 \to \mathbb{R}^2$ and using

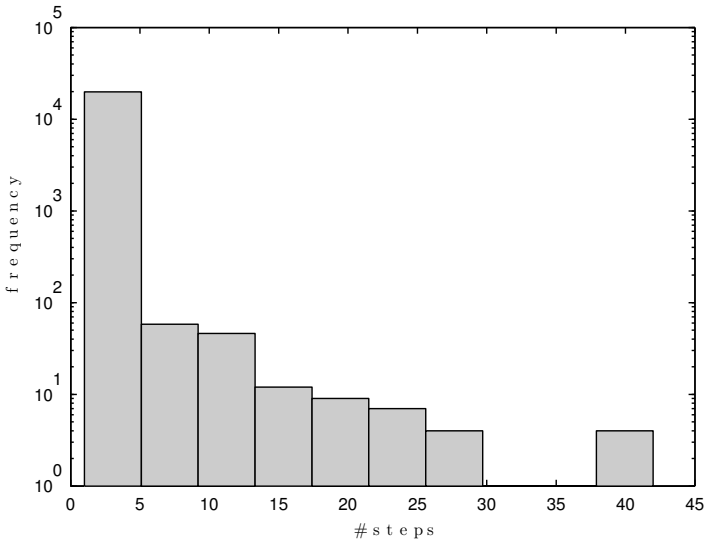

FIGURE 6.10. Number of steps the generalized Gauss-Newton procedure needs to converge for solving the reduced optimal control problem.

30 points in each direction leads to a grid with 810 000 points. Computing the manifold data needs 9.3 h and one interpolation including the scale optimization already needs around 15.7 h. Note that since there are two output dimensions the shape parameter optimization has to be done twice. A scatter search test as used in the other examples to find the optimal interpolation parameters would easily take several days, especially if it turns out that 30 points for each reaction progress variable are not enough.

Clearly, the offline method is not applicable for problems of this complexity in the current state. The two, partially connected problems are accuracy and parameter determination. They are partially connected because optimal parameters decrease the interpolation error. Beyond optimal parameters the accuracy can be increased by other means:

- Using other sets than Cartesian grids for the interpolation nodes.
- Use other basis functions.
- Use a more robust data structure for the partition of unity.
- Use smoother weighting functions in the partition of unity.
- Use a more robust scaling parameter optimization.

We mentioned that testing the interpolation takes far to long for problems of this size and besides making the interpolation process itself faster, for example via parallelization of the shape parameter optimization, an efficient method for determining the quality of the interpolation would be of great benefit.

In general this problem shows the limitations of the current implementation more than the limits of the method itself, since the optimal control results using the reduced model are fine. It is included here to show why and where improvements are necessary and could serve as a benchmark problem for reduced optimal control.

5. Summary

The topic of this chapter was the application of the numerical tools to 3 different examples. The key deliverables for model reduction in the context of optimal control are quality of the computed control and computational efficiency. The control

obtained using the reduced system has to produce a result close enough to the performance of the control from the full system and the computation of the reduced control should be faster than computing the full control. If both points are fulfilled, model reduction is a valuable tool in treating complex optimal control problems. Additionally to the sheer computation of the control one has always to consider the time of preparing the reduced model, the model reduction and the offline or online method for providing $h(x, u)$ when evaluating the usefulness of the approach.

The first example is based on an enzyme catalyzed reaction and is in explicit singularly perturbed form. Both modes (online and offline) produce controls that are on par with the full control in terms of objective performance. Additionally the offline mode provides computation of a control solution up to 5 times faster compared to the full problem. The reason for the speed up is mainly due to reduced effort in the numerical integration routine.

The second example, based on a voltage regulator, also in singularly perturbed form displayed how the model reduction approach fails, if the time scale separation is to small. Although the computation with the reduced model is as fast as using the full model, the reduced control leads to unusable results if applied to the full system. Only if the time scale separation is large enough we find similar results to the enzyme example in terms of quality and speed up.

The last example is a complex nonlinear model of a CSTR. Here, the time scale separation is large enough and the control results obtained with the online mode are very close to the results obtained with the full system. However, there is a significant computational disadvantage in using the model reduction. This is due to the expensive evaluation of the right hand side of the differential equation which makes the model reduction in its current implementation state too costly. The offline method also fails because of numerical instabilities caused by the interpolation error.

All three examples show how model reduction can be used to solve concrete optimal control problems. They were chosen to show how model reduction can be applied to the advantage of the user of such methods but also reveal where and the current implementation fails.

CHAPTER 7

Summary, Conclusions, and Outlook

1. Summary and Conclusion

Singularly perturbed systems based on ordinary differential equations are studied to analyze the mathematical implications of time scale separation for initial and boundary value problems. In two time scale singularly perturbed systems the time scale separation is made explicit through a small parameter ε, $0 < \varepsilon \ll 1$ and the state space is decomposed in slow modes evolving on a time scale $\mathcal{O}(1)$ and fast modes which evolve on $\mathcal{O}(\varepsilon)$. The explicit time scale structure of singularly perturbed systems can be exploited to the effect that two lower order systems (slow and fast) can be regarded independently. This way initial and boundary value problems can be analyzed and solutions can be represented as Taylor-like series in the parameter ε. Furthermore, a lower order, invariant manifold in the state space of the full system, parametrized by the slow states exists. This manifold is given through a function $y = h(x, \varepsilon)$ and again this function can be expanded into an ε series.

Systems open to external input are the topic of mathematical control theory. A central question is, if certain or all states of the state space can be reached by applying suitable controls. The issue of controllability is discussed separately for linear and non-linear systems because different tools are applied to each class. For linear systems controllability can be checked in the unconstrained control case by applying a rank condition to the controllability matrix. If controls are constrained, then additionally the uncontrolled system has to be stable. Nonlinear systems are analyzed using Lie-algebra methods and tools from differential geometry. We state a rank condition based on the repeated application of Lie brackets. The result is local and provides the existence of a non-empty neighborhood of points that can be reached around a point x_0. Optimal control problems and the Pontryagin minimum principle are introduced. It is shown, how the minimum principle can be used to formulate a boundary value problem for the states and co-states. If singularly perturbed systems are used in optimal control problems, the time scale structure transfers to the boundary value problem from the minimum principle and therefore also the optimal control solution exhibits a decomposition into slow and fast components. The slow manifold approach eliminates the fast modes from the system and thus also from the optimal control solution. Only the outer control solution without the boundary layer correction is generally obtainable in this case. Finally, numerical approaches to optimal control are discussed and especially the multiple shooting method is described in more detail. The infinite optimization problem is approximated by a finite dimensional NLP and solved with the help of a BDF integration routine and the NLP solver `IPOPT`.

Eventually, an approximation of the slow manifold is designated to be used in the numerical optimal control program. One way of providing a differentiable pointwise approximation of the manifold is to use multidimensional interpolation of sample data. To this end radial basis functions are introduced which allow interpolation on grid free data independent from the input dimension. Depending

on the basis function, exponential convergence of interpolants to functions from the native space of the basis function can be theoretically provided. However, the interpolation problem is numerically unstable and a high approximation quality can only be attained if the interpolation nodes are not too close to each other. The trade of between stability and quality can be balanced (in a certain range) by an additional shape parameter. This shape parameter is optimally chosen by a computationally efficient modified leave-one-out scheme. Finally, the evaluation of the interpolation object is made independent of the number of nodes by a partition of unity that decomposes the input domain into overlapping subdomains containing a constant number of points.

The numerical model reduction approach used in this work is based on approximating slow invariant manifolds (SIMs) in the state space of a model. They are a tool that can be used not only for singularly perturbed systems but also for more general systems where the time scale separation is not explicitly given. The model reduction problem is formulated as a minimization problem, based on the invariance of the slow manifold and the observation that the fast states relax to the SIM exponentially fast which implies that curvature is minimal for trajectories on the SIM. Two objectives are introduced, a global one, that is based on the integrated curvature of actual trajectory pieces and a local version. The global problem is discretized with a shooting approach whereas the local problem is solved with a generalized Gauss-Newton procedure. In any case the manifold and sensitivities with respect to the reaction progress variables are approximated pointwise. One way to apply the model reduction algorithm to control systems is to augment the differential equation system with a new variable representing the control parametrized with a finite number of parameters. This way, the control is eliminated from the system and the model reduction can be used without modification. Eventually, the manifold information is used to replace the fast variable in optimal control scenarios. To this end either the interpolation is used to approximate pre-computed data in an offline mode or the model reduction problem is solved for each point during the computation of the optimal control (online mode).

Finally, numerical results for three examples are given. There are two main issues that have to be considered for model reduction in the context of optimal control: The performance of the reduced optimal control solution in the full problem and the computational benefit of using the reduced system. The reduced control has to be "good enough" and its computation significantly faster. For the two singularly perturbed examples both goals are reached. Especially the offline mode, using the interpolation, provides a significant speed up and the reduced control leads to nearly indistinguishable results if ε is small. The third example fails to fulfill the speedup requirement for the online mode and fails completely in the offline mode due to loss of significance because of the interpolation error.

In conclusion, this work collected and integrated the theoretical background to show that model order reduction based on time scale separation is a fruitful approach. Based on the theoretical results numerical procedures are developed and implemented that provide ways to solve reduced optimal control problems faster than their full counterparts.

The numerical model reduction procedure allows to integrate the analytical results into standard optimal control algorithms. The method reliefs from carrying out the analytical computations to obtain a reduced system for singularly perturbed problems or even enables the use of model reduction at all for general systems without explicit time scale separation.

RBF interpolation is shown to be a reasonable choice for approximating the SIM information. Its grid and input dimension independence makes it easily integrable

and combinable with the model reduction toolkit. The use of the shape parameter optimization and the partition of unity approach lets the interpolator work robustly, reliably and fast on large data sets.

The results chapter showed that for practical examples there is a significant speed up due to the model reduction in computing the optimal control using the interpolator. From this experience and with the theoretical results in background it can be concluded that reduced optimal control is a valid strategy to produce useful results whilst using less computational resources.

2. Outlook

This work only represents the first step in making model reduction based on SIMs a viable option for real world optimal control applications. In many aspects it remains a proof of concept and the theoretical fundamentals as well as the numerical analysis and the actual implementation have to be developed further.

Singularly perturbed systems make time scale separation obvious but for general nonlinear system the theoretical background is less pronounced. From experience with the SIM based model reduction and the knowledge that is available it is clear that many concepts are analog. It should be clarified if and how a SIM can be defined and characterized formally for general systems. Additionally, the connection between general and singularly perturbed systems should be further explored. This would strengthen the theoretical understanding of using SIMs in optimal control scenarios. Other open problems have to do with adaptivity. We already mentioned the problem of systems with several time scales which demand for adjusting the order of the reduced model throughout the optimal control procedure. A related issue is that for general systems the SIM might cease to exist over some parts of the domain. Again the question turns up how this could be handled and what this means for the approximation quality of the reduced model.

This topic is also reaching over to the numerical domain. Any algorithm adapting to the order of the reduced model would depend on an online error estimation. This estimation has to be fast enough to not spoil the speed gain of model reduction. This could probably be achieved by including the online estimation of the SIM into the integration routine and the optimal control algorithm. Since full right-hand side information would be available an estimator for the time scale contribution for each state could be constructed. In any way, integrating the model reduction into the integrator would greatly benefit the performance of the online approach. As it was already mentioned this is similar to solving differential-algebraic equations numerically and it should be investigated if techniques from that area (e.g. adaptive step size and integration order) could be reused or adjusted.

By now the interpolation only uses the Gaussian kernel and problems might occur because of its small native space and numerical instability with regard to small fill distances. Although the scale parameter optimization and the partition of unity seem to make these problems less severe in practice other kernels might provide better performance. Transformed kernels, as described in [**91**, Section 12.3] have stability properties that do not depend on the fill distance but the number of interpolation nodes. Besides extending the interpolator in this direction it would also be helpful to have tools available that could be used to check the quality of the interpolation. A leave-one-out approach as in the shape parameter optimization could be used to check the output of the interpolator against the output of the model reduction routine.

A big chunk of possible future work concerns the implementation. The key aspect here is integration of the tools. So far, the model reduction toolkit `MoRe`, the interpolator, the BDF integrator, and the optimal control software `DOT` are all

independent programs or libraries. As already mentioned, connecting `MoRe` with the interpolator could increase performance. Integrating `MoRe` and the interpolator into `DOT` would improve the user experience, for example because implementing the model equation would only have to be done in one place. Overall, during the numerical experiments it turned out that organizing all the tools and the data transfer between them takes a considerable amount of time and effort and often leads to subtle bugs and errors. Generally, establishing or using standardized interfaces and data exchange formats would greatly contribute to the usability of the software.

Finally, the approach of using reduced models in optimal control has to be tested and extended to larger, more complex, and realistic control scenarios. A worthwhile goal seems to be the combination with nonlinear model predictive control as already described before. To this end one should also consider if ignoring the fast modes completely as it is done now is sufficient for realistic control scenarios. If measurements of the fast modes are available a feedback component could be added to the controller that drives the fast modes close to the SIM.

Bibliography

[1] Andrei A. Agrachev and Yuri Sachkov, *Optimal control from the geometric viewpoint*, Springer, 2004.

[2] A. Astolfi, *Model reduction by moment matching for linear and nonlinear systems*, Automatic Control, IEEE Transactions on **55** (2010), no. 10, 2321 –2336.

[3] Bradley M. Bell, *Automatic differentiation software cppad.*, 2010.

[4] Richard Ernest Bellman, *Dynamic programming*, Dover Publications Inc., 1957.

[5] S. V. Belokopytov and M. G. Dmitriev, *Direct scheme in optimal control problems with fast and slow motions*, Systems & Control Letters **8** (1986), no. 2, 129–135.

[6] Hans Georg Bock, *Numerical treatment of inverse problems in chemical reaction kinetics*, Modelling of Chemical Reaction Systems (K. H. Ebert, P. Deuflhard, and W. Jäger, eds.), Springer Series in Chemical Physics, vol. 18, Springer, Heidelberg, 1981, pp. 102–125.

[7] ———, *Recent advances in parameter identification techniques for ODE*, Numerical Treatment of Inverse Problems in Differential and Integral Equations (P. Deuflhard and E. Hairer, eds.), Birkhäuser, Boston, 1983, pp. 95–121.

[8] ———, *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*, Bonner Mathematische Schriften, vol. 183, University of Bonn, 1987.

[9] Hans Georg Bock and Karl J. Plitt, *A multiple shooting algorithm for direct solution of optimal control problems*, Proceedings of the Ninth IFAC World Congress, Budapest, Pergamon, Oxford, 1984.

[10] Carl Boor and Amos Ron, *On multivariate polynomial interpolation*, Constructive Approximation **6** (1990), 287–302.

[11] Boost Project, *Ublas numeric bindings*, `http://svn.boost.org/svn/boost/sandbox/numeric_bindings/boost/numeric/bindings/`.

[12] AE Bryson and YC Ho, *Applied optimal control: Optimization, estimation and control*, Taylor and Francis, London, 1975.

[13] G. M. Constantine and T. H. Savits, *A multivariate Faà di Bruno formula with applications*, Trans. Amer. Math. Soc. **348** (1996), no. 2, 503–520. MR 1325915 (96g:05008)

[14] Philip J. Davis, *Interpolation & approximation*, Dover Publications Inc., 1975.

[15] Peter Deuflhard and Folkmar Bornemann, *Numerische mathematik II: Gewöhnliche differentialgleichungen*, third ed., Walter de Gruyter, Berlin, 2008.

[16] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, *Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations*, Journal of Process Control **12** (2002), no. 4, 577–585.

[17] M. Dmitriev and G. Kurina, *Singular perturbations in control problems*, Automation and Remote Control **67** (2006), no. 1, 1–43.

[18] Markus Esenwein, *On the implementation of a direct numerical method for optimal control of ordinary differential equations*, Master's thesis, Freiburg University, 2011.

[19] Gregory Fasshauer and Qi Ye, *Reproducing kernels of generalized sobolev spaces via a green function approach with distributional operators*, Numerische Mathematik **119** (2011), 585–611, 10.1007/s00211-011-0391-2.

[20] H. O. Fattorini, *Infinite dimensional optimization and control theory*, Encyclopedia of Mahtematics and its Applications, vol. 62, Cambridge University Press, 1999.

[21] Neil Fenichel, *Geometric singular perturbation theory for ordinary differential equations*, Journal of Differential Equations **31** (1979), 53–98.

[22] Anthony V. Fiacco, *Sensitivity analysis for nonlinear programming using penalty methods*, Mathematical Programming **10** (1976), no. 1, 287–311.

[23] Rolf Findeisen, Lars Imsland, Frank Allgöwer, and Bjarne A. Foss, *State and output feedback nonlinear model predictive control: An overview*, European Journal of Control **9** (2003), no. 2–3, 190–207.

[24] B. Fornberg, T.A. Driscoll, G. Wright, and R. Charles, *Observations on the behavior of radial basis function approximations near boundaries*, Computers & Mathematics with Applications **43** (2002), no. 3–5, 473 – 490.

[25] Anders Forsgren, Philip E. Gill, and Margaret H. Wright, *Interior methods for nonlinear optimization*, SIAM Review **44** (2002), no. 4, 525–597.

[26] K. Fujimoto and J. Scherpen, *Balanced realization and model order reduction for nonlinear systems based on singular value analysis*, SIAM Journal on Control and Optimization **48** (2010), no. 7, 4591–4623.

[27] Ronald Garcia, Jeremy Siek, and Andrew Lumsdaine, `http://www.boost.org/doc/libs/1_52_0/libs/multi_array/doc/index.html`.

[28] Mariano Gasca and Thomas Sauer, *On the history of multivariate polynomial interpolation*, Journal of Computational and Applied Mathematics **122** (2000), no. 1-2, 23 – 35.

[29] Mariano Gasca and Thomas Sauer, *Polynomial interpolation in several variables*, Advances in Computational Mathematics **12** (2000), 377–410, 10.1023/A:1018981505752.

[30] C. W. Gear, T. J. Kaper, I. G. Kevrekidis, and A. Zagaris, *Projecting to a slow manifold: Singularly perturbed systems and legacy codes*, SIAM Journal on Applied Dynamical Systems **4** (2005), no. 3, 711–732.

[31] C. Geiger and C. Kanzow, *Numerische verfahren zur lösung unrestringierter optimierungsaufgaben*, Springer-Verlag, Berlin, 1999.

[32] Mathias Gerds, Roland Herzog, and Dirk Lebiedz, *Skript zur vorlesung kontrolltheorie und optimale steuerung*, Lecture notes by Dirk Lebiedz, winter term 2010/2011, Freiburg University.

[33] Alexander N. Gorban, Iliya V. Karlin, and Andrei Yu. Zinovyev, *Constructive methods of invariant manifolds for kinetic problems*, Physics Reports **396** (2004), 197–403.

[34] Alexander N. Gorban, Nikolaos K. Kazantzis, Ioannis G. Kevrekidis, Hans Christian Öttinger, and Constantinos Theodoropoulos, *Model reduction and coarse-graining approaches for multi-scale phenomena*, Springer, 2007.

[35] Andreas Griewank, *Evaluating derivatives: Principles and techniques of algorithmic differentiation*, Frontiers in Appl. Math., no. 19, SIAM, Philadelphia, PA, 2000.

[36] Juergen Hahn and Thomas F. Edgar, *An improved method for nonlinear model reduction using balancing of empirical gramians*, Computers & Chemical Engineering **26** (2002), no. 10, 1379 – 1397.

[37] Ernst Hairer and Gerhard Wanner, *Solving ordinary differential equations II: Stiff and differential-algebraic problems*, first ed., Springer Series in Computational Mathematics, no. 14, Springer, New York, 1991.

[38] Richard F. Hartl, Suresh P. Sethi, and Raymond G. Vickson, *A survey of the maximum principles for optimal control problems with state constraints*, SIAM Review **37** (1995), no. 2, 181–218.

[39] Harro Heuser, *Gewöhnliche differentialgleichungen*, 4 ed., Teubner, 2004.

[40] Frank Hoppensteadt, *Properties of solutions of ordinary differential equations with small parameters*, Communications on Pure and Applied Mathematics **24** (1971), 807–840.

[41] Warren P. Johnson, *The curious history of faà di bruno's formula*, Amer. Math. Monthly **109** (2002), 217–234.

[42] Christopher K. R. T. Jones, *Geometric singular perturbation theory*, Dynamical Systems (Russell Johnson, ed.), Lecture Notes in Mathematics, no. 1609, Springer, Heidelberg, 1995, pp. 44–118.

[43] Julia Kammerer, *Numerische verfahren zur dynamischen komplexitätsreduktion biochemischer reaktionssysteme*, Ph.D. thesis, University of Heidelberg, Heidelberg, Germany, October 2007.

[44] Hans G. Kaper and Tasso Joost Kaper, *Asymptotic analysis of two reduction methods for systems of chemical reactions*, Physica D **165** (2002), 66–93.

[45] Gaetan Kerschen, Jean-Claude Golinval, Alexander F. Vakakis, and Lawrence A. Bergman, *The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: An overview*, Nonlinear Dynamics **41** (2005), 147–169, 10.1007/s11071-005-2803-2.

[46] Petar V. Kokotović, *Applications of singular perturbation techniques to control problems*, SIAM Review **26** (1984), no. 4, 501–550.

[47] L. I. Kononenko and V. A. Sobolev, *Asymptotic decomposition of slow integral manifolds*, Siberian Mathematical Journal **35** (1994), no. 6, 1119–1132.

[48] Aditya Kumar, Panagiotis D. Christofides, and Prodromos Daoutidis, *Singular perturbation modeling of nonlinear processes with nonexplicit time-scale multiplicity*, Chemical Engineering Science **53** (1998), no. 8, 1491 – 1504.

[49] Karl Kunisch and Stefan Volkwein, *Proper orthogonal decomposition for optimality systems*, ESAIM: Mathematical Modelling and Numerical Analysis **42** (2008), no. 01, 1–23.

[50] Sanjay Lall, Jerrold E. Marsden, and Sonja Glavaški, *A subspace approach to balanced truncation for model reduction of nonlinear control systems*, International Journal of Robust and Nonlinear Control **12** (2002), no. 6, 519–535.

[51] S. H. Lam and D. A. Goussis, *The CSP method for simplifying kinetics*, International Journal of Chemical Kinetics **26** (1994), 461–486.

[52] Dirk Lebiedz, *Computing minimal entropy production trajectories: An approach to model reduction in chemical kinetics*, Journal of Chemical Physics **120** (2004), no. 15, 6890–6897.

[53] Dirk Lebiedz, Julia Kammerer, and Ulrich Brandt-Pollmann, *Automatic network coupling analysis for dynamical systems based on detailed kinetic models*, Physical Review E **72** (2005), no. 4, 041911.

[54] Dirk Lebiedz, Volkmar Reinhardt, and Jochen Siehr, *Minimal curvature trajectories: Riemannian geometry concepts for slow manifold computation in chemical kinetics*, Journal of Computational Physics **229** (2010), no. 18, 6512–6533.

[55] Dirk Lebiedz, Volkmar Reinhardt, Jochen Siehr, and Jonas Unger, *Geometric criteria for model reduction in chemical kinetics via optimization of trajectories*, Coping with Complexity: Model Reduction and Data Analysis (Alexander N. Gorban and Dirk Roose, eds.), Lecture Notes in Computational Science and Engineering, no. 75, Springer, Heidelberg, first ed., 2011, pp. 241–252.

[56] Dirk Lebiedz, Jochen Siehr, and Jonas Unger, *A variational principle for computing slow invariant manifolds in dissipative dynamical systems*, SIAM Journal on Scientific Computing **33** (2011), no. 2, 703–720.

[57] Daniel B. Leineweber, Irene Bauer, Hans Georg Bock, and Johannes P. Schlöder, *An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part I: theoretical aspects*, Computers and Chemical Engineering **27** (2003), 157–166.

[58] Daniel B. Leineweber, Andreas Schäfer, Hans Georg Bock, and Johannes P. Schlöder, *An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part II: Software aspects and applications*, Computers and Chemical Engineering **27** (2003), 167–174.

[59] R. A. Lorentz, *Multivariate hermite interpolation by algebraic polynomials: A survey*, Journal of Computational and Applied Mathematics **122** (2000), no. 1-2, 167 – 201.

[60] David G. Luenberger, *Linear and nonlinear programming*, second ed., Addison-Wesley, Reading, 1984.

[61] Jack Macki and Aaron Strauss, *Introduction to optimal control theory*, Springer, 1995.

[62] John Maddock, Paul A. Bristow, Hubert Holin, Xiaogang Zhang, Bruno Lalande, Johan Råde, Gautam Sewani, Thijs van den Berg, and Benjamin Sobotta, *Locating function minima: Brent's algorithm*, 2012, `http://www.boost.org/doc/libs/1_52_0/libs/math/doc/sf_and_dist/html/math_toolkit/toolkit/internals1/minima.html`.

[63] R. Marino and P.V. Kokotovic, *A geometric approach to nonlinear singularly perturbed control systems*, Automatica **24** (1988), no. 1, 31 – 41.

[64] W. Marquardt, *Nonlinear model reduction for optimization based control of transient chemical processes*, Chemical Process Control VI, Tuscon, Arizona, 7-12.1.2001 (J.W. Eaton J. B. Rawlings, B.A. Ogunnaike, ed.), no. 326, 2002, pp. 12–42.

[65] Michael McAsey and Libin Mou, *A proof of a general maximum principle for optimal controls via a multiplier rule on metric space*, Journal of Mathematical Analysis and Applications **337** (2008), no. 2, 1072 – 1088.

[66] Kurt Meyberg and Peter Vachenauer, *Höhere mathematik 1*, Springer, 2003.

[67] ———, *Höhere mathematik 2*, 4 ed., Springer, 2003.

[68] J. D. Murray, *Mathematical biology i: An introduction*, Springer, 1993.

[69] D. Subbaram Naidu, *Singular perturbation methodology in control systems*, Institution of Engineering and Technology, 1988.

[70] ———, *Singular perturbations and time scales in control theory and applications: an overview*, Dynamics of Continuous, Discrete and Impulsive Systems, Series B: Applications and Algorithms **9** (2002), 233–278.

[71] Jorge Nocedal and Stephen J. Wright, *Numerical optimization*, second ed., Springer Series in Operations Research and Financial Engineering, Springer, New York, 2006.

[72] R. O'Malley, *Singular perturbations and optimal control*, Mathematical Control Theory (W. Coppel, ed.), Lecture Notes in Mathematics, vol. 680, Springer Berlin / Heidelberg, 1978, 10.1007/BFb0065317, pp. 170–218.

[73] Rodrigo B. Platte, *How fast do radial basis function interpolants of analytic functions converge?*, IMA Journal of Numerical Analysis **31** (2011), no. 4, 1578–1597.

[74] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchen, *The mathematical theory of optimal processes*, 1. ed., L. S. Pontryagin Selected Works, vol. 4, Gordon and Breach Science Publishers, 1986.

[75] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical recipes in c – the art of scientific computing*, second ed., Cambridge University Press, Cambridge, 1997.

[76] Volkmar Reinhardt, *On the application of trajectory-based optimization for nonlinear kinetic model reduction*, Ph.D. thesis, University of Heidelberg, Heidelberg, Germany, 2008.

[77] Volkmar Reinhardt, Miriam Winckler, and Dirk Lebiedz, *Approximation of slow attracting manifolds in chemical kinetics by trajectory-based optimization approaches*, Journal of Physical Chemistry A **112** (2008), no. 8, 1712–1718.

[78] Shmuel Rippa, *An algorithm for selecting a good value for the parameter c in radial basis function interpolation*, Advances in Computational Mathematics **11** (1999), 193–210, 10.1023/A:1018975909870.

[79] Kunimochi Sakamoto, *Invariant manifolds in singular perturbation problems for ordinary differential equations*, Proceedings of the Royal Society of Edinburgh: Section A Mathematics **116** (1990), no. 1-2, 45 – 78.

[80] Shankar Sastry, *Nonlinear systems, analysis, stability and control*, Interdisciplinary Applied Mathematics, vol. 10, Springer, 1999.

[81] Jochen Siehr, *Numerical optimization methods within a continuation strategy for the reduction of chemical combustion models*, Ph.D. thesis, Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany, 2012.

[82] Sandeep Singh, Joseph M. Powers, and Samuel Paolucci, *On slow manifolds of chemically reactive systems*, Journal of Chemical Physics **117** (2002), no. 4, 1482–1496.

[83] Dominik Skanda, *Robust optimal experimental design for model discrimination of kinetic ode systems*, Ph.D. thesis, University of Freiburg, Freiburg im Breisgau, Germany, 2012.

[84] Eduardo D. Sontag, *Mathematical control theory: Deterministic finite dimensional systems*, second edition ed., Springer, New York, 1998.

[85] Josef Stoer and Roland Bulirsch, *Numerische mathematik*, fifth ed., vol. 2, Springer, Berlin, 2005.

[86] S.-K. Tin, N. Kopell, and C. K. R. T. Jones, *Invariant manifolds and singularly perturbed boundary value problems*, SIAM Journal on Numerical Analysis **31** (1994), no. 6, pp. 1558–1576 (English).

[87] Ireneusz Tobor, Patrick Reuter, and Christophe Schlick, *Efficient reconstruction of large scattered geometric datasets using the partition of unity and radial basis functions*, WSCG, 2004, pp. 467–474.

[88] A. B Vasil'eva and M. G. Dmitriev, *Singular perturbations in optimal control problems*, Journal of Mathematical Sciences **34** (1986), 1579–1629.

[89] Andreas Wächter and Lorenz T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Mathematical Programming **106** (2006), no. 1, 25–57.

[90] Joerg Walter, Mathias Koch, Gunter Winkler, and David Bellot, `http://www.boost.org/doc/libs/1_48_0/libs/numeric/ublas/doc/index.htm`.

[91] Holger Wendland, *Scattered data approximation*, Cambridge University Press, 2005.

[92] Antonios Zagaris, C. William Gear, Tasso Joost Kaper, and Yannis G. Kevrekidis, *Analysis of the accuracy and convergence of equation-free projection to a slow manifold*, ESAIM: Mathematical Modelling and Numerical Analysis **43** (2009), no. 4, 757–784.

Curriculum Vitae

Personal Data

Name	Marcel Rehberg
Date and Place of Birth	1983–03–28 in Grevesmühlen.

Professional Experience

05/2012 – 12/2012	Research Assistant, Ulm University, Institute for Numerical Mathematics, Group *Scientific Computing, Modeling and Simulation*, Project: Helmholz Heidelberg *SBCancer*: Modeling of biological systems.
01/2009 – 04/2012	Research Assistant, Freiburg University, Center for Biological Systems Analysis, Group *Scientific Computing, Modeling and Simulation*, Project FRISYS (Freiburg Initiative for Systems Biology) and Helmholz Heidelberg *SBCancer*: Modeling of biological systems.

Education

01/2009 – 5/2013	Phd thesis mathematics, Freiburg University (01/2009 – 05/2012), Ulm University (since 05/2012), Topic: *An approach to model reduction for optimal control of multiple time scale ODE*, Advisor: Prof. Dr. Dirk Lebiedz.
08/2006 – 11/2008	Master Sc., Lübeck University, Field of study: Computational Life Science, Masters thesis: *Mathematische Modelle für Lieferketten mit zugeordnetem Speicher* (Mathematical Models for supply chains with attached storage). Advisor: Prof. Dr. Dirk Langemann.
08/2006 – 12/2006	Semester Abroad, University of New Mexico, Albuquerque, USA.
10/2003 – 07/2006	Bachelor Sc., Lübeck University, Field of study: Computational Life Science, Bachelors thesis: *Vorstellung und Analyse eines Modells zum vesikulären Transport in der Zelle* (Presentation and Analysis of a Vesicular Transport Model). Advisor: Prof. Dr. Dirk Langemann.
1993 – 2002	Abitur (equivalent to A level), Gymnasium Lübz, Specialized in: Biology, Mathematics.

Acknowledgment

First and foremost I want to thank my advisor Prof. Dr. Dirk Lebiedz, not only for providing the inspiration for the topic of this work but also for creating a professional work environment. His advice was always welcome and a great deal of the concepts presented in this thesis can be traced back to his work and ideas.

Next, I would like to thank Prof. Dr. Knut Graichen for his effort of acting as the second reviewer for the thesis.

Also, I am indebted to the (former) members of the group "Scientific Computing, Modeling and Simulation": To Markus Esenwein for his excellent optimal control tool `DOT`, to Marc Fein for his help with the shape parameter optimization algorithm, to Dominik Skanda for his BDF integrator and to Jonas Unger for fruitful discussions on the topic of model reduction. A special thanks goes to Jochen Siehr for providing the `MoRe` toolkit, proofreading the thesis and answering many questions regarding his software and SIMs in general.

A thank you is in order for the members and staff of Freiburg University and the Center for Bioloical Systems Analysis for their professional handling of all the big and small administration issues. This extends also to the people at the Institute for Numerical Mathematics of Ulm University.

I would like to take the opportunity to thank my family. First of all my parents, their patient and enduring support throughout my life and educational career this work would not have been possible. They were always encouraging and their interest and curiosity were a great source of motivation while writing this thesis. Gratitude is also owed to my brother Mirko and my sister Nadine for their moral up hold.

Finally, I would like to thank Ilka Hellbrück. She provided that extra boost of motivation that was needed to get the job done.