Peer Methods in Optimal Control

Vom Fachbereich Mathematik der Technischen Universität Darmstadt
zur Erlangung des Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Dissertation von Dipl.-Math. Dirk Schröder aus Dieburg
Tag der Einreichung: 05.02.2016, Tag der Prüfung: 28.04.2016
Darmstadt 2016 — D 17

1. Gutachten: Prof. Dr. Jens Lang
2. Gutachten: Prof. Dr. Stefan Ulbrich

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Mathematik
AG Numerik und
Wissenschaftliches Rechnen

Peer Methods in Optimal Control

Genehmigte Dissertation von Dipl.-Math. Dirk Schröder aus Dieburg

1. Gutachten: Prof. Dr. Jens Lang
2. Gutachten: Prof. Dr. Stefan Ulbrich

Tag der Einreichung: 05.02.2016
Tag der Prüfung: 28.04.2016

Darmstadt 2016 — D 17

Für Helene

Zusammenfassung

In dieser Arbeit werden implizite und linear implizite Peer-Methoden im Kontext der Optimierung mit gewöhnlichen oder partiellen Differentialgleichungen als Nebenbedingungen untersucht.

In vielen Anwendungsproblemen, wie dem Kühlen von Glas, einer wandernden Flammenfront in einem gekühlten Kanal, oder dem Härten von Stahl, können die zugrunde liegenden physikalischen Prozesse durch gewöhnliche oder partielle Differentialgleichungen modelliert werden. Der Wunsch diese Vorgänge zu optimieren, führt zum Feld der optimalen Steuerung mit Differentialgleichungen.

In einem Optimierungsalgorithmus müssen die Nebenbedingungen, also in unserem Fall ein System von gewöhnlichen oder partiellen Differentialgleichungen, mehrmals ausgewertet werden. Daher ist eine effiziente Diskretisierung der Differentialgleichungen sehr wichtig. Für gewöhnliche Differentialgleichungen und zur Zeitdiskretisierung parabolischer partieller Differentialgleichungen werden gerne Runge-Kutta- und Rosenbrock-Methoden benutzt. Allerdings zeigen diese Methoden bei sehr steifen Problemen und bei der Diskretisierung parabolischer Probleme oft nicht die zu erwartende, klassische Konvergenzordnung. Dieses Phänomen wird Ordnungsreduktion genannt.

Eine vielversprechende Alternative sind Peer-Methoden. Wie Einschrittverfahren berechnen diese Methoden mehrere Näherungslösungen in einem Zeitschritt und wie Mehrschrittverfahren nutzen sie hierfür die Näherungslösungen des letzten Zeitschritts. Es wurde bewiesen, dass Peer-Methoden auch für steife Probleme die volle Konvergenzordnung zeigen. Eine Einführung zu Peer-Methoden findet sich in Kapitel 3.

Zwei populäre Ansätze zur Lösung von Optimalsteuerproblemen sind der *first-discretize-then-optimize* Ansatz und der *first-optimize-then-discretize* Ansatz. Während beim *first-discretize-then-optimize* Ansatz erst die Zielfunktion und die Nebenbedingungen diskretisiert und dann das sich ergebende endlich-dimensionale Optimierungsproblem gelöst wird, werden beim *first-optimize-then-optimize* Ansatz die unendlich-dimensionalen Optimalitätsbedingungen diskretisiert. In Kapitel 4 beschäftigen wir uns mit der Vertauschbarkeit dieser beiden Ansätze bei der Nutzung von Peer-Methoden. Wir zeigen, dass die beiden Ansätze zu sehr unterschiedlichen Resultaten führen und schließen daraus, dass Peer-Methoden nicht für den *first-discretize-then-optimize* Ansatz geeignet sind.

Daher konzentrieren wir uns im Folgenden auf den *first-optimize-then-discretize* Ansatz und nutzen Peer-Methoden in einem Multilevel Optimierungsalgorithmus. Hierfür leiten wir eine voll adaptive, also adaptiv sowohl in der Zeit als auch im Ort, Diskretisierung für parabolische partielle Differentialgleichungen in Kapitel 5 her. Wir nutzen den Rothe Ansatz und diskretisieren erst in der Zeit mit einer linear impliziten Peer-Methode. Dies führt auf mehrere lineare elliptische Probleme, die wir dann mit einer

Multilevel Finiten Element-Methode diskretisieren. Der Fehlerschätzer für den Fehler im Ort basiert hierbei auf hierarchischen Basen. Den Fehler in der Zeit schätzen wir ab, indem wir die berechnete Lösung mit einer Lösung niedrigerer Ordnung vergleichen. Wir betrachten die Effizienz des räumlichen Fehlerschätzers sowohl analytisch als auch numerisch. Schließlich vergleichen wir die Laufzeit von Peer-Methoden mit der von Rosenbrock-Methoden für drei partielle Beispiele in 2D. Peer-Methoden zeigen sich hierbei wettbewerbsfähig zu Rosenbrock-Methoden.

Diese voll adaptive Diskretisierung nutzen wir dann für die Multilevel Optimierung in Kapitel 6. Zuerst führen wir den Optimierungsalgorithmus ein und legen dabei besonderes Augenmerk auf die Rolle der Zeitintegration. Schließlich präsentieren wir Ergebnisse für drei Optimalsteuerprobleme mit partiellen Differentialgleichungen, wobei sich Peer-Methoden wieder wettbewerbsfähig zu Rosenbrock-Methoden zeigen.

Abstract

In this thesis we analyze implicit and linearly implicit peer methods in the context of optimization problems with ordinary or partial differential equations as constraints.

In many practical applications, like the cooling of glass, the propagation of a flame front in a cooled channel or the hardening of steel, the underlying physical process can be modeled by ordinary differential equations (ODE) or partial differential equations (PDE). The wish to optimize these processes leads to the field of ODE- and PDE-constrained optimization.

The constraints, in this case an ODE or PDE system, have to be evaluated several times in an optimization algorithm. Therefore it is very important to use efficient discretization methods for the arising differential equations. Runge-Kutta and Rosenbrock methods are a popular choice for ODEs and the time discretization of parabolic PDEs. However they suffer from order reduction when applied to stiff problems.

A promising alternative are peer methods. These methods construct several approximations to the solution in one time step like one-step methods and use the approximations of the last time step like multistep methods. Peer methods are proven to show no order reduction when applied to stiff problems. More details on peer methods are presented in Chapter 3. In this thesis we analyze peer methods within the optimal control with differential equations.

There are two popular approaches when solving optimal control problems. The first approach is called *first-discretize-then-optimize*, while the other is the *first-optimize-then-discretize* approach. In Chapter 4 we analyze the interchangeability of these two approaches when using peer methods. We find that the two approaches give quite different results for peer methods and especially conclude, that peer methods are not well suited for the first-discretize-then-optimize approach.

Therefore, we concentrate then on the first-optimize-then-discretize approach and especially want to employ peer methods within a multilevel optimization approach. To this end we derive a fully adaptive, that is adaptive in time and space, discretization for parabolic PDEs in Chapter 5. We follow the Rothe approach and discretize first in time by a linearly implicit peer method leading to several linear elliptic problems. These are then discretized by multilevel linear finite elements. We derive a spatial error estimator based on hierarchical bases. The time error is estimated by comparing the computed solution with a solution of lower order. We look at the efficiency of the spatial error estimator both analytically and numerically. Finally we compare the performance of peer methods to that of Rosenbrock methods for three PDE test examples in 2D. We see that peer methods are competitive to Rosenbrock methods.

This fully adaptive scheme is then used within a multilevel optimization in Chapter 6. We first introduce the optimization algorithm and especially look at the points where

the time integration plays a role. Finally, we present results for three PDE constrained control problems. Again the peer methods are competitive to Rosenbrock methods.

Acknowledgement

First of all, I thank Professor Dr. Jens Lang for his great advice and support during the development of this thesis.

I also thank the co-referee Professor Dr. Stefan Ulbrich and the members of the examining board Professor Dr. Karsten Grosse-Brauckmann, Professor Dr. Martin Kiehl and Professor Dr. Reinhard Farwig.

Especially I say thank you to Dr. Alf Gerisch for various interesting discussions, useful comments and for proofreading this thesis. Furthermore I thank Dr. Sebastian Ullmann for being always helpful and Dr. Pia Domschke for being a great office neighbour.

Special thanks and love go to my wife Helene for her neverending support.

Contents

1. Introduction

In this thesis we analyze implicit and linearly implicit peer methods within the optimization with ordinary or partial differential equations as constraints.

Ordinary or partial differential equations play an important role in many practical applications, as they can be used to describe the underlying physical process. Examples which are presented in this thesis are the cooling of glass, the propagation of flames in a cooled channel and the hardening of steel. Typically an analytical solution is not available and one has to solve the differential equations numerically in order to simulate the whole process.

Often we are not satisfied by just simulating, but we want to control the whole process in an optimal way. For example, we want to cool a glass workpiece as quickly as possible, but without causing cracks in the workpiece and we want to use as few energy as possible. This leads to the field of optimal control with differential equations.

In this thesis we especially look at time dependent problems, that is we consider problems of the form

$$\min_{u \in U, y \in Y} J(y, u) \tag{1.1}$$

$$\text{s.t. } e(y, u) = 0. \tag{1.2}$$

Here J denotes the objective function we want to minimize. u is the control parameter and y the state variable. The constraint (1.2) stands either for an ordinary differential equation or a parabolic partial differential equation. This problem will be discussed in more detail in Chapter 2.

To solve the optimal control problem (1.1) - (1.2) there are two popular approaches. In the *first-discretize-then-optimize* approach we start by discretizing the differential equation (1.2). This leads then to a finite dimensional nonlinear optimization problem

$$\min_{u_h \in U_h, y_h \in Y_h} J_h(y_h, u_h) \tag{1.3}$$

$$\text{s.t. } e_h(y_h, u_h) = 0. \tag{1.4}$$

For this, typically large scale, finite dimensional problem there are a lot of efficient numerical solvers available. This approach has two big advantages when using derivative based optimization methods. First the discrete gradient can often be computed by automatic differentiation within a sensitivity or adjoint approach. Second this discrete gradient is consistent with the discrete control problem. This ensures convergence of the optimization method. A drawback of this approach is that within an adjoint approach the discretization of the adjoint is fixed by the discretization of the state. This can lead

to an inconsistent approximation of the infinite dimensional adjoint system. Hence the discrete gradient might not be consistent with the original infinite dimensional optimization problem.

A first possibility to overcome this drawback is to use discretization methods where the discrete adjoint is also a consistent approximation of the infinite dimensional adjoint equation. This property is called adjoint consistency. In general, adjoint consistency can be directly achieved by applying Galerkin-type time discretizations as shown in Becker et al. [3] and Meidner and Vexler [36] for parabolic optimization problems. However, since these methods can be interpreted as special implicit Runge-Kutta schemes of Radau-type, they also join the main drawbacks with them: (i) a fully coupled, nonlinear system of stage values which challenges the storage requirements and the necessary iterative solution process and (ii) order reduction phenomena that usually occur when the system of ODEs arises from a semi-discretization of PDEs with boundary conditions. It is therefore reasonable to consider also alternative time integration methods such as diagonally implicit Runge-Kutta methods, which offer special structures to simplify the linear algebra. Additionally, multistep or peer methods also avoid order reduction to really benefit from higher order schemes.

The discrete adjoints of Runge-Kutta methods are themselves Runge-Kutta methods again. Hager [21] found that to achieve order three or higher of the discrete adjoint, additional order conditions have to be satisfied. First- or second-order Runge-Kutta schemes retain the same order. The consistency analysis can be done in the context of partitioned symplectic Runge-Kutta methods. Bonnans and Laurent-Varin [6] give order conditions up to order seven. Murua [37] already gives the same conditions for partitioned symplectic Runge-Kutta schemes. Walther [58] analyses reverse mode automatic differentiation applied to explicit Runge-Kutta methods and concludes that the order of the discretization is preserved under only mild additional conditions. The difference to the results of Hager [21] come from a different treatment of the control. While in [21] the control depends on the state and the adjoint, Walther [58] considers a fixed control. For the case of a control only acting in the initial conditions, Sandu [46] gives consistency properties of discrete adjoint Runge-Kutta methods. Lang and Verwer [32] show that W-methods also have to fulfill additional conditions, such that their discrete adjoints are consistent of order three or higher. They construct an L-stable W-method of third order in the state and adjoint variables. The special case of Crank-Nicolson schemes in the context of linear evolution equations were examined by Apel and Flaig [1]. They show that the Crank-Nicolson scheme is self-adjoint.

For multistep methods the situation is not so favorable. Sandu [47] shows that in the interior time domain discrete adjoint linear multistep methods have the same order of consistency as the corresponding one leg method for constant step sizes. However the adjoint initialization steps are usually inconsistent approximations. For adaptive step sizes even the consistency in the interior time domain is lost.

In Chapter 4 we will look at the adjoint consistency of implicit peer methods [2, 49]. These are two-step multistage methods, where all stage values share the same order of approximation. Beck et al. [2] show that implicit peer methods perform very well for

stiff problems and semi-discretizations of partial differential equations. Due to their high stage order they do not suffer from order reduction and work especially well for high accuracy requirements. Our main motivation is to clarify the potential of implicit peer methods to overcome the deficiencies of linear multistep methods when applied to optimal control problems. We construct peer methods with high adjoint consistency in the interior of the integration interval and show that the well-known backward differentiation formulas are optimal with respect to the achievable order. We will clearly identify that inappropriate adjoint initialization still remains a crucial issue for implicit peer methods, which restrict the overall consistency order to one. A fact that is also valid for linear multistep methods. Given the low consistency order of the discrete adjoints and therefore of the whole discretization, we have to conclude that implicit peer methods are not suitable for a first-discretize-then-optimize approach. The content of Chapter 4 has been published in [51].

Another possibility to avoid an inconsistent approximation of the infinite dimensional adjoint equation is the *first-optimize-then-discretize* approach. Here we begin by deriving optimality conditions on the infinite dimensional level. This allows us to discretize the state and adjoint equations separately by different, appropriate numerical solvers. While the discretized adjoint is now consistent with the infinite dimensional adjoint equation, the resulting gradient must not be consistent with the discrete optimal control problem. This inconsistency has to be controlled in an optimization method.

In [60, 61] an adaptive multilevel SQP method is presented. Here the approximation errors in the state and adjoint system are controlled with respect to a so called criticality measure. The criticality measure quantifies the optimality of the computed control iterate. This enables the algorithm to use coarse meshes if the optimization is still far away from an optimal solution. If the iteration comes near the optimal solution, that is the criticality measure became small, the approximation errors in the state and adjoint system are also required to be small. The algorithm will be explained in Section 6.1.

A critical point within the adaptive multilevel optimization is the efficient control of the state and adjoint errors. Therefore we are interested in an adaptive solution of the state and adjoint equations. In this work we take a closer look on the optimal control with parabolic partial differential equations, which read in their abstract form

$$\partial_t y = F(t, y), \quad 0 < t \leq T, \quad y(0) = y_0. \tag{1.5}$$

Popular discretization strategies for this problem are the method of lines and the Rothe method. In the method of lines we start by discretizing the right-hand side term $F(t, y)$ by for example a finite element method or finite differences. This leads to a large structured system of ordinary differential equations. This ODE system can then be solved by an appropriate ODE solver. Popular choices are for example the BDF-method based solver DASSL [42] or linearly implicit integrators. For fixed time steps and fixed spatial grids the Crank-Nicolson-scheme [15] is a common choice. While temporal adaptivity can be implemented very efficiently within the method of lines, spatial adaptivity is more difficult [17].

A more natural approach to combine adaptive time integration with spatial adaptive

methods is the Rothe-approach. Here we take the parabolic PDE as a Banach space valued ODE and first discretize it in time by a suitable integration method. In [28] a multilevel finite element solver was combined with an adaptive Rosenbrock solver to efficiently approximate nonlinear parabolic problems. The method was implemented in the software package *KARDOS* [19]. In [20] linearly implicit peer methods were introduced to Kardos and showed good results compared to Rosenbrock methods.

However the presented method in [20] is only adaptive in time and fixed spatial grids are used for the whole integration. There are problems like the propagation of a flame front which are solved more efficiently using also adaptive spatial grids. Therefore we combine a multilevel finite element method with a linearly implicit peer method in Chapter 5. The presented error estimator is proven to be efficient and reliable up to some perturbations. Furthermore some numerical tests show that the performance of the presented method is comparable to that of Rosenbrock methods.

In Chapter 6, we compare the derived, fully adaptive scheme to the combination of Rosenbrock methods with multilevel finite elements within the adaptive multilevel optimization algorithm presented in [60, 61]. The algorithm was already sucessfully applied to the optimal control of radiative heat transfer in two spatial dimensions in [12–14]. State constraints were added to the multivel optimization algorithm in [7, 11]. In [7] the algorithm was applied to the optimal control of radiative heat transfer in three spatial dimensions and the optimal control problem of heating a steel rack.

In this thesis, we consider three different test settings in two spatial dimensions. The first test problem is the optimal cooling of glass, where the occurring radiation is modelled by an $SP1$ approximation. This leads to a coupled system of a parabolic and an elliptic PDE with nonlinear source and boundary terms. The control acts on the boundary. The second test problem is the optimal control of a flame moving in a channel with the aim of burning the material only inside a part of the channel. The propagation of the flame is modelled by a system of linear parabolic equations with a nonlinear source term. The control enters the system again via the boundary. The last test problem is the optimal heating of a steel rack by a current which is induced on the boundary of the rack. Here the PDE system couples an instationary heat equation with a quasilinear potential equation. This system is known as the thermistor problem. Again we consider a boundary control.

Finally, in Chapter 7 we summarize the conclusions of this thesis.

2. Optimal control of differential equations

Following [22] we consider an optimal control problem of the following form:

$$\begin{aligned} &\min_{u\in U, y\in Y} J(y,u), \\ &\quad \text{s.t. } e(y,u) = 0. \end{aligned} \tag{2.1}$$

The objective function $J : Y \times U \mapsto \mathbb{R}$ models the quantities we want to minimize. Here we already used the typical differentiation between the state y from the state space Y and the control u taken from a control space U. The system of differential equations, either ordinary or partial, is hidden in the operator $e : Y \times U \mapsto Z$. The spaces Y, U and Z are considered to be Banach spaces.

In this work there are usually also constraints for the control function u. Hence we do not look for an optimal control in the whole control space U but rather in a smaller set of admissible functions $U_{ad} \subset U$. This set is assumed to be a closed convex set. In this thesis restrictions from above and below are used. That is we have the so-called box constraints:

$$U_{ad} := \{u \in U : \, u_{low} \leq u \leq u_{up}\}, \tag{2.2}$$

with u_{low}, $u_{up} \in U$ and $u_{low} \leq u_{up}$. Here, the relations are element- and point-wise.

The objective function J and the operator e are assumed to be continuously Fréchet-differentiable. Furthermore we assume that the state equation

$$e(y,u) = 0 \tag{2.3}$$

admits a unique solution $y(u)$ for every $u \in U$. Therefore, we have a solution operator $u \in U \mapsto y(u) \in Y$. Inserting this solution operator into the optimal control problem (2.1) leads to the reduced problem

$$\min_{u\in U_{ad}} \hat{J}(u) \quad \text{with } \hat{J}(u) := J(y(u),u). \tag{2.4}$$

For optimality conditions and optimization algorithms it is important to compute derivatives of the reduced objective function $\hat{J}$. We define the adjoint state $p \in Z^*$ fulfilling the adjoint equation

$$\partial_y J(y(u),u) + \partial_y e^*(y(u),u)p = 0. \tag{2.5}$$

Then the first derivative of $\hat{J}$ is given by

$$\hat{J}'(u) = \partial_u J(y(u),u) + \partial_u e^*(y(u),u)p. \tag{2.6}$$

Using the adjoint representation of the derivative we can give necessary first order optimality conditions for the reduced, control constrained optimal control problem (2.4). Let $(\bar{y}, \bar{u})$ be an optimal solution of (2.4). Then there exists an adjoint state $\bar{p} \in Z^*$ such that [22]

$$e(\bar{y}, \bar{u}) = 0, \qquad \text{(state equation)} \tag{2.7}$$

$$\partial_y e^*(\bar{y}, \bar{u})\bar{p} = -\partial_y J(\bar{y}, \bar{u}), \qquad \text{(adjoint equation)} \tag{2.8}$$

$$\bar{u} \in U_{ad}, \quad \langle \partial_u J(\bar{y}, \bar{u}) + \partial_u e^*(\bar{y}, \bar{u})\bar{p}, u - \bar{u} \rangle_{U^*,U} \geq 0 \quad \forall u \in U_{ad}. \tag{2.9}$$

Using the gradient (2.6) we can solve the optimization problem (2.1) with the Gradient method. Together with a step size search, like, for example, the Armijo rule, the Gradient method converges even globally [22]. The major drawback of the Gradient method is its slow, linear convergence. Faster convergence is obtained by using a Newton-type method, i.e. taking the curvature of the problem into account.

Assuming that J and e are twice continuously differentiable, we can compute the second derivative of $\hat{J}$ by differentiating $\hat{J}'(u)$ in a direction $s_u \in U$. First we introduce the linearized state s_y solving the linearized state equation

$$e_y(y(u), u)s_y = -s_u \tag{2.10}$$

and the second adjoint state w solving the second adjoint equation

$$\begin{aligned} e_y^*(y(u), u)w = &-\partial_{yy} J(y(u), u)s_y - \partial_{yy} e^*(y(u), u)ps_y \\ &- \partial_{yu} J(y, u)s_u - \partial_{yu} e^*(y(u), u)ps_u. \end{aligned} \tag{2.11}$$

Then we can compute the derivative of $\hat{J}'(u)$ in direction s_u by

$$\begin{aligned} \hat{J}''(u)s_u = &\partial_{uu} J(y(u), u)s_u + \partial_{uu} e^*(y(u), u)ps_u + \partial_u e^*(y(u), u)w \\ &+ \partial_{uy} J(y(u), u)s_y + \partial_{uy} e^*(y(u), u)ps_y. \end{aligned} \tag{2.12}$$

For the optimization problems in this thesis we use the Newton-type SQP method presented in [60, 61], which we introduce in Section 6.1. The SQP method converges at least locally superlinear and is therefore more attractive than a Gradient method.

3. Peer methods

3.1. Implicit peer methods

Initial value problems of the form

$$y'(t) = F(t, y(t)), \quad t \in [t_0, t_f], \quad y(t_0) = y_0, \tag{3.1}$$

with $F : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ are usually solved by one-step methods like Runge-Kutta or Rosenbrock methods or multistep methods like BDF methods. While one-step methods use only the last solution value to compute the approximation at the next time point, multistep methods use several old values. However unlike one-step method multistep methods do not construct intermediate stage values. A combination of the two approaches forms the class of general linear methods [8]. In [48] the class of linearly implicit parallel peer methods was introduced. They were called peer methods because there is no distinction between intermediate stage values and approximations at the time points. All stage values share the same accuracy and stability properties. This contrasts especially Runge-Kutta methods where the stage values only have low order of accuracy. The method was tailored for an independent solution of the stage values in parallel. It was then generalized to implicit parallel methods [50] and sequential linearly implicit methods [20, 43]. We start by giving the definition of implicit peer methods and look at their consistency and stability properties.

Let τ_n be the step size at time step $n \geq 1$. We set $t_n = t_{n-1,s}$. An s-stage implicit two-step peer method computes approximations $Y_{ni} \in \mathbb{R}^m$, $i = 1, \ldots, s$ to the exact solution of (3.1) at time points $t = t_{ni} = t_n + c_i\tau_n$, i.e., $Y_{ni} \approx y(t_{ni})$. Furthermore we introduce the step size ratio $\sigma_n = \frac{\tau_n}{\tau_{n-1}}$. The implicit peer method is then given by

$$\begin{aligned} i = 1, \ldots, s: \quad Y_{ni} &= \sum_{j=1}^{i} a_{ij}\tau_n F(t_{nj}, Y_{nj}) + \sum_{j=1}^{s} u_{ij}(\sigma_n) Y_{n-1,j}, \\ t_{ni} &= t_n + c_i\tau_n, \quad \sigma_n = \frac{\tau_n}{\tau_{n-1}}. \end{aligned} \tag{3.2}$$

Let $c \in \mathbb{R}^s$ denote the vector of pairwise distinct abscissae of the method. We set $c_s = 1$. There are no further restrictions on the values of c_i.

Different choices of the c_i have been considered in the literature. In [20] and [48] the values are chosen as the stretched Chebychev nodes

$$c_i := -\frac{\cos((i - \frac{1}{2})\frac{\pi}{s})}{\cos(\frac{\pi}{2s})}, \quad i = 1, 2, \ldots, s,$$

lying in the interval $[-1, 1]$. The reason of this choice is the small condition number of the corresponding Vandermonde matrix, which avoids inaccuracies in the computation of the method coefficients. In [2] the abscissae are restricted to the interval $(0, 1]$, which corresponds to the range used in classical methods of advancing forward in time and makes a dense output available.

In this thesis we will use the coefficients of singly implicit peer methods with positive nodes c_i as presented in [2]. We use a second, third and fourth order method which we will refer to as *peer*3*pos*, *peer*4*pos*, and *peer*5*pos*, respectively. Strictly positive nodes showed to be advantageous in the context of optimal control. Here, interpolation of the solution approximations is needed. Hence the possibility of a dense output is an advantage over methods which use also negative nodes. Overall the performance of peer methods with strictly positive nodes is very similar to that of the methods presented in [20] and [48]. Therefore we decided to only present numerical results for the singly implicit peer methods with positive nodes.

For a more compact notation we introduce stacked vectors

$$Y_n = \begin{pmatrix} Y_{n1} \\ Y_{n2} \\ \vdots \\ Y_{ns} \end{pmatrix}, \quad F(t_n, Y_n) = \begin{pmatrix} F(t_{n1}, Y_{n1}) \\ F(t_{n2}, Y_{n2}) \\ \vdots \\ F(t_{ns}, Y_{ns}) \end{pmatrix}.$$

Then we can rewrite (3.2) to

$$Y_n = (U(\sigma) \otimes I_m) Y_{n-1} + \tau_n (A \otimes I_m) F(t_n, Y_n). \tag{3.3}$$

Here $\otimes$ denotes the Kronecker product and I_m is the identity matrix of dimension m. The coefficients of the method are collected in the matrices $U(\sigma) = (u_{ij}(\sigma))_{i,j=1}^{s} \in \mathbb{R}^{s \times s}$ and the lower triangular matrix $A = (a_{ij})_{i,j=1}^{s} \in \mathbb{R}^{s \times s}$. Note that the matrix U depends on the step size ratio σ_n. Therefore, in an adaptive computation we have to recompute U at every time step. We will see later that this computation can be done efficiently. If $a_{ii} = \gamma$ for all $i = 1, \ldots, s$ we call the method singly-implicit, else it is called multi-implicit.

3.1.1. Starting procedure

The following strategy to compute the starting values of the implicit peer method was proposed in [20]. To compute the first step of the implicit peer method, i.e., $n = 1$, the s approximations $Y_{0i} \approx y(t_1 + (c_i - 1)\tau_0)$, the time point t_1 and the two step sizes τ_0 and τ_1 need to be known. We will compute Y_{0i}, $i = 1, \ldots, s$, by an one-step method with continuous output starting at t_0 and given the initial data y_0 of the problem. Let τ_{osm} denote the step size used by the one-step method. Then a numerical solution $\tilde{y}$ in the interval $[t_0, t_0 + \tau_{osm}]$ is available.

The smallest and largest time point where a starting solution is needed should equal t_0 and $t_0 + \tau_{osm}$. With c^- and c^+ denoting the minimum and maximum values of the abscissae c_i we get the equations

$$t_1 + (c^- - 1)\tau_0 = t_0 \text{ and } t_1 + (c^+ - 1)\tau_0 = t_0 + \tau_{osm}.$$

Solving for t_1 and τ_0 gives

$$t_1 = t_0 + \frac{1-c^-}{c^+-c^-}\tau_{osm} \text{ and } \tau_0 = \frac{1}{c^+-c^-}\tau_{osm}.$$

The starting values are hence given by

$$Y_{0i} := \tilde{y}(t_1 + (c_i - 1)\tau_0) = \tilde{y}\left(t_0 + \frac{c_i - c^-}{c^+ - c^-}\tau_{osm}\right), \quad i = 1, 2, .., s.$$

Note that the Y_{0i} corresponding to the node c^- is equal to the prescribed initial data y_0.

Since a time step ratio $\sigma = 1$ is beneficial for implicit peer methods, we require $\sigma_1 = 1$, i.e., $\tau_1 = \tau_0$. Finally the choice of τ_{osm} remains. As higher-order methods require sufficiently accurate starting values to show their expected performance, the starting step size τ_{osm} should be sufficiently small.

3.1.2. Consistency

One big difference between peer methods and Runge-Kutta methods is the full order of all stages. Hence we require that

$$Y_{ni} = y(t_{ni}) + \mathcal{O}(\tau_n^p)$$

where p denotes the order of the peer method. To derive the order conditions we replace Y_{ni} and $Y_{n-1,i}$ in (3.2) by the values of the true solution $y(t)$ and we use $F(t_{ni}, Y_{ni}) = y'(t_{ni})$. Taylor series expansion gives

$$y(t_n + c_i\tau_n) = \sum_{l=0}^{p} \frac{(c_i\tau_n)^l}{l!} y^{(l)}(t_n) + \mathcal{O}(\tau_n^{p+1}). \tag{3.4}$$

Like in [2] we use the node vector $c = (c_1, \ldots, c_s)^T$, the vector consisting only of ones $\mathbb{1} = (1, \ldots, 1)^T$ and the operator $z = \tau_n \frac{d}{dt}$ for a compact notation. With this notation we can define the consistency order of implicit peer methods by the following definition.

Definition 3.1.1. An implicit peer method has consistency order p, if the parameter matrices A, U and the node vector c fulfill the following set of equations for $z \to 0$:

$$\exp(cz) = U(\sigma_n)\ \exp((c - \mathbb{1})z) + Az\ \exp(cz) + \mathcal{O}(z^{p+1}). \tag{3.5}$$

The exponential of the vector is defined component wise.

By collecting the coefficients of $z^l/l!$ in columns we can rewrite (3.5) in matrix form. This gives

$$V_0^{(p)}\Sigma^{(p)} = U(\sigma_n)V_1^{(p)} + AV_0^{(p)}D^p(\Delta_0^{(p)})^T\Sigma^{(p)}, \tag{3.6}$$

where the matrices are given by

$$\begin{aligned}
\left(V_0^{(p)}\right)_{i,j} &= c_i^{j-1}, && 1 \le i \le s, 1 \le j \le p+1,\\
\left(V_1^{(p)}\right)_{i,j} &= (c_i - 1)^{j-1}, && 1 \le i \le s, 1 \le j \le p+1,\\
\left(\Delta_0^{(p)}\right)_{i,j} &= \delta_{i,j+1} = \begin{cases} 1 & i = j+1, \\ 0 & i \neq j+1, \end{cases} && 1 \le i \le s, 1 \le j \le p+1,\\
\Sigma^{(p)} &= \operatorname{diag}(1, \sigma_n, \sigma_n^2, \ldots, \sigma_n^p),\\
D^{(p)} &= \operatorname{diag}(1, 2, \ldots, p+1).
\end{aligned}$$

By demanding consistency order $p = s - 1$ the matrix $U(\sigma_n)$ is uniquely determined by the matrix A, the vector c and σ_n via the following system of equations:

$$U(\sigma_n) = \left(V_0^{(p)} - AV_0^{(p)} D^p (\Delta_0^{(p)})^T\right) \Sigma^{(p)} (V_1^{(p)})^{-1}. \tag{3.7}$$

Remark 3.1.1. The matrix $V_1^{(p)}$ is in this case quadratic. Since we assume pairwise distinct nodes c_i it is also regular. □

Remark 3.1.2. Note, that in the right-hand side in (3.7) the step size ratio σ_n enters the equation only through $\Sigma^{(p)}$. So one can compute $U(1)$ by (3.7) for the given matrix A of the implicit peer method. During integration with adaptive step sizes and hence step size ratios $\sigma_n \neq 1$, the matrix $U(\sigma_n)$ can be obtained via the system

$$U(\sigma_n) = U(1) V_1^{(p)} \Sigma^{(p)} (V_1^{(p)})^{-1}. \tag{3.8}$$

Since $V_1^{(p)}$ and $(V_1^{(p)})^{-1}$ are also independent of the actual problem, finding $U(\sigma_n)$ corresponds only to some multiplications of matrices of size $s \times s$. This is usually negligible compared to the overall computational cost. □

Requiring order $p = s - 1$ leaves the coefficients in A as free parameters. There are $\frac{s^2+s}{2}$ free coefficients for multi-implicit peer methods and $\frac{s^2-s}{2} + 1$ free coefficients for singly-implicit peer methods. Since consistency is not enough for convergence, we will first derive stability conditions to find the remaining coefficients.

3.1.3. Stability

Applying the implicit peer method (3.3) to the Dahlquist test equation $y' = \lambda y$ with $\lambda \in \mathbb{C}$ yields

$$Y_n = U(\sigma_n) Y_{n-1} + \tau \lambda A Y_n.$$

Setting $z = \tau\lambda$ and solving for Y_n gives

$$Y_n = (I - zA)^{-1} U(\sigma_n) Y_{n-1} := M(z) Y_{n-1}.$$

Here, $M(z)$ is the stability matrix of the method. For $\lambda = 0$ the stability matrix is $M(0) = U(\sigma_n)$. Since the exact solution is constant, the numerical solution should at least be bounded for an arbitrary number of steps. This is called zero-stability. For implicit peer methods we have the following definition from [2]:

Definition 3.1.2. An implicit peer method (3.2) is *zero-stable*, if

$$||U(\sigma_{n+l})U(\sigma_{n+l-1})\cdots U(\sigma_{n+1})U(\sigma_n)|| \leq K$$

holds for some constant $K < \infty$ and for all n and $l \geq 0$.

Now with consistency and zero-stability as defined above we have the following theorem from [2]:

Theorem 3.1.1. *If the implicit peer method is consistent of order p and zero-stable, it is also convergent of order p.*

It was shown in [43] that peer methods with the stronger property of optimal zero stability are guaranteed to be zero-stable for arbitrary step size sequences. Optimal zero stability means, that $M(0)$ has one eigenvalue equal to 1, which is required by consistency, and all other eigenvalues are equal to zero, i.e., vanishing parasitic roots. Choosing A such that $U(\sigma_n)$ has this property for every step size ratio $\sigma_n > 0$ leaves only one parameter $y = a_{ii}$ for singly-implicit peer methods and s parameters for multi-implicit methods. [2] uses the remaining degrees of freedom to construct multi-implicit methods of order $p = s$. For singly-implicit methods the remaining degree of freedom is chosen such that the method has coefficients of small magnitude, order $p = s$ for constant time step sizes, i.e., $\sigma = 1$, and finally also gives $L(\alpha)$-stable methods with a large angle α.

An implicit peer method is called $A(\alpha)$ stable [48], if

$$\forall z \text{ with } |\arg(z) - \pi| \leq \alpha : \rho(M(z)) \leq 1.$$

For $\alpha = \frac{\pi}{2}$ we have A-stability. Especially for the application to partial differential equations the stronger property of $L(\alpha)$-stability is desirable. An implicit peer method is $L(\alpha)$-stable if it is $A(\alpha)$-stable and

$$\lim_{z \to -\infty} M(z) = 0.$$

Since the stability matrix $M(z) = (I - zA)^{-1}U(\sigma_n)$ of peer methods vanishes at infinity, that is $M(\infty) = 0$, $L(\alpha)$-stability is equivalent to $A(\alpha)$-stability for peer methods.

3.2. Linearly implicit peer methods

For nonlinear problems (3.2) is a nonlinear system for each stage Y_{ni}. Especially when solving PDEs this nonlinear system can get very large and hence difficult to solve. We restrict ourself for this section to the case of singly-implicit peer methods.

Introducing an auxiliary vector

$$w_{ni} = \sum_{j=1}^{i-1} a_{ij}\tau_n F(t_{nj}, Y_{nj}) + \sum_{j=1}^{s} u_{ij}(\sigma_n) Y_{n-1,j} \tag{3.9}$$

leads to a new formulation of (3.2) given by

$$Y_{ni} = \tau_n \gamma F(t_{ni}, Y_{ni}) + w_{ni}, \quad i = 1, \ldots, s. \tag{3.10}$$

Rearranging (3.10) to

$$\tau_n F(t_{ni}, Y_{ni}) = \frac{1}{\gamma}(Y_{ni} - w_{ni}), \quad i = 1, \ldots, s,$$

and inserting this into (3.9) we get an equivalent formulation of the peer method:

$$Y_{ni} = \tau_n \gamma F(t_{ni}, Y_{ni}) + w_{ni}, \qquad i = 1, \ldots, s, \tag{3.11}$$

$$w_{ni} = \sum_{j=1}^{i-1} \frac{1}{\gamma} a_{ij}(Y_{nj} - w_{nj}) + \sum_{j=1}^{s} u_{ij}(\sigma_n) Y_{n-1,j}. \tag{3.12}$$

Equation (3.11) is a nonlinear system of equations for Y_{ni}. Given a good predictor Y^0_{ni} one iteration of Newton's method gives usually a sufficiently accurate solution. Performing this Newton step gives the linearly implicit peer method [20]

$$(I - \tau_n \gamma J_n)(Y_{ni} - Y^0_{ni}) = \tau_n \gamma F(t_{ni}, Y^0_{ni}) + w_{ni} - Y^0_{ni}, \tag{3.13a}$$

$$Y^0_{ni} = \sum_{j=1}^{i-1} \frac{1}{\gamma} a^0_{ij}(Y_{nj} - w_{nj}) + \sum_{j=1}^{s} u^0_{ij}(\sigma_n) Y_{n-1,j}, \tag{3.13b}$$

$$w_{ni} = \sum_{j=1}^{i-1} \frac{1}{\gamma} a_{ij}(Y_{nj} - w_{nj}) + \sum_{j=1}^{s} u_{ij}(\sigma_n) Y_{n-1,j}. \tag{3.13c}$$

The matrices $A^0 = (a^0_{ij})$ and $U^0 = (u^0_{ij})$ are additional $s \times s$ real coefficient matrices for the predictor. A^0 is strictly lower triangular and U^0 may depend again on the step size ratio. The matrix $J_n \in \mathbb{R}^{m\times m}$ is an approximation to the Jacobian of F,

$$J_n \approx \frac{\partial F(t_n, Y_{n-1,s})}{\partial y}.$$

Note that J_n is constant for each time step.

Remark 3.2.1. In [20] it is remarked, that by solving the recursions (3.13b) and (3.13c) for Y^0_{ni} and w_{ni}, we get the following equivalent computation formula

$$w_{ni} = \sum_{j=1}^{i-1} \bar{a}_{ij} Y_{nj} + \sum_{j=1}^{s} \bar{u}_{ij}(\sigma_n) Y_{n-1,j}, \tag{3.14}$$

$$Y^0_{ni} = \sum_{j=1}^{i-1} \bar{a}^0_{ij} Y_{nj} + \sum_{j=1}^{s} \bar{u}^0_{ij}(\sigma_n) Y_{n-1,j}, \tag{3.15}$$

with coefficients $\bar{A} := I - \gamma A^{-1}$, $\bar{U}(\sigma_n) := \gamma A^{-1} U(\sigma_n)$, $\bar{A}^0 := I - \gamma (A^0)^{-1}$ and $\bar{U}^0(\sigma_n) := \gamma (A^0)^{-1} U^0(\sigma_n)$. In this form only the approximations $Y_{n,j}$, $Y_{n-1,j}$, $j = 1, \ldots, s$, and the value w_{ni} of the current stage need to be stored within time step n. □

If the predictor has order $p = s - 1$ and the coefficients U and A fulfill the order conditions for an implicit peer method [2] of order $p = s - 1$, then the resulting linearly implicit peer method has also order $p = s - 1$ [20]. Furthermore, since the linearly implicit peer method is equivalent to the implicit peer method for linear problems, both have the same linear stability properties [20].

In [20] the authors propose the following strategy to construct the matrices A^0 and $U^0(\sigma_n)$. They impose order $p = s - 1$ for the predictor. This gives similar to the order conditions of implicit peer methods the following set of equations:

$$V_0^{(p)} \Sigma^{(p)} = U^0(1) V_1^{(p)} + A_0 V_0^{(p)} D^p (\Delta_0^{(p)})^T \Sigma^{(p)}. \tag{3.16}$$

Then they can solve the order conditions for $U^0 = U^0(A^0)$. The coefficients of A^0 are found by minimizing the Frobenius norm of U^0, i.e., we solve $\min_{A^0} ||U^0(A^0)||_F$. For $\sigma \neq 1$ we can compute $U^0(\sigma)$ via

$$U^0(\sigma) = U^0 V_1^{(p)} \Sigma^{(p)} (V_1^{(p)})^{-1}.$$

3.3. Linearly implicit peer methods with positive nodes

In this work we will use the coefficients of the singly implicit peer methods presented in [2]. The methods are of order $p = s - 1$ for general step sizes and of order $p = s$ for constant step sizes.We denote the linearly implicit methods using the coefficients of the methods with $s = 3$, $s = 4$ and $s = 5$ by *peer*3*pos*, *peer*4*pos* and *peer*5*pos*. All methods are $L(\alpha)$ stable with angles $\alpha = 86.1°$ for *peer*3*pos*, $\alpha = 83.2°$ for *peer*4*pos* and $\alpha = 75.7°$ for *peer*5*pos*. The coefficients of the methods are given in A.1 - A.3 in the appendix.

4. The discrete adjoint of implicit peer methods

4.1. Introduction

In this chapter we will look at implicit peer methods when applied to optimal control problems where the state equation is an ordinary differential equation. That is we consider the following nonlinear optimal control problem

$$\text{minimize } C(y(T)) \tag{4.1}$$
$$\text{subject to } y'(t) = f(y(t), u(t)), \quad t \in (0, T], \tag{4.2}$$
$$y(0) = y_0. \tag{4.3}$$

Here $y : [0, T] \to \mathbb{R}^m$ denotes the state, $u : [0, T] \to U \subset \mathbb{R}^d$ the control, $f : \mathbb{R}^m \times \mathbb{R}^d \to \mathbb{R}^m$ and $C : \mathbb{R}^m \to \mathbb{R}$ the objective function. The set of admissible controls $U \subset \mathbb{R}^d$ is closed and convex.

Remark 4.1.1. Note that it is enough to just look at final time observation objective functions. An optimal control problem with a tracking term in the objective of the form

$$\text{minimize } \int_0^T C_{track}(y(t), u(t))\, dt \tag{4.4}$$
$$\text{subject to } y'(t) = f(y(t), u(t)), \quad t \in (0, T], \tag{4.5}$$
$$y(0) = y_0. \tag{4.6}$$

can be transformed to a final time observation problem by adding an additional state variable $y_{track}(t) = \int_0^t C_{track}(y(s), u(s))\, ds$ leading to the problem

$$\text{minimize } y_{track}(T)\, dt \tag{4.7}$$
$$\text{subject to } y'(t) = f(y(t), u(t)), \quad y'_{track}(t) = C_{track}(y(t), u(t)) \quad t \in (0, T], \tag{4.8}$$
$$y(0) = y_0,\ y_{track}(0) = 0. \tag{4.9}$$

□

If the right-hand side $f(y, u)$ and the objective function C are sufficiently smooth, there exist associated Lagrange multipliers $p^* : [0, T] \to \mathbb{R}^m$ such that an optimal solution (y^*, u^*) satisfies the following first order optimality system:

$$(y^*)'(t) = f(y^*(t), u^*(t)), \qquad t \in (0, T],\ y^*(0) = y_0, \tag{4.10}$$
$$(p^*)'(t) = -\nabla_y f(y^*(t), u^*(t))^T p^*(t), \qquad t \in [0, T),\ p^*(T) = \nabla C(y^*(T)), \tag{4.11}$$
$$-\nabla_u f(y^*(t), u^*(t))^T p^*(t) \in N_U(u^*(t)), \quad t \in [0, T]. \tag{4.12}$$

Here $\nabla_y f$ and $\nabla_u f$ are the Jacobian matrices of f with respect to y and u. Finally, the normal cone mapping $N_U(u)$ is defined for any $u \in U$ as

$$N_U(u) = \{w \in \mathbb{R}^d : w^T(v-u) \leq 0 \text{ for all } v \in U\}. \tag{4.13}$$

There are basically two different approaches to solve the optimal control problem. In the *first-optimize-then-discretize* approach, one solves directly the first order optimality system. That is, one applies an appropriate numerical solver to the system (4.10)-(4.12). In the *first-discretize-then-optimize* approach, one discretizes first the optimal control problem (4.1)-(4.3), which leads to a finite dimensional optimization problem. Then one derives the first order optimality system and solves the system of nonlinear equations. It is desirable that this nonlinear system is an approximation to the infinite dimensional optimality system (4.10)-(4.12).

In Section 4.2 we derive the discrete optimal control problem by discretizing the constraint with an implicit peer method. Then in Section 4.3 and Section 4.4, we analyse the resulting discretization for consistency and stability. We solve the consistency conditions to construct implicit peer methods suitable for optimal control problems in Section 4.5. Finally we show some numerical experiments for the constructed methods in Section 4.6. The results of this chapter have been published in [51].

4.2. Discrete optimal control problem

In the *first-discretize-then-optimize* approach, we start by discretizing the differential equations (4.2) in the optimal control problem by an ODE solver.

Let's recall that the implicit peer method for constant step sizes in compact form as in equation (3.3) is given by

$$Y^n = (U \otimes I_m)Y^{n-1} + h(A \otimes I_m)F(Y^n, U^n), \quad 1 \leq n \leq N. \tag{4.14}$$

Matrix A is a lower triangular matrix with non vanishing diagonal elements, hence it is invertible. Let $G := A^{-1}$ and $B := A^{-1}U$. With these definitions we have the following alternative formulation of an implicit peer method:

$$(G \otimes I_m)Y^n = (B \otimes I_m)Y^{n-1} + hF(Y^n, U^n), \quad 1 \leq n \leq N. \tag{4.15}$$

Since G is again a lower triangular matrix, the different stages can be computed one after the other. If the original implicit peer method is singly diagonal implicit, the alternative method is also singly diagonal implicit. In the context of optimal control, the alternative formulation is easier to analyse as can be seen later.

Implicit peer methods require s initial values y_i^0, $1 \leq i \leq s$. These values are computed by a starting method $\Psi_s(y^0, U^0)$, e.g. by a Runge-Kutta method of the same or higher order. We collect the initial values in one vector,

$$Y^0 = \Psi_s(y_0, U^0). \tag{4.16}$$

Now we discretize the differential equation (4.2) from the optimal control problem with the implicit peer method (4.15) and the starting method (4.16). As a result we get the following finite dimensional optimal control problem:

$$\text{minimize } C(y_s^N) \tag{4.17}$$
$$\text{subject to } (G \otimes I_m)Y^n = (B \otimes I_m)Y^{n-1} + hF(Y^n, U^n), \qquad 1 \le n \le N, \tag{4.18}$$
$$Y^0 = \Psi_s(y_0, U^0). \tag{4.19}$$

We introduce Lagrange multipliers p_i^n for every state equation and combine them in the vector $P^n = (p_1^n, \dots, p_s^n)^T$ for $n = 0, \dots, N$. In compact form the first-order optimality conditions are

$$(G \otimes I_m)Y^n = (B \otimes I_m)Y^{n-1} + hF(Y^n, U^n), \tag{4.20}$$
$$Y^0 = \Psi_s(y_0, U^0), \tag{4.21}$$
$$(G^T \otimes I_m)P^N = (0, \dots, 0, \nabla C(y_s^N))^T + h\nabla_Y F(Y^N, U^N)^T P^N, \tag{4.22}$$
$$(G^T \otimes I_m)P^n = (B^T \otimes I_m)P^{n+1} + h\nabla_Y F(Y^n, U^n)^T P^n, \tag{4.23}$$
$$P^0 = (B^T \otimes I_m)P^1 + \nabla_Y \Psi_s(y_0, U^0)^T P^0, \tag{4.24}$$
$$-\nabla_{U^n} F(Y^n, U^n)^T P^n \in N_U(U^n), \tag{4.25}$$
$$-\nabla_{U^0} \Psi_s(y_0, U^0)^T P^0 \in N_U(U^0), \tag{4.26}$$
$$1 \le i \le s, \quad 1 \le n \le N. \tag{4.27}$$

We assume the existence of a local solution and sufficient smoothness of the optimal control problem. Then the control uniqueness property introduced in [21] states that if (y, p) are sufficiently close to the optimal solution (y^*, p^*), there is a locally unique minimizer $u = u(y, p)$ of the Hamiltonian $f(y, u)^T p$. Following [32], we remove the control variables u_i^n in the optimality system by inserting this unique minimizer.

With the functions

$$\phi(y, p) = -\nabla_y f(y, u(y, p))^T p, \quad g(y, p) = f(y, u(y, p)),$$

the transformed optimality system reads

$$(G \otimes I_m)Y^n = (B \otimes I_m)Y^{n-1} + hG(Y^n, P^n), \tag{4.28}$$
$$Y^0 = \Psi_s(y_0, U^0(Y^0, P^0)), \tag{4.29}$$
$$(G^T \otimes I_m)P^N = (0, \dots, 0, \nabla_y C(y_s^N))^T - h\Phi(Y^N, P^N), \tag{4.30}$$
$$(G^T \otimes I_m)P^n = (B^T \otimes I_m)P^{n+1} - h\Phi(Y^n, P^n), \tag{4.31}$$
$$P^0 = (B^T \otimes I_m)P^1 + \nabla_Y \Psi_s(y_0, U^0(Y^0, P^0))^T P^0, \tag{4.32}$$

where $\Phi(Y^n, P^n) = (\phi(y_1^n, p_1^n), \dots, \phi(y_s^n, p_s^n))^T$. This nonlinear system can be interpreted as a discretization of the two-point boundary-value problem:

$$y'(t) = g(y(t), p(t)), \quad y(0) = y_0, \tag{4.33}$$
$$p'(t) = \phi(y(t), p(t)), \quad p(T) = \nabla C(y(T)). \tag{4.34}$$

To obtain the optimal control of the problem, we compute the solution (Y, P) of the nonlinear system (4.28)-(4.32) and then we solve

$$-\nabla_{U^n} F(Y^n, U^n)^T P^n \in N_U(U^n), \quad 1 \le i \le s, 1 \le n \le N, \tag{4.35}$$

$$-\nabla_{U^0} \Psi_s(y_0, U^0)^T P^0 \in N_U(U^0), \quad 1 \le i \le s. \tag{4.36}$$

4.3. Consistency

Looking at the discretization of the adjoint equations (4.30)-(4.32) we remark, that there are three different schemes involved. The first set of equations (4.30) describe the adjoint initialization steps, i.e. the computation of P^N in the last time interval $(t_{N-1}, t_N]$. Equations (4.31) give a discretization of the adjoint in the interior time domain $(t_1, t_{N-1}]$. Finally, the adjoint termination steps in the time interval $[t_0, t_1]$ are computed by equations (4.32). Since the three schemes are quite different, we want to analyse them separately.

4.3.1. Consistency of the discrete adjoint initialisation and termination steps

The adjoint equations and their discretization march backwards in time. That is why we call the approximations computed by (4.30) in the time interval $(t_{N-1}, t_N]$ the adjoint initialisation steps. Computed by

$$(G^T \otimes I_m) P^N = \left(0, \dots, 0, \nabla C(y_s^N)\right)^T - h\Phi(Y^N, P^N), \tag{4.37}$$

they are generally inconsistent approximations to the solution of (4.34). Discrete adjoint peer methods share this problem with discrete adjoint linear multistep methods [47]. Since the adjoint equations are linear in the multipliers p, still linear convergence of the whole scheme can be obtained [5], if the method is consistent in the interior time domain.

Consider, for example, a general two-stage diagonally implicit peer method. The discrete adjoint initialisation steps are computed by

$$a_{22} p_2^N = \nabla C(y_2^N) - h\Phi(y_2^N, p_2^N), \tag{4.38}$$

$$a_{11} p_1^N + a_{21} p_2^N = -h\Phi(y_1^N, p_1^N). \tag{4.39}$$

If $a_{22} = 1$ than p_2^N would be an $\mathcal{O}(h)$ approximation to the exact value $\nabla C(y_2^N)$, else it is inconsistent. The equation for the first stage value (4.39) looks like an implicit Euler step marching backwards in time. But in general it holds for the coefficients that $a_{11} \neq 1$ and $a_{21} \neq -1$ due to the other consistency conditions in the interior time domain.

The discrete adjoint terminations steps are computed by

$$P^0 = (B^T \otimes I_m) P^1 + \nabla_Y \Psi_s(y_0, U^0(Y^0, P^0))^T P^0.$$

These are, like the adjoint initialization steps, in general inconsistent approximations to the infinite dimensional optimal control solution. This is also known from linear multistep methods [47]. In this work we do not consider these inconsistent steps, since already the discrete adjoint initialisation steps are quite problematic. In analogy to the first forward step, we approximate instead the discrete adjoint values P^0 by using the highly accurate state solution Y^0 and performing a further adjoint step

$$(A^T \otimes I_m)P^0 = (B^T \otimes I_m)P^1 + h\Phi(Y^0, P^0).$$

This approach contradicts however a first-discretize-then-optimize strategy. It remains an open question how to repair the discrete adjoint termination steps properly.

4.3.2. Consistency in the interior time domain

In the interior time domain, the discrete adjoint peer method for the adjoint equation (4.34) is given in the form

$$(G^T \otimes I_m)P^n = (B^T \otimes I_m)P^{n+1} - h\Phi(Y^n, P^n).$$

This is an implicit peer method marching backwards in time. We can now derive order conditions for the discrete adjoint like for the original peer method in 3.1.2.

We replace y_i^n and y_i^{n-1} by the values of the solution $y(t_n + c_i h)$ and $y(t_{n-1} + c_i h)$. Furthermore we use that $g(y_i^n, p_i^n) = y'(t_n + c_i h)$. Analogously we replace p_i^n and p_i^{n+1} by $p(t_n + c_i h)$ and $p(t_{n+1} + c_i h)$. Also we use that $\phi(y_i^n, p_i^n) = p'(t_n + c_i h)$. Taylor series expansion gives

$$y(t_n + c_i h) = \sum_{l=0}^{p} \frac{(c_i h)^l}{l!} y^{(l)}(t_n) + \mathcal{O}(h^{p+1}),$$
$$p(t_n + c_i h) = \sum_{l=0}^{q} \frac{(c_i h)^l}{l!} p^{(l)}(t_n) + \mathcal{O}(h^{q+1}).$$

The residual of the state and the adjoint discretization can be computed by using these Taylor expansions, cf. [49]. Like in [2] we use the node vector $c = (c_1, \ldots, c_s)^T$, the vector consisting only of ones $\mathbb{1} = (1, \ldots, 1)^T$ and the operator $z = h\frac{d}{dt}$ for a compact notation.

An implicit peer method has consistency order p and its discrete adjoint has consistency order q, if the parameter matrices G, B and the node vector c fulfill the following set of equations for $z \to 0$:

$$G \exp(cz) = B \exp((c - \mathbb{1})z) + z \exp(cz) + \mathcal{O}(z^{p+1}), \tag{4.40}$$
$$G^T \exp(cz) = B^T \exp((c + \mathbb{1})z) - z \exp(cz) + \mathcal{O}(z^{q+1}). \tag{4.41}$$

The exponentials of the vectors are defined component wise.

In the following we will also say that the implicit peer method is consistent of state order p and adjoint order q.

We can rewrite (4.40) and (4.41) in matrix form by collecting the coefficients of $z^l/l!$ in columns. This gives

$$GV_0^p = BV_1^p + V_0^p D^p (\Lambda_0^p)^T, \tag{4.42}$$
$$G^T V_0^q = B^T V_2^q - V_0^q D^q (\Lambda_0^q)^T, \tag{4.43}$$

where the matrices are given by

$$\begin{aligned}
(V_0^p)_{i,j} &= c_i^{j-1}, && 1 \le i \le s, 1 \le j \le p+1,\\
(V_0^q)_{i,j} &= c_i^{j-1}, && 1 \le i \le s, 1 \le j \le q+1,\\
(V_1^p)_{i,j} &= (c_i - 1)^{j-1}, && 1 \le i \le s, 1 \le j \le p+1,\\
(V_2^q)_{i,j} &= (c_i + 1)^{j-1}, && 1 \le i \le s, 1 \le j \le q+1,\\
(\Lambda_0^p)_{i,j} &= \delta_{i,j+1}, && 1 \le i \le p+1, 1 \le j \le p+1,\\
(\Lambda_0^q)_{i,j} &= \delta_{i,j+1}, && 1 \le i \le q+1, 1 \le j \le q+1,\\
D^p &= \operatorname{diag}(1,2,\dots,p+1), & D^q &= \operatorname{diag}(1,2,\dots,q+1).
\end{aligned}$$

Remark 4.3.1. A consistent implicit peer method of state order p and adjoint order q has to fulfill $(p+q) * s$ equations. On the other hand, we have $(3s^2+3s)/2$ degrees of freedom. If the node vector is already fixed, it reduces to $(3s^2+s)/2$. □

To understand the discrete adjoint peer method easier, we can transform it to a forward marching scheme. For example, the discrete adjoint of an implicit peer formulation of BDF methods regarded as a forward marching scheme is the BDF method itself.

First we transform the time grid by setting $\tilde{c} = (1+c_1)\mathbb{1} - Sc$ with the anti-diagonal unit matrix

$$S = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix} \in \mathbb{R}^{s,s}.$$

Furthermore we set $\tilde{P}^n = SP^{N-n}$ and $\tilde{Y}^n = SY^{N-n}$ for $1 \le n \le N$, the step size $\tilde{h} = -h$ and $\tilde{\Phi}(\tilde{Y}^n, \tilde{P}^n) = (\tilde{\phi}(\tilde{y}_1^n, \tilde{p}_1^n), \dots, \tilde{\phi}(\tilde{y}_s^n, \tilde{p}_s^n))^T$ with $\tilde{\phi}(\tilde{y}(\tau), \tilde{p}(\tau)) = \tilde{p}'(\tau) = p'(T-\tau)$ for $0 \le \tau \le T$. Then the discrete adjoint becomes

$$(G^T S \otimes I_m)\tilde{P}^n = (B^T S \otimes I_m)\tilde{P}^{n-1} + (S \otimes I_m)\tilde{h}\tilde{\Phi}(\tilde{Y}^n, \tilde{P}^n). \tag{4.44}$$

We can multiply this equation by S^{-1}, which leads to

$$(S^{-1}G^T S \otimes I_m)\tilde{P}^n = (S^{-1}B^T S \otimes I_m)\tilde{P}^{n-1} + \tilde{h}\tilde{\Phi}(\tilde{Y}^n, \tilde{P}^n). \tag{4.45}$$

The consistency equation for the discrete adjoint in matrix notation becomes

$$S^{-1}G^T S\tilde{V}_0^q = S^{-1}B^T S\tilde{V}_1^q + \tilde{V}_0^q D^q (\Lambda_0^q)^T, \tag{4.46}$$

with $(\tilde{V}_0^q)_{i,j} = (1 + c_1 - c_{s-i+1})^{j-1}$ and $(\tilde{V}_1^q)_{i,j} = (c_1 - c_{s-i+1})^{j-1}$ for $1 \leq i \leq s, 1 \leq j \leq q+1$.

In [2] implicit peer methods are constructed, with order $p = s$ for constant step sizes. It is desirable to construct implicit peer methods with consistency order $p = q = s$. Thus we have to fulfil the following order conditions:

$$GV_0^s = BV_1^s + V_0^p D^s (\Lambda_0^s)^T, \tag{4.47}$$

$$S^{-1}G^T S\tilde{V}_0^s = S^{-1}B^T S\tilde{V}_1^s + \tilde{V}_0^s D^s (\Lambda_0^s)^T. \tag{4.48}$$

For equidistant nodes, we have that $V_0^s = \tilde{V}_0^s$ and $V_1^s = \tilde{V}_1^s$. If for the parameter matrices, $G = S^{-1}G^T S$ and $B = S^{-1}B^T S$ hold, the adjoint consistency conditions are equal to the state consistency conditions. Implicit peer formulations of BDF methods have this property. The following theorem states, that these are also the only possible methods of state and adjoint order s for equidistant nodes.

Theorem 4.3.1. *An s-stage implicit peer method with equidistant nodes c_i, which is consistent of state and adjoint order s, is equivalent to an s-stage backward differentiation formula.*

Proof. Let the nodes $c_i = i/s$ and let the corresponding s-stage implicit peer method be of state and adjoint consistency order s. We show step by step, that every stage has to be an s-stage backward differentiation formula. After substituting the numerical approximation y_i^n by the values of the solution $y(t_n + c_i h)$, the first state equation reads

$$g_{11}y(t_n + h/s) = \sum_{i=1}^{s} b_{1i}y(t_n - \frac{(s-i)}{s}h) + hy'(t_n + h/s). \tag{4.49}$$

Since the stage order is s, this has to be an s-stage BDF method with constant step size h/s. From equation (4.45) follows for the first stage of the discrete adjoint, that

$$g_{ss}\tilde{p}(t_n + \tilde{c}_1\tilde{h}) = \sum_{i=1}^{s} b_{s+1-i,s}\tilde{p}(t_n - (1 - \tilde{c}_i)\tilde{h}) + \tilde{h}\tilde{p}'(t_n + \tilde{c}_1\tilde{h}). \tag{4.50}$$

Remember that $\tilde{c}_i = 1 + c_1 - c_{s+1-i} = 1 + 1/s - (s+1-i)/s = i/s = c_i$. Hence this equation again has to be equivalent to an s-stage BDF method.

Let us look at the second state equation:

$$g_{21}y(t_n + h/s) + g_{22}y(t_n + 2h/s) = \sum_{i=1}^{s} b_{2i}y(t_n - \frac{(s-i)}{s}h) + hy'(t_n + 2h/s). \tag{4.51}$$

Note, that b_{2s} is already fixed through equation (4.50). Hence we have $s+1$ degrees of freedom for $s+1$ linear independent consistency equations. Setting $b_{21} = 0$ leads to an s-stage BDF method, which solves the consistency equations. It follows, that it is the unique solution.

For the second stage of the discrete adjoint, we get:

$$g_{s,s-1}\tilde{p}(t_n + \tilde{c}_1\tilde{h}) + g_{s-1,s-1}\tilde{p}(t_n + \tilde{c}_2\tilde{h}) = \sum_{i=1}^{s} b_{s+1-i,s-1}\tilde{p}(t_n - (1 - \tilde{c}_i)\tilde{h}) + \tilde{h}\tilde{p}'(t_n + \tilde{c}_2\tilde{h}).$$

(4.52)

Note again, that $b_{1,s-1}$ and $b_{2,s-1}$ are already fixed by (4.49) and (4.51). As above the s-stage BDF method is the unique solution.

Repeating this argumentation for all s stages of the state and discrete adjoint proves the theorem. □

The example below for the case $s = 2$ and experiments with Mathematica [59] for $s = 3$ motivate the following conjecture.

Conjecture 4.3.2. *An s-stage implicit peer method can be of state and adjoint order s, if and only if the nodes c_i are chosen equidistant.*

If the nodes are chosen equidistant, Theorem 4.3.1 shows, that there is an s-stage implicit peer method of state and adjoint order s, namely an implicit peer formulation of the s-stage backward differentiation formula. Thus one direction of Conjecture 4.3.2 is clear. The other direction could not be proven yet for general s.

Example 4.3.1 (The case $s = 2$)**.** Consider the case for two stages. Then we have the following matrices:

$$G = \begin{pmatrix} g_{11} & 0 \\ g_{21} & g_{22} \end{pmatrix}, \quad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}, \quad c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}.$$

First, we solve the consistency equations for $p = q = 1$. That is we have to fulfill

$$G \begin{pmatrix} 1 & c_1 \\ 1 & c_2 \end{pmatrix} = B \begin{pmatrix} 1 & c_1 - 1 \\ 1 & c_2 - 1 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \text{ and}$$
$$G^T \begin{pmatrix} 1 & c_1 \\ 1 & c_2 \end{pmatrix} = B^T \begin{pmatrix} 1 & c_1 + 1 \\ 1 & c_2 + 1 \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}.$$

Solving for B in the first equation gives

$$B = \begin{pmatrix} g_{11} - \frac{g_{11}-1}{c_2-c_1} & \frac{g_{11}-1}{c_2-c_1} \\ g_{21} - \frac{g_{21}+g_{22}-1}{c_2-c_1} & g_{22} + \frac{g_{21}+g_{22}-1}{c_2-c_1} \end{pmatrix}.$$

Inserting B in the second equation and solving for A shows, that A has to be of the following form:

$$G = \begin{pmatrix} g_{11} & 0 \\ 2 - 2g_{11} & g_{11} \end{pmatrix}.$$

Now consider the second order consistency condition

$$G \begin{pmatrix} c_1^2 \\ c_2^2 \end{pmatrix} = B \begin{pmatrix} (c_1 - 1)^2 \\ (c_2 - 1)^2 \end{pmatrix} + 2 \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}.$$

Inserting G and B gives

$$\begin{pmatrix} g_{11}(c_1 - c_2 + 1) - c_1 + c_2 - 2 \\ -3g_{11}c_1 + 3(g_{11} - 1)c_2 - g_{11} + 3c_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

The second order consistency conditions for the adjoint method are the same equations. Solving for g_{11} and c_2 gives

$$c_2 - c_1 = \frac{1}{2} \text{ and } g_{11} = 3.$$

Since c_2 was chosen equal 1, the nodes have to be equidistant for state and adjoint order 2.

4.4. Stability

For the state discretization, we consider the Dahlquist test equation

$$y'(t) = \lambda y(t), \quad y(0) = y_0, \, \lambda \in \mathbb{C}, \, \mathrm{Re}(\lambda) < 0, \, t > 0. \tag{4.53}$$

Applying the implicit peer method with constant step size to the test equation leads to a recursion $Y^n = M_S(z)Y^{n-1}$ with the stability matrix

$$M_S(z) = (A - zI)^{-1}B, \tag{4.54}$$

where $z = h\lambda$. The corresponding adjoint test equation is

$$p'(t) = -\lambda p(t), \quad p(0) = p_0, \, \lambda \in \mathbb{C}, \, \mathrm{Re}(\lambda) < 0, \, t < 0. \tag{4.55}$$

Applying method (4.31) leads to the recursion $P^n = M_A(z)P^{n-1}$ with the stability matrix

$$M_A(z) = (A^T - zI)^{-1}B^T. \tag{4.56}$$

For its spectrum we have the following result.

Theorem 4.4.1. *The spectrum of $M_A(z)$ is the same as the spectrum of $M_S(z)$. If the implicit peer method is zero-stable (A-stable, L-stable), then the discrete adjoint is also zero-stable (A-stable, L-stable).*

Proof. Let λ be an eigenvalue of $M_S(z)$ with corresponding eigenvector v. Furthermore let $u = (A - zI)v$. From $(A - zI)^{-1}Bv = \lambda v$ we deduce $Bv = \lambda(A - zI)v = \lambda u$. Using the definition of u we get

$$B(A - zI)^{-1}u = B(A - zI)^{-1}(A - zI)v = Bv = \lambda u.$$

Thus λ is an eigenvalue of $B(A - zI)^{-1}$ and hence λ is also an eigenvalue of its transpose $(A - zI)^{-T}B^T = M_A(z)$. □

Remark 4.4.1. Note that A is a real, lower triangular matrix with only positive diagonal elements. Hence A has only real eigenvalues. Since the test equation only considers $\lambda \in \mathbb{C}$ with $\mathrm{Re}(\lambda) < 0$, the resolvent matrix $(A - zI)^{-1}$ is always defined. In fact, this is the reason why the diagonal of A is chosen positive.

4.5. Construction of implicit peer methods for optimal control

By an appropriate Mathematica [59] script (see Appendix C), we solve the consistency conditions (4.42)-(4.43) for a given node vector c.

To test our theoretical consistency results, we compare five methods. The first method is a two stage implicit peer formulation of a BDF2 method, whose coefficients are given in Table 4.1. The method is consistent of state and adjoint order 2.

$$c = \begin{pmatrix} 0.5 \\ 1 \end{pmatrix}, A = \begin{pmatrix} 3 & 0 \\ -4 & 3 \end{pmatrix}, B = \begin{pmatrix} -1 & 4 \\ 0 & -1 \end{pmatrix}$$

Table 4.1.: Coefficients of the BDF2 method in an implicit peer formulation. The BDF2 method is of state and adjoint consistency order 2.

Furthermore we compare four different 3-stage methods. Their coefficients are shown in Table 4.2. An implicit peer formulation of a BDF3 method is consistent of state and adjoint order 3. Method APEER330 is taken from [2]. It is consistent of state order 3, but it is not consistent for the adjoint equations. Finally we constructed two new methods. The adjoint consistent method APEER321 with the same nodes as APEER330. It is only state consistent of order 2 and adjoint consistent of order 1. For method APEER322 we have choosen different nodes, such that we have state and adjoint consistency order 2. All methods are zero stable.

4.6. Numerical illustrations

To illustrate the behaviour of implicit peer methods in optimal control problems, we use some common optimal control benchmarking problems, as in [32]. The underlying ODE systems range from linear, nonstiff to nonlinear and stiff problems. We solve a nonstiff problem with known exact solution [21], the nonlinear Rayleigh problem [24] and the stiff van der Pol oscillator.

The numerically observed convergence order is a least square fit of the errors to a function of the form ch^p. The obtained order is denoted by p_{fit}. For every problem we tested the convergence of the discrete adjoint scheme as it is, that is with inconsistent adjoint initial values, and with given exact adjoint initial values. While in the former case we only expect linear convergence, we should see the full consistency order in the latter case.

	BDF3	APEER330	APEER321	APEER322
State order	3	3	2	2
Adjoint order	3	0	1	2
c_1	1/3	0.296511126416765	0.296511126416765	0.414214
c_2	2/3	0.659116133261284	0.659116133261284	0.707107
c_3	1.0	1.0	1.0	1.0
a_{11}	5.5	5.941440596167428	5.300000000000000	8.000000000000000
a_{21}	-9.0	-12.830644360830028	-8.900000000000000	-14.818008334823144
a_{31}	4.5	10.584902609639268	4.406851053197821	9.696684165245848
a_{22}	5.5	5.951086773520019	5.426329330285161	6.939332504400435
a_{32}	-9.0	-10.905872879819768	-8.428049558441128	-14.818008334823132
a_{33}	5.5	5.745075267045352	5.194869174958150	8.000000000000000
b_{11}	1.0	0.740898324803885	0.265656061642047	7.121345812820955
b_{21}	0.0	-0.633437616377792	0.922864401472186	-6.646494669035405
b_{31}	0.0	0.685442977443295	-0.381669409916413	2.403824686637153
b_{12}	-4.5	-3.763501581749058	-2.224790104705090	-22.142204417002571
b_{22}	1.0	3.894545779328964	-1.721574636978474	20.910023255615279
b_{32}	0.0	-3.454494399598748	0.944644513527589	-6.646494669035413
b_{13}	9.0	8.964043853112601	7.259134043063039	23.020858604181619
b_{23}	-4.5	-10.140665750261180	-2.674960434208554	-22.142204417002613

Table 4.2.: Coefficients of the 3-stage peer methods BDF3, APEER330, APEER321 and APEER322

4.6.1. A nonstiff problem

This problem was proposed in [21]. We want to solve the following quadratic problem with a linear ODE given as constraint:

$$\text{Minimize } \frac{1}{2}\int_0^1 \left(u(t)^2 + 2y(t)^2\right)\, dt \tag{4.57}$$

$$\text{subject to } y'(t) = \frac{1}{2}y(t) + u(t), \quad t \in (0,1], \tag{4.58}$$

$$y(0) = 1. \tag{4.59}$$

We have the optimal solution

$$y^*(t) = \frac{2e^{3t} + e^3}{e^{1.5t}(2+e^3)}, \quad u^*(t) = \frac{2(e^{3t} - e^3)}{e^{1.5t}(2+e^3)}. \tag{4.60}$$

The first-order optimality system reads

$$y'(t) = \frac{1}{2}y(t) + u(t), \qquad t \in (0,1], \quad y(0) = 1, \tag{4.61}$$

$$p'(t) = -\frac{1}{2}p(t) - 2y(t), \quad t \in [0,1), \quad p(1) = 0, \tag{4.62}$$

$$0 = u(t) + p(t). \tag{4.63}$$

Solving (4.63) for u gives $u(t) = -p(t)$. Thus we get the following boundary-value problem

$$y'(t) = \frac{1}{2}y(t) - p(t), \qquad y(0) = 1, \tag{4.64}$$

$$p'(t) = -\frac{1}{2}p(t) - 2y(t), \quad p(1) = 0. \tag{4.65}$$

Numerical results for the five test methods BDF2, BDF3, APEER330, APEER321 and APEER322 with the discrete adjoint initialisation steps are shown in the left column of figure 4.1. APEER330 does not converge, since the adjoint discretization is not consistent. The other four methods only show linear convergence although the methods have higher order. Since the discrete adjoint initialisation steps are not consistent, the adjoint variables and consequently the control only converge with first order.

If we impose the exact adjoint initialisation values, the convergence behaviour changes. The method APEER330 still does not converge, but the other methods show their predicted convergence behaviour. The results are shown in the right column of figure 4.1.

Remark 4.6.1. Generally one does not know the exact adjoint initialisation values or has a reference solution available. Thus in real applications the order of approximation can not be repaired. Imposing exact initial values contradicts furthermore the first-discretize-then-optimize approach. □

4.6.2. The nonlinear unconstrained Rayleigh problem

The Rayleigh problem [24] describes the behaviour of a tunneldiode oscillator. The electric current $y_1(t)$ at time $t \in [0, T]$ is the state variable and the transformed voltage at the generator is the control $u(t)$. The unconstrained Rayleigh problem is in our problem setting given by

$$\text{Minimize } y_3(T) \tag{4.66}$$

$$\text{subject to } y_1'(t) = y_2(t), \tag{4.67}$$

$$y_2'(t) = -y_1(t) + y_2(1.4 - 0.14y_2(t)^2) + 4u(t), \tag{4.68}$$

$$y_3'(t) = u(t)^2 + y_1(t)^2, \quad t \in (0, T], \tag{4.69}$$

$$y_1(0) = -5,\ y_2(0) = -5,\ y_3(0) = 0. \tag{4.70}$$

In our computations we set $T = 2.5$.

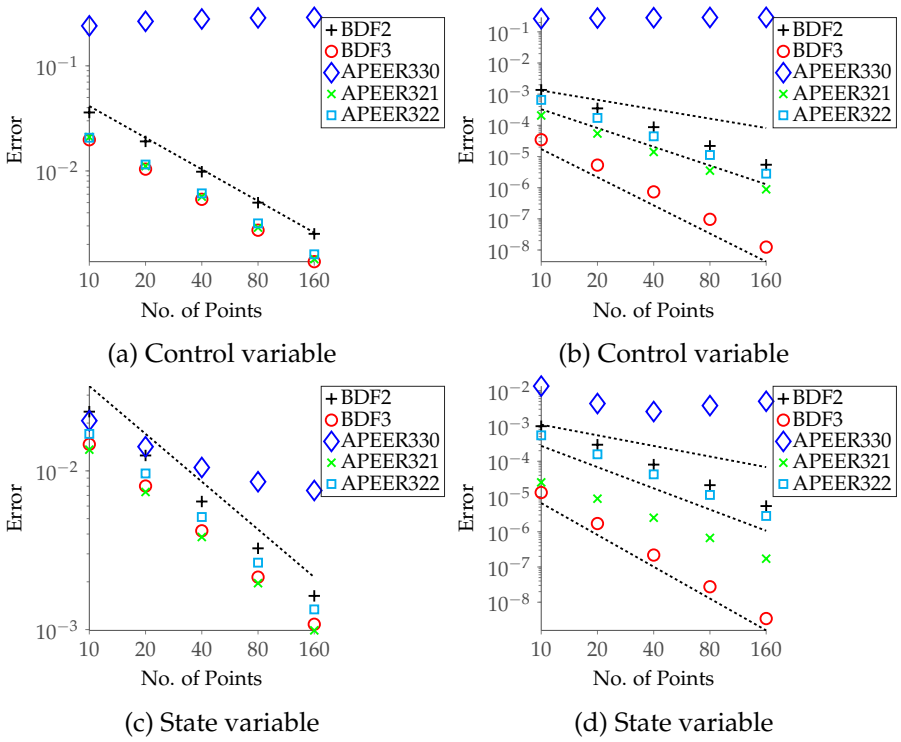
The adjoint equations and the condition for the control are again obtained by derivating the state system with respect to the state y and the control u. Thus we have

$$p_1'(t) = p_2(t) - 2y_1(t)p_3(t), \qquad p_1(T) = 0, \tag{4.71}$$

$$p_2'(t) = -p_1(t) - (1.4 - 0.42p_2(t)^2)p_2(t), \quad p_2(T) = 0, \tag{4.72}$$

$$p_3'(t) = 0, \qquad p_3(T) = 1, \tag{4.73}$$

$$0 = 4p_2(t) + 2u(t)p_3(t), \quad t \in (0, T). \tag{4.74}$$



(a) Control variable

(b) Control variable

(c) State variable

(d) State variable

Figure 4.1.: Test problem 1: Discrete state error $\max_{n=1,\dots,N} \max_{i=1,\dots,s} |y_i^n - y(t_n + c_i h)|$ and the discrete control error $\max_{n=1,\dots,N} \max_{i=1,\dots,s} |u_i^n - u(t_n + c_i h)|$ plotted over the number of time points for the five test methods. In the left column the adjoint initialisation steps were computed by the discrete adjoint scheme. In the right column they were given by the exact solution. The dashed line in the left column visualizes the slope of linear convergence. In the right column the dashed lines stand for linear, quadratic and cubic convergence.

From $p_3'(t) = 0$ and $p_3(T) =$ follows directly that $p_3(t) \equiv 1$. Hence by solving (4.74) for the control, we get that $u(t) = -2p_2(t)$. By separating equation (4.69) from the state system and eliminating the control in the remaining first order optimality conditions, we get the following nonlinear boundary value problem in $[0, T]$:

$$y_1'(t) = y_2(t), \qquad y_1(0) = -5, \tag{4.75}$$

$$y_2'(t) = -y_1(t) + y_2(t)\left(1.4 - 0.14y_2(t)^2\right) - 8p_2(t), \qquad y_2(0) = -5, \tag{4.76}$$

$$p_1'(t) = p_2(t) - 2y_1(t), \qquad p_1(T) = 0, \tag{4.77}$$

$$p_2'(t) = -p_1(t) - \left(1.4 - 0.42y_2(t)^2\right) p_2(t), \qquad p_2(T) = 0. \tag{4.78}$$

We computed a reference solution with the fourth order Runge Kutta method RK4 with step size $h = \frac{2.5}{320}$. The convergence results for the five test methods are shown in in the left column of figure 4.2. As predicted all methods except APEER330 converge linearly to the reference solution. APEER330 does not converge.

The convergence results for the computations with the adjoint initial values computed from the reference solution are shown in the right column of 4.2. Now, we see the full convergence of the methods. The convergence rate of the BDF3 method is disturbed by the bad behaviour for the first time discretization. Again APEER330 does not converge to the reference solution. The convergence of APEER321 for the adjoint is better than expected.

4.6.3. The stiff van der Pol oscillator

Our third problem illustrates the behaviour of the implicit peer methods for stiff problems. We consider an unconstrained optimal control problem for the van der Pol oscillator. Transformed into our setting, the problem reads as follows:

$$\text{Minimize } y_3(T) \tag{4.79}$$
$$\text{subject to } y_1'(t) = -y_2(t) + u(t), \tag{4.80}$$
$$y_2'(t) = \frac{1}{\epsilon}\left(y_1(t) + y_2(t) - \frac{y_2(t)^3}{3}\right), \tag{4.81}$$
$$y_3'(t) = \frac{1}{\epsilon^2}\left(y_1(t) + y_2(t) - \frac{y_2(t)^3}{3}\right)^2 + y_2(t)^2 + u(t)^2, \quad t \in (0, T], \tag{4.82}$$
$$y_1(0) = 2\epsilon,\ y_2(0) = 0,\ y_3(0) = 0. \tag{4.83}$$

In our computations we set $T = 2$ as final time and $\epsilon = 0.1$, thus we look at the van der Pol oscillator in its stiff region.

Using the same approach as before, that is eliminating the control $u(t)$, the auxiliary variable $y_3(t)$ and its adjoint, we get the following nonlinear boundary value problem in $[0, T]$ for the state and the adjoint variables:

$$y_1'(t) = -y_2(t) - \frac{p_1(t)}{2}, \qquad y_1(0) = 2\epsilon, \tag{4.84}$$
$$y_2'(t) = \frac{1}{\epsilon}\left(y_1(t) + y_2(t) - \frac{y_2(t)^3}{3}\right), \qquad y_2(0) = 0, \tag{4.85}$$
$$p_1'(t) = -\frac{1}{\epsilon}p_2(t) - \frac{2}{\epsilon^2}\left(y_1(t) + y_2(t) - \frac{y_2(t)^3}{3}\right), \quad p_1(T) = 0, \tag{4.86}$$
$$p_2'(t) = p_1(t) - 2y_2(t) - \frac{1}{\epsilon}(1 - y_2(t)^2)p_2(t) \tag{4.87}$$
$$- \frac{2}{\epsilon^2}\left(y_1(t) + y_2(t) - \frac{y_2(t)^3}{3}\right)(1 - y_2(t)^2), \quad p_2(T) = 0. \tag{4.88}$$

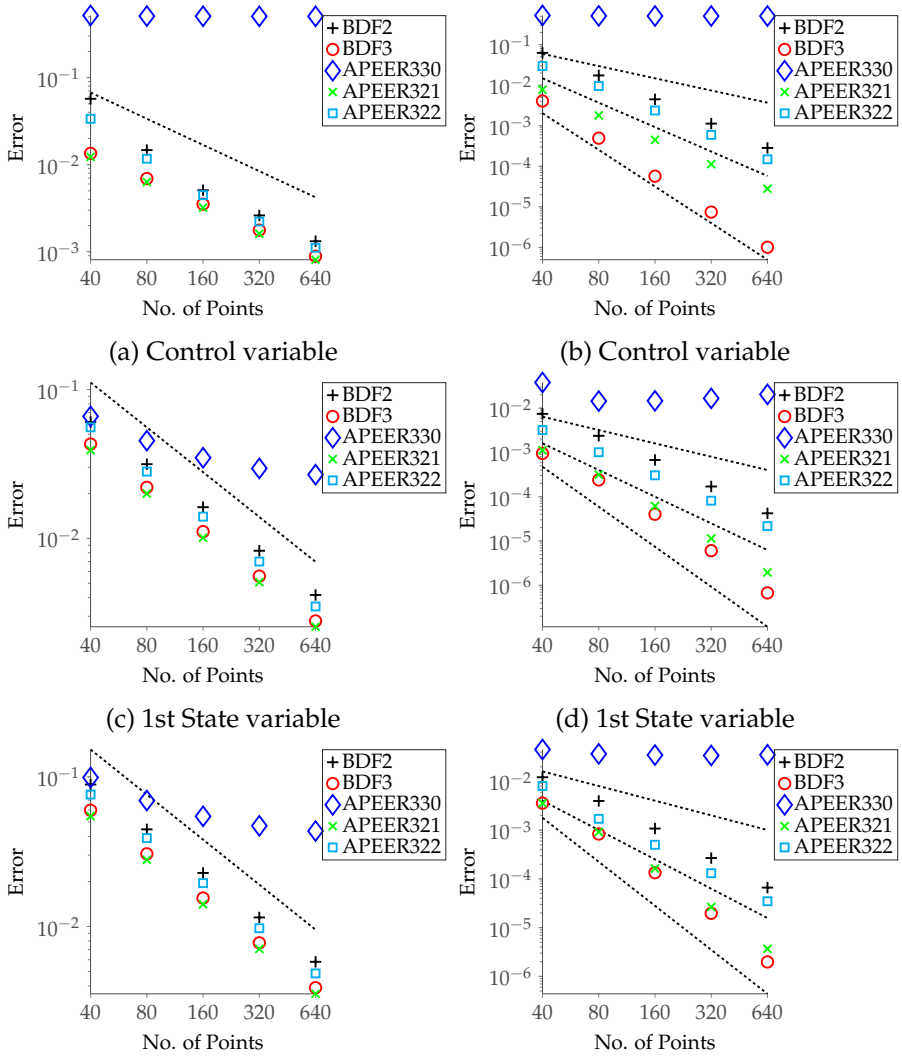
The optimal control can then be computed by $u(t) = -\frac{p_1(t)}{2}$.

Due to the stiffness, we can not use an explicit integrator to compute a reference solution. We computed a reference solution by applying the Rosenbrock W-Method

ROS3WO with $N = 5120$. ROS3WO is convergent of third order for the state and the adjoint. Furthermore it is an L-stable method [32].

The results for the discrete adjoint initialisation steps can be seen in the left column of figure 4.3. Again we see that the convergence order drops down to linear convergence for all methods except APEER330, which does not converge.

If we choose the adjoint initialisation steps corresponding to the reference solution the convergence behaviour changes. The results are plotted in the right column of figure 4.3. As expected the adjoint inconsistent method APEER330 does not converge to the reference solution. The BDF methods show their expected convergence order. APEER322 converges with second order, while APEER321 again shows a slightly better convergence rate than expected.

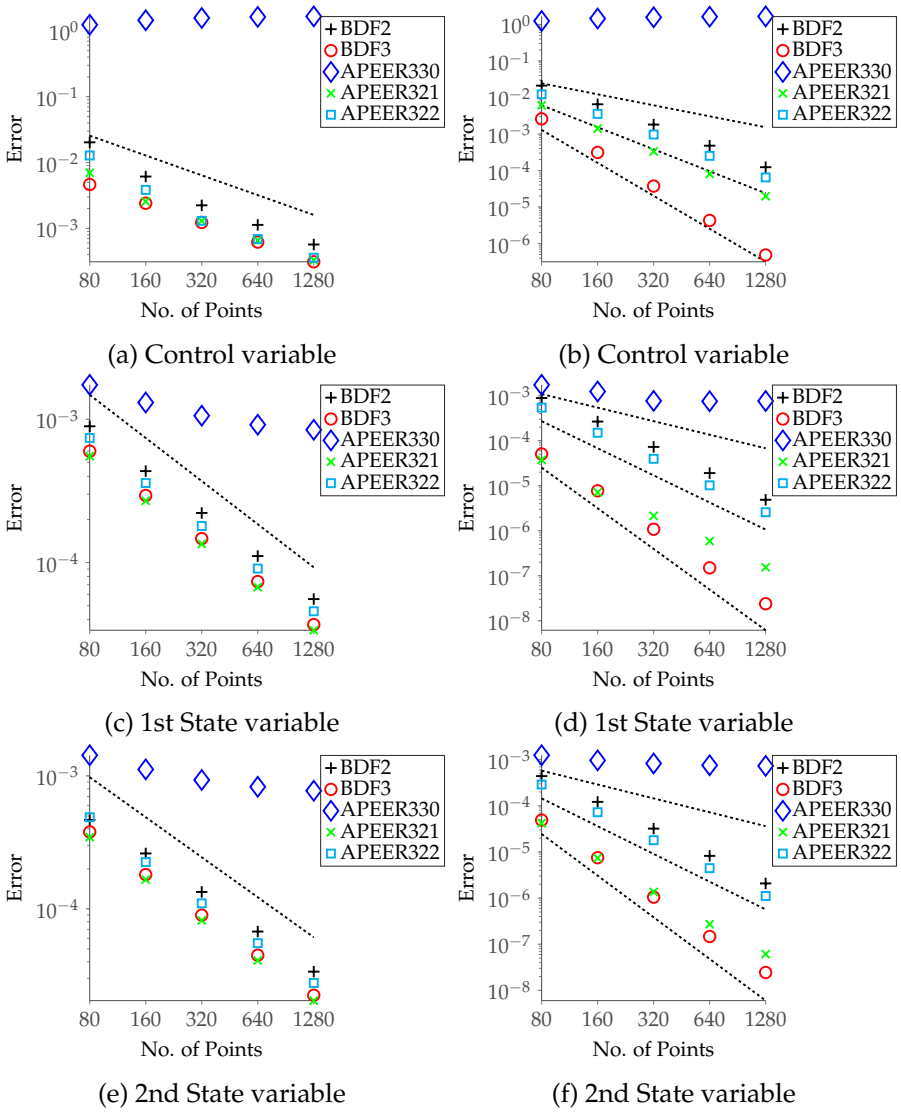


(a) Control variable (b) Control variable

(c) 1st State variable (d) 1st State variable

(e) 2nd State variable (f) 2nd State variable

Figure 4.2.: Rayleigh problem: Discrete state error $\max_{n=1,\dots,n} \max_{i=1,\dots,s} |y_i^n - y(t_n + c_i h)|$ and the discrete control error $\max_{n=1,\dots,n} \max_{i=1,\dots,s} |u_i^n - u(t_n + c_i h)|$ plotted over the number of time points for the five test methods. In the left column the adjoint initialisation steps are computed by the discrete adjoint scheme. In the right column they are given by the reference solution. The dashed line in the left column visualizes the slope of linear convergence. In the right column the dashed lines stand for linear, quadratic and cubic convergence.



(a) Control variable

(b) Control variable

(c) 1st State variable

(d) 1st State variable

(e) 2nd State variable

(f) 2nd State variable

Figure 4.3.: Van der Pol problem: Discrete state error $\max_{n=1,\dots,n} \max_{i=1,\dots,s} |y_i^n - y(t_n + c_i h)|$ and the discrete control error $\max_{n=1,\dots,n} \max_{i=1,\dots,s} |u_i^n - u(t_n + c_i h)|$ plotted over the number of time points for the five test methods. In the left column the adjoint initialisation steps are computed by the discrete adjoint scheme. In the right column they are given by the reference solution. The dashed line in the left column visualizes the slope of linear convergence. In the right column the dashed lines stand for linear, quadratic and cubic convergence.

4.7. Conclusion

In this chapter we analyzed implicit peer methods within the first-discretize-then-optimize approach for the numerical solution of optimal control problems. Consistency of order q in the interior time domain for the discretization of the adjoint equations introduces $s \times q$ new consistency conditions, where s is the number of stages. Implicit peer methods that are equivalent to backward differentiation formulas turn out to be optimal, since they are state and adjoint consistent of order s. It is possible to construct implicit peer methods with higher order in the interior time domain. However, due to the inappropriate adjoint initialisation step the convergence order drops down to one, as it is the case for linear multistep methods. The proper choice of the adjoint initialization step is crucial and can only be successfully handled in a first-optimize-then-discretize approach.

Solving the differential equations adaptively would introduce further inconsistencies also in the interior time domain. Hence it is not advisable to use discrete adjoint implicit peer methods in adaptive computations.

Eventually, we have to conclude that implicit peer methods are not suitable for optimal control problems when used in a first-discretize-then-optimize approach. Nevertheless, due to their good performance for stiff problems they are still an attractive option for the more general strategy first-optimize-then-discretize, where appropriate high-order adjoint initialization and termination steps can be easily included. Here, adjoint inconsistencies can be efficiently controlled, for example, by inexact SQP methods as demonstrated in [14] and [61]. This will be addressed in Chapter 6.

5. The adaptive solution of parabolic problems with peer methods

We have seen in the last chapter, that implicit peer methods are not well suited for a first-discretize-then-optimize-approach. To solve optimization problems with parabolic PDEs as constraints we therefore turn to the first-optimize-then-discretize-approach using an inexact SQP-method presented in Chapter 6. The inexact SQP-method demands space and time adaptive PDE solver.

To this aim, we combine a linearly implicit peer method within the Rothe approach with a multilevel finite element method. Thus we do not use a fixed spatial grid but we build a sequence of nested finite element spaces at every timestep. The nested spaces are constructed adaptively with respect to a local spatial error estimation based on a hierarchical basis.

We start by introducing parabolic PDEs in Section 5.1. In Section 5.2 we discuss the solution of the spatial problems and the spatial error estimation. Then we will repeat in Section 5.3 the strategy presented in [20] for time adaptivity of linearly implicit peer methods. We present some tests concerning the efficiency of the space and the time error estimator for some 1D test problems. Finally we compare the performance of the space-time adaptive peer methods to Rosenbrock methods in Section 5.5.

5.1. Parabolic partial differential equations

We consider the nonlinear initial boundary value problem

$$\begin{aligned} \partial_t y(x,t) &= f(x,t,y(x,t)) && \text{in } \Omega \times (0,T], \\ B(x,t,y(x,t))y(x,t) &= g(x,t,y(x,t)) && \text{on } \partial\Omega \times (0,T], \\ y(x,0) &= y_0(x) && \text{on } \bar{\Omega}. \end{aligned} \tag{5.1}$$

in the same setting as in [34] and [28]. $\Omega \subset \mathbb{R}^d, d = 1, 2$ or 3, denotes a bounded domain with sufficiently smooth boundary $\partial\Omega$. f is a partial differential operator. The boundary operator B stands for a system of boundary conditions interpreted in the sense of traces. g is a given function and y_0 is the initial condition.

We consider a Gelfand triple of separable Hilbert spaces $\mathcal{V}$, $\mathcal{H}$ and $\mathcal{V}'$ with $\mathcal{V} \overset{ds}{\hookrightarrow} \mathcal{H} \overset{ds}{\hookrightarrow} \mathcal{V}'$. We denote the norm on $\mathcal{H}$ induced by the scalar product $(\cdot,\cdot)$ with $|\cdot|$, the norm on $\mathcal{V}$ induced by the scalar product $((\cdot,\cdot))$ with $||\cdot||$, and the dual norm on $\mathcal{V}'$ by $||\cdot||_*$. The anti duality between $\mathcal{V}$ and $\mathcal{V}'$ is denoted by $\langle\cdot,\cdot\rangle$.

With the operator $F : (0,T] \times \mathcal{V} \to \mathcal{V}'$ we rewrite (5.1) as an abstract Cauchy problem of the form

$$\partial_t y(t) = F(t, y(t)), \quad 0 < t \leq T, \quad y(0) = y_0. \tag{5.2}$$

We assume that (5.2) has a unique, temporally smooth solution $y(t)$.

We suppose that F is sufficiently differentiable. We set

$$A(t,w) := -F_y(t,w). \tag{5.3}$$

We assume that $A(t,w) : \mathcal{V} \to \mathcal{V}'$ is a sectorial operator for $t \in (0,T]$ and $w \in \mathcal{W} \subset \mathcal{V}$. The operator $A(t,w)$ is associated with a sesquilinear form

$$a(t,w;v_1,v_2) = \langle A(t,w)v_1, v_2 \rangle, \quad v_1, v_2 \in \mathcal{V}. \tag{5.4}$$

We assume that for all $w \in \mathcal{W}$ and $t \in [0,T]$ the sesquilinear form $a(t,w;v_1,v_2)$ is continuous

$$|a(t,w;v_1,v_2)| \leq M_a ||v_1|| ||v_2|| \quad \forall v_1, v_2 \in \mathcal{V} \tag{5.5}$$

and $\mathcal{V}$-elliptic

$$a(t,w;v_1,v_1) \geq \mu_a ||v_1||^2, \quad \forall v_1 \in \mathcal{V}, \tag{5.6}$$

with constants M_a and μ_a independent of $t \geq 0, w, v_1$ and v_2. Furthermore, we require Lipschitz continuity of $t \mapsto A(t,w(t))$ in the $\mathcal{L}(\mathcal{V},\mathcal{V}')$-norm, i.e.

$$||A(t_2,w(t_2)) - A(t_1,w(t_1))||_{\mathcal{L}(\mathcal{V},\mathcal{V}')} \leq L|t_2 - t_1|, \quad \forall t_1, t_2 \in [0,T]. \tag{5.7}$$

Also we assume for the regularity of the second derivatives of F:

$$||F_{ty}(t,v)v_1||_* \leq C_1 ||v_1|| \qquad \forall v_1 \in \mathcal{V}, \tag{5.8}$$

$$||F_{yy}(t,v)[v_1,v_2]||_* \leq C_2 ||v_1|| ||v_2|| \qquad \forall v_1, v_2 \in \mathcal{V}, \tag{5.9}$$

with C_1, C_2 independent of v varying in bounded subsets of $\mathcal{V}$ and $t \in [0,T]$.

By setting $Q(t,v) = F(t,v) + A(t,v)v$ for all $v \in \mathcal{V}$, we can rewrite (5.2) in the form of a quasilinear Cauchy problem

$$\partial_t y + A(t,y)y = Q(t,y), \quad 0 < t \leq T, \quad y(0) = y_0. \tag{5.10}$$

From continuity (5.5) we deduce that $A(t,v)$ is uniformly bounded and has an uniformly bounded inverse A^{-1} [28]. Furthermore, $\mathcal{V}$-ellipticity (5.6) implies the existence of constants $M > 0$ and angle $\phi < \frac{\pi}{2}$ such that the resolvent bound

$$||(\lambda I + A(t,w))^{-1}||_{\mathcal{L}(\mathcal{V})} \leq \frac{M}{1+|\lambda|}$$

holds for all $w \in \mathcal{W}$ and for all $\lambda \in \mathbb{C}$ with $|\arg(\lambda)| \leq \pi - \phi$ [28]. This setting is the usual one for differential equations of parabolic type and includes the case of semilinear and quasilinear parabolic equations in two and three space dimensions [34].

5.2. Solving the spatial problems

To solve the PDE system (5.2), we follow the Rothe approach, that is, we first discretize in time and then in space. Discretizing in time with a linearly implicit peer method (3.13) leads to the following system:

$$Y_{ni} - Y_{ni}^0 - \tau_n \gamma J_n (Y_{ni} - Y_{ni}^0) = \tau_n \gamma F(t_{ni}, Y_{ni}^0) + w_{ni} - Y_{ni}^0, \quad 1 \leq i \leq s. \tag{5.11}$$

The matrix J_n is an approximation to the Jacobian of F at $(t_n, Y_{n-1,s})$. We only assume that there is a $\omega_n \in \mathcal{W}$ such that

$$J_n = F_y(t_n, \omega_n).$$

Note, that J_n is constant for each time step. Equation (5.11) is a system of linear elliptic equations. The linearity of the resulting elliptic systems is in fact the major motivation to use linearly implicit integration schemes. To solve the linear elliptic problems, we use a multilevel finite element method [28].

Following the Galerkin approach, the weak formulation of (5.11) is given by

$$\begin{aligned} \forall \varphi \in \mathcal{V}: \quad \langle Y_{ni} - Y_{ni}^0, \varphi \rangle - \tau_n \gamma \langle J_n (Y_{ni} - Y_{ni}^0), \varphi \rangle = \\ \langle \tau_n \gamma F(t_{ni}, Y_{ni}^0) + w_{ni} - Y_{ni}^0, \varphi \rangle, \quad 1 \leq i \leq s. \end{aligned} \tag{5.12}$$

Here, φ are the test functions taken from $\mathcal{V}$. Note that since $\mathcal{V}$ is continuously embedded in its dual space $\mathcal{V}'$, we can identify $v \in \mathcal{V}$ as an element in $\mathcal{V}'$. We introduce the stage update $Y_{ni}^U = Y_{ni} - Y_{ni}^0$ and the sesquilinear form

$$b_n(v_1, v_2) := \langle v_1, v_2 \rangle + \tau_n \gamma a(t_n, \omega_n; v_1, v_2), \quad v_1, v_2 \in \mathcal{V}.$$

Then we rewrite the weak formulation (5.12) in the following way:

$$\begin{aligned} \forall \varphi \in \mathcal{V}: \quad b_n(Y_{ni}^U, \varphi) &= \langle \tau_n \gamma F(t_{ni}, Y_{ni}^0) + w_{ni} - Y_{ni}^0, \varphi \rangle, \\ Y_{ni} &= Y_{ni}^0 + Y_{ni}^U, \quad 1 \leq i \leq s. \end{aligned} \tag{5.13}$$

For the following analysis, we introduce a τ-dependent error norm defined by

$$||v||_\tau^2 := \tau ||v||^2 + |v|^2, \quad v \in \mathcal{V} \tag{5.14}$$

and the associated sesquilinear form

$$a_\tau(v_1, v_2) = \tau((v_1, v_2)) + (v_1, v_2), \quad v_1, v_2 \in \mathcal{V}. \tag{5.15}$$

The system (5.13) is uniquely solvable by the Lax-Milgram Lemma, if the sesquilinear form $b_n(\cdot, \cdot)$ is bounded and elliptic. These conditions are satisfied according to the following Lemma cited from [28].

Lemma 5.2.1. *Assume that the sesquilinear form $a(\cdot,\cdot)$, defined in* (5.4)*, satisfies* (5.5) *and* (5.6) *with constants M_a and μ_a, respectively. Then there exist positive constants M_b and μ_b independent of τ_n such that for all functions $v_1, v_2 \in \mathcal{V}$:*

$$|b_n(v_1, v_2)| \leq M_b ||v_1||_\tau ||v_2||_\tau, \tag{5.16}$$

$$b_n(v_1, v_1) \geq \mu_b ||v_1||_\tau^2. \tag{5.17}$$

The constants are given by $M_b = \max(1, \gamma M_a)$ and $\mu_a = \min(1, \gamma \mu_a)$.

The idea of finite elements is to approximate the infinite dimensional solution space $\mathcal{V}$ by a finite dimensional subspace $\mathcal{V}_h$. Multilevel finite elements provide a sequence of nested finite element spaces

$$\mathcal{V}_h^{(0)} \subset \mathcal{V}_h^{(1)} \subset \cdots \subset \mathcal{V}_h^{(m)}$$

based on a sequence of increasingly adapted spatial meshes

$$\mathfrak{T}^{(0)} \subset \mathfrak{T}^{(1)} \subset \cdots \subset \mathfrak{T}^{(m)}.$$

We only consider triangulations of Ω as meshes. In our case the multilevel process stops if a scalar estimate ϵ_{ns}^h of the error $Y_{ns} - Y_{ns}^{h,(m)}$, where $Y_{ns}^{h,(m)}$ solves (5.18), see below, with $\mathcal{V}_h = \mathcal{V}_h^{(m)}$, is smaller than a given tolerance TOL_x, i.e.,

$$\epsilon_{ns}^h \leq TOL_x.$$

Remark 5.2.1. Note that we apply the multilevel process to the whole system (5.12) at once and not individually for each Y_{ni}, i.e., all stages in one time step are computed on the same mesh. Hence, at every time step we have to manage only two different meshes: the actual one and the mesh of the previous time step. □

The multilevel process splits the solution of the stage system into the following four parts:

1. **Solve** the stage system for given mesh $\mathfrak{T}^l$.
2. **Estimate** an error estimation $E_{ni,(m)}^h$.
3. If the computed error estimation is not small enough, we have to **refine** the mesh $\mathfrak{T}^l$ giving us a mesh $\mathfrak{T}^{l+1}$.
4. Else if the MFEM process computed successfully a solution, find an initial mesh for the next time step. That corresponds to **coarsen** the last mesh $\mathfrak{T}^m$.

In the following we will take a closer look at all of these four steps.

5.2.1. Discretization of the stage problems

Given a mesh $\mathfrak{T}$ and the approximations of the previous time step, $Y_{n-1,i} \in \mathcal{V}_{h,prev}$, $i = 1, \ldots, s$, on the mesh $\mathfrak{T}_{prev}$, we want to solve (5.13) on a finite dimensional subspace $\mathcal{V}_h(\mathfrak{T})$ of $\mathcal{V}$. Replacing $\mathcal{V}$ by the FE space $\mathcal{V}_h$ in (5.13) gives the following finite dimensional system to compute $Y_{ni}^{h,U} \in \mathcal{V}_h$:

$$\begin{aligned} \forall \varphi \in \mathcal{V}_h : \quad b_n(Y_{ni}^{h,U}, \varphi) =& \langle \tau_n \gamma F(t_{ni}, Y_{ni}^{h,0}) + w_{ni}^h - Y_{ni}^{h,0}, \varphi \rangle, \\ w_{ni}^h =& \sum_{j=1}^{i-1} \bar{a}_{ij} Y_{nj}^h + \sum_{j=1}^{s} \bar{u}_{ij}(\sigma_n) Y_{n-1,j}^h, \\ Y_{ni}^{h,0} =& \sum_{j=1}^{i-1} \bar{a}_{ij}^0 Y_{nj}^h + \sum_{j=1}^{s} \bar{u}_{ij}^0(\sigma_n) Y_{n-1,j}^h, \\ Y_{ni}^h =& Y_{ni}^{h,0} + Y_{ni}^{h,U}, \quad 1 \le i \le s. \end{aligned} \tag{5.18}$$

To compute the predictor $Y_{ni}^{h,0}$ and w_{ni}^h, the old stage values $Y_{n-1,i} \in \mathcal{V}_{h,prev}$, $i = 1, \ldots, s$, on the previous mesh $\mathfrak{T}_{prev}$ have to be projected to $Y_{n-1,i}^h \in \mathcal{V}_h$ on the current mesh $\mathfrak{T}$. For this projection we use a C^1 interpolation based on the idea of Lawson [33]. The C^1 interpolation gives smoother numerical solutions than for example an L_2-projection and improves the performance of our adaptive solution process. It was already observed in the context of BDF methods in [41] that smoother numerical solutions benefit multistep methods.

We choose the ansatz space $\mathcal{V}_h(\mathfrak{T})$ as the space of continuous, piecewise linear functions,

$$\mathcal{V}_h(\mathfrak{T}) = \{ v \in C(\Omega) : v|_T \in \mathbb{P}_1 \, \forall T \in \mathfrak{T} \} \subset \mathcal{V}.$$

Let $\{\psi_j(x) : j \in \mathcal{J}\}$ be a set of corresponding finite element basis functions on the mesh $\mathfrak{T}$ with the corresponding index set $\mathcal{J}$. The finite element solution Y_{ni}^h can then be written as

$$Y_{ni}^h(x) = \sum_{j \in \mathcal{J}} y_{ni}^{h,j} \psi_j(x).$$

We collect the coefficients $y_{ni}^{h,j}$ into the vector $\mathbf{Y}_{ni}^h$. Similarly we introduce $\mathbf{Y}_{ni}^{h,0}$, $\mathbf{Y}_{ni}^{h,U}$, and $\mathbf{W}_{ni}^h$ for the predictor $Y_{ni}^{h,0}$, the update $Y_{ni}^{h,U}$, and w_{ni}^h, respectively. Finally the coefficients of the projected old stage value $Y_{n-1,i}^h$ are denoted by $\mathbf{Y}_{n-1,i}^h$.

Inserting the finite element representations in (5.18), using linearity, and combining

the equations for all basis functions ψ_l, $l \in \mathcal{J}$, yields the following linear system

$$\begin{aligned}
(\mathcal{M} + \tau_n \gamma \mathcal{S})\, \mathbf{Y}_{ni}^{h,U} =& \tau_n \gamma \mathcal{F}_i + \mathcal{M}(\mathbf{W}_{ni}^h - \mathbf{Y}_{ni}^{h,0}), \\
\mathbf{W}_{ni}^h =& \sum_{j=1}^{i-1} \bar{a}_{ij} \mathbf{Y}_{nj}^h + \sum_{j=1}^{s} \bar{u}_{ij}(\sigma_n) \mathbf{Y}_{n-1,j}^h, \\
\mathbf{Y}_{ni}^{h,0} =& \sum_{j=1}^{i-1} \bar{a}_{ij}^0 \mathbf{Y}_{nj}^h + \sum_{j=1}^{s} \bar{u}_{ij}^0(\sigma_n) \mathbf{Y}_{n-1,j}^h, \\
\mathbf{Y}_{ni}^h =& \mathbf{Y}_{ni}^{h,0} + \mathbf{Y}_{ni}^{h,U}, \quad 1 \le i \le s,
\end{aligned} \tag{5.19}$$

with

$$\mathcal{M}_{kl} = \langle \varphi_l, \varphi_k \rangle,\ \mathcal{S}_{kl} = a(\varphi_l, \varphi_k) \text{ and } \mathcal{F}_{i,k} = \langle F(t_{ni}, Y_{ni}^{h,0}), \varphi_k \rangle,\ 1 \le k, l \le \#\mathcal{J}.$$

The matrix $\mathcal{M}$ is the mass matrix, the matrix $\mathcal{S}$ is the stiffness matrix and $\mathcal{F}_i$ is the load vector of the finite element approximation. The number of basis elements is denoted by $\#\mathcal{J}$. The mass matrix and the stiffness matrix are assembled only once per mesh. If using direct methods to solve the linear system in (5.19), the matrix factorization of the left hand side has to be computed only once and then can be used for all stages. The load vector changes in every stage and thus has to be computed for each i.

Finally the coefficients of the predictor are updated with the computed coefficients of the stage update to yield coefficients of the approximation Y_{ni}^h on the finite element mesh $\mathfrak{T}$, that is, $\mathbf{Y}_{ni}^h = \mathbf{Y}_{ni}^{h,0} + \mathbf{Y}_{ni}^{h,U}$. This is equivalent to $Y_{ni}^h(x) = Y_{ni}^{h,0}(x) + Y_{ni}^{h,U}(x)$ for all $x \in \Omega$.

5.2.2. Estimating the spatial error

To estimate the error of our FE solution Y_{ni}^h, we use the hierarchical basis technique presented for elliptic problems in [16]. The technique was used for the arising linear elliptic systems within a Rothe approach using Rosenbrock methods in [28]. In this section, we will adapt the results for Rosenbrock methods to linearly implicit peer methods.

To this aim, we consider the hierarchical composition

$$\mathcal{V}_h^+ = \mathcal{V}_h \oplus \mathcal{V}_h^\oplus \subset \mathcal{V}.$$

The extension space $\mathcal{V}_h^\oplus$ is built by all basis function which are needed to enrich our base space $\mathcal{V}_h$ to the extended space $\mathcal{V}_h^+$ of higher order. We can compute a solution

$Y_{ni}^{h,+} \in \mathcal{V}_h^+$ by solving

$$\begin{aligned}
\forall \varphi \in \mathcal{V}_h^+ : \quad b_n(Y_{ni}^{h,U,+}, \varphi) =& \langle \tau_n \gamma F(t_{ni}, Y_{ni}^{h,0,+}) + w_{ni}^{h,+} - Y_{ni}^{h,0,+}, \varphi \rangle, \\
w_{ni}^{h,+} =& \sum_{j=1}^{i-1} \bar{a}_{ij} Y_{nj}^{h,+} + \sum_{j=1}^{s} \bar{u}_{ij}(\sigma_n) Y_{n-1,j}^{h,+}, \\
Y_{ni}^{h,0,+} =& \sum_{j=1}^{i-1} \bar{a}_{ij}^0 Y_{nj}^{h,+} + \sum_{j=1}^{s} \bar{u}_{ij}^0(\sigma_n) Y_{n-1,j}^{h,+}, \\
Y_{ni}^{h,+} =& Y_{ni}^{h,0,+} + Y_{ni}^{h,U,+}, \quad 1 \le i \le s.
\end{aligned} \tag{5.20}$$

Here, $Y_{n-1,i}^{h,+}$ denotes the projection of the old stage value $Y_{n-1,i} \in \mathcal{V}_{h,prev}$ to the extended space $\mathcal{V}_h^+$ for $i = 1, \ldots, s$.

We define the error $E_{ni}^h \in \mathcal{V}$ of the finite element solution at time point t_{ni} as $E_{ni}^h = Y_{ni} - Y_{ni}^h$. Furthermore, we denote the difference between the solution in the base space to the solution in the extended space by $E_{ni}^{h,+} = Y_{ni}^{h,+} - Y_{ni}^h \in \mathcal{V}_h^+$.

In order to show that $E_{ni}^{h,+}$ is an efficient and reliable approximation to E_{ni}^h, we make use of the saturation assumption.

Assumption 5.2.1 (Saturation Assumption)**.** There exists a constant $\beta < 1$, such that

$$||Y_{ni} - Y_{ni}^{h,+}||_\tau \le \beta ||Y_{ni} - Y_{ni}^h||_\tau.$$

The saturation assumption is not valid in all cases. Just think of a right-hand side F, such that $Y_{ni}^{h,+} = Y_{ni}^h$. However, for fixed and sufficiently smooth right-hand sides the saturation assumption is fulfilled for sufficiently fine meshes [17]. Furthermore, the saturation assumption holds for the extension of linear elements with quadratic elements, if the data oscillation of F is small [18].

The saturation assumption and the triangle inequality imply

$$\frac{1}{1+\beta} ||E_{ni}^{h,+}||_\tau \le ||E_{ni}^h||_\tau \le \frac{1}{1-\beta} ||E_{ni}^{h,+}||_\tau, \tag{5.21}$$

i.e., $||E_{ni}^{h,+}||_\tau$ is an efficient and reliable error estimator.

Approximation of the spatial error estimator

For ease of notation, we define for vectors $y = (y_1, \ldots, y_s)^T$ and $z = (z_1, \ldots, z_s)^T$ the function

$$\begin{aligned}
r_{ni}(y, z) =& \tau_n \gamma F \left(t_{ni}, \sum_{j=1}^{s} \bar{u}_{ij}^0(\sigma_n) y_j + \sum_{j=1}^{i-1} \bar{a}_{ij}^0 z_j \right) \\
&+ \sum_{j=1}^{s} (\bar{u}_{ij}(\sigma_n) - \bar{u}_{ij}^0(\sigma_n)) y_j + \sum_{j=1}^{i-1} (\bar{a}_{ij} - \bar{a}_{ij}^0) z_j.
\end{aligned} \tag{5.22}$$

Also we denote, for example, the collection of all stage values in one big vector by

$$Y_n^h := \left(Y_{n1}^h, \ldots, Y_{ns}^h\right)^T \in (\mathcal{V}_h)^s.$$

With these definitions, applying the sesquilinear form $b_n(\cdot,\cdot)$ to $E_{ni}^{h,+}$ gives the following system for the spatial error estimator:

$$\begin{aligned} &\forall \varphi \in \mathcal{V}_h^+ : \\ b_n(E_{ni}^{h,+}, \varphi) =& b_n(Y_{ni}^{h,0,+} - Y_{ni}^{h,0}, \varphi) + b_n(Y_{ni}^{h,U,+} - Y_{ni}^{h,U}, \varphi) \\ =& \sum_{j=1}^{s} \bar{u}_{ij}^0(\sigma_n) b_n(Y_{n-1,j}^{h,+} - Y_{n-1,j}^h, \varphi) + \sum_{j=1}^{i-1} \bar{a}_{ij}^0 b_n(E_{nj}^{h,+}, \varphi) \\ &+ \langle r_{ni}(Y_{n-1}^{h,+}, Y_n^h + E_n^{h,+}) - r_{ni}(Y_{n-1}^h, Y_n^h), \varphi \rangle. \end{aligned} \tag{5.23}$$

Note that $r_{ni}(Y_{n-1}^{h,+}, Y_n^h + E_n^{h,+})$ only depends on the first $i-1$ entries of $Y_n^h + E_n^{h,+}$. Especially it does not depend on $E_{ni}^{h,+}$.

Solving (5.23) for $E_{ni}^{h,+}$ on the whole space $\mathcal{V}_h^+$ is very expensive. To be more efficient, we only compute an error approximation $E_{ni}^{h,\oplus}$ on $\mathcal{V}_h^\oplus$ by solving

$$\begin{aligned} &\forall \varphi \in \mathcal{V}_h^\oplus : \\ b_n(E_{ni}^{h,\oplus}, \varphi) =& \sum_{j=1}^{s} \bar{u}_{ij}^0(\sigma_n) b_n(Y_{n-1,j}^{h,+} - Y_{n-1,j}^h, \varphi) + \sum_{j=1}^{i-1} \bar{a}_{ij}^0 b_n(E_{nj}^{h,\oplus}, \varphi) \\ &+ \langle r_{ni}(Y_{n-1}^{h,+}, Y_n^h + E_n^{h,\oplus}) - r_{ni}(Y_{n-1}^h, Y_n^h), \varphi \rangle. \end{aligned} \tag{5.24}$$

One can ask now, whether the approximation $E_{ni}^{h,\oplus}$ is reliable and efficient. We answer this question in the following Theorem 5.2.4. To this end, we further make the assumption that the strengthened Cauchy-Schwarz inequality holds.

Assumption 5.2.2 (Strengthened Cauchy-Schwarz Inequality)**.** There exists a constant $\delta \in [0,1)$ independent of the finite element mesh $\mathfrak{T}$ and the time step size τ such that for the sesquilinear form (5.15) holds

$$|a_\tau(v,w)| \leq \delta ||v||_\tau ||w||_\tau \quad v \in \mathcal{V}_h, w \in \mathcal{V}_h^\oplus.$$

Then we cite the following Lemma from [28].

Lemma 5.2.2. *Let $\mathcal{V}_h^+ = \mathcal{V}_h \oplus \mathcal{V}_h^\oplus$ and $\bar{v} = \hat{v} + \check{v}$, where $\hat{v} \in \mathcal{V}_h$ and $\check{v} \in \mathcal{V}_h^\oplus$. Then the strengthened Cauchy-Schwarz inequality implies*

$$||\check{v}||_\tau \leq \frac{1}{\sqrt{1-\delta^2}} ||\bar{v} - v||_\tau \quad \forall v \in \mathcal{V}_h.$$

Furthermore, we have the following relation between $E_{ni}^{h,+}$ and $E_{ni}^{h,\oplus}$.

Lemma 5.2.3. *It holds for all $\varphi \in \mathcal{V}_h^{\oplus}$ that*

$$b_n(E_{ni}^{h,\oplus}, \varphi) = b_n(E_{ni}^{h,+}, \varphi) + \langle D_{ni}, \varphi \rangle \quad \text{for } i = 1, \dots, s, \tag{5.25}$$

with D_{ni} defined by

$$D_{ni} = \sum_{j=1}^{i} \tilde{a}_{ij}^0 \left(r_{nj}(Y_{n-1}^{h,+}, Y_n^h + E_n^{h,\oplus}) - r_{nj}(Y_{n-1}^{h,+}, Y_n^h + E_n^{h,+}) \right),$$

where $\tilde{a}_{ij}^0$ are the entries of the matrix $\tilde{A}^0 = (I - \bar{A}^0)^{-1}$.

Proof. Let

$$B_n(E, \varphi) = (b_n(E_i, \varphi))_{i=1,\dots,s}$$

and

$$R_n(Y, Z, \varphi) = (\langle r_i(Y, Z), \varphi \rangle)_{i=1,\dots,s}.$$

By collecting the error approximations $E_{ni}^{h,\oplus}$ and $E_{ni}^{h,+}$ in vectors $E_n^{h,\oplus}$ and $E_n^{h,+}$ we get for all $\varphi \in \mathcal{V}_h^{\oplus}$ that

$$\begin{aligned} B_n(E_n^{h,\oplus} - E_n^{h,+}, \varphi) =& \bar{A}^0 B_n(E_n^{h,\oplus} - E_n^{h,+}, \varphi) \\ &+ R_n(Y_{n-1}^{h,+}, Y_n^h + E_n^{h,\oplus}, \varphi) - R_n(Y_{n-1}^{h,+}, Y_n^h + E_n^{h,+}, \varphi). \end{aligned}$$

Hence we have that

$$B_n(E_n^{h,\oplus} - E_n^{h,+}, \varphi) = (I - \bar{A}^0)^{-1} \left(R_n(Y_{n-1}^{h,+}, Y_n^h + E_n^{h,\oplus}, \varphi) - R_n(Y_{n-1}^{h,+}, Y_n^h + E_n^{h,+}, \varphi) \right),$$

which concludes the proof. □

Using definition (5.22), we have the following splitting

$$D_{ni} = \tau_n D_{ni}^1 + D_{ni}^2 \tag{5.26}$$

with

$$\begin{aligned} D_{ni}^1 =& \gamma \sum_{k=1}^{i} \tilde{a}_{ik}^0 \Bigg(F\Big(t_{nk}, \sum_{j=1}^{s} \bar{u}_{kj}^0(\sigma_n) Y_{n-1,j}^{h,+} + \sum_{j=1}^{k-1} \bar{a}_{kj}^0 (Y_{nj}^h + E_{nj}^{h,\oplus})\Big) \\ &- F\Big(t_{nk}, \sum_{j=1}^{s} \bar{u}_{kj}^0(\sigma_n) Y_{n-1,j}^{h,+} + \sum_{j=1}^{k-1} \bar{a}_{kj}^0 (Y_{nj}^h + E_{nj}^{h,+})\Big) \Bigg). \end{aligned} \tag{5.27}$$

and

$$D_{ni}^2 = \sum_{k=1}^{i} \tilde{a}_{ik}^0 \sum_{j=1}^{k-1} (\bar{a}_{kj} - \bar{a}_{kj}^0)(E_{nj}^{h,\oplus} - E_{nj}^{h,+}). \tag{5.28}$$

The following theorem states that the error estimates $E_{ni}^{h,\oplus}$ are reliable and efficient up to some small perturbations.

Theorem 5.2.4. *Let $E_{ni}^{h,+} = \bar{E}_{ni}^{h} + \bar{E}_{ni}^{h,\oplus}$ where $\bar{E}_{ni}^{h} \in \mathcal{V}_h$ and $\bar{E}_{ni}^{h,\oplus} \in \mathcal{V}_h^{\oplus}$. D_{ni}^1 and D_{ni}^2 are given by equations* (5.27) *and* (5.28)*, respectively. Then we have*

$$\frac{\mu_b}{M_b}||E_{ni}^{h,\oplus}||_\tau \leq (1+\beta)||E_{ni}^{h}||_\tau + \frac{1}{M_b}(\sqrt{\tau_n}||D_{ni}^1||_* + |D_{ni}^2|) \tag{5.29}$$

and

$$\begin{aligned}(1-\beta)\mu_b||E_{ni}^{h}||_\tau \leq & \frac{M_b}{\sqrt{1-\delta^2}}||E_{ni}^{h,\oplus}||_\tau + M_b||\bar{E}_{ni}^{h}||_\tau \\ & + \frac{1}{\sqrt{1-\delta^2}}(\sqrt{\tau_n}||D_{ni}^1||_* + |D_{ni}^2|).\end{aligned} \tag{5.30}$$

Proof. Using Lemma 5.2.3 for $\varphi = E_{ni}^{h,\oplus}$ and the ellipticity of the sesquilinear form $b_n(\cdot,\cdot)$ gives

$$\mu_b||E_{ni}^{h,\oplus}||_\tau^2 \leq b_n(E_{ni}^{h,\oplus}, E_{ni}^{h,\oplus}) = b_n(E_{ni}^{h,+}, E_{ni}^{h,\oplus}) + \langle D_{ni}, E_{ni}^{h,\oplus}\rangle.$$

It follows with the continuity of $b_n(\cdot,\cdot)$ and (5.26),

$$\mu_b||E_{ni}^{h,\oplus}||_\tau^2 \leq M_b||E_{ni}^{h,+}||_\tau||E_{ni}^{h,\oplus}||_\tau + (\sqrt{\tau_n}||D_{ni}^1||_* + |D_{ni}^2|)||E_{ni}^{h,\oplus}||_\tau.$$

Then the first inequality follows with (5.21).

To prove the second inequality we use the splitting $E_{ni}^{h,+} = \bar{E}_{ni}^{h} + \bar{E}_{ni}^{h,\oplus}$. We get, again using the ellipticity of $b_n(\cdot,\cdot)$, Lemma 5.2.3, and (5.26),

$$\begin{aligned}\mu_b||E_{ni}^{h,+}||_\tau^2 \leq & \, b_n(E_{ni}^{h,+}, E_{ni}^{h,+}) \\ = & \, b_n(E_{ni}^{h,+}, \bar{E}_{ni}^{h}) + b_n(E_{ni}^{h,+}, \bar{E}_{ni}^{h,\oplus}) \\ = & \, b_n(E_{ni}^{h,+}, \bar{E}_{ni}^{h}) + b_n(E_{ni}^{h,\oplus}, \bar{E}_{ni}^{h,\oplus}) - \langle D_{ni}, \bar{E}_{ni}^{h,\oplus}\rangle \\ \leq & \, M_b\left(||E_{ni}^{h,+}||_\tau||\bar{E}_{ni}^{h}||_\tau + ||E_{ni}^{h,\oplus}||_\tau||\bar{E}_{ni}^{h,\oplus}||_\tau\right) \\ & + (\sqrt{\tau_n}||D_{ni}^1||_* + |D_{ni}^2|)||\bar{E}_{ni}^{h,\oplus}||_\tau.\end{aligned}$$

Applying Lemma 5.2.2 with $v = 0$ gives

$$||\bar{E}_{ni}^{h,\oplus}||_\tau \leq \frac{1}{\sqrt{1-\delta^2}}||E_{ni}^{h,+}||_\tau.$$

Hence

$$\mu_b||E_{ni}^{h,+}||_\tau \leq \frac{M_b}{\sqrt{1-\delta^2}}||E_{ni}^{h,\oplus}||_\tau + M_b||\bar{E}_{ni}^{h}||_\tau + \frac{1}{\sqrt{1-\delta^2}}(\sqrt{\tau_n}||D_{ni}^1||_* + |D_{ni}^2|).$$

Finally (5.21) gives the second inequality. □

Remark 5.2.2. Let us briefly discuss the size of the perturbations on the right hand sides of the inequalities (5.29) and (5.30). Using Taylor expansion, condition (5.9), and the uniform boundedness of the operator A, we get

$$\sqrt{\tau_n}||D^1_{ni}||_* + |D^2_{ni}| \leq C_* \sum_{j=1}^{i-1} ||E^{h,+}_{nj} - E^{h,\oplus}_{nj}||_\tau.$$

Hence, the perturbations depend on the differences $E^{h,+}_{nj} - E^{h,\oplus}_{nj}$ and $E^{h,+}_{nj} - \bar{E}^h_{nj}$, which are in general of moderate size compared to the errors $E^{h,+}_{nj}$ and $E^{h,\oplus}_{nj}$ themselves. Alternatively, making use of $D^1_{n1} = D^2_{n1} = 0$, one could explore the estimator $E^{h,\oplus}_{n1}$ to design a spatial mesh for all stage values and hope that it is especially a good choice for Y_{ns}. □

To make the computation of the error estimation even more computationally efficient, we can replace $b_n(\cdot,\cdot)$ by an approximation $\tilde{b}_n(\cdot,\cdot)$ in (5.24). Then we get an approximation $\tilde{E}^{h,\oplus}_{ni}$ to $E^{h,\oplus}_{ni}$ by solving for all $\varphi \in \mathcal{V}^\oplus_h$:

$$\begin{aligned}\tilde{b}_n(\tilde{E}^{h,\oplus}_{ni}, \varphi) = &\sum_{j=1}^{s} \bar{u}^0_{ij}(\sigma_n) b_n(Y^{h,+}_{n-1,j} - Y^h_{n-1,j}, \varphi) + \sum_{j=1}^{i-1} \bar{a}^0_{ij} \tilde{b}_n(\tilde{E}^{h,\oplus}_{nj}, \varphi) \\ &+ \langle r_{ni}(Y^{h,+}_{n-1}, Y^h_n + \tilde{E}^{h,\oplus}_n) - r_{ni}(Y^h_{n-1}, Y^h_n), \varphi \rangle.\end{aligned} \tag{5.31}$$

Similar to Lemma 5.2.3 we have for all $\varphi \in \mathcal{V}^\oplus_h$ that

$$\begin{aligned}\tilde{b}_n(\tilde{E}^{h,\oplus}_{ni}, \varphi) = &\, b_n(E^{h,\oplus}_{ni}, \varphi) \\ &+ \left\langle \sum_{j=1}^{i} \tilde{a}^0_{ij} \left(r_{nj}(Y^{h,+}_{n-1}, Y^h_n + \tilde{E}^{h,\oplus}_n) - r_{nj}(Y^{h,+}_{n-1}, Y^h_n + E^{h,\oplus}_n) \right), \varphi \right\rangle\end{aligned} \tag{5.32}$$

with the same coefficients $\tilde{a}^0_{ij}$, $i, j = 1, \ldots, s$, as in Lemma 5.2.3. We get the following result concerning the relation between $\tilde{E}^{h,\oplus}_{ni}$ and $E^{h,\oplus}_{ni}$.

Theorem 5.2.5. *Assume that there are positive constants $\tilde{M}_b$ and $\tilde{\mu}_b$ such that the sesquilinear form $\tilde{b}_n(\cdot,\cdot)$ satisfies for all $v_1, v_2 \in \mathcal{V}^\oplus_h$*

$$|\tilde{b}_n(v_1, v_2)| \leq \tilde{M}_b ||v_1||_\tau ||v_2||_\tau, \tag{5.33}$$

$$\tilde{b}_n(v_1, v_1) \geq \tilde{\mu}_b ||v_1||^2_\tau. \tag{5.34}$$

Then

$$\frac{\tilde{\mu}_b}{M_b} ||\tilde{E}^{h,\oplus}_{ni}||_\tau \leq ||E^{h,\oplus}_{ni}||_\tau + \frac{\sqrt{\tau_n}||\tilde{D}^1_{ni}||_* + |\tilde{D}^2_{ni}|}{M_b}, \tag{5.35}$$

$$||E^{h,\oplus}_{ni}||_\tau \leq \frac{\tilde{M}_b}{\mu_b} ||\tilde{E}^{h,\oplus}_{ni}||_\tau + \frac{\sqrt{\tau_n}||\tilde{D}^1_{ni}||_* + |\tilde{D}^2_{ni}|}{\mu_b}, \tag{5.36}$$

with

$$\tilde{D}^1_{ni} = \gamma \sum_{k=1}^{i} \tilde{a}^0_{ik} \Bigg(F\Big(t_{nk}, \sum_{j=1}^{s} \bar{u}^0_{kj}(\sigma_n) Y^{h,+}_{n-1,j} + \sum_{j=1}^{k-1} \bar{a}^0_{kj}(Y^h_{nj} + \tilde{E}^{h,\oplus}_{nj})\Big) - F\Big(t_{nk}, \sum_{j=1}^{s} \bar{u}^0_{kj}(\sigma_n) Y^{h,+}_{n-1,j} + \sum_{j=1}^{k-1} \bar{a}^0_{kj}(Y^h_{nj} + E^{h,\oplus}_{nj})\Big)\Bigg).$$

and

$$\tilde{D}^2_{ni} = \sum_{k=1}^{i} \tilde{a}^0_{ik} \sum_{j=1}^{k-1} (\bar{a}_{kj} - \bar{a}^0_{kj})(\tilde{E}^{h,\oplus}_{nj} - E^{h,\oplus}_{nj}).$$

Proof. The inequalities follow similar to the proof of Theorem 5.2.4 using the relation (5.32) and the conditions (5.16), (5.17), and (5.33), (5.34). □

Remark 5.2.3. To approximate the sesquilinear form we use a diagonalization over $\mathcal{V}^{\oplus}_h$. Instead of one big global computation for $E^{h,\oplus}_{ni}$ we then can compute $\tilde{E}^{h,\oplus}_{ni}$ element-wise. For details we refer to [16, 28]. □

Implementation of the error estimator

Computing the spatial error estimator for every stage value is still a lot of work. Intuitively one expects, that the solution does not change very much during a time step. Therefore it is reasonable to expect, that the spatial error is almost the same for every stage when we use the same mesh during the time step. Hence it would be sufficient to estimate the spatial error for only one stage.

For Rosenbrock methods, within the software *Kardos* the spatial error estimator for the first, suitably scaled stage is computed. This corresponds to estimating the spatial error as if one would use a linearly implicit Euler scheme for the full time step. As this is not possible for peer methods we decided to compute the error estimation for the last stage. The last stage Y^h_{ns} is an approximation at the last time point t_{ns} of the time step and we are using the latest information available, i.e. a function evaluation at t_{ns}. This approach has the disadvantage, that when the spatial grid has to be refined we have to recompute all stages. Rosenbrock methods only have to recompute the first stage. However, the computation on the last refinement level needs the biggest part of the overall computation time per time step. Hence this disadvantage is not as bad as it seems.

So we only compute $E^{h,\oplus}_{ns}$ and assume $E^{h,\oplus}_{ni} = 0$ for $i = 1, \cdots, s-1$. Then it holds that

$$r_{ns}(Y^{h,+}_{n-1}, Y^h_n + E^{h,\oplus}_n) = r_{ns}(Y^{h,+}_{n-1}, Y^h_n).$$

To compute an error estimation we solve the following equation

$$\begin{aligned} \forall \varphi \in \mathcal{V}^{\oplus}_h : b_n(E^{h,\oplus}_{ns}, \varphi) = &\sum_{j=1}^{s} \bar{u}^0_{ij}(\sigma_n) b_n(Y^{h,+}_{n-1,j} - Y^h_{n-1,j}, \varphi) \\ &+ \langle r_{ns}(Y^{h,+}_{n-1}, Y^h_n) - r_{ns}(Y^h_{n-1}, Y^h_n), \varphi \rangle. \end{aligned} \tag{5.37}$$

By adding $0 = \sum_{j=1}^{s-1} \bar{a}_{ij}^0 b_n(Y_{nj}^h, \varphi) - \sum_{j=1}^{s-1} \bar{a}_{ij}^0 b_n(Y_{nj}^h, \varphi)$ we can rearrange the equation to

$$\forall \varphi \in \mathcal{V}_h^{\oplus} : b_n(E_{ns}^{h,\oplus}, \varphi) = b_n(Y_{ns}^{h,0,+} - Y_{ns}^h, \varphi) + \langle r_{ns}(Y_{n-1}^{h,+}, Y_n^h), \varphi \rangle \tag{5.38}$$

with

$$Y_{ns}^{h,0,+} = \sum_{j=1}^{s} \bar{u}_{ij}^0(\sigma_n) Y_{n-1,j}^{h,+} + \sum_{j=1}^{s-1} \bar{a}_{ij}^0 Y_{nj}^h.$$

Let $\{\psi_j^{\oplus}(x) : j \in \mathcal{J}^{\oplus}\}$ be a system of basis functions of $\mathcal{V}_h^{\oplus}$. Then a finite element representation of $E_{ns}^{h,\oplus}$ is given by

$$E_{ns}^{h,\oplus} = \sum_{j \in \mathcal{J}^{\oplus}} e_{ns}^j \psi_j^{\oplus}(x).$$

Inserting this into the defining equation of the spatial error (5.38) gives

$$b_n(\sum_{j \in \mathcal{J}^{\oplus}} e_{ns}^j \psi_j^{\oplus}, \psi_l^{\oplus}) = b_n(Y_{ns}^{h,0,+} - Y_{ns}^h, \psi_l^{\oplus}) + \langle r_{ns}(Y_{n-1}^{h,+}, Y_n^h, \psi_l^{\oplus} \rangle, \quad l \in \mathcal{J}^{\oplus},$$

and using linearity of $b_n(\cdot, \cdot)$ leads to

$$\sum_{j \in \mathcal{J}^{\oplus}} e_{ns}^j b_n(\psi_j^{\oplus}, \psi_l^{\oplus}) = b_n(Y_{ns}^{h,0,+} - Y_{ns}^h, \psi_l^{\oplus}) + \langle r_{ns}(Y_{n-1}^{h,+}, Y_n^h), \psi_l^{\oplus} \rangle, \quad l \in \mathcal{J}^{\oplus}.$$

Inserting the definition of $b_n(\cdot, \cdot)$ and $r_{ns}(\cdot)$ gives for all $l \in \mathcal{J}^{\oplus}$:

$$\begin{aligned}\sum_{j \in \mathcal{J}^{\oplus}} e_{ns}^j \left(\langle \psi_j^{\oplus}, \psi_l^{\oplus} \rangle + \tau_n \gamma a(\psi_j^{\oplus}, \psi_l^{\oplus}) \right) &= \langle Y_{ns}^{h,0,+} - Y_{ns}^h, \psi_l^{\oplus} \rangle \\ &+ \tau_n \gamma a(Y_{ns}^{h,0,+} - Y_{ns}^h, \psi_l^{\oplus}) + \tau_n \gamma \langle F(t_{ns}, Y_{ns}^{h,0,+}), \psi_l^{\oplus} \rangle + \langle w_{ns}^{h,+} - Y_{ns}^{h,0,+}, \psi_l^{\oplus} \rangle,\end{aligned}$$

with

$$w_{ns}^{h,+} = \sum_{j=1}^{s} \bar{u}_{ij}(\sigma_n) Y_{n-1,j}^{h,+} + \sum_{j=1}^{s-1} \bar{a}_{ij} Y_{nj}^h.$$

Let $\mathcal{J}$ be the index set of the basis of $\mathcal{V}_h$ and $\mathcal{J}^{\oplus}$ be the index set of the basis of $\mathcal{V}_h^{\oplus}$. With the definitions

$$\begin{aligned}(\mathcal{A}_{QQ})_{ij} &= \langle \psi_i^{\oplus}, \psi_j^{\oplus} \rangle + \tau_n \gamma a(\psi_i^{\oplus}, \psi_j^{\oplus}), && i, j \in \mathcal{J}^{\oplus}, \\ (\mathcal{A}_{QL})_{ij} &= \langle \psi_i, \psi_j^{\oplus} \rangle + \tau_n \gamma a(\psi_i, \psi_j^{\oplus}), && i \in \mathcal{J}, j \in \mathcal{J}^{\oplus}, \\ (\mathcal{M}_{QQ})_{ij} &= \langle \psi_i^{\oplus}, \psi_j^{\oplus} \rangle, && i, j \in \mathcal{J}^{\oplus}, \\ (\mathcal{M}_{QL})_{ij} &= \langle \psi_i, \psi_j^{\oplus} \rangle, && i \in \mathcal{J}, j \in \mathcal{J}^{\oplus}, \\ (\mathcal{F}_Q)_i &= \langle F(t_{ns}, Y_{ns}^{h,0,+}), \psi_i^{\oplus} \rangle, && i \in \mathcal{J}^{\oplus},\end{aligned}$$

we get the system

$$\begin{aligned}\mathcal{A}_{QQ}\mathbf{e}_{ns} = &\begin{bmatrix}\mathcal{A}_{QL} & \mathcal{A}_{QQ}\end{bmatrix}\begin{bmatrix}\mathbf{Y}^{h,+,0}_{ns,L} - \mathbf{Y}^{h}_{ns} \\ \mathbf{Y}^{h,+,0}_{ns,Q}\end{bmatrix} + \tau_n\gamma\mathcal{F}_Q \\ &+ \begin{bmatrix}\mathcal{M}_{QL} & \mathcal{M}_{QQ}\end{bmatrix}\begin{bmatrix}\mathbf{W}^{h,+,0}_{ns,L} - \mathbf{Y}^{h,+,0}_{ns,L} \\ \mathbf{W}^{h,+,0}_{ns,Q} - \mathbf{Y}^{h,+,0}_{ns,Q}\end{bmatrix}.\end{aligned} \tag{5.39}$$

To make the computation even cheaper, we use not the whole matrix $\mathcal{A}_{QQ}$ but a diagonalization $\tilde{\mathcal{A}}_{QQ}$ given by

$$(\tilde{\mathcal{A}}_{QQ})_{ij} = \begin{cases}\langle\psi_i^{\oplus}, \psi_j^{\oplus}\rangle + \tau_n\gamma a(\psi_i^{\oplus}, \psi_j^{\oplus}), & \text{if } i = j, \\ 0, & \text{else,}\end{cases} \qquad \text{for } i, j \in \mathcal{J}^{\oplus},$$

on the left-hand side. That is we solve

$$\begin{aligned}\tilde{\mathcal{A}}_{QQ}\mathbf{e}_{ns} = &\begin{bmatrix}\mathcal{A}_{QL} & \mathcal{A}_{QQ}\end{bmatrix}\begin{bmatrix}\mathbf{Y}^{h,+,0}_{ns,L} - \mathbf{Y}^{h}_{ns} \\ \mathbf{Y}^{h,+,0}_{ns,Q}\end{bmatrix} + \tau_n\gamma\mathcal{F}_Q \\ &+ \begin{bmatrix}\mathcal{M}_{QL} & \mathcal{M}_{QQ}\end{bmatrix}\begin{bmatrix}\mathbf{W}^{h,+,0}_{ns,L} - \mathbf{Y}^{h,+,0}_{ns,L} \\ \mathbf{W}^{h,+,0}_{ns,Q} - \mathbf{Y}^{h,+,0}_{ns,Q}\end{bmatrix}.\end{aligned} \tag{5.40}$$

Essentially we now compute one-dimensional corrections in each of the directions of the hierarchical basis functions spanning $\mathcal{V}_h^{\oplus}$ instead of the global correction in $\mathcal{V}_h^{\oplus}$. Details can be found in [16].

5.2.3. Refining and coarsening the mesh

We refine and coarse the mesh identically to the way it is done in Kardos for Rosenbrock methods. We summarize at this point shortly the used strategy and refer for more details to [28].

The error estimation gives us for any element $K \in \mathfrak{T}^l$ the local quantity

$$\nu_K := ||E^h_{ns,(l)}||_{\tau,K}.$$

If this quantity is larger than a local barrier ν_{bar} defined by $\nu_{bar} := \frac{1}{4}\max_K \nu_K$, the element K is selected for refinement. In this work we consider only adaptive meshes in two dimensions. Hence we only have to look at the refinement of triangles. After selecting the elements for refinement, all selected triangles are refined into four congruent triangles. This is the so-called "red" refinement. To avoid slave-nodes, triangles with one refined edge are divided into two triangles ("green" refinement) and triangles with more than one refined edge are again refined "red", i.e. divided into four triangles. This process terminates with the refined mesh $\mathfrak{T}^{(l+1)}$.

If the multilevel process ended successfully at $l = m$ with $\mathfrak{T}^{(m)}$ we have to provide an initial mesh $\mathfrak{T}^{(0)}$ to the next integration step. Normally we can assume that a

solution does not alter very much from one time point to the following. So it is reasonable to coarsen the current mesh $\mathfrak{T}^{(m)}$ at regions where the errors are small and take this coarsened mesh as initial mesh $\mathfrak{T}^{(0)}_{next}$. To coarsen the current mesh we follow the "trimming-tree" strategy. Every element whose father has no refined sons is marked for deletion. Here we call a refined element the father of its subelements called sons. Let h denote the characteristic mesh size. Supposing an asymptotic behaviour $\nu_K \sim ch^p$ of the error we can predict the error of the element after deleting the element K by

$$\nu_{pred,K} \sim c(\alpha h)^p \sim \alpha^p \nu_K.$$

For our refinement strategy the parameter α can be chosen as $\alpha = 2$. Since we use linear finite elements, $p = 1$. If the predicted error $\nu_{pred,K}$ is smaller than the local error barrier ν_{bar} the element K is removed from the mesh.

5.3. Estimation of the error in time

5.3.1. Computation of an embedded solution of order $\tilde{p} = s - 2$

When the multilevel finite element method successfully solved the spatial problem (5.11), the error in time has to be estimated and the time step size adapted. Following [20] and similar to Runge-Kutta methods we estimate the local error in time by comparing the computed solution approximation Y^h_{ns} at t_{n+1} to one of lower order. All considered linearly implicit peer methods have order $p = s - 1$ for variable step sizes. We compute an additional solution $\tilde{Y}_{ns}$ at time point t_{n+1} of order $\tilde{p} = s - 2$ as a combination of the approximations Y^h_{ni}, $i = 1, \ldots, s-1$. Hence we compute

$$\tilde{Y}_{ns} = \sum_{i=1}^{s-1} \alpha_i Y^h_{ni}. \tag{5.41}$$

As described in [20] the coefficients α_i, $i = 1, \ldots, s-1$ are determined by looking at the Taylor expansion of $\tilde{Y}_{ns}$. Assume that an exact solution is given by $y(t)$. Then we have for $\tilde{Y}_{ns}$ that

$$\tilde{Y}_{ns} = \sum_{i=1}^{s-1} \alpha_i Y_{ni} = \sum_{i=1}^{s-1} \alpha_i y(t_n + c_i\tau_n) + \mathcal{O}(\tau_n^s) = \sum_{i=1}^{s-1} \alpha_i \sum_{l=0}^{s-2} y^{(l)}(t_n) \frac{c_i^l \tau_n^l}{l!} + \mathcal{O}(\tau_n^{s-1}). \tag{5.42}$$

On the other hand a Taylor expansion of $y(t_n + \tau_n)$ gives

$$y(t_n + \tau_n) = \sum_{i=0}^{s-2} y^{(l)}(\tau_n) \frac{\tau_n^l}{l!} + \mathcal{O}(\tau_n^{s-1}). \tag{5.43}$$

By equating equal powers of τ_n for $l = 0, \cdots, s-2$ we get for the vector of the coefficients $\alpha = (\alpha_1, \cdots, \alpha_{s-1})$ the equation

$$\alpha = {V_{0,s-1}}^{-T} \mathbb{1} \in \mathbb{R}^{s-1}. \tag{5.44}$$

Here $V_{0,s-1} = (c_i^l)$, $i = 1, \cdots, s-1$, $l = 0, \cdots, s-2$ is the Vandermonde matrix for $c_1, \cdots, c_{s-1}$.

With the coefficients of the lower order solution available we can compute an approximation of the local error in time:

$$\epsilon_n = Y_{ns}^h - \tilde{Y}_{ns}. \tag{5.45}$$

5.3.2. Time-step control

With m denoting the number of components of the PDE system (5.1) and the computed local temporal error approximation ϵ_n we define the following scaled scalar local error estimate

$$E_n^\tau := \left(\frac{1}{m} \sum_{i=1}^{m} \frac{||(\epsilon_n)_i||_{L_2}^2}{(ScalR_i \cdot ||(Y_{ns}^h)_i||_{L_2} + ScalA_i \cdot \sqrt{|\Omega|})^2} \right)^{\frac{1}{2}}$$

with user-prescribed relative and absolute scaling factors $ScalR_i$ and $ScalA_i$, respectively, for each component of the PDE system (5.1). If the computed E_n^τ is less than a tolerance TOL_t given by the user, the current step is accepted and otherwise rejected and repeated. In both cases, the step size for the next time step is defined as

$$\tau_{new} = \min \left\{ \tau_{max}, \min \left\{ \sigma_{max}, \max \left\{ \sigma_{min}, (TOL_t / E_n^\tau)^{1/(\tilde{p}+1)} \right\} \right\} \cdot \sigma_{safe} \cdot \tau_n \right\}. \tag{5.46}$$

For the parameters we set, c.f. [20],

$$\sigma_{min} = 0.2, \quad \sigma_{max} = 2, \quad \sigma_{safe} = 0.9.$$

The maximal step size τ_{max} is problem dependent.

An alternative step size controller based on control theory

In [53] the authors propose a controller based on control theory including the approach of using digital filters. These leads to the generic step size recursion

$$\tau_{n+1} = \left(\frac{\epsilon}{E_n^\tau}\right)^{\frac{\beta_1}{k}} \left(\frac{\epsilon}{E_{n-1}^\tau}\right)^{\frac{\beta_2}{k}} \left(\frac{\epsilon}{E_{n-2}^\tau}\right)^{\frac{\beta_3}{k}} \left(\frac{\tau_n}{\tau_{n-1}}\right)^{-\alpha_2} \left(\frac{\tau_{n-1}}{\tau_{n-2}}\right)^{-\alpha_3} \tau_n,$$

where $\epsilon = \theta \cdot TOL_t$, $\theta \leq 1$ is a safety factor and TOL_t is the local time error tolerance. The power k is chosen as $k = p + 1$ for an error-per-step (EPS) control and $k = p$ for an error-per-unit-step ($EPUS$) control with p denoting the order of convergence of the time-stepping method. Since we estimate the time error with an embedded method, we should use the order $\tilde{p}$ of the embedded method rather than the order p of the peer method.

Choosing $\beta_2 = \beta_3 = 0$, $\alpha_2 = \alpha_3 = 0$ and $\beta_1 = 1$ we get the standard controller (5.46) from above. From the control theoretic point view this is a pure I controller, where the

I stands for Integral. It only regulates the step size such that the last error estimation converges to the tolerance.

In [53] a general filter step size controller with $\beta_1 = \frac{5}{4}$, $\beta_2 = \frac{1}{2}$, $\beta_3 = -\frac{3}{4}$, $\alpha_2 = -\frac{1}{4}$ and $\alpha_3 = -\frac{3}{4}$ is proposed. By setting $\beta_2 \neq 0$ a Proportional control part is added which regulates also the change of the eror. Furthermore α_2 and α_3 regulate the change of step size to be more smooth. This is beneficial for multistep methods, because the coefficients of the method depend on the step size ratio. Finally β_3 is an additional Differential controller to control the change ratio of the error estimates to be as small as possible.

In the numerical experiments concerning the performance of peer methods in comparison to Rosenbrock methods we used the standard controller (5.46). Within the multilevel optimization in Chapter 6 we opted for the controller proposed by [53], as this controller gave better results concerning the stability of the method.

Limiting the step size

To ensure a smooth change of step sizes and disallow drastic changes of the step size, we use a step size limiter. Unlike the discontinuous limiter from above, [54] proposes a smooth limiter

$$\hat{\sigma}_n = 1 + \arctan(\sigma_n - 1)$$

for the step size change

$$\sigma_n = \frac{\tau_{n+1}}{\tau_n} = (\frac{TOL_t}{E_n^\tau})^{\beta_1} (\frac{TOL_t}{E_{n-1}^\tau})^{\beta_2} (\frac{TOL_t}{E_{n-2}^\tau})^{\beta_3} \sigma_{n-1}^{-\alpha_2} \sigma_{n-2}^{-\alpha_3}.$$

After computing the factor σ_n the step size is computed by

$$\tau_{new} = \min\{\tau_{max}, \min\{\sigma_{max}, \max\{\sigma_{min}, \sigma_n \cdot \sigma_{safe}\}\} \cdot \tau_n\}.$$

The parameters σ_{max}, σ_{min} and σ_{safe} are chosen as above. The maximal allowed step size τ_{max} is again problem dependent.

Finishing the integration

To avoid unreasonable small last time steps we follow the approach in [31]. Assume that the proposed stepsize is τ_{new}. To guarantee that we reach the endpoint T with an averaged normal step length, we adjust τ_{new} to

$$\tau_{new} \leftarrow \frac{(T - t_n)}{\lfloor (1 + (T - t_n)/\tau_{new}) \rfloor}.$$

5.4. Efficiency of the error estimation

To demonstrate the quality of the hierarchical error estimator and of the time error estimation by a lower order method we present the efficiency index for some test problems in one spatial dimension. We use the following four test problems in 1D.

1. **Linear problem.** This test problem is a heat equation with a linear source term. The domain is the unit interval $\Omega = (0, 1)$ and the equations are

$$\begin{aligned} \partial_t u - \partial_{xx} u &= u, \\ u(0, t) &= u(1, t) = 0, \\ u(x, 0) &= \sin(\pi x). \end{aligned} \tag{5.47}$$

   For this problem we have the following analytic solution:

$$u(x, t) = \exp((1 - \pi^2)t) \sin(\pi x).$$

2. **Ostermann's problem.** The following problem is taken from [39]. It is a heat equation with time-dependent source term on $\Omega = (0, 1)$ and reads

$$\begin{aligned} \partial_t u - \partial_{xx} u &= x \exp(-t), \\ u(0, t) &= u(1, t) = 0, \\ u(x, 0) &= \frac{1}{6} x(1 - x^2). \end{aligned} \tag{5.48}$$

   With the given initial condition, we have the following analytic solution:

$$u(x, t) = \left(\frac{\sin(x)}{\sin(1)} - x \right) \exp(-t).$$

3. **Tanh problem.** This test problem is a heat equation on the domain $\Omega = (-3, 3)$ with nonlinear source term and no-flux boundary conditions given by

$$\begin{aligned} \partial_t u - \partial_{xx} u &= p_3(1 - u^2) + 2p_2^2(u - u^3), \\ \partial_n u(-3, t) &= \partial_n u(3, t) = 0, \\ u(x, 0) &= \tanh(p_2(x - p_1)). \end{aligned} \tag{5.49}$$

   The problem is taken from [28] and was originally published in [38] to study moving-mesh strategies. The analytic solution is given by

$$u(x, t) = \tanh(p_2(x - p_1) + p_3 t).$$

   We set $p_1 = 0.05$ and $p_2 = p_3 = 6.0$. Here, homogeneous Neumann boundary conditions are justified for small t.

4. **Burgers equation.** The considered Burgers equation is taken from [28]. The domain is again the unit interval $\Omega = (0, 1)$ and the equation reads

$$\partial_t u - \mu \partial_x^2 u + u \partial_x u = 0. \tag{5.50}$$

   Initial and Dirichlet boundary conditions are chosen such that the exact solution is given by

$$u(x, t) = \frac{0.1 r_1 + 0.5 r_2 + r_3}{r_1 + r_2 + r_3}$$

   with

$$r_1 = exp(-\frac{x - 0.5}{20\mu} - \frac{99t}{400\mu}), r_2 = exp(-\frac{x - 0.5}{4\mu} - \frac{3t}{16\mu}), r_3 = exp(-\frac{x - 0.375}{2\mu}).$$

n/τ	5.0000e−3	2.5000e−3	1.2500e−3	6.2500e−4
20	0.68326	0.51968	0.35143	0.21334
40	0.90108	0.82037	0.69575	0.53364
80	0.97399	0.94937	0.90373	0.82447
160	0.99348	0.98702	0.97438	0.95007

(a) Efficiency Index for the linear problem

n/τ	5.0000e−3	2.5000e−3	1.2500e−3	6.2500e−4
20	0.68485	0.52084	0.35217	0.21374
40	0.90156	0.82081	0.69613	0.53391
80	0.97408	0.94949	0.90385	0.82458
160	0.99347	0.98704	0.97441	0.95010

(b) Efficiency Index for Ostermann's problem

n/τ	1.2500e−3	6.2500e−4	3.1250e−4	1.5625e−4
160	0.51261	0.34542	0.20903	0.11678
320	0.80885	0.67979	0.51532	0.34727
640	0.94434	0.89492	0.81008	0.68096
1280	0.98543	0.97152	0.94471	0.89529

(c) Efficiency Index for the tanh problem

Table 5.1.: Efficiency of the spatial error estimator

5.4.1. Efficiency of the spatial error estimation

A measure for the efficiency of the spatial error estimation at time point t_{ns} is the efficiency index

$$\text{Eff}_n^{Space} := \frac{||\tilde{E}_{ns}^{h,\oplus}||_\tau}{||Y_{ns} - Y_{ns}^h||_\tau}. \tag{5.51}$$

To test the efficiency index we perform one time step with the peer method on a uniform mesh with n denoting the number of linear finite elements on the mesh and given exact initial values. The time step τ should not be too large, such that the time error does not dominate the spatial error. But it should also not be too small, such that the interpolation error of the initial values does not dominate. We compare then the computed solution with the exact solution. Since we only compute one time step, the global error is the same as the local error and the local spatial error estimation should be the same as the global error. Hence the efficiency index should be approximatively one.

In Table 5.1 the results for three of the test problems, namely the linear, Ostermann's and the tanh problem, are shown. Starting at time point $t_0 = 0.01$ one time step with *peer*4*pos* for different mesh sizes n and different step sizes τ is computed. For fixed τ and increasing mesh size n, i.e. better spatial resolution, the efficiency index converges for all problems to the desired value one.

Note that the error estimator always underestimates the true global error. This stands

n/τ	2.5000e−1	1.2500e−1	6.2500e−2	3.1250e−2
320	1.3594	1.2082	1.0499	1.0182
640	1.3593	1.2082	1.0497	1.0169
1280	1.3593	1.2082	1.0497	1.0165
2560	1.3593	1.2082	1.0497	1.0164

(a) Efficiency index for the linear problem

n/τ	1.0000e−2	5.0000e−3	2.5000e−3	1.2500e−3
640	0.8370	0.9558	0.9471	0.5449
1280	0.8370	0.9595	0.9810	0.9125
2560	0.8370	0.9604	0.9870	0.9852
5120	0.8370	0.9606	0.9883	0.9946

(b) Efficiency index for the tanh problem

n/τ	2.5000e−1	1.2500e−1	6.2500e−2
640	1.0189	0.9980	1.0005
1280	1.0193	0.9979	0.9989
2560	1.0195	0.9979	0.9986

(c) Efficiency index for Burgers problem

Table 5.2.: Efficiency of the time error estimator

in contrast to Rosenbrock solvers where it was observed, that the spatial error estimator tends to overestimate the spatial error [28].

5.4.2. Efficiency of the time error estimation

As for the spatial error estimation we introduce the efficiency index

$$\mathrm{Eff}_n^{Time} := \frac{E_{ns}^{\tau}}{||Y_{ns} - Y_{ns}^{h}||_{L_2}}. \tag{5.52}$$

Again we perform one time step with the peer method on an uniform mesh with n denoting the number of linear finite elements on the mesh and given exact initial values. Now the spatial grid has to be chosen fine enough, such that the time error dominates the spatial error. We only compute one time step, hence the global error equals the local error and we expect of a good estimator an efficiency index of about one.

In Table 5.2 the results for the linear, the tanh and Burgers problem are shown. Like for the spatial error estimation we start at time point $t_0 = 0.01$ and compute one step with *peer*4*pos* for different mesh sizes n and different step sizes τ. For all test problems the efficiency error is about one for fine enough meshes and if the step size is not too large.

5.5. Comparison of peer-FE methods with Rosenbrock-FE methods

In this Section we present a comparison of some Rosenbrock schemes and linearly implicit peer methods for three different test problems in 2D, see Sections 5.5.1-5.5.3. The Rosenbrock schemes used are the second order method *ros*2 [57], the third order method *ros*3*pl* [29], and the fourth order method *rodas*4*p* [55]. See also Appendix B.1 - B.3 for the coefficients of the Rosenbrock methods. As stated in Section 3.3, the linearly implicit peer methods used are based on the coefficients of the singly implicit peer methods presented in [2]. These are the second order method *peer*3*pos*, the third order method *peer*4*pos*, and the fourth order method *peer*5*pos*.

The computations are all performed fully adaptive in time and space. The spatial tolerance TOL_x is always chosen equal to the time tolerance TOL_t, i.e. $TOL_x = TOL_t = TOL$. Several tests showed that this is usually a good choice with respect to accuracy and stability of the whole method.

If an exact solution is available for a test problem, then we approximate the exact $L_2(L_2)$ error in time and space by

$$||y_h - y||_{L_2(L_2)} \approx \left(\sum_{n=1}^{N} \tau_n ||Y_{ns}^h(\cdot) - y(\cdot, t_{ns})||_{L_2} \right)^{\frac{1}{2}}. \tag{5.53}$$

If only a reference solution $\hat{y}$ at the time point T is available, then we take the L_2-norm $||Y_{Ns}^h(\cdot) - \hat{y}(\cdot)||_{L_2}$ as numerical error.

5.5.1. Burgers problem

Our first test problem is the two dimensional Burgers equation [20],

$$\partial_t y_1 - \nabla \cdot (D \nabla u) = -a(y_1 \frac{\partial y_1}{\partial x_1} + y_2 \frac{\partial y_1}{\partial x_2}), \qquad \text{in } I \times \Omega, \tag{5.54}$$

$$\partial_t y_2 - \nabla \cdot (D \nabla v) = -a(y_1 \frac{\partial y_2}{\partial x_1} + y_2 \frac{\partial y_2}{\partial x_2}), \qquad \text{in } I \times \Omega. \tag{5.55}$$

Dirichlet and initial conditions are taken from the exact solution (5.56). The parameters are $D = 0.01$ and $a = 1$. The spatial domain is the unit square $\Omega = (0,1) \times (0,1)$ and the time interval is $I = [0,2]$. The exact solution is given by

$$\begin{aligned} y_1(x_1, x_2, t) &= \frac{3}{4} - \frac{1}{4a} \left(1 + \exp \frac{-4x_1 + 4x_2 - t}{32D} \right)^{-1}, \\ y_2(x_1, x_2, t) &= \frac{3}{4} + \frac{1}{4a} \left(1 + \exp \frac{-4x_1 + 4x_2 - t}{32D} \right)^{-1}. \end{aligned} \tag{5.56}$$

The exact solution depends in space only on the difference $x_1 - x_2$. Hence it is a wave starting at the diagonal $x_1 = x_2$ of the domain, and moving with a constant speed towards its north-west corner.

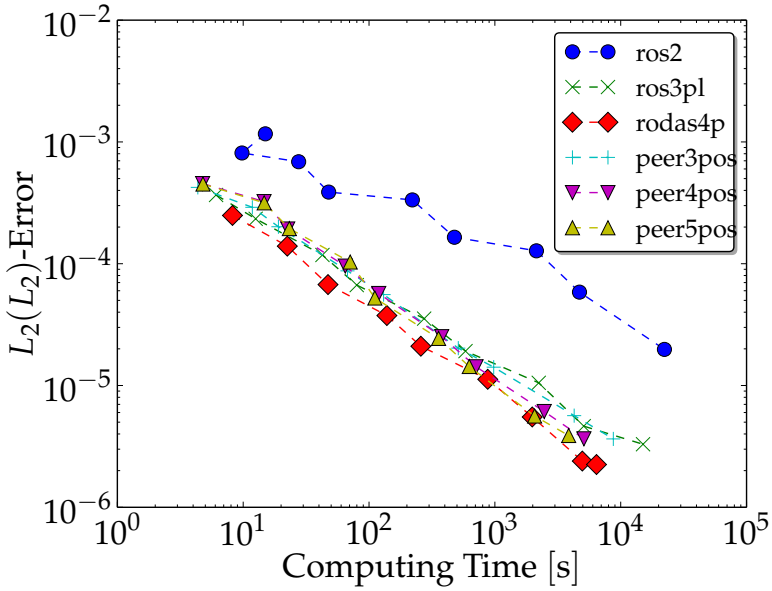


Figure 5.1.: The $L_2(L_2)$-error plotted against the computing time for Burgers problem.

We have performed several runs with decreasing tolerance $TOL = 0.5^n \times 10^{-3}$, $n = 0, \ldots, 8$. In Figure 5.1 we see the $L_2(L_2)$ error of the computed approximation plotted over the computing time. The second order method *ros*2 is the least effective method with respect to achieved error versus computing time. The other methods tested are very similar in their performance. The fourth order methods *rodas*4*p* and *peer*5*pos* are slightly better than the third order methods *ros*3*pl* and *peer*4*pos*. As the wave moves with constant speed the time steps used are constant after a short initial phase. Since *peer*3*pos* is a third order method for constant time steps, it is no surprise that it shows a similar performance to *ros*3*pl*.

In Table 5.3a we give the number of time steps required for the different tolerance values. While the required number of time steps increases by a factor 17 for the finest tolerance compared to the coarsest tolerance for *ros*2, it only doubles for *rodas*4*p* and *peer*5*pos*. For *peer*4*pos* it increases by a factor three, for *ros*3*pl* by a factor four, and for *peer*3*pos* by a factor five, respectively.

Finally in Table 5.3b we see the maximal number of spatial mesh points used for the given tolerance. During the integration with *ros*2 the number of spatial nodes is much smaller than for the other methods. The integration with *rodas*4*p* uses the finest meshes. While the mesh sizes for the same spatial tolerance are similar for peer methods, they differ drastically for Rosenbrock methods.

n/Method	ros2	ros3pl	rodas4p	peer3pos	peer4pos	peer5pos
0	37	20	17	18	17	15
1	53	23	18	22	21	20
2	76	28	20	26	24	21
3	108	34	23	33	27	24
4	155	41	25	42	30	25
5	218	51	28	54	35	27
6	309	63	31	71	41	29
7	436	81	35	94	47	32
8	615	104	39	127	56	34

(a) Number of time steps for a given local tolerance

n/Method	ros2	ros3pl	rodas4p	peer3pos	peer4pos	peer5pos
0	729	968	1088	729	729	729
1	754	1391	2398	1230	1249	1279
2	1201	2782	3709	1477	1536	1600
3	1456	4184	8592	3650	3666	3905
4	3634	11625	13973	4785	5091	5385
5	4876	16782	35826	13802	13157	14792
6	13644	46353	58730	17938	18996	20316
7	18524	70160	119783	51242	50484	58499
8	52705	147994	136896	70430	72998	78243

(b) Maximal number of spatial mesh points for a given local tolerance

Table 5.3.: Number of time steps and maximal number of spatial mesh points for Burgers problem and tolerances $TOL = (0.5)^n \times 10^{-3}$.

5.5.2. Semilinear problem

As a second test problem we consider the following semilinear problem on the unit square $\Omega = (0,1) \times (0,1)$:

$$\partial_t u - \Delta u = f - u^3, \qquad \text{in } I \times \Omega, \tag{5.57}$$

$$u = 0, \qquad \text{on } I \times \partial\Omega, \tag{5.58}$$

$$u(0) = 0. \tag{5.59}$$

The time interval is $I = [0, 0.5]$. This problem was published as a test problem for discontinuous Galerkin methods in [45]. The function f is defined by:

$$f(t,x,y) = \sin(\pi x)\sin(\pi y)\Big(\pi^2\cos(\pi^2 t) + 10 + \Big(\sin(\pi^2 t) + 10t\Big) \Big(2\pi^2 + \big(\sin(\pi^2 t) + 10t\big)^2 \sin(\pi x)^2 \sin(\pi y)^2\Big)\Big).$$

With these settings the problem has an analytic solution given by

$$u(t,x,y) = \big(\sin(\pi^2 t) + 10t\big)\sin(\pi x)\sin(\pi y).$$

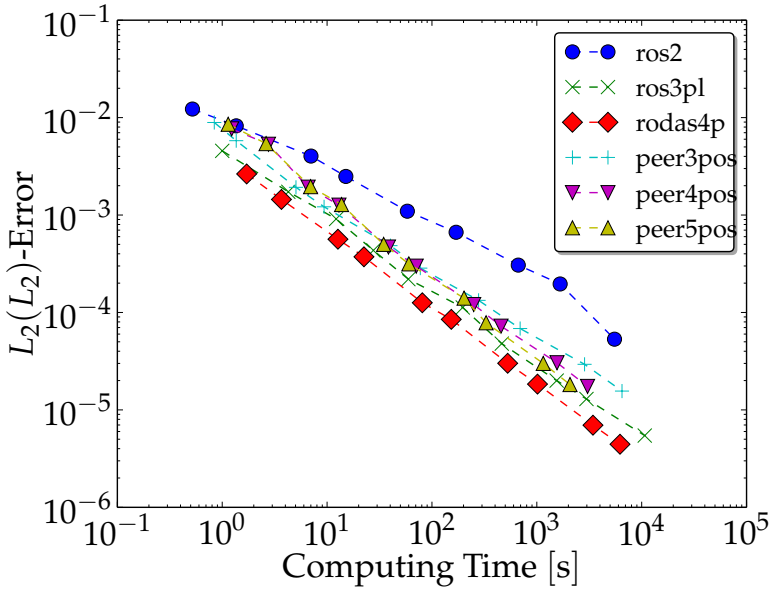


Figure 5.2.: The $L_2(L_2)$-error plotted against the computing time for the semilinear problem

n/Method	ros2	ros3pl	rodas4p	peer3pos	peer4pos	peer5pos
0	11	11	8	10	13	13
1	21	15	10	14	26	23
2	42	19	12	17	20	19
3	58	24	14	23	29	27
4	89	30	16	30	25	23
5	147	38	19	39	32	25
6	222	47	22	53	35	25
7	328	59	25	72	42	28
8	495	74	29	98	50	32

(a) Number of time steps for a given tolerance

n/Method	ros2	ros3pl	rodas4p	peer3pos	peer4pos	peer5pos
0	180	352	653	217	247	233
1	288	914	999	299	300	299
2	633	1565	2819	879	881	874
3	984	3568	3854	1164	1161	1162
4	2310	5809	11288	3346	3433	3156
5	3859	14131	15835	4629	4608	4635
6	9197	24307	44971	11915	13345	13001
7	14844	61703	67981	18841	18748	18405
8	28109	86076	192497	54655	51168	50594

(b) Maximal number of spatial mesh points for a given tolerance

Table 5.4.: Number of time steps and maximal number of spatial mesh points for the semilinear problem.

Again we computed solutions for different tolerances $TOL = 0.5^n \times 10^{-2}$, $n = 0, \ldots, 9$. In Figure 5.2 the achieved error is plotted over the needed computing time. Like for Burgers problem *ros*2 shows the worst performance of the method. The other methods show again a quite similar performance with slight advantages for the Rosenbrock methods *rodas*4*p* and *ros*3*pl*.

The number of time steps used are shown in Table 5.4a. The results are similar to the Burgers problem. *ros*2 needs the most time steps, while *rodas*4*p* and *peer*5*pos* only need a small number of time steps to achieve the given tolerance. *peer*3*pos* is again comparable to *ros*3*pl* and *peer*4*pos*.

In Table 5.4b the maximal number of spatial mesh points required for a given tolerance are shown. Like for Burgers problem, the integration with *rodas*4*p* gives the finest meshes, while the meshes used in the integration with *ros*2 are the coarsest.

5.5.3. Flame problem

As the last test problem, we consider the propagation of a flame front through a cooled channel. This problem was published in [28] as a test problem for fully adaptive Rosenbrock-FE methods.

With a dimensionless temperature T, species concentration Y, and constant diffusion coefficients we look at the following system of equations

$$\begin{aligned} \partial_t T - \Delta T &= \omega(T,Y), && \text{in } \Omega \times (0,60], \\ \partial_t Y - \frac{1}{Le}\Delta Y &= -\omega(T,Y), && \text{in } \Omega \times (0,60], \\ T(\cdot,0) &= T_0(\cdot), && \text{on } \Omega, \\ Y(\cdot,0) &= Y_0(\cdot), && \text{on } \Omega. \end{aligned} \tag{5.60}$$

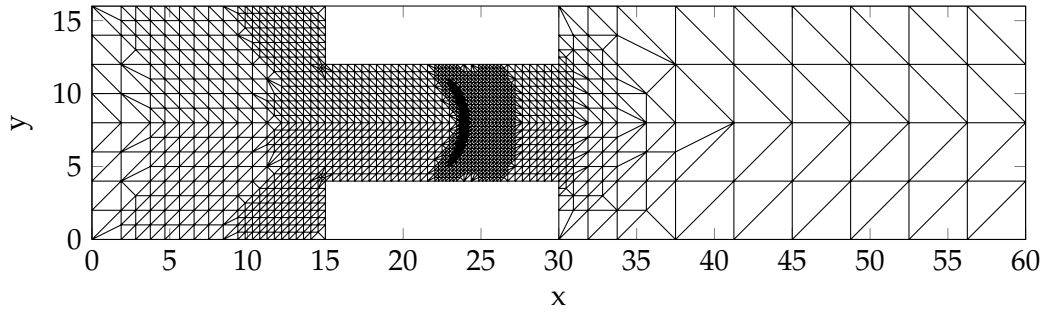
The Lewis number Le is the ratio of heat and mass diffusivity. The reaction is described by an Arrhenius law giving the following one-species reaction mechanism

$$\omega(T,Y) = \frac{\beta^2}{2Le} Y \exp\left(\frac{-\beta(1-T)}{1-\alpha(1-T)}\right). \tag{5.61}$$

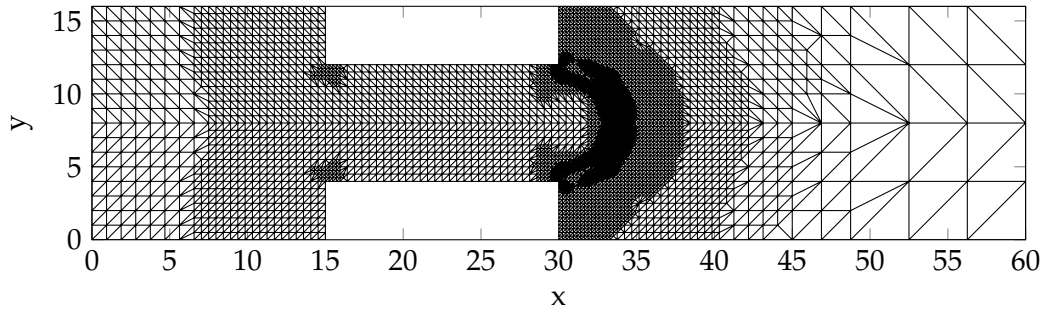
The computational domain is a channel with width $H = 16$ and length $L = 60$. An obstacle with half of the width and length $L/4$ is positioned at $L/4$. The freely propagating laminar flame described by (5.60) is cooled at the obstacle. The heat absorption is modeled by a Robin boundary condition on $\partial\Omega_R$. On the left boundary of the domain, we prescribe Dirichlet conditions. The remaining boundary conditions are of homogeneous Neumann type. All this is represented by the following boundary conditions:

$$T = 1 \text{ on } \partial\Omega_D \times I, \quad \partial_n T = 0 \text{ on } \partial\Omega_N \times I, \quad \partial_n T = -\kappa T \text{ on } \partial\Omega_R \times I, \tag{5.62}$$

$$Y = 0 \text{ on } \partial\Omega_D \times I, \quad \partial_n Y = 0 \text{ on } \partial\Omega_N \times I, \quad \partial_n Y = 0 \text{ on } \partial\Omega_R \times I. \tag{5.63}$$



(a) Flame at time point $t = 20$.



(b) Flame at time point $t = 40$.

Figure 5.3.: Spatial meshes for the Flame problem within a integration with *peer*4*pos* and $TOL = 2.0 \times 10^{-4}$. Above at time point $t = 20$ the flame is inside the channel and in the figure below the flame already left the channel at time point $t = 40$.

As initial condition we set

$$T_0(x,y) = \begin{cases} 1 & \text{for } x \leq x_0 \\ \exp(-(x - x_0)) & \text{for } x_1 > x_0 \end{cases}, \tag{5.64}$$

$$Y_0(x,y) = \begin{cases} 0 & \text{for } x_1 \leq x_0 \\ 1 - \exp(-Le(x - x_0)) & \text{for } x > x_0 \end{cases}. \tag{5.65}$$

The remaining parameters are chosen as

$$Le = 1, \quad \alpha = 0.8, \quad \kappa = 0.1, \quad \beta = 10, \quad x_0 = 9.$$

In Figure 5.3 the spatial meshes at time points $t = 20$ and $t = 40$ are shown for an integration with *peer*4*pos* and $TOL = 2.0 \times 10^{-4}$. The plots illustrate well the importance of spatial adaptivity for the flame problem. At $t = 20$ the flame is inside the channel and that is where most of the mesh refinement takes place. At $t = 40$ the flame has left the channel and now the mesh needs to be refined outside the channel, while inside the channel a coarser mesh is sufficient.

For this problem, we do not have an analytical solution. We computed a reference solution $\hat{y}$ at the time point T with $TOL = 10^{-5}$ using *ros*3*pl* and take the L_2-norm $||Y^h_{Ns}(\cdot) - \hat{y}(\cdot)||_{L_2}$ at time point $t_{end} = 60$ as numerical error.

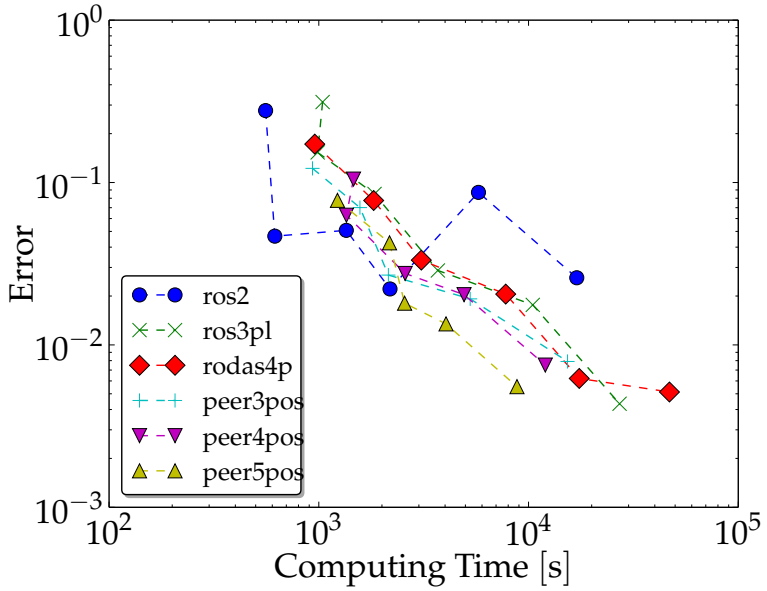


Figure 5.4.: The numerical error $||Y^h_{Ns}(\cdot) - \hat{y}(\cdot)||_{L_2}$ at the final time point $t_{end} = 60$ plotted against the computing time for the flame problem.

n/Method	ros2	ros3pl	rodas4p	peer3pos	peer4pos	peer5pos
0	283	211	276	-	-	-
1	423	262	309	198	249	282
2	641	330	350	254	279	287
3	987	426	398	343	287	268
4	1501	551	456	472	360	282
5	2315	716	524	653	425	284

(a) Number of time steps for a given local tolerance

n/Method	ros2	ros3pl	rodas4p	peer3pos	peer4pos	peer5pos
0	546	1601	2516	-	-	-
1	932	3078	5514	2116	2281	2142
2	1815	6707	10962	3605	3658	3595
3	3357	12926	23836	7072	7204	7204
4	6959	28223	45196	13000	13377	13512
5	13370	52585	89514	27292	27397	27355

(b) Maximal number of spatial mesh points for a given local tolerance

Table 5.5.: Number of time steps and maximal number of spatial mesh points for the flame problem and tolerances $TOL = 0.5^n \times 10^{-3}$.

We ran computations with different tolerances $TOL = 0.5^n \times 10^{-3}$, $n = 0, \ldots, 5$. For the coarsest tolerance $TOL = 10^{-3}$ the peer methods could not successfully compute a solution. For stricter tolerance values there were no difficulties. The performance of the methods is shown in Figure 5.4. With respect to the achieved error versus computing

time *peer*5*pos* shows the best performance. The other methods, except *ros*2, have a similar performance, while *ros*2 is again the worst method.

In Table 5.5a the number of time steps used is shown. *peer*5*pos* needs the smallest number of time steps to satisfy the given error tolerance. Even for tighter tolerances the number of time steps used is only increasing a little. *rodas*4*p* and *peer*4*pos* need almost the same number of time steps for the same tolerance, while *ros*3*pl* needs the same number of time steps as *peer*3*pos*. This demonstrates the advantage of using time integrators not suffering from order reduction.

Again the maximal number of spatial mesh points needed differs a lot for Rosenbrock methods for a tolerance, while the peer methods require almost the same number of mesh points to fulfill a given tolerance, as can be seen in Table 5.5b.

6. PDE constrained optimization with linearly implicit peer methods

In this chapter we look at the performance of peer methods within a first-optimize-then-discretize approach. In [10, 13] a combination of an adaptive multilevel optimization method and an adaptive PDE solver is presented. We will present this algorithm in the first section. In the second section we compare the performance of peer methods with that of Rosenbrock methods for three PDE-constrained optimization problems.

6.1. Multilevel optimization

6.1.1. Multilevel strategy

Using an optimization method for the optimal control problem (2.1) with a fixed level of accuracy, that is all arising differential equations are solved on a fixed mesh, has two disadvantages. First the computed reduced gradient is not necessarily the exact gradient for the current approximation of the optimal control problem. That is, the discrete adjoint is not consistent with the discrete state. The result is a stagnation of the optimization method as the computed gradient is no descent direction. As we have seen earlier this inexact reduction of the reduced gradient can be avoided by choosing the right, adjoint consistent integration methods. As we have also seen earlier peer methods unfortunately are not adjoint consistent. Second the optimization method will reach a point where the accuracy of the computed control is higher than the accuracy of the approximation of the optimal control problem. At this point it is better to refine the control problem approximation, i.e., the state and adjoint system, than to improve the optimal control, i.e., going on with the optimization on the current refinement level.

In [11, 13, 61] optimization methods were combined with a discretization error based multilevel strategy. Here, the presented optimization methods converge on an infinite dimensional level to the infinite dimensional optimal control [61].

Now assume a discrete control iterate $u_k^h \in U_{ad}^h$ is given. We always use a linear finite element space for U_{ad}^h. We then can discretize the state equation (2.3) with the discrete state solution y_k^h by

$$e^h(y_k^h, u_k^h) = 0.$$

Furthermore, we discretize the adjoint equation (2.5) with the discrete adjoint solution p_k^h by

$$\partial_y J^h(y_k^h, u_k^h) + \partial_y e^{h,*}(y_k^h, u_k^h) p_k^h = 0.$$

By discretizing (2.6) we can compute an inexact reduced gradient g_k^h by

$$g_k^h = \partial_u J^h(y_k^h, u_k^h) + \partial_u e^{h,*}(y_k^h, u_k^h) p_k^h.$$

The multilevel strategy is based on controlling the infinite dimensional residual norms of the state and the adjoint equations by a so called criticality measure cm_k. The criticality measure is an indicator of the optimality of the current control iterate u_k^h. In an optimization without control constraints, the norm of the discrete gradient is a good choice for cm_k, i.e.,

$$\mathrm{cm}_k = ||g_k^h||_{U_h}. \tag{6.1}$$

In the presence of control constraints, we consider instead of (6.1) the norm of the negative discrete gradient projected to the admissible control space, i.e.,

$$\mathrm{cm}_k = ||\mathbb{P}_{U_{ad}^h - u_k^h}(-g_k^h)||_{U_h}. \tag{6.2}$$

Here, the element-wise and point-wise projection $\mathbb{P}_{U_{ad}-u_k^h}(s)$ for $s \in U$ is given by

$$\mathbb{P}_{U_{ad}^h - u_k^h}(s) = \begin{cases} u_{low} - u_k^h & \text{if } s \leq u_{low} - u_k^h, \\ u_{up} - u_k^h & \text{if } s \geq u_{up} - u_k^h, \\ s & \text{otherwise.} \end{cases}$$

Given the criticality measure we control the infinite dimensional residual norms by

$$||e(y_k^h, u_k^h)||_{V^*} \leq k_y \,\mathrm{cm}_k, \tag{6.3}$$

$$||\partial_y e^*(y_k^h, u_k^h) p_k^h + \partial_y J(y_k^h, u_k^h)||_{Y^*} \leq k_p \,\mathrm{cm}_k\,. \tag{6.4}$$

Furthermore, the difference between the projection $\mathbb{P}_{U_{ad}-u_k^h}(-g_k^h)$ to the shifted infinite dimensional feasible set $U_{ad} - u_k^h$ and the projection $\mathbb{P}_{U_{ad}^h - u_k^h}(-g_k^h)$ to the shifted finite dimensional set $U_{ad}^h - u_k^h$ is controlled for pointwise control constraints by

$$||\mathbb{P}_{U_{ad}-u_k^h}(-g_k^h) - \mathbb{P}_{U_{ad}^h - u_k^h}(-g_k^h)||_U \leq k_u \,\mathrm{cm}_k\,. \tag{6.5}$$

The infinite dimensional residual norms are hard to compute. Therefore, [60] showed that also appropriate global error estimates with

$$||y_k^h - y_k||_Y \leq \mu_y(y_k^h), \tag{6.6}$$

$$||p_k^h - p_k||_P \leq \mu_p(p_k^h), \tag{6.7}$$

$$||\mathbb{P}_{U_{ad}-u_k^h}(-g_k^h) - \mathbb{P}_{U_{ad}^h - u_k^h}(-g_k^h)||_U \leq \mu_u(g_k^h), \tag{6.8}$$

can be used to control the refinement level. We get the following accuracy conditions

$$\mu_y(y_k^h) \leq c_y \,\mathrm{cm}_k, \tag{6.9}$$

$$\mu_p(p_k^h) \leq c_p \,\mathrm{cm}_k, \tag{6.10}$$

$$\mu_u(g_k^h) \leq c_u \,\mathrm{cm}_k\,. \tag{6.11}$$

If these conditions are fulfilled, the optimization can compute a control update. In this thesis we use a generalized SQP-method which will be described shortly in Section 6.1.3.

6.1.2. Global error estimation

The error estimations presented in Chapter 5 in time and space are both only local error estimators. To control the step size and the mesh size this is sufficient. To control the refinement level with respect to the criticality measure we need global error estimates $\mu_y(y_k^h)$ and $\mu_p(p_k^h)$ for the state and adjoint approximation, respectively.

Let Y_{ms} be the exact solution of the problem (5.2) and Y_{ms}^h the computed approximation at the final time $t_{ms} = T$. The global error is then defined by

$$gerr_{ms} = ||Y_{ms} - Y_{ms}^h||.$$

We bound this error by a temporal and a spatial part, that is

$$gerr_{ms} \leq gerr_{ms}^{space} + gerr_{ms}^{time}.$$

Now we could try to estimate $gerr_{ms}^{space}$ and $gerr_{ms}^{time}$. However this is usually very hard. Therefore we compute instead controllable quantities q_{time} and q_{space} which are proportional to $gerr_{ms}^{time}$ and $gerr_{ms}^{space}$. So even if we do not have global error estimations, we still can control it via the quantities q_{time} and q_{space} [10].

Assume we have computed solutions Y_{ns}^h for $n = 1, \cdots, m$ at time points $t_0 < t_1 < \cdots < t_m = T$. Furthermore we have computed the local error estimations E_{ns}^τ and ϵ_{ns}^h, $n = 1, \cdots, m$. Then we compute a quantity describing the overall temporal error by

$$q_{time} = (T - t_0)^{-\frac{1}{2}} ||(E_{1s}^\tau, \cdots, E_{ms}^\tau)||_2$$

and a quantity describing the spatial error by

$$q_{space} = (T - t_0)^{-\frac{1}{2}} ||(\epsilon_{1s}^h, \cdots, \epsilon_{ms}^h)||_2.$$

This gives us the four estimates $q_{s,time}$, $q_{s,space}$, $q_{a,time}$, and $q_{a,space}$ for the state time error, the state space error, the adjoint time error, and the adjoint space error, respectively. These quantities are then used to steer the multilevel strategy within the optimization algorithm by replacing the accuracy conditions (6.9)-(6.10) with the following four conditions

$$q_{s,time} \leq C_{s,time}\, \mathrm{cm}_k, \tag{6.12}$$
$$q_{s,space} \leq C_{s,space}\, \mathrm{cm}_k, \tag{6.13}$$
$$q_{a,time} \leq C_{a,time}\, \mathrm{cm}_k, \tag{6.14}$$
$$q_{a,space} \leq C_{a,space}\, \mathrm{cm}_k, \tag{6.15}$$

where $C_{s,time}$, $C_{s,space}$, $C_{a,time}$ and $C_{a,space}$ are real constants. For more details we refer to [10]. This approach relies on the assumption that by controlling local errors, we can control the overall global error. In the literature this is known as tolerance proportionality which states that the global error is indeed proportional to the local errors [52].

6.1.3. Iterative trust region SQP-methods

When the multilevel strategy found the right refinement level, we are ready to compute an update s_u to the control u_k giving us a new control iterate u_{k+1}. In a gradient method we would use the negative gradient $-g_k^h$ as control update. As the gradient method is only linearly convergent, we use instead a Newton-type generalized SQP method as presented in [10, 60, 61]. For more details on the implementation we refer to [10].

The idea behind Newton-type optimization is to set the derivative of the reduced optimal control problem (2.4) to zero

$$\hat{J}'(u) = 0,$$

and to apply Newton's method to the resulting nonlinear system. This gives the following iteration formula

$$\hat{J}''(u_k)(u_{k+1} - u_k) = -\hat{J}'(u_k). \tag{6.16}$$

While this is technically correct near the optimal solution, it is not when we are far away [25]. In the latter case the reduced Hessian is not guaranteed to be positive definite and the solution of (6.16) may not be a descent direction.

It is more precise to say that u_{k+1} should be a minimizer of a local quadratic model of $\hat{J}$ around u_k. That gives with the control update $s_u := u_{k+1} - u_k$:

$$m_k(s_u) = \hat{J}(u_k) + \langle \hat{J}'(u_k), s_u \rangle_{U^*,U} + \frac{1}{2}\langle \hat{J}''(u_k)s_u, s_u \rangle_{U^*,U}, \quad \text{s.t. } e(y(u_k), u_k) = 0. \tag{6.17}$$

If the reduced Hessian $\hat{J}''(u_k)$ is positive definite, then the minimizer of (6.17) is the solution of (6.16). However, we have to take care of the case where $\hat{J}''(u_k)$ is not positive definite. One common solution is to add a trust region to the quadratic model. That is we introduce a value Δ_k which restricts the length of the control update s_u. We only trust our quadratic model in a ball with radius Δ_k around the iterate u_k.

The introduction of the trust region globalises the normally only local convergent Newton-type method. So, instead of minimizing (6.17), we solve

$$\begin{aligned} &\min_{s_u} m_k(s_u) = \hat{J}(u_k) + \langle \hat{J}'(u_k), s_u \rangle_{U^*,U} + \frac{1}{2}\langle \hat{J}''(u_k)s_u, s_u \rangle_{U^*,U}, \\ &\quad \text{s.t. } e(y(u_k), u_k) = 0, \quad ||s_u|| \leq \Delta_k. \end{aligned} \tag{6.18}$$

The most direct way to find a minimizer of (6.18) is to search only in the direction of steepest descent bounded by the trust region. Solving

$$\min_{\sigma} m_k(-\sigma \hat{J}'(u_k)), \quad \text{s.t. } \sigma||\hat{J}'(u_k)|| \leq \Delta_k. \tag{6.19}$$

gives the Cauchy direction $s_u = -\bar{\sigma}\hat{J}'(u_k)$ with

$$\bar{\sigma} = \begin{cases} \frac{\Delta_k}{||\hat{J}'(u_k)||_U} & \text{if } \langle \hat{J}''(u_k)\hat{J}'(u_k), \hat{J}'(u_k)\rangle_{U^*,U} \leq 0, \\ \min\left\{ \frac{||\hat{J}'(u_k)||_U^2}{\langle \hat{J}''(u_k)\hat{J}'(u_k), \hat{J}'(u_k)\rangle_{U^*,U}}, \frac{\Delta_k}{||\hat{J}'(u_k)||_U} \right\} & \text{if } \langle \hat{J}''(u_k)\hat{J}'(u_k), \hat{J}'(u_k)\rangle_{U^*,U} > 0, \end{cases}$$

and the Cauchy point $u_k - \bar{\sigma}\hat{J}'(u_k)$.

Since using the Cauchy direction already guarantees global convergence, any solution of the quadratic trust region problem which gives at least a fraction of the Cauchy decrease ensures global convergence. One could use the Cauchy direction as control update. We would get essentially a steepest descent method with the step size restricted by the trust region. This method is called unidirectional trust region method. It suffers from the same problems as steepest descent namely only linear convergence.

It is possible, that the control constraints become active for some or even all parts of the control iterate. If we would know exactly the active and inactive parts of the control constraints for the optimal control, we could restrict our problem to the inactive set and treat it as an unconstrained problem. However we only have the active and inactive set of the current control iterate u_k available. [60] solves this problem by overestimating the active set and solving the Newton problem for the control update only on the underestimated inactive part.

We therefore minimize (6.18) by solving the linear subproblems (6.16) on the underestimated inactive part. Using a direct method is very expensive for PDE-constrained optimization problems. Consequently we apply an iterative method. To distinguish between the optimization loop and the linear solver, we follow the convention in [25] and call the optimization iteration the *outer iteration* and the iteration to solve the linear subproblems the *inner iteration*.

Since in a first-optimize-then-discretize approach the discrete Hessian $\mathbf{H}^h \in \mathbb{R}^{n_u \times n_u}$ is not necessarily symmetric, we use the BiCGStab algorithm [56] to compute the trial step s_u. The inner iteration stops if at the current iterate $s_u^{(i)}$ with residual $r^{(i)}$ defined by $\hat{J}''(u_k)s_u^{(i)} = -\hat{J}'(u_k) + r^{(i)}$ the following inequality holds

$$\frac{||r^{(i)}||_U}{||\hat{J}'(u_k)||_U} \leq \mu_{term}$$

with the termination tolerance μ_{term}. Also we only allow a certain maximal number of iterations dependent on the considered problem.

After computing the control update s_u, an Armijo line search gives then a scaled trial step $s_{u,proj} := \mathbb{P}_{U_{ad}-u_k}(\sigma_k s_u)$ such that

$$m_k(0) - m_k(s_{u,proj}) > \kappa_4 \langle \hat{J}'(u_k), s_{u,proj} \rangle_{U^*,U} \quad \text{and} \quad ||s_{u,proj}||_U < \Delta_k.$$

If the generalized Cauchy decrease condition

$$m_k(0) - m_k(s_{u,proj}) \geq \kappa_1 ||\mathbb{P}_{U_{ad}-u_k}(-\hat{J}'(u_k))||_U \min\left\{\kappa_2 ||\mathbb{P}_{U_{ad}-u_k}(-\hat{J}'(u_k))||_U, \kappa_3 \Delta_k\right\}$$

is fulfilled, we accept the trial step else we use the projected negative gradient step $\mathbb{P}_{U_{ad}-u_k}(-\sigma_k \hat{J}'(u_k))$ with σ_k coming again out of an Armijo line search augmented by the trust region condition. We set the constants to $\kappa_1 = \kappa_2 = \kappa_3 = 0.01$.

Acceptance of steps

Having computed a trial step s_u with a certain trust region radius Δ_k, we now have to decide whether we accept the trial step, the trust region or both. We base this decision on the ratio of the achieved reduction in the objective function

$$\text{ared}_k := J(y_k(u_k), u_k) - J(y_{k+1}(u_{k+1}), u_{k+1}) \tag{6.20}$$

to the reduction as predicted by the quadratic model (6.17)

$$\text{pred}_k := m_k(0) - m_k(s_u). \tag{6.21}$$

Introducing three control parameters $0 < \mu_0 \leq \mu_{low} \leq \mu_{high}$, we accept the trial step and update u_k if $\frac{\text{ared}_k}{\text{pred}_k} > \mu_0$. Furthermore we adjust the trust region according to

$$\Delta_{k+1} = \begin{cases} \omega_{down}\Delta_k & \text{if } \frac{\text{ared}_k}{\text{pred}_k} < \mu_{low}, \\ \min\{\omega_{up}\Delta_k, \Delta_{max}\} & \text{if } \frac{\text{ared}_k}{\text{pred}_k} > \mu_{high}. \end{cases}$$

The adjustment parameters are chosen as $\omega_{down} = 0.5$ and $\omega_{up} = 2$. To avoid an infinite expansion of the trust region, we bound it from above by $\Delta_k \leq \Delta_{max}$.

If the control update s_u is accepted by the algorithm, the difference between exact and inexact gradient is sufficiently small and the optimization algorithm can start the next iteration step with the new control iterate $u_{k+1} = u_k + s_u$.

If the control update is not accepted, there are essentially two possible reasons. First the local quadratic problem is not a good approximation of the original problem, hence the trust region is too big. Second the inexact gradient does not approximate the exact gradient good enough. We can compare the inexact gradient to the exact gradient by the following gradient condition

$$\begin{aligned} |\langle \partial_{(y,u)} J(y_k, u_k), (s_y, s_u)\rangle_{V^*,V} - \langle \hat{J}'(u_k), s_u\rangle_{U^*,U} \\ \leq \kappa \min\left\{ ||\mathbb{P}_{U_{ad}-u_k}(-\hat{J}'(u_k))||_U, \Delta_k \right\} ||u_k||_U. \end{aligned} \tag{6.22}$$

In our computations we set $\kappa = 0.1$. When the gradient condition holds the inexact gradient is good enough and we have to decrease the trust region. Otherwise the tolerances for the state and the adjoint system have to be refined, i.e., we use a refined approximation of the control problem. There is always a refinement level such that (6.22) and (6.9)- (6.11) are satisfied [60].

6.1.4. The role of the time integrator in the multilevel optimization algorithm

There are essentially three points where we have to solve PDE systems during the optimization. When computing the objective function, we need to solve a state system (2.3) and for the reduced gradient an adjoint system (2.5). Finally when solving (6.16)

we need to solve a linearized state system (2.10) and the second adjoint system (2.11) to compute the reduced Hessian applied to a direction.

In our computations the time mesh of an adaptive solution of the state system is also used for the solution of the adjoint, linearized state and second adjoint systems. This is done to avoid computationally expensive interpolations of the state solutions for the adjoint and linearized state systems, and of the state, adjoint and linearized state solutions for the second adjoint system. Within the optimal control with parabolic problems the adjoint, linearized state and second adjoint are linear parabolic equations. Here we can use the fourth order method *rodas*4*p* [55] which shows no order reduction for linear parabolic systems. Hence the overall computational cost of an optimization step is controlled by the numerical solution of the state system.

Using high order integrators should be very beneficial, as they should need fewer time steps as lower order methods to satisfy a given tolerance for the time error. However, Runge-Kutta and Rosenbrock methods are known to suffer from order reduction when applied to stiff ODEs and within the discretization of PDEs [34,35,39]. Order reduction can be avoided by using methods which fulfill additional order conditions. *rodas*4*p* [55], for example, fulfills the additional order conditions for order 4 for linear parabolic systems. In [30] the third order Rosenbrock method *ros*3*p* was designed which shows no order reduction even for nonlinear parabolic problems. In [29] the third order Rosenbrock method *ros*3*pl* was presented. It is L-stable and avoids order reduction by fulfilling additional order conditions up to order three.

Constructing Runge-Kutta or Rosenbrock methods of order higher than three, which do not suffer from order reduction when applied to nonlinear problems, is difficult. Here peer methods are an attractive alternative. Because of their multistep structure, peer methods allow for a very systematic computation of order conditions and it is not difficult to construct higher order methods. A problem when constructing higher order peer methods is the stability, but we saw already in Section 3.1, that there are stable high order methods available. Peer methods are shown to not suffer from order reduction [40]. Therefore, we expect peer methods to give good results, when used within the optimization of stiff ODEs or PDEs. In the following we present some numerical test examples regarding the performance of peer methods within the multilevel optimization.

6.2. Numerical results for PDE-constrained test problems

In this Section we compare the performance of the peer methods *peer*3*pos*, *peer*4*pos* and *peer*5*pos* within a multilevel optimization to that of the Rosenbrock methods *ros*3*pl* and *rodas*4*p*. We consider three different PDE-constrained optimal control problems.

6.2.1. Glass cooling

Our first example is the optimal cooling of glass presented in [11]. After heating and shaping glass it is an important step to cool down the material to room temperature. This is done by placing the hot glass in a preheated furnace and subsequently reducing the

temperature of the furnace until everything reached the desired temperature. The choice of the temperature distribution over time of the furnace leads to an optimal control problem.

First we derive a suitable model of the cooling process. Glass is heated to high temperatures and hence the direction- and frequency-dependent thermal radiation field and the spectral radiative properties of semi-transparent glass have to be considered. Since the energy transport by radiation is a lot faster than the transport by diffusion, we model the radiation in a quasi-static manner. The computational domain Ω is an unit square, i.e., $\Omega = (0,1) \times (0,1)$. Our control $u(t)$, the furnace temperature, enters through the boundary of the workpiece. The control is bounded from above by u_{upper} and from below by u_{lower}, i.e., $u_{lower} \leq u(t) \leq u_{upper}$ for all $t \in [0, t_e]$. We model the glass temperature $T(x,t)$ by a heat equation with a non-linear exchange of energy between the radiation field and the temperature.

In our example we consider an SP_1 approximation of the frequency spectrum of the radiation field, i.e., the continuous frequency spectrum is discretized by a single band. The resulting model is called grey scale model, because it neglects the dependency on wavelength or frequency. Let ϵ denote the optical thickness coefficient. The heat conduction coefficient is denoted by k_c and the convective heat transfer coefficient by h_c. Furthermore we have the radiated energy coefficient a_2. We consider the model in a dimensionless form, see [26]. Additionally we have two dimensionless constants κ and σ. The state system for the state $y := (T, \phi)^T$ is then given by the following set of equations:

$$\partial_t T - k_c \Delta T - \frac{1}{3(\kappa+\sigma)} \Delta \phi = 0 \tag{6.23}$$

$$-\frac{\epsilon^2}{3(\kappa+\sigma)} \Delta \phi = -\kappa\phi + 4\pi\kappa a_2 T^4 \tag{6.24}$$

with boundary and initial conditions

$$k_c n \cdot \nabla T + \frac{1}{3(\kappa+\sigma)} n \cdot \nabla \phi = \frac{h_c}{\epsilon}(u - T) + \frac{1}{2\epsilon}\left(4\pi a_2 u^4 - \phi\right) \tag{6.25}$$

$$\frac{\epsilon^2}{3(\kappa+\sigma)} n \cdot \nabla \phi = \frac{\epsilon}{2}(4\pi a_2 u^4 - \phi) \tag{6.26}$$

$$T(0,x) = T_0(x). \tag{6.27}$$

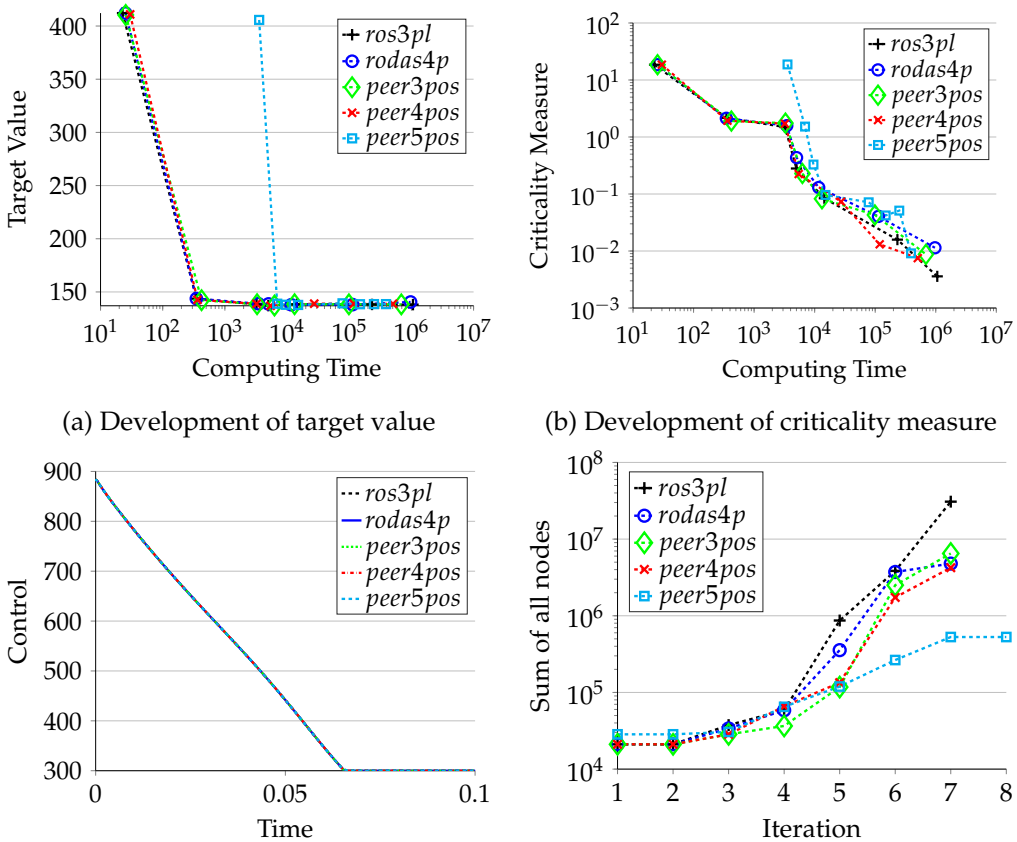
We now want to optimize three quantities. First the glass piece should cool down as close as possible to a given spatially uniform temperature profile $T_d(t)$. Second the final temperature $T(t_e, \cdot)$ should be as close as possible to a given temperature $T_d(t_e)$. And third the control $u(t)$ should follow a desired spatially uniform control profile $u_d(t)$ as close as possible. This leads to the following objective function:

$$\begin{aligned} J(T,u) = &\frac{1}{2}\int_0^{t_e} ||T(t,\cdot) - T_d(t)||^2_{L^2(\Omega)}\, dt + \frac{0.1}{2}||T(t_e,\cdot) - T_d(t_e)||^2_{L^2(\Omega)}\, dt \\ &+ \frac{0.1}{2}\int_0^{t_e} |u(t) - u_d(t)|^2\, dt. \end{aligned} \tag{6.28}$$

Table 6.1.: Problem and model specific quantities

u_{lower}	lower bound for the control	300	k_c	conductivity coefficient	1.0
u_{upper}	upper bound for the control	900	h_c	convection coefficient	$1.0e{-}3$
t_e	final time	$1.0e{-}1$	ϵ	optical thickness coefficient	$5.0e{-}1$
$T_0(x)$	initial glass temperature	$9.0e+2$	κ	dimensionless constant	10.0
$T_d(t)$	desired glass temperature	$T_0 \cdot e^{\frac{-\log(\frac{T_0}{300})t}{t_e}}$	a_2	radiated energy coefficient	$1.8e{-}8$
$u_0(t)$	initial control	$T_d(t)$	σ	dimensionless constant	1.0
$u_d(t)$	desired control	$T_d(t)$			

The used values for problem and model specific quantities are shown in Table 6.1. As linear solver we use BiCGStab with a maximal number of 50 iterations. As an optimization solver we use a projected SQP method. For the refinement of tolerances we use $C_{s,space} = C_{s,time} = 5.0e-3$, $C_{a,space} = C_{a,time} = 5.0e-2$. The initial tolerance for all errors is $5.0e-2$.



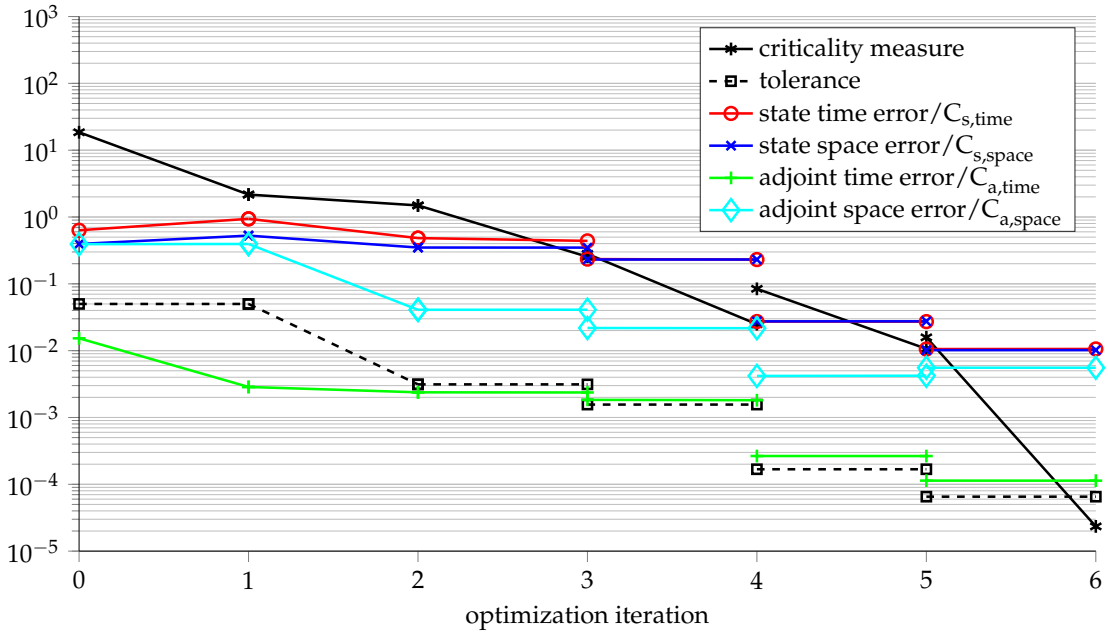
(a) Development of target value

(b) Development of criticality measure

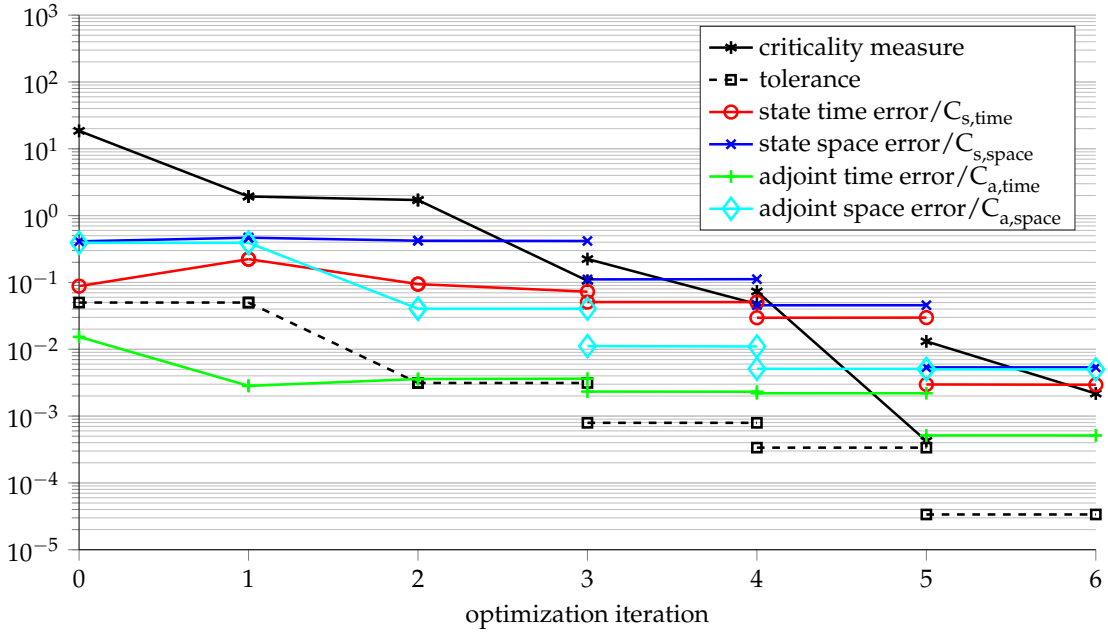
(c) Computed optimal control

(d) Degrees of freedom for the state

Figure 6.1.: Multilevel optimization of grey-scale problem



(a) Multilevel strategy using *ros3pl* as time integrator.



(b) Multilevel strategy using *peer4pos* as time integrator.

Figure 6.2.: Comparison of the multilevel strategy used within the optimization of the grey-scale problem. When a scaled error estimate cuts the criticality measure, the local error tolerance is refined.

We use different time integrators to compute the state system. To make the results more comparable, we decided to use always the same integrator for the adjoint, linearized state and second adjoint sytems. Since these systems are all linear, the fourth order method *rodas*4*p* can be applied without order reduction. In the computations we always used *rodas*4*p* as time integrator for these systems regardless of the integrator used for the state system. Hence the only difference between the shown optimization runs comes from the used time integrator for the state system.

In Figure 6.1a the development of the target value is shown. All optimization runs converge to the same value. The criticality measure at the control iterates is shown in Figure 6.1b. There is not much difference between the integrators. *peer*5*pos* needed some time to complete the first iteration step due to several refinements in the beginning.

All optimization runs computed the same optimal control as can be seen in Figure 6.1c. The number of degrees of freedom used in the state system at every control iteration is shown in Figure 6.1d. For later optimization iterations the state discretization using *peer*5*pos* need the smallest number of degrees of freedom, while for the other time integrators the needed number of degrees of freedom is quite similar. In the last iteration of *ros*3*pl* the discretization of the state needs more degrees of freedom due to the relatively small criticality measure.

In Figure 6.2 the multilevel strategy of the optimization algorithm is shown for the time integrators *ros*3*pl* and *peer*4*pos*. The criticality measure, the error tolerance and the four different global error estimations divided by the corresponding refinement constant are plotted for every optimization iteration. If one or several refinements took place the values before the first refinement and after the last refinement are shown. Whenever one of the scaled error estimates exceeds the criticality measure, the local error tolerance has to be refined. This leads usually to an increase of the criticality measure and a decrease of the error norms.

For both methods, *ros*3*pl* and *peer*4*pos*, the multilevel strategies used are very similar. The state time error is lower when using *peer*4*pos*. This leads to a smaller number of time steps used especially for fine tolerances. Since we use the time mesh of the state problem and we always take *rodas*4*p* as time integrator for the adjoint problem, the adjoint time error is larger for *peer*4*pos* than for *ros*3*pl*. The state errors dominate the adjoint errors. Therefore, the larger adjoint time error had no influence on the multilevel strategy.

6.2.2. Flame propagation through a cooled channel

We now look at an optimal control problem for the flame problem already considered in Section 5.5.3. We assume, that we can control the temperature at the boundary of the obstacle. The objective function is given by

$$J(T, Y, u) = \frac{0.1}{2} \iint_{\Omega} (T(x, y, t_{end}) - T_D(x, y))^2 \, dx \, dy + \frac{0.1}{2} \int_0^{t_{end}} u^2 \, dt.$$

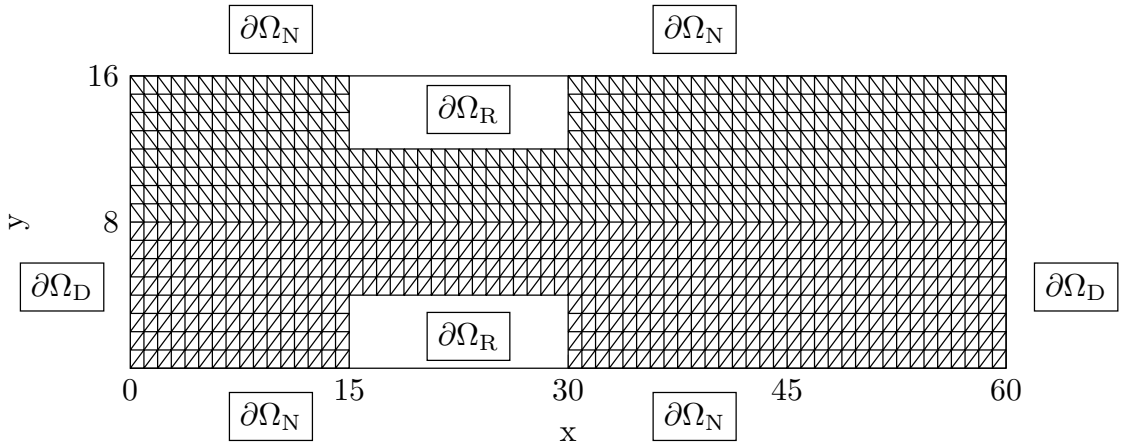


Figure 6.3.: Initial mesh for Flame Problem

Here T_D denotes a desired distribution of the temperature and is given by

$$T_D(x,y) = \begin{cases} 1 & \text{if } x \leq 30.0 \\ 0 & \text{else.} \end{cases}$$

Hence our objective is to control the burning process in such a way, that everything left of and inside the obstacle is burned, but nothing is burned right of the obstacle. We add a Tikhonov regularization term for the control, which penalizes the energy consumption of the process. The final time t_{end} is set to 20. The state system is then given by

$$\partial_t T - \Delta T = \omega(T,Y), \qquad \text{in } \Omega \times I, \tag{6.29}$$

$$\partial_t Y - \frac{1}{Le}\Delta Y = -\omega(T,Y), \qquad \text{in } \Omega \times I, \tag{6.30}$$

$$T(\cdot,0) = T_0(\cdot) \qquad \text{on } \Omega, \tag{6.31}$$

$$Y(\cdot,0) = Y_0(\cdot) \qquad \text{on } \Omega, \tag{6.32}$$

with the following boundary conditions

$$\begin{aligned} T &= 1, & Y &= 0, & &\text{on } \partial\Omega_D \times I, \\ \partial_n T &= 0, & \partial_n Y &= 0, & &\text{on } \partial\Omega_N \times I, \\ \partial_n T &= -\kappa(T - u(t)), & \partial_n Y &= 0, & &\text{on } \partial\Omega_R \times I. \end{aligned} \tag{6.33}$$

Different to the setting in Section 5.5.3 we added a control u, which enters the system via the Robin boundary condition on $\partial\Omega_R$. The Lewis number Le is set to 1 and the heat loss parameter κ to 0.1. The initial condition is again

$$T_0(x,y) = \begin{cases} 1 & \text{for } x \leq x_0 \\ \exp(-(x - x_0)) & \text{for } x > x_0 \end{cases}, \tag{6.34}$$

$$Y_0(x,y) = \begin{cases} 0 & \text{for } x \leq x_0 \\ 1 - \exp(-Le(x - x_0)) & \text{for } x > x_0 \end{cases}, \tag{6.35}$$

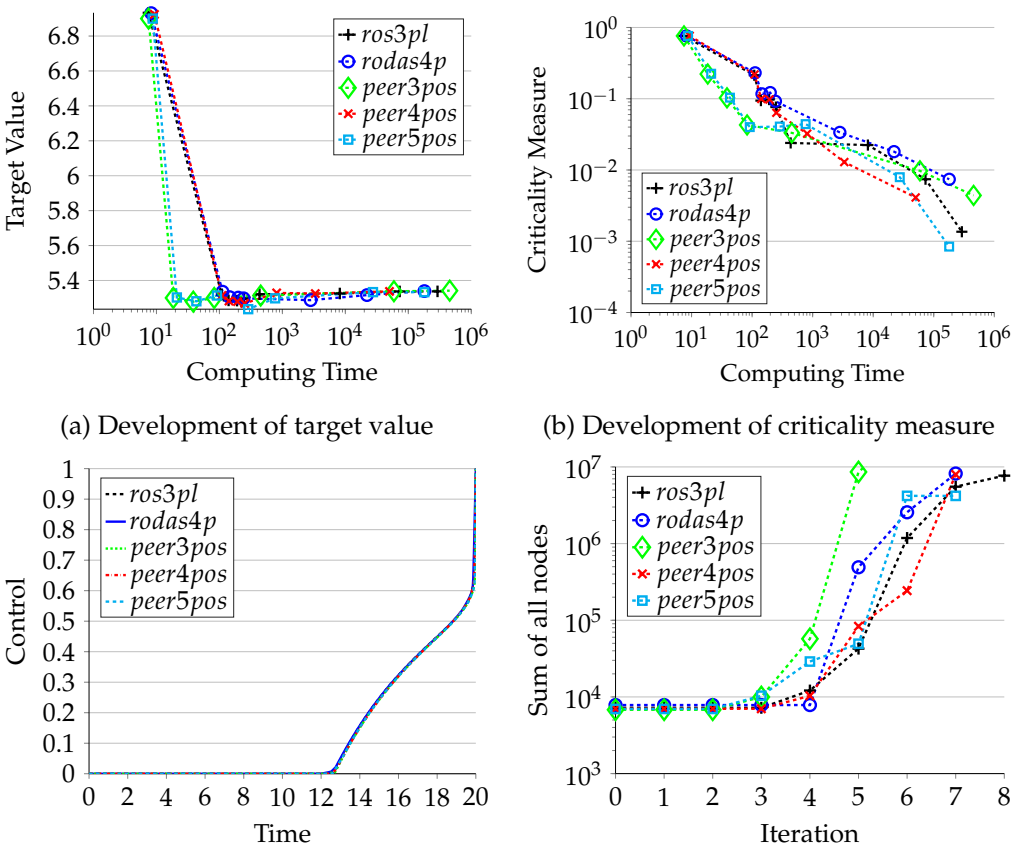
where $x_0 = 9$. Finally the source function $\omega(T, Y)$ is given by

$$\omega(T, Y) = \frac{\beta^2}{2Le} Y \exp\left(\frac{-\beta(1-T)}{1-\alpha(1-T)}\right).$$

The heat release parameter α is set to 0.8 and β is set to 1. The choice of β is different to Section 5.5.3, where it was set to 10. For $\beta = 10$ the systems are much harder to solve than for $\beta = 1$. Especially the adjoint and second adjoint system require very fine discretizations even for the initial tolerances. For this reason we used a smaller β, at the cost of a less sharp transition from burnt to unburnt material.

The control function is constrained by

$$0 \leq u(t) \leq 1.0.$$



(a) Development of target value

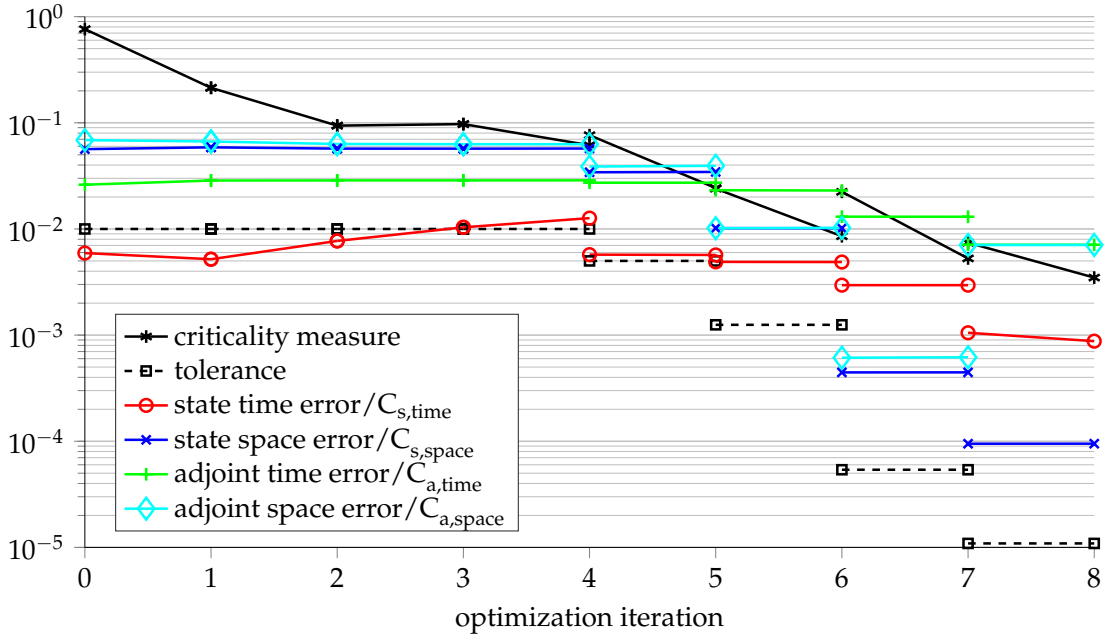
(b) Development of criticality measure

(c) Computed optimal control

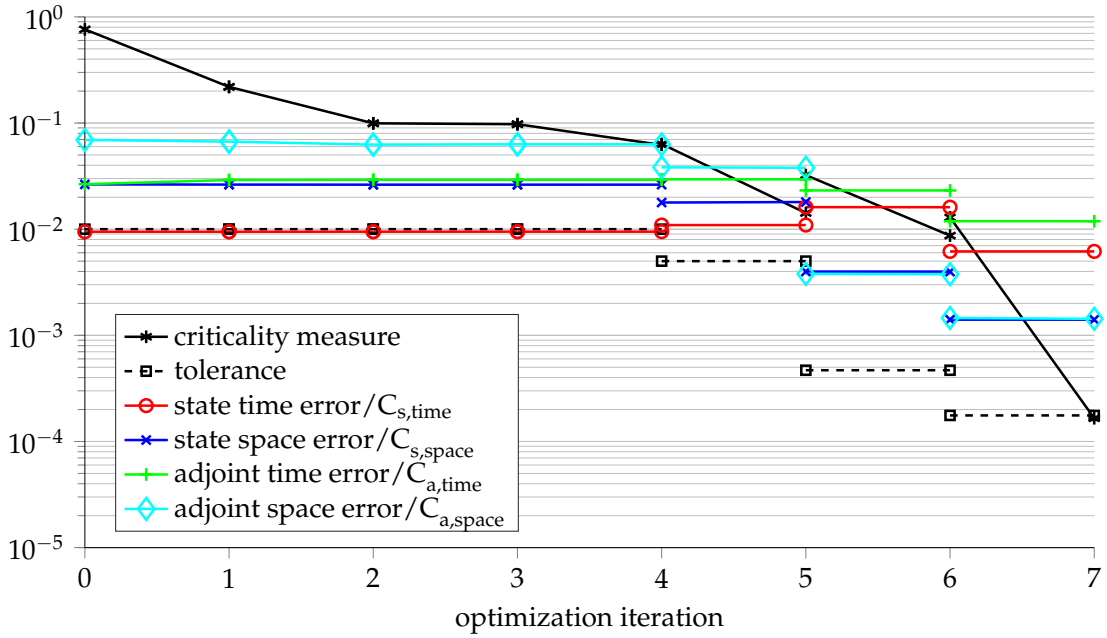
(d) Degrees of freedom for the state

Figure 6.4.: Multilevel optimization of flame problem

As for the glass cooling problem we use the time integrator *rodas4p* for the adjoint, linearized state and the second adjoint systems. Only the time integration solver for the



(a) Multilevel strategy using *ros3pl* as time integrator.



(b) Multilevel strategy using *peer4pos* as time integrator.

Figure 6.5.: Comparison of the multilevel strategy used within the optimization of the flame problem. When a scaled error estimate cuts the criticality measure, the local error tolerance is refined.

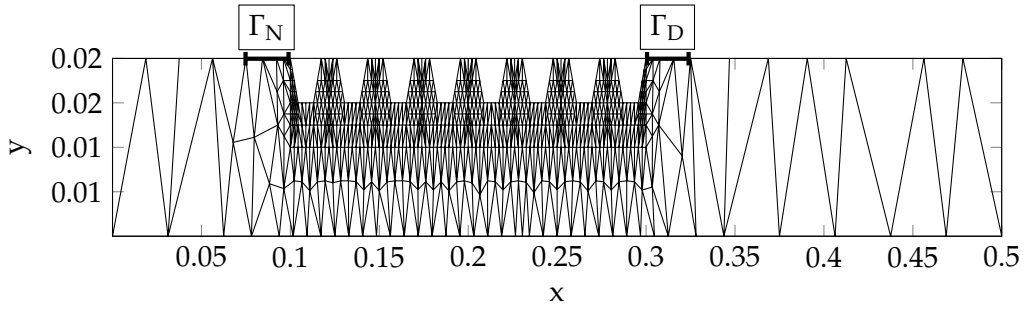


Figure 6.6.: Initial mesh for Thermistor Problem

state system is changed. The initial control is $u_0(t) = 0.2$. As initial tolerance we set for all errors $1.0e-2$. The refinement criterion constants are $C_{s,space} = C_{a,space} = 1.0e-1$ and $C_{s,time} = C_{a,time} = 1.0e-2$.

The results of the different optimization runs are shown in Figure 6.4. All methods compute almost the same optimal control as can be seen in Figure 6.4c. The flame enters the channel at about $t = 13$. As a consequence the optimal control is set to 0 before this time to save energy. In Figure 6.4a the development of the target value is shown. As all optimization runs yield the same optimal control, it is no surprise that the target values also converge to the same value.

The criticality measure at the control iterates is shown in Figure 6.4b. All optimization runs decrease the criticality measure similarly. Concerning the computing time *peer*5*pos* and *peer*4*pos* have slight advantages compared to *ros*3*pl*. *rodas*4*p* and *peer*3*pos* have the worst performance of the considered methods.

Looking at the used degrees of freedom for the state system in Figure 6.4d, we see that *peer*3*pos* as a second order method needs finer meshes to fulfill the given tolerances than the other methods. The other methods need a similar number of degrees of freedom.

The details of the multilevel strategy for *ros*3*pl* and *peer*4*pos* are shown in Figure 6.5. The dominating scaled error estimate is now the adjoint time error. The scaled state time error influences the multilevel strategy only for *peer*4*pos* in the last two iterations. Note, that the last plotted value of the criticality measure is the value before any refinements. Therefore, it is different to the last criticality measure in Figure 6.4b, where the values after refinement are shown.

6.2.3. Steel hardening

In [23] an optimal control problem for the hardening of a steel rack is presented. The rack is heated by inducing a direct current on a part Γ_N on the boundary of the rack Ω. The induced current flows to the anode at another part of the boundary Γ_D, where it is absorbed.

We want to minimize the objective function

$$J(y,u) := \frac{1}{2}\int_\Omega (\theta(x,T) - \theta_d)^2\,dx + \frac{0.1}{2}\int_0^T \int_{\Gamma_N} |u(t)|^2\,ds\,dt$$

where $y = (\theta, \varphi)$. The variable $\theta(x,t)$ stands for the temperature of the rack at point x and at time t. Similarly $\varphi(x,t)$ denotes the electrical potential and the control $u(t)$ describes the current induced at the Neumann boundary Γ_N. Note, that it only depends on time and is chosen constant in space. By the first part of the objective function we reinforce an temperature distribution in the rack which is close to a desired temperature distribution θ_D. Furthermore we add a Tikhonov regularization which penalizes large values of the control u. Physically spoken we penalize the energy consumption of the heating process.

Since we can not realistically create an arbitrary large current, we bound the allowed control, i.e. we demand

$$\forall t \in [0,T]: \quad 0 \leq u(t) \leq u_{max}(t).$$

Given a control u the state $y = (\theta, \varphi)$ has to satisfy the system

$$\begin{aligned}
C_p \rho \partial_t \theta - \nabla \cdot (\kappa \nabla \theta) &= \sigma(\theta) |\nabla \varphi|^2 && \text{in } (0,T) \times \Omega, \\
n^\intercal \kappa \nabla \theta &= \alpha(\theta_l - \theta) && \text{on } (0,T) \times \Gamma, \\
\theta(x,0) &= \theta_0 && \text{in } \Omega, \\
-\nabla \cdot (\sigma(\theta) \nabla \varphi) &= 0 && \text{in } (0,T) \times \Omega, \\
n^\intercal \sigma(\theta) \nabla \varphi &= C_u u && \text{on } (0,T) \times \Gamma_N, \\
\varphi &= 0 && \text{on } (0,T) \times \Gamma_D, \\
n^\intercal \sigma(\theta) \nabla \varphi &= 0 && \text{on } (0,T) \times \Gamma_R.
\end{aligned}$$

The system couples an instationary heat equation with a quasilinear potential equation. The boundary condition for the temperature models a linear heat exchange with the ambient temperature θ_l with heat transfer coefficient α. For the potential equation we have mixed boundary conditions. The current inflow is modeled via a Neumann boundary condition at $\Gamma_N = \{(x, 0.02) : 0.75 \leq x \leq 1\}$ while the anode on the boundary part $\Gamma_D = \{(x, 0.02) : 1 \leq x \leq 1.25\}$ is modeled by a Dirichlet condition. We assume isolation on the other parts $\Gamma_R = \Gamma \setminus (\Gamma_N \cup \Gamma_D)$ and hence set a no-flow boundary condition.

The electric conductivity σ depends nonlinearly on θ and is modeled by

$$\sigma(\theta) = \frac{1}{a + b\theta + c\theta^2 + d\theta^3}$$

with the parameters

$$a = 4.9659E-7, \quad b = 8.4121E-10, \quad c = -3.7245E-13, \quad d = 6.1960E-17.$$

Furthermore we set the heat capacity $C_p = 470 \frac{J}{kg\,K}$, the density $\rho = 7900 \frac{kg}{m^3}$, the heat conduction coefficient $\kappa = 50 \frac{W}{m\,K}$ and the heat transfer coefficient $\alpha = 20 \frac{W}{m^2\,K}$. The parameter C_u which is used to balance the order of magnitude of control and the reduced derivatives is set to $1.0E5$.

One important feature is the use of mixed boundary conditions for the potential equation. However this leads to some problems for an adaptive solution process. According to [27] the solution of the potential will behave locally like $r^{\frac{1}{2}}\cos(\frac{\omega}{2})$ in polar coordinates with the origin located at the transition point of the boundary. In space θ and φ belong to $H^{\frac{3}{2}}(\Omega)$ [9]. Because of the Neumann-Dirichlet singularity, we can only expect a convergence order of $\mathcal{O}(h^{\frac{1}{2}})$ in the energy norm for uniform mesh refinement [44]. This highlights the need for an adaptive mesh refinement. Unfortunately we also have an unbounded derivative behaving like $r^{-\frac{1}{2}}\cos(\frac{\omega}{2})$ in tangential direction of the domain at the transition point. This leads to an higher temperature around the transition point after refinement. While this is a drawback of the model, we can deal with it numerically. A unexpected growth of the temperature due to the spatial refinement gives larger temporal errors. Hence we benefit from time adaptivity, as the step size controller registers this growth and adapts by reducing the step size.

In Kardos the spatial error of Rosenbrock methods is measured after the computation of the first stage. So rather than estimating the spatial error of the method itself, they estimate the spatial error of an suitable scaled, implicit Euler method. This works very well in practice. Until now we measured the spatial error of the last stage for peer methods. As in Chapter 5 said, we neglect a possible nonlinear error transport through the stages. This is meaningful for a lot of problems, but this does not work well for the thermistor problem regarding its singularity. However one can follow Rosenbrock methods and estimate the error of the first stage. Unlike Rosenbrock methods we do not measure the spatial error at the new time point $t_n + \tau$ but at $t_n + c_1\tau$. This is similar to the Method of Lines approaches, where the spatial mesh is also fixed before the time step using the old solution. Giving sufficiently small step sizes τ one intuitively expect this to work good enough. In numerical computations we see that measuring the spatial error in the first stage for peer methods actually leads to almost the same meshes as the spatial error estimation of Rosenbrock methods.

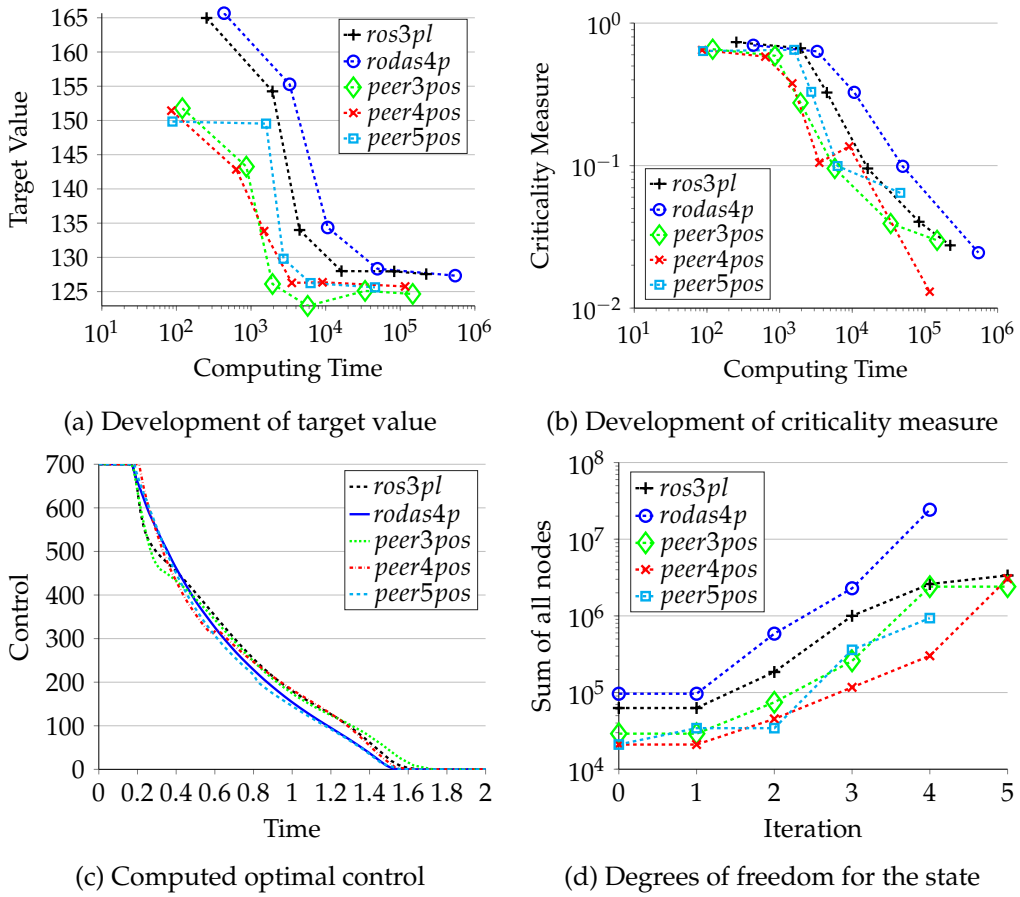
Now we look on the results of the multilevel optimization algorithm applied to the thermistor optimal control problem. The control bounds are chosen as

$$u_{min} = 0.0 \text{ and } u_{max} = 7.0E7.$$

We use a projected SQP method and the BiCGStab algorithm to solve the linear system. We restrict the maximal number of iterations to 5. The initial control is $u_0(t) = 1.5 \times 10^7 \cdot (2-t)^2$. As initial tolerance we set $1.0e-3$ and we set $C_{s,space} = C_{a,space} = 2.0e-3$ and $C_{s,time} = C_{a,time} = 1.0e-3$. The adjoint errors were always smaller than the respective state errors, hence the tolerances were only refined due to high errors of the state solution.

Again *rodas*4*p* is used for adjoint, linearized state and second adjoint systems, only the time integrator for the state system is changed. In Figure 6.7 the results of an optimization run with *ros*3*pl*, *rodas*4*p*, *peer*3*pos*, *peer*4*pos* and *peer*5*pos* as time integrators for the state are shown.

The target values at each optimization iteration are shown in Figure 6.7a. The peer methods compute different target values than the Rosenbrock methods even for very



(a) Development of target value

(b) Development of criticality measure

(c) Computed optimal control

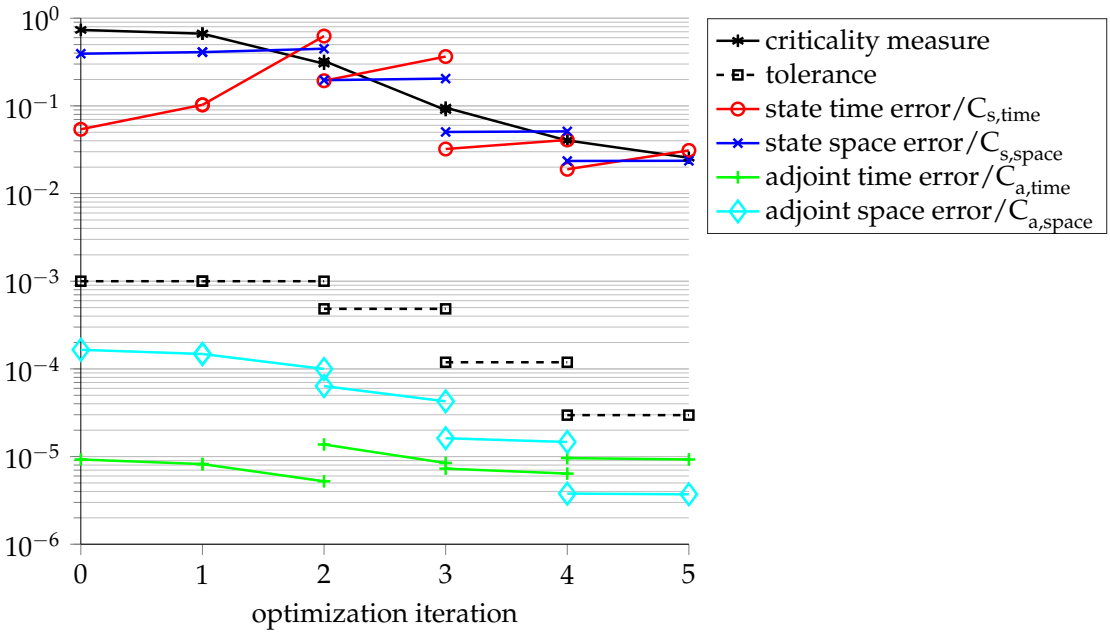
(d) Degrees of freedom for the state

Figure 6.7.: Multilevel optimization of the Thermistor problem

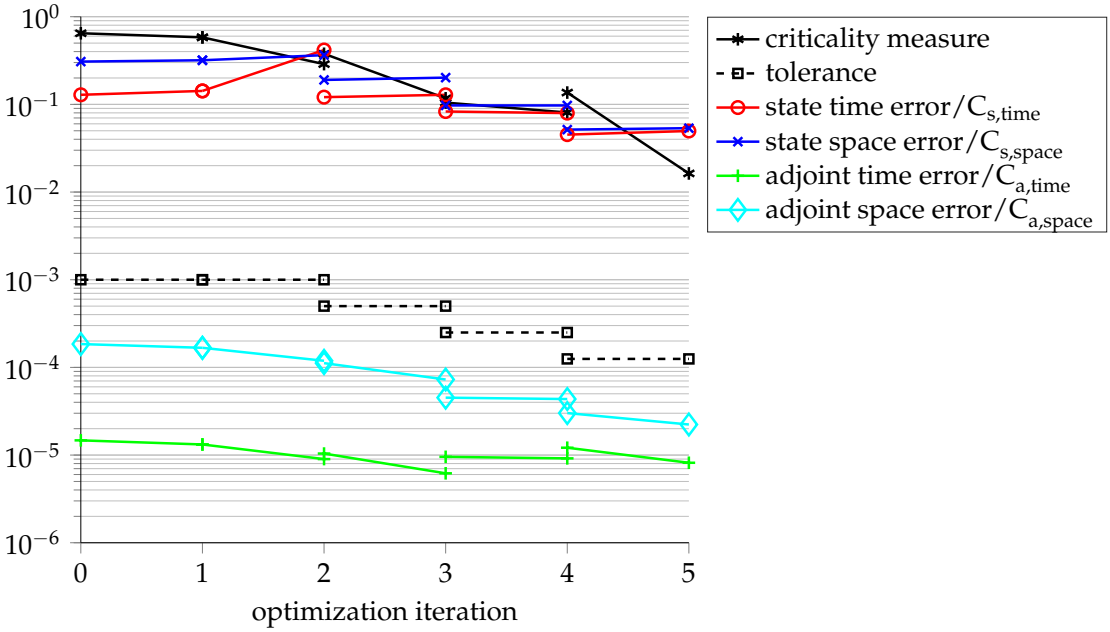
similar control iterates. The target value is very sensitive with respect to the used spatial mesh. In Figure 6.7d the number of degrees of freedom for the state system at the control iterates is shown. The Rosenbrock methods use finer meshes in the beginning than the peer methods, therefore the computed target value gets larger.

All methods converge with a similar reduction in the criticality measure as can be seen in Figure 6.7b. Since *rodas4p* uses the finest meshes throughout the optimization it is the slowest method in our comparison. The optimization runs using peer methods and *ros3pl* decrease the criticality measure similarly. *peer5pos* is a little bit slower than *peer4pos*, but the convergence stalled after 5 iterations. The criticality measure could not be reduced anymore, since it failed to refine its mesh. Hence for *peer5pos* one would have to choose other optimization parameter to ensure a refinement of the mesh after 5 optimization iterations.

The computed optimal controls are shown in Figure 6.7c. They are slightly different for every method, but they all share the same qualitative behaviour. After a strong heating phase in the beginning to time point 0.2, the induced current is reduced to zero



(a) Multilevel strategy using *ros3pl* as time integrator.



(b) Multilevel strategy using *peer4pos* as time integrator.

Figure 6.8.: Comparison of the multilevel strategy used within the optimization of the Thermistor problem. When a scaled error estimate cuts the criticality measure, the local error tolerance is refined.

at about 1.6. This is in accordance to the results obtained in [23] and [7].

A comparison of the multilevel strategies of the optimizations using *ros*3*pl* and *peer*4*pos* as time integrators is shown in Figure 6.8. Again the strategies used are very similar. The scaled state error estimates dominate the scaled adjoint error estimates, which subsequently do not influence the multilevel strategy.

6.2.4. Conclusion

Using peer methods gave comparable results to using Rosenbrock methods within the optimization. Hence, for the presented test problems, the lack of order reduction was not a big advantage. As the order is an asymptotic result, the involved error constants play an important role for large time steps. The error constants of Rosenbrock methods are better than the one of the peer methods investigated. Hence for large tolerances Rosenbrock methods might perform better as peer methods even though they have actually a reduced order. Furthermore in [2] it was noted that peer methods start to get really superior when high accuracy is required. We couple the time integration with linear finite elements in space. It is not meaningful to require very small time errors while allowing for large spatial errors, hence the tolerances for time and space should be balanced. However high accuracy requirements in space are very expensive and usually not feasible when using linear finite elements. Thus we will not reach high accuracy regions with our optimization algorithm, as the computational cost gets too high. All in all we therefore do not have an extraordinary better performance of peer methods in comparison with Rosenbrock solver. This was already observed in Section 5.5 where peer methods only showed better results for the problem of a moving flame in a cooled channel. We expect, that using peer methods, and avoiding order reduction in the time integration, starts to pay off when using higher order finite elements in space or solving problems with high accuracy requirements.

7. Conclusions

The aim of this thesis was to analyze implicit and linearly implicit peer methods in the context of optimal control with ordinary differential and partial differential equations. We first looked at the interchangeability of the first-discretize-then-optimize and first-optimize-then-discretize approaches. We found out, that **the discrete adjoint of implicit peer methods is not a consistent discretization of the adjoint of the infinite dimensional problem.** Furthermore if we look only on the interior time domain, a consistency order $p = s - 1$ for the discrete adjoint method leads to an equivalent of a BDF method. Hence, BDF methods are optimal with respect to the order of consistency. Due to these negative results, implicit peer methods are not recommended to be used in a first-discretize-then-optimize approach.

For this reason we switched our attention to a first-optimize-then-discretize environment. In preparation of the application of linearly implicit peer methods within a multilevel optimization, **we derived and implemented linearly implicit peer methods with adaptive time step control and adaptive spatial meshes within the software Kardos.** We showed that, similar to Rosenbrock methods, a spatial error estimator based on the hierarchical basis approach is robust up to some perturbations. In numerical experiments the presented methods gave satisfying and competitive results compared to Rosenbrock methods.

After this preliminary work **we sucessfully used linearly implicit peer methods within multilevel optimization for three nontrivial test problems.** For the first test problem, the optimal cooling of class, using peer methods as time integrator for the arising state problems gave no significant advantage over using the well established Rosenbrock solvers *ros*3*pl* and *rodas*4*p*. Within the optimization of the propagation of a flame in a cooled channel, we could see that using *peer*4*pos* and *peer*5*pos* gave slightly better results than the Rosenbrock methods. For the last problem, the optimal heating of a steel rack, *peer*4*pos* was the most efficient method.

For the considered test problems, using linearly implicit peer methods did not improve the performance of the multilevel optimization significantly. One reason is that we solve the PDE systems with rather low accuracy requirements. Therefore it would be interesting to couple the peer methods with higher order finite element methods in space.

Another interesting topic would be to consider goal-oriented error estimation [4]. In the context of optimal control problem, this approach is attractive because normally we are not so interested in the state system itself but rather in the objective function. Hence it makes sense to refine the discretization according to the needs of the objective function.

A. Coefficients of peer methods

A.1. *peer3pos*

$c_1 = 4.385371847140350e-01$	$c_2 = 8.743710492192502e-01$
$c_3 = 1.000000000000000e+00$	
$A_{11} = 1.869928069686809e-01$	$A_{12} = 0.000000000000000e+00$
$A_{13} = 0.000000000000000e+00$	
$A_{21} = 4.358338645052150e-01$	$A_{22} = 1.869928069686809e-01$
$A_{23} = 0.000000000000000e+00$	
$A_{31} = 4.805420905198220e-01$	$A_{32} = 8.092072476614261e-02$
$A_{33} = 1.869928069686809e-01$	
$A^0_{11} = 0.000000000000000e+00$	$A^0_{12} = 0.000000000000000e+00$
$A^0_{13} = 0.000000000000000e+00$	
$A^0_{21} = 8.739363601379309e-01$	$A^0_{21} = 0.000000000000000e+00$
$A^0_{22} = 0.000000000000000e+00$	
$A^0_{31} = 8.589765039122383e-01$	$A^0_{32} = 1.410234960877617e-01$
$A^0_{33} = 0.000000000000000e+00$	

A.2. *peer4pos*

$c_1 = 1.661225026730741e-01$	$c_1 = 4.145497896735533e-01$
$c_3 = 7.042604619720084e-01$	$c_4 = 1.000000000000000e+00$
$A_{11} = 1.205215848722435e-01$	$A_{12} = 0.000000000000000e+00$
$A_{13} = 0.000000000000000e+00$	$A_{14} = 0.000000000000000e+00$
$A_{21} = 2.484272870004788e-01$	$A_{22} = 1.205215848722435e-01$
$A_{23} = 0.000000000000000e+00$	$A_{24} = 0.000000000000000e+00$
$A_{31} = 2.243553795746857e-01$	$A_{32} = 3.137825797242478e-01$
$A_{33} = 1.205215848722435e-01$	$A_{34} = 0.000000000000000e+00$
$A_{41} = 2.112962998724119e-01$	$A_{42} = 3.138914292536173e-01$
$A_{43} = 3.086897682008952e-01$	$A_{44} = 1.205215848722435e-01$
$A^0_{11} = 0.000000000000000e+00$	$A^0_{12} = 0.000000000000000e+00$
$A^0_{13} = 0.000000000000000e+00$	$A^0_{14} = 0.000000000000000e+00$
$A^0_{21} = 4.173897839175595e-01$	$A^0_{22} = 0.000000000000000e+00$
$A^0_{23} = 0.000000000000000e+00$	$A^0_{24} = 0.000000000000000e+00$
$A^0_{31} = 1.651295614765928e-01$	$A^0_{32} = 5.387989102421881e-01$
$A^0_{33} = 0.000000000000000e+00$	$A^0_{34} = 0.000000000000000e+00$
$A^0_{41} = 4.927828853168685e-01$	$A^0_{42} = 2.102950292666084e-01$
$A^0_{43} = 7.175121439497394e-01$	$A^0_{44} = 0.000000000000000e+00$

A.3. *peer5pos*

$c_1 = 2.068377401453823e-01$	$c_2 = 3.951241118982431e-01$
$c_3 = 6.199266734460809e-01$	$c_4 = 8.406000177315648e-01$
$c_5 = 1.000000000000000e+00$	
$A_{11} = 9.477265336778397e-02$	$A_{12} = 0.000000000000000e+00$
$A_{13} = 0.000000000000000e+00$	$A_{14} = 0.000000000000000e+00$
$A_{15} = 0.000000000000000e+00$	
$A_{21} = 1.882863717528613e-01$	$A_{22} = 9.477265336778397e-02$
$A_{23} = 0.000000000000000e+00$	$A_{24} = 0.000000000000000e+00$
$A_{25} = 0.000000000000000e+00$	
$A_{31} = 1.664873086357186e-01$	$A_{32} = 2.466016246649800e-01$
$A_{33} = 9.477265336778397e-02$	$A_{34} = 0.000000000000000e+00$
$A_{35} = 0.000000000000000e+00$	
$A_{41} = 1.510411365150738e-01$	$A_{42} = 2.590889022811269e-01$
$A_{43} = 2.236322387899817e-01$	$A_{44} = 9.477265336778397e-02$
$A_{45} = 0.000000000000000e+00$	
$A_{51} = 1.531895778100870e-01$	$A_{52} = 2.234013037887999e-01$
$A_{53} = 2.999378263874698e-01$	$A_{54} = 1.166335518682616e-01$
$A_{55} = 9.477265336778397e-02$	
$A^0_{11} = 0.000000000000000e+00$	$A^0_{12} = 0.000000000000000e+00$
$A^0_{13} = 0.000000000000000e+00$	$A^0_{14} = 0.000000000000000e+00$
$A^0_{15} = 0.000000000000000e+00$	
$A^0_{22} = 3.944355830316005e-01$	$A^0_{23} = 0.000000000000000e+00$
$A^0_{24} = 0.000000000000000e+00$	$A^0_{25} = 0.000000000000000e+00$
$A^0_{21} = 0.000000000000000e+00$	
$A^0_{31} = 2.687561117109394e-01$	$A^0_{32} = 3.508845549385570e-01$
$A^0_{33} = 0.000000000000000e+00$	$A^0_{34} = 0.000000000000000e+00$
$A^0_{35} = 0.000000000000000e+00$	
$A^0_{41} = 5.837572805490611e-01$	$A^0_{42} = -3.261779079210321e-01$
$A^0_{43} = 5.830218812575633e-01$	$A^0_{44} = 0.000000000000000e+00$
$A^0_{45} = 0.000000000000000e+00$	
$A^0_{51} = 5.143425950232470e-01$	$A^0_{52} = -1.045955037674921e-01$
$A^0_{53} = 2.774411211068372e-01$	$A^0_{54} = 3.128117876374074e-01$
$A^0_{55} = 0.000000000000000e+00$	

B. Coefficients of Rosenbrock methods

A Rosenbrock method is given by

$$\begin{aligned}\left(\frac{I}{\tau\gamma} - F_y(t_{n-1}, y_{n-1})\right) k_{ni} =& F\left(t_{n-1} + \alpha_i\tau_n, y_{n-1} + \sum_{j=1}^{i-1} a_{ij}u_{nj}\right) \\ &+ \gamma_i\tau_n F_t(t_{n-1}, y_{n-1}) + \sum_{j=1}^{i-1} \frac{c_{ij}}{\tau_n} u_{nj}, \quad i = 1, \ldots, s, \\ y_n =& y_{n-1} + \sum_{j=1}^{s} m_j u_{nj}.\end{aligned}$$

An embedded solution $\hat{y}_n$ of lower order is computed by

$$\hat{y}_n = y_{n-1} + \sum_{j=1}^{s} \hat{m}_j u_{nj}.$$

B.1. *ros*2

$\gamma =$	1.7071067811865474
$a_{21} =$	0.5857864376269049
$c_{21} =$	1.1715728752538099
$\alpha_1 =$	0.0000000000000000
$\alpha_2 =$	1.0000000000000000
$\gamma_1 =$	1.7071067811865474
$\gamma_2 =$	−1.7071067811865474
$m_1 =$	0.8786796564403574
$m_2 =$	0.2928932188134524
$\hat{m}_1 =$	0.5857864376269049

B.2. *ros3pl*

$\gamma =$	$4.358665215084590000102927193070 81e-01$
$a_{2,1} =$	$1.147140180139520858381210755005 95e+00$
$a_{3,1} =$	$2.463070773030053223565105646741 81e+00$
$a_{3,2} =$	$1.147140180139520858381210755005 95e+00$
$a_{4,1} =$	$2.463070773030053223565105646741 81e+00$
$a_{4,2} =$	$1.147140180139520858381210755005 95e+00$
$a_{4,3} =$	$0.000000000000000000000000000000 00e+00$
$c_{2,1} =$	$2.631861185781064730367789783471 72e+00$
$c_{3,1} =$	$1.302364158113094747817525820288 15e+00$
$c_{3,2} =$	$-2.769432022251303537191402526218 55e+00$
$c_{4,1} =$	$1.552568958732399758185770133334 10e+00$
$c_{4,2} =$	$-2.587743501215152911116082479203 73e+00$
$c_{4,3} =$	$1.416993298352020490857121337580 08e+00$
$\alpha_0 =$	$0.000000000000000000000000000000 00e+00$
$\alpha_1 =$	$5.000000000000000000000000000000 00e-01$
$\alpha_2 =$	$1.000000000000000000000000000000 00e+00$
$\alpha_3 =$	$1.000000000000000000000000000000 00e+00$
$\gamma_0 =$	$4.358665215084590000102927193070 81e-01$
$\gamma_1 =$	$-6.413347849154099998970728069291 88e-02$
$\gamma_2 =$	$1.110281725125051094035539406279 51e-01$
$\gamma_3 =$	$-3.339342691255353656742954626679 42e-17$
$m_0 =$	$2.463070773030053311385481618067 67e+00$
$m_1 =$	$1.147140180139520935251144784228 22e+00$
$m_2 =$	$0.000000000000000000000000000000 00e+00$
$m_3 =$	$1.000000000000000000000000000000 00e+00$
$\hat{m}_0 =$	$2.346947683513665235772657768897 89e+00$
$\hat{m}_1 =$	$4.565305694518950809261334522481 95e-01$
$\hat{m}_2 =$	$5.694924394549456833180926806914 36e-02$
$\hat{m}_3 =$	$7.386849361662243750275939901950 30e-01$

B.3. *rodas4p*

$\gamma = 2.50000000000000000e-01$	
$a_{2,1} = 3.00000000000000000e+00$	$a_{3,1} = 1.83103679348675906e+00$
$a_{3,2} = 4.95518396743379520e-01$	$a_{4,1} = 2.30437658269266898e+00$
$a_{4,2} = -5.24927524574300139e-02$	$a_{4,3} = -1.17679876183278200e+00$
$a_{5,1} = -7.17045496242302545e+00$	$a_{5,2} = -4.74163667148178555e+00$
$a_{5,3} = -1.63100263133097102e+01$	$a_{5,4} = -1.06200404411140100e+00$
$a_{6,1} = -7.17045496242302497e+00$	$a_{6,2} = -4.74163667148178539e+00$
$a_{6,3} = -1.63100263133097102e+01$	$a_{6,4} = -1.06200404411140100e+00$
$a_{6,5} = 1.00000000000000000e+00$	
$c_{2,1} = 1.20000000000000000e+01$	$c_{3,1} = 8.79179517394703520e+00$
$c_{3,2} = 2.20786558697351808e+00$	$c_{4,1} = -1.08179305685715297e+01$
$c_{4,2} = -6.78027061142826596e+00$	$c_{4,3} = -1.95348594464240992e+01$
$c_{5,1} = -3.41909500674967680e+01$	$c_{5,2} = -1.54967115372596328e+01$
$c_{5,3} = -5.47476087596413036e+01$	$c_{5,4} = -1.41600539214853400e+01$
$c_{6,1} = -3.46260583093053308e+01$	$c_{6,2} = -1.53008497611447335e+01$
$c_{6,3} = -5.69995557866266747e+01$	$c_{6,4} = -1.84080700979309498e+01$
$c_{6,5} = 5.71428571428571696e+00$	
$\alpha_1 = 0.00000000000000000e+00$	$\alpha_2 = 7.50000000000000000e-01$
$\alpha_3 = 2.10000000000000000e-01$	$\alpha_4 = 6.30000000000000000e-01$
$\alpha_5 = 1.00000000000000000e+00$	$\alpha_6 = 1.00000000000000000e+00$
$\gamma_1 = 2.50000000000000000e-01$	$\gamma_2 = -5.00000000000000000e-01$
$\gamma_3 = -2.35040000000000000e-02$	$\gamma_4 = -3.62000000000000000e-02$
$\gamma_5 = 0.00000000000000000e+00$	$\gamma_6 = 0.00000000000000000e+00$
$m_1 = -7.17045496242302567e+00$	$m_2 = -4.74163667148178555e+00$
$m_3 = -1.63100263133097105e+01$	$m_4 = -1.06200404411140100e+00$
$m_5 = 1.00000000000000000e+00$	$m_6 = 1.00000000000000000e+00$
$\hat{m}_1 = -7.17045496242302497e+00$	$\hat{m}_2 = -4.74163667148178539e+00$
$\hat{m}_3 = -1.63100263133097102e+01$	$\hat{m}_4 = -1.06200404411140100e+00$
$\hat{m}_5 = 1.00000000000000000e+00$	

C. Mathematica scripts

Listing C.1: Script to solve the state and adjoint consistency equations for a given node vector c

```
DiscAdjointPeer[c_] := Module[
  {s, p, q, V0, V1, V2, V0Adj,
   V1Adj, V2Adj, a, b, A, B, AT, BT},
  (* define order p for state and order q for adjoint*)

  s = Length[c];
  p = 2;
  q = 2;
  (* ------------------------------------------------------------*)
  (* create Matrices V0, V1, V2
     for state consistency *)
  (* V2 = V0.D.F' (see Weiner) *)
  V0 = ConstantArray[1, s];
  V1 = ConstantArray[1, s];
  V2 = ConstantArray[0, s];
  For[k = 0, k < p, k += 1;
   V0 = Join[V0, c^k];
   V1 = Join[V1, (c - 1)^k];
   V2 = Join[V2, k*c^(k - 1)];
   ];
  V0 = Transpose[Partition[V0, s]];
  V1 = Transpose[Partition[V1, s]];
  V2 = Transpose[Partition[V2, s]];
  (* create Matrices V0Adj, V1Adj, V2Adj
     for adjoint consistency *)

  V0Adj = ConstantArray[1, s];
  V1Adj = ConstantArray[1, s];
  V2Adj = ConstantArray[0, s];
  For[k = 0, k < q, k += 1;
   V0Adj = Join[V0Adj, (c - 1)^k];
   V1Adj = Join[V1Adj, c^k];
   V2Adj = Join[V2Adj, k*(c - 1)^(k - 1)];
   ];
```

```
V0Adj = Transpose[Partition[V0Adj, s]];
V1Adj = Transpose[Partition[V1Adj, s]];
V2Adj = Transpose[Partition[V2Adj, s]];
(* ------------------------------------------------------------------------*)
(* A is a lower triangular matrix *)
a[i_, j_] := 0 /; j > i;
A = Array[a, {s, s}];
AT = Transpose[A];
(* B is a full matrix *)
B = Array[b, {s, s}];
BT = Transpose[B];
(* ------------------------------------------------------------------------*)
(* Solve Consistency Equations *)
linsol = Solve[
  {Table[(A.V0 - B.V1 - V2)[[i, j]], {i, 1, s},
      {j, 1, p + 1}] == 0,
   Table[(AT.V0Adj - BT.V1Adj + V2Adj)[[i, j]], {i, 1, s},
      {j, 1, q + 1}] == 0},
  Flatten[{Table[a[i, j], {i, 1, s}, {j, 1, i}],
    Table[b[i, j], {i, 1, s}, {j, 1, s}]}]
  ];
(* Print Solution if one could be found *)

If[linsol == {}, Print["No Solution could be found!"],
 Print[MatrixForm /@ {A, B} /. linsol]];

] (* End DiscAdjointPeer *)
```

Bibliography

[1] Thomas Apel and Thomas G. Flaig. Crank-Nicolson schemes for optimal control problems with evolution equations. *SIAM Journal on Numerical Analysis*, 50(3):1484–1512, 2010.

[2] Steffen Beck, Rüdiger Weiner, Helmut Podhaisky, and Bernhard A. Schmitt. Implicit peer methods for large stiff ODE systems. *Journal of Applied Mathematics and Computing*, 38(1-2):389–406, 2012.

[3] Roland Becker, Dominik Meidner, and Boris Vexler. Efficient numerical solution of parabolic optimization problems by finite element methods. *Optimisation Methods and Software*, 22(5):813–833, 2007.

[4] Roland Becker and Rolf Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica 2001*, 10:1–102, 2001.

[5] Dörte Beigel, Mario S. Mommer, Leonard Wirsching, and Hans Georg Bock. Approximation of weak adjoints by reverse automatic differentiation of BDF methods. *Numerische Mathematik*, 126(3):383–412, 2014.

[6] Frederic J. Bonnans and Julien Laurent-Varin. Computation of order conditions for symplectic partitioned Runge-Kutta schemes with application to optimal control. *Numerische Mathematik*, 103:1–10, 2006.

[7] Stefanie Bott, Debora Clever, Jens Lang, Stefan Ulbrich, Jan Carsten Ziems, and Dirk Schröder. On a fully adaptive SQP method for PDAE-constrained optimal control problems with control and state constraints. In *Trends in PDE Constrained Optimization*, pages 85–108. Springer, 2014.

[8] John C. Butcher. General linear methods. *Acta Numerica*, 15:157–256, 2006.

[9] Claire Chauvin, Pierre Saramito, and Christophe Trophime. Convergence properties and numerical simulation by an adaptive FEM of the thermistor problem. https://hal.archives-ouvertes.fr/hal-00449960, 2007.

[10] Debora Clever. *Adaptive multilevel methods for PDAE-constrained optimal control problems*. Verlag Dr. Hut, 2013.

[11] Debora Clever and Jens Lang. Optimal control of radiative heat transfer in glass cooling with restrictions on the temperature gradient. *Optimal Control Applications and Methods*, 33(2):157–175, 2012.

[12] Debora Clever, Jens Lang, and Dirk Schröder. Model hierarchy-based optimal control of radiative heat transfer. *International Journal of Computational Science and Engineering 2*, 9(5-6):509–525, 2014.

[13] Debora Clever, Jens Lang, Stefan Ulbrich, and Jan Carsten Ziems. Combination of an adaptive multilevel SQP method and a space-time adaptive PDAE solver for optimal control problems. *Procedia Computer Science*, 1(1):1435–1443, 2010.

[14] Debora Clever, Jens Lang, Stefan Ulbrich, and Jan Carsten Ziems. Generalized multilevel SQP-methods for PDAE-constrained optimization based on space-time adaptive PDAE solvers. In G. Leugering, S. Engell, A. Griewank, M. Hinze, R. Rannacher, V. Schulz, M. Ulbrich, and S. Ulbrich, editors, *Constrained Optimization and Optimal Control for Partial Differential Equations*, pages 37–60. International Series of Numerical Mathematics, Vol. 160, Basel, 2012.

[15] John Crank and Phyllis Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 43, pages 50–67. Cambridge Univ Press, 1947.

[16] Peter Deuflhard, Peter Leinen, and Harry Yserentant. Concepts of an adaptive hierarchical finite element code. *IMPACT of Computing in Science and Engineering*, 1(1):3–35, 1989.

[17] Peter Deuflhard and Martin Weiser. *Numerische Mathematik 3: adaptive Lösung partieller Differentialgleichungen*. De Gruyter Studium. De Gruyter, 2011.

[18] Willy Dörfler and Ricardo H. Nochetto. Small data oscillation implies the saturation assumption. *Numerische Mathematik*, 91(1):1–12, 2002.

[19] Bodo Erdmann, Jens Lang, and Rainer Roitzsch. KARDOS-User's Guide. Manual, Konrad-Zuse-Zentrum Berlin, 2002.

[20] Alf Gerisch, Jens Lang, Helmut Podhaisky, and Rüdiger Weiner. High-order linearly implicit two-step peer - finite element methods for time-dependent PDEs. *Applied Numerical Mathematics*, 59:634–638, 2009.

[21] William W. Hager. Runge-Kutta methods in optimal control and the transformed adjoint system. *Numerische Mathematik*, 87:247–282, 2000.

[22] Michael Hinze, Rene Pinnau, Michael Ulbrich, and Stefan Ulbrich. *Optimization with PDE constraints*, volume 23. Springer Science & Business Media, 2008.

[23] Dietmar Hömberg, Christian Meyer, Joachim Rehberg, and Wolfgang Ring. Optimal control for the thermistor problem. *SIAM Journal on Control and Optimization*, 48(5):3449–3481, 2010.

[24] David H. Jacobson and David Q. Mayne. *Differential dynamic programming*. Modern analytic and computational methods in science and mathematics. American Elsevier Pub. Co., 1970.

[25] C.T. Kelley. *Iterative Methods for Optimization*, volume 18. SIAM Frontiers in Applied Mathematics, 1999.

[26] Axel Klar, Jens Lang, and Mohammed Seaïd. Adaptive solution of SP_N-approximations to radiative heat transfer in glass. *International Journal of Thermal Science*, 44:1013–1023, 2005.

[27] Alois Kufner and Anna-Margarete Sändig. *Some applications of weighted Sobolev spaces*, volume 100. BG Teubner Gmbh, 1987.

[28] Jens Lang. *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems - Theory, Algorithm, and Applications*. Springer, Berlin, Heidelberg, 2001.

[29] Jens Lang and Delia Teleaga. Towards a fully space-time adaptive FEM for magnetoquasistatics. *IEEE Transactions on Magnetics*, 44,6:1238–124, 2008.

[30] Jens Lang and Jan G. Verwer. ROS3P - an accurate third-order Rosenbrock solver designed for parabolic problems. *BIT*, 41:730–737, 2001.

[31] Jens Lang and Jan G. Verwer. On global error estimation and control for initial value problems. *SIAM Journal on Scientific Computing*, 29(4):1460–1475, 2007.

[32] Jens Lang and Jan G. Verwer. W-methods in optimal control. *Numerische Mathematik*, 124(2):337–360, 2013.

[33] C. L. Lawson. C^1-Compatible interpolation over a triangle. *Technical Memorandum Jet Propulsion Labaratory*, pages 33–370, 1976.

[34] Christian Lubich and Alexander Ostermann. Linearly implicit time discretization of non-linear parabolic equations. *IMA journal of numerical analysis*, 15(4):555–583, 1995.

[35] Christian Lubich and Alexander Ostermann. Runge-Kutta approximation of quasi-linear parabolic equations. *Mathematics of computation*, 64(210):601–627, 1995.

[36] Dominik Meidner and Boris Vexler. Adaptive space-time finite element methods for parabolic optimization problems. *SIAM Journal on Control and Optimization*, 46(1):116–142, 2007.

[37] Ander Murua. On order conditions for partitioned symplectic methods. *SIAM Journal on Numerical Analysis*, 34:2204–2211, 1997.

[38] Ulrich Nowak. Adaptive Linienmethoden für nichtlineare parabolische Systeme in einer Raumdimension. *Ph.D. Thesis, Freie Universiät Berlin*, 1993.

[39] Alexander Ostermann and M. Roche. Rosenbrock methods for partial differential equations and fractional orders of convergence. *SIAM journal on numerical analysis*, 30(4):1084–1098, 1993.

[40] Bettina Peth. *Adaptive Two-Step Peer Methods for Incompressible Flow Problems*. Verlag Dr. Hut, 2010.

[41] Linda Petzold, Yang Cao, Shengtai Li, and Radu Serban. Adaptive numerical methods for sensitivity analysis of differential-algebraic equations and partial differential equations. *Workshop on Modelling and Simulation in Chemical Engineering, Coimbra, Portugal*, 2003.

[42] Linda R. Petzold. A description of DASSL: A differential/algebraic system solver. In *Proc. IMACS World Congress*, pages 430–432, 1982.

[43] Helmut Podhaisky, Rüdiger Weiner, and Bernhard A. Schmitt. Rosenbrock-type peer two-step methods. *Applied numerical mathematics*, 53(2):409–420, 2005.

[44] Sergey Repin, Stefan Sauter, and Anton Smolianski. A posteriori error estimation for the Poisson equation with mixed Dirichlet/Neumann boundary conditions. *Journal of computational and applied mathematics*, 164:601–612, 2004.

[45] Thomas Richter, Andreas Springer, and Boris Vexler. Efficient numerical realization of discontinuous Galerkin methods for temporal discretization of parabolic problems. *Numerische Mathematik*, 124(1):151–182, 2013.

[46] Adrian Sandu. On the properties of Runge-Kutta discrete adjoints. *Lecture Notes in Computer Science*, 3394:550–557, 2006.

[47] Adrian Sandu. Reverse automatic differentiation of linear multistep methods. In Christian Bischof, Martin Bäcker, Paul Hovland, Uwe Naumann, and Jean Utke, editors, *Advances in Automatic Differentiation*, volume 64 of *Lecture Notes in Computational Science and Engineering*, pages 1–12. Springer Berlin Heidelberg, 2008.

[48] Bernhard A. Schmitt and Rüdiger Weiner. Parallel two-step W-methods with peer variables. *SIAM Journal on Numerical Analysis*, 42(1):265–282, 2004.

[49] Bernhard A. Schmitt, Rüdiger Weiner, and Kilian Erdmann. Implicit parallel peer methods for stiff initial value problems. *Applied Numerical Mathematics*, 53:547–570, 2005.

[50] Bernhard A. Schmitt, Rüdiger Weiner, and Helmut Podhaisky. Multi-implicit peer two-step W-methods for parallel time integration. *BIT Numerical Mathematics*, 45(1):197–217, 2005.

[51] Dirk Schröder, Jens Lang, and Rüdiger Weiner. Stability and consistency of discrete adjoint implicit peer methods. *Journal of Computational and Applied Mathematics*, 262:73–86, 2014.

[52] Lawrence F. Shampine. Tolerance proportionality in ODE codes. In *Numerical Methods for Ordinary Differential Equations*, pages 118–136. Springer, 1989.

[53] Gustaf Söderlind. Digital filters in adaptive time-stepping. *ACM Transactions on Mathematical Software (TOMS)*, 29(1):1–26, 2003.

[54] Gustaf Söderlind and Lina Wang. Adaptive time-stepping and computational stability. *Journal of Computational and Applied Mathematics*, 185(2):225–243, 2006.

[55] Gerd Steinebach. Order reduction of ROW-methods for DAEs and method of lines applications. Technical report, Technische Hochschule Darmstadt, 1995.

[56] Henk A. Van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on scientific and Statistical Computing*, 13(2):631–644, 1992.

[57] Jan G. Verwer, Edwin J. Spee, Joke G. Blom, and Willem Hundsdorfer. A second-order Rosenbrock method applied to photochemical dispersion problems. *SIAM Journal on Scientific Computing*, 20(4):1456–1480, 1999.

[58] Andrea Walther. Automatic differentiation of explicit Runge-Kutta methods for optimal control. *Journal of Computational Optimization and Applications*, 36:83–108, 2007.

[59] Wolfram Research Inc. Mathematica edition: Version 7.0, 2008.

[60] Jan Carsten Ziems. *Adaptive multilevel SQP-methods for PDE-constrained optimization*. PhD thesis, Technische Universität Darmstadt, 2010.

[61] Jan Carsten Ziems and Stefan Ulbrich. Adaptive multilevel inexact SQP-methods for PDE-constrained optimization. *SIAM Journal on Optimization*, 21:1–40, 2011.

Wissenschaftlicher Werdegang

Dirk Schröder,
geboren am 18.09.1986 in Dieburg

02/2011-04/2016	Wissenschaftlicher Mitarbeiter am Fachbereich Mathematik der Technischen Universität Darmstadt, Arbeitsgruppe Numerik und Wissenschaftliches Rechnen
10/2005-01/2011	Studium Mathematik mit Nebenfach Informatik an der Technischen Universität Darmstadt
06/2005	Abitur an der Alfred-Delp-Schule Dieburg