INAUGURAL-DISSERTATION
zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen Gesamtfakultät
der
Ruprecht-Karls-Universität
Heidelberg

vorgelegt von
Diplom-Mathematiker Oliver Großhans, M.Sc.
aus Landau/Pfalz

Tag der mündlichen Prüfung: 13.12.2001

# Optimal Control of a Reactive Stagnation Point Flow on a Catalytic Plate

Gutachter: Prof. Dr. Dr. h.c. Hans Georg Bock
Prof. Dr. Rolf Rannacher

## Acknowledgments

At this point I would like to thank my advisors Prof. Dr. Dr. h. c. H. G. Bock and Dr. O. Deutschmann, Dr. J. Schlöder and Prof. Dr. V. Schulz for many helpful discussions and their support. In addition, I thank the Deutsche Forschungsgemeinschaft (DFG) for the support within the Sonderforschungsbereich 359 "Reaktive Strömungen, Diffusion und Transport".

Also I would like to thank all my colleagues in the Simulation and Optimization group at the IWR for their friendship and their assistance. I especially thank my former colleague and roommate Dr. Angelika Dieses for always encouraging me not to give up even when the problems seemed unsolvable and for the pleasure sharing the office with her.

And last, but not least I thank my parents for their continuous support and my fiancé Corinna for her love and her encouraging support while completing this thesis.

# Contents

# List of Symbols

### Lowercase Latin Characters

| | | |
|---|---|---|
| $c_p$ | specific heat capacity of mixture at constant pressure | $[\mathrm{J\,kg^{-1}\,K^{-1}}]$ |
| $c_{p,i}$ | specific heat capacity of the species $i$ at constant pressure | $[\mathrm{J\,kg^{-1}\,K^{-1}}]$ |
| $h_i$ | specific enthalpy | $[\mathrm{J\,kg^{-1}}]$ |
| $j_i$ | diffusion flux density | $[\mathrm{kg\,m^{-2}\,s^{-1}}]$ |
| $k_{f_k}$ | reaction rate coefficient of reaction $k$ | $[\mathrm{mol\,m\,s}]$ |
| $p$ | pressure | $[\mathrm{N\,m^{-2}}]$ |
| $\dot{s}_i$ | reaction rate of species $i$ on the surface | $[\mathrm{mol\,m^{-2}\,s^{-1}}]$ |
| $v_x$ | axial velocity | $[\mathrm{m\,s^{-1}}]$ |
| $u$ | Stefan velocity | $[\mathrm{m\,s^{-1}}]$ |

### Uppercase Latin Characters

| | | |
|---|---|---|
| $A_k$ | pre-exponential factor of reaction $k$ | $[\mathrm{mol\,m\,s}]$ |
| $D_{i,M}$ | effective diffusion coefficient of species $i$ in the mixture | $[\mathrm{m^2\,s^{-1}}]$ |
| $D_{ij}$ | binary diffusion coefficient | $[\mathrm{m^2\,s^{-1}}]$ |
| $D_i^T$ | thermal diffusion coefficient of species $i$ | $[\mathrm{kg\,m^{-1}\,s^{-1}}]$ |
| $E_{a_k}$ | activation energy of reaction $k$ | $[\mathrm{J\,mol^{-1}}]$ |
| $K_g$ | number of elementary gas phase reactions | [—] |
| $K_s$ | number of elementary surface reactions | [—] |
| $\bar{M}$ | mean molecular mass | $[\mathrm{kg\,mol^{-1}}]$ |
| $M_i$ | molecular mass of species $i$ | $[\mathrm{kg\,mol^{-1}}]$ |
| $N_b$ | number of bulk species | [—] |
| $N_g$ | number of gas phase species | [—] |
| $N_s$ | number of surface species | [—] |
| $R$ | universal gas constant | $[\mathrm{J\,K^{-1}\,kg^{-1}}]$ |
| $T$ | temperature | [K] |
| $V$ | scaled radial momentum $v_r/r$ | $[\mathrm{s^{-1}}]$ |
| $X_i$ | species mol fraction | [—] |
| $[X_i]$ | species concentration on the surface | $[\mathrm{mol\,m^{-2}}]$ |
| | in the gas phase | $[\mathrm{mol\,m^{-3}}]$ |
| $Y_i$ | species mass fraction | [—] |

### Lowercase Greek Characters

| | | |
|---|---|---|
| $\beta_k$ | temperature exponent of reaction $k$ | [—] |
| $\epsilon_{ik}$ | coverage dependent activation energy of reaction $k$ | $[\mathrm{J\,mol^{-1}}]$ |
| $\lambda$ | heat conductivity | $[\mathrm{J\,m^{-1}\,K^{-1}\,s^{-1}}]$ |

| | | |
|---|---|---|
| $\lambda_i$ | heat conductivity of species $i$ | $[\mathrm{J\,m^{-1}\,K^{-1}\,s^{-1}}]$ |
| $\mu$ | mean viscosity | $[\mathrm{N\,s\,m^{2}}]$ |
| $\mu_i$ | viscosity of species $i$ | $[\mathrm{N\,s\,m^{2}}]$ |
| $\mu_{ik}$ | parameter for the modification of the reaction order | [—] |
| $\nu'_{ik}, \nu''_{ik}$ | stoichiometric coefficients | [—] |
| $\dot{\omega}_i$ | reaction rate of species $i$ in the gas phase | $[\mathrm{mol\,m^{-3}\,s^{-1}}]$ |
| $\rho$ | density | $[\mathrm{kg\,m^{-3}}]$ |
| $\sigma_i$ | number of surface sites occupied by species $i$ | [—] |
| $\chi_i$ | chemical symbol of species $i$ | [—] |

## Uppercase Greek Characters

| | | |
|---|---|---|
| $\Gamma$ | surface site density | $[\mathrm{mol\,m^{-2}}]$ |
| $\Lambda$ | eigenvalue of pressure equation | $[\mathrm{N\,m^{-4}}]$ |
| $\Theta_i$ | surface coverages | [—] |

# 1. Introduction

Chemical reactions on catalytic surfaces are a very active and intense research area. The catalyst offers the possibility to accelerate and to control certain features of the reactions, which can lead to higher selectivities and conversion. In other words less natural and financial resources are needed.

A sophisticated approach for the study of this type of chemistry is based on elementary reaction models. This means all "net" or "overall" reactions are split up into elementary reactions which occur on a molecular level exactly in the way which is described by the reaction equation. For example the reaction of hydroxy radicals OH with molecular hydrogen $H_2$ forming water and hydrogen atoms is such an elementary reaction,

$$OH + H_2 \rightarrow H_2O + H.$$

On the contrary, the reaction

$$2H_2 + O_2 \rightarrow 2H_2O$$

is not an elementary one. Detailed investigations show that water is not produced by a single collision between the three reacting molecules. Instead, many reactive intermediates like H, O, OH are formed. So this reaction can be described by 38 elementary reactions [59].

For the validation of the proposed reaction mechanisms computer simulations are performed and the results are compared to experimental data. Methods for the mathematical parameter estimation [11,47] are still not yet as commonly used as they should be. In many cases, it is not sufficient to purely study the reaction mechanism on the catalytic surface, because the transport of the chemical species to and from the catalyst is also important. Therefore it is necessary to couple the equations for the surface reactions with the equations of the flow field. This coupling makes the solution of the problem more complex, because chemical reactions can be expressed as a set of *ODE* but the addition of the flow equations means the addition of a set of partial differential equations.

One experimental configuration, which has received much attention over the last few years, is the reactive stagnation point flow on a catalytic plate [5,6,17,18,26,34], in which a flow of chemical species in gas-phase is directed towards a catalytic plate. This setting has been used to assist in the development of reaction mechanisms for several chemical problems. Their application has shown very good agreement with corresponding experiments [17,18,60].

The software for the numerical simulation of this problem has been developed in the research group of Prof. Warnatz over the last 10 years. It is summarized in the package *DIFRUN* and simple user interfaces are provided. The reaction mechanisms

and process parameters are taken from a number of parameter files and all other necessary data for the reaction equations are then computed in a pre-processing step. The model equations are discretized in space by finite differences and the resulting differential-algebraic equations are solved by a modified *LIMEX* [5,38] which is based on the original *LIMEX* [16].

After evaluation of the reaction mechanisms the next step is to look for optimal process conditions, that is for example the best values for temperature, pressure, chemical composition, or flow velocity to obtain a maximum (or minimum) of the desired output. So far, trial-and-error methods have been widely used. But this optimization by trial-and-error is extremely time consuming. A complete simulation—possibly for a very large system of differential-algebraic equations—has to be performed for each new value of the objective function. So this way of optimization is only feasible for a very small number of control variables. Stationary problems with a constant control are possible examples. Transient problems with time dependent controls, leading to 10–100 control variables after discretization, cannot be practically solved on a trial-and-error basis anymore.

On the other hand, there has been an enormous development in the field of optimization of applied problems: For example in mechanics the optimal path planning for satellite mounted robots [51–53], or in chemistry the optimal control of distillation columns [21, 22] to mention only a few.

Mathematical sophistication, algorithm and software development, and powerful computing resources do now enable to tackle optimal control problems with several hundreds or even thousands of differential algebraic equations and a high number of control variables.

Two powerful software packages have been developed in the research group of Prof. Bock over the last several years: *MUSCOD*(II) [12,36,37] and *OCPRSQP* [48]. Both are based on the boundary value problem approach [11, 12], that is simulation and optimization are done at the same time, and both have been applied very successfully to many application problems (see above). *MUSCOD* solves the optimal control problem with a reduced *SQP*-method based on a multiple shooting discretization [37]. *OCPRSQP* uses a collocation method to discretize the *DAE* describing the model and solves the resulting optimization problem with a partially reduced *SQP*-method [48]. So far *OCPRSQP* is better suited for applications with a very large number of variables and only a few control variables. It has also been successfully used in a former investigation of homogeneous reaction systems [56, 57].

In the aforementioned studies, optimal control problems have been solved which can be described as sets of *DAE*. This is an established but still very active field. The next step is now to approach optimal control problems in partial differential equations. This is an emerging field of research and it is possible to deal with these very large systems only in recent times. Examples for recent research are the shape optimization of turbine blades [24,25,50,54] and optimization problems in water flow and transport processes in soils [23], to mention only a few.

Over the last couple of years other research groups have started working on systems similar to that we are considering here [42–44]. But this research is always restricted to one chemical system. In this thesis we aim for the largest possible flexibility and efficiency of the software. That means, it takes only minimal effort to go from one

chemical system to another and it is possible to use the software with only a rough understanding of mathematical optimization.

Other optimal control packages have been developed based on collocation, for example [58]. But they usually have the disadvantage that very large linear systems have to be solved compared to codes based on multiple shooting [40]. Therefore the development has concentrated on multiple shooting based algorithms [40, 41, 55]. A solution to this disadvantage of the algorithms based on collocation has been a partially reduced *SQP*-method developed in [48].

Hence, a software package is needed for the practical application that includes an easy-to-use user-interface to study various chemical systems and an optimal control code which is efficient, robust and also easy-to-use.

In this thesis a new software package has been developed based on the simulation code *DIFRUN* and the optimal control package *OCPRSQP*. This new code provides us for the first time with a software tool for the optimal control of a reactive stagnation point flow on a catalytic plate for different chemical processes. Only minimal effort is needed going from one chemical system to another. This required an overall design to use the desired features of *DIFRUN* and *OCPRSQP* and several conceptual changes, new algorithms and modifications to *OCPRSQP*. In the present work, this package is applied for computing the optimal solution to several practical problems.

**Organization of this Thesis**

The second chapter introduces the three application problems: the catalytic partial oxidation of methane to syngas [17, 18], the epoxidation of ethylene on silver [39] and the catalytic oxygen-free conversion of methane to ethane [60, 61]. For these applications we want to find the optimal process conditions and these systems are used as typical examples from everyday practice to show the performance of our new package. A solution from a simulation is presented for each problem, which can be used as a reference to which the optimal solution presented in Chapter 6 can be compared.

Chapter 3 introduces the partial differential equations, which model the reactive stagnation point flow on a catalytic plate [5, 17].

Chapter 4 shows how these model equations are discretized in space using finite differences leading to a system of differential-algebraic equations. The dependency of these equations on the variables is emphasized because it leads to a block band structure of the Jacobian of this system. It is shown that this *DAE* is of index 1. Afterwards a short overview of the software package *OCPRSQP* is given: The collocation discretization and the partially reduced *SQP*-algorithm are presented. The presentation of algorithmic details concentrates on the computation of the derivatives and the recursion formulas for the condensing step of the partially reduced *SQP*-algorithm. This complements the information from [48]

The fifth chapter is—besides the results in chapter 6—the heart of this thesis. The development of the new software package based on *DIFRUN* and *OCPRSQP* is described, especially some of the major obstacles on that way and possible solutions: non standard interfaces, the need for analytic derivatives in contrast to the used finite differences; to mention only a few. It turns out that the accuracy of the finite differences to compute the derivatives for the optimization is not enough. Therefore

an interface to *ADIFOR* [8–10] has been developed to provide the access to analytic derivatives. Computing the derivatives, solving the linear systems in the condensing step and solving the QP-problem are the most time consuming steps in the partially reduced *SQP*-algorithm. The structure of the derivatives has been analyzed with the aid of the dependency relations of chapter 4. This shows a characteristic block band structure which enables the use of a condensed mode of *ADIFOR* [10] for the computation the derivatives. This reduces the computing time between 20-30%. As a next step, a sparse LU-factorization and a corresponding sparse linear system solver was developed, taking into account the block band structure of the matrices. These new algorithms give a speed up of almost a factor 2 compared to the non-sparse version. In a next step the basic formulation of the partially reduced *SQP*-algorithm had to be changed. Up to that point the initial values were treated as possible control variables but in our problem class the initial values are always fixed so it is possible to reduce the size of the QP-problems quite considerably. New recursion algorithms are presented taking this fact into account. Reducing the size of the QP-problem accelerates the solution of the QP-problem by a factor of 4 for the typical problem sizes treated in this thesis.

In the sixth chapter optimal solutions to the aforementioned applications are presented. The behavior of the control and the objective function is shown when refining the collocation mesh or using a different spatial discretization. It can be observed that already with a very coarse discretization in space and time a fairly good approximation of the control and the objective function can be obtained. It can also be seen that all the modifications described in chapter 5 are necessary to be able to obtain the results in a reasonable computing time.

The seventh chapter summarizes the results and gives an outlook to further research directions.

# 2. Applications from Chemistry

All the processes described in this chapter are modeled by a reactive stagnation point flow on a catalytic plate (figure 2.1) because in laboratory experiments, the chemical processes are often considered in a stagnation flow configuration. The model equations are discussed in the next chapter. In this thesis the process optimization for

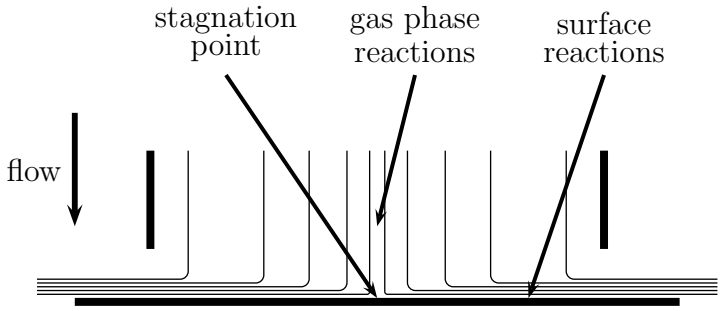


Figure 2.1.: Reactive stagnation point flow on a catalytic plate.

three different chemical application problems has been studied. These systems are shortly described in this chapter. The first process is the catalytic partial oxidation of methane to syngas—an industrial important process and of high potential to substitute the energy-costly conventional steam reforming [17, 18]. The second one, the epoxidation of ethylene on silver is a well established large scale chemical process [39]. The third and last process concerns the catalytic oxygen-free conversion of methane to ethane is currently mainly of academic interest [60, 61].

## 2.1. Catalytic Partial Oxidation of Methane to Syngas

Catalytic conversion of methane, the main component of natural gas, has recently received extensive experimental and theoretical attention because of its potential to synthesize useful chemicals. The catalytic reactors used for these processes have a complex interaction between the reactive flow and reactions on the catalytic surface. Therefore, the description of these heterogeneous reactors requires a detailed description of the coupling of the flow field and the catalyst.

Synthesis gas is a very important chemical intermediate for many relevant processes including the production of methanol and synthetic fuels by Fischer-Tropsch synthesis. Syngas is currently mainly produced by the endothermic steam reforming

of methane or by homogeneous oxidation in auto-thermal reforming. There has recently been significant interest in alternate routes to syngas production. The direct partial oxidation of light alkanes in a monolithic catalyst at very short contact times has been shown to offer a promising route to convert light alkanes to syngas, higher hydrocarbons, and oxygenates [13, 33]. Using this type of reactor, selectivities to syngas in excess of 90% have been achieved at methane conversions over 90% [13].

Syngas formation from methane/oxygen mixtures on noble metal catalysts is characterized by the competition between a complete oxidation channel globally written as

$$\begin{gathered} CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O \\ \triangle H_R = -890 \text{ kJ/mol} \end{gathered}$$

and a partial oxidation channel written as

$$\begin{gathered} CH_4 + \tfrac{1}{2}O_2 \rightarrow CO + 2H_2O \\ \triangle H_R = -36 \text{ kJ/mol.} \end{gathered}$$

This process, which can be run nearly auto-thermally and adiabatically, exhibits an extremely fast variation of temperature, velocity, and transport coefficients of the reactive mixture near the catalyst entrance.

**Optimal Process Control Problem**

One main contribution to the control of the syngas production is the ratio of methane to oxygen in the flow. If the mixture is too lean or rich, too much or too less $CO_2$ is obtained. Hence an optimal control problem can be stated as follows:

> Maximize the ratio of the mass fluxes of CO and $CO_2$ at the outflow depending on the ratio of the mol fractions of $CH_4$ and $O_2$ at the inflow. The sum of the mol fractions of $CH_4$ and $O_2$ at the inflow shall be constant.

This process is modeled by a reaction mechanism consisting of 11 surface species, 7 gas-phase species and with 32 elementary reactions which is shown in table A.1. This leads to a system of 11 *ODE* coupled with 11 partial differential equations (7 species + 4 flow equations). The following simulation uses a spatial discretization of 28 non-uniformly distributed grid points (the distribution is automatically computed by *DIFRUN*, for more details [6, 18]). This leads to an optimal control problem in 319 *DAE* (227 *ODE* and 92 *AE*) and one control function.

After a short initial phase this problem is stationary so it is justified not to consider time dependency. For this reason the coverages of the surface species are scalar values and in all other figures it is sufficient to draw the spatial dependence of the variables.

Table 2.1 shows the values of the coverages of the surface species on the catalytic plate. Figure 2.2 shows the mol fractions of the gas phase species and figure 2.3 the flow variables. This simulation is performed with a value of 1.41 for the ratio of the mol fractions of $CH_4$ and $O_2$ at the inflow. This yields a value of 149.46 for the ratio of the mass fluxes CO and $CO_2$ at the outflow.

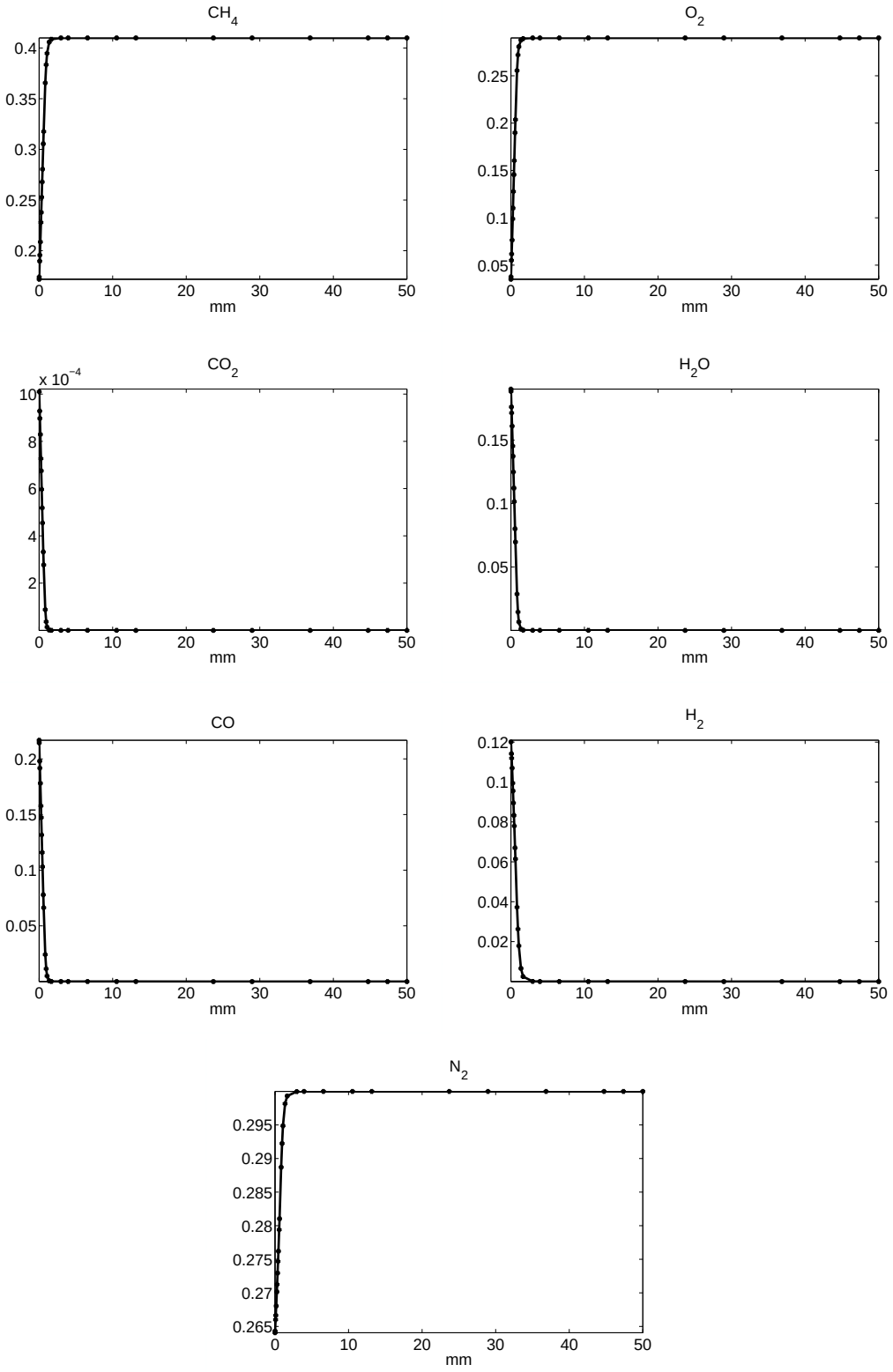


Figure 2.2.: Mol fraction of gas phase species of a solution from a simulation for the syngas problem. The catalytic plate is on the 'left' of the figures (0 mm) and the inflow on the 'right' (50 mm).

| Species | Coverage | Species | Coverage |
|---|---|---|---|
| Pt(s) | $6.10 \cdot 10^{-1}$ | O(s) | $4.34 \cdot 10^{-7}$ |
| H(s) | $8.54 \cdot 10^{-3}$ | $H_2O$(s) | $1.71 \cdot 10^{-4}$ |
| $CH_3$(s) | $4.41 \cdot 10^{-7}$ | OH(s) | $2.45 \cdot 10^{-5}$ |
| $CH_2$(s) | $3.05 \cdot 10^{-9}$ | CO(s) | $3.76 \cdot 10^{-1}$ |
| CH(s) | $1.11 \cdot 10^{-10}$ | $CO_2$(s) | $4.74 \cdot 10^{-10}$ |
| C(s) | $5.44 \cdot 10^{-3}$ | | |

Table 2.1.: Coverages for the surface species of a solution from a simulation for the syngas problem

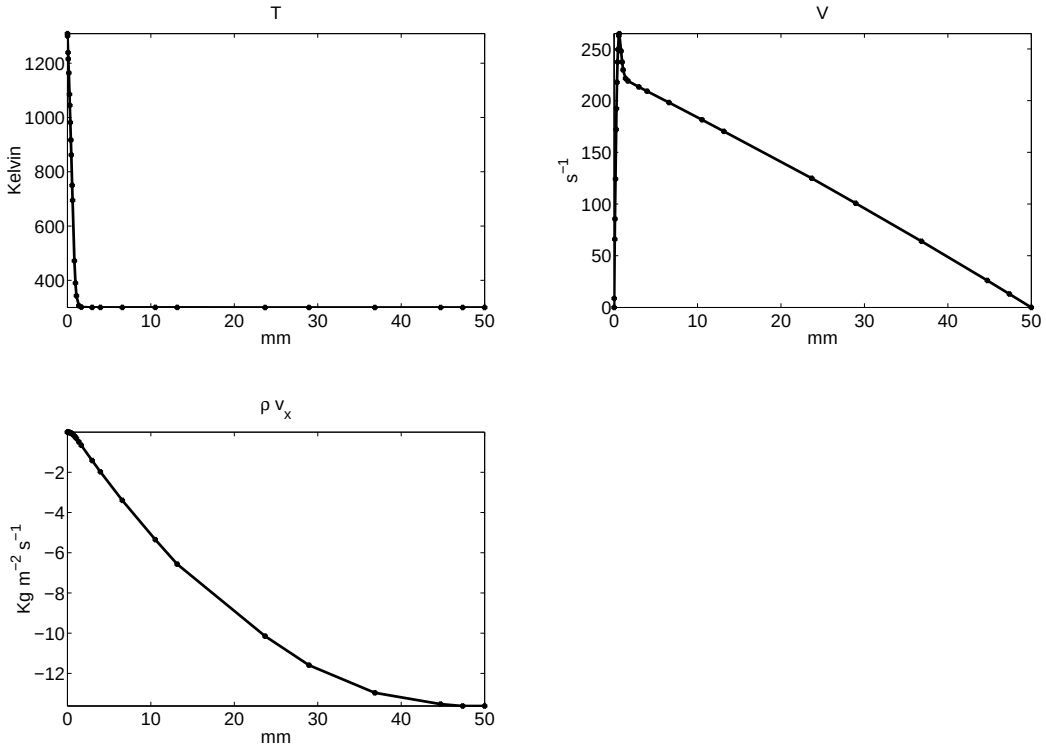


Figure 2.3.: Flow variables of a solution from a simulation for syngas problem. The catalytic plate is on the 'left' of the figures (0 mm) and the inflow on the 'right' (50 mm).

## 2.2. Epoxidation of Ethylene on Silver

The heterogeneous oxidation of olefines to epoxides by silver catalysts is of great industrial importance. The significance of epoxides comes from their potential to be intermediate for the production of glycols, polymers and other useful chemicals.

The process of the ethylenoxide production involves the interaction of ethylene with oxygen over a silver catalyst to produce the corresponding ethylene epoxide along with undesirable combustion products $CO_2$ and $H_2O$.

$$C_2H_4 + \tfrac{1}{2}O_2 \rightarrow C_2H_4O$$
$$C_2H_4 + 3O_2 \rightarrow 2CO_2 + 2H_2O$$

The task is to find optimal process parameters to maximize the production rate of ethylenoxid.

**Optimal Process Control Problem**

One important parameter to control the production rate of ethylenoxid is the ratio of ethylene to oxygen at the inflow. Also the temperature of the catalytic plate might play a role. Hence an optimal control problem can be stated as follows

> Maximize the production rate of $C_2H_4O$ at the outflow depending on the ratio of mol fractions $C_2H_4/O_2$ at the inflow and the temperature of the plate. The sum of mol fractions $C_2H_4 + O_2$ at the inflow shall be constant.

Here a reaction mechanism with 12 surface, 5 gas-phase species and 26 elementary reactions is considered which is shown in A.2. This results in a system of 12 *ODE* and 9 partial differential equations. The following simulation uses a uniform spatial discretization of 40 grid points which leads after spatial discretization of the partial differential equation to a system of 245 *ODE* and 127 *AE* and two control functions.

After a short initial phase the solution is stationary so it is again justified to focus on this stationary part. Table 2.2 shows the values of the coverages for the surface species on the catalytic plate. Figure 2.4 shows the mol fractions of the gas phase species and figure 2.5 shows the flow variables. This simulation is done with a value of 0.13 for the ratio of the mol fractions $C_2H_4$ and $O_2$ and a temperature of the catalytic plate of 525 K. This yields a production rate of $6.1 \cdot 10^{-6}$ mol/m$^2$s for $C_2H_2O$ at the outflow.

| Species | Coverage | Species | Coverage |
|---|---|---|---|
| Ag(s) | $9.69 \cdot 10^{-1}$ | HCOO(s) | $1.50 \cdot 10^{-2}$ |
| O(s) | $2.05 \cdot 10^{-5}$ | H(s) | $6.32 \cdot 10^{-3}$ |
| $C_2H_3$(s) | $5.97 \cdot 10^{-6}$ | $CO_3$(s1) | $6.73 \cdot 10^{-7}$ |
| $H_2O$(s) | $5.97 \cdot 10^{-4}$ | $C_2H_4$(s) | $2.17 \cdot 10^{-5}$ |
| OH(s) | $1.28 \cdot 10^{-9}$ | | |
| $C_2H_3O$(s) | $5.13 \cdot 10^{-3}$ | | |
| HCO(s) | $8.31 \cdot 10^{-4}$ | | |
| $CH_2O$(s) | $2.70 \cdot 10^{-3}$ | | |

Table 2.2.: Coverages for the surface species of a solution from a simulation for the epoxidation of ethylene.

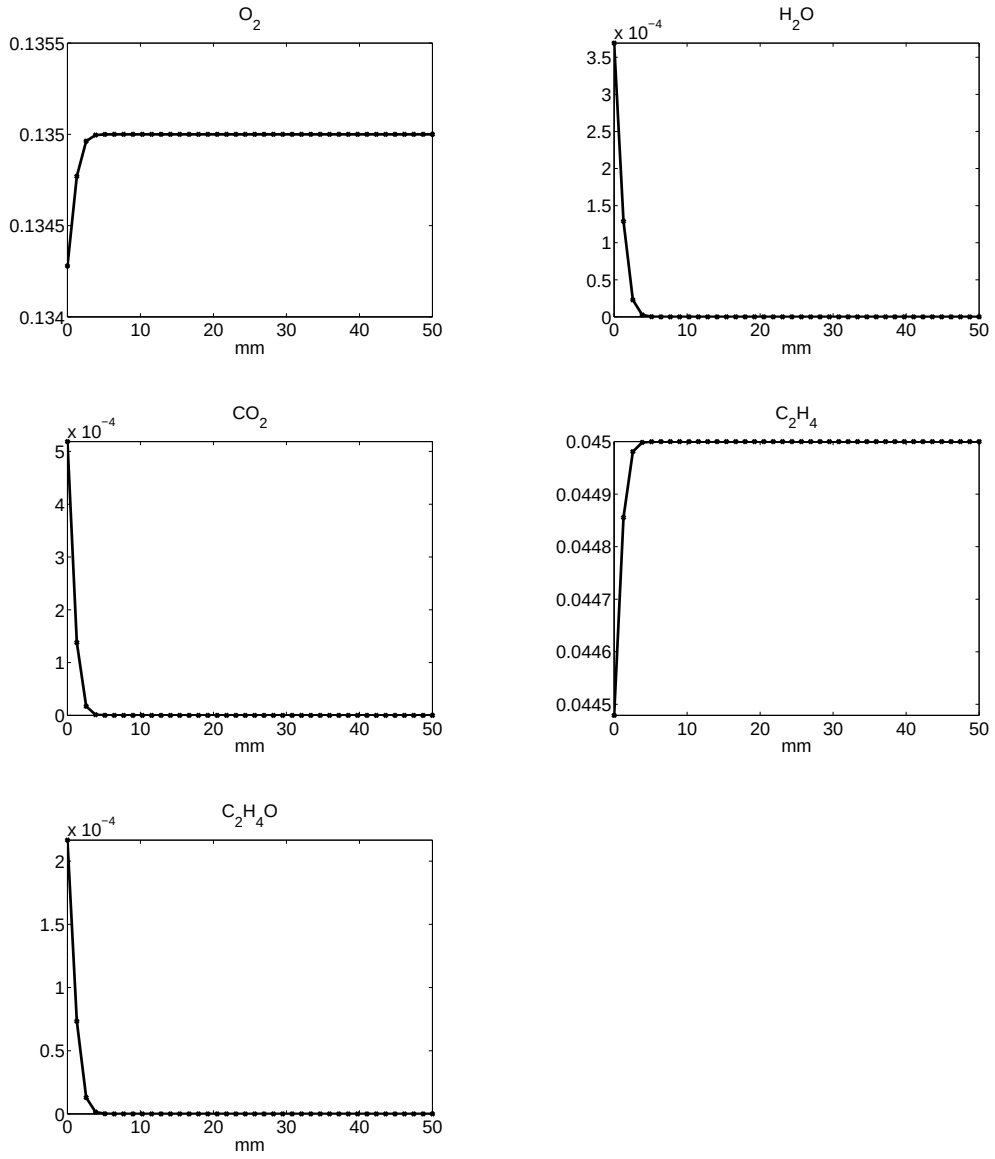


Figure 2.4.: Mol fraction of gas phase species of a solution from a simulation for the epoxidation of ethylene. The catalytic plate is on the 'left' of the figures (0 mm) and the inflow on the 'right' (50 mm).

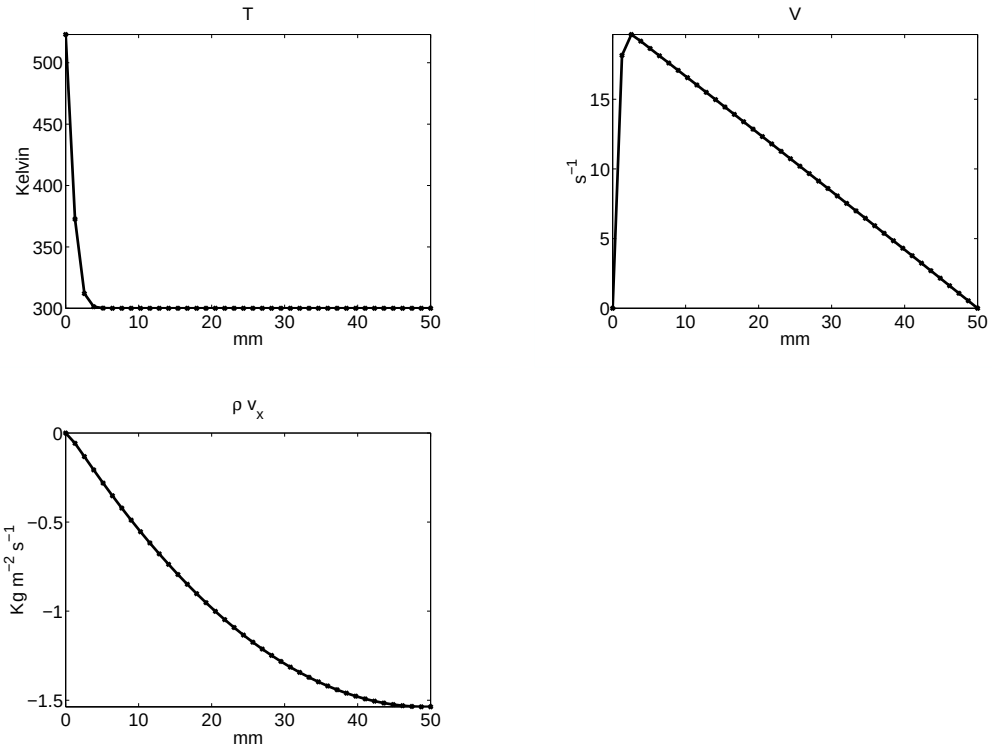


Figure 2.5.: Flow variables of a solution from a simulation for epoxidation of ethylene. The catalytic plate is on the 'left' of the figures (0 mm) and the inflow on the 'right' (50 mm).

## 2.3. Catalytic Oxygen-Free Conversion of Methane to Ethane

One way to obtain ethane from methane is via the syngas formation step of section 2.1. An alternate approach is the direct oxygen-free conversion of methane to ethane:

$$2CH_4 \rightarrow C_2H_6 + H_2$$

This application possesses a very high selectivity (80%) but the conversion is rather low. The aim is now to find process conditions which lead to a higher conversion.

**Optimal Process Control Problem**

The problems mentioned in the last two sections posses a stationary solution whereas this one is transient. The production rate of ethane depends mainly on three parameters: the velocity of the flow at the inflow, the temperature of the plate and the pressure. An optimal control problem can be stated as follows

> Maximize the production rate of $C_2H_6$ at the outflow depending on the flow velocity at the inflow, the temperature of the catalytic plate, and the pressure.

This problem is described by 14 surface, 4 gas-phase species and 39 elementary reactions with the reaction mechanism shown in table A.3. This leads to a system of 14 *ODE* and 8 partial differential equations. In the following simulation a uniform spatial discretization with 30 grid points has been used which the leads to a system of 158 *ODE* and 96 *AE* and three control functions. The whole simulation last for 300 s. Figures 2.6 and 2.7 show the coverages of the surface species. The next figure 2.8 shows the gas phase species. The flow variables are shown in figure 2.9. This simulation is done with a constant inflow velocity of 0.001 m/s, a constant temperature of the plate of 523 K and a constant pressure of 1 bar. So an integrated production rate of $9.817 \cdot 10^{-6}$ mol/m$^2$s is obtained for this simulation.

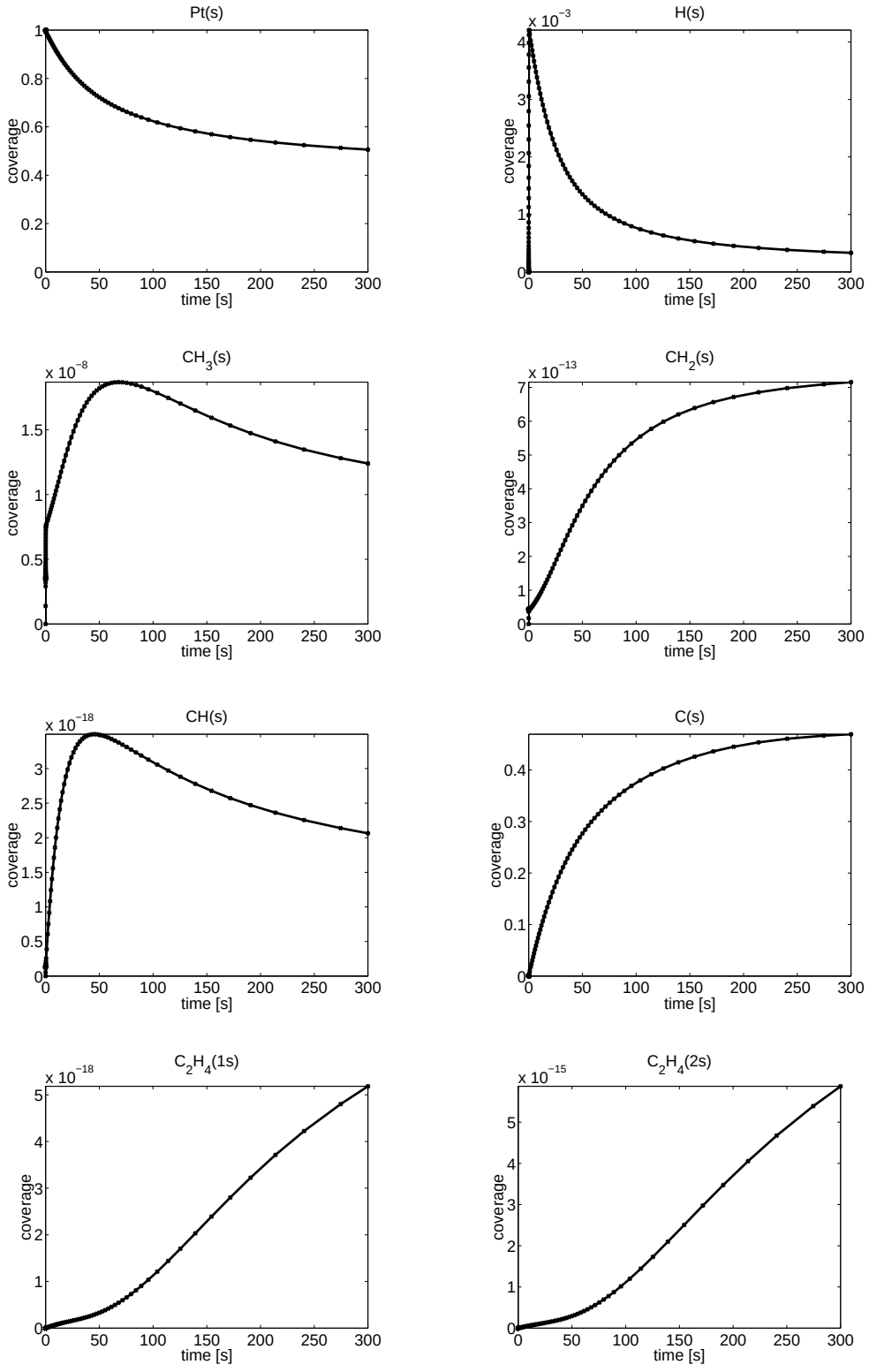


Figure 2.6.: Coverages for the surface species of a solution from a simulation for the catalytic oxygen-free conversion of methane to ethane.

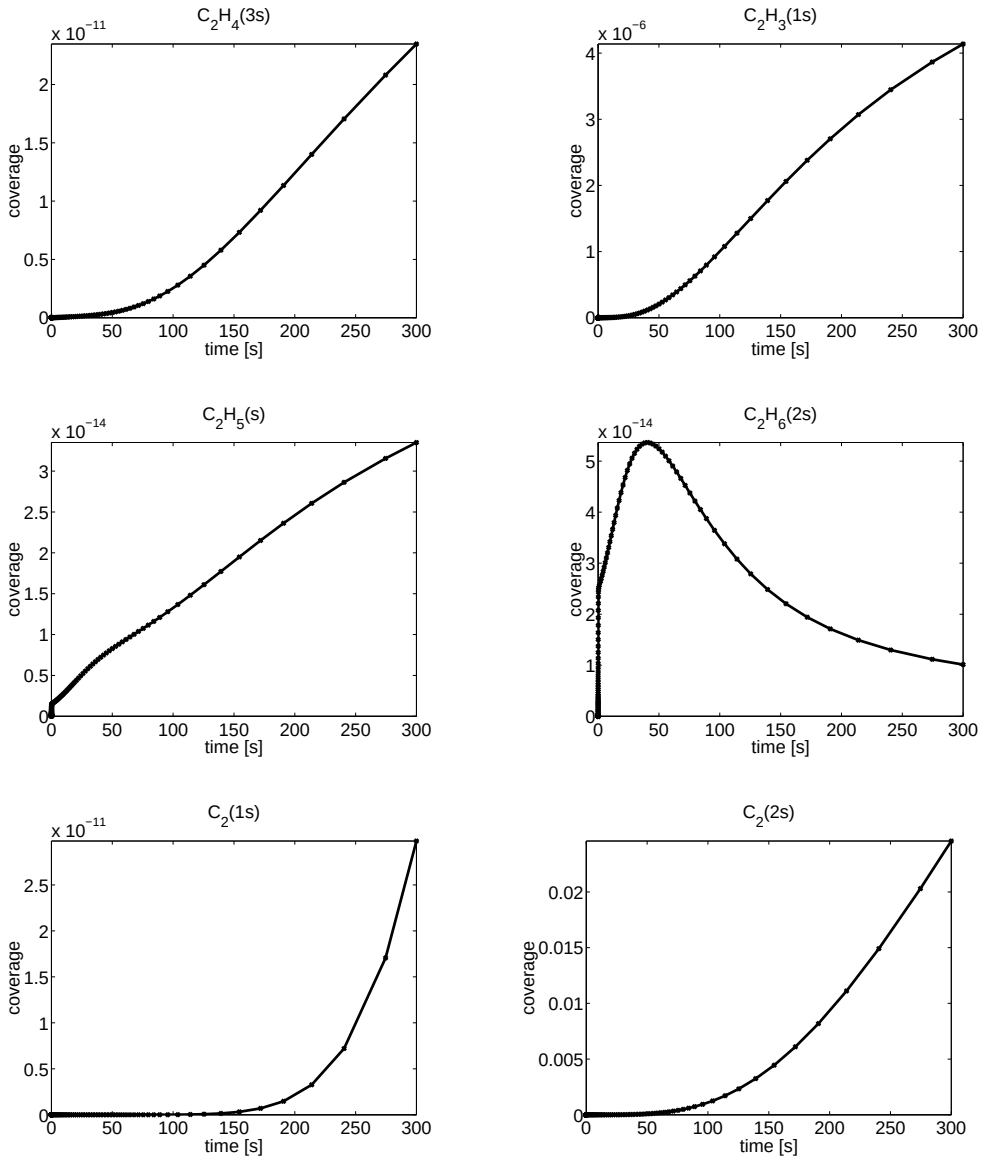


Figure 2.7.: Coverages for the surface species of a solution from a simulation for the catalytic oxygen-free conversion of methane to ethane.

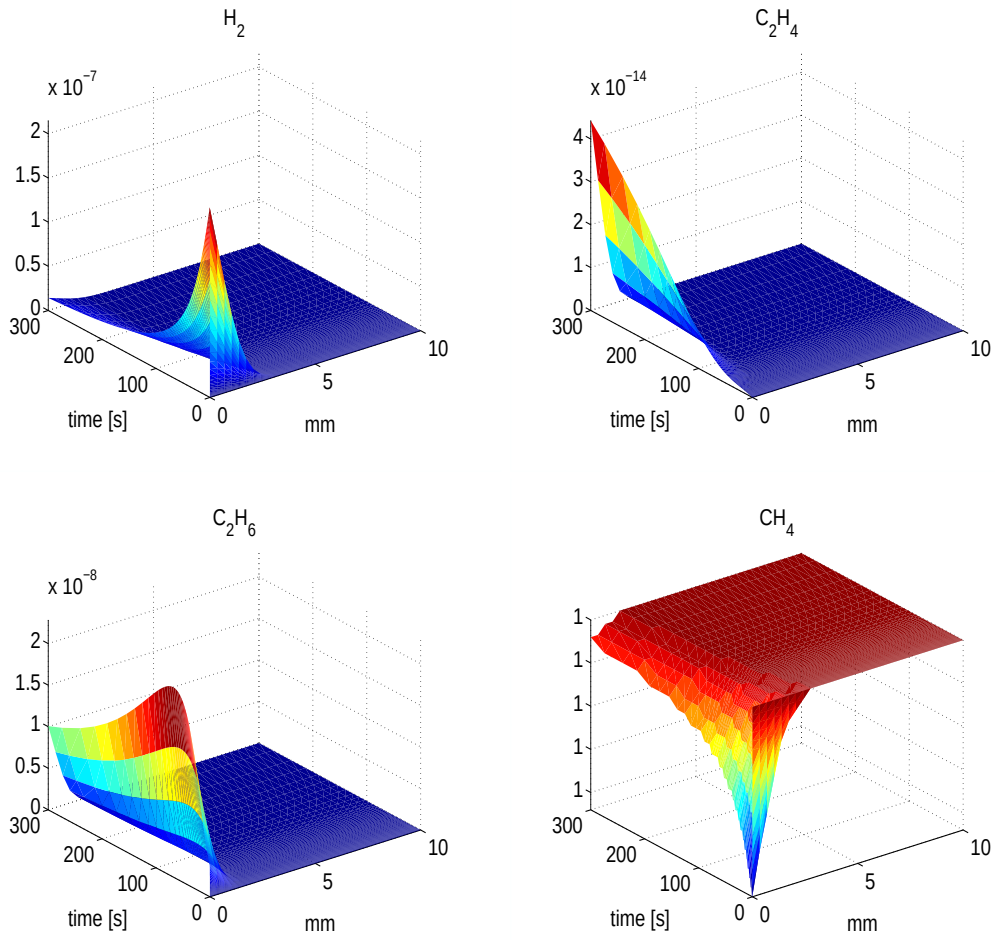


Figure 2.8.: Mol fraction of gas phase species of a solution from a simulation for the catalytic oxygen-free conversion of methane to ethane. The extreme variations in the beginning have to do with the setup of the problem. The initial data are not consistent a-priori so rapid changes can be seen before a reasonable solution is established.

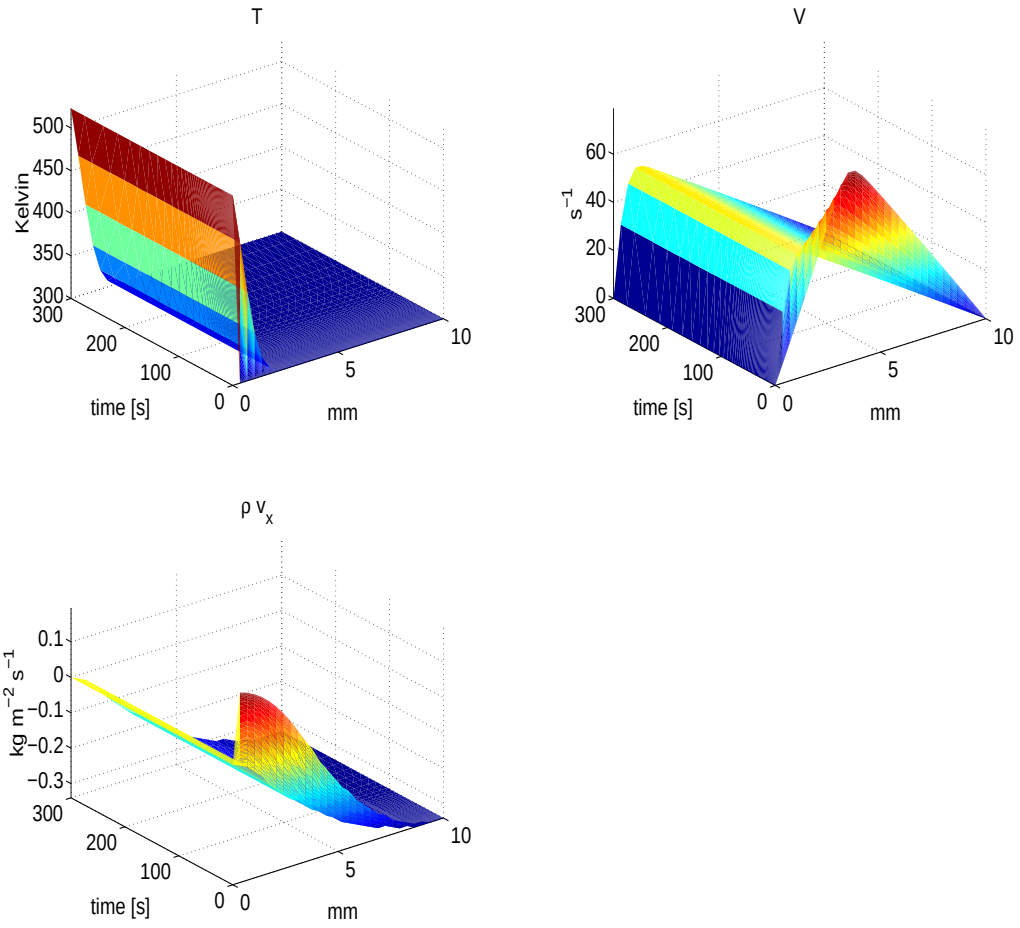


Figure 2.9.: Flow variables of a solution from a simulation for the catalytic oxygen-free conversion of methane to ethane. The extreme variations in the beginning have to do with the setup of the problem. The initial data are not consistent a-priori so rapid changes can be seen before a reasonable solution is established.

# 3. Reactive Stagnation Point Flow on a Catalytic Plate

The aim of this chapter is to describe the model equations of a reactive stagnation point flow on a catalytic plate. This gives an impression of the complexity of the model and furthermore its *structure* is needed in the optimization context. For a more detailed account of the derivation of the model equations it is refered to [5, 6, 17, 18, 20, 26, 34].

In a reactive stagnation point flow on a catalytic plate one considers an initially uniform flow of reacting chemicals in the gas phase leaving a pipe and hitting a catalytic plate before escaping through a small gap between plate and pipe (figure 3.1). Only an axial symmetric configuration is considered, which allows with certain approximations the reduction of this 3-dimensional problem to a one-dimensional. Furthermore only chemical reactions in the gas phase and on the catalytic plate are taken into account.

This configuration extends investigations on catalytic surface reactions, coupling the catalytic surface reactions with the characteristics of the—possibly time dependent—transport phenomena.

The whole model is built from three contributions: The equations describing the flow and the chemical reactions therein, the equations modeling the chemical reactions on the catalytic plate and the boundary conditions to the flow equations closing the system and coupling the flow equations with the equations describing the catalytic surface chemistry.

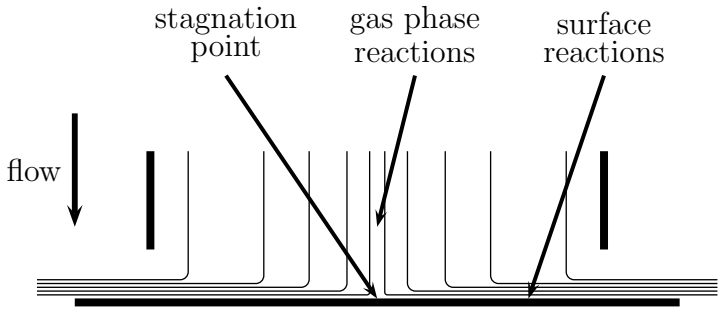


Figure 3.1.: Reactive stagnation point flow on a catalytic plate.

## 3.1. Gas Phase Equations

At first a look is taken at the equations describing the gas phase of the stagnation point flow. As mentioned above the flow equations—temperature, mass, momentum—are coupled with chemistry—species—and additionally an equation obtained through the dimensional reduction process. The equation for the temperature

$$\partial_t T = -\left[\frac{\rho v_x}{\rho} + \frac{1}{\rho c_p}\sum_{i=1}^{N_g} c_{p,i} j_i\right]\partial_x T - \frac{1}{\rho c_p}\sum_{i=1}^{N_g}\dot{\omega}_i M_i h_i + \frac{1}{\rho c_p}\partial_x(\lambda\partial_x T) \tag{3.1}$$

and the $N_g$ equations for the species

$$\partial_t Y_i = -\frac{\rho v_x}{\rho}\partial_x Y_i + \frac{1}{\rho}\dot{\omega}_i M_i - \frac{1}{\rho}\partial_x j_i \tag{3.2}$$

describe a time evolution. The equation for the mass conservation

$$0 = \frac{p}{R}\frac{\bar{M}^2}{T^2}\left[T\sum_i \frac{\partial_t Y_i}{M_i} + \frac{\partial_t T}{\bar{M}}\right] - 2\rho V - \partial_x(\rho v_x) \tag{3.3}$$

and the two equations for the momentum

$$0 = -\frac{\rho v_x}{\rho}\partial_x V - V^2 - \frac{\Lambda}{\rho} + \frac{1}{\rho}\partial_x(\mu\partial_x V) \tag{3.4}$$

$$0 = \partial_x \Lambda. \tag{3.5}$$

contain no time derivative and are treated as algebraic equations in time. The independent variables in this system are the temperature $T$, the species mass fraction $Y_i$, the axial mass flux $\rho v_x$, the radial momentum $V$, and the eigenvalue $\Lambda$ of the momentum equation. Please observe that the term $\rho v_x$ is considered to be an independent variable and not the radial velocity $v_x$. The state equation

$$\rho = \frac{p\bar{M}}{RT} \tag{3.6}$$

with fixed pressure $p$, mean molecular mass $\bar{M}$, and universal gas constant $R$, can be used to compute the density $\rho$.

In the following these model equations are explained in more detail. The equation (3.1) describes the evolution of the temperature $T$. The first term of the temperature equation (3.1) plays the role of a convection term. The "convection velocity" $v_x$ is corrected by an additional term

$$\frac{1}{\rho c_p}\sum_{i=1}^{N_g} c_{p,i} j_i$$

in which $c_p$ is the specific heat capacity of the mixture at constant pressure, $c_{p,i}$ the specific heat capacity of the species $i$ at constant pressure and $N_g$ the number of gas

phase species. The diffusion flux density $j_i$ is given by

$$\hat{j}_i = -\left(\rho D_{i,M}\partial_x Y_i + \frac{D_i^T}{T}\partial_x T + \frac{\rho D_{i,M} Y_i}{\bar{M}}\partial_x \bar{M}\right)$$
$$j_i = \hat{j}_i - Y_i \sum_{k=1}^{N_g} \hat{j}_k$$

where the mean molecular mass $\bar{M}$ is

$$\bar{M} = \frac{1}{\sum_{i=1}^{N_g} \frac{Y_i}{M_i}}$$

with $M_i$ the molecular mass of species $i$, and $D_{i,M}$ is the effective diffusion coefficient of the species $i$ in the mixture, $D_i^T$ the thermal diffusion coefficient. As a final equation for this first term in equation (3.1) it is necessary to compute $D_{i,M}$ from binary diffusion coefficients $D_{ij}$

$$D_{i,M} = \frac{1 - Y_i}{\sum_{j \neq i} \frac{X_j}{D_{ij}}}$$

where the mol fraction $X_i$ is related to the mass fraction through

$$X_i = \frac{\bar{M}}{M_i} Y_i.$$

The second term in equation (3.1) is a source term. It represents the fact that due to the chemical reactions the temperature might change. The chemistry in the gas phase is described by elementary reactions

$$\sum_{i=1}^{N_g} \nu'_{ik} \chi_i \rightarrow \sum_{i=1}^{N_g} \nu''_{ik} \chi_i$$

with stoichiometric coefficients $\nu'_{ik}, \nu''_{ik}$ of species $i$ in reaction $k$, $\chi_i$ species symbol and $K_g$ the number of elementary reactions. The reaction rate $\dot{\omega}_i$ is given by

$$\dot{\omega}_i = \sum_{k=1}^{K_g} \nu_{ik} k_{f_k} \prod_{i=1}^{N_g} [X_i]^{\nu'_{ik}}$$

where $\nu_{ik} = \nu''_{ik} - \nu'_{ik}$ and the velocity coefficients $k_{f_k}$ are temperature dependent and follow a modified Arrhenius law

$$k_{f_k} = A_k T^{\beta_k} \exp(-\frac{E_{a_k}}{RT})$$

with $A_k$ as pre-exponential factor, $\beta_k$ as temperature exponent and $E_{a_k}$ as activation energy. The specific enthalpy $h$ is defined as

$$h = \sum_i^{N_g} Y_i h_i \quad \text{with} \quad h_i = \int_{T_{\text{ref}}}^{T} c_{p,i}\, dT$$

where $T_{\text{ref}}$ is a reference temperature and $c_{p,i}$ is the specific heat at constant pressure of species $i$. $c_{p,i}$ is modeled as a polynomial function of temperature. More details can be found in [17].

The last term in equation (3.1) is a diffusion contribution. The mean viscosity of the mixture $\lambda$ is given by

$$\lambda = \frac{1}{2}(\sum_{i=1}^{N_g} X_i\lambda_i + (\sum_{i=1}^{N_g} \frac{X_i}{\lambda_i})^{-1})$$

with $\lambda_i$ being the heat conductivity coefficient of the species $i$.

The species equation (3.2) has a very similar structure to the temperature equation. The first term is of convective type, the second one represents a source term since during chemical reactions a species might be the product of a chemical reaction of other species, and the last one is a diffusion term.

Since it is assumed that the pressure is constant, the state equation (3.6) has been used to eliminate the explicit time dependence which can usually be found in the mass conservation equation, and this leads to (3.3) with the additional source term $2\rho V$ and the convection term $\partial_x(\rho v_x)$.

The first term of the momentum equation (3.4) has the form of a convection term, the second and third term represent a source and the last term is a diffusive contribution with the mean viscosity $\mu$

$$\mu = \frac{1}{2}(\sum_{i=1}^{N_g} X_i\mu_i + (\sum_{i=1}^{N_g} \frac{X_i}{\mu_i})^{-1})$$

and $\mu_i$ the viscosity coefficient of the species $i$. The last equation (3.5) for the pressure eigenvalue $\Lambda$ is a result of reducing the dimension of the problem from two to one dimensions. For further details it is refered again to [17].

## 3.2. Surface Equations

The chemical reactions on the catalytic plate are described by an additional set of equations, which compute the change of the coverage $\Theta_i$ for the species $i$ on the surface.

$$\partial_t\Theta_i = \frac{\dot{s}_i\sigma_i}{\Gamma} \quad [i = 1, \ldots, N_s] \tag{3.7}$$

with the surface site density $\Gamma$, the number of surface sites $\sigma_i$ needed for adsorption of species $i$ and reaction rate $\dot{s}_i$. Furthermore it has to be ensured that the coverages $\Theta_i$ add up to one

$$\sum_{i=1}^{N_s} \Theta_i = 1$$

The reaction rate $\dot{s}_i$ in (3.7) is given by

$$\dot{s}_i = \sum_{k=1}^{K_s} \nu_{ik} k_{f_k} \prod_{i=1}^{N_g+N_s+N_b} [X_i]^{\nu'_{ik}}$$

with the number of elementary surface reactions $K_s$, the number of species $N_s$ and the number of bulk species $N_b$. The velocity coefficients $k_{f_k}$ can again be described by a modified Arrhenius law

$$k_{f_k} = A_k T^{\beta_k} \exp(-\frac{E_{a_k}}{RT}) f_k(\Theta_1, ..., \Theta_{N_s})$$

in which $f_k$ depends on the coverages

$$f_k = \prod_{i=1}^{N_s} \Theta_i^{\mu_{i_k}} \exp(\frac{\epsilon_{ik}\Theta_i}{RT})$$

The changed reaction order is described by $\mu_{i_k}$ and the coverage dependent activation energy by $\epsilon_{ik}$.

## 3.3. Boundary Conditions

Besides the initial values one has to choose $2N_g + 7$ boundary values to close the system (3.1)–(3.5). At the inflow Dirichlet conditions are assumed for the temperature $T$, the species mass fraction $Y_i$, the axial mass flux $\rho v_x$, and the momentum $V$, that is

$$\begin{aligned} T &:= T^0 & Y_i &:= Y_i^0 \\ (\rho v_x) &:= \rho^0 v_x^0 & V &:= 0 \end{aligned}$$

which gives us $N_g + 3$ conditions. At the outflow the situation is more complicated and leads to a coupling between gas phase and surface equations. For the temperature equation we have a differential equation

$$\begin{aligned} \left(\rho c_\mathrm{p} \triangle x^+ + \rho_\mathrm{cat} c_\mathrm{cat} d\right) \partial_t T = \lambda_b \partial_x T - 2\sigma\epsilon(T^4 - T_\mathrm{ref}^4) - \sum_{i=1}^{N_g+N_s} \dot{s}_i M_i h_i + \frac{I^2 \rho_\mathrm{el}}{db^2} \\ -\lambda_b (T - T_b)^\kappa - \sum_{i=1}^{N_g} \dot{\omega}_i M_i h_i \triangle x^+ \end{aligned}$$

with $\rho_\mathrm{cat}$ the density, $c_\mathrm{cat}$ the heat capacity and $d$ the thickness of the catalytic plate. The first term describes the conductivity of the catalytic plate with the Fourier heat conductivity law $\lambda_b \partial_x T$ in which $\lambda_b$ is an experimental value. The second term contains the radiative contribution as a Stefan-Boltzmann law with the Boltzmann constant $\sigma$, the temperature dependent radiation value $\epsilon$ and the reference temperature $T_\mathrm{ref}$. The third term is a contribution from chemical surface reactions and the forth one describes the heating of the catalytic plate. $I$ is the electric current, $\rho_\mathrm{el}$ the specific electric resistant and $b$ the length of the catalytic plate. The fifth term encompasses the heat loss of the catalytic plate and the values of $\lambda_b$, $T_0$, and $\kappa$ are purely experimental. Finally the last term is a source term for the energy changes during chemical reactions.

For the species equation (3.2) a balance of species in the gas phase just above the plate and the species on the plate is assumed

$$\rho \partial_t Y_i \triangle x^+ = -j_i - \rho u Y_i + \dot{s}_i M_i + \dot{\omega}_i M_i \triangle x^+$$

with the Stefan velocity

$$u = \frac{1}{\rho} \sum_{i=1}^{N_g} \dot{s}_i M_i.$$

For the mass conservation equation (3.3) and for the momentum equation (3.4) one has

$$(\rho v_x)_0 = \rho u \quad \text{and} \quad V = 0.$$

Altogether this gives another $N_g + 3$ conditions. To compensate for the lack of a genuine boundary condition for equation (3.5) a zero gradient condition is used for the mass flux at the inflow

$$\partial_x (\rho v_x) = 0$$

With these boundary conditions the system is well defined [17].

# 4. Numerical Methods

After describing the model equations in the last chapter it is now shown how these equations are discretized and how the resulting optimal control problem is solved.

Since *OCPRSQP* is used to solve the optimal control problem the model equations are needed in the form of a system of differential-algebraic equations. Internally the simulation tool *DIFRUN* uses a modified version of *LIMEX* which is a solver for a system of *DAE*. Therefore the model equations already have the desired form. In section 4.1 it is shown how the model equations are discretized in space by finite differences to obtain the desired *DAE*. The dependency of the right hand sides on the discretized variables is of particular interest. It is this dependency which is used later on exploring the structure of the Jacobian, that is the derivative of the right hand site of the *DAE* with respect to the discretized variables. It is shown in section 4.2 that the *DAE* from section 4.1 is of index 1 which is a prerequisite for the usability of *OCPRSQP*. Section 4.3 is devoted to a short summary of the collocation method which is used in *OCPRSQP* to discretize the *DAE* in time. Section 4.4 finally contains a short summary of the partially reduced *SQP* algorithm implemented in *OCPRSQP*. The presentation concentrates on a detailed account of the formulas and recursions for which new ones are presented in chapter 5 to obtain a better performance of the algorithm. For further information regarding the partially reduced *SQP* method the reader is refered to [48]

## 4.1. The Discretization of the Model Equations

The convention used for the labeling of the spatial discretization is shown in figure 4.1. The first grid point $x_1$ is assumed to be just above the catalytic plate and the last one, $x_N$, at the inflow. The spatial discretization of the model equations uses one-sided and central finite differences. For the first derivative either first order one-sided

$$(\partial_x f)_l := \frac{f_{l+1} - f_l}{\triangle x_l}$$

or second order central finite differences

$$(\partial_x f)_l := \frac{1}{\triangle x_l + \triangle x_{l-1}} \left( \frac{\triangle x_l}{\triangle x_{l-1}} (f_l - f_{l-1}) + \frac{\triangle x_{l-1}}{\triangle x_l} (f_{l+1} - f_l) \right)$$

are used and for the second derivatives first order central finite differences

$$\partial_x (f \partial_x g)_l := \frac{1}{\triangle x_l + \triangle x_{l-1}} \left( (f_{l+1} + f_l) \frac{g_{l+1} - g_l}{\triangle x_l} - (f_l + f_{l-1}) \frac{g_l - g_{l-1}}{\triangle x_{l-1}} \right) \qquad (4.1)$$

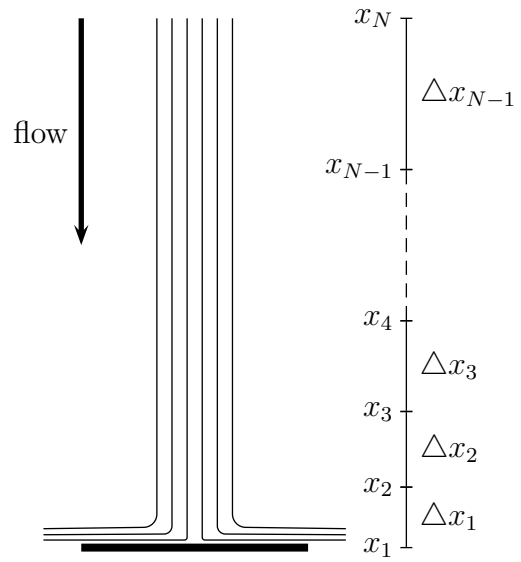


Figure 4.1.: Labeling for the spatial discretization.

An important observation is that using these formulas an discretized equation only depends on the variables of at most three neighboring grid points: the grid point at which the equation is computed and the two neighboring grid points.

The spatial discretization of the equations is now straightforward. The discretized gas phase equations and boundary conditions are not presented in full detail but the presentation concentrates on the dependency of these equations on the discretized variables because this dependency is used later on to reduce the computational complexity of the optimization algorithm.

**Gas Phase Equations**

For ease of presentation a slightly different notation is used in the remaining of this section: the indices for the species are now superscript and the indices for the discretization subscript. The species equation becomes

$$\partial_t Y_l^i = -\frac{(\rho v_x)_l}{\rho_l} \frac{Y_{l+1}^i - Y_l^i}{\triangle x_l} + \frac{1}{\rho_l} \dot{\omega}_l^i M^i - \frac{1}{\rho_l} J_l^i \tag{4.2}$$

with

$$\rho_l = \rho(T_l, \{Y_l^k\}_{k=1}^{N_g})$$
$$\dot{\omega}_l^i = \dot{\omega}^i(T_l, \{Y_l^k\}_{k=1}^{N_g})$$

and for $J_l^i$

$$J_l^i = (\partial_x \hat{j}_i)_l - Y_l^i \sum_{k=1}^{N_g} (\partial_x \hat{j}_k)_l$$

with

$$(\partial_x \hat{j}_i)_l = -[\partial_x(\rho D_{i,M}\partial_x Y_i)]_l - [\partial_x(\frac{D_i^T}{T}\partial_x T)]_l - [\partial_x(\frac{\rho D_{i,M} Y_i}{\bar{M}}\partial_x \bar{M})]_l$$

in which formula (4.1) can be used to get

$$J_l^i = \mathbf{J}_l^i\left(T_{l-1}, T_l, T_{l+1}, \{Y_{l-1}^k, Y_l^k, Y_{l+1}^k\}_{k=1}^{N_g}\right)$$

The boldface $\mathbf{J}_l^i$ is a complicated function of all the variables listed in the brackets. In the following the precise form of this function is of no interest but the dependency on the different variables because this defines the structure of the problem. Therefore one obtains the following

$$\partial_t Y_l^i = \mathbf{Y}_l^i\left((\rho v_x)_l, T_{l-1}, T_l, T_{l+1}, \{Y_{l-1}^k, Y_l^k, Y_{l+1}^k\}_{k=1}^{N_g}\right)$$

It is a bit more work to find the dependency of the temperature equation on the discretized variables. One has

$$\partial_t T_l = -\left[\frac{(\rho v_x)_l}{\rho_l} + \frac{1}{\rho_l c_{\mathrm{p}l}}\sum_{i=1}^{N_g} c_{\mathrm{p}l}^{\,i}(j_i)_l\right]\frac{T_{l+1}-T_l}{\triangle x_l} - \\ - \frac{1}{\rho_l c_{\mathrm{p}l}}\sum_{i=1}^{N_g}\dot{\omega}_l^i M^i h_l^i + \frac{1}{\rho_l c_{\mathrm{p}l}}[\partial_x(\lambda\partial_x T)]_l \quad (4.3)$$

But using the results from the species equation and since there is no dependency on $V$ or $\Lambda$ one obtains

$$\partial_t T_l = \mathbf{T}_l\left((\rho v_x)_l, T_{l-1}, T_l, T_{l+1}, \{Y_{l-1}^k, Y_l^k, Y_{l+1}^k\}_{k=1}^{N_g}\right)$$

which can already be seen from the first and the last term of the equation. The next step is the mass conservation equation. Since the calculations for the temperature and species equation have already been done this is not too difficult. For

$$0 = \frac{p}{R}(\frac{\bar{M}_l}{T_l})^2\left[T_l\sum_{i=1}^{N_g}\frac{[\partial_t Y^i]_l}{M^i} + \frac{[\partial_t T]_l}{\bar{M}_l}\right] - 2\rho_l V_l - \frac{(\rho v_x)_{l+1}-(\rho v_x)_l}{\triangle x_l} \quad (4.4)$$

one obtains

$$0 = (\rho \mathbf{v_x})_\mathbf{l}\left((\rho v_x)_l, (\rho v_x)_{l+1}, V_l, T_{l-1}, T_l, T_{l+1}, \{Y_{l-1}^k, Y_l^k, Y_{l+1}^k\}_{k=1}^{N_g}\right)$$

The discretization of the momentum equation is straightforward

$$0 = -\frac{(\rho v_x)_l}{\rho_l}\frac{V_{l+1}-V_l}{\triangle x_l} - V_l^2 - \frac{\Lambda_l}{\rho_l} + \frac{1}{\rho_l}[\partial_x(\mu\partial_x V)]_l \quad (4.5)$$

and this equation can therefore be seen as a function $\mathbf{V}_l$ depending on

$$0 = \mathbf{V}_l\left((\rho v_x)_l, \Lambda_l, T_l, V_{l-1}, V_l, V_{l+1}, \{Y_{l-1}^k, Y_l^k, Y_{l+1}^k\}_{k=1}^{N_g}\right)$$

The equation for $\mathbf{\Lambda}_l$ is easily discretized and one obtains

$$0 = \mathbf{\Lambda}_l(\Lambda_l, \Lambda_{l+1}) = \frac{\Lambda_{l+1} - \Lambda_l}{\triangle x_l} \tag{4.6}$$

**Boundary Conditions**

The first step is to discretize the boundary conditions at the inflow. One sets

$$T_N := T^0 \qquad Y_N^i := Y^{0i} \tag{4.7}$$

$$(\rho v_x)_N := \rho^0 v_x^0 \qquad V_N := 0 \tag{4.8}$$

with $T^0, Y^{0i}, \rho^0 v_x^0$ the fixed boundary values. So there are no dependencies on other variables. At the outflow the situation is more complex. For the temperature one has

$$\begin{aligned}\left(\rho_1 c_{\mathrm{p}_1} \triangle x^+ + \rho_{\mathrm{cat}} c_{\mathrm{cat}} d\right) \partial_t T_1 = \lambda_b \frac{T_1 - T_2}{\triangle x_1} - 2\sigma\epsilon_1 (T_1^4 - T_{\mathrm{ref}}^4) - \sum_{i=1}^{N_g+N_s} \dot{s}^i M^i h_l^i + \frac{I^2 \rho_{\mathrm{el}}}{db^2} \\ -\lambda_b (T_l - T_b)^\kappa - \sum_{i=1}^{N_g} \dot{\omega}^i M^i h_l^i \triangle x^+\end{aligned} \tag{4.9}$$

with $\triangle x^+ = \frac{x_1}{2}$ and using the results from above this yields

$$\partial_t T_1 = \mathbf{T}_1(T_1, T_2, \{Y_1^k, Y_2^k\}_{k=1}^{N_g}, \{\Theta^j\}_{j=1}^{N_s})$$

For the species one has

$$\rho_1 \partial_t Y_1^i \triangle x^+ = -j_1^i - \rho_1 u Y_1^i + \dot{s}^i M^i + \dot{\omega}_1^i M^i \triangle x^+ \tag{4.10}$$

with Stefan velocity $u$ and again with the results from above

$$\partial_t Y_1^i = \mathbf{Y}_1^i(T_1, T_2, \{Y_1^k, Y_2^k\}_{k=1}^{N_g}, \{\Theta^j\}_{j=1}^{N_s})$$

Since $u = \mathbf{u}(T_1, \{Y_1^k\}_{k=1}^{N_g}, \{\Theta^j\}_{j=1}^{N_s})$ the mass flux equation can be written as

$$(\rho v_x)_1 = \rho_1 u \tag{4.11}$$

and therefore

$$0 = (\rho \mathbf{v_x})_1((\rho v_x)_1, T_1, \{Y_1^k\}_{k=1}^{N_g}, \{\Theta^j\}_{j=1}^{N_s})$$

At the outflow the momentum has to be zero and therefore we set

$$V_1 = 0. \tag{4.12}$$

The zero gradient condition reads

$$0 = \frac{(\rho v_x)_N - (\rho v_x)_{N-1}}{\triangle x_{N-1}} = \mathbf{\Lambda}_1((\rho v_x)_N, (\rho v_x)_{N-1}). \tag{4.13}$$

## 4.2. Index of the DAE model

In this section the index of the *DAE* is computed which has been obtained in the last section discretizing the model equations by finite differences in space. Further information about the index of a *DAE* can be found for example in [31,45]. In contrast to the system considered in [44] which is of index 2 a computation shows that the *DAE* of the last section is of index 1.

The following equations show again the dependencies of the model equations on the different variables which is a summary of the results obtained in the last section.

surface equations

$$\partial_t \Theta^k = (\mathbf{\Theta^k}) \left( T_1, \{Y_1^j\}_{j=1}^{N_g}, \{\Theta_1^j\}_{j=1}^{N_s} \right) \quad k = 1, \ldots, N_s \tag{4.14}$$

first spatial grid point

$$\partial_t Y_1^i = (\mathbf{Y_1^i}) \left( T_1, T_2, \{Y_1^k, Y_2^k\}_{k=1}^{N_g}, \{\Theta_1^j\}_{j=1}^{N_s} \right) \tag{4.15}$$

$$\partial_t T_1 = (\mathbf{T_1}) \left( T_1, T_2, \{Y_1^k, Y_2^k\}_{k=1}^{N_g}, \{\Theta_1^j\}_{j=1}^{N_s} \right) \tag{4.16}$$

$$0 = ((\rho \mathbf{v_x})_\mathbf{1}) \left( (\rho v_x)_1, T_1, \{Y_1^k\}_{k=1}^{N_g}, \{\Theta_1^j\}_{j=1}^{N_s} \right) \tag{4.17}$$

$$0 = (\mathbf{V_1}) (V_1) \tag{4.18}$$

$$0 = (\mathbf{\Lambda_1}) (\Lambda_1, \Lambda_2) \tag{4.19}$$

spatial grid points $l = 2, \ldots, N-1$

$$\partial_t Y_l^i = (\mathbf{Y_l^i}) \left( (\rho v_x)_l, T_{l-1}, T_l, T_{l+1}, \{Y_{l-1}^k, Y_l^k, Y_{l+1}^k\}_{k=1}^{N_g} \right) \tag{4.20}$$

$$\partial_t T_l = (\mathbf{T_l}) \left( (\rho v_x)_l, T_{l-1}, T_l, T_{l+1}, \{Y_{l-1}^k, Y_l^k, Y_{l+1}^k\}_{k=1}^{N_g} \right) \tag{4.21}$$

$$0 = ((\rho \mathbf{v_x})_\mathbf{l}) \left( (\rho v_x)_l, (\rho v_x)_{l+1}, V_l, T_{l-1}, T_l, T_{l+1}, \{Y_{l-1}^k, Y_l^k, Y_{l+1}^k\}_{k=1}^{N_g} \right) \tag{4.22}$$

$$0 = (\mathbf{V_l}) \left( (\rho v_x)_l, \Lambda_l, T_l, V_{l-1}, V_l, V_{l+1}, \{Y_{l-1}^k, Y_l^k, Y_{l+1}^k\}_{k=1}^{N_g} \right) \tag{4.23}$$

$$0 = (\mathbf{\Lambda_l}) (\Lambda_l, \Lambda_{l+1}) \tag{4.24}$$

last spatial grid point

$$0 = (\mathbf{Y_N^i}) \left( \{Y_N^k\}_{k=1}^{N_g} \right) \tag{4.25}$$

$$0 = (\mathbf{T_N}) (T_N) \tag{4.26}$$

$$0 = ((\rho \mathbf{v_x})_\mathbf{N}) ((\rho v_x)_N) \tag{4.27}$$

$$0 = (\mathbf{V_N}) (V_N) \tag{4.28}$$

$$0 = (\mathbf{\Lambda_l}) ((\rho v_x)_N, (\rho v_x)_{N-1}) \tag{4.29}$$

Now all the algebraic equations are differentiated once with respect to $t$. The following table then shows that it is possible to get a complete system of *ODE*. Therefore the *DAE* is of index 1.

| equation number | index | variable | |
|---|---|---|---|
| 4.14 | 0 | $\Theta^k$ | |
| 4.15 | 0 | $Y_1^i$ | |
| 4.16 | 0 | $T_1$ | |
| 4.17 | 1 | $(\rho v_x)_1$ | |
| 4.18 | 1 | $V_1$ | |
| 4.19 | 1 | $\Lambda_1$ | |
| 4.20 | 0 | $Y_l^i$ | for $l = 2, \ldots, N-1$ |
| 4.21 | 0 | $T_l$ | for $l = 2, \ldots, N-1$ |
| 4.22 | 1 | $(\rho v_x)_l$ | for $l = 2, \ldots, N-2$ |
| | 1 | $V_{N-1}$ | for $l = N-1$ |
| 4.23 | 1 | $V_l$ | for $l = 2, \ldots, N-2$ |
| | 1 | $\Lambda_{N-1}$ | for $l = N-1$ |
| 4.24 | 1 | $\Lambda_l$ | for $l = 2, \ldots, N-2$ |
| | 1 | $\Lambda_N$ | for $l = N-1$ |
| 4.25 | 1 | $Y_N^i$ | |
| 4.26 | 1 | $T_N$ | |
| 4.27 | 1 | $(\rho v_x)_N$ | |
| 4.28 | 1 | $V_N$ | |
| 4.29 | 1 | $(\rho v_x)_{N-1}$ | |

Table 4.1.: This table shows which equation gives rise to an *ODE* for a certain variable. The first column lists the equation number, the second column the index of the variable and the third column the variable for which one obtains an *ODE*.

## 4.3. Collocation Discretization of a DAE

In section 4.1 the model equations have been discretized to obtain a *DAE*. In general this can be written as

$$\begin{aligned} \dot{y} &= f(y(t), x(t)) \\ 0 &= g(y(t), x(t)) \\ 0 &= r(y(0), x(0), y(t_{\text{end}}), x(t_{\text{end}})) \end{aligned}$$

with time $t$, differential variables $y \in \mathbb{R}^{n_D}$, algebraic variables $x \in \mathbb{R}^{n_A}$, and boundary conditions $r \in \mathbb{R}^{n_D}$. A collocation method is used to transform this *DAE* to a set of nonlinear equations. The idea is to discretize the *DAE* in time using a fixed mesh $\{\tau_j\}_{j \in J}$ with $t_0 = \tau_0 \leq \cdots \leq \tau_j \leq \cdots \leq \tau_{\text{end}} = t_{\text{end}}$ and to approximate the solution on each interval $[\tau_j, \tau_{j+1}]$ with length $h_j$ with a collocation polynomial $y^\pi$(see figure 4.2) [31,48]. The collocation polynomial $y^\pi$ of degree $k$ is defined such that for $k$ a positive integer and $\rho_1, \ldots, \rho_k$ real numbers ($\in [0,1]$)

$$\begin{aligned} y^\pi(t_j) &= y_j \\ \dot{y}^\pi(t_j + \rho_i h_j) &= f(t_j + \rho_i h_j, y^\pi(t_j + \rho_i h_j), x(t_j + \rho_i h_j)) \end{aligned}$$

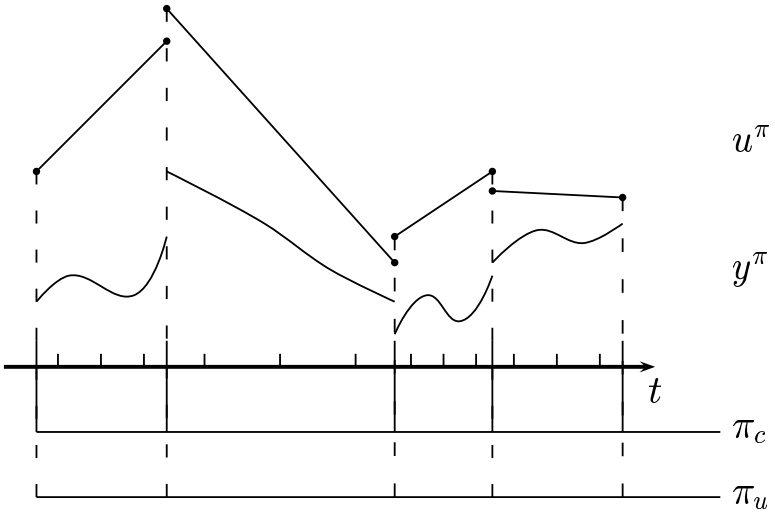


Figure 4.2.: Collocation discretization.

which means that on each interval the collocation polynomial satisfies the *DAE* at the collocation points. To compute the polynomial the following ansatz is used

$$y^\pi(t) = y_j + h_j \sum_{l=1}^{k} z_{jl} \psi_l(\frac{t-\tau_j}{h_j}) \qquad \forall t \in [\tau_j, \tau_{j+1}]$$

on each interval where $\psi_i$ is a polynomial basis of $\mathbb{P}_{k+1}[0,1]$ with

$$\psi_i(0) = 0 \quad \text{and} \quad \dot{\psi}_i(\rho_l) = \begin{cases} 1 & l = i \\ 0 & \text{else.} \end{cases}$$

The algebraic variables are discretized by the vectors

$$x_{jl} \in \mathbb{R}^{n_A}, \qquad \begin{matrix} l = 1, \ldots, k \\ j = 1, \ldots, m-1 \end{matrix}$$

representing the solution values $x(t_{jl})$ with $t_{jl} := \tau_j + \rho_j h_j$ at the collocation points. A polynomial interpolation of $\{x_{j1}, \ldots, x_{jk}\}$ yields an approximation $x^\pi(t)$ in the whole interval $[\tau_j, \tau_{j+1}]$. The discretized form of the *DAE* is then

$$\begin{aligned} z_{jl} &= f\left(y_j + h_j \sum_{s=1}^{k} z_{js} \psi_s(\rho_l), x_{jl}\right) \\ 0 &= g\left(y_j + h_j \sum_{s=1}^{k} z_{js} \psi_s(\rho_l), x_{jl}\right) \end{aligned} \qquad \text{for} \quad \begin{matrix} j = 1, \ldots, m-1 \\ l = 1, \ldots, k \end{matrix} \tag{4.30}$$

and additionally the continuity conditions

$$y_j + h_j \sum_{s=1}^{k} z_{js} \psi_s(1) - y_{j+1} = 0 \tag{4.31}$$

and discretized boundary constraints

$$0 = r(y_1, x^\pi(0), y_m, x^\pi(t_{\text{end}}))$$

Further details about this collocation approach can be found in [48] and more details about collocation in for example [2–4, 31].

## 4.4. The Optimal Control Algorithm

In sections 4.1 and 4.2 the model equations, a set of 1 dimensional transient partial-differential equations have been discretized in space and time to obtain a set of non-linear equations. The optimal control problem is therefore transformed into a large scale *structured* optimization problem. 'Structured' in this context means the special structures one obtains using specific discretizations, in this case finite differences for the space and collocation for the time dependence. This section describes some parts of the implementation of the partially reduced *SQP*-method in *OCPRSQP* in more detail. The algorithm solves a nonlinear optimal control problem of the form

$$\begin{gathered}
\min \phi(y(t_{\text{end}}), x(t_{\text{end}})) \\
\dot{y} = f(y(t), x(t), u(t)) \\
0 = g(y(t), x(t), u(t)) \\
r(y(0), x(0), y(t_{\text{end}}), x(t_{\text{end}})) = 0 \\
s(y(t), x(t), u(t)) \geq 0 \\
u_i^{\min} \leq u_i \leq u_i^{\max} \qquad i = 1, \ldots, n_U
\end{gathered}$$

in which $s$ represents the path constraints and the last expression bounds on the controls. The *DAE* is now discretized with a collocation method as described in section 4.3 and for ease of presentation we assume that the control is always constant on each interval $[\tau_j, \tau_{j+1}]$. Furthermore we assume that the path constraints and the bounds for the controls have only to be satisfied at the collocation mesh points $\tau_j$. This transforms the whole problem into a high dimensional optimization problem of the form

$$\min \phi(y_m, x^\pi(t_{\text{end}})) \tag{4.32}$$

subject to the constraints — the collocation conditions $c^{\text{col}}$ —

$$c_{jl}^{\text{col}} = \begin{cases} f\left(y_j + h_j \sum_{s=1}^k z_{js}\psi_s(\rho_l), x_{jl}, u_{jl}\right) - z_{jl} = 0 \\ g\left(y_j + h_j \sum_{s=1}^k z_{js}\psi_s(\rho_l), x_{jl}, u_{jl}\right) = 0 \end{cases} \quad \text{for} \quad \begin{matrix} j = 1, \ldots, m-1 \\ l = 1, \ldots, k \end{matrix} \tag{4.33}$$

the continuity conditions $c^{\text{con}}$

$$c_j^{\text{con}} = y_j + h_j \sum_{s=1}^k z_{js}\psi_s(1) - y_{j+1} = 0 \qquad \text{for} \quad j = 1, \ldots, m-1 \tag{4.34}$$

the discretized boundary constraints

$$r(y_1, x^\pi(0), y_m, x^\pi(t_{\text{end}})) = 0 \tag{4.35}$$

the bounds for the controls

$$u_j^{\min} \le u_j \le u_j^{\max} \qquad \text{for} \quad j = 1, \ldots, m-1 \tag{4.36}$$

and the discretized path constraints

$$s(y_j, x^\pi(\tau_j), u^\pi(\tau_j)) \le 0 \qquad \text{for} \quad j = 1, \ldots, m \tag{4.37}$$

The details concerning the theoretical foundations of the partially reduced *SQP*-method can be found in V. Schulz thesis [48]. Further information about the (reduced) *SQP*-method can be found for example in [7, 14, 15, 27–30, 32, 35, 49]

For a closer look at some parts of the algorithm it is convenient to introduce a grouping of the variables

$$\begin{aligned} \xi_j &= \begin{bmatrix} z_{j1} & \ldots & z_{jk} \end{bmatrix} \\ \underline{y} &= \begin{bmatrix} \xi_1 & y_2 & \xi_2 & y_3 & \ldots & \xi_{m-1} & y_m \end{bmatrix} \\ \underline{u} &= \begin{bmatrix} y_1 & u_1 & u_2 & \ldots & u_{m-1} \end{bmatrix} \end{aligned}$$

the collocation and continuity conditions

$$\begin{aligned} c_j^{\text{col}} &= \begin{bmatrix} c_{j_1}^{\text{col}} & \ldots & c_{j_k}^{\text{col}} \end{bmatrix} \\ c &= \begin{bmatrix} c_1^{\text{col}} & c_1^{\text{con}} & c_2^{\text{col}} & c_2^{\text{con}} & \ldots & c_{m-1}^{\text{col}} & c_{m-1}^{\text{con}} \end{bmatrix} \end{aligned}$$

and the discretized boundary conditions and path constraints

$$d = \begin{bmatrix} d_1 & d_2 & \ldots & d_{m-1} & d_m \end{bmatrix}$$

in which $d_j$ is the collection of boundary constraints and path constraints at the $j$-th collocation mesh point.[1] The original version of the algorithm keeps the initial values of the *DAE* as possible controls to the problem. The formal partially reduced *SQP*-algorithm is presented in figure 4.3. Computing the derivatives of the constraints $C_{\underline{y}} := \frac{\partial c}{\partial \underline{y}}$, $C_{\underline{u}} := \frac{\partial c}{\partial \underline{u}}$, $D_{\underline{y}} := \frac{\partial d}{\partial \underline{y}}$, and $D_{\underline{u}} := \frac{\partial d}{\partial \underline{u}}$, computing $C_{\underline{y}}^{-1}$ and $C_{\underline{y}}^{-T}$ and solving the *QP*-subproblem seem to be promising starting points for the reduction of computing time. Therefore a closer look is taken on these points. A glance at the collocation conditions 4.33, and the continuity conditions 4.34 reveals $C_{\underline{y}}$ and $C_{\underline{u}}$ have a certain block structure which is shown in figures 4.4 and 4.5 The matrices $W_j$, $V_j$, and $F_j$ are the derivatives of the collocation equations (4.33) with respect to $\xi_i$, $y_i$, $u_i$:

$$W_j := \frac{\partial c_j^{\text{col}}}{\partial \xi_j} = \begin{bmatrix} W_{j11} & \cdots & W_{j1k} \\ \vdots & & \vdots \\ W_{jk1} & \cdots & W_{jkk} \end{bmatrix} \tag{4.38}$$

[1] Actually it makes only sense to speak about the boundary constraints for $j = 1$ or $j = m$, in all other cases there are only path constraints to be considered.

(1) evaluate all constraints, the objective criterion
(2) determine the derivatives of the objective criterion:

$$\nabla_{\underline{y}}\Phi \quad \text{and} \quad \nabla_{\underline{u}}\Phi$$

and of the constraints

$$C_{\underline{y}} := \frac{\partial c}{\partial \underline{y}}, \quad C_{\underline{u}} := \frac{\partial c}{\partial \underline{u}}, \quad D_{\underline{y}} := \frac{\partial d}{\partial \underline{y}}, \quad D_{\underline{u}} := \frac{\partial d}{\partial \underline{u}}$$

(3) compute the reduced gradient

$$\gamma := \nabla_{\underline{u}}\Phi - C_{\underline{u}}^T C_{\underline{y}}^{-T} \nabla_{\underline{y}}\Phi$$

(4) solve the quadratic program

$$\min_{\triangle \underline{u}} \triangle \underline{u}^T B_k \triangle \underline{u} + \gamma^T \triangle \underline{u}$$
$$(D_{\underline{u}} - D_{\underline{y}} C_{\underline{y}}^{-1} C_{\underline{u}}) \triangle \underline{u} + D_{\underline{y}} C_{\underline{y}}^{-1} c = 0$$

in order to obtain $\triangle \underline{u}$ and the adjoint variables $\lambda_d$
(5) determine the null space step $\triangle \underline{y}^{\mathcal{N}}$

$$\triangle \underline{y}^{\mathcal{N}} := -C_{\underline{y}}^{-1} C_{\underline{u}} \triangle \underline{u}$$

the range space step $\triangle \underline{y}^{\mathcal{R}}$

$$\triangle \underline{y}^{\mathcal{R}} := -C_{\underline{y}}^{-1} c$$

and the adjoint variables $\lambda_c$

$$\lambda_c = -C_{\underline{y}}^{-T} (D_{\underline{y}}^T \lambda_d + \nabla_{\underline{y}}\Phi)$$

(6) perform a line search and get step length $\alpha$
(7) update of the Hessian
(8) iterate $y = y + \alpha(\triangle \underline{y}^{\mathcal{R}} + \triangle \underline{y}^{\mathcal{N}})$ and $u = u + \alpha(\triangle \underline{u}^{\mathcal{N}})$
(9) $k = k + 1$, go to (1) until convergence

Figure 4.3.: PRSQP algorithm

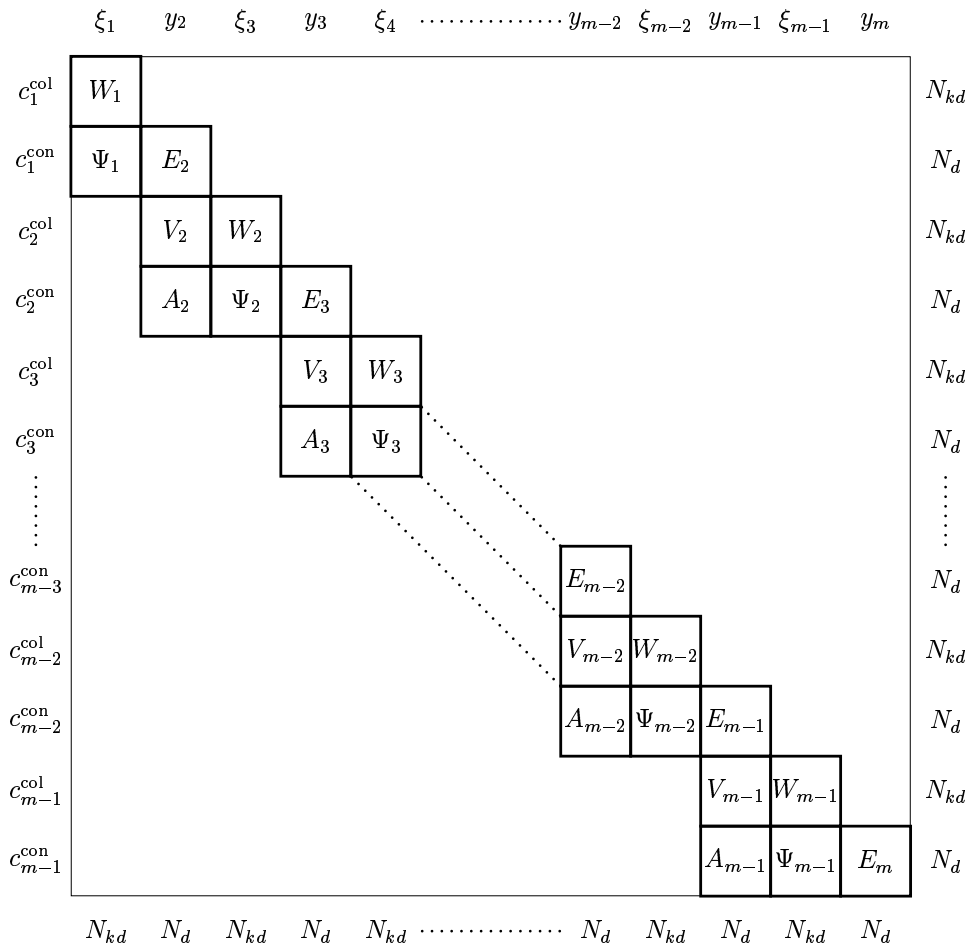


Figure 4.4.: Block structure of $C_{\underline{y}}$. $N_d$ is the number of dynamic equations, that is the number of differential plus algebraic equations. $N_{kd}$ is $k$ times $N_d$, $k$ is the number of collocation points per interval.

with

$$W_{jlm} := \frac{\partial c_{jl}^{\mathrm{col}}}{\partial \xi_{jm}} = -\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} + h_j \psi_m(\rho_l) \begin{bmatrix} \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial y} \end{bmatrix}_{x_j + \rho_l h_j} \quad \text{in} \quad [x_j, x_{j+1}] \tag{4.39}$$

$$V_j := \frac{\partial c_j^{\mathrm{col}}}{\partial y_j} = \begin{bmatrix} V_{j1} \\ \vdots \\ V_{jk} \end{bmatrix} \quad \text{with} \quad V_{jl} = \begin{bmatrix} \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial y} \end{bmatrix}_{x_j + \rho_l h_j} \quad \text{in} \quad [x_j, x_{j+1}]$$

$$F_j := \frac{\partial c_j^{\mathrm{col}}}{\partial u_j} = \begin{bmatrix} F_{j1} \\ \vdots \\ F_{jk} \end{bmatrix} \quad \text{with} \quad F_{jl} = \begin{bmatrix} \frac{\partial f}{\partial u} \\ \frac{\partial g}{\partial u} \end{bmatrix}_{x_j + \rho_l h_j} \quad \text{in} \quad [x_j, x_{j+1}]$$

The matrices $A_j, E_j$ and $\Psi_j$ result from the derivatives of the continuity condi-

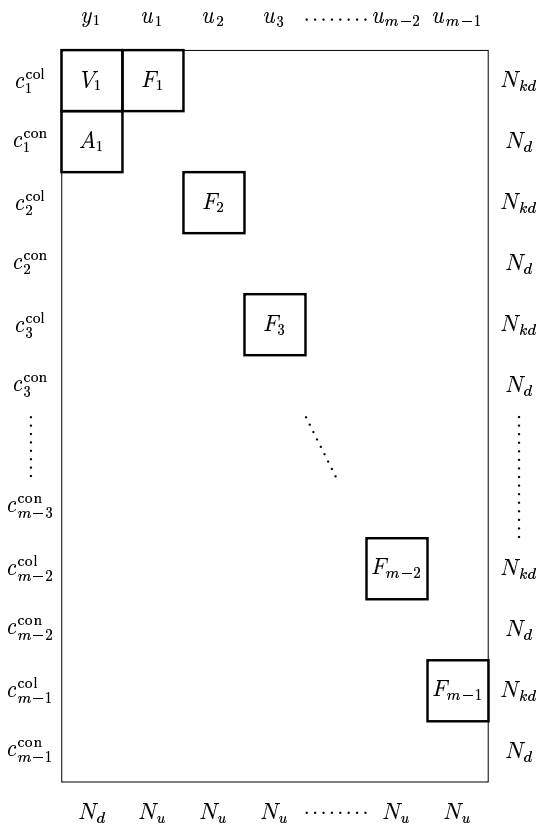


Figure 4.5.: Block structure of $C_{\underline{u}}$. $N_d$ is the number of dynamic equations, that is the number of differential plus algebraic equations. $N_{kd}$ is $k$ times $N_d$, $k$ is the number of collocation points per interval.

tions (4.34) and the first two are given by

$$A_j := \frac{\partial c_j^{\text{con}}}{\partial y_j} = I$$
$$E_j := \frac{\partial c_j^{\text{con}}}{\partial y_{j+1}} = -I$$

whereas the matrix $\Psi_j$ has a slightly more complicated form, namely

$$\Psi_j := \frac{\partial c_j^{\text{con}}}{\partial \xi} = h_j \left[ \begin{array}{c|c|c} \psi_1(1)I & \dots & \psi_k(1)I \end{array} \right]$$

with $I$ the $N_{kd}$ dimensional unit matrix. Because of its particular structure $\Psi_j$ has not to be stored explicitly and multiplication with a matrix $A$ is done easily through the formula

$$\Psi_j A = \Psi_j \begin{bmatrix} A_1 \\ \vdots \\ A_k \end{bmatrix} = h_j \left[ \sum_{s=1}^{k} \psi_s(1) A_s \right]$$

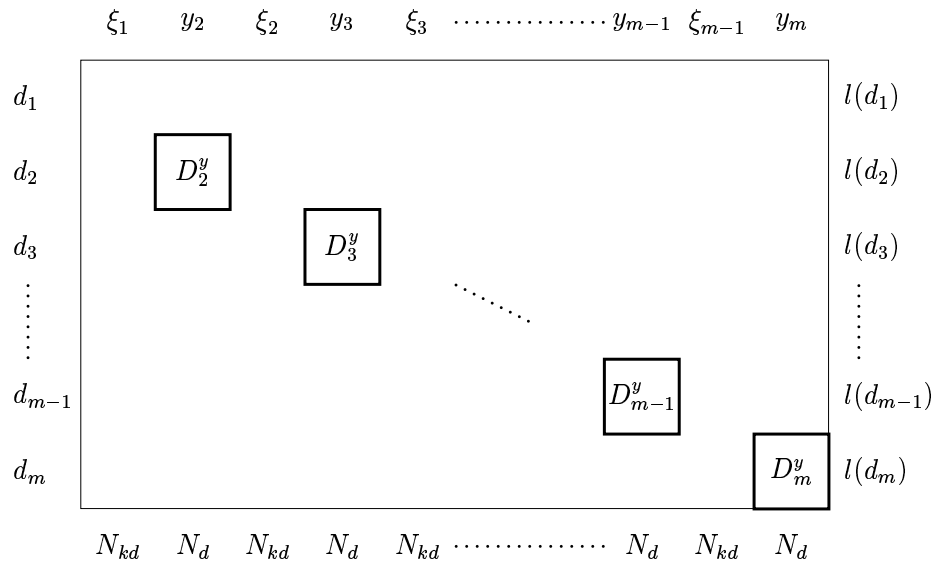


Figure 4.6.: Block structure of $D_{\underline{y}}$. $N_d$ is the number of dynamic equations, that is the number of differential plus algebraic equations. $N_{kd}$ is $k$ times $N_d$, $k$ is the number of collocation points per interval. $l(d_j)$ is the number of constraints (boundary + path) at the $j$-th collocation mesh point.

At each collocation mesh point $j$ one has a certain number of path constraints $d_j$.[2] In the following it is assumed that $d_j$ only depends on $y_j$ and $u_j$ (and at $j = m$ only on $y_m$) which gives the structure for $D_{\underline{y}}$ and $D_{\underline{u}}$ shown in figures 4.6 and 4.7 where $D_j^y := \frac{\partial d_j}{\partial y}$ and $D_j^u := \frac{\partial d_j}{\partial u}$ is used.

### Recursion Formulas

It has been shown in the last section that the matrices $C_{\underline{y}}$, $C_{\underline{u}}$, $D_{\underline{y}}$, $D_{\underline{u}}$ posses an rich internal structure. Taking a closer look at the partially reduced $SQP$ algorithm shown in figure 4.3 reveals that $C_{\underline{y}}^{-1}$ and $C_{\underline{y}}^{-T}$ have to be computed in combination with matrix multiplications. In the following a set of recursion formulas is presented which uses this particular structure of $C_{\underline{y}}$ to obtain $C_{\underline{y}}^{-1}$ (or $C_{\underline{y}}^{-T}$) in a very efficient way. Two vectors $x$ and $c$ are defined with the following internal grouping

$$x := \begin{bmatrix} \xi_1 & y_2 & \xi_2 & \dots & \xi_{m-1} & y_m \end{bmatrix}$$
$$c := \begin{bmatrix} c_1^{\text{col}} & c_1^{\text{con}} & \dots & c_{m-1}^{\text{col}} & c_{m-1}^{\text{con}} \end{bmatrix}$$

in which the variables $\xi_i$, $y_i$, $c_i^{\text{col}}$ and $c_i^{\text{con}}$ indicate the dimension of the different parts. First the system $C_{\underline{y}} x = c$ is to be solved. The block structure of $C_{\underline{y}}$ shown in

[2] And possibly boundary constraints if $j = 1$ or $j = m$

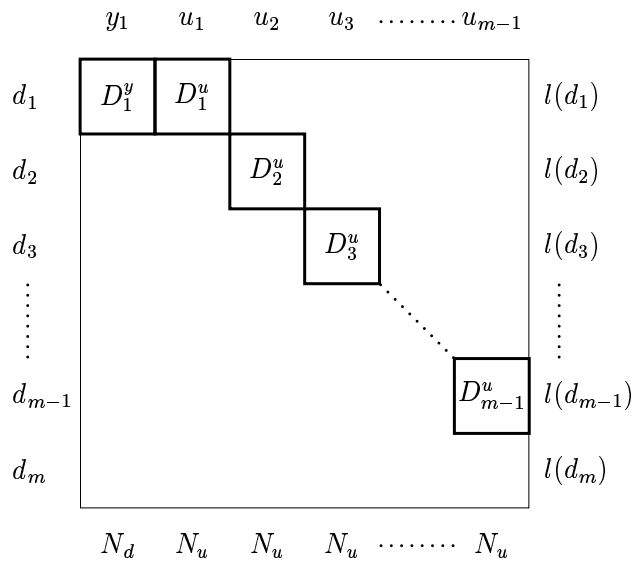


Figure 4.7.: Block structure of $D_{\underline{u}}$. $N_d$ is the number of dynamic equations, that is the number of differential plus algebraic equations. $N_u$ is the number of controls. $l(d_j)$ is the number of constraints (boundary + path) at the $j$-th collocation mesh point.

figure 4.4 leads to the recursion formula

$$\left.\begin{aligned} \xi_1 &= W_1^{-1} c_1^{\text{col}} \\ y_2 &= \Psi_1 \xi_1 - c_1^{\text{con}} \\ \xi_j &= W_j^{-1} \left( c_j^{\text{col}} - V_j y_j \right) \\ y_{j+1} &= y_j + \Psi_j \xi_j - c_j^{\text{con}} \end{aligned}\right\} \quad j = 2, \ldots, m-1$$

When solving $C_y^T c = x$ the block structure of $C_y^T$ leads to

$$\left.\begin{aligned} c_{m-1}^{\text{con}} &= -y_m \\ c_{m-1}^{\text{col}} &= W_{m-1}^{-T} \left( \xi_{m-1} - \Psi_{m-1}^T c_{m-1}^{\text{con}} \right) \\ c_j^{\text{con}} &= c_{j+1}^{\text{con}} + V_{j+1}^T c_{j+1}^{\text{col}} - y_{j+1} \\ c_j^{\text{col}} &= W_j^{-T} \left( \xi_j - \Psi_j^T c_j^{\text{con}} \right) \end{aligned}\right\} \quad j = m-2, \ldots, 1$$

The algorithm to compute $D_{\underline{y}} - D_{\underline{u}} C_{\underline{y}}^{-1} C_{\underline{u}}$ is more complex than the previous ones. The idea is to first solve the system $C_{\underline{y}} X = C_{\underline{u}}$ and then multiplying $X$ by $-D_{\underline{u}}$ and adding to $D_{\underline{y}}$. $\bar{D}_j^y$ is defined to be a column 'vector' with $-D_j^y$ ($D_j^y$ if $j = 1$) at the 'j-th' position, that is equal to the matrix we obtain if we take the columns of $D_{\underline{y}}$ (or $D_{\underline{u}}$ for $j = 1$) containing $D_j^y$ and multiplying by $-1$ if $j \neq 1$. The same is done to define $\bar{D}_j^u$ but without the multiplication by $-1$. This leads to a recursion of the

form

$$[\ \bar{D}_j^y \ \big|\ \bar{D}_j^u\ ] := [\ \bar{D}_j^y \ \big|\ \bar{D}_j^u\ ] + \bar{D}_{j+1}^y[\ I - \Psi_j W_j^{-1} V_j \ \big|\ \Psi_j W_j^{-1} F_j\ ] \quad j = m-1, \ldots, 2$$
$$[\ \bar{D}_1^y \ \big|\ \bar{D}_1^u\ ] := [\ \bar{D}_1^y \ \big|\ \bar{D}_1^u\ ] + \bar{D}_2^y[\ -I + \Psi_j W_j^{-1} V_1 \ \big|\ \Psi_1 W_1^{-1} F_1\ ]$$

Since this is not obvious it is illustrate with a simple example with four collocation mesh points. The first step is to solve $C_{\underline{y}} X = C_{\underline{u}}$, that is

$$\begin{bmatrix} W_1 & & & & & \\ \Psi_1 & -I & & & & \\ & V_2 & W_2 & & & \\ & I & \Psi_2 & -I & & \\ & & & V_3 & W_3 & \\ & & & I & \Psi_3 & -I \end{bmatrix} \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} \\ X_{31} & X_{32} & X_{33} & X_{34} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} \\ X_{41} & X_{42} & X_{43} & X_{44} \end{bmatrix} = \begin{bmatrix} V_1 & F_1 & & \\ I & & & \\ & & F_2 & \\ & & & \\ & & & F_3 \end{bmatrix}$$

which gives the solution

$$\begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} \\ X_{31} & X_{32} & X_{33} & X_{34} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} \\ X_{41} & X_{42} & X_{43} & X_{44} \end{bmatrix} = \begin{bmatrix} W_1^{-1}V_1 & W_1^{-1}F_1 & 0 & 0 \\ -I + \Psi_1 Y_{11} & \Psi_1 Y_{12} & 0 & 0 \\ -W_2^{-1}V_2 X_{21} & -W_2^{-1}V_2 X_{22} & W_2^{-1}F_2 & 0 \\ X_{21} + \Psi_2 Y_{21} & X_{22} + \Psi_2 Y_{22} & \Psi_2 Y_{23} & 0 \\ -W_3^{-1}V_3 X_{31} & -W_3^{-1}V_3 X_{32} & -W_3^{-1}V_3 X_{33} & W_3^{-1}F_3 \\ X_{31} + \Psi_3 Y_{31} & X_{32} + \Psi_3 Y_{32} & X_{33} + \Psi_3 Y_{33} & \Psi_3 Y_{34} \end{bmatrix}$$

Multiplying $X$ by $-D_{\underline{y}}$ and adding to $D_{\underline{u}}$ yields

$$\begin{bmatrix} D_1^y & D_1^u & 0 & 0 \\ -D_2^y X_{21} & -D_2^y X_{22} & D_2^u - D_2^y X_{23} & -D_2^y X_{24} \\ -D_3^y X_{31} & -D_3^y X_{32} & -D_3^y X_{33} & D_3^u - D_3^y X_{34} \\ -D_4^y X_{41} & -D_4^y X_{42} & -D_4^y X_{43} & -D_4^y X_{44} \end{bmatrix}$$

Since only the $X_{ij}$ entries are needed one can define $X'$

$$X' := \begin{bmatrix} X_{21} & X_{22} & X_{23} & X_{24} \\ X_{31} & X_{32} & X_{33} & X_{34} \\ X_{41} & X_{42} & X_{43} & X_{44} \end{bmatrix}$$

which is then

$$X' = \begin{bmatrix} -I + \Psi_1 W_1^{-1} V_1 & \Psi_1 W_1^{-1} F_1 & 0 & 0 \\ [I - \Psi_2 W_2^{-1} V_2] X_{21} & [I - \Psi_2 W_2^{-1} V_2] X_{22} & \Psi_2 W_2^{-1} F_2 & 0 \\ [I - \Psi_3 W_3^{-1} V_3] X_{31} & [I - \Psi_3 W_3^{-1} V_3] X_{32} & [I - \Psi_3 W_3^{-1} V_3] X_{33} & \Psi_3 W_3^{-1} F_3 \end{bmatrix}$$

Therefore $D_{\underline{u}} - D_{\underline{y}} C_{\underline{y}}^{-1} C_{\underline{u}}$ can be written as

$$\begin{bmatrix} D_1^y & D_1^u & & \\ -D_2^y(-I + \Psi_1 W_1^{-1} V_1) & -D_2^y \Psi_1 W_1^{-1} F_1 & & \\ -D_3^y[I - \Psi_2 W_2^{-1} V_2] X_{21} & -D_3^y[I - \Psi_2 W_2^{-1} V_2] X_{22} & & \\ -D_4^y[I - \Psi_3 W_3^{-1} V_3] X_{31} & -D_4^y[I - \Psi_3 W_3^{-1} V_3] X_{32} & & \\ & & 0 & 0 \\ & & D_2^u & 0 \\ & & -D_3^y \Psi_2 W_2^{-1} F_2 & D_3^u \\ & & -D_4^y[I - \Psi_3 W_3^{-1} V_3] X_{33} & -D_4^y \Psi_3 W_3^{-1} F_3 \end{bmatrix} \tag{4.40}$$

This form of the solution leads to a natural recursion which is illustrate in the following. One starts with the matrices $\bar{D}_i^y$ and $\bar{D}_j^u$ which are arranged in the following order

| $\bar{D}_4^y$ | $\bar{D}_3^y$ | $\bar{D}_2^y$ | $\bar{D}_1^y$ | $\bar{D}_1^u$ | $\bar{D}_2^u$ | $\bar{D}_3^u$ |
|---|---|---|---|---|---|---|
| | | | $D_1^y$ | $D_1^u$ | | |
| | | $-D_2^y$ | | | $D_2^u$ | |
| | $-D_3^y$ | | | | | $D_3^u$ |
| $-D_4^y$ | | | | | | |

Now an update of the columns $\bar{D}_3^y$ and $\bar{D}_3^u$ is computed

$$\bar{D}_3^y = \bar{D}_3^y + \bar{D}_4^y[I - \Psi_3 W_3^{-1} V_3] \qquad \bar{D}_3^u = \bar{D}_3^u + \bar{D}_4^y[\Psi_3 W_3^{-1} F_3]$$

and one obtains

| $\bar{D}_3^y$ | $\bar{D}_2^y$ | $\bar{D}_1^y$ | $\bar{D}_1^u$ | $\bar{D}_2^u$ | $\bar{D}_3^u$ |
|---|---|---|---|---|---|
| | | $D_1^y$ | $D_1^u$ | | |
| | $-D_2^y$ | | | $D_2^u$ | |
| $-D_3^y$ | | | | | $D_3^u$ |
| $-D_4^y[I - \Psi_3 W_3^{-1} V_3]$ | | | | | $-D_4^y[\Psi_3 W_3^{-1} F_3]$ |

in which the column $\bar{D}_4^y$ is discarded because it has not changed and it is not needed any further. A glance at the solution 4.40 shows that $\bar{D}_3^u$ now contains the last column of the solution 4.40. It is not going to change nor contribute any further and therefore it is not shown in the further steps. The next step computes updates to $\bar{D}_2^y$ and $\bar{D}_2^u$

$$\bar{D}_2^y = \bar{D}_2^y + \bar{D}_3^y[I - \Psi_2 W_2^{-1} V_2] \qquad \bar{D}_2^u = \bar{D}_2^u + \bar{D}_3^y[\Psi_2 W_2^{-1} F_2]$$

and one obtains

| $\bar{D}_2^y$ | $\bar{D}_1^y$ | $\bar{D}_1^u$ | $\bar{D}_2^u$ |
|---|---|---|---|
| | $D_1^y$ | $D_1^u$ | |
| $-D_2^y$ | | $D_2^u$ | |
| $-D_3^y[I - \Psi_2 W_2^{-1} V_2]$ | | | $-D_3^y[\Psi_2 W_2^{-1} F_2]$ |
| $-D_4^y[I - \Psi_3 W_3^{-1} V_3][I - \Psi_2 W_2^{-1} V_2]$ | | | $-D_4^y[I - \Psi_3 W_3^{-1} V_3][\Psi_2 W_2^{-1} F_2]$ |

The updated column $\bar{D}_2^u$ now contains the third column of the solution 4.40. In the final step of this example updates to $\bar{D}_1^y$ and $\bar{D}_1^u$ are computed.

$$\bar{D}_1^y = \bar{D}_1^y + \bar{D}_2^y[I - \Psi_1 W_1^{-1} V_1] \qquad \bar{D}_1^u = \bar{D}_1^u + \bar{D}_2^y[\Psi_1 W_1^{-1} F_1]$$

One obtains

| $\bar{D}_1^y$ |
|---|
| $D_1^y$ |
| $-D_2^y[-I + \Psi_1 W_1^{-1} V_1]$ |
| $-D_3^y[I - \Psi_2 W_2^{-1} V_2][-I + \Psi_1 W_1^{-1} V_1]$ |
| $-D_4^y[I - \Psi_3 W_3^{-1} V_3][I - \Psi_2 W_2^{-1} V_2][-I + \Psi_1 W_1^{-1} V_1]$ |

| $\bar{D}_1^u$ |
|---|
| $D_1^u$ |
| $-D_2^y[\Psi_1 W_1^{-1} F_1]$ |
| $-D_3^y[I - \Psi_2 W_2^{-1} V_2][\Psi_1 W_1^{-1} F_1]$ |
| $-D_4^y[I - \Psi_3 W_3^{-1} V_3][I - \Psi_2 W_2^{-1} V_2][\Psi_1 W_1^{-1} F_1]$ |

$\bar{D}_1^y$ and $\bar{D}_1^u$ contain the first and second column of the solution 4.40. This example calculation leads to the recursion

$$\begin{aligned}
&[\ \bar{D}_j^y \ \big|\ \bar{D}_j^u\ ] := [\ \bar{D}_j^y \ \big|\ \bar{D}_j^u\ ] + \bar{D}_{j+1}^y [\ I - \Psi_j W_j^{-1} V_j \ \big|\ \Psi_j W_j^{-1} F_j\ ] \quad j = m-1, \ldots, 2 \\
&[\ \bar{D}_1^y \ \big|\ \bar{D}_1^u\ ] := [\ \bar{D}_1^y \ \big|\ \bar{D}_1^u\ ] + \bar{D}_2^y [\ -I + \Psi_j W_j^{-1} V_1 \ \big|\ \Psi_1 W_1^{-1} F_1\ ]
\end{aligned}$$

which leads to the matrix

$$\begin{bmatrix} \bar{D}_1^y & \bar{D}_1^u & \bar{D}_2^u & \ldots & \bar{D}_{m-1}^u \end{bmatrix}$$

It can explicitly be written in a closed form as

$$\begin{bmatrix}
D_1^y & D_1^u & 0 & 0 & \ldots & 0 \\
D_2^y Z_{21} & D_2^y Z_{22} & D_2^u & 0 & \ldots & 0 \\
D_3^y Z_{31} & D_3^y Z_{32} & D_3^y Z_{33} & D_3^u & \ddots & 0 \\
\vdots & & & & & \vdots \\
D_{m-1}^y Z_{m-1,1} & D_{m-1}^y Z_{m-1,2} & D_{m-1}^y Z_{m-1,3} & D_{m-1}^y Z_{m-1,4} & \ldots & D_{m-1}^u \\
D_m^y Z_{m,1} & D_m^y Z_{m,2} & D_m^y Z_{m,3} & D_m^y Z_{m,4} & \ldots & D_m^y Z_{m,m}
\end{bmatrix}$$

with

$$\begin{aligned}
Z_{i,1} &= -\left[\prod_{k=3}^{i} (I - \Psi_{i-k+2} W_{i-k+2}^{-1} V_{i-k+2})\right] \left[-I + \Psi_1 W_1^{-1} V_1\right] \qquad i \geq 2 \\
Z_{i,j} &= -\left[\prod_{k=j+1}^{i} (I + \Psi_{i-k+2} W_{i-k+2}^{-1} V_{i-k+2})\right] \Psi_{j-1} W_{j-1}^{-1} F_{j-1} \qquad i, j \geq 2
\end{aligned}$$

but the recursion formula computes this matrix much more efficiently then the explicit computation with this product formula.

# 5. New Strategies

A straightforward application of the optimal control package *OCPRSQP* in combination with the simulation tool *DIFRUN* was not possible. Several new developments, changes and modifications to *OCPRSQP* were necessary to obtain a working software package and to reduce the computing times to a reasonable level. In this chapter these new developments, changes and modifications are presented and explained.

The first section describes the design and development of the new package based on *DIFRUN* and *OCPRSQP*.

The finite differences used so far in *OCPRSQP* to obtain derivatives do not yield enough accuracy in this problem class. An interface to use analytic derivatives has been implemented. These derivatives are computed with *ADIFOR*. The second section shows how to speed up the computation using these derivatives.

One of the most time consuming steps during the optimization is the condensing step. The development of a fast direct band solver is described in the third section. This solver is optimized to take advantage of a block band structure of the linear systems which have to be solved.

A much deeper change of the optimization can be done considering the condensing itself. In the original version of *OCPRSQP* the QP-variables have essentially been the initial values and the controls. In this problem class the initial values are always fixed. Therefore one can consider QP-problems which have only the controls as free variables which reduces the size of the QP-problem and therefore reduces the computing time. It is shown how the recursion formulas have to be changed for this new situation.

The convention is used that names in small capitals are subroutines. So *OCPRSQP* denotes the whole software package whereas OCPRSQP is just a subroutine.

## 5.1. A New Software Tool based on DIFRUN and OCPRSQP

The development is based on two existing software packages: *DIFRUN* a simulation tool and *OCPRSQP* an optimal control package.

Formally these two packages fit together very well. *DIFRUN* is a tool which is very flexible to use to test new chemical systems in a reactive stagnation point flow on a catalytic plate setting. The package already contains the spatial discretized form of the model equations since it uses a modified version of *LIMEX* [5, 38]—a *DAE* solver—for the simulation. *OCPRSQP* on the other hand is an optimal control package which allows *DAE* as constraints and is especially well suited to this situation in which there are a very large number of problem variables and constraints (around 100–1000) but only a small number of controls (1–3).

| Number of variables | Contents |
|---|---|
| $N_s$ | surface coverages $\Theta^j$ |
| $N_g$ | species $Y_1^i$ at first grid point |
| 1 | temperature $T_1$ at first grid point |
| 1 | mass $(\rho v_x)_1$ at first grid point |
| 1 | momentum $V_1$ at first grid point |
| 1 | $\Lambda_1$ at first grid point |
| $\vdots$ | $\vdots$ |
| $N_g$ | species $Y^i_{m_{\text{spa}}}$ at grid point $m_{\text{spa}}$ |
| 1 | temperature $T_{m_{\text{spa}}}$ at grid point $m_{\text{spa}}$ |
| 1 | mass $(\rho v_x)_{m_{\text{spa}}}$ at grid point $m_{\text{spa}}$ |
| 1 | momentum $V_{m_{\text{spa}}}$ at grid point $m_{\text{spa}}$ |
| 1 | $\Lambda_{m_{\text{spa}}}$ at grid point $m_{\text{spa}}$ |

Table 5.1.: Ordering of the discretized variables and equations in *DIFRUN*.

Like in many other cases the problems are due to the practical realizations. Since a modified version of *LIMEX* is used to solve the *DAE* system in *DIFRUN* there is no standard form of an interface for function evaluations of the right hand side. Standard form means just the dimensions of the problem and values of the variables as input and the function values as output. Moreover the excessive use of "common blocks" makes it even more difficult to obtain such an interface. The first step has therefore been to provide a new subroutine ODE for all the necessary setup to call the right hand side subroutine DFRFCN in *DIFRUN* for function evaluations in a standardized way.

This new interface has been tested using a standard form of *LIMEX* [16] which confirmed the results already obtained. Since *OCPRSQP* uses a collocation method for the time discretization the *RADAU* [31] solver based on collocation has been used and again very satisfactory results have been obtained.

Another problem for the coupling was the ordering of differential and algebraic equations. In *DIFRUN* the algebraic equations are marked by an additional array whereas in the original *OCPRSQP* a fixed ordering—first differential and the algebraic equations—has been used. All the relevant subroutines of *OCPRSQP* have been changed to adapt to the way *DIFRUN* handles this ordering. This is done because the right hand sides coded in *DIFRUN* should be taken as they are and a manual reordering of variables and functions in the subroutine ODE is just too time consuming. Furthermore the ordering of variables in *DIFRUN* reveals a block band structure of the Jacobian when computing the derivatives of the right hand sides. The ordering of the variables is shown in table 5.1 and corresponds to the ordering of the equations.

Since all the parameters needed to compute the right hand side are kept in "common blocks" and since a minimal coupling between the source of *DIFRUN* and the optimization package is desired, a point in *DIFRUN* has been selected to call an interface subroutine INTERFACE which does all the setup for OCPRSQP. With such a minimal coupling it is very easy to update *DIFRUN* with newer versions with minimal effort if necessary. This approach is illustrated in figure 5.1. One could view

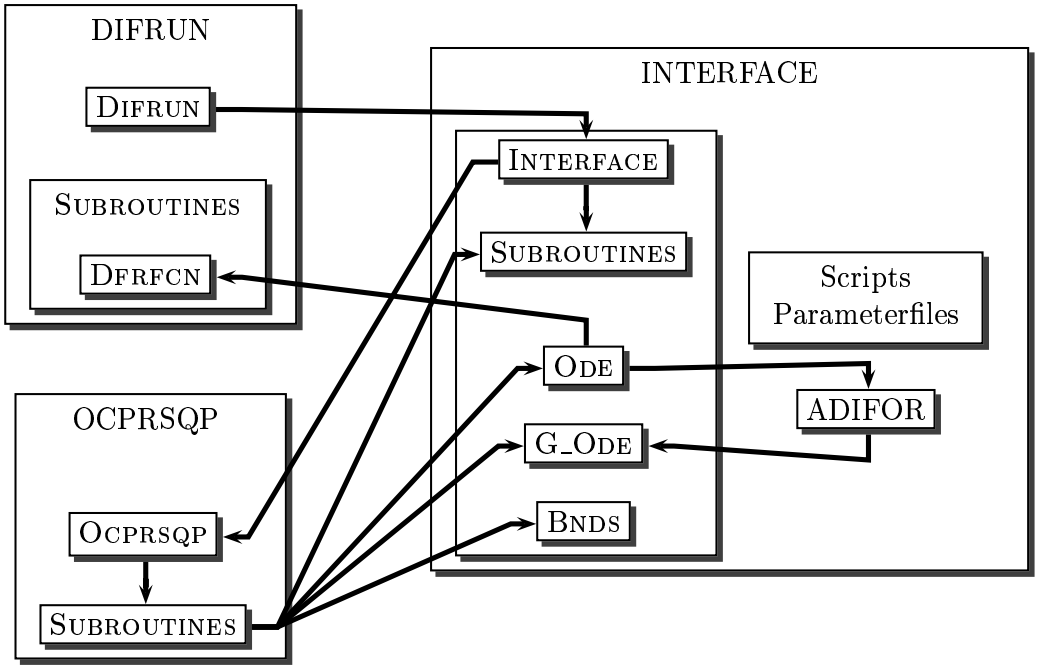


Figure 5.1.: Coupling between the software packages *DIFRUN* and *OCPRSQP*.

the new package as an extension to *DIFRUN* and setting the appropriate flag in INTERFACE, the new package remains just a simulation tool. *DIFRUN* provides all the initialization for the chemical system and the calls the subroutine INTERFACE. This subroutine provides the setup for OCPRSQP which is then called for the optimization. The whole interface package includes some additional subroutines which are called from INTERFACE or from subroutines of *OCPRSQP*, the ODE subroutine to provide access to the right hand side DFRFCN in *DIFRUN*, G_ODE which is the analytic derivative of ODE obtained with *ADIFOR* and BNDS for the path constraints of the problem.

Computational tests showed that the finite difference approach used in the original *OCPRSQP* cannot be used for this new problem class. So it has been decided to provide analytic derivatives using *ADIFOR* [8–10]. The ODE subroutine is therefore also the starting point for the automatic differentiation. To provide this new functionality several internal changes had to made in *OCPRSQP*. Additionally some "scripts" have been developed to automatize the process of taking all the needed subroutines from *DIFRUN* and calling *ADIFOR*, for example for the case that something has changed in ODE and a new G_ODE has to be computed. This makes it now a very easy to use extension to the original package. The necessity of using an automatic differentiation tool like *ADIFOR* instead of computing the derivatives with paper and pencil can also be seen from figure 5.2. It shows that DFRFCN on its own has a very complicated structure because of all the subroutines which are called.

The major goal is to keep *OCPRSQP* as some kind of "black box" hiding all the difficulties from a potential user who is not too interested in algorithmic details con-

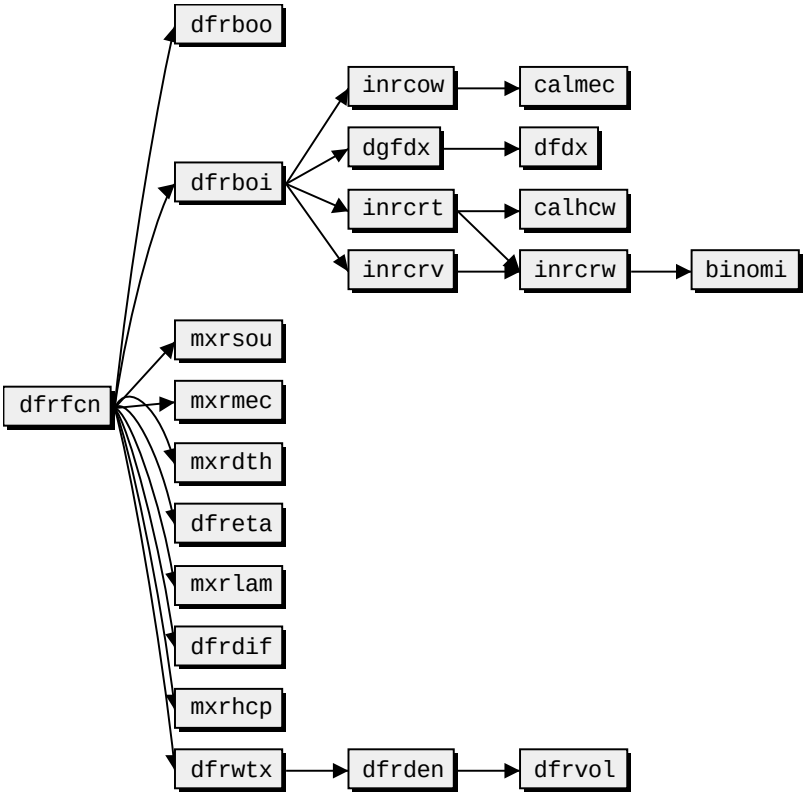


Figure 5.2.: Subroutines of DFRFCN.

cerning optimization algorithms. All the features should be controllable through parameters such that the core of the code does not have to be changed when going to a new application. The aim is that one should have to change only a few well defined portions of the code. This cannot be avoided because of restrictions of the programming language FORTRAN77—the dimension of arrays for example—or because of changed problem settings—boundary formulations or a different objective function for example. So far we have succeeded in keeping this necessary changes to a minimum and therefore are able to switch to a new application problem in minimal time.

To give an idea of the complexity of *OCPRSQP* figure 5.3 shows the most important subroutines.

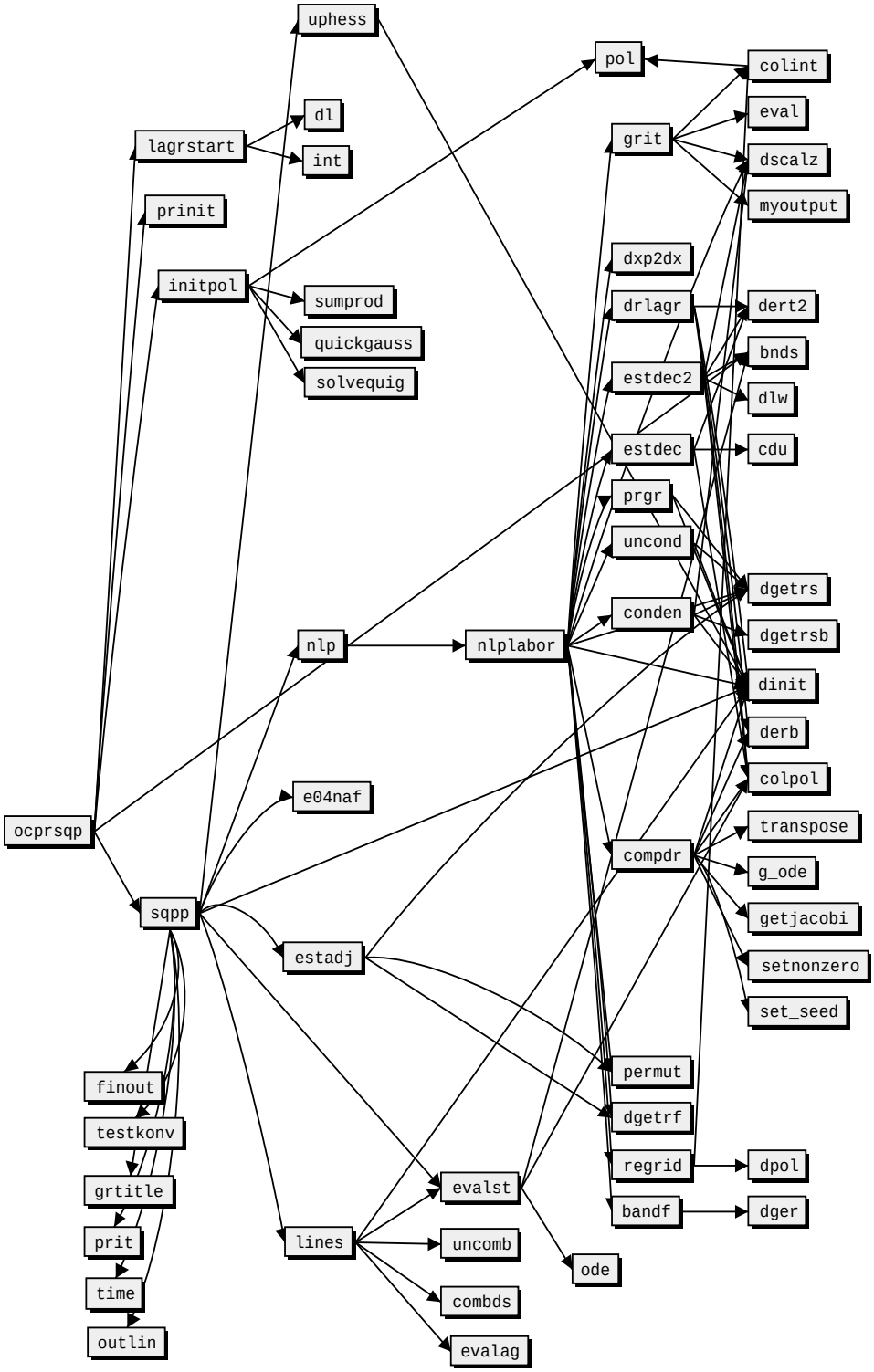


Figure 5.3.: Most important subroutines of OCPRSQP.

## 5.2. Block Band Structure of Derivatives

Providing accurate derivatives is crucial for the optimization algorithm. Computational tests showed that for the problem class under consideration computing the derivatives with finite differences is not accurate enough. Therefore an automatic differentiation tool, *ADIFOR* [8–10], has been used. It is used as a substitute for the finite differences: *ADIFOR* is applied to the subroutines describing the functions which have to be differentiated and then generates a new set of subroutines providing the analytic form of the derivatives for computing the Jacobian.

As can be seen in figure 4.4 and from equations 4.38 and 4.39 the Jacobian, that is $W_j$ has to be computed at each collocation point. Besides the number of species there are two important variables controlling the problem size: the number of collocation mesh points $m_{\text{col}}$ and the number of spatial grid points $m_{\text{spa}}$. Taking a closer look at the equations 4.38, 4.39 and the figure 4.4 reveals that the number of operations to compute all entries of $C_{\underline{y}}$ depends linearly on $m_{\text{col}}$ whereas it depends quadratically on $m_{\text{spa}}$ because $m_{\text{spa}}$ controls the size of the vectors $f$ and $g$ and the variables $y$ and $\xi$.

The subroutine LINEAR of *OCPRSQP* computes all the necessary entries of $C_{\underline{y}}$, $C_{\underline{u}}$, $D_{\underline{y}}$, and $D_{\underline{u}}$. Most of the time is spend on computing the $W_j$, $j = 1, \ldots, m_{\text{col}}$ which is done using the subroutines G_ODE computing the derivatives obtained with *ADIFOR*.

The absolute computing time does not play a role in the following considerations and therefore the time measurements have been normalized to show the observed relations more clearly.

Figure 5.4 shows the time spend in the subroutine LINEAR depending on the number of collocation mesh points $m_{\text{col}}$. It shows a very good agreement with the linear dependence expected from the considerations from above.

The next figure 5.5 shows that the time spend in the subroutine LINEAR depends quadratically on the number of grid points. A closer look at the $W_j$ matrices reveals that they are not dense but have a special block band structure which is shown in figure 5.6. This is due to the finite difference discretization of the spatial derivatives of the model equations and can be obtained using the dependency relations of section 4.1. It is tempting to use this structure for computing the $W_j$ to avoid computing "zero" entries. Fortunately *ADIFOR* offers the possibility to compute "structural independent columns" at the same time by setting a seed matrix [10]. These structural independent columns have been marked in figure 5.7. It is easy to see that all columns marked by diagonal lines from top left to down right are structural independent. The same applies to the columns marked by diagonals from down left to up right and cross hatched columns. The compressed $W_j$ is then shown in figure 5.8. The number of columns of this matrix is $N_s + 3(N_g + 4)$, that is independent of the discretization and therefore does not depend on $m_{\text{spa}}$ anymore. As a result it has the dimension

$$[N_s + m_{spa}(N_g + 4)] \times [N_s + 3(N_g + 4)]$$

and depends now only linear on $m_{\text{spa}}$. Computing these compressed $W_j$ is the most important and time consuming part of the sparse subroutine LINEAR.

Figure 5.9 shows the time spent in the sparse LINEAR depending on the number

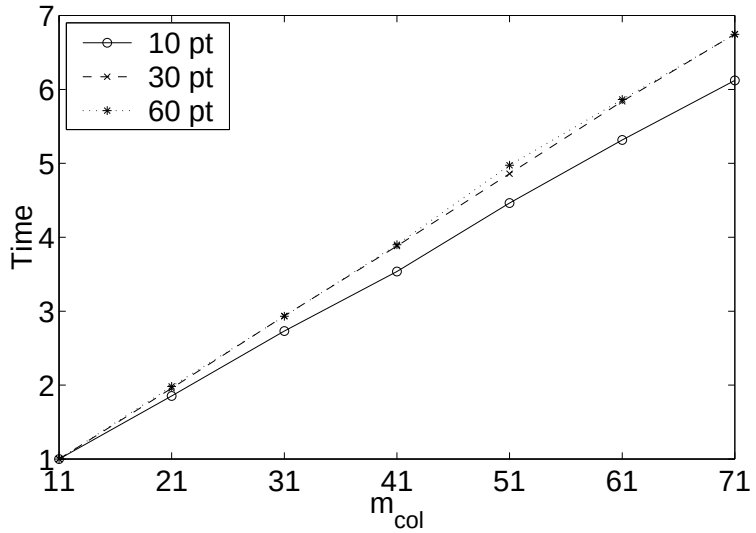


Figure 5.4.: The time spend in LINEAR which computes essential parts of $C_{\underline{y}}$, $C_{\underline{u}}$, $D_{\underline{y}}$, and $D_{\underline{u}}$ depends linearly on the number of collocation mesh points $m_{\mathrm{col}}$. The results are shown for different numbers of spatial grid points $m_{\mathrm{spa}}$.

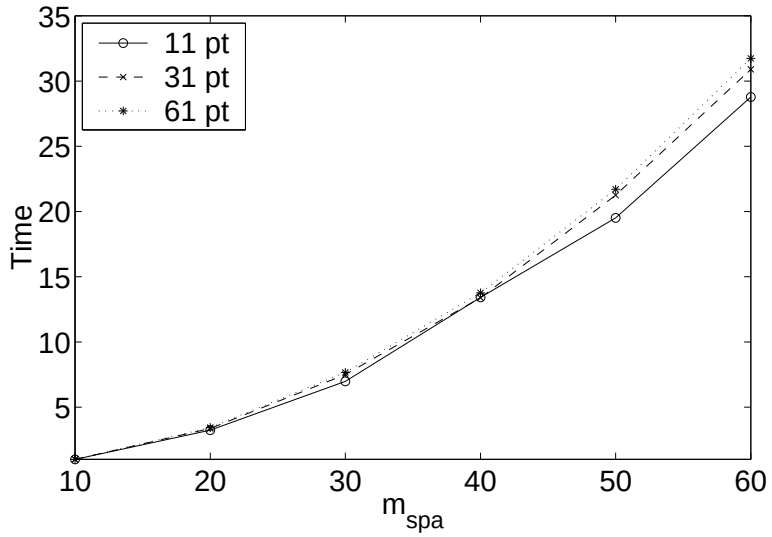


Figure 5.5.: The time spend in LINEAR which computes essential parts of $C_{\underline{y}}$, $C_{\underline{u}}$, $D_{\underline{y}}$, and $D_{\underline{u}}$ depends quadratically on the number of spatial grid points $m_{\mathrm{spa}}$. The results are shown for different numbers of collocation mesh points $m_{\mathrm{col}}$.

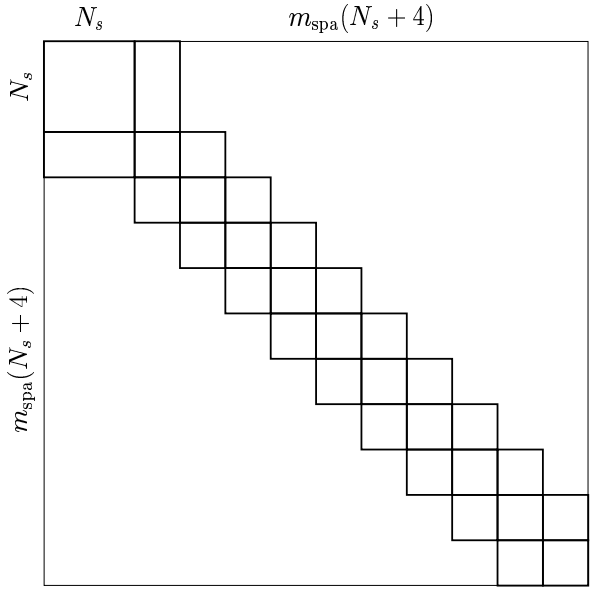


Figure 5.6.: Block band structure of $W_j$

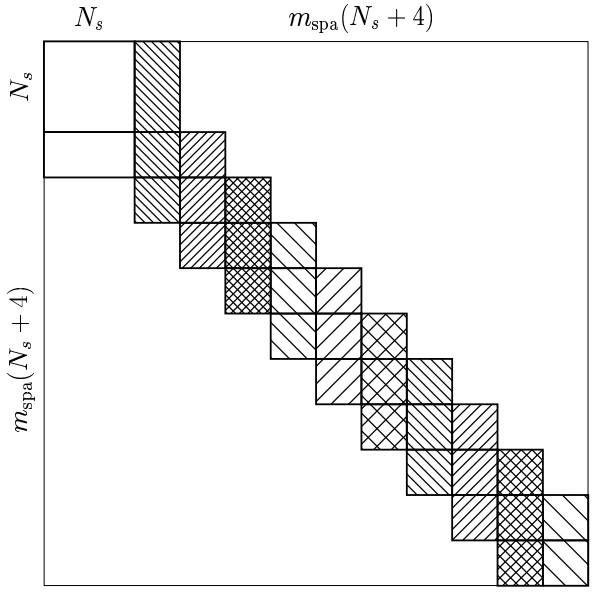


Figure 5.7.: Structural independent columns of $W_j$

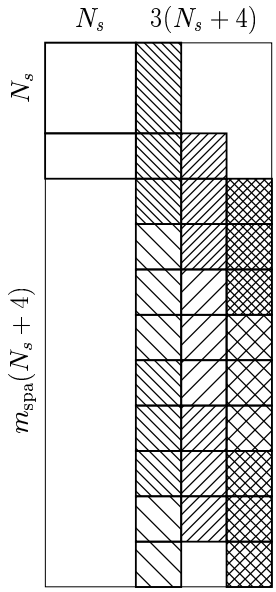


Figure 5.8.: Compressed $W_j$

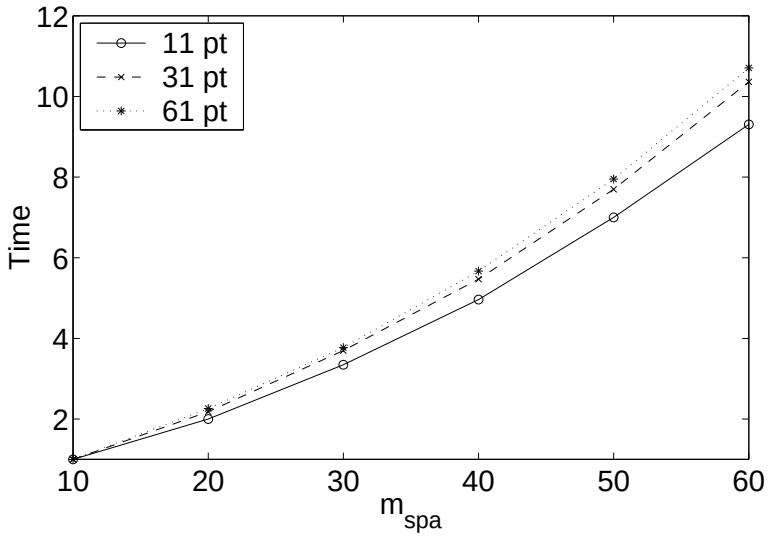


Figure 5.9.: The time spend in LINEAR depending on the number of spatial grid points $m_{\mathrm{spa}}$. The results are shown for different numbers of collocation mesh points $m_{\mathrm{col}}$.

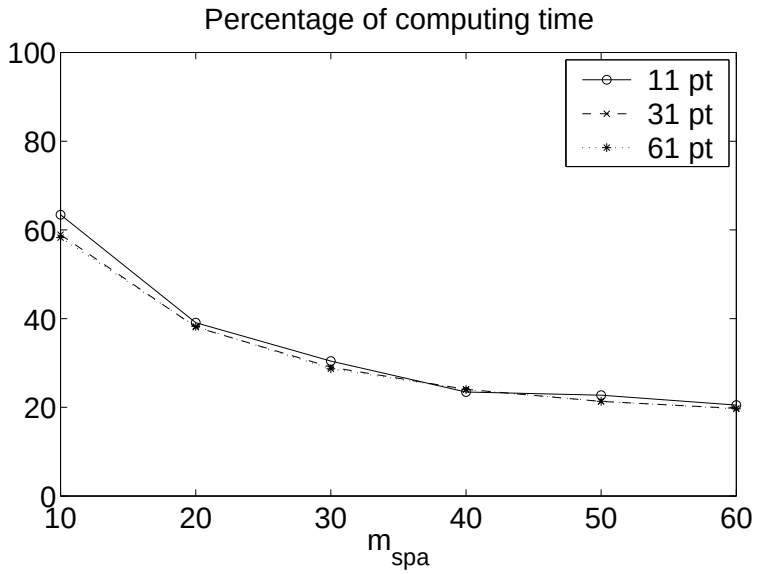


Figure 5.10.: Percentage of computing time needed for sparse LINEAR compared to 100% for the non-sparse LINEAR depending on the number of spatial grid points $m_{\text{spa}}$ for different numbers of collocation mesh points $m_{\text{col}}$.

of spatial grid points $m_{\text{spa}}$ for different numbers of collocation mesh points $m_{\text{col}}$. There is almost a linear dependence on $m_{\text{spa}}$ but the number of other computations in LINEAR grows with higher then linear order ($D_{\underline{y}}$ for example). Still the result is much better then using the non-sparse LINEAR as can be seen comparing the results with the one shown in figure 5.5

The sparse LINEAR needs 70–80% less computing time than the non-sparse LINEAR for medium to large size problems. This can be seen in figure 5.10 in which the percentage of time which the sparse LINEAR needs is compared to the 100% of the non-sparse LINEAR. Looking at the total time it is possible to save around 20–30% computational time using the sparse LINEAR. Figure 5.11 shows the percentage of total computing time needed with the sparse LINEAR compared to the 100% with the non-sparse LINEAR.

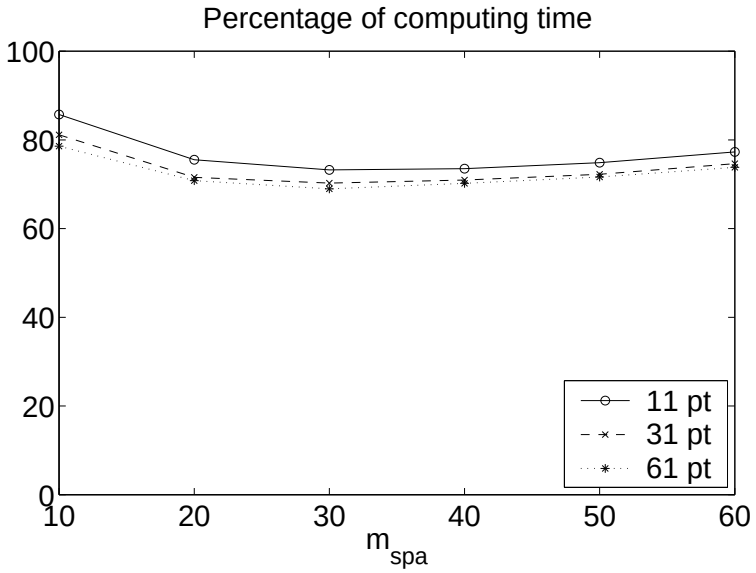


Figure 5.11.: Percentage of total computing time needed with sparse LINEAR compared to 100% with the non-sparse LINEAR depending on the number of spatial grid points $m_{\mathrm{spa}}$ for different numbers of collocation mesh points $m_{\mathrm{col}}$.

## 5.3. Sparse Linear Algebra Solver

The last section showed that one bottleneck in the partially reduced *SQP*-algorithm is the computation of the derivatives and one possibility has been shown to considerably reduce the computing time. A closer inspection of the partially reduced *SQP*-algorithm reveals that another bottleneck is the computation of the inverse of the matrix $C_{\bar{y}}$ (or $C_{\bar{y}}^T$). The recursion formulas of section 4.4 show that this reduces to compute the inverse of $W_j$ (or $W_j^T$). In the non-sparse implementation of *OCPRSQP* inverting is done by first computing an LU decomposition and the solving linear systems using standard linear algebra subroutines (DGETRF for the LU factorization and DGETRS for the solution of the linear systems [1]). Figure 5.12 shows time measurements for computing the LU decomposition of all $W_j$, $j = 1, \ldots, m_{\text{col}}$. This clearly depends linear on $m_{\text{col}}$ because the LU factorization has to be computed for all $W_j$, $1 \leq j \leq m_{\text{col}}$ as can easily be seen from the recursion formulas. The condensing step, that is computing all the data for the *QP* problem does also linearly depend on $m_{\text{col}}$ which can be expected from the recursion formulas and which is confirmed through time measurements as can be seen in figure 5.13. Looking at the dependence of the linear algebra on the number of spatial grid points $m_{\text{spa}}$ shows a dramatically worse behavior. It is well known that the number of operations of the algorithms for a LU decomposition is of the order $O(n^3)$, with $n$ being the dimension of the problem. Since the size of the matrices $W_j$ depend on $m_{spa}$—the size of $W_j$ is $[N_s + (N_g + 4)m_{\text{spa}}]n_k$—the time for factorization is expected to grow with $O(m_{\text{spa}}^3)$. This can be seen in figure 5.14 in which the DGETRF LU factorization subroutine has been used. Again it is well known that solving a linear system after obtaining a LU decomposition of the coefficient matrix is of order $O(n^2)$. Since the condensing subroutine contains the repeated solution of $n$ linear systems an $O(n^3)$ behavior of the algorithm is expected (as can be seen from the recursion formulas in section 4.4). This is confirmed by computational tests as shown in figure 5.15. In the following a modified LU factorization and corresponding algorithm for the solution of the linear systems is presented. It takes into account the particular block band structure of $W_j$.

### LU Factorization

As mentioned above it takes order $O(n^3)$ operations to compute a LU factorization if the structure of the matrix is not take into account. The structure of $W_j$ for which the LU factorization has to be computed is shown in figure 5.6. Using a standard LU decomposition unnecessarily operates on too many zeros and produces unwanted fill in through pivoting. One can get around these disadvantages by restricting the pivoting to certain areas and only performing computations on nonzero entries. The algorithm is illustrated in the following pictures. One starts in the upper left corner of figure 5.16(left) and computes a LU decomposition of the cross hatched part using column pivoting, that is looking only up to the $N_s$-th row for a pivot element. This yields the situation shown in figure 5.16(right) in which the horizontal lines mark the upper part (U) and the vertical lines the lower part (L) of the LU decomposition. Now the entries in the cross hatched part of figure 5.16(right) can be eliminated which leads to the situation shown in figure 5.17(left) in which the cross hatched part might have been modified through the elimination process. Now a LU decomposition

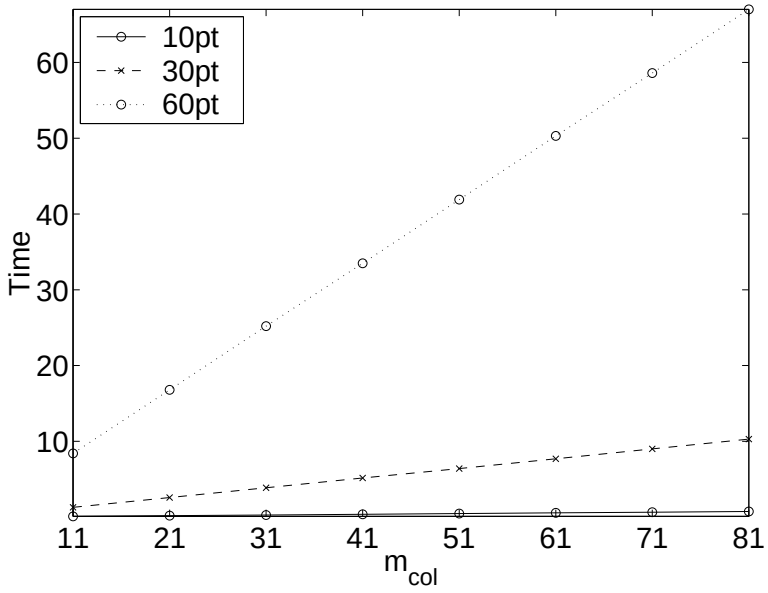


Figure 5.12.: The Time for LU factorization of $W_j$, $j = 1, \ldots, m_{\text{col}}$ depending linearly on the number of collocation mesh points $m_{\text{col}}$ for different numbers of spatial grid points $m_{\text{spa}}$.

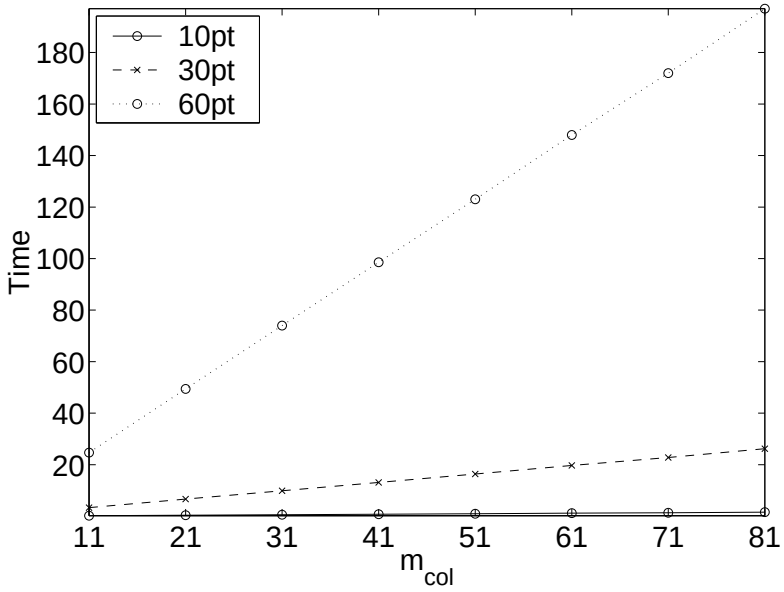


Figure 5.13.: The time for the condensing depends linearly on the number of collocation mesh points $m_{\text{col}}$ for different numbers of spatial grid points $m_{\text{spa}}$.

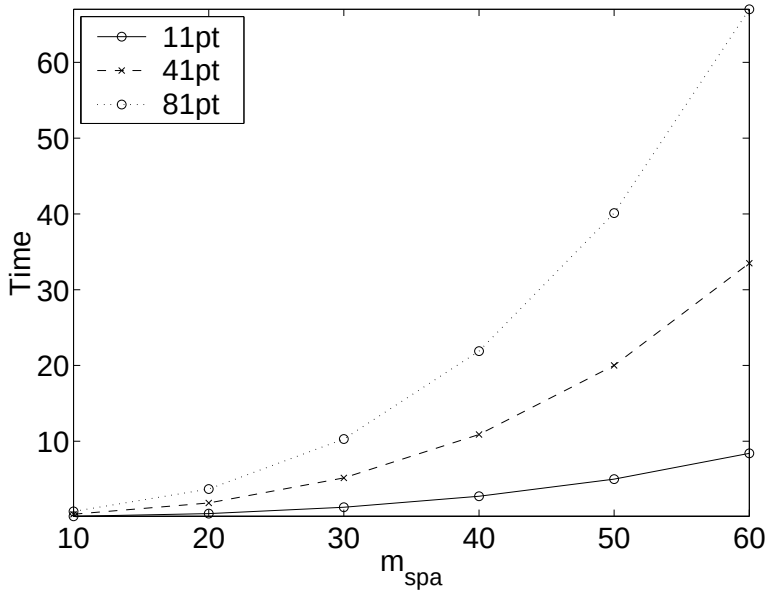


Figure 5.14.: Time for LU factorization of $W_j$ depending on the number of spatial grid points $m_{\text{spa}}$ for different numbers of collocation grid points $m_{\text{col}}$.

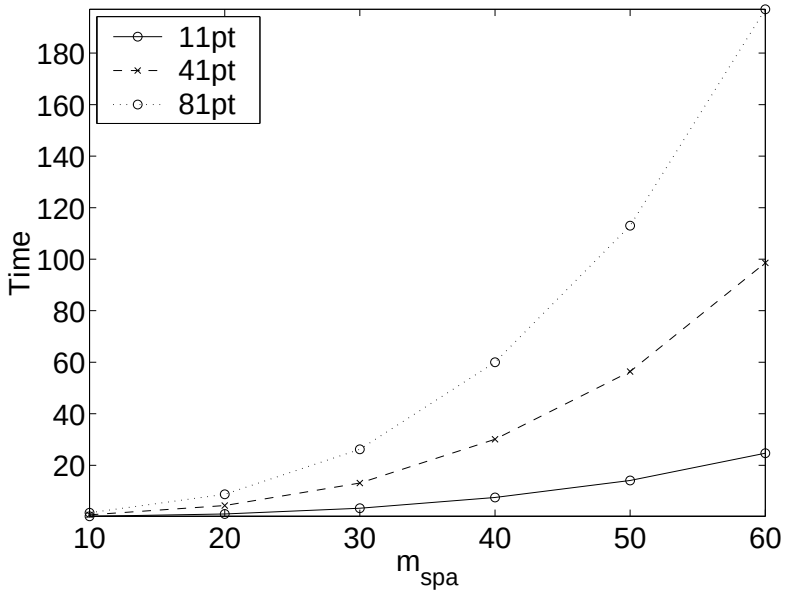


Figure 5.15.: The computing time for the condensing depending on the number of spatial grid points $m_{\text{spa}}$ for different numbers of collocation mesh points $m_{\text{col}}$.

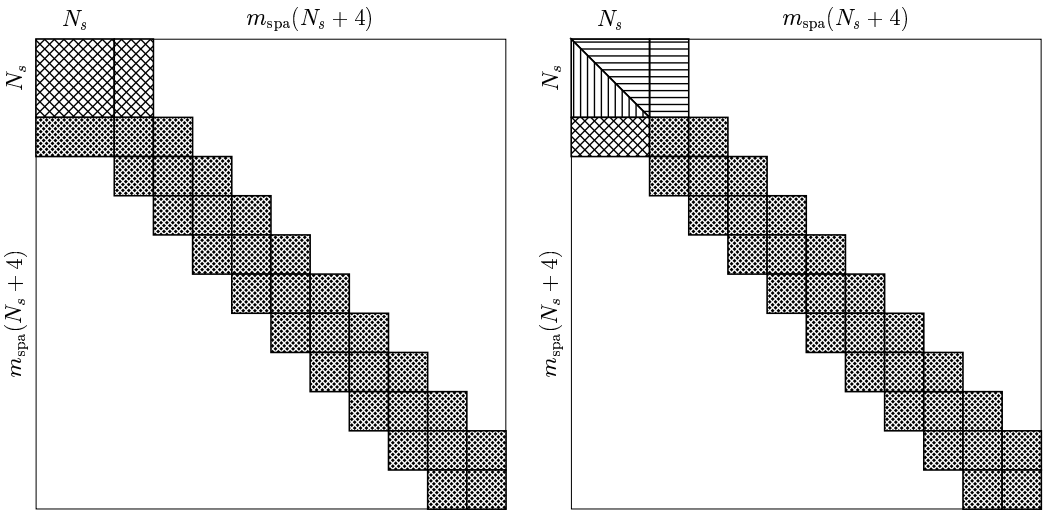


Figure 5.16.: First step (left) and second step (right) in modified LU algorithm.

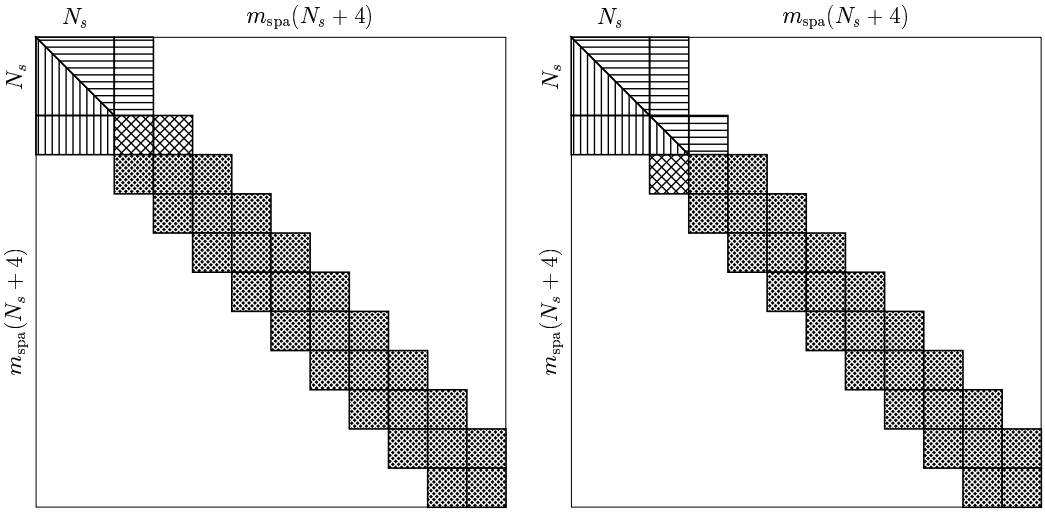


Figure 5.17.: Third step (left) and forth step (right) in modified LU algorithm.

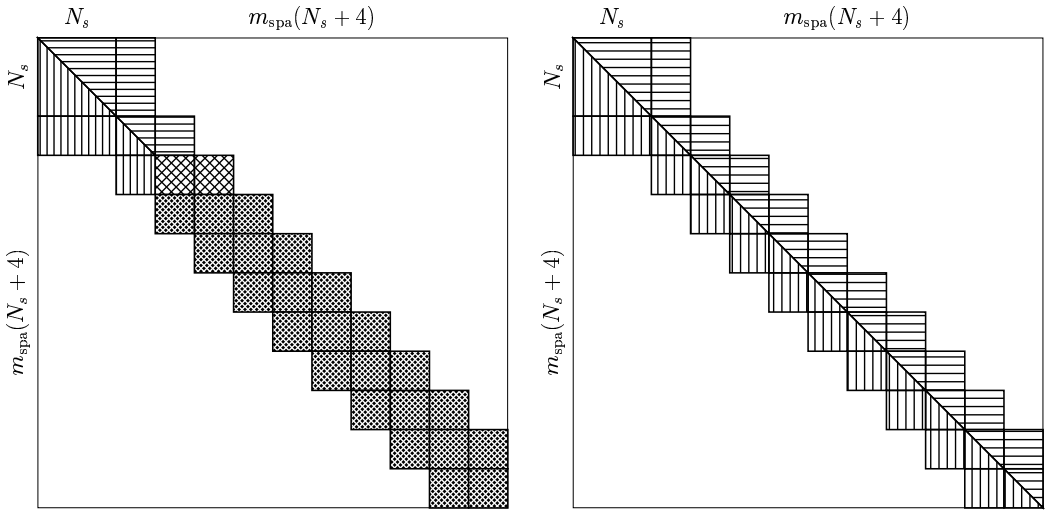


Figure 5.18.: Third step (left) and after the last step (right) in modified LU algorithm.

is performed on the cross hatched part of figure 5.17(left) which has as the result shown in figure 5.17(right). Again the cross hatched area of figure 5.17(right) can be eliminated can be eliminated with the result shown in figure 5.18(left). Here the cross hatched area has been modified during the elimination. Continuing this procedure leads to the particular form of the LU factorized matrix shown in figure 5.18(right). It has the advantage that there is no additional fill in and that all necessary operations are performed purely on the non-zero entries of this particular problem structure. It is not difficult to see that this algorithm depends only linearly on $m_{\text{spa}}$ since by construction essentially $m_{\text{spa}}$ times a fixed size LU factorization has to be performed. Computational test confirm this behavior as can be seen in figure 5.19. Not only the linear dependence, but also a much lower computing time for the factorization has been obtained which can also be seen in figure 5.19 compared to figure 5.14. The result is even more obvious if the sparse factorization is compared directly with the non-sparse factorization which is done in figure 5.20. A speed up of a factor of more than 200 for large problems is obtained.

### Solving the Linear System

In the last paragraph it has been shown that it is possible to considerably speed up the factorization of $W_j$ by taking into account the problem structure, that is the structure of the spatial discretization of the model equations. The condensing algorithm mainly solves $n$ linear systems in order $O(n^2)$ each. This leads to a $O(n^3)$ behavior as shown in figure 5.12. An algorithm corresponding to the factorization algorithm described in the last section has been developed to solve these linear systems. For similar reasons as above this algorithm is only linearly dependent on $m_{\text{spa}}$. But since $O(m_{\text{spa}})$ systems have to be solved an $O(m_{\text{spa}}^2)$ behavior is expected for the sparse condensing. This is fully confirmed by computational experiments as can be seen in figure 5.21. Figure 5.22 shows the factors which the sparse solver is faster than the non-sparse solver. This can be up to factor of 4.5 for large problems.

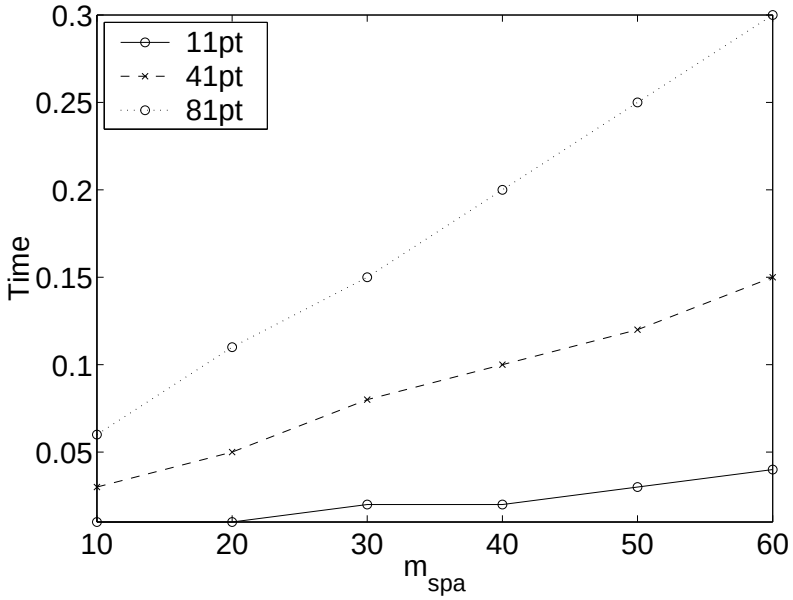


Figure 5.19.: The time for sparse LU factorization of $W_j$, $j = 1, \ldots, m_{\text{col}}$ depending on the number of spatial grid points $m_{\text{spa}}$ for different numbers of collocation mesh points $m_{\text{col}}$.

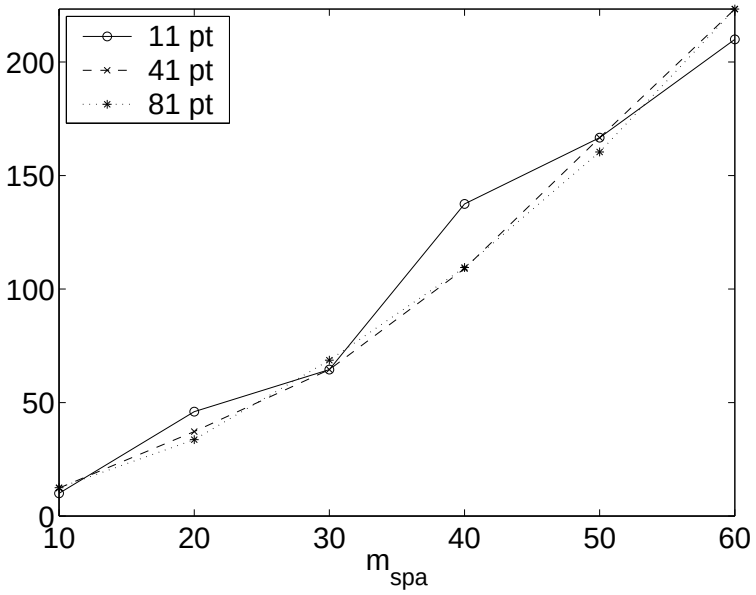


Figure 5.20.: Sparse LU factorization is several times faster than none-sparse LU factorization of $W_j$, $j = 1, \ldots, m_{\text{col}}$ depending on the number of spatial grid points $m_{spa}$ for different numbers of collocation mesh points $m_{\text{col}}$.

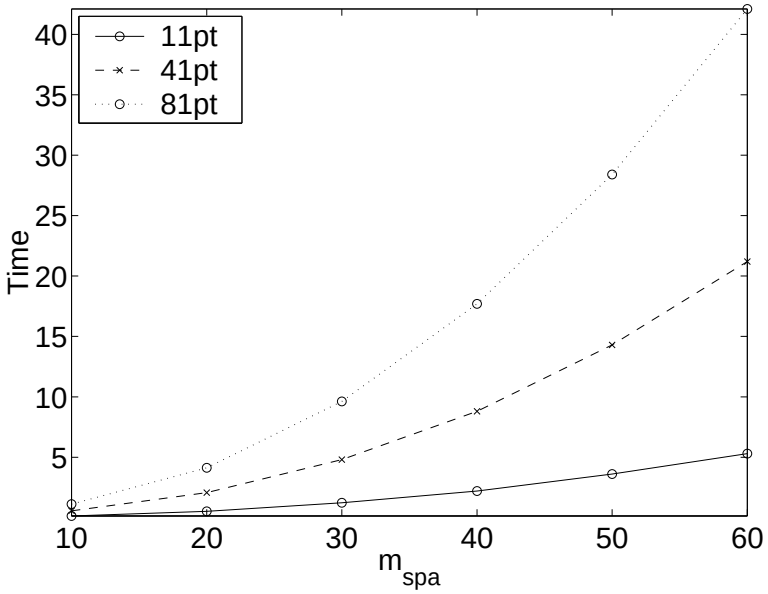


Figure 5.21.: Time for sparse condensing depending on the number of spatial grid points $m_{\text{spa}}$ for different numbers of collocation mesh points $m_{\text{col}}$.

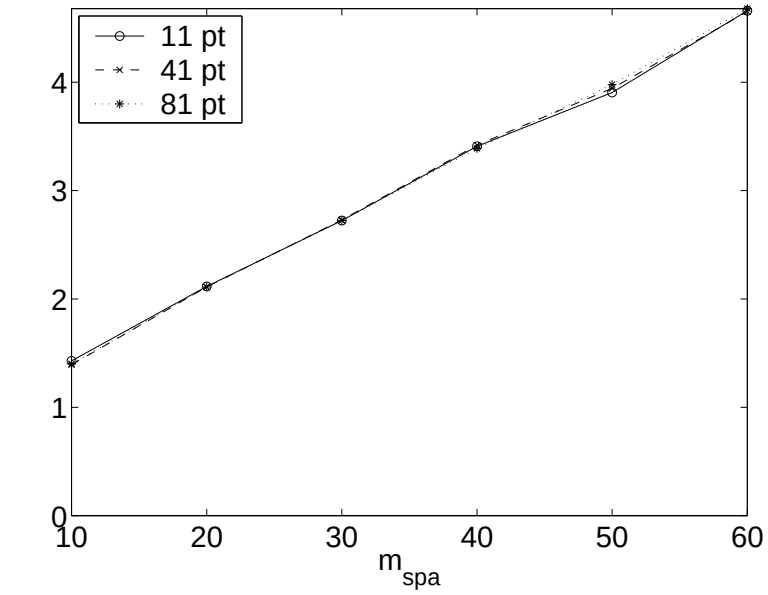


Figure 5.22.: Sparse condensing is several times faster than non-sparse condensing depending on the number of spatial grid points $m_{\text{spa}}$ for different numbers of collocation mesh points $m_{\text{col}}$.

**Results**

It has been shown that taking the problem structure into account when developing solvers for linear systems can lead to quite dramatic reductions of computing time. But this has to be considered in relation to the total computing time and there the improvements are still remarkable. Using the sparse factorization and sparse solver in the condensing gives almost a factor of 2 as can be seen in figure 5.23. If in addition the sparse LINEAR is used, then improvements of more than a factor 5 are obtained for large problems. The results are presented in figure 5.24. The stability of this sparse algorithm for the LU factorization and the linear system solver has been observed in all computations of this problem class.

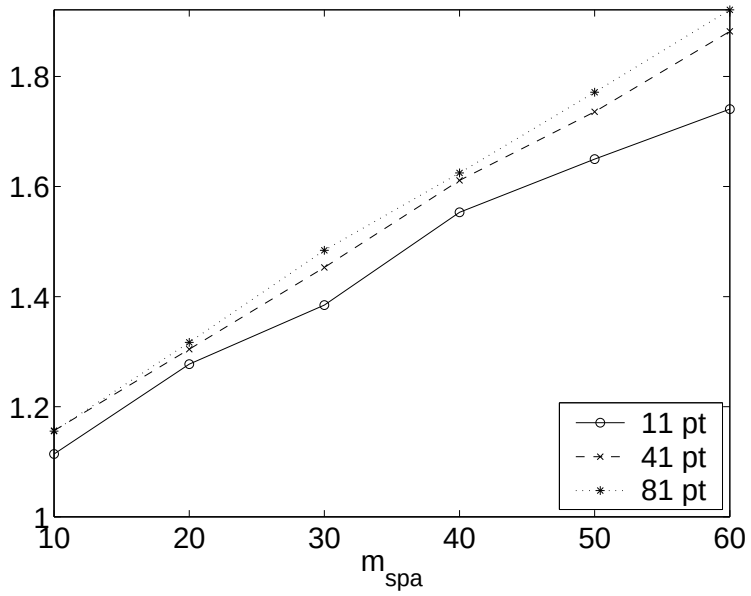


Figure 5.23.: The sparse LU factorization and sparse solver for the condensing is several times faster than the non-sparse algorithm depending on the number of spatial grid points $m_{spa}$ for different numbers of collocation mesh points $m_{\mathrm{col}}$.

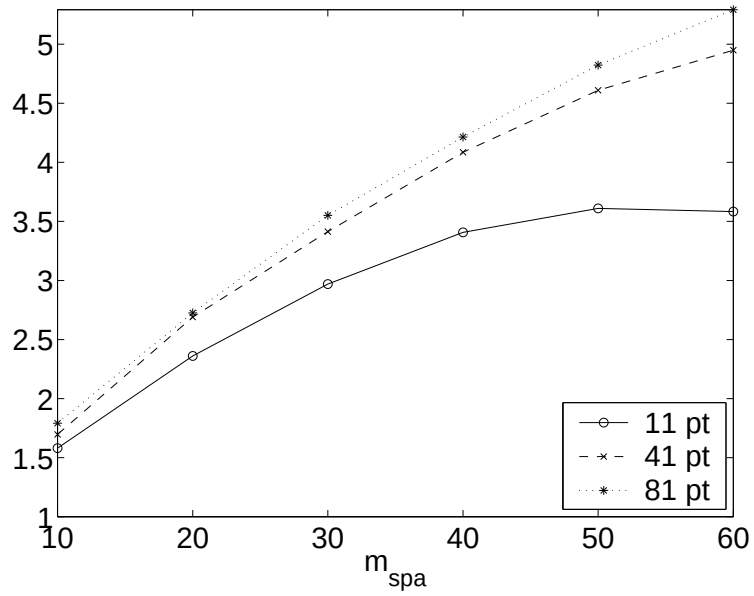


Figure 5.24.: Sparse LU factorization, sparse solver for the condensing and sparse LINEAR is several times faster than the non-sparse algorithm depending on the number of spatial grid points $m_{spa}$ for different numbers of collocation mesh points $m_{\mathrm{col}}$.

## 5.4. Reduced Problem Size of QP

The partially reduced *SQP*-algorithm described in section 4.4 uses the discretized controls and the initial values as possible 'control' variables, the variables denoted by $\underline{u}$. In this thesis a problem class is considered in which the initial values are always fixed. So it seems desirable to reformulate the algorithm in such a way that this fact is recognized, that is to have only the original discretized controls as control variables. This would reduce the number of variables for the QP-problem considerably because usually there are several hundred equations and therefore several hundred initial values and far less than a hundred control variables. To fix the initial values $y$ at $t = 0$ another $N_{\mathrm{ND}}$ equations—$c_0^{\mathrm{con}}$—are added to the continuity conditions 4.34

$$c_0^{\mathrm{con}} = y^0 - y_1 = 0$$

with the initial values $y^0 := y(t = 0)$ fixed. Furthermore the grouping of the variables and the collocation and continuity conditions is changed in the vectors $\underline{y}$, $\underline{u}$, $c$ and is now

$$\begin{aligned}
\underline{y} &= \begin{bmatrix} y_1 & \xi_1 & y_2 & \xi_2 & \dots & \xi_{m-1} & y_m \end{bmatrix} \\
\underline{u} &= \begin{bmatrix} u_1 & u_2 & \dots & u_{m-1} \end{bmatrix} \\
c &= \begin{bmatrix} c_0^{\mathrm{con}} & c_1^{\mathrm{col}} & c_1^{\mathrm{con}} & c_2^{\mathrm{col}} & c_2^{\mathrm{con}} & \dots & c_{m-1}^{\mathrm{col}} & c_{m-1}^{\mathrm{con}} \end{bmatrix}
\end{aligned}$$

Through this new ordering the matrices $C_{\underline{y}}$, $C_{\underline{u}}$, $D_{\underline{y}}$ and $D_{\underline{u}}$ are changed and the figures 5.25–5.28 show the modified matrices in which the derivatives are computed like in the original algorithm. In the $C_{\underline{y}}$ matrix $N_d$ rows and columns have to be added which are additionally marked by diagonal lines in figure 5.25. In the modified $C_{\underline{u}}$ the first $N_d$ columns have been removed from the original $C_{\underline{u}}$ and additionally $N_d$ rows, which are marked in figure 5.26 have been added. $N_d$ columns have been added to the original $D_{\underline{y}}$ and are marked by diagonal lines in figure 5.27. Finally the first $N_d$ columns have been removed from the original $D_{\underline{u}}$ as can be seen in figure 5.28 compared to figure 4.7. In a next step the recursion formulas are adapted to the block band structure of these matrices. As above a partition of two variables $x$ and $c$ is defined as

$$\begin{aligned}
x &:= \begin{bmatrix} y_1 & \xi_1 & y_2 & \xi_2 & \dots & \xi_{m-1} & y_m \end{bmatrix} \\
c &:= \begin{bmatrix} c_0^{\mathrm{con}} & c_1^{\mathrm{col}} & c_1^{\mathrm{con}} & \dots & c_{m-1}^{\mathrm{col}} & c_{m-1}^{\mathrm{con}} \end{bmatrix}
\end{aligned}$$

With these and taking into account the block band structure of $C_{\underline{y}}$ (figure 5.25) one obtains the following recursion to solve $C_{\underline{y}} x = c$

$$\left.\begin{aligned}
y_1 &= -c_0^{\mathrm{con}} \\
\xi_j &= W_j^{-1} \left( c_j^{\mathrm{col}} - V_j y_j \right) \\
y_{j+1} &= y_j + \Psi_j \xi_j - c_j^{\mathrm{con}}
\end{aligned}\right\} \quad j = 1, \dots, m-1$$

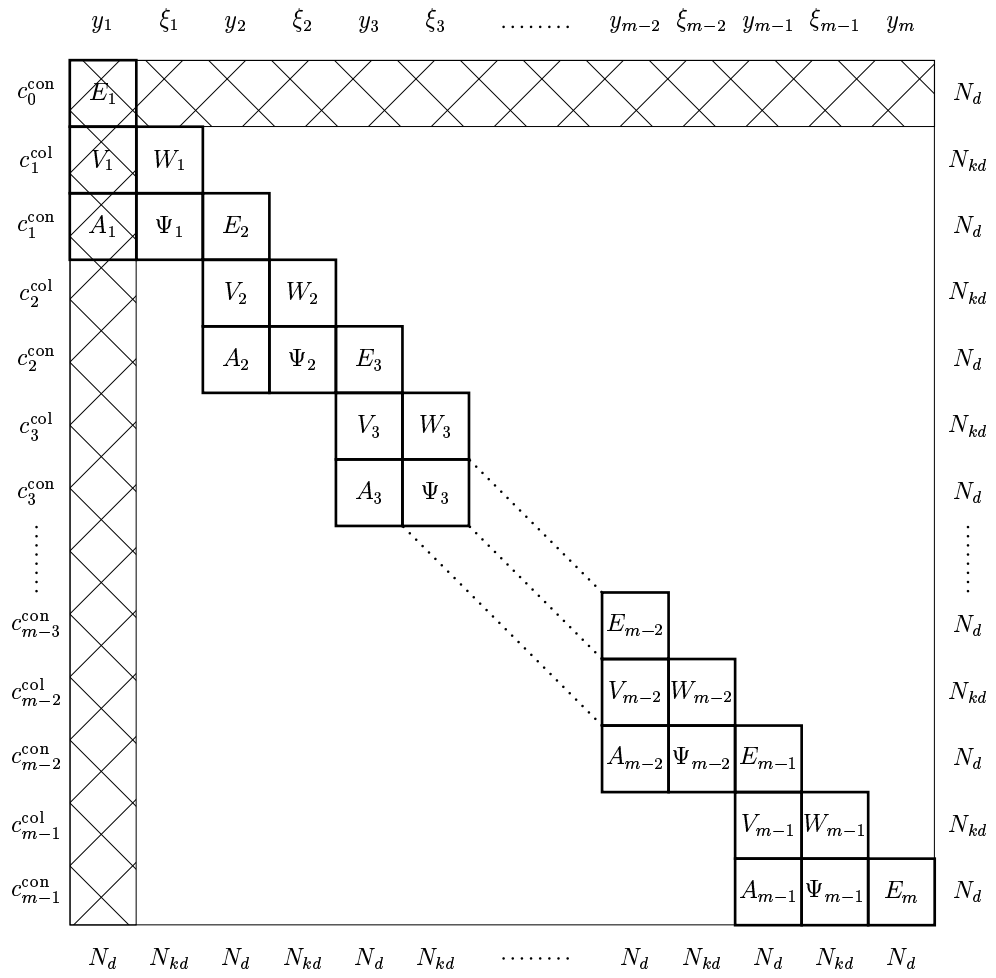


Figure 5.25.: Block structure of modified $C_{\underline{y}}$

Similarly one obtains the recursion to solve $C_{\underline{y}}^T c = x$

$$\left.\begin{aligned} c_{m-1}^{\text{con}} &= -y_m \\ c_j^{\text{col}} &= W_j^{-T}\left(\xi_j - \Psi_j^T c_j^{\text{con}}\right) \\ c_{j-1}^{\text{con}} &= c_j^{\text{con}} - y_j - V_j^T c_j^{\text{col}} \end{aligned}\right\} \quad j = m-1, \ldots, 1$$

As in the original algorithm the recursion to compute $D_{\underline{u}} - D_{\underline{y}} C_{\underline{y}}^{-1} C_{\underline{u}}$ is more complex. To illustrate the recursion a simple example with four collocation mesh points is

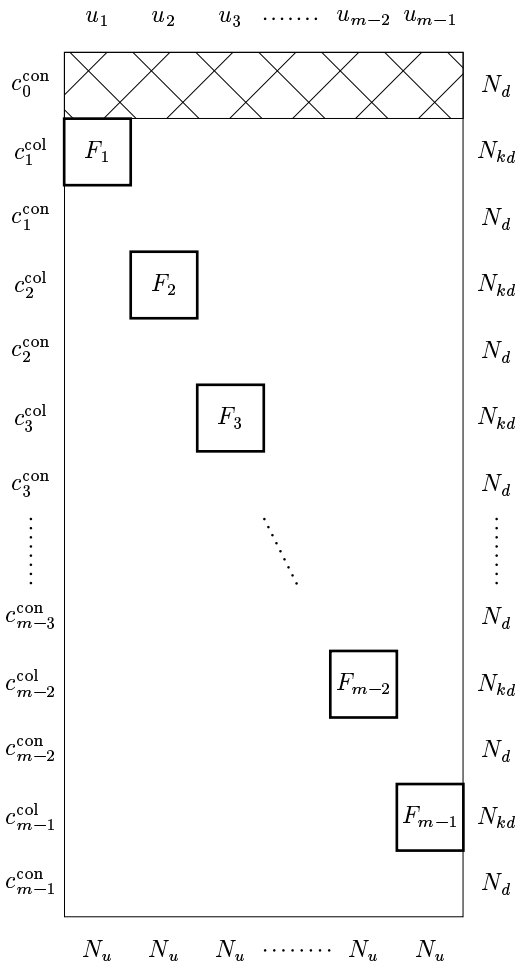


Figure 5.26.: Block structure of modified $C_{\underline{u}}$

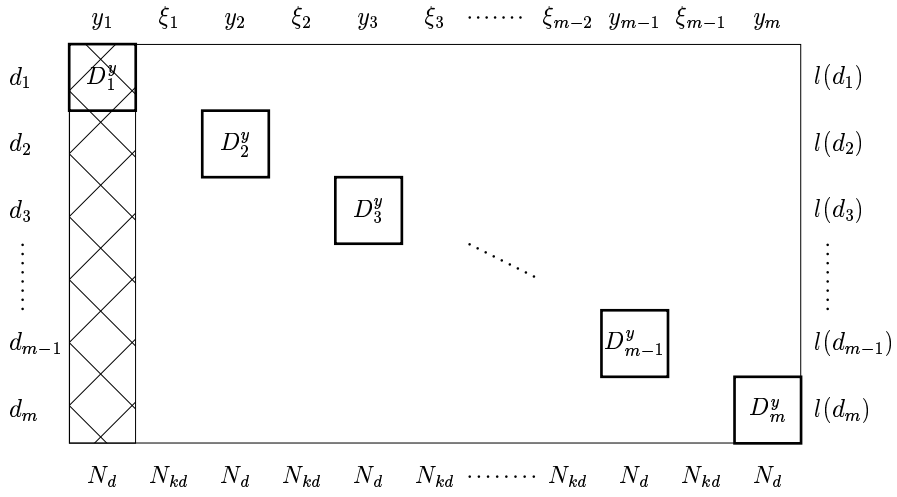


Figure 5.27.: Block structure of modified $D_{\underline{y}}$

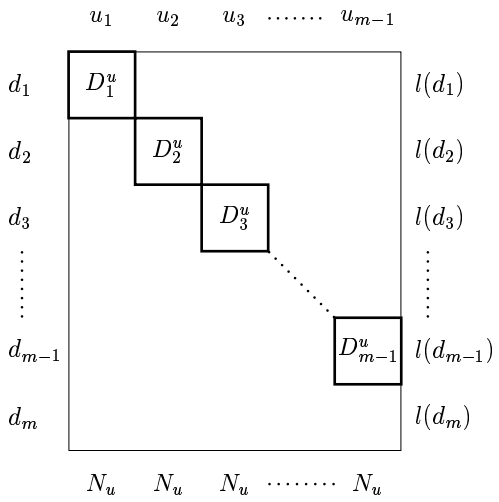


Figure 5.28.: Block structure of modified $D_{\underline{u}}$

considered. The first step is to solve $C_{\underline{y}}X = C_{\underline{u}}$

$$\begin{bmatrix} -I & & & & & & \\ V_1 & W_1 & & & & & \\ I & \Psi_1 & -I & & & & \\ & & V_2 & W_2 & & & \\ & & I & \Psi_2 & -I & & \\ & & & & V_3 & W_3 & \\ & & & & I & \Psi_3 & -I \end{bmatrix} \begin{bmatrix} X_{11} & X_{12} & X_{13} \\ Y_{11} & Y_{12} & Y_{13} \\ X_{21} & X_{22} & X_{23} \\ Y_{21} & Y_{22} & Y_{23} \\ X_{31} & X_{32} & X_{33} \\ Y_{31} & Y_{32} & Y_{33} \\ X_{41} & X_{42} & X_{43} \end{bmatrix} = \begin{bmatrix} F_1 & & \\ & & \\ & F_2 & \\ & & \\ & & \\ & & \\ & & F_3 \end{bmatrix}$$

which gives the solution

$$\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ Y_{11} & Y_{12} & Y_{13} \\ X_{21} & X_{22} & X_{23} \\ Y_{21} & Y_{22} & Y_{23} \\ X_{31} & X_{32} & X_{33} \\ Y_{31} & Y_{32} & Y_{33} \\ X_{41} & X_{42} & X_{43} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ W_1^{-1}F_1 & 0 & 0 \\ \Psi_1 Y_{11} & 0 & 0 \\ -W_2^{-1}V_2X_{21} & W_2^{-1}F_2 & 0 \\ X_{21} + \Psi_2 Y_{21} & \Psi_2 Y_{22} & 0 \\ -W_3^{-1}V_3X_{31} & -W_3^{-1}V_3X_{32} & W_3^{-1}F_3 \\ X_{31} + \Psi_3 Y_{31} & X_{31} + \Psi_3 Y_{32} & \Psi_3 Y_{33} \end{bmatrix}$$

Multiplying $X$ by $-D_{\underline{y}}$ and adding to $D_{\underline{u}}$ yields

$$\begin{bmatrix} D_1^u - D_1^y X_{11} & -D_1^y X_{12} & -D_1^y X_{13} \\ -D_2^y X_{21} & D_2^u - D_2^y X_{22} & -D_2^y X_{23} \\ -D_3^y X_{31} & -D_3^y X_{32} & D_3^u - D_3^y X_{33} \\ -D_4^y X_{41} & -D_4^y X_{42} & -D_4^y X_{43} \end{bmatrix}$$

Since only the $X_{ij}$ entries are needed one can define $X'$ by

$$X' := \begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \\ X_{41} & X_{42} & X_{43} \end{bmatrix}$$

and obtains

$$X' = \begin{bmatrix} 0 & 0 & 0 \\ \Psi_1 W_1^{-1} F_1 & 0 & 0 \\ [I - \Psi_2 W_2^{-1} V_2] X_{21} & \Psi_2 W_2^{-1} F_2 & 0 \\ [I - \Psi_3 W_3^{-1} V_3] X_{31} & [I - \Psi_3 W_3^{-1} V_3] X_{32} & \Psi_3 W_3^{-1} F_3 \end{bmatrix}$$

Therefore $D_{\underline{u}} - D_{\underline{y}} C_{\underline{y}}^{-1} C_{\underline{u}}$ can be written as

$$\begin{bmatrix} D_1^u & 0 & 0 \\ -D_2^y \Psi_1 W_1^{-1} F_1 & D_2^u & 0 \\ -D_3^y [I - \Psi_2 W_2^{-1} V_2] X_{21} & -D_3^y \Psi_2 W_2^{-1} F_2 & D_3^u \\ -D_4^y [I - \Psi_3 W_3^{-1} V_3] X_{31} & -D_4^y [I - \Psi_3 W_3^{-1} V_3] X_{32} & -D_4^y \Psi_3 W_3^{-1} F_3 \end{bmatrix} \tag{5.1}$$

This form of the solution leads to a natural recursion which is illustrate in the following. Starting with the matrices $\bar{D}_i^y$ and $\bar{D}_j^u$ which are arranged in the following order

| $\bar{D}_4^y$ | $\bar{D}_3^y$ | $\bar{D}_2^y$ | $\bar{D}_1^y$ | $\bar{D}_3^u$ | $\bar{D}_2^u$ | $\bar{D}_1^u$ |
|---|---|---|---|---|---|---|
| | | | $D_1^y$ | | | $D_1^u$ |
| | | $-D_2^y$ | | | $D_2^u$ | |
| | $-D_3^y$ | | | $D_3^u$ | | |
| $-D_4^y$ | | | | | | |

an update of the columns $\bar{D}_3^y$ and $\bar{D}_3^u$ is computed

$$\bar{D}_3^y = \bar{D}_3^y + \bar{D}_4^y [I - \Psi_3 W_3^{-1} V_3] \qquad \bar{D}_3^u = \bar{D}_3^u + \bar{D}_4^y [\Psi_3 W_3^{-1} F_3]$$

and one obtains

| $\bar{D}_3^y$ | $\bar{D}_2^y$ | $\bar{D}_1^y$ | $\bar{D}_3^u$ | $\bar{D}_2^u$ | $\bar{D}_1^u$ |
|---|---|---|---|---|---|
| | | $D_1^y$ | | | $D_1^u$ |
| | $-D_2^y$ | | | $D_2^u$ | |
| $-D_3^y$ | | | $D_3^u$ | | |
| $-D_4^y [I - \Psi_3 W_3^{-1} V_3]$ | | | $-D_4^y [\Psi_3 W_3^{-1} F_3]$ | | |

in which the column $\bar{D}_4^y$ is not shown anymore because it has not changed and it is not needed any further. A glance at the solution 5.1 shows that $\bar{D}_3^u$ contains now the last column of the solution 5.1. This will not change nor contribute any further and therefore it is not displayed in further steps. The next step computes updates to $\bar{D}_2^y$ and $\bar{D}_2^u$

$$\bar{D}_2^y = \bar{D}_2^y + \bar{D}_3^y [I - \Psi_2 W_2^{-1} V_2] \qquad \bar{D}_2^u = \bar{D}_2^u + \bar{D}_3^y [\Psi_2 W_2^{-1} F_2]$$

and one obtains

| $\bar{D}_2^y$ | $\bar{D}_1^y$ | $\bar{D}_2^u$ | $\bar{D}_1^u$ |
|---|---|---|---|
| | $D_1^y$ | | $D_1^u$ |
| $-D_2^y$ | | $D_2^u$ | |
| $-D_3^y [I - \Psi_2 W_2^{-1} V_2]$ | | $-D_3^y [\Psi_2 W_2^{-1} F_2]$ | |
| $-D_4^y [I - \Psi_3 W_3^{-1} V_3][I - \Psi_2 W_2^{-1} V_2]$ | | $-D_4^y [I - \Psi_3 W_3^{-1} V_3][\Psi_2 W_2^{-1} F_2]$ | |

The updated column $\bar{D}_2^y$ now contains the second column of the solution 5.1. In the final step of this example updates to $\bar{D}_1^y$ and $\bar{D}_1^u$ are computed.

$$\bar{D}_1^y = \bar{D}_1^y + \bar{D}_2^y[I - \Psi_1 W_1^{-1} V_1] \qquad \bar{D}_1^u = \bar{D}_1^u + \bar{D}_2^y[\Psi_1 W_1^{-1} F_1]$$

and one obtains

$$\begin{array}{c} \bar{D}_1^y \\ \hline D_1^y \\ -D_2^y[I - \Psi_1 W_1^{-1} V_1] \\ -D_3^y[I - \Psi_2 W_2^{-1} V_2][I - \Psi_1 W_1^{-1} V_1] \\ -D_4^y[I - \Psi_3 W_3^{-1} V_3][I - \Psi_2 W_2^{-1} V_2][I - \Psi_1 W_1^{-1} V_1] \end{array}$$

$$\begin{array}{c} \bar{D}_1^u \\ \hline D_1^u \\ -D_2^y[\Psi_1 W_1^{-1} F_1] \\ -D_3^y[I - \Psi_2 W_2^{-1} V_2][\Psi_1 W_1^{-1} F_1] \\ -D_4^y[I - \Psi_3 W_3^{-1} V_3][I - \Psi_2 W_2^{-1} V_2][\Psi_1 W_1^{-1} F_1] \end{array}$$

$\bar{D}_1^u$ contains the first column of the solution 5.1. In contrast to the original algorithm the computation of the update to $\bar{D}_1^y$ is not really needed. This example calculation leads to the recursion

$$[\ \bar{D}_j^y \mid \bar{D}_j^u\ ] := [\ \bar{D}_j^y \mid \bar{D}_j^u\ ] + \bar{D}_{j+1}^y[\ I - \Psi_j W_j^{-1} V_j \mid \Psi_j W_j^{-1} F_j\ ] \quad j = m-1, \ldots, 1$$

which is very similar to the original one. But now one is only interested in the solution matrix of the form

$$\begin{bmatrix} \bar{D}_1^u & \bar{D}_2^u & \ldots & \bar{D}_{m-1}^u \end{bmatrix}$$

Again this can explicitly be written in closed form as

$$\begin{bmatrix} D_1^u & 0 & 0 & \ldots & 0 \\ D_2^y Z_{11} & D_2^u & 0 & \ldots & 0 \\ D_3^y Z_{21} & D_3^y Z_{22} & D_3^u & \ddots & 0 \\ \vdots & & & & \vdots \\ D_{m-1}^y Z_{m-2,1} & D_{m-1}^y Z_{m-1,2} & D_{m-1}^y Z_{m-2,3} & \ldots & D_{m-1}^u \\ D_m^y Z_{m-1,1} & D_m^y Z_{m-1,2} & D_m^y Z_{m-1,3} & \ldots & D_m^y Z_{m-1,m-1} \end{bmatrix}$$

with

$$Z_{i,j} = \left[ \prod_{k=j+1}^{i} (I - \Psi_{i-k+2} W_{i-k+2}^{-1} V_{i-k+2}) \right] \Psi_j W_j^{-1} F_j \qquad 1 \le i \le j \le m-1$$

**Comparison with Original Algorithm**

Implementing the necessary regrouping of the variables and changing the recursion subroutines is unfortunately not a trivial task and would be a major reprogramming of the software package *OCPRSQP*. But fortunately one important aspect of this

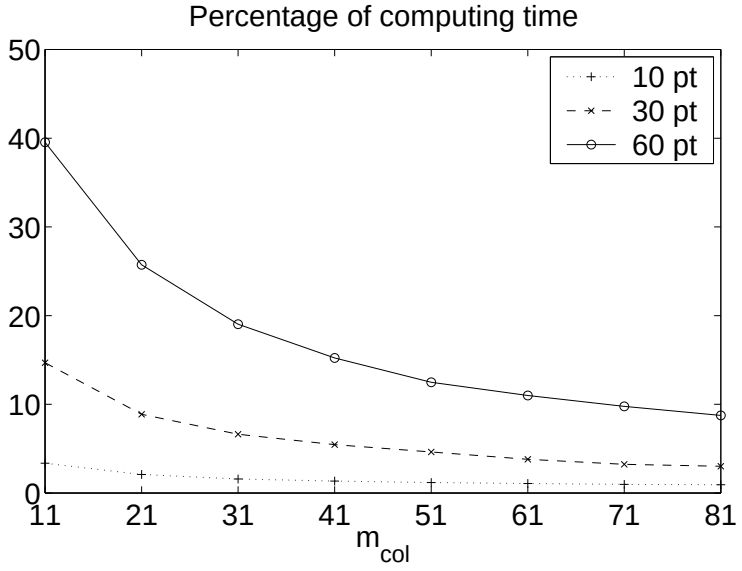


Figure 5.29.: Percentage of total computing time of optimization algorithm spend for solving the QP-problem depending on the number of collocation mesh points $m_{\mathrm{col}}$ for different numbers of spatial grid points $m_{\mathrm{spa}}$.

reduction can be tested. The speed up through solving only the much smaller QP-problem. The QP-solver E04NAF which is used in *OCPRSQP* allows to fix some of the variables and takes advantage of it. So it is possible to obtain a very good estimation of the speed up due to the reduction of the size of the QP problem. Figure 5.29 shows the percentage of the computing time of the otherwise optimized code which is spend for solving the QP-problem. For medium sized problems this is around 5–15%. Fixing the initial values in the QP-solver, reduces this to 2.5–5% which can be seen in figure 5.30. Figure 5.31 shows the computing time spend on solving the QP-problem during an optimization. Figure 5.32 shows the computing time spent on solving the QP-problem for fixed initial data for the same optimization. So fixing the initial data leads to a speed up for solving the QP-problem of at least 2 to up to 5 as can be seen in figure 5.33.

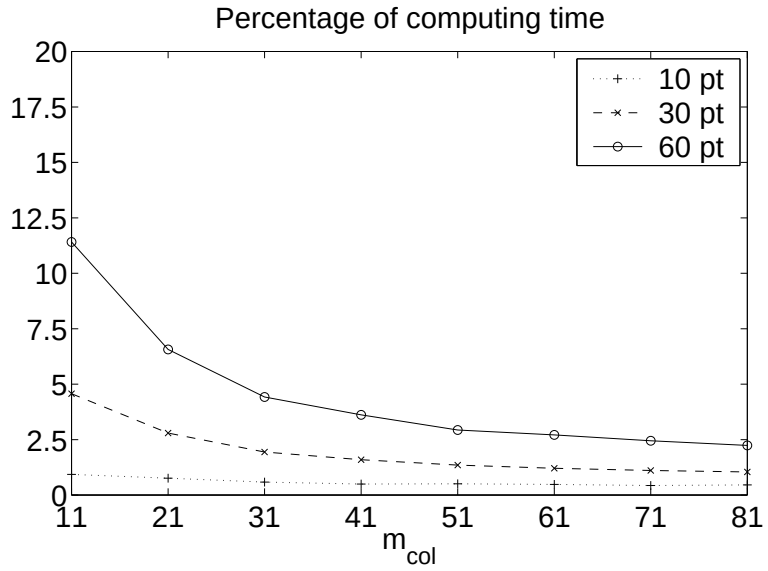


Figure 5.30.: Percentage of total computing time of optimization algorithm spend for solving the QP-problem with fixed initial data depending on the number of collocation mesh points $m_{\text{col}}$ for different numbers of spatial grid points $m_{\text{spa}}$.

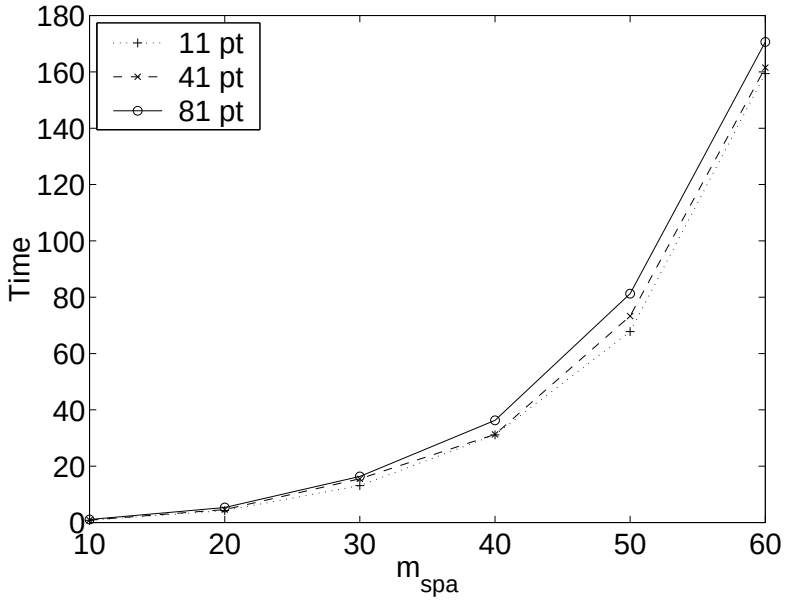


Figure 5.31.: Time spend on solving the QP-problem depending on the number of spatial grid points $m_{\text{spa}}$ for different numbers of collocation mesh points $m_{\text{col}}$.

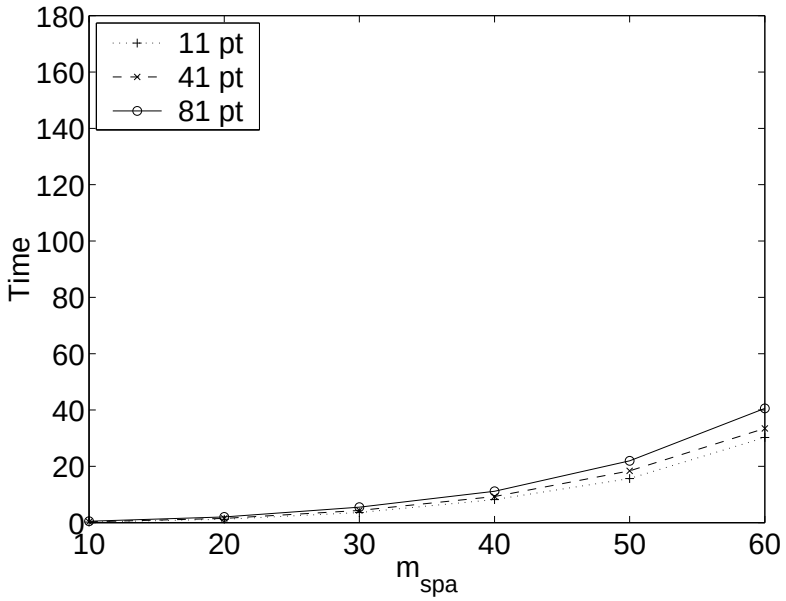


Figure 5.32.: Time spend on solving the QP-problem with fixed initial data depending on the number of spatial grid points $m_{\text{spa}}$ for different numbers of collocation mesh points $m_{\text{col}}$.

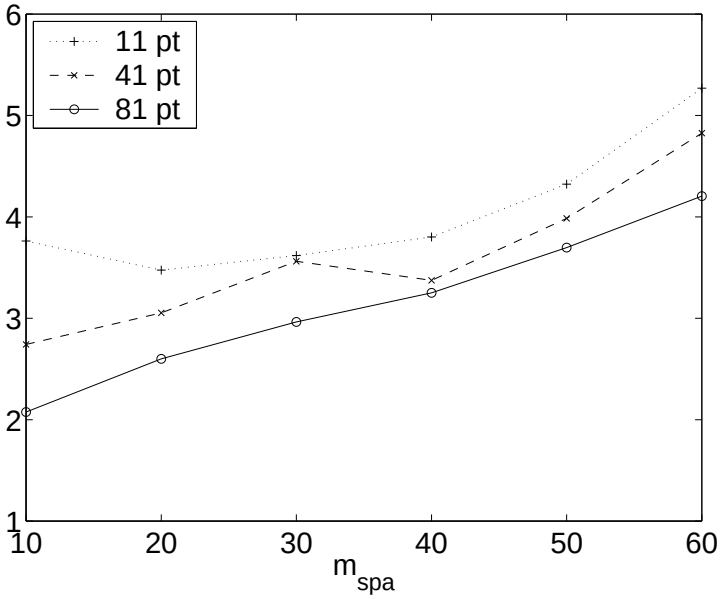


Figure 5.33.: The solver for the QP-problem with fixed initial data is several times faster than without fixed initial data. The results are shown depending on the number of spatial grid points $m_{\text{spa}}$ for different numbers of collocation mesh points $m_{\text{col}}$

# 6. Numeric Results

In this chapter optimal solutions to the application problems introduced in chapter 2 are presented. Several observations have been made during the optimization process which might show directions for further research.

The first problem, the catalytic conversion of methane to syngas, mainly serves as a benchmark problem. The optimal solution to this problem is presented and it is shown how the control and the objective function depend on the number of collocation mesh points $m_{\text{col}}$. The computational time spent in a $SQP$ iteration of the sparse algorithm is compared to the time for the non-sparse algorithm and it is shown that the sparse code is much faster then the non-sparse one in a real application. This enables to study many different scenarios in an acceptable time.

Then the optimal solution to the epoxidation of ethylene is presented. For this application it is not too difficult to find suitable spatial discretizations so one can look at how the optimal control code performs with respect to different spatial and time discretizations. In this context it is especially interesting to look at the behavior of the objective function and the controls.

The last section of this chapter is devoted to the optimal control problem of the catalytic oxygen free conversion of methane to ethane. From the viewpoint of optimal control this is the most interesting example because it is a transient system. An optimal solution is presented which could not have been found with the current simulation tool on a trial-and-error basis. Again it is shown how the control and the objective function behave when the time discretization is changed.

## 6.1. Catalytic Conversion of Methane to Syngas

A solution to this problem which has been obtained through simulation is presented in section 2.1. This solution is now taken as a starting point. So starting from a mol fraction of 0.41 for $CH_4$ at the inflow on has to find an optimal ratio of $CH_4/O_2$ of mol fractions (keeping the sum of the mol fractions of $CH_4$ and $O_2$ constant) such that the ratio of the fluxes $CO/CO_2$ at the outflow is maximized. A non-uniform spatial grid with 28 points, which has been obtained from the simulation, and a uniform time discretization of 28 points has been used to obtain an optimal solution which is shown in table 6.1 and figures 6.1, and 6.2. Table 6.1 shows the coverages of the different surface species and figure 6.1 the mol mass of the gas phase species. Figure 6.2 finally shows the behavior of the flow variables. Starting from an initial solution with the mol fraction of 0.41 for $CH_4$ at the inflow the algorithm needs 29 $SQP$ iterations to converge. It takes 287 s for the sparse algorithm to converge whereas the non-sparse algorithm needs 1201 s. In this optimal control problem the value of the control has

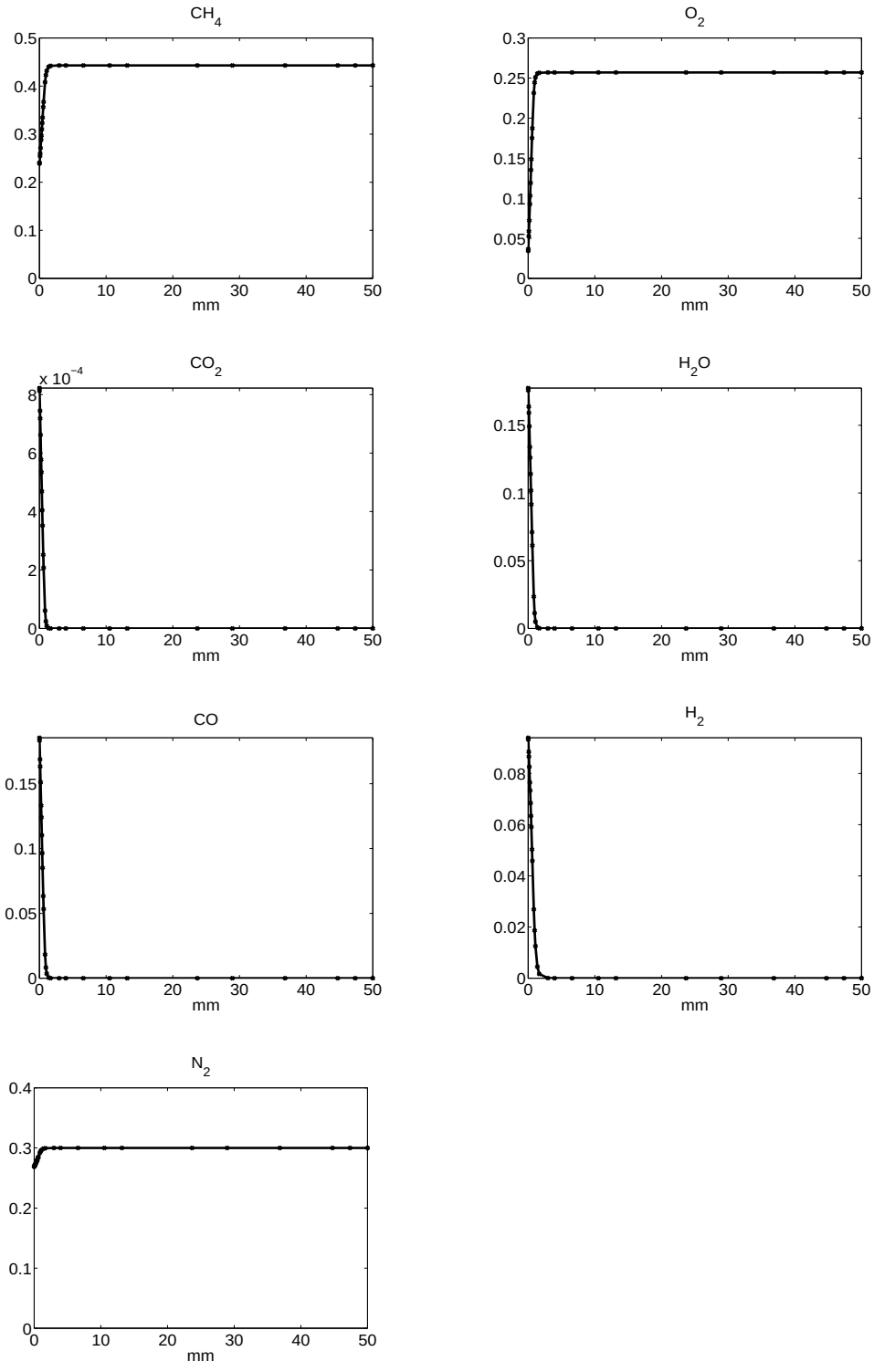


Figure 6.1.: Mol fraction of gas phase species for the optimal solution of the syngas problem.

| Species | Coverage |
|---|---|
| Pt(s) | $5.58 \cdot 10^{-1}$ |
| H(s) | $8.35 \cdot 10^{-3}$ |
| $CH_3$(s) | $5.28 \cdot 10^{-7}$ |
| $CH_2$(s) | $2.85 \cdot 10^{-9}$ |
| CH(s) | $6.87 \cdot 10^{-10}$ |
| C(s) | $5.04 \cdot 10^{-3}$ |

| Species | Coverage |
|---|---|
| O(s) | $3.92 \cdot 10^{-7}$ |
| $H_2O$(s) | $1.73 \cdot 10^{-4}$ |
| OH(s) | $2.90 \cdot 10^{-5}$ |
| CO(s) | $4.28 \cdot 10^{-1}$ |
| $CO_2$(s) | $4.04 \cdot 10^{-10}$ |

Table 6.1.: Coverages for surface species for the optimal solution of the syngas problem

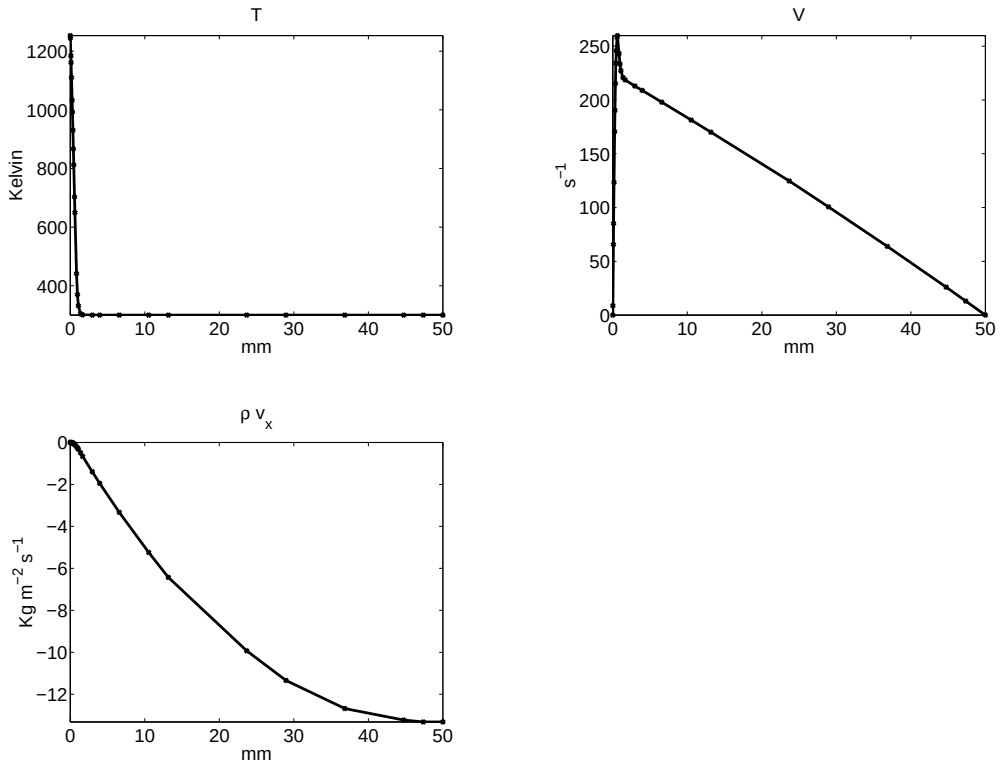


Figure 6.2.: Flow variables for the optimal solution of the syngas problem.

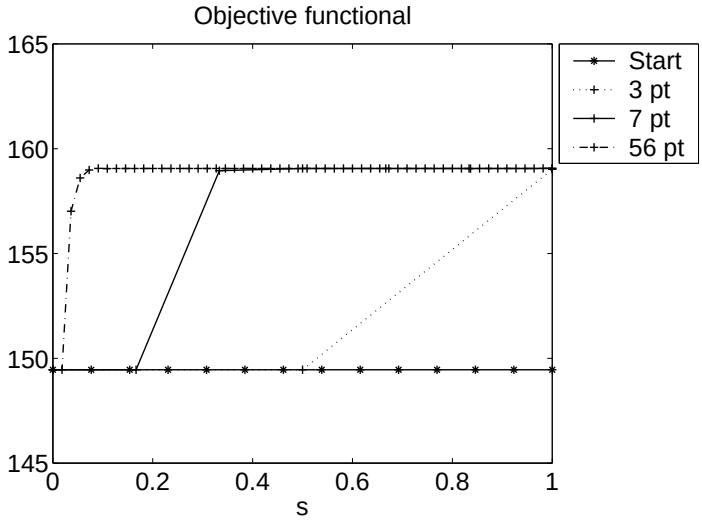


Figure 6.3.: Objective functional, the ratio of the fluxes $CO/CO_2$, depending on time and different numbers of collocation mesh points $m_{\text{col}}$.

been fixed at $t = 0$ to the initial value. The control, that is the mol fraction of the $CH_4$ of the optimal solution has increased from 0.41 to 0.443, that is the ratio of $\frac{CH_4}{O_2}$ has increased from 1.414 to 1.724. At the same time the value of the objective function, that is the ratio of the fluxes CO and $CO_2$ has increased from 149.46 (section 2.1) to now 159.06. Figure 6.3 shows the objective function for the initial solution and for different time discretizations. Of interest for the final problem solution is only the constant value and one can see that this value is already obtained at the end of the time interval with a time discretization of only three points. The next figure 6.4 shows how the control behaves when the time discretization changes. Already with three points the optimal value is obtained. Increasing the number of mesh points shows that some oscillations at the beginning of the time interval occur. At the end of the time interval there is a non-physical decrease of the control when the discretization increases. So far these observations cannot be accounted for. The last figure 6.5 in this section shows that the effort spent in identifying the structure of this problem class and using it in several places to reduce the computational cost has really payed off because for the problem with 56 collocation mesh points the sparse algorithm is 4.3 times faster than the non-sparse.

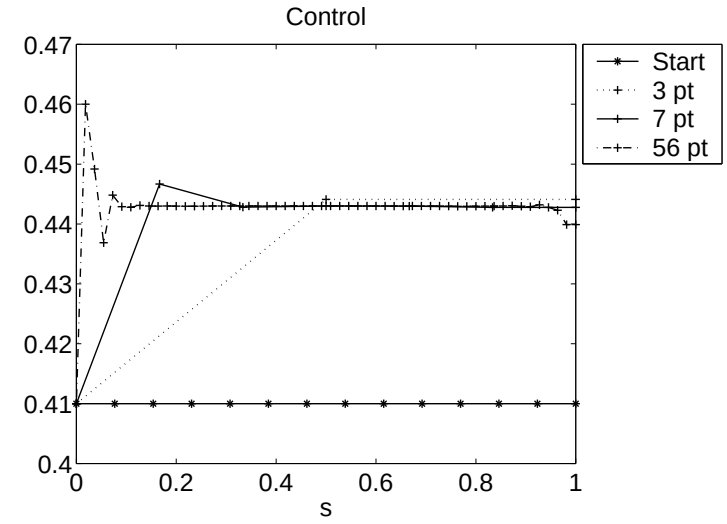


Figure 6.4.: Optimal control, mol fraction of $CH_4$, depending on time and different numbers of collocation mesh points $m_{\text{col}}$.

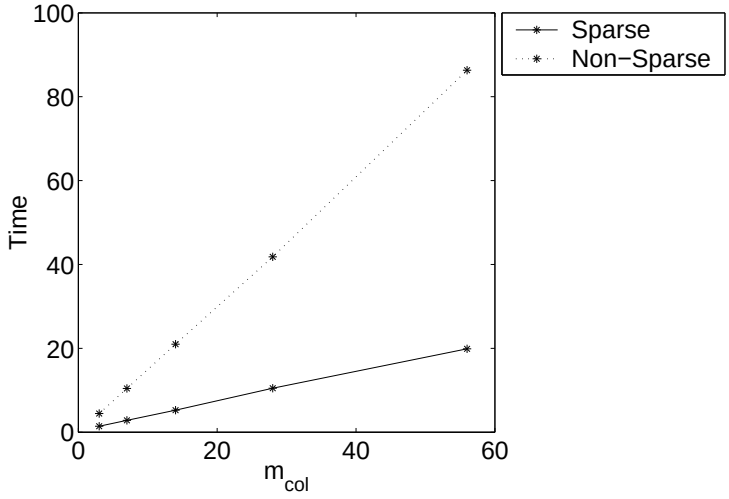


Figure 6.5.: Time per SQP-iteration for non-sparse and sparse code.

## 6.2. Epoxidation of Ethylene on Silver

The results of a simulation of this process is presented in section 2.2. The objective function of the optimization is the production rate of $C_2H_4O$ which has to be maximized. The parameters which can be controlled are the $\frac{C_2H_4}{O_2}$ ratio of mol fractions and the temperature of the catalytic plate. This optimal control problem has been solved with 40 points for the spatial and 40 points for the time discretization. Table 6.2 shows the coverages of the surface species and figure 6.6 the mol fractions of the gas phase species. The flow variables can be seen in figure 6.7. The optimization decreases the mol fraction of $C_2H_4$ at the inflow from 0.045 to 0.0368 yielding an optimal ratio of $\frac{C_2H_4}{O_2} = 0.257$. The temperature has decreased to 525 K which is set as a lower bound since. It would decrease further so it has been decided to set a value of 525 K as operating value for this process. An increase of the production rate from $6.10 \cdot 10^{-5}$ mol/m$^2$s to $6.50 \cdot 10^{-5}$ mol/m$^2$s is obtained. This is about 6.6 percent. The sparse algorithm needs 22 *SQP* iterations and 530 s in contrast to the non-sparse algorithm which needs 2450 s.

As in the last section it is interesting to look at the performance of the algorithm if the discretization is changed. The primary interest is not so much a very accurate solution of the model equations but the accuracy of the control and the objective function. In figure 6.8 the number of spatial grid points has been fixed to 40 and the objective function is shown for different numbers of collocation mesh points $m_{\text{col}}$. The values of the objective function for the starting solution are plotted as a reference. Already with 14 collocation mesh points a good approximation of the final values of the objective function is obtained. Almost a constant since a stationary problem is considered. In figure 6.9 the control is shown for 40 spatial grid points and different numbers of collocation points. This is the control corresponding to the objective function shown in figure 6.8. There again a good approximation of the constant control of a mol fraction of 0.0368 for $C_2H_4$ is already obtained with 14 collocation mesh points. So far it is not possible to account for the oscillations of the control at the beginning of the time interval and the behavior of the control and the objective function at the end of the time interval.

In the next two figures 6.10 and 6.11 the objective function and the control are shown for an optimization wit 20 spatial grid points and the same number of col-

| Species | Coverage | Species | Coverage |
|---|---|---|---|
| Ag(s) | $9.72 \cdot 10^{-1}$ | HCOO(s) | $1.33 \cdot 10^{-2}$ |
| O(s) | $2.54 \cdot 10^{-5}$ | H(s) | $5.94 \cdot 10^{-3}$ |
| $C_2H_3$(s) | $5.31 \cdot 10^{-6}$ | $CO_3$(s1) | $9.47 \cdot 10^{-7}$ |
| $H_2O$(s) | $6.84 \cdot 10^{-4}$ | $C_2H_4$(s) | $1.82 \cdot 10^{-5}$ |
| OH(s) | $1.49 \cdot 10^{-9}$ | | |
| $C_2H_3O$(s) | $4.56 \cdot 10^{-3}$ | | |
| HCO(s) | $7.58 \cdot 10^{-4}$ | | |
| $CH_2O$(s) | $2.40 \cdot 10^{-3}$ | | |

Table 6.2.: Coverages of the surface species for optimal solution of epoxidation of ethylene.

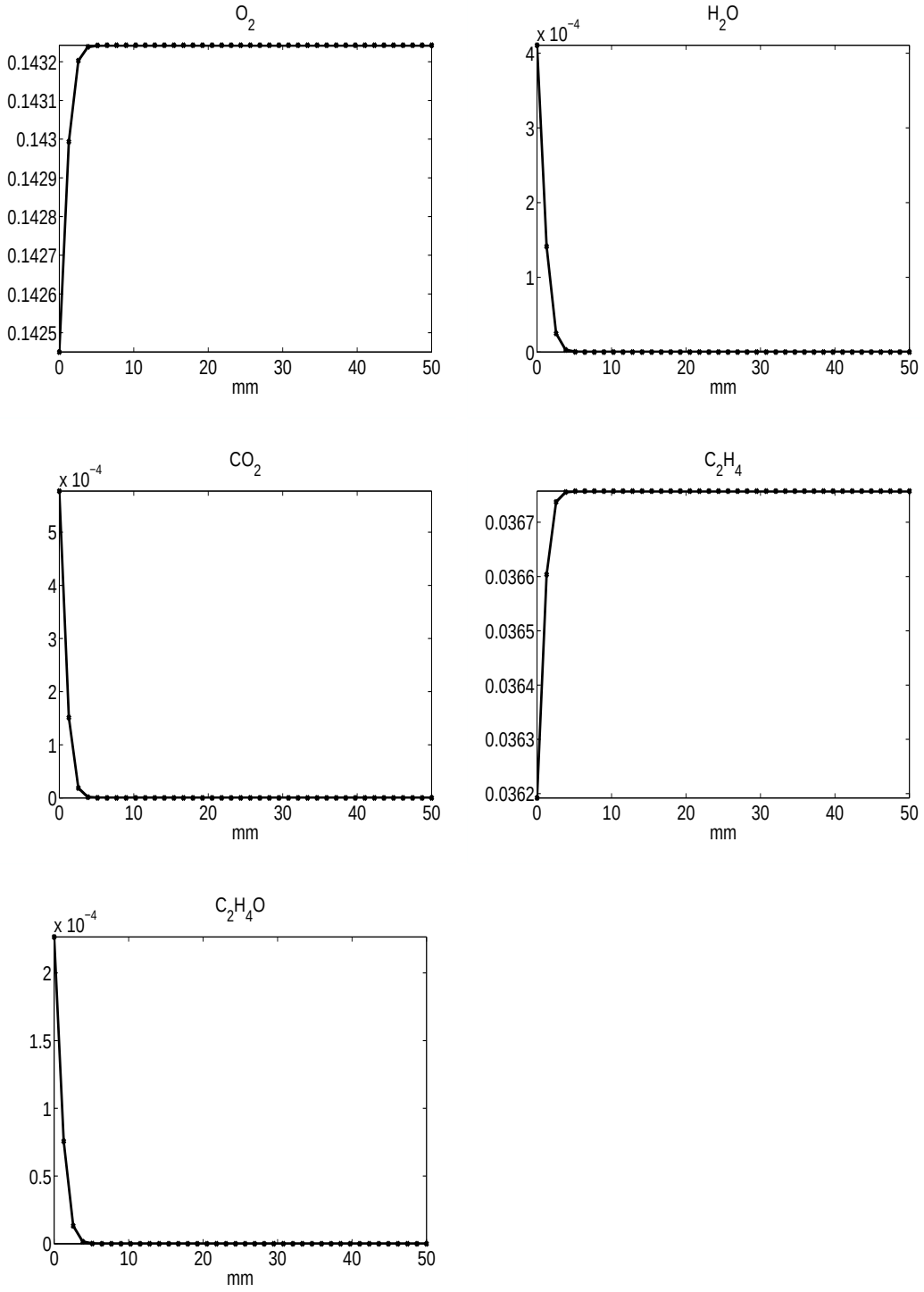


Figure 6.6.: Mol fraction of gas phase species for optimal solution of epoxidation of ethylene.

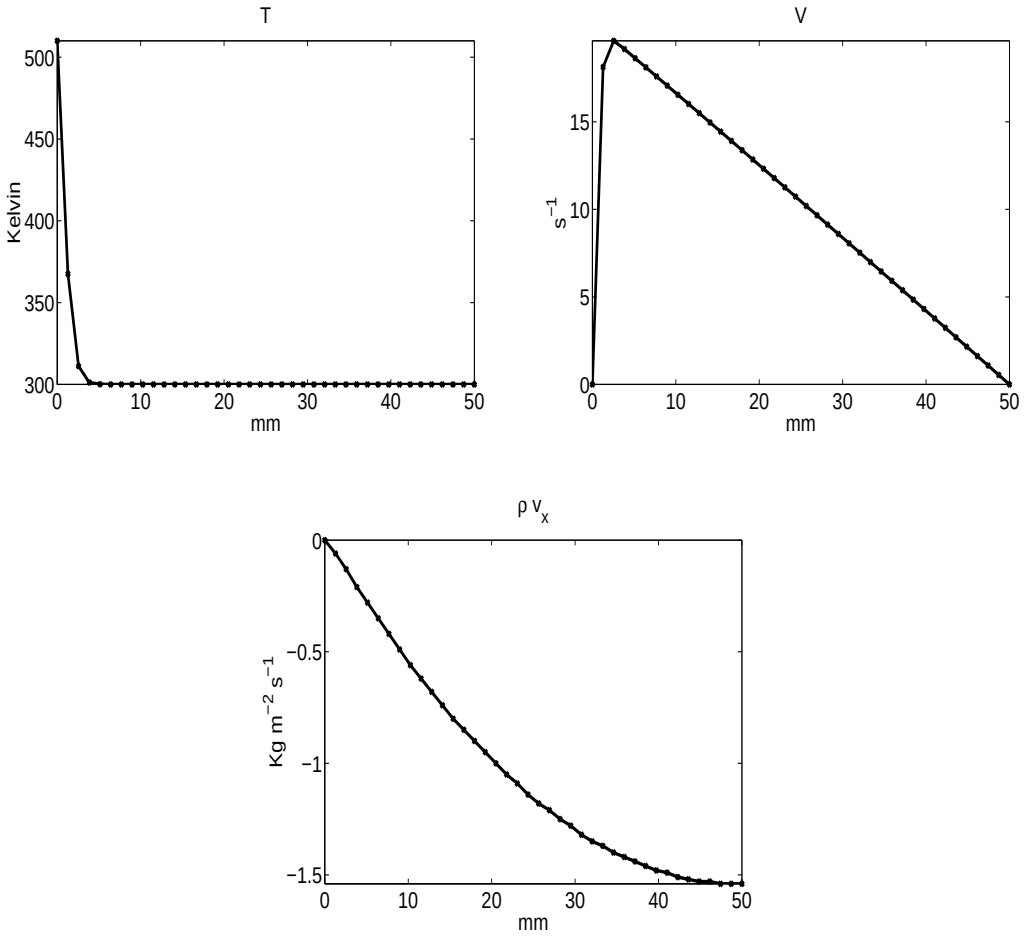


Figure 6.7.: Flow variables for optimal solution of epoxidation of ethylene.

location points as before. All other parameters are kept the same. The interesting observation is that it has almost no influence on the final shape of the objective function or control. But, the constant values of both, that are the numbers we are finally interested in, are slightly different than before. So one could say that only a rather coarse discretization is needed to solve the optimal control problem but to get better values a finer discretization is necessary.

As in the last section a look is taken at the computing times. The table 6.3 shows that there is a great difference between the times needed to compute a solution with the non-sparse and the sparse version of the algorithm. The sparse version is up to a factor of 5 faster than the non-sparse. The observations of the last paragraph show that only a coarse discretization of the problem is necessary to obtain good first values for the control and the objective function. So choosing 20 points in space and 14 points in time gives already a satisfactory answer in about 1 minute.

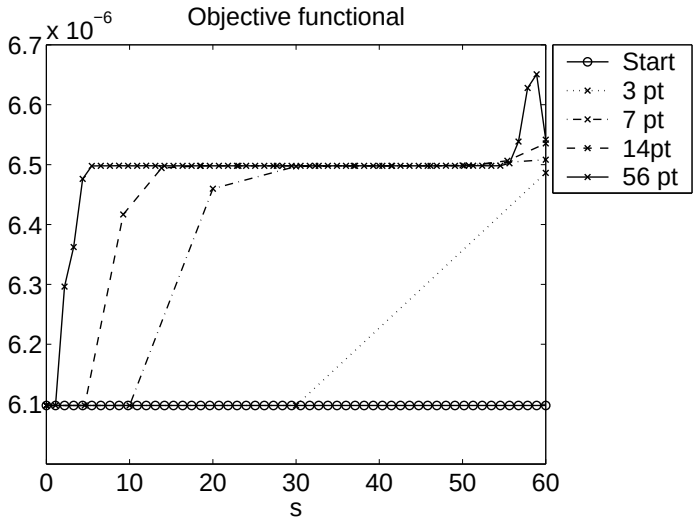


Figure 6.8.: Objective function, the production rate of $C_2H_4O$, depending on time and different numbers of collocation mesh points $m_{\text{col}}$ for a fixed number of spatial grid points $m_{\text{spa}} = 40$.

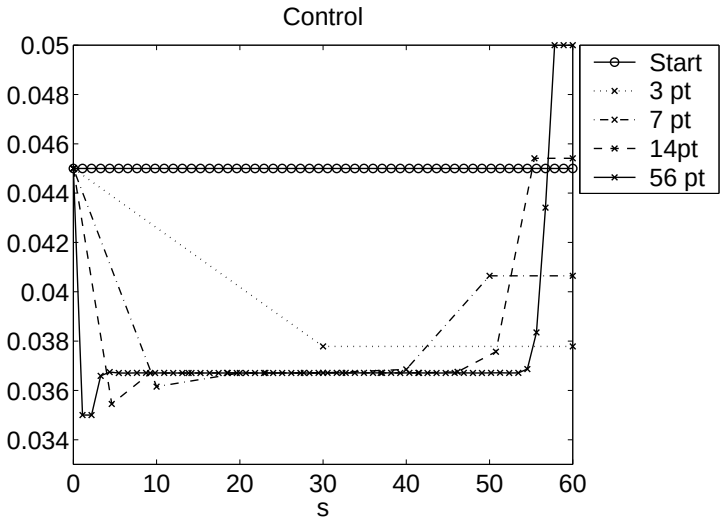


Figure 6.9.: Optimal control, the $C_2H_4$ mol fraction at the inflow, depending on time and different numbers of collocation mesh points $m_{\text{col}}$ for a fixed number of spatial grid points $m_{\text{spa}} = 40$.

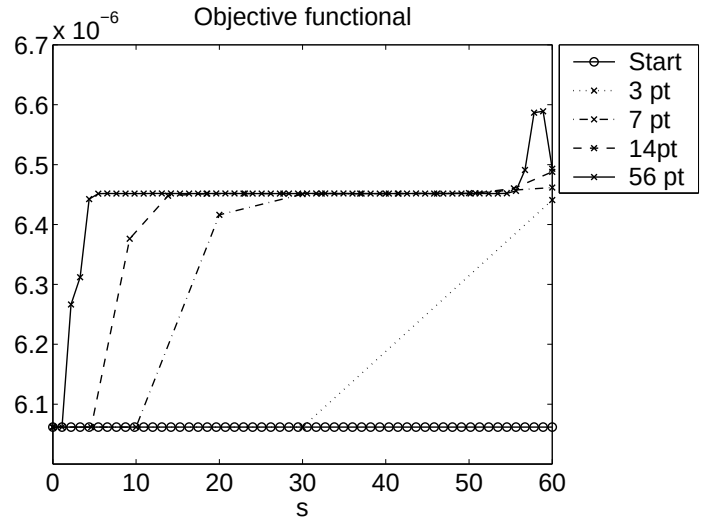


Figure 6.10.: Objective function, the production rate of $C_2H_4O$, depending on time and different numbers of collocation mesh points $m_{\text{col}}$ for a fixed number of spatial grid points $m_{\text{spa}} = 40$.

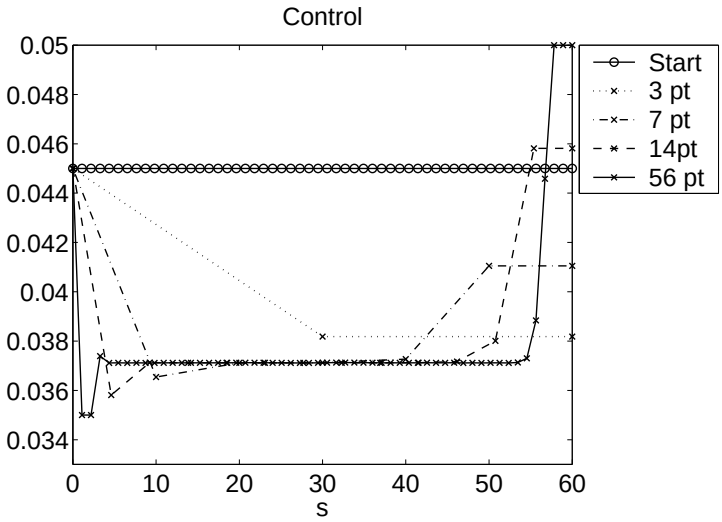


Figure 6.11.: Optimal control, the $C_2H_4$ mol fraction at the inflow, depending on time and different numbers of collocation mesh points $m_{\text{col}}$ for a fixed number of spatial grid points $m_{\text{spa}} = 40$.

| | | 3pt | 7 pt | 14 pt | 28 pt | 56 pt |
|---|---|---|---|---|---|---|
| 40 pt | full | 156 | 462 | 865 | 1695 | 3427 |
| | sparse | 38 | 84 | 163 | 332 | 654 |
| 20 pt | full | 41 | 91 | 172 | 349 | 733 |
| | sparse | 16 | 27 | 57 | 111 | 237 |

Table 6.3.: Total time for *OCPRSQP* in seconds.

## 6.3. Catalytic Oxygen-Free Conversion of Methane to Ethane

The solution of a simulation of the catalytic oxygen-free conversion of methane to ethane is presented in section 2.3. As mentioned earlier and in contrast to the previous two applications this one is time-dependent. So a time-dependent control is expected. A difficulty for the setup of this problem comes directly from the chemistry: After a certain time the catalytic surface is covered to a high percentage with carbon reducing considerably the surface activity. In the previous applications it has been possible to generate initial values by just computing a simulation and then using the result as starting point for the optimization. This is not possible in this case because due to the pollution with carbon the optimization would start with a not so interesting situation. Since well defined initial values are needed for the optimization it is not possible to start from 'scratch' as it is done in simulations using *DIFRUN*. The solution of these difficulties is to do a simulation for a short time compared to the overall time interval of the optimization and then using this initial data for an optimization on the remaining time interval. 'Short' in this context means that this time should not play a role for the optimization process but should be long enough to obtain well defined initial data. For the application under investigation it turns out that computing a simulation for 6 s can provide the necessary initial data and is still short enough compared to the whole process time of 300 s. For the reduction of the computing time, if testing different time discretizations, a simulation is computed for the first 5 s—which takes around 2 min—and then used as input to *DIFRUN* which then computes another simulation of 1 s—this takes about 2 s—before starting the optimization.

As mentioned in section 2.3 there are three possible controls: the temperature of the catalytic plate, the pressure, and the inflow velocity. A few simulations have been done to obtain an idea for good starting values for these parameters. It turns out that the temperature and the pressure have a fairly clear influence on the objective function, the production rate of $C_2H_6$. The higher the temperature or pressure, the higher the production rate of $C_2H_6$. These values would always approach the bounds and for this reason they are set to 523 K and 1 bar to study the effect of the inflow velocity on the production rate of $C_2H_6$. Like in [60] three different fixed inflow velocities $v = 0.01$ cm/s, or 0.1 cm/s, or 53 cm/s are chosen and the results for the production rate for these simulations can be seen in figure 6.12. It is obvious that the inflow velocity $v = 0.1$ cm/s produces the largest integrated production rate of all three simulations. This can also be seen from the following table:

| $v$[ cm/s] | integrated production rate [ mol/m$^2$] |
|---|---|
| 53 | $4.4194 \cdot 10^{-6}$ |
| 0.1 | $9.8185 \cdot 10^{-6}$ |
| 0.01 | $5.8552 \cdot 10^{-6}$ |

The inflow velocity of $v = 0.1$ cm/s is now used as a starting value for the optimization. As described above a simulation is then computed with these parameter values for the first six seconds and then the optimization starts. The values for these first

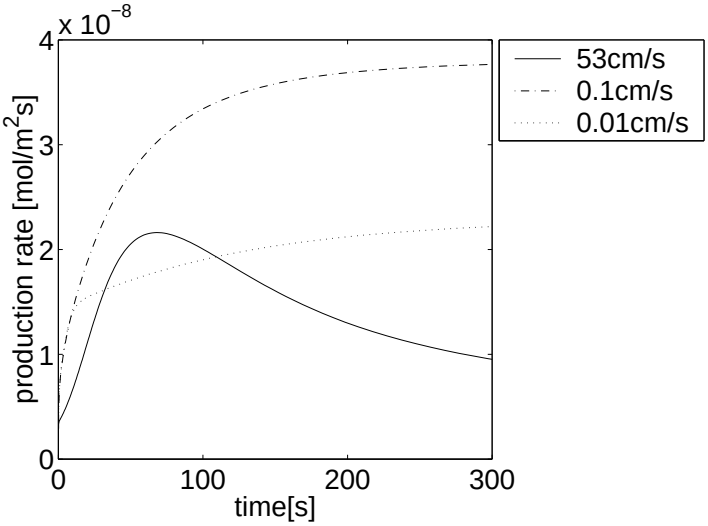


Figure 6.12.: Production rate of $C_2H_6$ depending on time for different inflow velocities.

six seconds are not shown in the following figures because they do not play any role for the optimization.

For practical reasons the control is bounded from below by 0.05 cm/s and from above by 1 cm/s. A uniform mesh of 30 collocation points and 30 spatial grid points is chosen. Figure 6.13 then shows the result of this optimization for the production rate of $C_2H_6$ (dotted line) compared to the corresponding production rate for the constant inflow velocity (solid line). Integrating this rate gives a rise of 3% compared to the initial profile. The next figure 6.14 shows the corresponding control to this optimal solution. The inflow velocity increases immediately to 1 cm/s, the upper bound, then after around 40 s decreases fast to approach a value around 0.1 cm/s. It then decreases slowly until hitting the lower bound of 0.05 cm/s at around 210 s at which it stays up to the end at 300 s. The shape of this solution is stable if the collocation discretization is changed (figures 6.15 and 6.16). Already with 10 collocation mesh points the general shape of the new control and the objective function are visible. If the collocation mesh gets refined the solution then seems to converge to a specific shape which can be seen looking at the solutions for 20, 30, and 50 collocation mesh points.

To complete the presentation of an optimal solution to this application problem figures 6.17 and 6.18 show the coverages of the surface species. Figure 6.19 shows the mol fraction of the gas phase species and figure 6.20 the flow variables. To conclude this section a look is taken at the number of variables and computing times for this problem.

This application contains 14 surface and 4 gas-phase species which leads to a system of 14 *ODE* and 8 partial differential equations. For the spatial discretizations 30 grid points are used which leads to a system of 254 *DAE*. The following table shows the

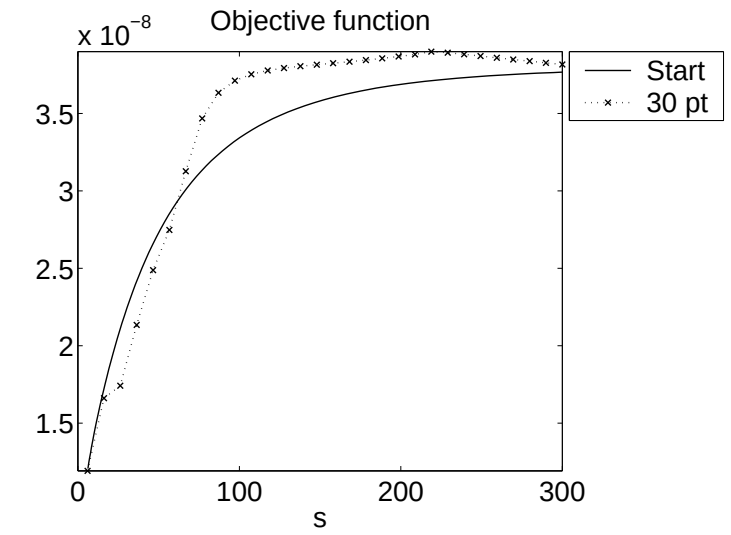


Figure 6.13.: Objective function, production rate of $C_2H_6$, depending on time for an optimal solution.

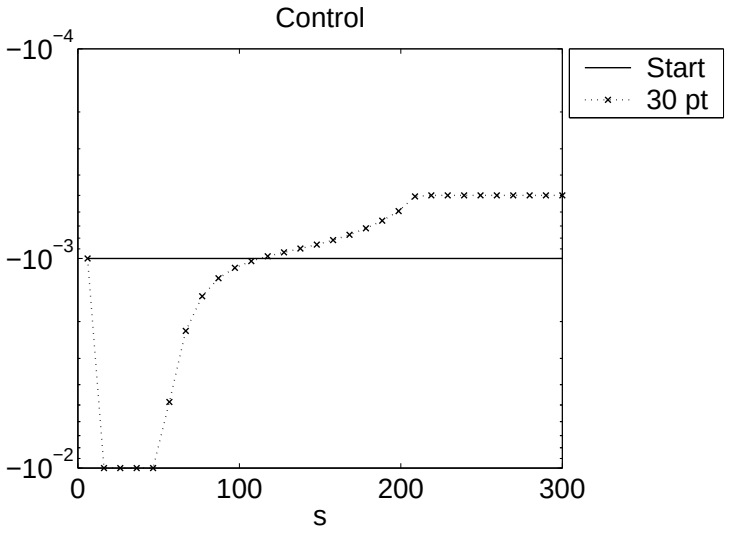


Figure 6.14.: Optimal control, inflow velocity, depending on time for an optimal solution.

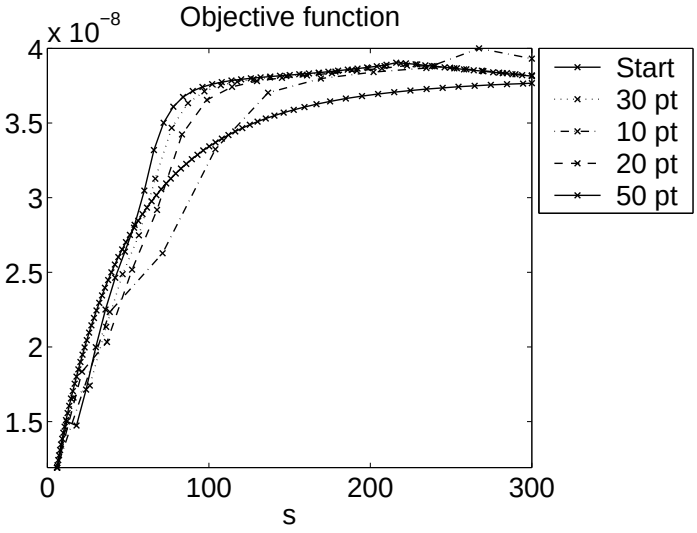


Figure 6.15.: Objective function, production rate of $C_2H_6$, depending on time for optimal solution for different numbers of collocation mesh points $m_{\text{col}}$.

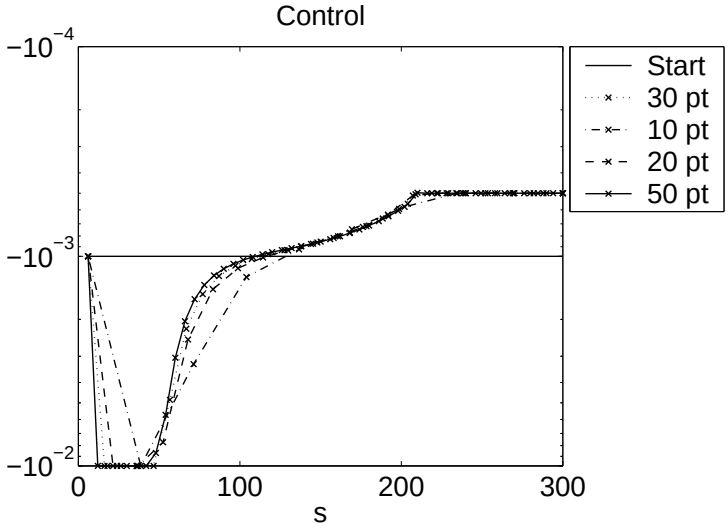


Figure 6.16.: Optimal control, inflow velocity, for different numbers of collocation mesh points $m_{\text{col}}$.

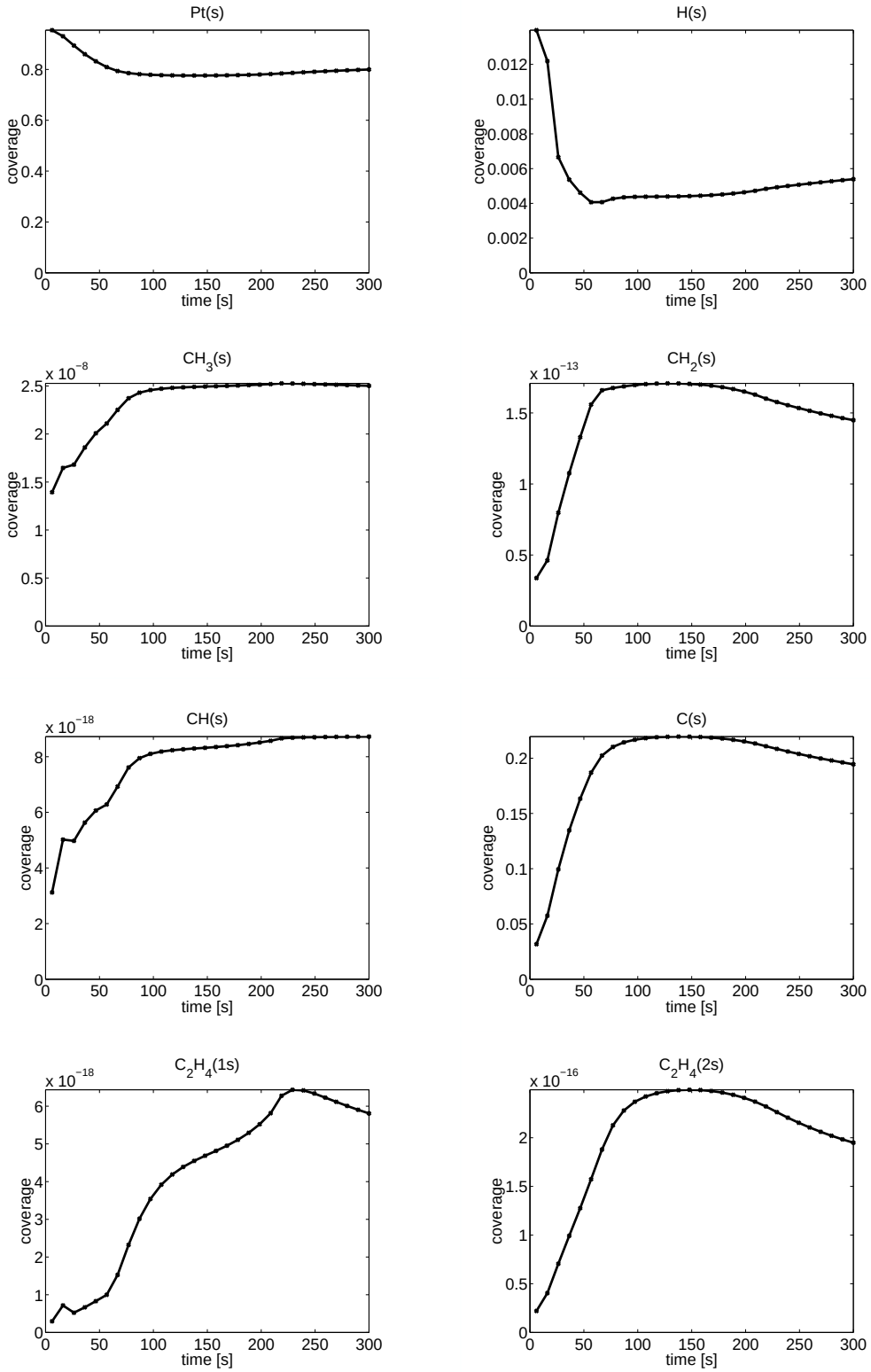


Figure 6.17.: Coverages of the surface species for optimal solution of the catalytic oxygen-free conversion of methane to ethane.

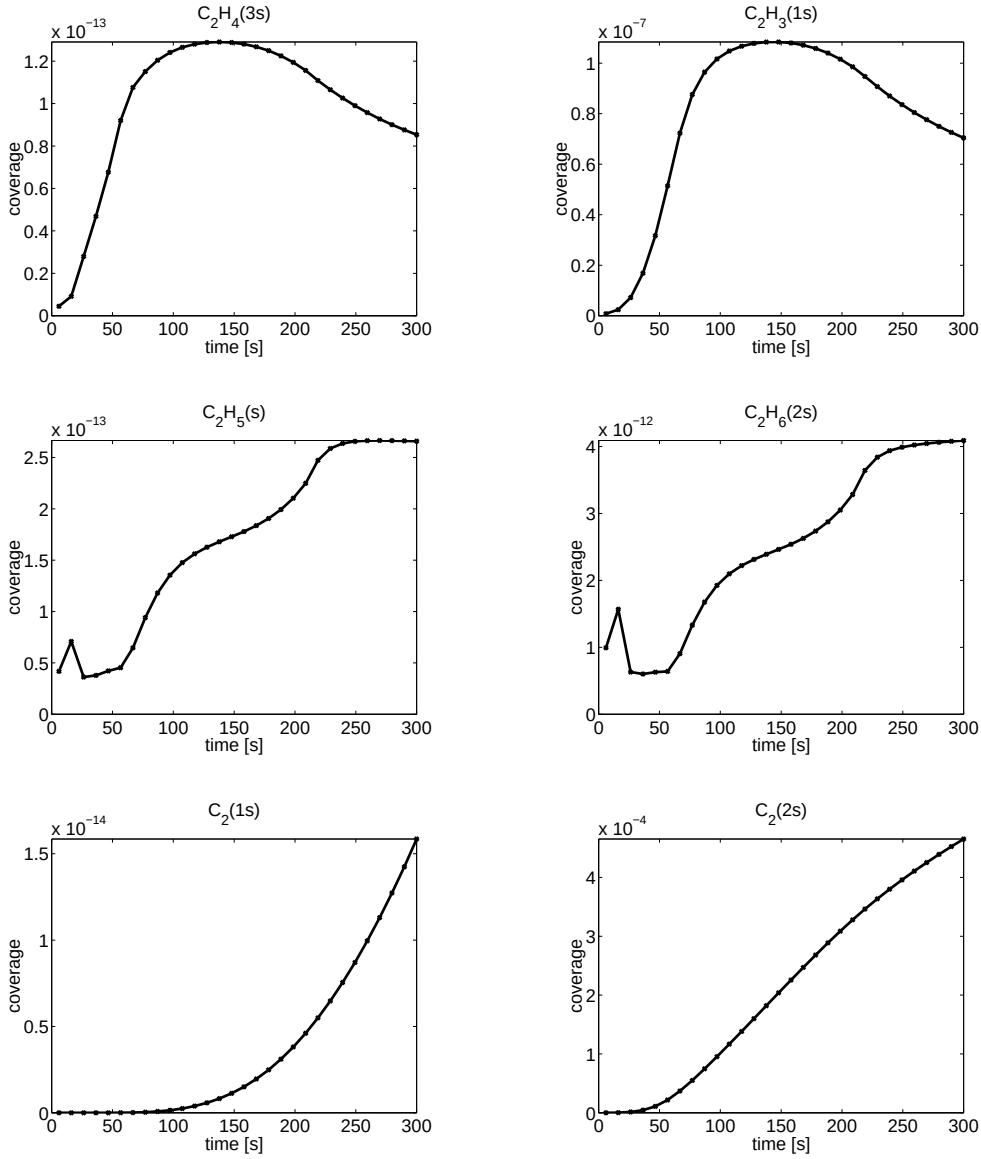


Figure 6.18.: Coverages of the surface species for optimal solution of the catalytic oxygen-free conversion of methane to ethane.

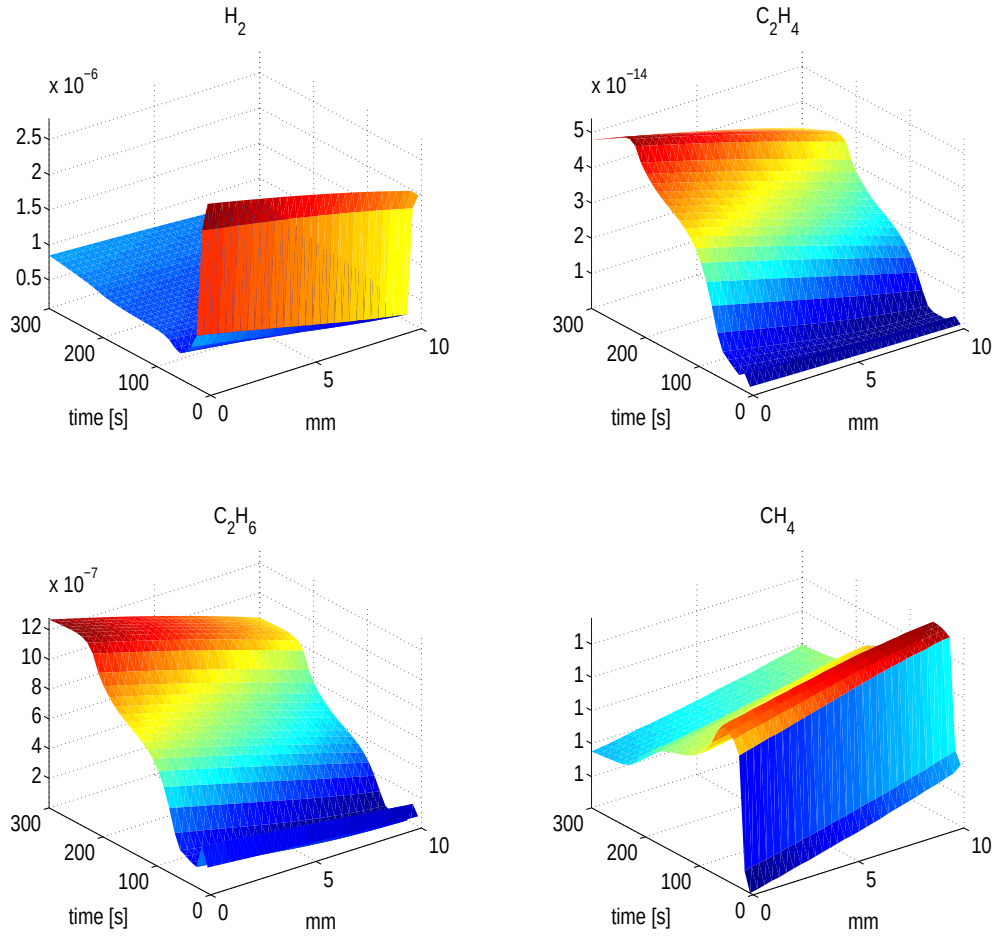


Figure 6.19.: Mol fraction of gas phase species for optimal solution of the catalytic oxygen-free conversion of methane to ethane.

total number of variables and constraints—collocation, continuity, boundary, and path constraints—for this problem after the collocation discretization.

| collocation mesh points | variables | constraints |
|---|---|---|
| 10 | 6070 | 5964 |
| 20 | 12460 | 12344 |
| 30 | 18850 | 18724 |
| 50 | 31630 | 31484 |

The computing times are quite high, between 6 min for the sparse version and $m_{\text{col}} = 10$ and 5 h for the non-sparse algorithm and $m_{\text{col}} = 50$. Again the great benefit of the sparse algorithm can be seen very clearly from the following table

| | | 10pt | 20 pt | 30 pt | 50 pt |
|---|---|---|---|---|---|
| 30 pt | full | 1417 | 5038 | 12725 | 13353 |
| | sparse | 408 | 1042 | 2438 | 2974 |

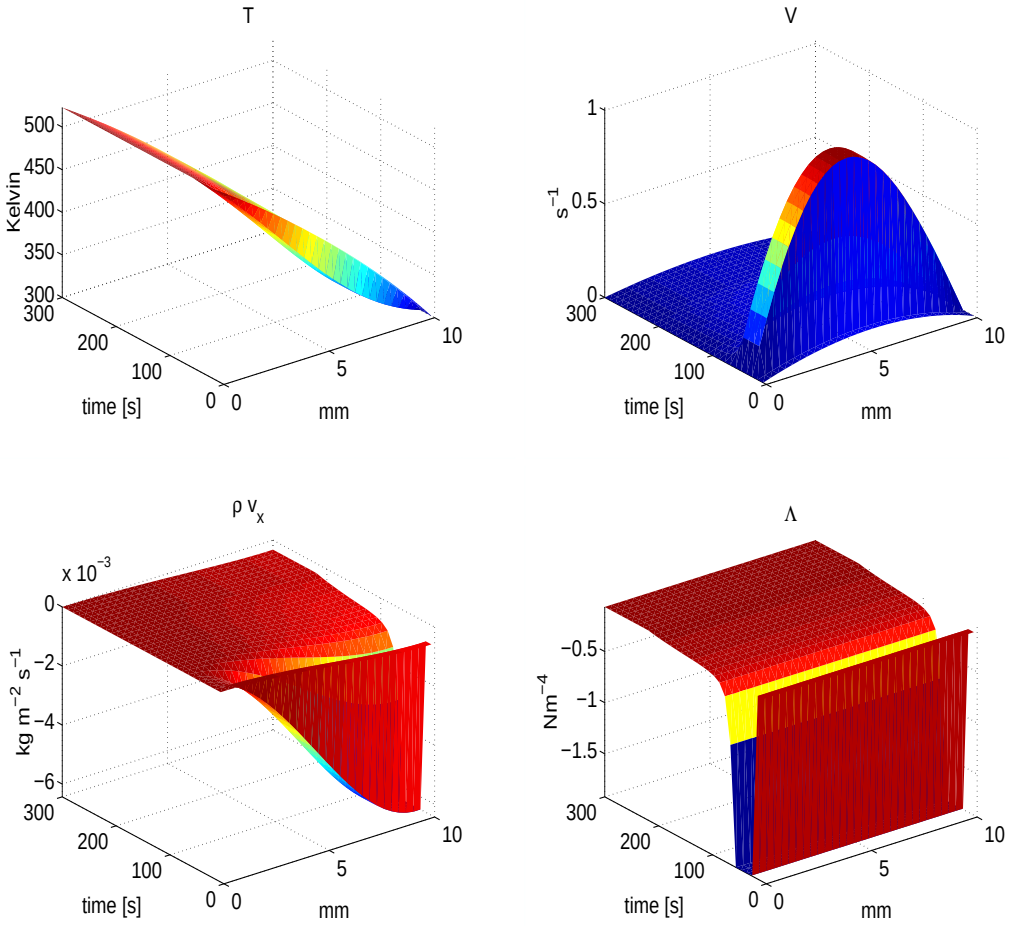


Figure 6.20.: Flow variables for optimal solution of the catalytic oxygen-free conversion of methane to ethane.

in which the computing time for the optimization is given in seconds.

# 7. Conclusions and Outlook

The work in this thesis focused on the development of a reliable and easy-to-use software package for the optimal control of a reactive stagnation point flow on a catalytic plate. Several problems on the software engineering, the algorithmic and the application level had to be solved to reach this goal. The code has been applied to three application problems to illustrate its performance.

This thesis also shows the need for further research in this area. There is still potential in reducing the computing time even further. For example, the time for computing the Jacobian could be reduced considerably by using new specialized automatic differentiation tools, which would have to be developed along the lines of the work by Rücker [46] taking into account the problem structure and the structure of chemical equations. This would also reduce the high memory requirements, which in turn enables to study more realistic chemical models with dozens of species resulting in thousands of *DAE*.

The methodology of the successful coupling can now be imitated and applied to other reactor configurations such as the reactive channel flow. For this system the simulation tool is now finished and available. Another potential area of research is the study of the effect of the spatial and time discretization on the quality of the control.

The new software is based on two existing packages: *DIFRUN*, a tool for the simulation of a reactive stagnation point flow on a catalytic plate, and *OCPRSQP*, an optimal control package based on a collocation discretization in time and a partially reduced *SQP* method to solve the optimization problem.

In a first step these two packages have been coupled to provide a solid basis for further developments. Here the major two difficulties were the identification of the interface to the discretized model equations in *DIFRUN* and the development of a new interface for *OCPRSQP* to use analytic derivatives provided by *ADIFOR* as a substitute for the finite differences, which had been used before but turned out to be not accurate enough for this problem class. This first step was successfully finished, keeping the enormous flexibility of *DIFRUN* to modify the chemical systems studied with minimal effort. Furthermore only a very limited number of unavoidable changes have to be made to the new code to switch from one chemical system to another.

Since the number of equations in the considered problem class can easily reach several hundred or even thousands of *DAE*, the pure coupling of the two packages did not provide an efficient tool. The computing times were too high and more effort had to be made on the algorithms, for example using the problem structure when computing the Jacobian, or for the condensing step of the partially reduced *SQP*-algorithm. This new algorithms reduced considerably the computing time.

Furthermore it has been focused to apply the new package for the optimization of practical problems. Three problems have been studied to demonstrate the performance of this new tool: The catalytic partial oxidation of methane to syngas, the epoxidation of ethylene on silver and the catalytic oxygen-free conversion of methane to ethane. Optimal solutions have been presented. In the case of the catalytic oxygen-free conversion of methane to ethane, it has been possible to present a solution which could not have been obtained on a trial-and-error basis using the original *DIFRUN*.

For the first time a software package is available for the very fast computation of the optimal control for *arbitrary* chemical systems in a reactive stagnation point flow on a catalytic plate. The user of the software needs only little effort to set up a new problem.

# A. Reaction Mechanisms

In this chapter, all surface reaction mechanisms are listed. The major reference for the format of these mechanisms is [19]

Species with the suffix (s) are adsorbed species. The species named Pt(s) and Ag(s) denote uncovered surface sites available for adsorption on platinum and silver, respectively.

The kinetic data are given according to [19]. The units are $A$ [mol, cm, s], $S^0$, $\beta$ and $\mu$ [−], $E_a$ and $\epsilon$ [kJ mol$^{-1}$]. $S^0$ denotes the initial (uncovered surface) sticking coefficient. If the reaction kinetic exhibits an additional dependence on surface coverage, the line under the reaction equation names the species, to which the dependence is referred, and gives the kinetic parameters $\mu$ and $\epsilon$, $\mu$ and $\epsilon$ are zero for all other reactions.

## A.1. Catalytic Partial Oxidation of Methane to Syngas

| Reaction mechanisms | $A$ / $S^0$ | $\beta$ / $\mu$ | $E_a$ / $\epsilon$ |
|---|---|---|---|
| Adsorption | | | |
| $H_2$ +Pt(s) +Pt(s) → H(s) +H(s) | 0.046E-00 | 0.0 | 0.0 |
| \$Pt($s$) | 0.0 | -1.0 | 0.0 |
| $O_2$ +Pt(s) +Pt(s) → O(s) +O(s) | 1.891E+21 | -0.5 | 0.0 |
| $CH_4$ +Pt(s) +Pt(s) → $CH_3$(s) +H(s) | 0.600E+00 | 0.0 | 52.0 |
| $H_2O$ +Pt(s) → $H_2O$(s) | 7.500E-01 | 0.0 | 0.0 |
| $CO_2$ +Pt(s) → $CO_2$(s) | 5.000E-03 | 0.0 | 0.0 |
| CO +Pt(s) → CO(s) | 8.400E-01 | 0.0 | 0.0 |
| Desorption | | | |
| H(s) +H(s) → Pt(s) +Pt(s) +$H_2$ | 3.700E+21 | 0.0 | 67.4 |
| \$H(s) | 0.0 | 0.0 | 6.0 |
| O(s) +O(s) → Pt(s) +Pt(s) +$O_2$ | 3.700E+21 | 0.0 | 213.2 |
| \$O(s) | 0.0 | 0.0 | 188.3 |
| $H_2O$(s) → $H_2O$ +Pt(s) | 4.500E+12 | 0.0 | 41.8 |
| CO(s) → CO +Pt(s) | 1.000E+13 | 0.0 | 146.0 |
| \$CO(s) | 0.0 | 0.0 | 33.0 |
| $CO_2$(s) → $CO_2$ +Pt(s) | 1.000E+13 | 0.0 | 27.1 |
| CH3(s)+H(s) → $CH_4$ +Pt(s) +Pt(s) | 1.850E+22 | 0.0 | 21.5 |
| \$H(s) | 0.0 | 0.0 | 7.0 |

| Surface Reactions | | | |
|---|---|---|---|
| O(s) +H(s) → OH(s) +Pt(s) | 1.280E+22 | 0.0 | 11.2 |
| OH(s) +Pt(s) → O(s) +H(s) | 1.070E+21 | 0.0 | 82.5 |
| $O(s) | 0.0 | 0.0 | 73.2 |
| $H(s) | 0.0 | 0.0 | -3.0 |
| H(s) +OH(s) → $H_2O$(s) +Pt(s) | 2.040E+22 | 0.0 | 66.2 |
| $H_2O$(s) +Pt(s) → H(s) +OH(s) | 1.680E+20 | 0.0 | 106.8 |
| $O(s) | 0.0 | 0.0 | -167.4 |
| $H(s) | 0.0 | 0.0 | -3.0 |
| OH(s) +OH(s) → $H_2O$(s) +O(s) | 7.400E+21 | 0.0 | 74.0 |
| $H_2O$(s) +O(s) → OH(s) +OH(s) | 1.000E+21 | 0.0 | 43.1 |
| $O(s) | 0.0 | 0.0 | -240.6 |
| CO(s) +O(s) → $CO_2$(s) +Pt(s) | 3.700E+21 | 0.0 | 118.0 |
| $CO(s) | 0.0 | 0.0 | 33.0 |
| $CO_2$(s) +Pt(s) → CO(s) +O(s) | 3.700E+21 | 0.0 | 173.3 |
| $O(s) | 0.0 | 0.0 | -94.1 |
| C(s) +O(s) → CO(s) +Pt(s) | 3.700E+21 | 0.0 | 0.0 |
| CO(s) +Pt(s) → C(s) +O(s) | 3.700E+21 | 0.0 | 236.9 |
| $CO(s) | 0.0 | 0.0 | -95.0 |
| $O(s) | 0.0 | 0.0 | 31.3 |
| CH3(s)+Pt(s) → $CH_2$(s) +H(s) | 1.262E+22 | 0.0 | 64.8 |
| $CH_2$(s) +H(s) → CH3(s)+Pt(s) | 3.700E+21 | 0.0 | 0.0 |
| $H(s) | 0.0 | 0.0 | 7.0 |
| $CH_2$(s) +Pt(s) → CH(s) +H(s) | 7.000E+22 | 0.0 | 61.2 |
| CH(s) +H(s) → $CH_2$(s) +Pt(s) | 3.700E+21 | 0.0 | 0.0 |
| $H(s) | 0.0 | 0.0 | 7.0 |
| CH(s) +Pt(s) → C(s) +H(s) | 3.000E+22 | 0.0 | 0.0 |
| $H(s) | 0.0 | 0.0 | 7.0 |
| C(s) +H(s) → CH(s) +Pt(s) | 3.700E+21 | 0.0 | 126.3 |
| $H_2$ +C(s) +Pt(s) → CH(s) +H(s) | 5.580E+20 | 0.0 | 83.8 |
| CH(s) +H(s) → Pt(s) +C(s) +$H_2$ | 3.090E+22 | 0.0 | 30.1 |
| C(s) +$H_2$ → $CH_2$(s) | 5.870E+11 | 0.0 | 17.6 |
| $CH_2$(s) → C(s) +$H_2$ | 7.690E+13 | 0.0 | 25.1 |

Table A.1.: Surface reaction mechanisms for syngas problem. [17]

## A.2. Epoxidation of Ethylene on Silver

| Reaction mechanisms | | | | | | | $A$ / $S^0$ | $\beta$ / $\mu$ | $E_a$ / $\epsilon$ |
|---|---|---|---|---|---|---|---|---|---|
| Adsorption | | | | | | | | | |
| $O_2$ | +Ag(s) | +Ag(s) | → | O(s) | +O(s) | | 1.000E-06 | 0.0 | 0.0 |
| $C_2H_4$ | +Ag(s) | | → | $C_2H_4$(s) | | | 1.500E-03 | 0.0 | 0.0 |
| $C_2H_4$ | +Ag(s) | +Ag(s) | → | $C_2H_3$(s) | +H(s) | | 6.000E+19 | 0.0 | 8.0 |
| $CO_2$ | +O(s) | | → | CO3(s) | | | 1.000E-03 | 0.0 | 46.2 |
| $H_2O$ | +Ag(s) | | → | $H_2O$(s) | | | 7.500E-01 | 0.0 | 0.0 |
| Desorption | | | | | | | | | |
| O(s) | +O(s) | | → | Ag(s) | +Ag(s) | +$O_2$ | 2.800E+20 | 0.0 | 135.0 |
| $O(s) | | | | | | | 0.0 | -0.7 | 60.0 |
| $C_2H_4$(s) | | | → | Ag(s) | +$C_2H_4$ | | 1.000E+13 | 0.0 | 44.9 |
| $C_2H_3$(s) | +H(s) | | → | Ag(s) | +Ag(s) | +$C_2H_4$ | 3.700E+21 | 0.0 | 12.0 |
| CO3(s) | | | → | $CO_2$ | +O(s) | | 1.000E+13 | 0.0 | 147.0 |
| $O(s) | | | | | | | 0.0 | 0.0 | -40.0 |
| $H_2O$(s) | | | → | $H_2O$ | +Ag(s) | | 1.000E+13 | 0.0 | 50.4 |
| Surface Reactions | | | | | | | | | |
| $C_2H_4$(s) | +O(s) | | → | Ag(s) | +$C_2H_4O$ | +Ag(s) | 3.500E+19 | 0.0 | 8.0 |
| $O(s) | | | | | | | 0.0 | 0.0 | -40.0 |
| $C_2H_4O$ | +Ag(s) | +Ag(s) | → | $C_2H_4$(s) | +O(s) | | 1.000E+13 | 0.0 | 31.2 |
| $C_2H_3$(s) | +O(s) | | → | $C_2H_3O$(s) | +Ag(s) | | 3.700E+21 | 0.0 | 23.8 |
| $C_2H_3O$(s) | +Ag(s) | | → | $C_2H_3$(s) | +O(s) | | 3.700E+21 | 0.0 | 224.3 |
| $C_2H_3O$(s) | +O(s) | | → | HCO(s) | +$CH_2O$(s) | | 3.700E+21 | 0.0 | 54.3 |
| HCO(s) | +$CH_2O$(s) | | → | $C_2H_3O$(s) | +O(s) | | 3.700E+21 | 0.0 | 119.8 |
| $CH_2O$(s) | +O(s) | | → | HCOO(s) | +H(s) | | 3.700E+21 | 0.0 | 51.4 |
| HCOO(s) | +H(s) | | → | $CH_2O$(s) | +O(s) | | 3.700E+21 | 0.0 | 131.2 |
| HCO(s) | +O(s) | | → | HCOO(s) | +Ag(s) | | 3.700E+21 | 0.0 | 46.2 |
| HCOO(s) | +Ag(s) | | → | HCO(s) | +O(s) | | 3.700E+21 | 0.0 | 210.0 |
| HCOO(s) | +O(s) | | → | $CO_2$ | +OH(s) | +Ag(s) | 3.700E+21 | 0.0 | 56.0 |
| $CO_2$ | +OH(s) | +Ag(s) | → | HCOO(s) | +O(s) | | 3.700E+21 | 0.0 | 192.5 |
| H(s) | +O(s) | | → | OH(s) | +Ag(s) | | 3.700E+21 | 0.0 | 88.2 |
| OH(s) | +Ag(s) | | → | H(s) | +O(s) | | 3.700E+21 | 0.0 | 67.2 |
| OH(s) | +H(s) | | → | $H_2O$(s) | +Ag(s) | | 3.700E+21 | 0.0 | 8.4 |
| $H_2O$(s) | +Ag(s) | | → | OH(s) | +H(s) | | 3.700E+21 | 0.0 | 168.8 |

Table A.2.: Surface reaction mechanisms for epoxidation of ethylene. [39]

## A.3. Catalytic Oxygen-Free Conversion of Methane to Ethane

| Reaction mechanisms | $A$ / $S^0$ | $\beta$ / $\mu$ | $E_a$ / $\epsilon$ |
|---|---|---|---|
| Adsorption | | | |
| $H_2$ +Pt(s) +Pt(s) → H(s) +H(s) | 0.046E-00 | 0.0 | 0.0 |
| \$Pt(s) | 0.0 | -1.0 | 0.0 |
| $CH_4$ +Pt(s) +Pt(s) → $CH_3$(s) +H(s) | 9.000E-04 | 0.0 | 72.2 |
| $CH_4$ +C(s) → $C_2H_4$(2s) | 7.290E+01 | 0.5 | 0.0 |
| \$C(s) | 0.0 | 0.0 | -47.5 |
| \$$C_2$(2s) | 0.0 | 0.0 | -47.5 |
| $C_2H_6$ +Pt(s) +Pt(s) → $C_2H_6$(2s) | 1.000E+00 | 0.0 | 0.0 |
| $C_2H_4$ +Pt(s) → $C_2H_4(1s)$ | 1.000E-03 | 0.0 | 0.0 |
| Desorption | | | |
| H(s) +H(s) → $H_2$ +Pt(s) +Pt(s) | 3.700E+21 | 0.0 | 75.0 |
| \$H(s) | 0.0 | 0.0 | 15.0 |
| \$$C_2$(2s) | 0.0 | 0.0 | 15.0 |
| \$C(s) | 0.0 | 0.0 | 15.0 |
| $CH_3$(s) +H(s) → $CH_4$ +Pt(s) +Pt(s) | 1.000E+21 | 0.0 | 50.0 |
| \$H(s) | 0.0 | 0.0 | 15.0 |
| $C_2H_4$(2s) → $CH_4$ +C(s) | 1.000E+14 | 0.0 | 2.5 |
| \$C(s) | 0.0 | 0.0 | -47.5 |
| \$$C_2$(2s) | 0.0 | 0.0 | -47.5 |
| $C_2H_6$(2s) → Pt(s) +Pt(s) +$C_2H_6$ | 1.000E+16 | 0.0 | 20.9 |
| $C_2H_4(1s)$ → Pt(s) +$C_2H_4$ | 1.000E+14 | 0.0 | 50.2 |
| Surface reactions | | | |
| $CH_3$(s) +Pt(s) → $CH_2$(s) +H(s) | 1.262E+22 | 0.0 | 70.3 |
| $CH_2$(s) +H(s) → $CH_3$(s) +Pt(s) | 3.090E+22 | 0.0 | 0.0 |
| $CH_2$(s) +Pt(s) → CH(s) +H(s) | 7.314E+22 | 0.0 | 58.9 |
| \$C(s) | 0.0 | 0.0 | -50.0 |
| \$$C_2$(2s) | 0.0 | 0.0 | -50.0 |
| CH(s) +H(s) → $CH_2$(s) +Pt(s) | 3.090E+22 | 0.0 | 0.0 |
| CH(s) +Pt(s) → C(s) +H(s) | 3.090E+22 | 0.0 | 0.0 |
| C(s) +H(s) → CH(s) +Pt(s) | 1.248E+22 | 0.0 | 138.0 |
| $C_2H_4(1s)$ → $C_2H_4$(2s) | 1.000E+13 | 0.0 | 83.3 |
| $C_2H_4$(2s) → $C_2H_4(1s)$ | 1.000E+13 | 0.0 | 75.3 |
| $C_2H_5$(s) +H(s) → $C_2H_6$(2s) | 3.700E+22 | 0.0 | 41.8 |
| $C_2H_6$(2s) → $C_2H_5$(s) +H(s) | 1.000E+13 | 0.0 | 57.7 |
| $CH_3$(s) +$CH_3$(s) → $C_2H_6$(2s) | 1.000E+21 | 0.0 | 00.0 |
| $C_2H_6$(2s) → $CH_3$(s) +$CH_3$(s) | 1.000E+13 | 0.0 | 124.5 |
| $CH_2$(s) +$CH_3$(s) → $C_2H_5$(s) +Pt(s) | 1.370E+22 | 0.0 | 00.0 |
| $C_2H_5$(s) +Pt(s) → $CH_2$(s) +$CH_3$(s) | 1.370E+20 | 0.0 | 128.9 |
| $C_2H_5$(s) +Pt(s) → $C_2H_4$(2s)+H(s) | 1.370E+22 | 0.0 | 54.4 |
| $C_2H_4$(2s)+H(s) → $C_2H_5$(s) +Pt(s) | 1.370E+22 | 0.0 | 29.3 |
| $C_2H_5$(s) +Pt(s) +Pt(s) → $C_2H_4$(3s)+H(s) | 1.370E+22 | 0.0 | 16.7 |
| $C_2H_4$(3s)+H(s) → $C_2H_5$(s) +Pt(s) +Pt(s) | 1.370E+20 | 0.0 | 28.9 |
| $C_2H_4$(3s) → $C_2H_4$(2s)+Pt(s) | 1.000E+13 | 0.0 | 87.4 |
| $C_2H_4$(2s)+Pt(s) → $C_2H_4$(3s) | 1.370E+21 | 0.0 | 37.2 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $C_2H_4(2s)$ | +Pt(s) | → | $C_2H_3(1s)$ | +H(s) | 1.370E+20 | 0.0 | 99.1 |
| $C_2H_3(1s)$ | +H(s) | → | $C_2H_4(2s)$ | +Pt(s) | 1.370E+22 | 0.0 | 75.3 |
| $C_2H_3(1s)$ | +Pt(s) | → | $CH_3$(s) | +C(s) | 1.370E+22 | 0.0 | 46.9 |
| $C(s) | | | | | 0.0 | 0.0 | -50.0 |
| $C_2(2s) | | | | | 0.0 | 0.0 | -50.0 |
| $CH_3$(s) | +C(s) | → | $C_2H_3(1s)$ | +Pt(s) | 1.370E+22 | 0.0 | 46.0 |
| $C_2(2s)$ | +$C_2(2s)$ | → | $C_2(1s)$ | | 1.370E+21 | 0.0 | 220.0 |
| $C_2(2s) | | | | | 0.0 | 0.0 | 60.0 |
| C(s) | +C(s) | → | $C_2(2s)$ | +Pt(s) | 1.370E+21 | 0.0 | 180.0 |
| $C(s) | | | | | 0.0 | 0.0 | 50.0 |
| $C_2(2s)$ | +Pt(s) | → | C(s) | +C(s) | 1.370E+21 | 0.0 | 185.0 |
| $C(s) | | | | | 0.0 | 0.0 | 50.0 |
| $H_2$ | +C(s) | → | $CH_2$(s) | | 4.000E-02 | 0.0 | 29.7 |
| $C(s) | | | | | 0.0 | 0.0 | 4.6 |
| $C_2(2s) | | | | | 0.0 | 0.0 | 4.6 |
| $CH_2$(s) | | → | C(s) | +$H_2$ | 7.690E+13 | 0.0 | 25.1 |
| $C(s) | | | | | 0.0 | 0.0 | -50.0 |
| $C_2(2s) | | | | | 0.0 | 0.0 | -50.0 |

Table A.3.: Surface reaction mechanisms for catalyitc oxygen-free conversion of methane to ethane. [61]

# Bibliography


[1] E. Anderson, Z. Bai, C. Bischof, J. W. Demmel, J. J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *Lapack Users' Guide*. SIAM Philadelphia, 2nd edition edition, 1995.

[2] U. M. Ascher, R. M. M. Mattheij, and R. D. Russel. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. SIAM, 1995.

[3] Uri Ascher. Collocation for two-point boundary value problems revisited. *SIAM J. Numer. Anal.*, 23(3):596–609, 1986.

[4] Uri M. Ascher and Raymond J.Spiteri. Collocation software for boundary value differential-algebaraic equations. *SIAM J. Sci. Comput.*, 15(4):938–952, 1994.

[5] F. Behrendt. *Simulation laminarer Gegenstromdiffusionsflammen unter Verwendung detaillierter Reaktionsmechanismen*. PhD thesis, Ruprecht-Karls-Universität, Heidelberg, 1989.

[6] F. Behrendt, O. Deutschmann, and J. Warnatz. Simulation and sensitivity analysis of heterogenous oxidation of methane on a platin foil. *J. Vac. Sci. Technol. A*, 13(3):1373–1377, 1995.

[7] Lorenz T. Biegler, Jorge Nocedal, and Claudia Schmid. A reduced hessian method for large-scale constrained optimization. *SIAM J. Optimization*, 5(2):314–347, 1995.

[8] C. H. Bischof, A. Carle, P. M. Khademi, and A. Maurer. The ADIFOR 2.0 system for the automatic differentiation of Fortran 77 programs. Technical Report CRPC-TR94491, Center for Research on Parallel Computation. Rice University, 1994.

[9] C. H. Bischof, A. Carle, P. M. Khademi, and A. Maurer. ADIFOR 2.0: Automatic differentiation of Fortran 77 programs. *IEEE Computational Science and Engineering*, 3(3):18–32, 1996.

[10] C. H. Bischof, A. Carle, P. M. Khademi, A. Maurer, and P. Hovland. The ADIFOR 2.0 user's guide. Technical Report No. 192, Mathematics and Computer Science Division. Rice University, 1995.

[11] H. G. Bock. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*. Bonner Mathematische Schriften 183. University of Bonn, 1987.

[12] H. G. Bock and K. J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proccedings of the 9th IFAC World Congress, Budapest*. Pergamon Press, 1984.

[13] A. S. Bodke and L. D. Schmidt. The effect of ceramic supports on partial oxidation of hydrocarbons over noble metal coated monoliths. *J. Catal.*, 179:138, 1998.

[14] Paul T. Boggs and John W. Tolle. Sequential quadratic programming. In *Acta Numerica*, pages 1–51. 1995.

[15] J. E. Dennis, JR. and Jorge J. Moré. Quasi-Newton methods, motivation and theory. *SIAM Review*, 1977.

[16] P. Deuflhard, E. Hairer, and J. Zugck. One-step and extrapolation methods for differential-algebraic systems. *Numerische Mathematik*, 51:501–516, 1987.

[17] O. Deutschmann. *Modellierung von Reaktionen an Oberflächen und deren Kopplung mit chemisch reagierenden Strömungen.* PhD thesis, Ruprecht-Karls-Universität, Heidelberg, 1996.

[18] O. Deutschmann, F. Behrendt, and J. Warnatz. Modelling and simulation of heterogenous oxidation of methane on a platinum foil. *Catal. Today*, 21:461–470, 1994.

[19] O. Deutschmann, C. Correa, S. Tischer, D. Chatterjee, and J. Warnatz. *DETCHEM, User manual, Version 1.4.1.*

[20] O. Deutschmann, R. Schmidt, F. Behrendt, and J. Warnatz. Numerical modeling of catalytic ignition. *Proc. Combust. Inst.*, 26:1747–1754, 1996.

[21] M. Diehl, H. G. Bock, J. P. Schloeder, R. Findeisen, Z. Nagy, and F. Allgoewer. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *J. Proc. Contr.*, 2001.

[22] M. Diehl, I. Uslu, R. Findeisen, S. Schwarzkopf, F. Allgöwer, H. G. Bock, T. Bürner, E. D. Gilles, A. Kienle, J. P. Schlöder, and E. Stein. Real-time optimization of large scale process models: Nonlinear model predictive control of a high purity distillation column. In M. Groetschel, S. O. Krumke, and J. Rambau, editors, *Online Optimization of Large Scale Systems: State of the Art.* Springer, 2001.

[23] Angelika Dieses. Numerical methods for optimization problems in water flow and reactive solute transport processes of xenobiotics in soils. Technical Report SFB Preprint 2001-07, University of Heidelberg, 2001. Ph.D. Thesis.

[24] Wolfgang Egartner. Working range optimization for turbine and compressor blading. *Journal for Computational and Applied Mathematics*, 120(1–2):59–65, 2000.

[25] Wolfgang Egartner and Volker H. Schulz. Partially reduced SQP methods for optimal turbine and compressor blade design. In *ENUMATH 97, Proceedings of the 2nd European Conference on Numerical Mathematics and Advanced Applications.* World Scientific Publishing, November 1998.

[26] G. Evans and R. Greif. A numerical model of the flow and heat transfer in a rotating disk cemical vapor deposition reactor. *Transactions of the ASME, Journal of Heat Transfer*, 109:928–935, 1987.

[27] Daniel Gabay. Minimizing a differentiable function over a differential manifold. *Journal of Optimization Theory and Applications*, 37(2):177–219, 1982.

[28] Daniel Gabay. Reduced quasi-newton methods with feasibility improvement for nonlinearly constrained optimization. *Mathematical Programming Study*, 16:18–44, 1982.

[29] Daniel Gabay and David G. Luenberger. Efficiently converging minimization methods based on the reduced gradient. *SIAM J. Control and Optimization*, 14(1):42–61, 1976.

[30] Chaya Bleich Gurwitz and Michael L. Overton. Sequential quadratic programming methods based on approximating a projected hessian matrix. *SIAM J. Sci. Stat. Comput.*, 10(4):631–653, 1989.

[31] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and differential algebraic problems.* Springer, Berlin-Heidelberg-New York, 2nd edition, 1996.

[32] Matthias Heinkenschloss. Projected sequential quadratic programming methods. *SIAM J. Optimization*, 6(2):373–417, 1996.

[33] D. A. Hickman and L. D. Schmidt. Steps in $CH_4$ oxidation on Pt and Rh surfaces: High-temperature reactor simulations. *AIChE J.*, 39:1164, 1993.

[34] Robert J. Kee, James A. Miller, Gregory H. Evans, and Graham Dixon-Lewis. A computational model of the structure and extinction of strained, opposed flow, premixed methane-air flames. In *Twenty-Second Symposium (International) on Combustion*, pages 1479–1494, 1988.

[35] F.-S. Kupfer. An infinite-dimensional convergence theory for reduced SQP methods in hilbert space. *SIAM J. Optimization*, 6(1):126–163, 1996.

[36] D.B. Leineweber. Analyse und Restrukturierung eines Verfahrens zur direkten Lösung von Optimal-Steuerungsproblemen. Master's thesis, Ruprecht-Karls-Universität, 1995.

[37] D.B. Leineweber. *Efficient Reduced SQP Methods for the Optimization of Chemical Processes Described by Large Sparse DAE Models*. VDI, Düsseldorf, 1999.

[38] U. Maas. *Mathematische Modellierung instationärer Verbrennungsprozesse unter Verwendung detaillierter Reaktionsmechanismen*. PhD thesis, Ruprecht-Karls-Universität, Heidelberg, 1989.

[39] L. I. Maier, O. Deutschmann, and J. Warnatz. Modeling the partial oxidation of ethylene on silver. *to appear*, 2001.

[40] Linda Petzold, J. Ben Rosen, Philip E. Gill, Laurent O. Jay, and Kihong Park. Numerical optimal control of parabolic pdes using dasopt. In *Proc. IMA Workshop on Large-Scale Optimization*, 1996.

[41] Linda Petzold, Radu Serban, Shengtai Li, Soumyendu Raha, and Andrew Strelzoff. A problem solving environment for dynamic optimization of partial differential-algebraic equation systems. In *16th IMACS World Congress, Lausanne, Switzerland*, 2000.

[42] Laxminarayan L. Raja, Robert J. Kee, and Linda R. Petzold. Simulation of the transient, compressible, gas-dynamic, behavior od catalytic-combustion ignition in stagnation flows. In *Twenty-Seventh Symposium (International) on Combustion*, Boulder, 1998.

[43] Laxminarayan L. Raja, Robert J. Kee, Radu Serban, and Linda R. Petzold. Dynamic optimization of chemically reacting stagnation flows.

[44] Laxminarayan L. Raja, Robert J. Kee, Radu Serban, and Linda R. Petzold. Computational algorithm for dynamic optimization of chemical vapor deposition processes in stagnation flow reactors. *Journal of the Electrochemical Society*, 147(7):2718–2726, 2000.

[45] Werner C. Rheinboldt. Differential-algebraic systems as differential equations on manifolds. *Mathematics of Computation*, 43(168):473–482, 1984.

[46] G. Rücker. Automatic differentiation with applications in optimization of chemical reaction systems. Master's thesis, Ruprecht-Karls-Universität, 1999.

[47] J. P. Schlöder. *Numerische Methoden zur Behandlung hochdimensionaler Aufgaben der Parameteridentifizierung.* Bonner Mathematische Schriften 187. University of Bonn, 1988.

[48] V. Schulz. *Reduced SQP Methods for Large Scale Optimal Control Problems in DAE with Application to Path Planning for Satellite Mounted Robots.* PhD thesis, Ruprecht-Karls-Universität, Heidelberg, 1996.

[49] V. Schulz. Solving discretized optimization problems by partially reduced SQP methods. *Computing and Visualization in Science*, 1:83–96, 1998.

[50] V. H. Schulz. Numerical optimization of the cross-sectional shape of turbine blades. In *Proceedings of ICIAM 95, Special Issue of ZAMM*, Berlin, 1995. Akademie-Verlag.

[51] V. H. Schulz. Optimal paths for satellite mounted robots. In *Proceedings of ICIAM 95, Special Issue of ZAMM*, Berlin, 1995. Akademie-Verlag.

[52] V. H. Schulz, H. G. Bock, and R. W. Longman. Optimal path planning for satellite mounted robot manipulators. In *Proceedings of the 1993 AAS/AIAA Spaceflight Mechanics Meeting*, volume 73 of *Advances in Astronautical Sciences*. American Astronautical Society, 1993.

[53] V. H. Schulz, H. G. Bock, and R. W. Longman. Shortest paths for satellite mounted robot manipulators. In D. Kraft R.Bulirsch, editor, *Computational Optimal Control*, pages 357–366. Birkhäuser, Basel-Boston-Berlin, 1994.

[54] V. H. Schulz, Th. Dreyer, Th. Speer, and H. G. Bock. Optimum shape design of turbine blades. In P. Kleinschmidt, A. Bachem, U. Derigs, D. Fischer, U. Leopold-Wildburger, and R. Möhring, editors, *Operations Research Proceedings 1995*, pages 190–195, Heidelberg, 1996. Springer.

[55] Radu Serban and Linda R. Petzold. Coopt - a software package for optimal control of large-scale differential-algebraic equation systems. *to appear, J. Math. and Computers in Simulation*, 2000.

[56] M. von Schwerin. *Numerische Methoden zur Schätzung der Reaktionsgeschwindigkeiten bei der katalytischen Methankonversion und zur Optimierung von Essigsäure- und Methanprozessen*. PhD thesis, Ruprecht-Karls-Universität, Heidelberg, 1998.

[57] Marianne von Schwerin, Olaf Deutschmann, and Volker Schulz. Process optimization of reactive systems by partially reduced SQP methods. *Computers and Chemical Engineering*, 24:89–97, 2000.

[58] Oskar von Stryk. Numerical solution of optimal control problems by direct collocation. In R. Bulirsch, A. Miele, J. Stoer, and K.-H. Wells, editors, *Optimal Control—Calculus of Variations, Optimal Control Theory and Numerical Methods*, volume 111 of *International Series of Numerical Mathematics*, pages 129–143, 1993.

[59] J. Warnatz, U. Maas, and R.W. Dibble. *Combustion*. Springer, Berlin-Heidelberg, 2nd edition, 1999.

[60] M. Wolf. *Simulation der katalytischen Methankonversion an Platin*. PhD thesis, Ruprecht-Karls-Universität, Heidelberg, 2000.

[61] M. Wolf, O. Deutschmann, F. Behrendt, and J. Warnatz. Kinetic model of an oxygen-free methane conversion on a platinum catalyst. *Catal. Lett.*, 61, 1999.